



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA DE MINAS

**APLICACIÓN DE DEEP LEARNING COMO COMPLEMENTO AL
MODELAMIENTO FENOMENOLÓGICO DEL PROCESO DE FLOTACIÓN A
ESCALA INDUSTRIAL**

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN MINERÍA

CARLOS ANDRÉS YANTÉN VERGARA

PROFESOR GUÍA:
Willy Kracht Gajardo

MIEMBROS DE LA COMISIÓN:
Julian Ortiz Cabrera
Sebastian Ávalos Sotomayor
Gonzalo Díaz Rodríguez

SANTIAGO DE CHILE
2022

APLICACIÓN DE DEEP LEARNING COMO COMPLEMENTO AL MODELAMIENTO FENOMENOLÓGICO DEL PROCESO DE FLOTACIÓN A ESCALA INDUSTRIAL

Se ha propuesto durante el desarrollo de esta tesis hacer uso de algoritmos de Deep Learning, con el fin de apoyar el modelamiento fenomenológico del proceso de flotación. El principal apoyo viene de la mano del método de predicción denominado redes neuronales. El objetivo principal de esta tesis es generar un modelo de predicción de variables de flotación que combine el uso de algoritmos de Deep Learning con métodos fenomenológicos.

Se identificaron tres posibles puntos de aplicación para estos algoritmos: (1) El cálculo de las distribuciones de tiempos de residencias, (2) la predicción del valor de la ley de concentrado en función exclusiva de la ley de alimentación, y (3) la identificación y clasificación de Unidades Geometalúrgicas, para definir qué variables cinéticas son adecuadas usar en cada predicción. Un modelo global que combine estos tres puntos de aplicación podría recibir un set de leyes elementales o minerales y en función de eso, identificar la unidad geometalúrgica y asociar las constantes cinéticas correspondientes. Luego, podría derivar a un modelo más específico para la predicción de la ley de concentrado.

Para el primer punto de aplicación, se estudió la predicción de distribuciones de tiempos de residencia. Se hicieron pruebas de predicción con redes neuronales y con el método de estimación de densidad con Kernel. Ambos métodos alcanzaron valores de precisión importantes con respecto a la base de datos utilizada. Las redes neuronales específicamente tuvieron grados de precisión que involucraron una divergencia de Kullback Liebler de 10^{-4} , pero son mucho menos flexibles, y pierden capacidad de predicción cuando se sale de su dominio de entrenamiento. Por otro lado, la aplicación de la estimación de densidad con Kernel genera predicciones con divergencia de Kullback Liebler de 24×10^{-4} , pero que son más flexibles, dado que no requieren una base de entrenamiento.

Para el segundo punto de aplicación, se estudió la utilización de redes neuronales frente al uso de modelos de regresión simples. Se pudo ver que ambos métodos presentaban resultados muy similares en cuanto a precisión y tiempo de cálculo. Esto se debe probablemente al hecho que no se tuvieron los datos necesarios para hacer predicción de acuerdo a diferentes unidades metalúrgicas, situación que podría haber diferenciado el comportamiento de las leyes de concentrado frente a una misma ley de alimentación.

Finalmente, para el tercer punto, se hicieron pruebas con métodos de clasificación directa, los cuales presentaron un rendimiento adecuado pero mejorable. Frente a esto, se probó la alternativa de implementar un algoritmo de clasificación en base a un árbol binario, el cual si presento una mejora con respecto al primero, yendo desde una precisión en la clasificación 74.4 % a 76.7 %. También se probó la posibilidad de usar un método de clasificación difusa o probabilística, mediante la función de activación softmax, el cual presento un rendimiento ligeramente superior, de aproximadamente 78 %.

*And then one day you find
Ten years have got behind you
No one told you when to run
You missed the starting gun*

Time - Pink Floyd

Agradecimientos

Quería iniciar agradeciendo a mi familia por todo el apoyo brindado durante este proceso. Especiales agradecimientos a mi mamá, mi papá, mi hermana Lisette y mi cuñado Jonathan por la paciencia estos últimos meses.

A los profesores que prestaron todo el apoyo y atención durante el desarrollo de esta tesis. Willy Kracht, Gonzalo Díaz y Sebastian Ávalos, gracias por el tiempo y la dedicación de enseñarme todo lo que he aprendido durante este proceso.

Un especial agradecimiento a las secretarias docentes y coordinadoras de minas, Carol Tapia y Gloria Jiménez. Gracias por el apoyo, la paciencia y la dedicación entregadas en este proceso. Gracias por estar ahí, apoyar y no dejarme bajar los brazos cuando las cosas se pusieron cuesta arriba.

A todo el equipo de Alges, por el recibimiento, el apoyo constante y la comprensión entregada en estos meses.

Por último, Madeline Valdivia, muchas gracias por el apoyo dado durante el desarrollo de esta tesis, el tiempo que me brindaste para ayudarme a organizar las ideas, y la contención entregada cuando el camino se puso difícil.

Tabla de Contenido

1. Introducción	1
1.1. Objetivo general	3
1.2. Objetivos específicos	3
1.3. Alcances	3
2. Marco Teórico	4
2.1. Flotación de Minerales	4
2.1.1. Introducción a la flotación	4
2.1.2. Contexto operacional	5
2.1.3. Pruebas de flotación Batch	8
2.2. Modelamiento	9
2.2.1. Modelamiento fenomenológico de flotación	10
2.2.1.1. Modelo de flotación de primer orden en flotación Batch	10
2.2.1.2. Modelo de flotación de primer orden en flotación continua	12
2.2.1.2.1. Distribución de tiempos de residencia	13
2.2.1.2.2. Distribución de constantes cinéticas	16
2.2.1.2.3. Estado del arte en modelos de flotación	18
2.2.2. Modelos caja negra de simulación	19
2.2.2.1. Clasificación de modelos de caja negra según su función	20
2.2.2.1.1. Modelo de Regresión	20
2.2.2.1.2. Modelo de clasificación	21
2.2.2.2. Redes Neuronales	21
2.2.2.2.1. Definición y creación	21
2.2.2.2.2. Backpropagation	23
2.2.2.3. Estimación de densidad con Kernel	24
2.3. Definiciones Matemáticas relevantes en redes neuronales	25
2.3.1. Funciones de Activación	25
2.3.1.1. Función de Activación tipo sigmoide	26
2.3.1.2. Función de Activación tipo ReLU y Leaky ReLU	27
2.3.1.3. Función de activación tipo tangente hiperbólica	28
2.3.1.4. Función de activación tipo softmax	29
2.3.2. Funciones de Perdida	29
2.3.2.1. Función de pérdida de error cuadrático medio y derivadas	29
2.3.2.2. Función de Pérdida de Kullback Leibler	30
2.3.2.3. Función de Pérdida Binary Cross Entropy	30
2.4. Programacion de modelos de caja negra	31
2.4.1. Construcción de una red neuronal	31

2.4.1.1.	Programación de Red en Keras	32
2.4.1.2.	Optimización de Red	33
2.4.2.	Algoritmos auxiliares en el uso de redes neuronales	33
2.4.2.1.	Normalización de datos	33
2.4.2.2.	Aplicación de vectores de latencia	34
2.5.	Definición y caracterización de UGMs	35
3.	Metodología	37
3.1.	Desarrollo del modelo fenomenológico	37
3.1.1.	Definición de variables pertinentes	37
3.1.2.	Calculo de DTR por método de Montecarlo	38
3.1.3.	Discretización del modelo de recuperación	39
3.1.4.	Cálculo de recuperación en un banco	40
3.2.	Análisis de método de implementación	43
3.3.	Modelos de caja negra	44
3.3.1.	Modelos aplicados a calculo de DTRs	44
3.3.1.1.	Motivación de uso	44
3.3.1.2.	Base de datos utilizada	45
3.3.1.2.1	Base de datos original	45
3.3.1.2.2	Base de datos modificada	46
3.3.1.3.	Implementación	47
3.3.1.3.1	Redes Neuronales	47
3.3.1.3.2	Estimación de Densidad con Kernel	49
3.3.2.	Modelos aplicados a predicción de ley de concentrado	50
3.3.2.1.	Motivación de uso	51
3.3.2.2.	Base de datos utilizada	51
3.3.2.3.	Implementación	53
3.3.3.	Modelos aplicados a identificación de UGMs	53
3.3.3.1.	Motivación de uso	53
3.3.3.2.	Base de datos utilizada	54
3.3.3.3.	Implementación	55
3.3.3.3.1	Clasificación directa	55
3.3.3.3.2	Clasificación por árbol binario	56
3.3.3.3.3	Clasificación probabilística	56
4.	Resultados	58
4.1.	Modelo fenomenológico puro	58
4.2.	Modelos aplicados a cálculo de DTRs	60
4.3.	Predicción de leyes de concentrado	64
4.4.	Identificación de Unidad Geometalúrgica	65
5.	Análisis de resultados	69
5.1.	Predicciones de distribuciones de tiempos de residencia	71
5.2.	Predicciones de ley de concentrado	74
5.3.	Clasificación de unidades geometalúrgicas	74
5.4.	Propuesta de un modelo de predicción de recuperación	76
6.	Conclusiones	77

Bibliografía

79

Anexo

82

Índice de Tablas

3.1.	Estadísticas de datos para modelo de leyes de Concentrado	51
3.2.	Estadísticas Generales de categorías de UGMs	55
4.1.	Estructuras de modelos de predicción de DTR según etapa en estudio	60
4.2.	Resultados de precisión en estudio de predicción de DTRs con redes neuronales	60
4.3.	Estructuras usadas en estudio de estimación de densidad con Kernel para pre- dicción de DTRs	61
4.4.	Resultados de precisión en estudio de predicción de DTRs con estimación de densidad con Kernel	61
4.5.	Modelos de regresión para predicción de leyes de concentrado	64
4.6.	Modelos de redes neuronales para predicción de leyes de concentrado	65
4.7.	Estructura de Redes neuronales para clasificación directa de UGMs	66
4.8.	Resultados de clasificación directa de UGMs	66
4.9.	Estructura de Redes neuronales para clasificación de UGMs con árbol binario .	66
4.10.	Resultados de clasificación de UGMs por árbol binario	67
4.11.	Estructura de Redes neuronales para clasificación de UGMs con método proba- bilístico	67
4.12.	Resultados de clasificación de UGMs por método difuso	68
4.13.	Matrices de confusión por método	68

Índice de Ilustraciones

2.1.	Circuito de flotación estándar	7
2.2.	Ejemplo de Curva de recuperación Acumulada	9
2.3.	Comparación entre datos reales y curva de recuperación modelada	12
2.4.	Distribuciones de tiempo de residencia con tiempo de residencia medio variable	15
2.5.	Distribuciones de tiempos de residencia con número de celdas variable	16
2.6.	Representaciones gráfica de la distribución de constantes cinéticas de Klimpel y Garcia Zuñiga	17
2.7.	Esquema del funcionamiento de una red neuronal	22
2.8.	Esquematización de labor de cálculo de una neurona	23
2.9.	Gráfico de función sigmoide	26
2.10.	Grafico de Función ReLU	27
2.11.	Grafico de función Tangente hiperbólica	28
2.12.	Esquema de algoritmo de normalización	34
3.1.	Esquema de funcionamiento de Montecarlo para generación de DTRs	39
3.2.	Esquema para calculo de recuperación en una celda	42
3.3.	Histograma de leyes de alimentación	52
3.4.	Histograma de leyes de concentrado	52
4.1.	Comparación de DTRS según número de datos	58
4.2.	Aumento de tiempo de calculo de DTR frente a número de datos aleatorios . .	59
4.3.	Aumento de tiempo de calculo de recuperación frente a número de datos aleatorios	59
4.4.	Comparación de distribuciones de tiempos de residencia con tiempo de residencia medio dentro del rango de entrenamiento	62
4.5.	Comparación de distribuciones de tiempos de residencia con tiempo de residencia medio fuera del rango de entrenamiento	62
4.6.	Curva de recuperación con flujo dentro de limites de entrenamiento	63
4.7.	Curva de recuperación con flujo fuera de limites de entrenamiento	63
4.8.	Comparación de tiempos de calculo de recuperación según número de celdas en banco y método	64
4.9.	Comparacion de densidades entre datos de leyes de concentrado	65
5.1.	Esquema de funcionamiento de modelo creado	76

Capítulo 1

Introducción

La industria minera es una de las más importantes a nivel nacional. Para el año 2021, el porcentaje que aportó al PIB fue de un 10 % [1]. La minería, ya sea desde el sector público o privado, es una importante fuente de empleos, especialmente en el sector zona norte del país, donde hasta un 17 % de la población se dedica a la industria minera o a un área asociada a ella.

El proceso de obtención de metales en forma general puede dividirse en dos etapas principales: Minería y Metalurgia. La primera consiste en todos aquellos procesos enfocados en planificar, extraer y transportar minerales desde la tierra. Una vez obtenidos estos minerales, vienen las etapas de metalurgia, las cuales consisten en procesar y alterar estos minerales, tanto física como químicamente, con el fin de separar aquellos metales específicos que se buscan.

En la minería del cobre, los minerales que se tratan con mayor regularidad son los llamados sulfuros, es decir, minerales con azufre como uno de sus compuestos principales. La metalurgia de sulfuros de cobre posee un proceso bien establecido, que consiste en las etapas de chancado, molienda, flotación, fundición y electrorefinación, además de procesos auxiliares como lo pueden ser el espesamiento y el filtrado [2]. Cada uno de estos procesos recibe el producto de la etapa anterior, por lo que una modificación en las actividades más tempranas tiene repercusiones en todas las etapas subsiguientes. Estos cambios que se pueden producir pueden afectar negativamente la eficiencia y producción de la planta, por lo que se hace necesario tener un método para predecir el comportamiento que tendrán los distintos procesos frente a cambios en sus variables de entrada.

Con esto en mente, nace el área del modelamiento y simulación de procesos. El modelamiento consiste en describir la relación de las variables de entrada y salida de un proceso por medio de diversas herramientas matemáticas. Una vez que se tenga un modelo planteado, se ingresan las variables de entrada y se evalúan los parámetros de salida como una predicción de lo que sucedería en una situación real. A esto se le conoce como simulación de procesos [3].

Existen diversos tipos de modelos que se diferencian en la manera en que tratan los datos de ingreso para calcular los datos de salida. Estos se pueden categorizar en tres tipos principales: (1) Modelo fenomenológico, (2) Modelo de Caja Negra y (3) Modelo de Caja gris. Un modelo fenomenológico es aquel que busca describir la relación entre variables de entrada y salida por medio de ecuaciones basadas en fenómenos físicos y/o químicos reales que

se pueden medir por experimentación. Un modelo de caja negra es aquel que utiliza algún tipo de inteligencia artificial con el fin de encontrar relaciones matemáticas directas entre los parámetros de entrada y los parámetros de salida, sin que exista la descripción de algún fenómeno en específico, sino que la relación encontrada sea meramente empírica. Por último, existen los modelos de caja gris, que mezclan las características de ambos tipos descritos anteriormente. Por lo general, un modelo de caja gris tiene una base fenomenológica a la que se le hacen ajustes por medio de un modelo de caja negra para poder acercarlo lo más posible a la realidad.

Esta tesis se enfoca en estudiar el modelamiento del proceso de flotación. Este proceso es aquel que separa los minerales que tienen valor, conocidos como minerales de mena, de aquellos que no lo tienen, conocidos como minerales de ganga. El proceso de flotación ocurre en un equipo denominado celda de flotación, que es una cámara de gran tamaño descubierta en su parte superior, donde la pulpa, que es la mezcla de minerales disminuidos en tamaño y agua, es ingresada. Este equipo cuenta con un sistema de inyección de aire, que se introduce en forma de burbujas, a las cuales, las partículas de cobre se adhieren debido a sus propiedades químicas. En la superficie de la celda afloran todas las burbujas, acumulándose en una zona denominada área de espuma. La espuma se desborda en la celda, y al ser limpiada y secada se convierte en concentrado, que es el producto que se envía a fundición. Por otro lado, aquel mineral que no ha flotado se desecha por la parte inferior de la celda.

El modelamiento de la flotación se enfoca principalmente en la utilización de modelos fenomenológicos, los cuales dependen de parámetros como la alimentación de la celda, el flujo de aire y la ley de mineral que ingresa en la misma. Existen ecuaciones ya desarrolladas que permiten predecir algunas de las variables operacionales más importantes relacionadas con el proceso. Sin embargo, estos modelos pueden presentar deficiencias por falta de información o por cálculos que pueden resultar muy engorrosos.

Para el desarrollo de esta tesis se plantea como objetivo generar un modelo de caja gris para poder simular estados transitorios en las variables principales del proceso de flotación. Para esto, se debe generar, en primer lugar, un modelo fenomenológico lo más cercano posible a la realidad y con los inputs necesarios que tengan injerencia en el proceso. Luego, es necesario investigar e idear un plan para complementar esto con un modelo de caja negra, con el fin de ajustar los datos para hacerlos más cercanos a los datos reales. El principal método caja negra que se ha elegido se denomina redes neuronales.

Las redes neuronales son un método de regresión y predicción de datos, que se basa en el procesamiento de un set de variables de entrada o inputs en una serie de unidades denominadas neuronas. Estas realizan una serie de diferentes operaciones matemáticas, para obtener un valor de salida u output, el que corresponde a la predicción realizada. Las neuronas pueden ser entrenadas automáticamente por diversos métodos, con el objetivo de tener un modelo lo más representativo posible de la realidad.

Considerando el contexto anteriormente descrito, se plantea como hipótesis que un modelo de caja gris, que utiliza de forma combinada los beneficios entregados de un modelo fenomenológico y los ajustes de un modelo de redes neuronales, tiene un mejor rendimiento que un modelo que sea puramente fenomenológico. Esto considerando tanto la precisión a la

hora de representar la realidad, como el tiempo de cálculo que se tome en realizar todas las operaciones necesarias.

1.1. Objetivo general

Diseñar un modelo caja gris, utilizando base fenomenología y soporte con redes neuronales u otros algoritmos de *Machine Learning* y *Deep Learning*, para simulación de variables de flotación.

1.2. Objetivos específicos

1. Analizar modelos fenomenológicos utilizados en la actualidad, identificando y seleccionando los más adecuados.
2. Estudiar los modelos de caja negra usados en aplicaciones industriales en la actualidad, haciendo énfasis en modelos basados en redes neuronales profundas.
3. Identificar cómo se combinan ambos tipos de modelos con el objetivo de generar un modelo caja gris.
4. Estudiar algoritmos de clasificación basados en metodologías de *Deep Learning* para ser utilizados en clasificación de unidades geometalúrgicas.
5. Generar y testear un prototipo de modelo caja gris de modelación de flotación.

1.3. Alcances

1. El trabajo de simulación se ha dividido entre el uso de Matlab para generar y tratar sets de datos, y Python, para la etapa de modelamiento. También se han utilizado herramientas de simulación, con el fin de simular los estados transitorios en un ambiente continuo.
2. El modelo se ha generado con el fin de simular el proceso de flotación de la manera más fehaciente posible desde un punto de vista académico e investigativo. Este modelo, si bien puede ser usado en ámbitos directamente industriales, como predicción de procesos y control de estos, no está optimizado para ello. Si se desea usar con ese fin, se deben aplicar ciertos cambios para adecuarlo a la situación correspondiente.
3. La etapa de entrenamiento de las neuronas en redes neuronales se ha realizado con datos reales provenientes de plantas de flotación, por lo que el ajuste que presenta el modelo deriva de esos datos específicos. Si se desea probar este modelo en otra planta de flotación, resultaría conveniente contar con datos de esa planta en específico para hacer la etapa de entrenamiento.

Capítulo 2

Marco Teórico

2.1. Flotación de Minerales

2.1.1. Introducción a la flotación

La concentración de minerales es una de las etapas del proceso minero. Esta consiste en separar selectivamente los minerales de interés, también conocidos como minerales de mena, de aquellos que no tiene valor económico para el negocio minero, también conocidos como minerales de ganga. La separación se hace en función de alguna característica, física o química, específica, en que ambos tipos de minerales difieran, y que sea capaz de ser aprovechada por algún mecanismo o sistema [2].

En la actualidad existe una gran variedad de métodos de separación que se usan para concentrar alguna especie en específico. Algunos se basan en la diferencia de densidad existente entre los minerales, como lo pueden ser la separación por medios densos o la decantación. Sin embargo, de entre todos los métodos de concentración, en la minería del cobre hay uno que destaca por sobre los demás, que es la flotación de minerales. Este método se basa en separar los minerales de acuerdo con su hidrofobicidad, es decir, de acuerdo con su capacidad de interactuar con el agua o de repelerla [4].

Esta se hace en equipos denominados celdas de flotación, las cuales se alimentan con pulpa mineralúrgica, que es la mezcla de mineral previamente disminuido en tamaño más agua y reactivos, a la que se le inyecta aire en forma de burbujas. El aire es ingresado por la parte inferior de la celda, y agitado para poder producir burbujas, las cuales, por diferencia de densidad con el líquido, tenderán a subir a la superficie de la celda. El mecanismo principal de separación se produce cuando las partículas que son hidrofóbicas tienden a adherirse a la superficie de las burbujas y ascender junto a estas, mientras que aquellas que son hidrofílicas tienden a quedarse en el agua y ser descargada junto a esta [4].

Del proceso de flotación se obtienen dos productos, que son el concentrado y el relave. El primero se retira por la parte superior de la celda, y está compuesto en su mayoría por aquellas partículas que se adhirieron a las burbujas, y que, por ende, tienen propiedades hidrofóbicas. En una etapa de flotación normal, se busca que sea el mineral más valioso y, por ende, que se quiere concentrar, aquel que flote.

Por otro lado, el relave es todo aquel material que debido a sus propiedades no se adhirió a las burbujas cuando estuvo en contacto con ellas, o que no tuvo la oportunidad de estar en contacto a ellas. Este material es retirado por la parte inferior de la celda, y en su mayoría, está compuesto por mineral de ganga, aunque también suele incluir partículas de mena que no pudieron llegar a la zona de concentrado [4].

En una celda, para que ocurra el proceso de flotación, es necesario que coexistan dos zonas bien definidas [4]:

1. Zona de Colección: Es el lugar donde se produce la interacción de las partículas con las burbujas y ascenso de estas hasta la zona de espuma. Esta ocupa el mayor volumen de la celda y en ella se encuentran fases en los tres estados: Solido como mineral, liquido como agua de procesos y gases. Por la zona inferior se produce la descarga del relave.
2. Zona de espuma: Esta se encuentra en la parte superior de la celda. Es donde se acumula todo el mineral que se adhirió a las burbujas en la zona de colección. Como su nombre lo indica, esta zona se caracteriza por contar con la presencia de una espuma que contiene la carga de mineral, la cual es retirada por desborde de la celda, para formar el producto que se denomina concentrado.

Para que ocurra el proceso de flotación, es necesario contar con la ayuda de agentes químicos que faciliten las condiciones para que el proceso sea posible. Estos reactivos se pueden dividir en tres categorías principales [5]:

1. Colectores: Son los agentes que entregan las propiedades hidrofóbicas a las partículas de mineral de interés, por lo que son la clave para el proceso de flotación. Suelen ser compuestos de carácter heteropolar y funcionan por medio de reacciones de adsorción, es decir, enlazamiento químico superficial del mineral con la parte polar del reactivo. Por otro lado, el resto de estructura polar queda en contacto directo con las moléculas de agua, que al ser apolares, tienden a rechazar el conjunto colector-partícula.
2. Espumantes: Son los agentes que se encargan de generar las condiciones adecuadas en cuanto al aire y las burbujas presentes dentro del sistema. Algunas de sus funciones son dar estabilidad y resistencia a la espuma superficial, y evitar la coalescencia de las burbujas en el interior del equipo, para contar con mayor área expuesta.
3. Modificadores: Son los agentes encargados de generar un ambiente físico químico adecuado para que ocurra el proceso de flotación. Un ejemplo esencial es la cal, la cual permite regular el pH de la pulpa dentro del equipo, la cual debe mantenerse en intervalos básicos. Otros ejemplos son los depresantes, los cuales se encargan de otorgarles propiedades hidrofílicas a minerales que no se desean flotar, lo que facilita su eliminación por el relave.

2.1.2. Contexto operacional

Previo a referirse al contexto operacional de la flotación, existen un par de conceptos que deben ser explicados, que son “ley” y “recuperación”. La ley de un elemento o mineral se define como el porcentaje existente de esa especie en específico en una masa determinada o en un flujo conocido. Por otro lado, la recuperación se define como la masa de la especie que se

obtuvo al final de proceso dividida por la masa total de la especie que fue ingresada al proceso.

Un esquema de flotación típico cuenta, en primer lugar, con un tanque de mezcla que se alimenta con la pulpa proveniente desde las etapas de conminución, donde se le añade agua y los primeros reactivos. Estos se añaden en este punto puesto que la reacción de adsorción no es inmediata, sino que lleva tiempo en realizarse, por lo que el tiempo de residencia que tiene la pulpa dentro del estanque debe ser el suficiente como para que el colector logre adsorberse superficialmente [6].

Luego, la pulpa es transportada a las celdas de flotación. Lo más común es que no haya una sola celda, sino que haya varias de estas y que estén organizadas en un circuito. Este circuito está diseñado con el fin de poder maximizar la recuperación de cobre en la planta y minimizar la recuperación de ganga en forma global, obteniéndose como producto final un concentrado de alta ley que ha recuperado la mayor cantidad de mineral que se ha ingresado en el circuito.

Para lograr esto, se cuenta con dos tipos principales de celdas de flotación: Las celdas convencionales y las celdas columnares. Las celdas convencionales están diseñadas con el fin de maximizar la recuperación de cobre, en desmedro de la ley del concentrado que producen. Estas celdas se alimentan por la parte inferior, zona donde también se alimenta el flujo de aire. Esta configuración favorece la colisión burbuja-partícula, por la agitación que se genera en el fondo de la celda. Esto hace muy probable que las partículas de mineral de mena logren llegar a la zona de recolección, aunque también hace que el ascenso de ganga adherido a las burbujas se vea incrementado.

Por otro lado, las celdas columnares o columnas de flotación están diseñadas para maximizar la ley del mineral en el concentrado que producen, en desmedro de la recuperación de este. Estas celdas se alimentan por la parte superior, y el mineral cae por gravedad a través del fluido. Al mismo tiempo, las burbujas ascienden, produciéndose el fenómeno denominado flujo contracorriente, donde las burbujas y las partículas se encuentran al moverse en direcciones distintas. Este fenómeno hace menos probable el encuentro entre las dos fases, disminuyendo la recuperación. Pero, por otro lado, al no generar una intensa agitación, la recuperación de ganga se ve desfavorecida, lo que asegura que los minerales que se vayan recuperando sean en su mayoría de mena, maximizando la ley del concentrado.

Para maximizar la recuperación y ley en la planta, se deben combinar ambas en un circuito de flotación [6]. El circuito más usado de flotación de cobre es el denominado Rougher/Cleaner-Scavenger. La configuración de este circuito se muestra en la Fig 2.1, donde las flechas rojas representan flujos de alimentación, las amarillas de concentrado y las verdes de relave:

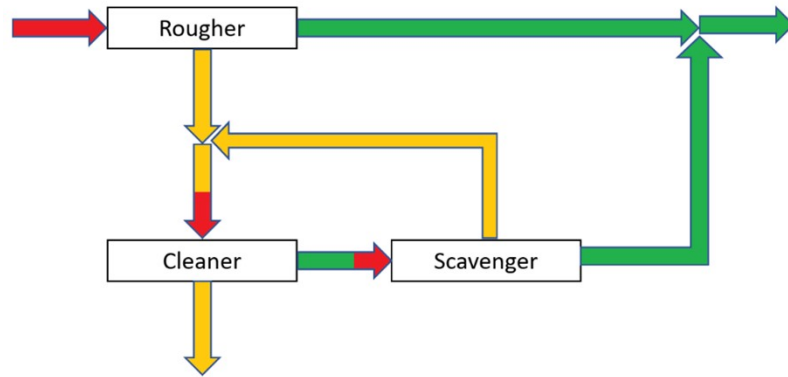


Figura 2.1: Circuito de flotación estándar

Cada una de estas etapas tiene una función en específico:

- Celdas Rougher: Estas son las primeras en un circuito convencional, y se alimentan directamente con el flujo de pulpa proveniente del estanque. Su función es poder recuperar la mayor cantidad de mineral de mena, por lo que se utilizan celdas del tipo convencional. El concentrado de estas celdas tiene una ley muy baja como para ser comercializado o enviado a fundición, por lo que debe ser re-tratado en la siguiente etapa. Por otro lado, el relave tiene baja ley y es considerado como un desecho, por lo que es enviado a los tranques.
- Celdas de Limpieza o Cleaner: Estas celdas son alimentadas con el concentrado proveniente de las celdas rougher y su labor es limpiar este concentrado de la ganga que presenta, maximizando la ley, por lo que se deben utilizar celdas columnares. Por regla general, el concentrado que se produce en esta etapa es el final que produce la planta. Por otro lado, su relave aún tiene una considerable cantidad de cobre que no se pudo recuperar, por lo que es vuelto a flotar en la siguiente etapa.
- Celda de Repaso o Scavenger: Estas celdas son alimentadas con el relave de las columnas de flotación. Al tener un contenido alto de mena y ganga, se busca recuperar toda la mena que sea posible. Para hacer esto, se hace uso de celdas convencionales. El concentrado que producen es reenviado a las columnas, mientras que el relave, al tener bajo contenido de cobre, es desechado y enviado al tranque.

Hay que mencionar que lo normal es que cada etapa no es efectuada por una sola celda del tipo correspondiente, sino que se organiza una configuración de estas en serie y paralelo. Cada grupo de celdas en serie se denomina banco de celdas. En un banco, la alimentación de una celda es el relave de la celda anterior. Esto se hace con el objetivo de poder aumentar el tiempo de residencia del mineral dentro de las celdas, y maximizar la recuperación. La suma de todos los concentrados forma el concentrado final de la etapa, mientras que el relave de la celda final forma el relave final de la etapa [6].

Ahora bien, si una etapa debe tratar un flujo muy grande como para que un solo banco sea capaz de procesar y obtener la recuperación deseada, la solución más común es instalar varias líneas de procesamiento paralelas, donde el flujo a alimentar se divide en flujos más pequeños y capaces de tratar por las celdas que se tienen.

2.1.3. Pruebas de flotación Batch

Las pruebas de flotación batch son realizadas en un laboratorio, con un equipo denominado celda de flotación batch. El objetivo de realizar este tipo de pruebas es obtener parámetros de flotación que pueden ser utilizados en el diseño y operación de celdas en escala real. Una prueba de flotación batch se diferencia de una flotación normal en el hecho de que no es continua, es decir, no se cuenta con una alimentación constante a la celda, sino que es un proceso con una duración predeterminada y que cuenta con un único instante de alimentación al equipo. Además, la producción de concentrado tampoco es constante en el tiempo, sino que tiene un comportamiento decreciente [7].

Las celdas de flotación batch son equipos de pequeño tamaño, a diferencia de las celdas reales. Cuentan con un recipiente en el cual es ingresado el mineral a flotar junto con agua y los reactivos requeridos. También tiene un sistema de inyección de aire y agitación para poder simular condiciones similares a las obtenidas en un ambiente industrial. Al momento de encenderse el equipo, desde uno de los bordes empieza a rebalsar el material de interés junto con espuma, lo que forma el concentrado. Este se acumula en intervalos de tiempo definidos durante un cierto periodo, hasta que se detiene la prueba. Todo el material que se ha recolectado se considera como concentrado, mientras que el material que queda dentro de la celda sin flotar se considera como relave.

Los pasos generales con los que cuentan una prueba de flotación batch son los siguientes:

1. Se ingresa el material junto con el agua a la celda. La masa por ingresar depende de la dimensión de la celda. El volumen de agua por ingresar es tal que se cumpla que la pulpa generada tenga una concentración en peso de aproximadamente 30 %.
2. Se ingresan los reactivos necesarios y se inicia la etapa de acondicionamiento. Esta consiste en dejar encendida la celda con agitación, pero sin aire. La idea es dar tiempo para que ocurra el proceso de adsorción del colector en la superficie del mineral. El acondicionamiento suele durar en torno a los 10 minutos.
3. Se enciende el flujo de aire y se empieza con la recolección de concentrado desde el borde de la celda. Este proceso se hace en bandejas, las cuales reciben el mineral y la espuma que rebalsan. La recolección de concentrado se hace en intervalos definidos de tiempo, entre los cuales se hace un cambio en la bandeja que recolecta el mineral. Al finalizar la prueba, se tiene un número de bandejas equivalentes al número de intervalos de tiempo que se hayan definido.
4. Durante la prueba, se debe reponer el agua que se pierde con la espuma, con el fin de mantener el ambiente dentro de la celda a una concentración en peso constante. Esta agua debe contener la misma proporción de reactivos que el agua ingresada en un inicio, con el objetivo de mantener las condiciones químicas inalteradas dentro de la celda.
5. Al finalizar la prueba, se tiene una serie de bandejas con los concentrados obtenidos en los distintos intervalos de tiempo, más el material que haya quedado dentro de la celda sin flotar como relave. Todos estos son secados por separado y sometidos a pruebas de análisis elemental y/o mineralógico, con el fin de conocer la composición química de estos.

En una prueba de flotación normal se espera que la masa de concentrado recolectado y la ley vayan disminuyendo con el tiempo. Esto debido a que, al encenderse el equipo, las condiciones generadas son las adecuadas para que el mineral de interés flote de manera inmediata, lo que hace que la mayor parte de este salga al inicio de la prueba, mientras que los restos que hayan quedado sin flotar lo hagan de forma más lenta y paulatina.

Una vez realizados los análisis químicos, se puede calcular la masa de cobre que se ha recuperado en función del tiempo, lo que permite deducir la recuperación de cobre a lo largo de la prueba, dividiendo la masa de cobre que se recuperó en cada intervalo por la masa de cobre total que se ingresó en la celda. Sumando los valores de recuperación, se obtiene la curva de recuperación acumulada, la cual es el resultado más relevante que se puede conseguir de las pruebas, ya que a partir de esta se pueden deducir parámetros que son necesarios a la hora de hacer modelamiento [7]. En la Fig. 2.2 se puede apreciar la forma típica que esta presenta

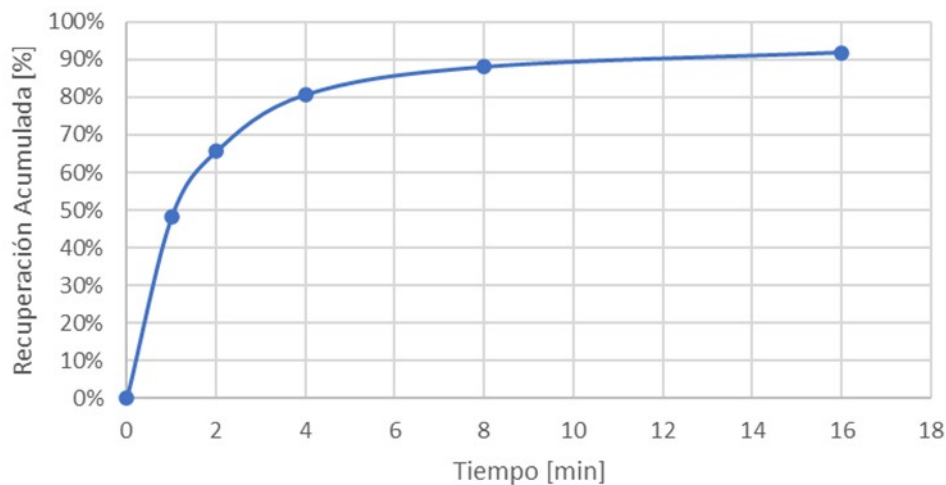


Figura 2.2: Ejemplo de Curva de recuperación Acumulada

2.2. Modelamiento

El desarrollo de esta tesis involucra constantemente el concepto de modelamiento, por lo que resulta importante explicar previamente que se entiende por modelo en ciencias. Un modelo es todo aquel sistema que busca representar un cierto fenómeno, y que puede entregar información valiosa acerca de las leyes naturales que rigen su funcionamiento. Un modelo, por norma general, debería ser más simple y accesible de estudiar que el mismo fenómeno que se está buscando representar [8].

Un modelo puede ser clasificado según la manera en la que se busca representar el fenómeno en específico. Los principales tipos de modelos que se pueden deducir de esta diferenciación son los siguientes:

1. Fenomenológico: Es todo aquel modelo que busca explicar un fenómeno por medio de relaciones empírica y consistentemente con las teorías fundamentales del campo en el que se desenvuelve el fenómeno.

2. Modelo caja negra: Es todo aquel modelo que se obtiene directamente desde datos, buscando encontrar una relación matemática que permita enlazar los datos de entrada con los datos de salida, sin darle un significado físico real a los parámetros que lo definen [9].
3. Modelo caja gris: Es todo aquel modelo que cuenta con una base fenomenológica y que internamente cuenta con parámetros que han sido calculados por medio de métodos caja negra, es decir, es un modelo que combina el conocimiento de las leyes que rigen el fenómeno en cuestión, y la información deducida desde el entrenamiento de modelos caja negra [10].

2.2.1. Modelamiento fenomenológico de flotación

Como se pudo apreciar anteriormente, la flotación es una operación que depende de un gran número de parámetros operacionales, por lo que hacer predicciones sobre el comportamiento de esta puede ser una tarea un tanto compleja. Con el objetivo de poder simular el proceso de flotación, se han generado a lo largo del tiempo una serie de diferentes modelos, los cuales han ido incorporando los distintos parámetros operacionales necesarios para poder representar el proceso de la manera más fehaciente posible.

Los modelos de flotación buscan predecir principalmente la recuperación que se puede obtener de una celda o de un número de celdas determinado. Para esto se deben tomar una serie de supuestos acerca de los parámetros que definen el proceso. A continuación, se presentan los distintos modelos que se han ido planteando con los años, cuáles son los supuestos que toman y cuál es su efectividad a la hora de simular las condiciones reales de una planta.

2.2.1.1. Modelo de flotación de primer orden en flotación Batch

El primer modelo de flotación propuesto se denomina Modelo de flotación de primer Orden, o modelo de García Zúñiga. Este modelo está primariamente pensado para flotación batch, por lo que es un modelo que depende del tiempo. A partir de este se pueden deducir la existencia de dos parámetros cinéticos que son de gran relevancia para los modelos de flotación en continuo. Lo anterior posiciona al modelo de Garcia Zuñiga como base de estudio [11].

El modelo comienza planteando la ecuación de concentración de partículas de una especie o elemento en específico (C_p). Se inicia planteando que la concentración disminuye con el tiempo, lo cual tiene sentido ya que las partículas son cada vez menos a medida que se avanza con la prueba de flotación. Se considera que esta disminución depende de la concentración en sí misma, además de la concentración de burbujas (C_B). Esto se multiplica por una constante (k) para poder realizar un ajuste, con lo que se obtiene la Ec. 2.1:

$$-\frac{dC_P}{dt} = k \times C_P \times C_B \quad (2.1)$$

Con esta ecuación planteada, se pueden hacer algunos supuestos para simplificarla:

1. Se espera que la concentración de burbujas en condiciones estables de laboratorio y de planta no tenga grandes variaciones, por lo que podría considerarse como constante. Con esto en mente, puede ser absorbida por el valor de k .

2. La concentración de partículas de interés se puede ver como la masa de partículas que existen en un volumen definido. Sin embargo, el volumen de la celda de flotación batch es constante, por lo que se puede trabajar con la masa directamente.

Con estas consideraciones, la ecuación se puede reescribir como se muestra en al Ec. 2.2:

$$-\frac{dM}{dt} = k \times M(t) \quad (2.2)$$

Con esto, se plantea una ecuación diferencial ordinaria, la cual tiene por solución la Ec 2.3:

$$M(t) = M_0 \times e^{-k \times t} \quad (2.3)$$

En esta ecuación, M_0 representa la masa inicial ingresada a la celda al inicio de la prueba de flotación, mientras que $M(t)$ es la masa de material valioso que queda dentro de la celda en un tiempo t . Con esto en mente, la recuperación debe ser calculada con respecto a la masa que ha salido de la celda, bajo la Ec. 2.4:

$$R(t) = \frac{M_0 - M(t)}{M_0} \quad (2.4)$$

Finalmente, reemplazando las relaciones se obtiene la ecuación de recuperación de flotación de primer orden, llamada también ecuación de García Zúñiga. Esta se presenta en la Ec. 2.5

$$R(t) = 1 - e^{-k_{GZ} \times t} \quad (2.5)$$

De esta ecuación se puede deducir la existencia de un parámetro de flotación denominado constante cinética de flotación, representada por la letra “k”. Esta constante representa la velocidad con la que un mineral o elemento es recuperado desde una celda de flotación batch. Mientras mayor sea este valor, el mineral tiende a salir más rápido desde la celda de flotación. La constante cinética de flotación es dependiente de varios parámetros, tanto intrínsecos del mineral, como de la operación en sí misma. Los principales parámetros de los que depende el valor de k son:

- Mineralogía: El mineral que se está flotando es muy relevante a la hora de definir el valor de la constante cinética, puesto que ciertos minerales tienen mayor o menor tendencia natural a flotar, lo que afecta cuán rápido sale de la celda.
- Reactivos utilizados: El tipo de reactivo, así como la dosis utilizada definen las condiciones químicas del mineral y de la celda, lo que tiene una gran repercusión en cuán rápido flota el mineral [5].
- Flujo de gases: Tal como se explicó anteriormente, la constante cinética de flotación incluye de manera implícita al flujo de aire dentro de la celda, por lo que también es un parámetro a considerar. Si se tiene una concentración alta de burbujas, se espera un aumento en la constante cinética de flotación [12].

Una vez deducida la expresión indicada anteriormente, se puede ver que, si el tiempo tiende a infinito, la recuperación obtenida es de 100 %. Sin embargo, esta afirmación podría alejarse de la realidad debido al fenómeno de oclusión de mineral de interés. Este consiste en que hay porciones de mineral de interés que se encuentran totalmente rodeadas de ganga,

por lo que estas nunca entran en contacto directo con las burbujas y, por ende, no pueden flotar.

Para corregir este error, se debe integrar un nuevo termino en la ecuación de recuperación batch, el cual se denomina R_∞ y que representa cual es el valor de recuperación que podría alcanzar la celda si esta se deja funcionando un tiempo infinito. Este valor suele deducirse de las pruebas de flotación batch, siendo el valor aproximado superior a la recuperación obtenida al final de la prueba. Teniendo en cuenta esto, la ecuación final de García Zúñiga para flotación batch se puede escribir como se muestra en la Ec. 2.6:

$$R(t) = R_\infty (1 - e^{-k_{GZ} \times t}) \quad (2.6)$$

Esta ecuación puede usarse para modelar la curva de recuperación acumulada mostrada en la Fig. 2.2, y de esta manera, obtener una aproximación de los valores que se deben utilizar para las siguientes etapas de modelamiento. En la Fig. 2.3 se muestra el desempeño del modelo al representar datos industriales.

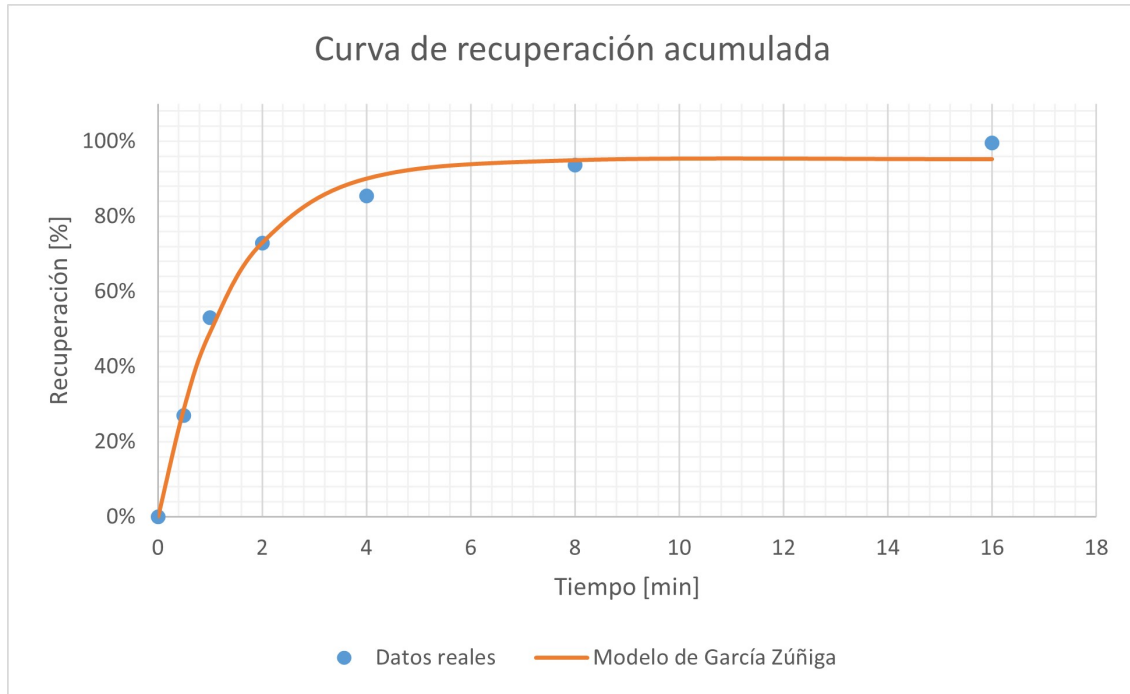


Figura 2.3: Comparación entre datos reales y curva de recuperación modelada

2.2.1.2. Modelo de flotación de primer orden en flotación continua

La ecuación de García Zúñiga anteriormente estudiada presenta buen rendimiento a la hora de representar una flotación batch. Sin embargo, este modelo presenta serias diferencias al momento de simular una flotación de escala industrial. Esto es debido a que, como se explicó previamente, la flotación a escala industrial es un proceso continuo, donde se tiene una alimentación constante, al igual que una producción constante de relave y de concentrado, a diferencia de una flotación batch, donde solo se tiene un instante de alimentación y una sola masa de relave al final del proceso. Esto hace que sea necesario introducir nuevos conceptos para poder adaptar la ecuación de García Zúñiga ya existente.

La forma que se ha planteado para realizar la adaptación es utilizar un análisis en continuo, es decir, realizar un análisis partícula a partícula. Esto se hace considerando el hecho que cada una de ellas tiene una distinta probabilidad de flotar en la celda, dado las diferencias entre los tiempos que pasa cada una dentro y las diferentes características de flotación que tiene cada una por sus diferencias mineralógicas. Con esto en mente, surgen dos nuevos conceptos que se deben representar por medio de funciones matemáticas:

1. Distribución de tiempos de residencia: Es la expresión que representa la probabilidad que tiene cada partícula de permanecer un periodo de tiempo específico dentro de la celda, antes de abandonarla como relave. Este concepto reemplaza la variable temporal vista en la ecuación de García Zúñiga y se representa con el termino $E(t)$ [13].
2. Distribución de constantes cinéticas: Es la expresión que asigna una cierta constante cinética de flotación a cada partícula por separado, por lo que representa qué tan flotables son cada una de las partículas dentro de la celda. Este concepto reemplaza la constante cinética de flotación que, en la ecuación de García Zúñiga original, es única para todas las partículas. Se representa con el termino $f(k)$ [14].

Con lo anteriormente dicho, se puede plantear esta nueva adaptación para calcular la recuperación de una celda o de un banco de celdas, conocida como la ecuación de Polat y Chandler [15], mostrada en la Ec. 2.7:

$$R = R_{\infty} \int_0^{\infty} \int_0^{\infty} E(t) \times f(k) \times (1 - e^{-k \times t}) dk dt \quad (2.7)$$

Esta doble integral representa una suma considerando todos los posibles tiempos que una partícula puede pasar dentro de la celda, y otra suma considerando todas las posibles constantes cinéticas que se pueden asignar. Se puede ver que la recuperación deja de ser una variable dependiente del tiempo, pasando a ser una constante. De esto se puede deducir, que si se mantienen invariables las condiciones operacionales dentro de la celda y los flujos de alimentación permanecen inalterados en todos los aspectos, la recuperación debería mantenerse constante en el tiempo.

La ecuación de Polat y Chandler es la base del modelo fenomenológico planteado para la utilización de esta tesis, por lo que comprender cada uno de sus términos y saber aplicarlo de manera correcta, resulta vital para el trabajo que se ha planeado realizar. Estudiando cada uno de sus términos, es posible detectar cuales son los que presentan mayor flaqueza a la hora de representar la realidad, y encontrar los puntos de aplicación donde se podrían hacer ajustes usando redes neuronales u otro sistema de *Machine Learning*.

2.2.1.2.1. Distribución de tiempos de residencia

Con lo anterior en mente, se debe estudiar, en primer lugar, cómo afecta la variable temporal en este caso. Tal como se dijo anteriormente, las celdas de flotación industrial presentan una alimentación continua, es decir, existen un flujo constante de partículas, tanto de mineral como de ganga, entrando en la celda y distribuyéndose en su interior, teniendo la posibilidad de salir como concentrado o como relave. Como primera aproximación, se puede considerar que la agitación existente en el interior de la celda es la suficiente como para que las concentraciones de partículas sean equitativas en cualquier parte de esta. Esta aproximación se denomina modelo de estanque perfectamente agitado.

Con esta consideración, se puede ver que cada partícula pasará un determinado tiempo dentro de la celda desde el instante en que entra hasta que sale. Para este análisis se considera estudiar solamente las partículas de relave, dado que son aquellas que estuvieron tiempo en la celda, con posibilidades de flotar y no lo hicieron. El tiempo que pase cada una de estas partículas dentro de la celda se denomina tiempo de residencia. Al extender este concepto a todas las partículas de la celda, se deduce un modelo se denomina distribución de tiempos de residencia, que indica cuál es la probabilidad de que una partícula que ha entrado en la celda permanezca un tiempo t^* dentro de ella antes de salir como relave. Conocer la distribución de tiempos de residencia de una celda resulta vital en el estudio de esta tesis, ya que es uno de los ítems que permite adaptar la fórmula de recuperación de García Zúñiga de un proceso batch a un proceso en continuo.

Al conocer el tiempo de residencia de todas las partículas que han entrado en la celda en un momento en específico, se puede deducir cuál es la fracción de mineral que ha permanecido un tiempo t^* dentro de la celda y han salido como relave. Este valor puede ser usado para predecir cuánto tiempo pasará una partícula que ha entrado en la celda. Sin embargo, resultaría muy complejo poder medir de manera experimental el tiempo exacto que cada partícula pasa dentro de la celda antes de salir, por lo que se recurre a la utilización de modelos para poder obtener esa función de probabilidades. Existe un modelo básico, el cual representa la distribución de tiempos de residencia como una función con distribución exponencial [13], tal como se muestra en la Ec. 2.8:

$$E(t) = \frac{1}{\tau_m} \times e^{-\frac{t}{\tau_m}} \quad (2.8)$$

El parámetro τ_m se conoce como tiempo de residencia medio, y corresponde al valor promedio que todas las partículas pasan dentro de la celda. Este valor puede ser obtenido operacionalmente desde la celda de flotación, mediante la Ec. 2.9:

$$\tau_m = \frac{V_{ef}}{Q_{tr}} \quad (2.9)$$

En esta ecuación se tienen dos términos que es vital comprender, ya que ambos pueden verse afectados por los parámetros del proceso:

- Volumen efectivo (V_{ef}): Es el volumen dentro de la celda de flotación que puede ser utilizado efectivamente por mineral. La manera de calcularlo es el volumen nominal de la celda, es decir, el que se obtiene de sus dimensiones físicas, restando todos aquellos volúmenes dentro de la celda que no pueden ser ocupados por mineral. Estos son el volumen de espuma en la superficie, el volumen de mecanismos dentro de la celda y el volumen de aire en forma de burbujas dentro.
- Flujo volumétrico de Relave (Q_{tr}): Corresponde al flujo total que abandona la celda. Este flujo considera tanto el mineral sólido en forma de partículas, ya sean de mineral o ganga, y el agua.

Al estudiar el comportamiento que posee la función de tiempos de residencia en el tiempo, se puede ver que presenta la tendencia de la Fig. 2.4 :

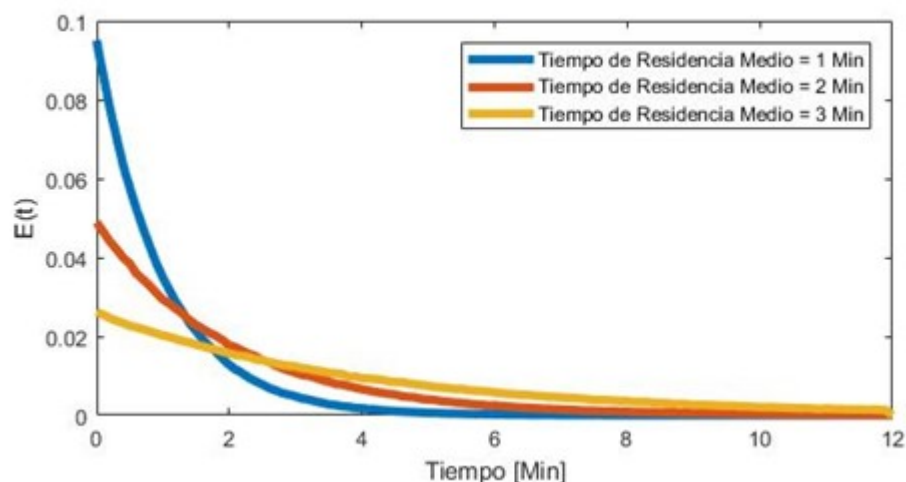


Figura 2.4: Distribuciones de tiempo de residencia con tiempo de residencia medio variable

Se puede apreciar en todos los gráficos un comportamiento decreciente en cuanto al valor de la distribución de tiempos de residencia, lo que significa que en una sola celda es más probable que una partícula pase un periodo de tiempo muy breve dentro del equipo antes de pasar a formar parte del relave. Esto es perjudicial para el proceso, dado que las partículas no tienen el tiempo suficiente para estar dentro de la celda, encontrarse con una burbuja y ascender hasta el concentrado. Una forma de corregir esto puede observarse dentro del mismo gráfico y corresponde a aumentar el tiempo de residencia medio de las partículas dentro de la celda. Esto hace que la curva se aplane, lo que significa que a menor tiempo disminuye la probabilidad, mientras que a mayor tiempo aumenta. Dos formas de aumentar el tiempo de residencia medio podrían ser, por un lado, instalar celdas más grandes para tener mayor volumen efectivo o, por otro lado, disminuir los flujos de alimentación. Sin embargo, esto último no suele ser rentable, ya que implicaría una disminución en la producción de concentrado desde las celdas.

A priori, podría pensarse que disminuir el flujo de aire en la celda o disminuir la altura de espuma podrían ser soluciones, dado que esto implicaría un aumento en el volumen efectivo. Sin embargo, estas dos variables también definen otros procesos que ocurren dentro de la celda y que tienen injerencia directa en el valor de la recuperación, por lo que disminuir su valor puede resultar contraproducente para el proceso. Aumentar el C_p de un flujo también podría parecer una idea razonable, ya que permitiría disminuir el flujo volumétrico y aumentar el tiempo de residencia medio. Sin embargo, esto tampoco es una buena idea, ya que se requiere que el C_p esté dentro de ciertos límites definidos, con el objetivo que el relave pueda tener un comportamiento más fluido y que no se embanque en el fondo de la celda.

La principal forma de aumentar el tiempo de residencia de las partículas consiste en añadir más celdas en serie dentro del banco, ya que de esta manera se logra minimizar la probabilidad que una partícula entre en cada una de estas y pase un periodo de tiempo mínimo dentro de ella. Mientras más celdas se agreguen en serie, es más probable que una partícula pase un periodo de tiempo prolongado dentro, a diferencia de las partículas que pasan un periodo de tiempo muy breve en cada celda a la que entran, y posteriormente abandonan el equipo. Esto

puede verse gráficamente en la representación de las distribuciones de tiempo de residencia, tal como se muestra en la Fig. 2.5:

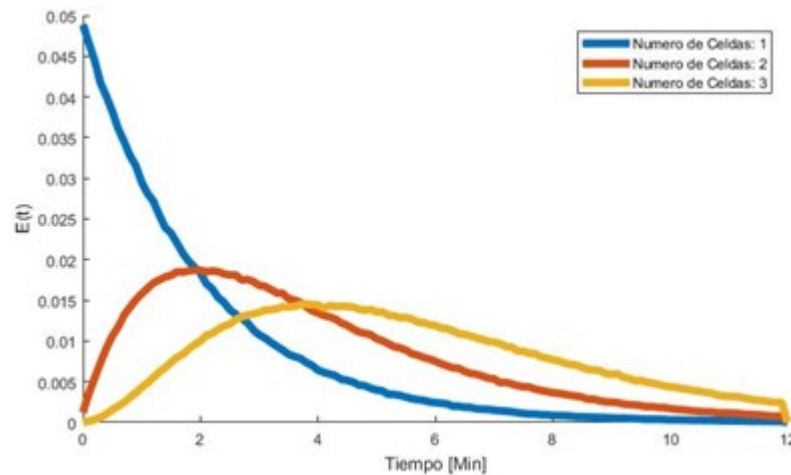


Figura 2.5: Distribuciones de tiempos de residencia con número de celdas variable

Se puede apreciar cómo aumentar el número de celdas sirve para que las probabilidades más altas no sean asignadas a los tiempos menores, sino que a estadías más prolongadas dentro de la celda. Con esto en mente, la planeación del número de celdas que debe tener un banco suele ser un trade off entre tener una distribución de tiempos de residencia que maximice la recuperación, y el costo operacional de añadir celdas extra.

2.2.1.2.2. Distribución de constantes cinéticas

Como se explicó en el capítulo anterior, la constante cinética de flotación “k” representa en la ecuación de García Zúñiga qué tan flotable es una partícula frente a determinadas y específicas condiciones químicas y físicas de sí misma y de la celda de flotación. Mientras más alto sea este valor, se puede interpretar que el mineral tiende a flotar más rápidamente y por ende abandona de forma más abundante la celda, generando una recuperación más alta en el proceso.

Este concepto, al ser trasladado a una celda de flotación convencional, debe ser adaptado de la misma forma en que se adaptó la distribución de tiempos de residencia. Sin embargo, a diferencia de ésta, la distribución de constantes cinéticas debe ser estudiada por separado según el tipo de mineral o elemento. Esto dado que, como se ha descrito anteriormente, existen minerales, denominados mena, que presentan propiedades que permiten facilitar su recuperación por el proceso de flotación, a diferencia de aquellas que no flotan, denominadas ganga.

Además, también se hace importante destacar que es posible contar con más de un mineral que tenga el elemento de interés en la mineralogía que se esté tratando. Esto es relevante puesto que no todos los minerales se comportan de la misma forma a la hora de ser procesados.

Como caso base y supuesto general se puede trabajar con dos aproximaciones a la hora de trabajar con mineralogías que incluyan más de una especie de interés:

1. Conocer solo la constante cinética de la especie en cuestión, y de ella y de una serie de supuestos, buscar deducir el comportamiento de los minerales por separado.
2. Conocer la constante cinética de cada mineral por separado, por lo que se puede conocer explícitamente la recuperación de cada uno.

En esta tesis, se busca la mayor cercanía posible con aquellos datos con los que se podría contar en la operación real de una planta de flotación. Por ello, se ha recurrido al primer método, aplicando supuestos que son presentados y fundamentados más adelante.

Si se estudia la flotación de una especie en específico, se pueden hacer dos aproximaciones sobre cómo se espera sea el comportamiento de esta propiedad:

1. Todas las partículas de una especie en específico presentan las mismas propiedades a la hora de flotar, independiente de su tamaño o su grado de liberación. Esta aproximación es equivalente a la propuesta en la ecuación de García Zúñiga, por lo que es la traslación directa de la misma a flotación continua.
2. Las partículas de una especie en específico presentan distintas capacidades de flotar, diferenciadas por el tamaño o grado de liberación. Esta aproximación busca parametrizar mediante una función, la distribución de constantes cinéticas. De esta aproximación se puede deducir una serie de modelos, de entre los cuales destaca uno denominado modelo de Klimpel [16].

A continuación, se presentan las dos ecuaciones que se han utilizado con el fin de representar ambos casos. La Ec. 2.10 muestra la distribución asociada a Klimpel, mientras que la Ec. 2.11 muestra la distribución asociada a García Zúñiga. Además, se presenta la Fig. 2.6, donde se muestran gráficamente.

$$f(k) = \begin{cases} \frac{1}{K_k} & \text{si } k \leq K_k \\ 0 & \sim \end{cases} \quad (2.10)$$

$$f(k) = \begin{cases} 1 & \text{si } k = K_{GZ} \\ 0 & \sim \end{cases} \quad (2.11)$$

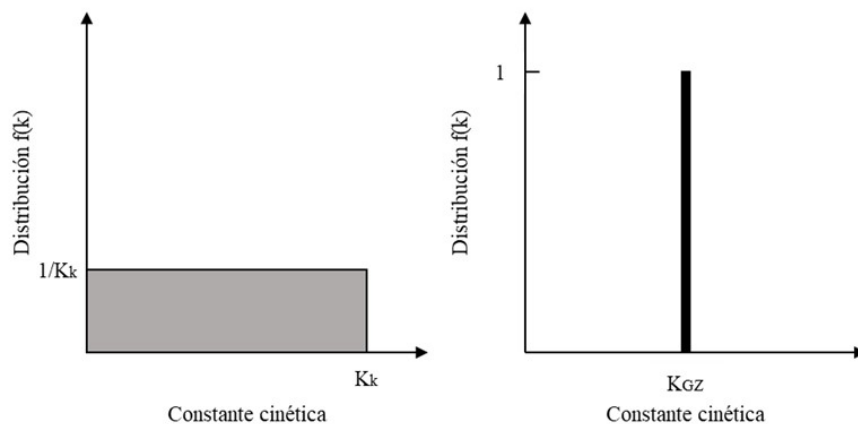


Figura 2.6: Representaciones gráfica de la distribución de constantes cinéticas de Klimpel y Garcia Zuñiga

2.2.1.2.3. Estado del arte en modelos de flotación

Al hacer una revisión bibliográfica acerca de los modelos de flotación existentes, se puede ver una gran variedad de enfoques desde los cuales se ha abordado el problema. Uno de los primeros puntos a estudiar es acerca del modelamiento ya existente que se tiene sobre las distribuciones de tiempos de residencia. En este campo, se ha desarrollado avances en el modelamiento, como lo es en el trabajo de Yianatos et al. [17], en el cual se presenta una propuesta para una expresión analítica que representa la distribución de tiempos de residencia, escrita en la Ec. 2.12.

$$E(t) = \frac{N^N \times \left(\frac{t}{\tau}\right)^{N-1} \times \exp(-N \times \frac{t}{\tau})}{\tau \times (N-1)!} \quad (2.12)$$

Donde N es el número de celdas del banco y τ es el tiempo medio de residencia media promedio de todas ellas. Desde esta ecuación, se puede deducir la expresión para la recuperación, por medio de la fórmula de Polat y Chandler y los supuestos de García Zuñiga, la cual está indicada en la Ec. 2.13.

$$R = R_{\infty} \left(1 - \frac{N^N}{\tau^{N-1} \times (N-1)! \times (N + k \times \tau)}\right) \quad (2.13)$$

Se puede ver como ambas ecuaciones presentan una limitante en cuanto a su uso, y es que ambas consideran en la misma ecuación tanto el valor de N , es decir, el número de celdas, y el valor de τ , es decir, el tiempo de residencia medio. Esto implica que ambos modelos consideran que todas las celdas de un banco celdas tienen el mismo tiempo de residencia medio, lo cual es un error, dado que en cada celda, una porción de mineral es retirado desde el flujo por el concentrado, haciendo que en un banco los flujos tratados sean cada vez menores a medida que se avanza y haciendo crecer el tiempo de residencia medio. Considerando esto, el modelo fenomenológico propuesto para desarrollar esta tesis corrige este error, asignando un valor medio de tiempo de residencia por cada celda y luego combinando valores por medio de Montecarlo.

Otro de los puntos a evaluar es cuales son los modelos que se usan los simuladores industriales. Para esto, se investigó acerca de uno de los software de flotación más usados en la industria, como lo es JKSim, el cual es un simulador de procesos de conminución, separación por tamaño y flotación [18]. El software recibe una serie de señales provenientes desde la planta y con ellos simula diversas variables operacionales. En el caso de la etapa de flotación, el modulo específico se denomina JKSimFloat. Al investigarse sobre el funcionamiento de este modulo, se puede ver que el modelo de recuperación resumido utilizado para modelar la flotación verdadera es el mostrado en la Ec. 2.14 [19].

$$R = \frac{k \times \tau}{k \times \tau + 1} \quad (2.14)$$

La Ec. 2.14 es una extensión de la ecuación de García Zuñiga de flotación batch a flotación continua. Se puede ver que solo depende del valor de la constante cinética y del tiempo de residencia medio. No está presente la variable N que hace referencia al número de celdas. De esto se deduce para tratar la recuperación de un banco no se ve sistema como un todo, sino que se aborda el problema celda a celda. Eso es una diferencia clara con el modelo fenomenológico deducido desde la ecuación de Polat y Chandler (Ec. 2.7), ya que esta, al estudiar el problema globalmente, utiliza la verdadera distribución de tiempos de residencia

que describe el banco, situación que no se aprecia en el modelo de JKSimFloat.

2.2.2. Modelos caja negra de simulación

Ya se ha explicado la base de funcionamiento de los modelos fenomenológicos a utilizar en el desarrollo de esta tesis. Sin embargo, tal como plantea el título de esta tesis, se hace necesario explicar en general los modelos de caja negra, y más específicamente, el funcionamiento de las redes neuronales, ya que son el ítem complementario a un modelo fenomenológico que permite construir un modelo de caja gris.

En líneas generales, se define un modelo de caja negra como todo aquel método que busque representar un fenómeno de cualquier índole buscando una relación matemática entre dos sets de variables, uno de entrada y uno de salida, sin poner el foco en buscar reproducir o representar la física o química que se encuentre detrás, sino que solo defina una relación que ajuste bien los datos existentes [20].

Al momento de estudiar la aplicación de los modelos de caja negra, se puede encontrar un termino muy nombrado como forma de implementación, que es el concepto de Machine Learning. Se conoce como Machine Learning a todo el conjunto de técnicas utilizadas con el fin que un sistema informático realice una operación o acción para la cual ha sido entrenada. El fundamento pilar de este conjunto de metodologías es darle ciertas directrices bases al sistema, con el objetivo de que él mismo aprenda cual es el método más efectivo para realizar la acción u operación solicitada. Existen diferentes metodologías de Machine Learning, que se diferencian entre sí en la profundidad del método que utilizan para realizar la acción requerida y las condiciones iniciales que usan [21].

Desde lo anterior se puede deducir lo desafiante que implica la implementación de modelos caja negra y una ventaja esencial por sobre los modelos fenomenológicos. El principal desafío presentado es encontrar cuál es aquella ecuación que mejor relaciona variables inputs y outputs del sistema estudiado. En la actualidad existen varios métodos para encontrar esta ecuación, entre los cuales se pueden nombrar los más comunes como regresiones lineales o polinomiales, o algunos más avanzados como lo son las redes neuronales. Por otro lado, la principal ventaja que puede entregar un modelo caja negra por sobre uno fenomenológico es la simpleza de las ecuaciones utilizadas, dado que estas suelen ser conjunciones de diversas operaciones matemáticas, a diferencia de las simulaciones fenomenológicas, que requieren un análisis más profundo y exhaustivo, como métodos aleatorios o discretizaciones de ecuaciones diferenciales.

Sin embargo, un método de caja negra presenta dos claras desventajas frente a los métodos de simulación fenomenológica. En primer lugar, es que no todos los fenómenos pueden ser representados por cualquier método de caja negra. Existen métodos que son muy simples, como las regresiones, incapaces de modelar distribuciones complejas de datos. Esto hace que, si se tiene un fenómeno más complejo, se deba recurrir a métodos caja negra más avanzados, como las redes neuronales. La otra clara desventaja es que requiere de la existencia de una base de datos para poder generar la ecuación que represente el fenómeno, por ende, es necesario realizar pruebas experimentales o medir desde un proceso industrial para poder trabajar con este método, a diferencia de la simulación fenomenológica, que solo requiere comprender

la dinámica del estudio a realizar. Dependiendo del método a utilizar, la dimensión de esta base de datos debe ser de mayor o menos envergadura, aunque por lo general se espera contar con una que contenga todos los casos que se esperan observar en la realidad.

Por lo general, cualquier modelamiento que use un método de caja negra, debe contar con dos etapas básicas para generar el modelo. La primera es la etapa de entrenamiento, que consiste en tomar la base de datos y encontrar cuál es la ecuación o sistema que mejor representa este proceso. El modelo que es entrenado toma los input en la base de datos y ajusta sus parámetros internos con el fin de hacer que el output sea lo más similar posible a la salida esperada en la base de datos. Cada uno de estas evaluaciones y modificaciones se denominan *batch*, y el tamaño de *batch* es la cantidad de veces que se realiza este proceso [22]. El tamaño de *batch* óptimo es aquel que implica presentar todos los posibles casos al modelo al momento de entrenar, pero sin ser demasiado grande, ya que se puede caer en el *overfitting* [23], que es sobre ajustar el modelo solo a los datos de entrenamiento, haciendo que pierda generalidad predecir para casos nuevos.

La segunda etapa es la de validación. Esta consiste en evaluar los inputs de la base de datos con los modelos creados y comparar los outputs reales que se tienen con aquellos predichos. Esta etapa permite evaluar el desempeño del modelo, ya que se puede observar cuánto se acerca lo que el modelo entrega a los datos reales y cuantificarlo mediante diversas herramientas estadísticas. Si se han realizado diversos modelos, esta etapa permite determinar y elegir cual es el modelo más representativo de la realidad.

Otro aspecto importante por revisar es la partición que se debe realizar a la base de datos en ambas etapas. La forma más básica de hacerlo es tomar toda la base para crear el modelo y después usar cada dato por separado para verificar su funcionamiento. Sin embargo, este método no suele ser muy utilizado, dado que se prefiere trabajar con bases por separado para entrenamiento y testeo, para que de esta forma los datos por testear no tengan inferencia en el modelo. Por regla general, se reparten los datos de manera aleatoria entre una base de entrenamiento y una de testeo, en una proporción predefinida y dependiente del fenómeno a estudiar.

2.2.2.1. Clasificación de modelos de caja negra según su función

Tal como se explicó anteriormente, los modelos de caja negra buscan simular un proceso matemático con el fin de entregar un resultado similar al resultado real obtenido. Esto implica que los modelos de caja negra pueden ser usados en varios ámbitos para representar diferentes tipos de funciones. A continuación, se presentan dos de los principales objetivos que puede presentar un modelo de caja negra.

2.2.2.1.1. Modelo de Regresión

Un modelo de regresión es aquel que busca predecir un valor o set de valores de salida tras la transformación a un valor o set de valores de entrada, es decir, su objetivo es simular una función de variable continua de alguna clase, la cual puede estar bien definida, o desconocerse el funcionamiento [24].

Un modelo de regresión puede tener varias opciones para ser creado, siendo algunas de las principales las regresiones lineales y polinomiales, y algunos casos de redes neuronales.

Estos campos tienen en común que las variables a trabajar, tanto las de salida como las de entrada, suelen ser números continuos, lo que permite hacer la relación directa con una función también continua.

2.2.2.1.2. Modelo de clasificación

Los modelos de clasificación son aquellos que, al recibir un set de valores, los procesa y predice a que tipo o clase pertenecen esos datos, categorizándolos por medio de marcadores o índices discretos. Considerando esto, este tipo de modelos tiene la función de identificar datos en grupos que, previamente, han sido identificados como poblaciones distintas [24]. Hacer esto puede tener diversos propósitos. Por ejemplo, para el desarrollo de esta tesis, se buscó desarrollar un modelo que permitiera identificar y categorizar unidades geometalúrgicas, lo que puede ser usado para asignar constantes cinéticas adecuadas al momento de modelar el proceso de flotación.

La principal forma de desarrollar algoritmos de clasificación es por medio de la categorización, es decir, la asignación de un Número a las distintas poblaciones. Esto se hace con el fin de poder transportar las diferencias cualitativas o cuantitativas entre dos o más sets a un espacio numérico, el cual pueda ser modelado, calculado y predicho. Con esta idea en mente, se crea un algoritmo que permita estimar aquel valor numérico que se haya asignado.

Existen varias metodologías que permiten clasificar datos según un set de valores de entrada, como los son el método de los k vecinos más cercanos, el Support Vector Machine y algunas técnicas utilizadas en redes neuronales. El método de los vecinos más cercanos presenta una mayor utilidad cuando se busca clasificar variables que tienen una fuerte correlación espacial. Por otro lado, los otros dos métodos son usados para hacer clasificaciones de cualquier tipo, sin importar el origen de la data a utilizar. Otras metodologías no predicen directamente el valor de la categoría, sino que entregan la probabilidad que una cierta variable o sets de variables pertenezca a un set de clases. Este valor es más útil puesto que puede mostrar el grado de certeza que posee el algoritmo a la hora de hacer la clasificación, información que puede ser usada por el desarrollador para elegir como categorizar un cierto valor input.

2.2.2.2. Redes Neuronales

2.2.2.2.1. Definición y creación

Dentro del mundo de Machine Learning, uno de los campos de investigación, desarrollo y aplicación más avanzados y útiles es el Deep Learning. Este consiste en utilizar algoritmos de capas o layers con el fin de transformar una serie de inputs en outputs. La idea base es que en cada capa la información es transformada desde la data original a una más abstracta [25]. Todo este marco de estudio es recae en una unidad básica denominada neurona. Esta recibe una serie de señales de entrada y las transforma mediante operaciones matemáticas de diversa índole, con el fin de obtener una serie única de señales de salida [26].

Con lo anterior en mente, una red neuronal se puede definir como un ordenamiento interconectado de neuronas. Estas se ordenan internamente, pudiendo recibir como input una señal dada por una variable de entrada o una señal proveniente de otra neurona. Al momento de ordenar las neuronas, se pueden identificar tres capas: (1) La capa inicial, correspondientes a las señales ingresadas como inputs en el modelo; (2) las capas ocultas, que son aquellas

que acumulan todas aquellas neuronas que realizan el trabajo de procesamiento de los datos; (3) la capa de salida, que es aquella que entrega las señales correspondientes a las variables output del modelo [24]. A continuación, en la Fig. 2.7 se muestra una representación visual del proceso de cálculo que implica una red neuronal:

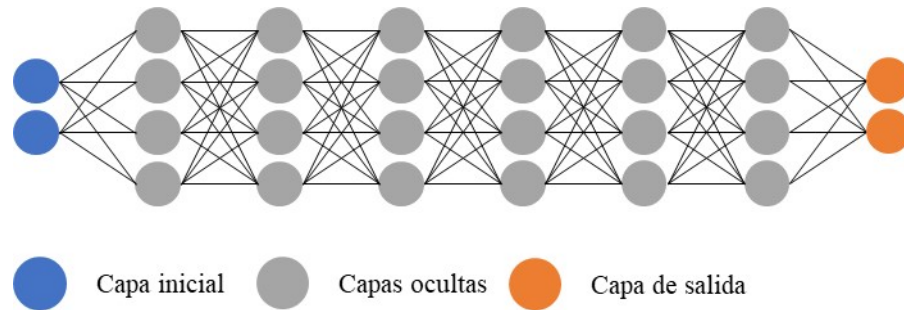


Figura 2.7: Esquema del funcionamiento de una red neuronal

El número de capas ocultas y número de neuronas en estas puede ser definido por el creador de la red. Más neuronas, implica mayor tiempo de entrenamiento para hallar cuál es la operación que mejor ajusta en cada una, pero puede entregar una mejor precisión a la hora de modelar el fenómeno en cuestión.

El fundamento matemático que posee una red neuronal se puede dividir en dos aspectos principales: (1) Las operaciones matemáticas que conforman el cálculo de cada una de estas neuronas; (2) La metodología utilizada para que los parámetros dentro de la neurona sean los óptimos.

El cálculo matemático que se realiza dentro de cada neurona se puede dividir en las siguientes etapas [26]:

1. Se ingresa los diversos valores de entrada, a los que también se les llama señales, las que pueden venir desde la capa inicial o desde una capa previa dentro de la capa oculta. Cada uno de estos valores es multiplicado por distintas constantes, a las que se les denomina “pesos”. Todos estos productos se suman. A lo anterior, se suma un valor constante, llamado sesgo.
2. De las operaciones anteriores, se puede deducir que se ha producido la suma de varias ecuaciones lineales, lo que resulta en una sola ecuación lineal o, equivalentemente, una suma ponderada de las variables de ingreso.
3. Se le aplica una función a la ecuación lineal, con el fin de poder acotar el valor de las posibles salidas a ciertos valores específicos. Esta función se denomina función de activación, que debe ser una función de transferencia no lineal.

En la Fig 2.8, se esquematizan los pasos anteriormente mencionados. Las señales de entrada están representadas por Y , los pesos por λ , el sesgo por S y la función de transferencia por ϕ :

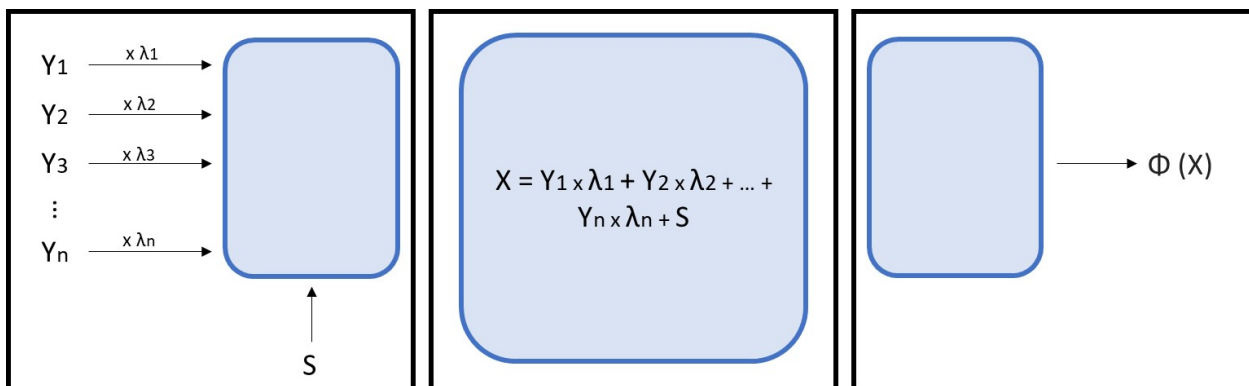


Figura 2.8: Esquemización de labor de cálculo de una neurona

Las funciones de activación pueden tener distinta forma dependiendo de cómo calculan la salida de la neurona. Esto puede realizarse para tres objetivos: (1) Restringirla a un set de datos discretos; (2) Acotarla a un cierto intervalo; (3) Imponerle que sea mayor a un número específico. Además, como se debe usar el método de backpropagation para la etapa de entrenamiento, es necesario que estas funciones cuenten con primera derivada.

Con las explicaciones anteriores, ya se ha dejado clara la estructura que posee una red neuronal, que es la conjunción de estas unidades llamadas neuronas que trabajan en conjunto para producir un valor de salida. Sin embargo, aún no se ha hablado sobre cómo se hace para encontrar los valores óptimos a utilizar en los pesos y en los sesgos de cada neurona. Los algoritmos de Deep Learning cuentan con la particularidad de que no es necesario encontrar los valores óptimos por cuenta del programador, sino que el mismo sistema se auto entrena con el objetivo de lograr esto. Para conseguir lo anterior, se han diseñado una serie de métodos que se usan para que el sistema encuentre por sí solo cuáles son los valores óptimos, aunque la mayoría se basan en la utilización de una misma metodología base llamada backpropagation.

2.2.2.2.2. Backpropagation

La metodología de backpropagation consiste en recorrer la red neuronal desde los valores de salida hasta los de entrada, con el fin de estudiar el error entre el valor que entrega la red y el valor que se espera recibir. En paralelo a esto, se analiza cuáles son las neuronas que más interferencia presentan a la hora de producir ese error, para luego cambiar el valor de los pesos y los sesgos de esas neuronas. Por lo general, al ser creadas las redes neuronales, se asignan valores aleatorios en las neuronas, por lo que al inicio las predicciones pueden parecer que no tengan sentido. Sin embargo, al aplicar el algoritmo de backpropagation una cantidad suficiente de veces, se puede ver que el ajuste y la correlación de los datos de salida con los datos de esperados empieza a crecer, prueba de que el modelo está cumpliendo su labor cada vez mejor. La medida del error existente entre los datos predichos y los datos reales se llama función de pérdida. Esta función puede tener varias posibles definiciones según el caso lo requiera, las cuales son explicadas con posterioridad.

Para entender como funciona el algoritmo de backpropagation, se puede tomar un ejemplo simplificado de una red con sola neurona en la capa de salida y N neuronas en la capa previa. El valor de la salida de la red se puede escribir como se muestra en la Ec. 2.15 [?].

$$F(x) = \psi(F_1(x) \times W_1 + F_2(x) \times W_2 + \dots + F_N(x) \times W_N + S) \quad (2.15)$$

Donde se tiene que ψ es la función de activación en la neurona de salida, F_n son las señales provenientes de las neuronas de la capa anterior, W_N son los pesos asociados a esas señales y x es el valor de la capa inicial ingresado al modelo.

Considerando esto, se puede definir la función de pérdida L como una medida de diferencia entre el valor de predicción $F(x, W, S)$ y los valores reales entregados para entrenamiento $Z(x)$. Al entrenar una red, se busca que los valores de $Z(x)$ y $F(x, W, S)$ sean lo más similares posibles, por lo que se desea es minimizar el valor de la función L , lo cual se logra mediante la derivada de los valores que se buscan optimizar, es decir, los pesos y sesgos de la red neuronal, lo que se plantea en la Ec. 2.16.

$$\frac{\partial L}{\partial W_N, S} = 0 \quad (2.16)$$

Por ejemplo, si se está trabajando con la función de pérdida MSE, se puede reescribir tal como se muestra en la Ec. 2.17 [27].

$$L(x) = \frac{1}{2N} \left(\sum_{i=1}^N Z(x) - F(x, W, S) \right)^2 \quad (2.17)$$

Considerando lo anterior, por regla de la cadena se tiene que es necesario conocer igualmente el valor de la derivada de $F(x)$, la cual se puede escribir como se muestra en la Ec. 2.18 [?].

$$F'(x) = \psi'(s) \times (F'_1(x) \times W_1 + F'_2(x) \times W_2 + \dots + F'_N(x) \times W_N + S) \quad (2.18)$$

Donde se tiene que $s = \psi(F_1(x) \times W_1 + F_2(x) \times W_2 + \dots + F_N(x) \times W_N + S)$.

Teniendo a el método de backpropagation ya explicado, hace falta también entender el concepto de optimizador. Este corresponde a algún algoritmo que tiene como fin guiar y encontrar en cada iteración los valores de pesos y sesgos que hacen que la red sea a cada paso más óptima para la labor que se requiere. Para el desarrollo de esta tesis, se ha usado el algoritmo de optimización Adam, el cual, según sus diseñadores, es computacionalmente eficiente, con requerimiento mínimo de memoria y bien preparado para problemas con muchos datos [28].

2.2.2.3. Estimación de densidad con Kernel

Otra metodología de predicción dentro del ámbito de Machine Learning es la estimación de densidad con Kernel. A diferencia de las redes neuronales, esta metodología solo puede ser utilizada en la distribución de funciones de densidad, es decir, distribuciones que indican la probabilidad continua de que un determinado suceso ocurra. Con esto en consideración, se ha visto que esta metodología tiene un gran éxito a la hora de representar esta clase de resultados.

Para poder aplicar este método, se necesita tener un set de datos de derivado de la función de densidad $f(t)$ que se quiera modelar, o que distribuya de la manera que esta lo hace. Si se tiene n datos conocidos, entonces este set de datos se puede escribir como $[x_1, x_2, \dots, x_n]$. Luego, la función de densidad deducida por estimación con Kernel $\hat{f}(t)$ se puede escribir

como se muestra en la Ec. 2.19 [29].

$$\hat{f}(t) = \frac{1}{n} \sum_{i=1}^n K(x_i, t) \quad (2.19)$$

Donde se tiene que el termino $K(x_i, t)$ se le conoce como función de Kernel. Existen varios tipos de funciones que pueden ser usadas como Kernel, como lo son funciones de tipo exponencial, gaussiano o rectangular. Sin embargo, todas ellas deben compartir un par de características [29]:

- No pueden ser menores que 0, o diverger a ∞ en ningún valor de x_i ni de t .
- Su integral en todo su dominio con respecto a t debe ser igual a 1.

Las funciones de Kernel pueden ser simétricas o asimétricas. En el primer caso, gracias a las propiedades simétricas, el Kernel se puede reescribir como se muestra en la Ec. 2.20 [29].

$$K_{Simétrico}(x_i, t) = \frac{1}{h} K\left(\frac{x_i - t}{h}\right) \quad (2.20)$$

Donde se tiene que el parámetro h es llamado ancho de banda. Este parámetro permite controlar el grado de suavidad que tendrá la curva de la distribución de densidad con respecto a los datos a los que se debe ajustar.

Cuando se aplica un ajuste por función de estimación de densidad con Kernel a un set de datos $[x_1, x_2, \dots, x_n]$ que conforman solamente un input, la labor realizada es un equivalente a buscar estimar el histograma porcentual relativo a ese set de datos. Considerando esto, la utilización de este método para calcular funciones de probabilidad, como lo puede ser una DTR, podría ser vista como redundante, considerando el hecho que para calcular esta ya se tiene un método que hace esa labor directamente con Montecarlo. Sin embargo, se ha decidido tratar como un método por separado por dos motivos relativos a estudios preliminares: (1) La utilización de este método conlleva un menor tiempo de calculo que el método de Montecarlo, y (2) Requiere una menor cantidad de números aleatorios a generar para lograr grados de precisión similares.

2.3. Definiciones Matemáticas relevantes en redes neuronales

Anteriormente ya se mencionaron los conceptos de función de pérdida y función de activación, siendo la primera la función que mide la diferencia existente entre los valores teóricos que se conocen y los valores que han sido predichos por la red, mientras que la segunda agrupa todas las funciones que se usan como transformadas a la salida de las neuronas con el fin de generar los cambios suficientes en los datos para que se pueda hacer la predicción. A continuación, se hablan de las más importantes que se estudiaron durante en el desarrollo de esta tesis.

2.3.1. Funciones de Activación

2.3.1.1. Función de Activación tipo sigmoide

Esta función esta definida en todos los reales por la Ec. 2.22 [30]:

$$P(t) = \frac{1}{1 + e^{-t}} \quad (2.21)$$

La salida de esta función esta acotada entre 0 y 1, siendo estos dos puntos excluidos. La forma que tiene esta función se muestra en la Fig. 2.9.

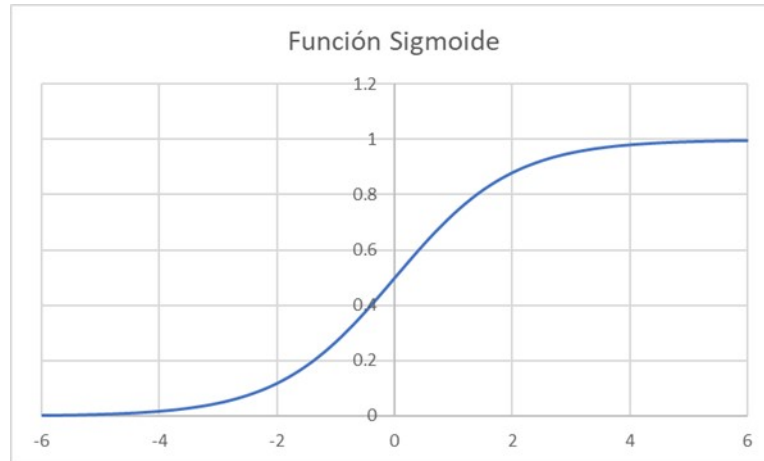


Figura 2.9: Gráfico de función sigmoide

Se puede ver como esta función esta acotada a un espacio muy limitado en su dominio de entrada, teniendo valores con variación importante solo entre los valores de -4 y 4 en el eje X aproximadamente, y con pendientes simétricas con respecto al eje Y. Esta función es útil para aquellos casos donde los valores que se desean simular son de ordenes de magnitud bajos, o cuando no existe mucha variabilidad en los inputs del modelo, ya que esta representa bien la variación existente en puntos que son cercanos entre sí.

Una utilidad que presenta la función sigmoide es para los procesos de clasificación de variables. Si se desea clasificar una variable en función de un set de datos con los que se cuenta, se puede hacer uso de la función sigmoide a la salida del modelo de redes neuronales. En primer lugar, se debe categorizar la base de datos de entrenamiento de manera binaria, es decir con 0 y 1. Al hacer uso de la función sigmoide a la salida de la red, esta presenta valores entre esos mismos números, por lo que, por lo general, se toma el punto medio de 0.5 como umbral bajo el cual se clasifica como 0, mientras que sobre él se clasifica como 1. Teniendo esto en cuenta, es importante destacar la limitante que esto solo puede ser hecho con bases de datos que cuentan solo con dos tipos de datos. Sin embargo, es posible utilizar esto para el caso donde se cuente con más tipos de datos, mediante el uso de árboles binarios, técnica que se explica con posteridad.

Como se explico anteriormente, las funciones de activación deben ser derivables en todo su dominio, con el fin de poder aplicar el algoritmo de backpropagation en las neuronas. Considerando esto, se puede ver que la función sigmoide cumple con esta característica, y que además su derivada es una función simple.

$$\frac{dP(t)}{dt} = \frac{e^{-t}}{(1 + e^{-t})^2} \quad (2.22)$$

2.3.1.2. Función de Activación tipo ReLU y Leaky ReLU

Esta función, también llamada Rectificador, está definida en todos los reales por la Ec. 2.23 [30]:

$$P(t) = \max(0, t) \quad (2.23)$$

La forma que tiene es mostrada en la Fig. 2.10.

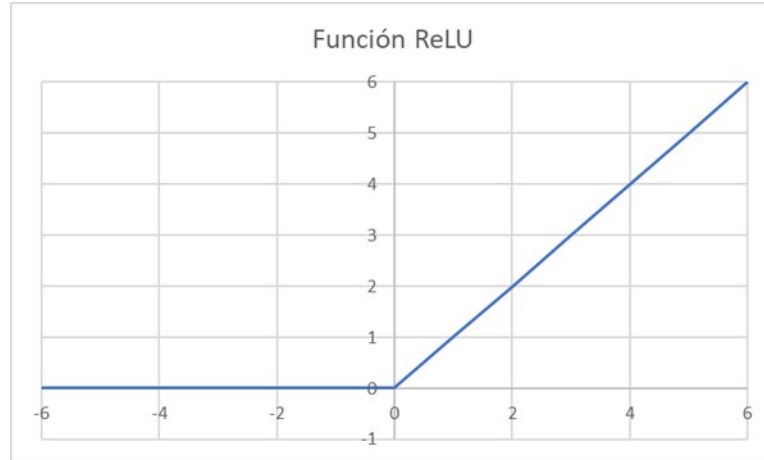


Figura 2.10: Grafico de Función ReLU

Los valores de ingreso a una red creada con este método pueden ser los valores directos, o las desviaciones provenientes de la etapa de normalización. A diferencia de la función sigmoide, esta no está acotada a superiormente, pero si inferiormente, siendo 0 su valor mínimo, alcanzado en todos los valores reales negativos. Teniendo esto en cuenta, no presenta una utilidad directa para ser usada en algoritmos de clasificación como función de salida. Por otro lado, esta función ha demostrado ser de gran utilidad para ser utilizada como función de activación de neuronas internas, tanto para algoritmos de clasificación, como para algoritmos de regresión. La particular definición que posee esta función permite que a la salida de las neuronal solo se presenten valores positivos, lo que resulta útil cuando todos los valores de entrada se mueven dentro de ese dominio. Por otro lado, al presentar una derivada simple de calcular, facilita el cálculo del algoritmo de Backpropagation.

Por otro lado, presenta una desventaja en casos donde se trabaja con posibles inputs negativos, como pueden ser los valores normalizados. Se debe recordar que inicialmente los valores de los pesos y sesgos asociados a una neurona son valores aleatorios. Si se da el caso en que la combinación de variables, pesos y sesgos da como resultado solo valores negativos, la red entera se anula, entregando únicamente valores de 0, situación que si se dio durante el desarrollo de esta tesis. Para evitar esto, se puede utilizar versión modificada de la función ReLU original, llamada Leaky ReLU [30], la cual tiene por expresión la Ec. 2.24.

$$f(x) = \begin{cases} x & \text{si } x > 0 \\ 0.01 \times x & \text{si } x \leq 0 \end{cases} \quad (2.24)$$

Esta función elimina el riesgo que los valores generados por los posibles números negativos sean absorbidos por la función y que se pueda perder información en la construcción de la red.

2.3.1.3. Función de activación tipo tangente hiperbólica

Esta función esta definida en todos los reales por la Ec. 2.25 [30]:

$$P(t) = \frac{e^t - e^{-t}}{e^t + e^{-t}} \quad (2.25)$$

La salida de esta función esta acotada entre -1 y 1, siendo estos dos puntos excluidos. La forma que tiene esta función se muestra en l Fig. 2.11.

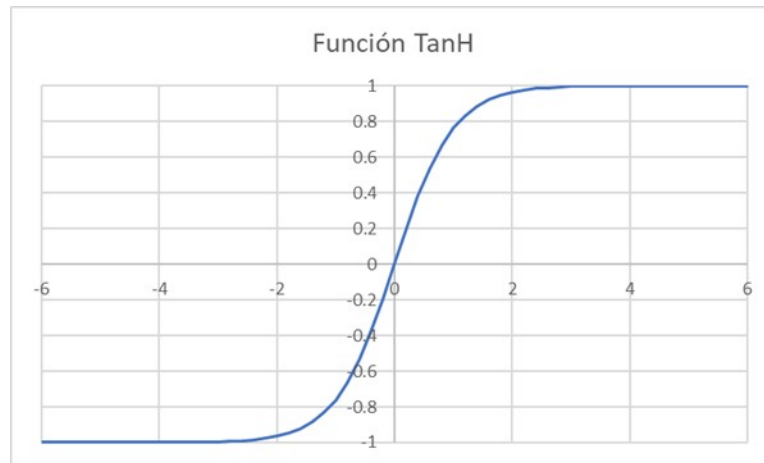


Figura 2.11: Grafico de función Tangente hiperbólica

Se puede ver que tiene un comportamiento similar al que presenta la función sigmoidea, con la diferencia que esta si tiene la posibilidad de entregar valores negativos, acotados hasta un máximo y mínimo no alcanzables de 1 y -1. Además, el dominio en el que la función posee valores diferenciados entre si es más estrecho, yendo aproximadamente desde -2 hasta 2. Su utilidad es similar a la de la función sigmoidea, con el alcance que también permite hacer predicciones negativas, lo que es una ventaja en general cuando se hacen predicciones de variables normalizadas.

Una de las principales formas de utilizar la función tangente hiperbólica es en algoritmos de clasificación. Dado que la salida de esta función puede aproximarse a tres valores enteros, puede ser usada para la clasificación directa de tres o dos variables. Puede combinarse con funciones tipo ReLU o Leaky ReLU, para de esta forma recuperar los valores negativos que se pierden con esta última.

La derivada de esta función, mostrada en la Ec. 2.26 también es simple de implementar en un sistema de optimización.

$$\frac{dP(t)}{dt} = \frac{4 \times e^{2t}}{(e^{2t} + 1)^2} \quad (2.26)$$

2.3.1.4. Función de activación tipo softmax

La función de activación softmax presenta ciertas peculiaridades que las diferencias de las demás funciones de activación. Por regla general, solo suele ocuparse en la salida de un sistema de redes, y su uso está enfocado al de los algoritmos de clasificación. La ecuación que la expresa considera la existencia que a la salida de la capa final existe un número n específico de neuronas, que debe coincidir con el número de categorías que se tienen. Para la neurona j -ésima, la ecuación de la función softmax se puede escribir como se muestra en la Ec. 2.27 [30].

$$P(t_j) = \frac{e^{t_j}}{\sum_{k=1}^n e^{t_k}} \quad (2.27)$$

Como se puede apreciar, esta función hace que las salidas de la capa estén conectadas entre sí por medio de un algoritmo de normalización. Esto implica, además, que las salidas de esta capa serán números entre 0 y 1, los que sumarán un total de 1. Con esto en mente, esta salida puede ser interpretada como una distribución de probabilidades de pertenencia del dato a clasificar a cada una de las posibles categorías. Además, el uso de una función exponencial implica que cuando la salida de la neurona es un valor más elevado con respecto a los demás, esto incrementará aún más la distancia, lo que se interpreta como un alto grado de certeza por parte del algoritmo. Con esto, se debe añadir algún algoritmo que permita elegir la categoría que se debe asignar a la predicción. El método natural es elegir aquel que presente una mayor probabilidad.

Dada la descripción presentada de cómo funciona la función softmax, esta requiere que la base de datos de entrenamiento sea diferente a las demás que se usan para hacer clasificación y categorización de datos. En este caso, lo que se requiere es que se tengan como datos de salida un vector de largo equivalente al número de categorías existentes, y que en el índice correspondiente al valor al que pertenece se ingrese un número 1, mientras que todos los demás valores deben tener un valor de 0.

2.3.2. Funciones de Perdida

2.3.2.1. Función de pérdida de error cuadrático medio y derivadas

La función de pérdida más común entre dos sets de variables es la función del error cuadrático medio (MSE). Si se tienen dos sets de datos X e Y , cada uno con un número igual de datos n , y con un set ordenado (x_i, y_i) , con $i = 1, 2, \dots, n$, se tiene que la ecuación que define el MSE esta presentada en la Ec. 2.28.

$$MSE(X, Y) = \frac{1}{n} \sum_i^n (X_i - Y_i)^2 \quad (2.28)$$

Se puede ver que esta formula calcula la diferencia entre cada par de datos del set ordenado, y la eleva al cuadrado para tener números positivos. Es el método más simple, pero presenta una limitante: No es posible identificar de forma directa cuales son los puntos que

generan un mayor problema, por lo que no se puede idear una estrategia para buscar trabajar con mayor fortaleza en esos puntos.

Derivada de esta, se puede deducir una función de pérdida con una estructura similar, que es el error absoluto medio (MAE), el cual en vez de definirse con las diferencias al cuadrado del set ordenado, se define con el valor absoluto, tal como se muestra en la Ec. 2.29

$$MAE(X, Y) = \frac{1}{n} \sum_i^n |X_i - Y_i| \quad (2.29)$$

2.3.2.2. Función de Pérdida de Kullback Leibler

Para utilizar esta métrica de comparación, se debe considerar como supuesto que se están comparando dos distribuciones de datos que distribuyen de manera gaussiana. Teniendo en cuenta esto, existe una medición denominada divergencia de Kullback–Leibler, la cual se define como una distancia entre dos distribuciones. Esta divergencia es menor si dos distribuciones son más parecidas y vale 0 cuando se compara dos distribuciones que son idénticas [31].

Para dos distribuciones normales multivariadas Y_1 e Y_2 , cada una con sus respectivas funciones de densidad $S_1(x)$ y $S_2(x)$, la divergencia de Kullback-Leibler puede ser escrita como se muestra en la Ec. 2.30.

$$D_{KL}(Y_1||Y_2) = \int_{-\infty}^{\infty} S_1(t) \times \ln \frac{S_1(x)}{S_2(x)} dx \quad (2.30)$$

2.3.2.3. Función de Pérdida Binary Cross Entropy

Esta función de pérdida es la única de las estudiadas que está definida únicamente para ser usada en algoritmos de clasificación de datos. Sin embargo, solo puede ser usada para separar dos sets de datos distintos, los cuales deben estar categorizados de manera binaria. La ecuación que rige esta función de pérdida es la presentada en la Ec. 2.31.

$$H(p, q) = - \sum_i p_i \ln(q_i) \quad (2.31)$$

Esta función estudia las probabilidades de pertenencia a una categoría o no, motivo por el cual solo puede ser usada de forma binaria. El funcionamiento básico que posee esta función consiste en comparar el dato en cuestión con la base de datos de entrenamiento, buscando similitudes a su alrededor, para, posteriormente, aplicar la función de pérdida mostrada. El resultado entregado es la probabilidad de pertenencia a la categoría asignada con 1. Con esto en mente, es necesario aplicar algún algoritmo de rectificación de datos. Por lo general, se impone un valor límite, usualmente 0.5, bajo el cual se considera que el dato pertenece a la categoría 0, mientras que, si es mayor, se implica que pertenece a la categoría 1.

El principal problema que presenta el uso de esta función de pérdida es que la clasificación no es precisa en las zonas intermedias, es decir, zonas que rodean el valor de 0.5, lo cual implica una dificultad en su implementación cuando se busca usar en bases de datos donde la diferencia entre categorías es probabilística y no está totalmente clara.

2.4. Programacion de modelos de caja negra

Los modelos de redes neuronales debe ser diseñado en un lenguaje de programación que sea combinable con modelo fenomenológico base que ya se ha creado. En este caso, se ha recurrido a utilizar Python. Dentro de las librerías estudiadas, se ha probado con la utilización de Tensorflow, y más específicamente, Keras. Otra librería de apoyo es Pandas, que permite importar datos directamente desde Excel para ser procesados y utilizados en la creación del modelo de Deep Learning.

Las variables que pueden ser manipuladas a la hora de construir el modelo de redes neuronales son las siguientes:

1. Número de capas ocultas y neuronas: Es posible asignar el número de capas que se desea que tenga el modelo, además del número de neuronas por capa, lo que permite establecer de forma indirecta el número total de neuronas y de parámetros por optimizar.
2. Función de activación: Es posible asignar el tipo de función de activación que se desea aplicar a la salida de cada una de las neuronas de una capa específica, lo que permite acotar el valor de salida de las neuronas.
3. Número de repeticiones de entrenamiento o *epochs*: Es posible asignar la cantidad de veces que se desea optimizar los valores de los parámetros que definen la red neuronal. Por cada repetición, el algoritmo modifica ligeramente el valor del sesgo y de los ponderadores de cada neurona, buscando aproximarse lo más posible a los datos de entrenamiento.
4. Ratio de Aprendizaje: Es posible asignar el valor objetivo de la similitud en la representación del modelo frente a los datos de entrenamiento [32].

Se deben realizar un número extenso de pruebas para poder encontrar la combinación ideal de los parámetros indicados anteriormente, con el fin de que el modelo a entregar sea lo más cercano a la realidad. El tiempo que dura el entrenamiento es variable, dependiendo de la cantidad de neuronas, del tamaño de la base de datos de entrenamiento y del método usado en el algoritmo de backpropagation para optimizar los valores de los sesgos y los pesos.

2.4.1. Construcción de una red neuronal

La construcción de las redes neuronales requiere, en primer lugar, conocer la estructura base de la función que se quiere representar, lo que incluye conocer los rangos de valores de entrada y de salida de la función. También es deseable conocer aproximadamente el comportamiento esperado en la función, es decir, conocer su regularidad, crecimiento, mínimos y máximos. Esto con el fin de encontrar la estructura interna más óptima.

Es relevante recalcar que una red más compleja, es decir, que posea un número mayor de neuronas y/o capas, no es necesariamente mejor. Un exceso de parámetros puede ser contraproducente y complicar el cálculo de las variables deseadas. En ocasiones, si estas han sido configuradas correctamente, las redes en sí mismas tienden a eliminar neuronas que no han sido útiles, dejando todos sus ponderadores con valor de cero.

Keras es la librería de Tensorflow que usado para crear las redes en esta tesis. A diferencia de los lenguajes más tradicionales y directos, Keras no programa la estructura de la red a nivel de neuronas, sino que a nivel de capa. Esto implica que al crearse una capa específica, todas las neuronas que la componen poseen los mismos parámetros en cuanto a función de activación y ratio de aprendizaje. Esto tiene la ventaja que facilita mucho la programación de redes que poseen una gran número de neuronas y las hace más ordenadas.

2.4.1.1. Programación de Red en Keras

Los pasos por seguir para programar una red en Keras son los siguientes:

1. Definir la capa de entrada: Se debe definir cuál es la capa inicial que ingresa al sistema. Es importante que la definición incluya explícitamente cuál es el orden en que las variables ingresan, puesto que esta debe ser concordante con la obtenida con el modelo.
2. Definir el número de capas y número de neuronas por capas: Se debe definir el número de capas que se presentan en paralelo. También, se define el número de neuronas que tiene cada capa. Este valor puede ser diferente para cada una. Por lo general, se hace un estudio con el fin de encontrar la cantidad de neuronas que maximice la precisión del modelo. Por otro lado, una cantidad muy elevada de neuronas puede ser redundante, complicar innecesariamente el modelo y elevar el tiempo de cálculo.
3. Definir las funciones de activación que se tendrán entre cada capa: Cada capa que se tiene, aplica una función de activación a la ponderación lineal que se calcula al ingreso de las señales de entrada. Esta función puede tener distintos dominios, considerando el comportamiento y distribución de los valores esperados en la señal de salida. Para funciones con valores pequeños se recomienda el uso de funciones sigmoideas o hiperbólicas, que tienen dominios que van entre 0 y 1, y -1 y 1 respectivamente. Por otro lado, se recomienda el uso de funciones tipo ReLU en caso de que se tengan dominios mayores.

Al programarse las redes en Keras, se cuentan con las siguientes limitantes generales:

1. Dentro de una misma capa, debido a la definición de capa que presenta Keras, solo es posible usar una sola función de activación.
2. Las redes generadas por este método son del tipo *Fully Conected*, es decir, todas las neuronas de una capa están conectadas con todas las neuronas de la capa siguiente.
3. Se contó con un número limitado de funciones de activación y funciones de pérdida, dado el poco nivel de profundidad que presenta esta librería.

A continuación, se muestra como se traduce esto en términos de código para un ejemplo genérico de una red de predicción de DTRs.

```
model = keras.Sequential()
model.add(keras.layers.Dense(50,activation=keras.layers.LeakyReLU(),
input_dim=N_Celdas))
model.add(keras.layers.Dense(50,activation=keras.layers.LeakyReLU(),))
model.add(keras.layers.Dense(50,activation=keras.layers.LeakyReLU(),))
model.add(keras.layers.Dense(50,activation=keras.layers.LeakyReLU(),))
model.add(keras.layers.Dense(600,activation='sigmoid'))
```

2.4.1.2. Optimización de Red

Una vez armada la estructura base, se debe ingresar el set de entrenamiento. El proceso de optimización de los pesos y de los sesgos es un proceso iterativo y automático:

1. Inicialmente el sistema asigna una serie de valores aleatorios a los pesos y sesgos. Con eso, calcula la salida y la compara con la esperada.
2. El sistema recorre la red en retroceso, desde las salidas hasta las entradas, optimizando los valores de los sesgos y los pesos de tal forma que se minimice la diferencia entre la salida esperada y la obtenida. El porcentaje de mejora esperado en este paso se define inicialmente en el valor del ratio de aprendizaje.
3. Ambos pasos son repetidos un número definido de veces, valor representado en el número de repeticiones de entrenamiento.

Con lo anteriormente explicado, se puede ver que el sistema calcula automáticamente cuáles son los valores dentro de las neuronas que optimizan el predictor. Sin embargo, existen varios parámetros al momento de construir una red que deben ser definidos por el mismo programador, tales como el número de capas, el número de neuronas que posea cada una y las funciones de activación, así como los ratios de aprendizaje y el número de repeticiones de entrenamiento. Por otro lado, la forma de calcular el error entre la predicción realizada y los datos reales a la hora de realizar el algoritmo backpropagation no es única.

A continuación, se muestra como se traduce esto en términos de código para un ejemplo genérico de entrenamiento y optimización una red de predicción de DTRs.

```
model.compile(loss='mse',optimizer=tf.keras.optimizers.Adam())
model.summary()
Historial=model.fit(Taus, DTRS,epochs=1500, verbose=1,)
```

2.4.2. Algoritmos auxiliares en el uso de redes neuronales

Para la utilización de redes neuronales, en ocasiones es necesario el uso de algoritmos auxiliares, que permiten procesar los datos con el fin de adaptarlos a las necesidades del modelo o transportar las predicciones a un dominio más adecuado.

2.4.2.1. Normalización de datos

Tal como se mencionó en un principio, en la predicción por medio de redes neuronales, es inusual utilizar los datos de manera directa. Lo más normal es que sea necesario hacer una etapa de preprocesamiento con el fin de adaptarlos a un dominio que sea más sencillo de predecir. Esto se hace debido a que es posible que las variables de entrada o de salida del modelo pueden presentar una gran variabilidad en cuanto a los valores en sí mismos, e incluso, en cuanto a los órdenes de magnitud.

Dado esto, el procedimiento estándar a seguir es tratar los datos previamente por medio de un proceso de normalización de datos. Esto consiste en restar el dato ingresado con el valor promedio de todos los datos en la base de entrenamiento y posteriormente dividirlo por la desviación estándar. El hacer esto implica que no es el valor directo lo que se busca predecir, sino que la distancia que se tenga hasta los valores promedio. Esta distancia suele ser un valor bastante más acotado que el valor del dato en sí mismo, lo que ayuda a que la

predicción pueda ser realizada con mayor facilidad y exactitud.

Se puede hablar en términos generales que deben existir dos etapas de procesamiento de datos:

1. Normalización de entrada: Esta consiste en normalizar cada uno de los datos de entrada con respecto a las medias y las desviaciones estándar correspondientes obtenidas desde la base de datos de entrenamiento.
2. Desnormalización de salida: Como ya se mencionó, la salida de una red neuronal planteada correctamente ya está normalizada, por lo que hay que hacer el proceso inverso con cada una de ellas para obtener el dato predicho por el sistema. En este caso el cálculo correspondería multiplicar por el valor de la desviación estándar correspondiente y posteriormente sumarle la media

En la Fig 2.12 se presenta un esquema que muestra el proceso general que se debe realizar con los datos que se desean usar en el modelo:

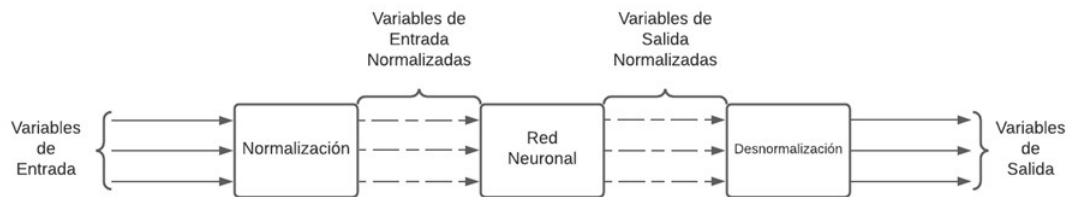


Figura 2.12: Esquema de algoritmo de normalización

La ecuación utilizada para hacer la normalización de datos es la presentada en la Ec. 2.32.

$$X_N = \frac{X - X_{Promedio}}{\sigma_X} \quad (2.32)$$

Para poder realizar esta operación correctamente es necesario que la base de datos que se tenga sea representativa y sin sesgos, dado que de estos valores se obtiene el promedio y la desviación estándar que son utilizados en la ecuación previamente mostrada. Si no se cuenta con una base representativa, es posible que la red realice predicciones sesgadas del fenómeno a representar, lo que generaría un modelo no útil para su uso.

El fundamento al aplicar el algoritmo de normalización y desnormalización es que ya no se predice directamente un valor a partir de otro, sino que se predice la distancia o separación desde un valor promedio, lo que ayuda disminuir la variabilidad de los parámetros de entrada, dejándolos en un entorno más acotado, situación que es útil al momento de aplicarse un sistema de redes neuronales.

2.4.2.2. Aplicación de vectores de latencia

Un vector de latencia se define como un vector en el cual todos sus componentes son ruido aleatorio que distribuye de alguna manera específica. Por lo general, se suele usar un ruido de tipo Gaussiano para no tenga sesgos. Estos están pensados para servir de apoyo en muchos

aspectos de las redes neuronales.

Uno de los principales puntos de apoyo es poder aumentar el tamaño del input que ingresa en la red si es que esta tiene muy pocos valores en la capa inicial. Esto es porque de esta manera puede mantener encendidas algunas conexiones neuronales que podrían apagarse si solo hay una variable de ingreso. Por otro lado, al ser solo ruido aleatorio, se espera que no afecte negativamente la predicción, ya que esta no debería estar sesgada hacia ningún valor en específico.

2.5. Definición y caracterización de UGMs

Para el desarrollo de esta memoria se hace importante mencionar el concepto de UGM, puesto que este es muy relevante a la hora de definir la cinética dentro de una celda. La sigla UGM significa unidad geometalúrgica, y hace referencia a un dominio o espacio dentro de un yacimiento que es lo suficientemente homogéneo en cuanto a composición mineralógica, geología y alteración, como para predecir que cualquier actividad minera o metalúrgica que se realice con material extraído desde ese sector, tendrá un comportamiento similar e invariable en el tiempo [33]. En la actualidad minera, se le ha dado una gran relevancia al trabajo en minería desde una perspectiva geometalúrgica, ya que se ha descubierto que tener en cuenta desde un principio este cruce de informaciones permite tener mayor control en el proceso completo, y optimizarlo con el fin de tener una minería más eficiente y sustentable.

Por lo general, al momento de realizar la exploración minera de un yacimiento y construir el modelo de bloques, una de las informaciones más relevantes a catalogar es el dominio geológico. Esto consiste en separar el yacimiento en zonas según su alteración geoquímica, tipo de roca, condiciones de formación y ubicación espacial específica. La mineralogía de la zona también está incluida dentro de esta clasificación. De esta forma, inicialmente, se pueden estudiar las leyes de cada dominio y, a partir de esto, elegir y separar aquellos que se evalué son económicamente explotables, de aquellos que no lo son.

Paralelamente al trabajo de planificación de la extracción mencionado en el párrafo anterior, se deben realizar las pruebas metalúrgicas en los distintos dominios del yacimiento. Se realizan pruebas de flotación batch con el fin de conocer el comportamiento esperado en la etapa de flotación, determinando durante el proceso las constantes cinéticas de las especies de interés. Además, se pueden hacer estimaciones de la recuperación de ganga que se podría tener en el proceso, para tener una idea base de las leyes de mineral de interés que se podrían obtener en los concentrados. Además de las flotaciones batch, se realizan otras pruebas metalúrgicas, como pruebas de dureza para los chancadores y los molinos, pruebas de abrasividad y pruebas de sedimentación para los espesadores.

Conociendo los valores de las pruebas metalúrgicas en general, se hace un cruce de información con la información geológica que divide el yacimiento en dominios. Se busca agrupar aquellos dominios con mineralogías y resultados en pruebas metalúrgicas similares en una sola unidad. A estas unidades se les denomina unidades geometalúrgicas [33]. Por lo general, para realizar este agrupamiento se requiere del juicio de todos los involucrados en el proceso minero, como los geólogos, ingenieros de minas e ingenieros metalurgistas.

Hay que tener en cuenta que una Unidad geometalúrgica no es totalmente homogénea al momento de ser extraída, ya que pueden existir diferencias innatas dentro de ella, dado el proceso de formación de los minerales que se puedan encontrar ahí. Lo que es importante para definirla es que al momento de ser explotada, extraída, procesada y concentrada no existan diferencias sustanciales en los resultados obtenidos a una escala relevante para la planificación y control de procesos. Por ello al diseñarse la estrategia de explotación de una UGM en específico, se suelen utilizar los valores promedios para mineralogías, leyes y otras variables operacionales, y luego se añade una cierta variabilidad para tener resultados y estrategias de trabajo más robustas.

Con esto en mente, para poder simular la flotación de una UGM en específico se debe contar con la siguiente información:

- Leyes de Minerales: Son las leyes de todas las especies de minerales que se pueden encontrar en una UGM específica. Por lo general, se consideran solo las especies que contengan el elemento de interés. La ganga puede considerarse como una única especie, a menos que su comportamiento afecte de alguna manera la recuperación.
- Constantes cinéticas de las especies: Se debe conocer la constante cinética de las especies por separado, para ponderar la recuperación que tiene cada una en el proceso de flotación.

En una planta de flotación, por lo general, no se conoce cuál es la mineralogía exacta que está alimentando en las celdas. Lo más común es conocer cuál es la UGM o mezcla de UGM que se está alimentando. Esta información viene de la zona de los acopios de mineral, donde generalmente el material se conserva en zonas separadas según su procedencia. Esta información no suele tenerse en tiempo real, sino que con un retraso que depende de los instrumentos con los que la planta cuenta, por lo que se considera importante incorporar alguna herramienta que permita hacer frente a este retraso.

Capítulo 3

Metodología

Para el desarrollo de esta tesis se ha planteado un trabajo en tres etapas principales: (1) Desarrollar un modelo fenomenológico en un ambiente de programación que permita realizar simulación en tiempos continuos; (2) Desarrollar un modelo de redes neuronales asociado al modelo fenomenológico; (3) Estudiar el orden de implementación que deben tener ambos modelos con el fin de que el modelo de caja gris creado sea lo más representativo posible a la realidad.

3.1. Desarrollo del modelo fenomenológico

3.1.1. Definición de variables pertinentes

Las variables que ingresan al modelo se pueden dividir en dos clases principales: (1) Las variables que pueden ser definidas externamente por el usuario, y (2) Las que están definidas internamente al momento de ser programadas. Si bien ambos tipos de variables pueden ser modificadas por el usuario, las primeras son un parámetro que necesariamente deben ser incorporadas al inicio del trabajo, mientras que las segundas tienen un valor estándar. La principal diferencia desde un punto de vista operacional es que las primeras representan a aquellos parámetros que, en una planta de flotación, están en constante variación, mientras que las segundas representan a aquellos que suelen ser fijos en una operación normal.

Con respecto al primer tipo de variables mencionadas, es decir, las que están en constante cambio, se pueden nombrar las siguientes:

1. Flujo de Alimentación de Sólidos: Es el flujo de alimentación de mineral sólido y seco que alimenta la planta en un instante determinado. La unidad utilizada en este caso para medirlo es Tonelada/Día.
2. Vector de Mineralogía de Alimentación: Este vector es el que indica las leyes de los minerales que ingresan en la planta en la UGM específica que se esté procesando.
3. Vector de Leyes de Cobre: Este vector contiene las leyes de cobre de cada uno de los minerales contenidos en el Vector de mineralogía.
4. Alturas de Espuma: Es la altura de espuma que posee cada celda dentro del banco. Este valor es relevante al momento de calcular el volumen efectivo de cada celda y puede ser usado como parámetro para definir la recuperación en espuma.

5. Flujos de Aire: Es el flujo de aire inyectado dentro de la celda y que se usa para estimar el porcentaje de aire existente dentro, valor que es utilizado a la hora de definir el volumen efectivo.
6. Número de Celdas: Corresponde al número de celdas que forma el banco, y es un parámetro que define la cantidad de veces que el cálculo de recuperaciones debe ser realizado por su cuenta.

Por otro lado, dentro las variables que se mantienen constantes en el tiempo se pueden encontrar las siguientes:

1. Tamaños de las celdas: Corresponde a la altura y el área transversal de las celdas de flotación que componen un banco. Se puede tener la opción que todas estas sean iguales, o que sean diferenciados según la posición dentro del banco.
2. Parámetros cinéticos: Son los parámetros cinéticos de modelamiento que poseen los modelos de García Zúñiga para calcular la recuperación, tales como las constantes cinéticas y los valores de R infinito.
3. Concentración en Peso de la alimentación: Es el porcentaje de sólidos que posee la alimentación con respecto al flujo completo. Permite deducir el flujo de líquidos con el que se está alimentando la celda.
4. Densidades: Son las densidades de todas las especies que contiene el flujo de alimentación.

Por otro lado, las principales variables de respuesta que se buscaron modelar son la recuperación de las especies de interés y la ley de cobre. Ambas pueden obtenerse de los modelos que se presentan a continuación, los cuales pueden poseer una naturaleza fenomenológica, de caja negra o de caja gris.

3.1.2. Cálculo de DTR por método de Montecarlo

La distribución de tiempos de residencia requirió el uso de un método de Montecarlo para poder ser representada en primeras instancias. A modo de resumen, un método de Montecarlo consiste en generar un set de números aleatorios que sigan una cierta distribución predefinida, con el objetivo de utilizar estos datos como base para recrear un fenómeno. En este caso en específico, la creación del modelo para distribución de constantes cinéticas siguió los siguientes pasos:

- I. En el caso de ser una sola celda, se calcula el tiempo medio de residencia, que, como se explicó anteriormente en la Ec. 2.9, es el cociente entre el volumen efectivo de la celda y el flujo volumétrico alimentado, denominado τ_{Medio} .
- II. Se genera un set de datos para obtener la distribución de tiempos de residencia. Estos datos aleatorios deben seguir una distribución exponencial, cuyo valor característico lambda sea equivalente a $[1/\tau_{Medio}]$, tal como se muestra en la Ec. 2.8. El número de datos a generar depende del usuario. Un número de datos bajo genera un modelo con bajo tiempo de cómputo, pero con poca precisión.
- III. Se genera un set de intervalos lo suficientemente grandes como para contener todos los valores que se hayan generado aleatoriamente. Luego se cuenta cuántos de los datos caen

dentro de cada uno de los intervalos generados, y se normalizan por el número total de datos, obteniéndose de esta manera un histograma normalizado. Este histograma, en este caso, es utilizado como un modelo que indica la probabilidad de que una partícula pase un tiempo en específico en la celda.

- IV. En caso de contar con más de una celda, en el paso II es necesario conocer el tiempo medio de residencia para cada una. Luego, se genera un set de datos para cada celda por separado, y estos se suman, obteniéndose así en cada suma el tiempo total que una partícula modelada pasa dentro del banco. Luego, se repite el paso III.

En la Fig. 3.1 se muestra un esquema que grafica esa labor para el caso de más de una celda. En este caso, el índice m representa el número de datos aleatorios a generar, mientras que el índice n representa el número de celdas a simular y T corresponde al banco en su totalidad.

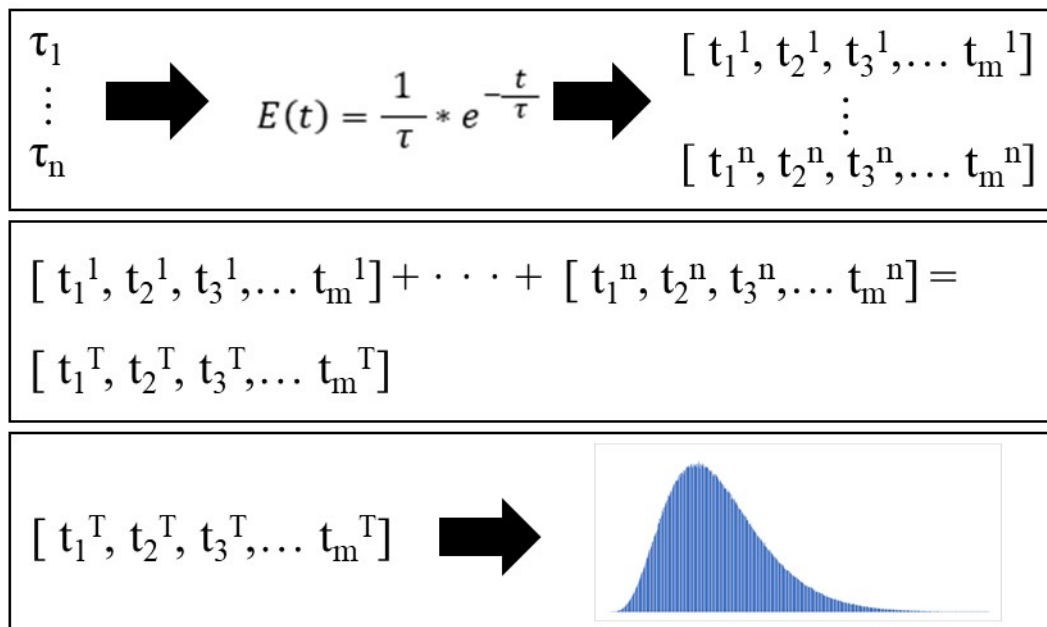


Figura 3.1: Esquema de funcionamiento de Montecarlo para generación de DTRs

Este es el método estándar para calcular las distribuciones de tiempos de residencia cuando se tiene más de una celda. En próximos capítulos se explica como este puede ser reemplazado por otras metodologías pertenecientes al ámbito del Machine Learning y se evalúa el desempeño comparativo.

3.1.3. Discretización del modelo de recuperación

El modelo fenomenológico creado utiliza como ecuación base la ecuación de Polat y Chandler Ec. 2.7. Cada uno de sus términos ha sido adaptado de forma específica para poder ser calculado como una función de Python. Como se recuerda, esta ecuación consiste en una doble integral desde cero hasta infinito. La primera para considerar todos los posibles tiempos de residencia de las partículas, y la segunda para considerar todas las posibles constantes cinéticas. Sin embargo, las funciones de Python que son programadas desde la consola sólo

pueden incluir términos discretos, por lo que se debe discretizar esta función, transformando estas integrales en sumatorias. Al hacer esto, estas sumatorias ya no se consideran hasta infinito, sino que hasta un valor lo suficientemente grande como para que todos los valores numéricos a partir de ese punto puedan aproximarse a cero.

Para el caso de esta tesis, se mencionaron los dos principales modelos planteados previamente, que son el de García Zúñiga y el Klimpel. A modo de simplificación del problema, se considero solamente el uso de la distribución de García Zúñiga. Al aplicarle la integral a la definición de esta ecuación, el termino queda con valor de 1 y solo queda representada en el termino exponencial.

Si se aplica la discretización de manera correcta, tomando el supuesto de García Zúñiga con respecto a la distribución de constantes cinéticas, se puede llegar a la expresión mostrada en la Ec. 3.1:

$$R = \sum_{t=0}^T R_{\infty} \times E'(t) \times (1 - e^{-K_{GZ} \times t}) \quad (3.1)$$

El conteo temporal se lleva a cabo un delta de tiempo definido previamente. Esta nueva función es la que ha sido programada para simular la recuperación en una celda de flotación. Los términos por utilizar tienen los siguientes significados:

- T: Es el tiempo hasta el cual se decide calcular el valor de la recuperación. Este debe ser escogido por el usuario del sistema. Por el planteamiento mismo de la ecuación, se puede ver que, a tiempos muy altos, el aporte a la sumatoria se hace muy cercano a cero, por lo que se puede escoger un valor a partir del cual se aprecie que no existe crecimiento en el valor de la recuperación.
- K_{GZ} : Es la constante cinética de García Zúñiga que se usa para representar la facilidad de flotación de todas las partículas de una especie en específico.
- $E'(t)$: Es la versión modificada de la ecuación de distribución de tiempos de residencia, que incluye la opción de modelar más de una celda. Además, se ha implementado de tal forma que el detiene el conteo en el tiempo T.

Esta ecuación modificada es la base a partir de la cual se construye el modelo de una planta de flotación, por lo que resulta crucial comprender completamente los valores que se utilizan y saber escogerlos.

3.1.4. Cálculo de recuperación en un banco

Ya se ha definido el método para calcular la recuperación a partir de una distribución de tiempos de residencia. Sin embargo, el método para obtener el τ_{Medio} no suele ser tan directo en una celda o en un banco.

El modelo global para calcular la recuperación sigue un orden secuencial celda a celda, realizando operaciones iterativas, tanto dentro de una misma celda, como en el banco completo. A continuación, se presenta un paso a paso para realizar el calculo de la recuperación en una celda:

1. Inicialmente se debe calcular el flujo volumétrico total de alimentación a la celda inicial y el volumen efectivo de la celda. Con ello se puede obtener una primera aproximación al τ_{Medio} . Con ella se calcula una primera DTR y desde ella se extrae una primera aproximación de la recuperación mediante la Ec. 3.1. Esta labor es ejecutada por una función que se ha denominado “Caso Base”. Este valor de recuperación esta, por lo general, subestimado, ya que ha sido calculado en base a un tiempo de residencia medio que incluye todas las partículas que ingresan en la celda, en vez de hacerlo considerando solo las de relave, que el método usual.
2. Conociendo los valores de las primeras aproximaciones recuperación para minerales de mena, se puede hacer una estimación del flujo de cada uno de estos en el relave. Sin embargo, también es necesario conocer el valor del flujo en el relave correspondiente a la ganga. Para hacer esto, podría trabajares la ganga como otro mineral más, asignándole su propia constante cinética. Sin embargo, esto no es muy común, ya que no se suelen hacer estudios sobre cinética de flotación de ganga. Otra forma de hacerlo seria mediante el estudio de la flotación por arrastre, pero esto involucraría tener que hacer estudios acerca de la recuperación de agua [34]. Frente a esto, se ha elegido usar un algoritmo auxiliar que permita calcular una aproximación de ley de concentrado a partir de la ley de alimentación, con el fin de aportar la información necesaria para calcular el flujo de ganga en el relave. Este modelo en primeras instancias fue obtenido mediante un algoritmo de regresión, aunque en etapas posteriores se hicieron pruebas con otro tipo de metodologías.
3. Conociéndose las recuperaciones, se procede a calcular el flujo en el relave de cada especie minera, ya sea ganga o mena. Esto se hace calculado el flujo correspondiente por la expresion $(1 - R)$.
4. Una vez calculados todos los flujos en el relave, se vuelve a calcular una segunda aproximación al τ_{Medio} de la celda, esta vez considerando el flujo volumétrico total de relave. Con este valor, se calcula una nueva DTR y con ella se calculan los valores de recuperación corregidos. Con estos valores de recuperación el proceso es repetido hasta que el valor de la recuperación converja a de la iteración anterior. Esta labor es ejecutada por una función denominada “Iterador”.

En la Fig. 3.2 se muestra un diagrama que esquematiza el funcionamiento del algoritmo ya explicado de manera visual.

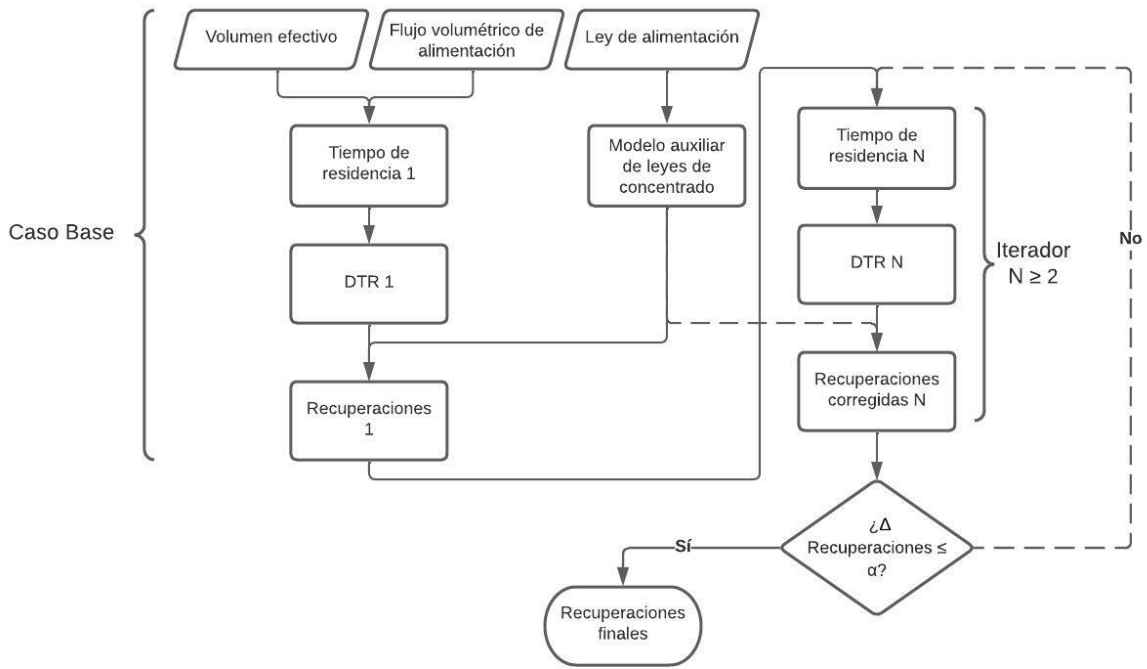


Figura 3.2: Esquema para calculo de recuperación en una celda

Con el método ya explicado es posible deducir la recuperación de una celda. Sin embargo, cuando se presenta el caso de un banco, esta metodología debe ser extendida a las demás celdas. A continuación se presenta un paso a paso para realizar el calculo de la recuperación en un banco de dos celdas:

1. Se deben tener los valores de flujos en relave desde la primera celda inicialmente, obtenidos desde el método anterior. Con ellos se calcula una primera aproximación τ_{Medio} para la segunda celda. Como ya se conoce el τ_{Medio} de la primera celda, es posible calcular una primera aproximación a la distribución de tiempos de residencia del banco.
2. Con esta DTR se calcula la primera aproximación de recuperación del banco. Cabe destacar que esta es realizada viendo el banco como un todo, por lo que los valores de flujo de concentrado que se pueden calcular con las recuperaciones obtenidas, con respecto a la alimentación de la primera celda, corresponde a la suma de los concentrados de todas la celdas, mientras que los valores para el relave corresponden a los valores de la última celda.
3. Conociendo los valores de relave de la ultima celda, puede seguirse un procedimiento similar al explicado en el caso de una sola celda, donde este es utilizado para calcular un nuevo valor τ_{Medio} para la segunda celda y repetir el procedimiento la cantidad de veces necesarias hasta que converja.

Como se puede ver, ya se ha planteado un sistema que permite calcular la recuperación desde un banco de celdas y que permite corregir el hecho que no todas las celdas dentro del mismo poseen el mismo tiempo de residencia medio. Esto permite tener una aproximación más realista. Sin embargo, este sistema también presenta las siguientes dificultades:

- Requiere contar con información que en ocasiones puede ser difícil obtener, específicamente hablando de las constantes cinéticas de flotación para ganga, o información general para predecir las recuperaciones de ganga.
- El sistema presupone que se tiene bien identificada la mineralogía específica que se está alimentando, por lo que se le puede asociar una UGM correspondientes y sus consiguientes constantes cinéticas.

Ambos puntos generan problemas en el correcto modelamiento de la recuperación en una planta de flotación, por lo que se evaluó la opción de usar algoritmos de tipo caja negra para aportar a su solución.

3.2. Análisis de método de implementación

Una vez que se tiene programado el modelo de caja negra y la base del modelo de redes neuronales, se debe trabajar en combinarlos de tal manera que se cree un modelo de caja gris. Es necesario establecer previamente qué grado de intromisión se desea que posea cada modelo, ya que de esta manera se puede definir el grado de profundidad que se le debe asignar a las redes neuronales, o el grado de precisión en la escala temporal que se debe incluir en el modelo fenomenológico.

Con lo anterior, se puede plantear la creación de un modelo con una fuerte base fenomenológica al que se le realicen ajustes con un modelo de redes neuronales. Por otro lado, se puede diseñar un modelo que posea las nociones básicas que entrega un modelo fenomenológico, y usar las redes como metodología para obtener los valores reales. Entre ambas opciones, puede estar la de un modelo que posea una base fenomenológica que actúe paralelamente con un conjunto de redes neuronales para obtener un modelo más balanceado.

Los puntos de aplicación de redes neuronales dentro del modelo fenomenológico se han dividido en tres grupos:

- Redes neuronales de reemplazo: Las redes con esta denominación han sido creadas con el fin de reemplazar algún aspecto o cálculo que está bien definido o planteado dentro del modelo fenomenológico, pero que su implementación implica un gran gasto computacional para el sistema en general, por lo que el uso del método de Deep Learning permitiría reducir los tiempos generales de cómputo del modelo.
- Redes neuronales de soporte: Las redes con esta denominación cumplen la función de representar algún fenómeno físico o químico implicado en el desarrollo del modelo fenomenológico, del cual no se conoce con claridad cuál es la relación directa que involucra a las variables de entrada y salida, por lo que se decide usar el sistema de Deep Learning con el fin de obtener una representación del modelo.
- Redes neuronales de Ajuste: Las redes con esta denominación han sido ideadas bajo el supuesto que es posible que un modelo de caja gris, en general, no pueda representar con total fidelidad el fenómeno deseado. Considerando esto, se han pensado sistemas de Deep Learning de ajuste, cuya función es poder transformar el valor de salida del modelo en un valor lo más cercano posible al real. Para esto, se consideran como inputs,

el valor que se desea corregir o ajustar, y los valores de variables operacionales que sean considerados de relevancia.

Dependiendo del enfoque que se plantee usar durante el desarrollo de esta tesis, es necesario crear bases de datos diferenciadas para la etapa de entrenamiento de la red neuronal. Esto es otro factor por considerar, puesto que probablemente la base de datos con la que se cuente sea limitada, con un número no muy elevado de datos y con variables que no fueron medidas, por lo que se les dará prioridad a aquellos modelos de redes neuronales que sean capaces de ser entrenados con una cantidad mínima de datos.

Tal como se planteó previamente, en esta etapa se planea poner a prueba la hipótesis propuesta al inicio del desarrollo de esta tesis, la que indica que se espera que un modelo que incluye un enfoque fenomenológico y un enfoque de Deep Learning podría ser más preciso y eficiente en cuanto a tiempo de cómputo. Con el fin de demostrar lo anterior, se espera hacer un análisis de la eficiencia del modelo de caja gris frente a uno puramente fenomenológico, que incluya analizar tanto la precisión entregada, como el tiempo de cálculo que le lleva y las limitaciones que podría presentar.

3.3. Modelos de caja negra

3.3.1. Modelos aplicados a calculo de DTRs

3.3.1.1. Motivación de uso

Como ya se explicó anteriormente, el cálculo de las distribuciones de tiempos de residencia requiere del uso un método de Montecarlo, es decir, hacer uso de un algoritmo de números aleatorios, para modelar el caso cuando se tienen más de una celda en serie en un banco. Además, también se mostró anteriormente cómo la cantidad de datos a ser calculados con el método Montecarlo, incide directamente en la calidad del modelo, dado que una simulación hecha con un número elevado de datos presenta una curva mucho más continua y regular que una simulación realizada con pocos datos, aunque esto también implica un aumento sustancial en el tiempo de cálculo. Esto se ve particularmente reflejado al momento de estudiar el tiempo de cálculo del modelo completo a medida que se van agregando celdas en el banco.

El modelo de redes neuronales a usar debe ser un reemplazo completo del modelo de Montecarlo con el que se contaba previamente. Por esto, se debe ser cumplir las siguientes exigencias para que pueda ser utilizado:

1. Debe ser un modelo sustitutivo pero que cumpla la misma función, es decir, debe recibir los mismos inputs que recibía el modelo de Montecarlo, o sea, las distribuciones de tiempos de residencia medios individuales de cada celda; y debe entregar el mismo output, o sea, la distribución de tiempos de tiempos de residencia.
2. La distribución de tiempos de residencia entregada debe ser lo más similar posible a la distribución que se obtendría si se utilizase el método de Montecarlo, tanto en la forma de la curva en sí misma, como en los valores que se obtengan de esta.
3. De ser necesario, se deben incorporar dentro de esta, todos los algoritmos que sean precisos con el fin de ajustar el modelo.

3.3.1.2. Base de datos utilizada

Para el desarrollo de esta etapa, se generaron una serie de bases de datos artificiales, con el fin de comparar el desempeño del método original de Montecarlo con el que podrían entregar los algoritmos caja negra. A continuación, se describe la evolución que fueron experimentando estas bases en relación a las necesidades encontradas por los modelos.

3.3.1.2.1. Base de datos original

Se decidió usar el algoritmo fenomenológico previamente explicado con el fin de generar la base de datos adecuada. Se decidió usar la proporción de 70/30 para dividir la base entre entrenamiento y testeo, ya que es la más habitual en estos casos. Se decidió crear una base de 1000 datos para que esta tuviera robustez suficiente para representar todos los casos deseados.

Para crear la base de datos, se hizo uso del modelo fenomenológico anteriormente mencionado, pero en lugar de que este entregara los valores de recuperación esperados, se programó para que diera como outputs: (1) los valores de los tiempos de residencias medios de las celdas que forman parte del banco, (2) la distribución de tiempos de residencia asociada a esos valores, y (3) otros valores operacionales que se consideraron importantes.

Los valores ingresados en el modelo fenomenológico fueron los siguientes:

- Flujo solido de alimentación: Por cada uno de los 1000 casos que se simularon, se asignó un flujo con valor aleatorio entre 5.000 y 30.000 [tpd]. Esto se hizo con la idea de crear un modelo que fuese capaz de simular plantas con una gran variedad de flujos de alimentación.
- Mineralogía: Como simplificación inicial, se decidió trabajar con un solo mineral de cobre. En este caso, se trató de calcopirita. Por otro lado, se consideró una ganga homogénea y no reactiva, es decir, que no favorece ni castiga la recuperación de calcopirita. Leyes de Alimentación: Por cada uno de los 1000 casos que se simularon, se asignó un valor de ley de cobre aleatorio, con valores entre 0,5 % y 2 %. Igual a como ocurre con el flujo de alimentación, esto se hizo con la idea de tener una base más robusta y que pudiese simular una gran cantidad casos posibles.
- Número de celdas: Por cada uno de los 1000 casos que se simularon, se asignó un número aleatorio de celdas a formar el banco a utilizar, con un número global entre 1 y 3. La idea original es que la red neuronal sea capaz de predecir cualquier valor dentro de un amplio espectro de configuraciones en la planta.
- Datos operacionales variables: Como se mencionó anteriormente, el flujo areal superficial de gas y la altura de espuma son parámetros operacionales que cuentan con la cualidad de ser variables dentro del modelo, es decir, pueden cambiar en periodos cortos de tiempo, y, por ende, se incluyen como inputs. Para este estudio en específico se decidió usar valores de 1 cm/s y 40 cm respectivamente, dado que son los más comunes en plantas reales.
- Datos operacionales variables: Como se mencionó anteriormente, el flujo areal superficial de gas y la altura de espuma son parámetros operacionales que cuentan con la cualidad de ser variables dentro del modelo, es decir, pueden cambiar en periodos cortos de tiempo, y, por ende, se incluyen como inputs. Para este estudio en específico se decidió usar

valores de 1 cm/s y 40 cm respectivamente, dado que son los más comunes en plantas reales.

- Datos operacionales fijos: Estos son los parámetros que dada su naturaleza no pueden ser modificados operacionalmente hablando, por lo que se incluyen como variables internas. Aquí los principales son el área y la altura de la celda $35 m^2$ y 4 m respectivamente, la constante cinéticas de cobre de valor $1,2 [min^{-1}]$ y el Cp de alimentación en 33 %.
- Modelo de estimación de leyes de concentrado: Tal como se mencionó anteriormente, el simulador creado necesita un medio para estimar la recuperación de la ganga desde las celdas. Se barajaron varias ideas sobre cómo incluir esto, las cuales son explicadas con posterioridad. En particular, esta etapa uso un modelo de regresion.

Las distribuciones de tiempos de residencias creadas fueron simuladas hasta los 60 minutos, con un paso de 0,1 minutos, lo que da un total de 601 (Considerando el tiempo inicial $T_0 = 0$) valores que son entregados como outputs.

Con toda la información anteriormente entregada, se puede ver que la idea inicial al momento de crear este sistema de redes neuronales fue que este pudiera simular las condiciones de cualquier planta que tratase un mineral en específico, sin importar el número de celdas, ni los flujos de alimentación. Más adelante se muestra que esta idea no tuvo el éxito esperado, por lo que se debió recurrir a otros métodos.

3.3.1.2.2. Base de datos modificada

Al momento de trabajar con la base de datos anteriormente explicada, se vio que generaba resultados deficientes en cuanto a la precisión, dado que los posibles resultados a entregar presentan una gran variabilidad, situación que no es capaz de reproducirse fehacientemente con una red neuronal. Frente a esto se tomaron las siguientes medidas para mejorar el modelo base:

1. Bases de datos diferenciadas: Tal como se mencionó anteriormente, se abandonó la idea de hacer un modelo único sin importar el número de celdas, para dar paso a modelos diferenciados según este parámetro. Ante esto, surgió la necesidad de crear modelos diferenciados para cada uno de los casos, considerando el hecho que cada red programada por separado necesita el mismo número de datos de entrenamiento y testeo para validar su funcionamiento. Cada base fue creada con rangos similares en cuanto al flujo de alimentación y las leyes de los minerales de interés.
Considerando lo anterior, implica que se debió adaptar un algoritmo que incluyese la opción de seleccionar el modelo a utilizar en función del largo del vector de tiempos de residencia medios, además de contarse con bases de datos de pre procesamiento diferenciadas igualmente por cada número de celdas.
2. Acotar los valores de las variables operacionales en la base de datos de entrenamiento: Se redujo la variabilidad de varios de los parámetros operacionales que definen el proceso. Esto implica que se pierde una representación universal de todas las plantas de procesamiento con un solo modelo, pero es necesario con el fin de tener más precisión. Se contó con criterio experto para definir los nuevos valores límites con los que se pudiera contar. Los flujos de alimentación se generaron aleatoriamente con valores entre el 60 % y el 120 % de un valor nominal que, en este caso, se fijó en 20.000 [tpd]. La ley elemental

se fijó con una variabilidad de entre 70 % y 120 % en torno a un valor nominal de 1 %. Al hacer esto, se disminuye sustancialmente los valores y las formas que alcanzan las DTR, por lo que se espera que se produzca una predicción más precisa.

Este cambio implica que el modelo solo es aplicable a una planta en específico, ya que esas variabilidades son las típicas que se podrían presentar por motivos operacionales y por la propia variabilidad geometalúrgica que presente el mineral a ser tratado.

3. Dado que el tiempo inicial $T_0=0$ es irrelevante ya que se elimina en el calculo de recuperación visto en la Ec. 3.1, y que incluirlo puede traer dificultades extra a la hora de predecir la DTR, se decidió eliminar.

3.3.1.3. Implementación

Obtenida la base de datos, se siguió con el trabajo de implementación de ambos métodos. A continuación, se muestra la labor realizada con cada uno.

3.3.1.3.1. Redes Neuronales

La implementación de este método fue realizándose en etapas, en las cuales se fue avanzando al ir adquiriendo mayor conocimiento y manejo en el área de las redes neuronales.

a. Modelo de predicción único

La estructura base de la primera red neuronal para calcular la distribución de tiempos de residencia consiste en: (1) el vector de ingreso de los tiempos de residencia medios de las celdas del banco, (2) sus consiguientes capas intermedias, sobre las cuales hay que hacer un estudio general sobre la cantidad de capas, el número de neuronas de cada una, y las funciones de actuación a utilizar y (3) la capa de salida que, en este caso, representa los valores de la Distribución de tiempos de residencia.

Se probaron varias estructuras con el fin de ver cual maximiza la precisión. Si se analizan más a fondo la entrada y salida del modelo, se puede ver que la capa inicial contaba con un total de 3 valores de entrada sin importar el número de celdas con los que contase el banco. Si se modelaba un banco con un número inferior de celdas, los valores sin utilizar por tiempos de residencia medios eran llenados con valor de 0. Por otro lado, la capa de salida es de 601 valores, para emular los valores que tendría la distribución de tiempos de residencia a 60 minutos con un paso de 0.1 minutos.

b. Modelo diferenciado según número de celdas

El primer modelo creado presenta claras deficiencias en cuanto a su precisión, lo que indica que este no es útil para el reemplazo del cálculo de Montecarlo que debe ser realizado por el algoritmo en el cálculo de las distribuciones de tiempos de residencia. Ante esto, se decidió investigar posibilidades de mejora de este algoritmo base, con el fin de llevarlo a uno que si tuviese un desempeño más adecuado.

El primer error principal que se pudo notar en el modelo base fue que este fuese único, sin importar el número de celdas. Se puede ver el hecho que los valores de entrada al modelo planteado tienen la característica de ser un valor nominal, deducido como el cociente entre el volumen efectivo y el flujo volumétrico de alimentación a la celda. Sin embargo, al querer representar la presencia de menos celdas que el largo de vector completo, se incurrió en el uso de reemplazar esos valores faltantes con ceros, que tienen un significado más bien categórico,

puesto que un tiempo de residencia de cero no tiene sentido desde su definición.

Al notarse este error en el planteamiento inicial, se decidió que la mejor forma de contrarrestar esto era haciendo uso de modelos diferenciados. Estos modelos implican generar una base de datos por cada uno, entrenarlas y usarlas por separado, siendo cada uno de ellos llamado cuando la situación lo requiera. Cada modelo recibe como input un vector con el número de datos equivalente, y entrega como salida un vector de un Número de datos especificados, que representan la DTR, omitiéndose así el uso de variables categóricas.

El aumento observado tanto en coste computacional como en tiempo de cálculo debería ser mayor al caso de un solo modelo en las etapas de entrenamiento validación y carga de los modelos. Sin embargo, a la hora de la utilización en un ambiente más operacional, este debería ser mínimo, puesto que los únicos pasos extra que se estarían integrando incluyen un árbol de decisión con trabaja con condicionales y la carga de los archivos con los datos para realizar la normalización, algoritmo que se desarrolla a continuación.

c. Modelo normalizado y escalado

Tal como se explicó antes, en el modelo original no se contempló la incorporación de un algoritmo de normalización de datos de entrada ni salida. La ausencia de esta etapa, en discusiones con expertos, se considera como el principal motivo de la poca representatividad de los resultados entregados.

Frente a esto, se decidió incorporar un algoritmo de normalización de datos, tanto para la entrada, como para la salida. Como se mencionó en el punto anterior, al separarse en modelos diferenciados según el número de celdas, esto implica que los datos deben ser tratados diferenciadamente según este parámetro. A continuación, se explica cómo funciona este algoritmo:

1. Se tomaron las bases de datos de entrenamiento para los valores de entrada y de salida para cada uno de los números de celdas a tratar y se calculó la media y la desviación estándar para cada uno de esos valores. Cada una es incorporada en un archivo distinto
2. El algoritmo recibe la variable de entrada, que son los valores de tiempos de residencia medios para las celdas, y selecciona el archivo correspondiente según número de celdas. Desde aquí extrae los valores necesarios de promedios y desviaciones estándar, y se aplica la normalización. En caso de que una desviación estándar tenga valor de 0, la normalización no es realizada. De esta forma la variable de entrada queda redefinida en este punto, siendo esta la que efectivamente entra al sistema de redes neuronales.
3. Se obtienen los resultados desde la red. A estos se les realiza el proceso inverso al explicado. Se llama al archivo correspondiente según el número de celdas, desde donde se extraen cada uno de los 600 valores de medias y desviaciones estándar obtenidas desde los datos de entrenamiento. Se realiza la desnormalización de datos. Estos datos de salida son los que finalmente se entregan como output del algoritmo, y son los que son utilizados para calcular la recuperación.

Además de esto, también se recurrió al uso de un algoritmo de escalamiento. Una red neuronal típica presenta dificultades si se busca trabajar bajo ciertas condiciones específicas. Una de ellas es trabajar en escalas u ordenes de magnitud muy pequeños. Esto significa que

una red no está optimizada para predecir valores que se encuentran en escalas de milésimas o más pequeñas. La explicación que se tiene a esto es que una red busca minimizar el error existente entre los valores predichos y los valores de entrenamiento, por lo que, si ambos son muy pequeños, el error existente entre ellos también lo será, por lo que la red puede interpretar que ya son lo suficientemente parecidos y detenga la optimización.

Esto se cumple muy claramente en este caso. Los valores de la distribución de tiempos de residencia son muy pequeños. Dado que su resultado se interpreta como una probabilidad, deben obligatoriamente tener valores entre 0 y 1. Sin embargo, en la mayoría de los casos, no se logra alcanzar esos ordenes de magnitud, trabajándose normalmente entre los órdenes de magnitud de 10^{-4} a 10^{-7} .

Las opciones que se tienen para corregir esto es el uso del escalamiento de las variables a predecir. Esto significa aumentar artificial y externamente el valor de todos parámetros que son usados para entrenar, para que de esta manera se puedan trabajar en un ambiente de programación que facilite minimizar la diferencia entre datos. Este aumento se realiza mediante una ponderación por algún número en específico, el cual debe ser el mismo a usar en toda la base de datos. Una vez que se haya hecho la predicción, se debe dividir todo por el ponderados que se haya usado y así se tiene el resultado predicho.

Para el caso de esta tesis, se ha decidido ponderar solamente los valores de la salida de la red, puesto que los inputs están en un orden de magnitud que se ha considerado que es lo suficientemente alto como para que se puedan hacer optimización sin problemas. Por otro lado, los outputs han sido ponderados por 10^9 , lo que permite que los valores más pequeños con los que se podría contar estén alcanzan un orden de magnitud que permita trabajar con ellos.

d. Inclusión de vectores de latencia

Como última propuesta de mejora, se decidió probar con la incorporación de vectores de latencia. Para el caso específico de esta tesis se ha decidido que el vector de entrada este compuesto en total de 8 valores, los que incluyen aquellos valores de los tiempos de residencia medios, que son los inputs originales, más los valores del vector de latencia. De esta forma se tiene que los modelos diferenciados reciben toda la misma cantidad de variables, pero se diferencian en cuantas de esas variables realmente representan datos operacionales reales, y cuantos de estos representan ruido aleatorio. Se ha incorporado un algoritmo a los modelos que permite acoplar el ruido aleatorio para dejar el largo del vector en 8, sin importar la cantidad de celdas a representar.

La distribución por utilizar para generar los vectores de latencia es una distribución normal estándar. Esta fue elegida puesto que es sencilla de implementar, el ruido generado presenta una variabilidad adecuada para el problema planteado y los números usados tienen un orden de magnitud similar, lo que impide posibles desviaciones por parte de la red.

3.3.1.3.2. Estimación de Densidad con Kernel

Paralelamente al trabajo realizado con el método de redes neuronales, también se exploró la posibilidad de calcular la distribución de tiempos de residencia con el método de estimación de densidad con Kernel. Se vio en estudios preliminares que esta posibilidad, si bien tiene un tiempo de cálculo más alto en comparación con las redes, produce distribuciones sin tanto

ruido, además de poder usar un algoritmo único que hiciera la labor demandada.

En reglas generales, se puede ver que el planteamiento del algoritmo en cuanto a los inputs y outputs es el mismo que el que presentan las redes neuronales, es decir, se recibe un vector de tiempos de residencia de cada una de las celdas que forman un banco, y retorna el vector discretizado de la distribución de tiempos de residencia. El procedimiento interno que se realiza es el siguiente:

1. Se hace uso del mismo algoritmo de generación de números aleatorios que se usa en el método original. Este genera un número determinado de datos que distribuyen de manera exponencial con parámetro λ equivalente al tiempo de residencia medio. Si se tiene más de una celda, se generan un número equivalente de vectores con datos aleatorios, los cuales se suman al final de proceso, lo que entrega un set de datos que representa el tiempo de residencia dentro del banco completo.
2. Se realiza el procesamiento de los datos. Para el desarrollo de esta tesis, se trabajó en Python igualmente, haciendo uso de las librerías de scikit-learn y Keras, las cuales poseen herramientas que permiten realizar estas labores de manera más simple y automática. Se debe elegir el Kernel a utilizar para ajustar el modelo. Dados los datos con los que se cuenta, se decidió trabajar tanto con Kernels exponenciales y Gaussianos.
3. El algoritmo genera de forma interna la curva de distribución de tiempos de residencia. Se debe definir el vector de tiempos discretizados que puede ser interpretado como los rangos del histograma.

Se puede ver que la programación de este método es bastante más simple que de las redes neuronales, además de adaptable, ya que con un solo modelo se puede cubrir todo el espectro de situaciones. Con esto en mente, la idea de usar este algoritmo en contraposición del de redes neuronales es la de simplificar el algoritmo, eliminando las ramificaciones que existen con las redes, además de los algoritmos auxiliares, como la normalización, el escalamiento y el ajuste a la salida del modelo.

La base de datos a utilizar para probar la efectividad de este método difiere de la base de datos usada en el modelo de redes neuronales, ya que, en este caso, se decidió no separar según el número de celdas. La base de datos consta de un vector de largo único, el cual contiene distintos valores de tiempos de residencia para distintos números de celda en bancos. En los casos que se tenga un menor número de celdas que el largo total del vector, se ingresan los valores restantes como ceros. Estos son eliminados internamente dentro del algoritmo, para así tener el número exacto de tiempos de residencia de las celdas dentro del banco.

Por otro lado, las salidas corresponden al vector de la distribución de tiempos de residencia. Para este estudio, se consideró equiparar la salida de este algoritmo con la que se obtiene de las redes neuronales, por lo que se consideró obtener el valor de probabilidad de 600 datos, partiendo desde el minuto 0 al minuto 60, con discretización de 0.1.

3.3.2. Modelos aplicados a predicción de ley de concentrado

3.3.2.1. Motivación de uso

Tal como se mencionó anteriormente, para la creación del modelo fenomenológico se hizo necesaria la incorporación de un modelo que permitiese predecir la ley de un concentrado en función de la ley de alimentación a las celdas. Esta labor indirectamente incorpora los conocimientos acerca de recuperación de ganga, que, al no ser el objetivo de una celda de flotación, no es el foco por estudiar, lo que genera que los conocimientos que se tengan en este aspecto suelen ser limitados.

Para sortear este desconocimiento, se recurre a la información obtenida desde las pruebas de flotación batch. En estas se dividen las muestras por UGM, con el fin de estudiar el comportamiento que tiene el mineral de cada una al momento de ser flotado en condiciones industriales simuladas. Al obtenerse los resultados, se puede apreciar la cantidad de mineral que se recupera de cada una, obteniéndose de esta forma un estimado de la ley de salida que se podría obtener al tratar cada una por separado.

Es importante destacar que es esperable que cada UGM posea un modelo diferenciado según las características de esta, por lo que es necesario conocer el valor de esta categorización, ya sea por medio de información directa, o por medio de un modelo de clasificación.

3.3.2.2. Base de datos utilizada

La base de datos utilizada en esta etapa constaba de un modelo de bloques con información geometalúrgica. El dato principal que se usó para crear los modelos de esta etapa es la ley de Cobre en la alimentación, mientras que la única salida esperada es la ley de cobre en el concentrado. Este último dato estaba presente en la base de datos. Se estima que este fue producido mediante interpolación espacial, usando algún algoritmo como lo puede ser el Kriging, basado en datos experimentales reales provenientes de pruebas de flotación batch. Más específicamente hablando, los datos en cuestión pertenecen a un modelo de leyes de alimentación y concentrado para una etapa Rougher.

Esta base de datos no contaba con diferenciación inicial de los bloques según UGM, por lo que se consideró el uso de un modelo único con el fin de generar los resultados esperados. Ante esto, se planteó previamente como hipótesis primaria, que este modelo no presentaría una precisión muy elevada, dada la alta dispersión esperada entre los datos por no poder segmentar según diferencias mineralógicas, que a su vez pueden generar diferencias en los comportamientos en la etapa de flotación.

A continuación, se presenta la Tab. 3.1 con las principales estadísticas e histogramas relativos a los datos utilizados para hacer el entrenamiento de los modelos:

Tabla 3.1: Estadísticas de datos para modelo de leyes de Concentrado

Estadístico	Leyes de alimentación [%]	Leyes de concentrado [%]
Promedio	0.87	7.25
Desv. Estándar	0.6	4.54
Mínimo	0.01	0.04
Maximo	9.81	49.61

Histograma Leyes de Alimentación

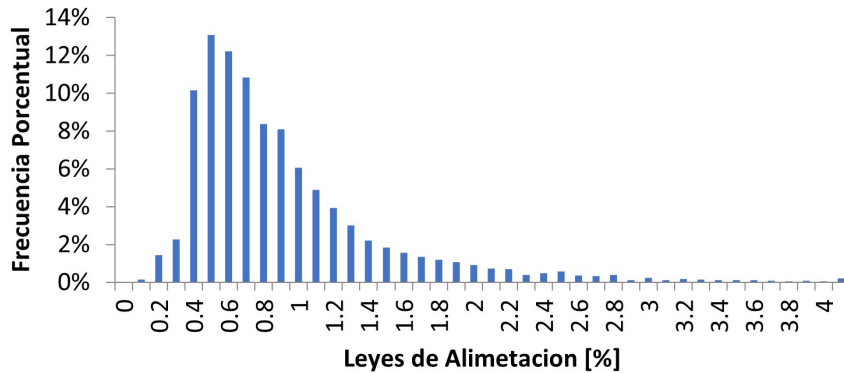


Figura 3.3: Histograma de leyes de alimentación

Histograma Leyes de Concentrado

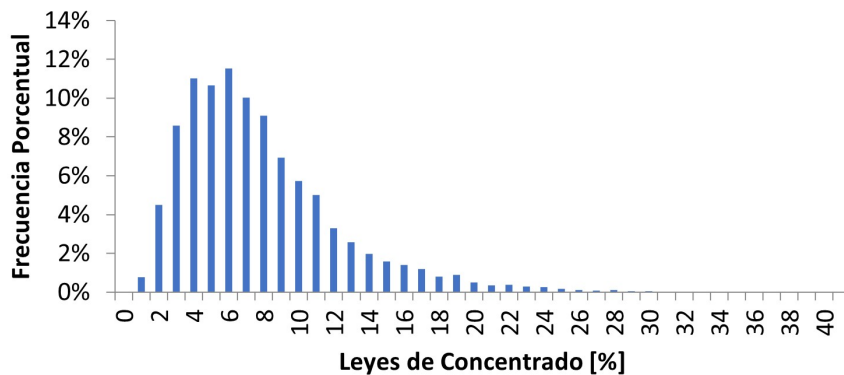


Figura 3.4: Histograma de leyes de concentrado

Se puede apreciar como los valores promedio rondan en torno al 0.8 % y 7.25 % para ley de alimentación y concentrado respectivamente. El primer valor se puede considerar adecuado, dadas las condiciones de los yacimientos que se presentan actualmente, donde se ha tendido a la baja en las leyes que se extraen [35]. Por otro lado, la ley de concentrado promedio si se puede ver que está por debajo de lo normal y esperado, ya que, al ser concentrado Rougher, se espera que la ley este en torno al 15 %. Esto puede deberse al hecho que la base presenta una gran cantidad de bloques con leyes de alimentación muy bajas, en torno al 0.4 %, de los cuales no puede obtenerse una gran cantidad del metal de interés, lo que implica que al ser procesado las leyes generadas tampoco serán muy altas.

Relacionado a lo anterior, se puede ver cómo, si bien la ley de alimentación promedio se encuentra dentro de un umbral realista, existen un gran número de bloques con leyes muy bajas, siendo el mínimo con una ley de 0.01 %. Dependiendo del tipo de faena, existe la posibilidad de que estos bloques no sean tratados, lo que cambiaría la base de datos al momento de entrenar. Por ejemplo, en una faena a rajo abierto, existe la posibilidad de

transportar los bloques bajo una ley de corte determinada a un botadero, lo que hace que no sean tratados por la planta. Por otro lado, en una faena subterránea del tipo Block Caving o Panel Caving, no existe la posibilidad de conocer de forma directa que zona en específico se está extrayendo, por lo que los bloques de baja ley sí son enviados a planta de flotación, lo que implica que deberían ser incluidos en el modelo.

3.3.2.3. Implementación

En esta etapa se comparó el desempeño que podrían mostrar dos métodos de predicción de leyes. El primero se basa en el uso de regresiones de datos, es decir, se buscó ajustar los parámetros de alguna ecuación conocida con el fin de acercar los valores de la salida a los valores esperados. Se probaron diversas funciones bases con este fin, como lo fueron funciones lineales, funciones exponenciales, potenciales o polinomiales.

Por otro lado, se busco también estudiar la efectividad de usar un método de redes neuronales para realizar esta predicción. Se creyó que su uso podría resultar útil si se añadiese el algoritmo de normalización de batch, pero al verse que se su desempeño era peor al de hacer una predicción directa, se decanto por descartar completamente su uso.

3.3.3. Modelos aplicados a identificación de UGMs

3.3.3.1. Motivación de uso

Ya se ha estudiado en profundidad como simular la función de distribución de tiempos de residencia dentro del modelo de recuperación planteando por Polat y Chandler. Recordando esta ecuación, esta posee un segundo término integral, que involucra a las constantes cinéticas y como distribuyen. Tal como se explicó en la construcción del modelo fenomenológico, se decidió usar el modelo de García Zúñiga para ser incorporado, considerando este que todas las partículas de un mismo mineral pueden usar la misma constante cinética para modelar su recuperación.

Este supuesto anteriormente dado tiene validez siempre y cuando se esté tratando a ese mineral en específico en condiciones estándar invariantes, es decir, que la mineralogía global que alimenta la celda es siempre la misma, al igual que las condiciones fisicoquímicas y operacionales. Considerando lo anterior, se puede asignar el valor de la constante cinética adecuada.

En una operación minera habitual el mineral se divide por UGMs para su extracción y posterior procesamiento. Cada UGM cuenta con sus propias características, lo que implica que el momento de realizarse las pruebas de flotación batch, se deduzcan diferentes constantes cinéticas para cada una. Estas diferencias son las causantes que se pueden tener recuperaciones distintas al tratar distintas UGMs bajo iguales condiciones operacionales. Es importante mencionar que, en la realidad, cada UGM en sí misma no es una unidad totalmente homogénea. Estas pueden presentar una cierta variabilidad. Sin embargo, se suele usar el promedio de estos valores como una simplificación del problema, con el fin de tener parámetros certeros para ser usados en los modelos.

Considerando todo lo anterior, se hace necesario conocer qué UGM se está tratando en un momento en específico, ya que esto permite asignar el set de constantes cinéticas correspondiente a la alimentación de la celda. Primariamente, se podría tener una idea global del

origen del material que se esté tratando en un instante en específico, sabiendo el tiempo de procesamiento de las etapas previas y la gestión desde los stocks, aunque esto solo es posible si existe una buena gestión de los equipos de procesamiento y se cuenta con las herramientas e instrumentos necesarios. Lo deseable es que, en una faena minera, el mineral extraído desde distintos sectores sea enviado a acopias distintos, donde es almacenado hasta ser procesado. Luego, considerando el tiempo promedio que la mineral demora en ser procesado en las etapas de conminución puede rastreadarse hasta su entrada en la celda.

Sin embargo, no siempre es posible tener ese rótulo acerca del origen del mineral, por lo que se deben recurrir a otros métodos para conocer esta información. El uso de herramientas como lectores de Ley en línea son útiles en este caso, ya que entregan información de la composición elemental y mineralógica de un flujo de alimentación en tiempo real. Frente a esto, se propone como opción usar las leyes de alimentación como datos de alimentación a un modelo de redes neuronales o de otra metodología de Machine o Deep Learning, que permita hacer clasificación de UGMs, con el fin de usar esta información para seleccionar el set de constantes cinéticas a usar.

Los datos requeridos que se utilizaron fueron los siguientes:

- **Leyes elementales de Alimentación:** Se contó con las leyes de cada bloque relativas a los elementos Cobre, Oro, Plata, Arsénico, Hierro y Molibdeno. Si bien estos datos provienen de un modelo de bloques, se espera que en una planta de flotación se pueda tener esta misma información cada cierto tiempo gracias a los equipos lectores de leyes en línea, por lo que su inclusión se consideró plausible en un modelo a escala industrial.
- **Relación entre elementos:** Se incluye además de las leyes en sí mismas, la relación entre un par de elementos, que en este caso es la Relación Cobre/Hierro. Se decidió incluir por ser un parámetro que normalmente se maneja en plantas de flotación y por normalmente ser definitorio al momento de identificar diferentes unidades Geometalúrgica.
- **Categoría:** Es la unidad Geometalúrgica en la que se categorizaron cada uno de los datos presentes. En los datos originales se trabajó con tres categorías principales. Al momento de estudiar los datos presentes en ellas, se puede ver como una de ellas está claramente diferenciada de las otras dos, con datos de leyes visiblemente mayores, lo que da a entender una zona de mayor enriquecimiento mineralógico.

3.3.3.2. Base de datos utilizada

Para realizar este trabajo, se contó con una base de datos real, consistente en un modelo de bloques, que incluye información relativa a las leyes de Cobre, Hierro, Arsénico, Azufre, Oro, Plata y Molibdeno, además de datos adicionales como la densidad y el W_i esperado de la roca. Por otro lado, cada bloque está categorizado con un número del 1 a 3, teniéndose de esta manera 3 posibles UGMs. Los datos de ubicación espacial de cada bloque se descartaron como posible parámetro de ingreso a la red. Esto debido a que en una planta de flotación no tiene sentido incorporar información sobre la ubicación espacial del material cuando éste ya ha sido extraído y procesado.

Además de las leyes en sí mismas, se decidió incorporar una relación entre leyes como input adicional, siendo específicamente la relación entre Cobre y Hierro. Los motivos que llevaron a añadir esta variable como input son que es muy común contar con esta información en

plantas de procesamiento, y que las diferentes zonas en un yacimiento suelen presentar una variabilidad muy apreciable en esta variable, siendo más alta en UGMs ricas en sulfuros, y más baja en UGMs ricas en óxidos.

La base de datos utilizada se dividió en la proporción estándar de 70/30 para entrenamiento y testeo respectivamente. En la Tab. 3.2 se muestran las el promedio y desviación estándar de leyes y relaciones relevantes para la identificación:

Tabla 3.2: Estadísticas Generales de categorías de UGMs

	Categoría 1		Categoría 2		Categoría 3	
	Promedio	Desv. Est.	Promedio	Desv. Est.	Promedio	Desv. Est.
Ley Plata [ppm]	5,54	4,79	3,27	3,65	2,38	2,81
Ley Arsénico [%]	0,05	0,06	0,03	0,07	0,03	0,06
Ley Oro [ppm]	0,02	0,01	0,01	0,01	0,01	0,01
Ley Cobre Total [%]	0,8	0,49	0,54	0,45	0,35	0,34
Ley Hierro [%]	1,68	0,79	1,8	0,64	1,96	13,21
Ley Molibdeno [%]	0,05	0,05	0,04	0,05	0,03	0,05
Ley Azufre [%]	2,33	1,13	1,83	1,05	1,54	0,92
Fracción Cu/Fe	0,55	0,34	0,35	0,3	0,23	0,24

Se puede apreciar como en líneas generales la ley de elementos de interés (Cobre, oro, plata y molibdeno) es mayor en la categoría 1 que en la 2 y en la 3, tendencia que se repite para el azufre. Por otro lado, se puede apreciar un comportamiento contrario en la ley de hierro, la cual presenta su valor más alto en la categoría 3. Haciendo el cruce de información, se puede inferir que la categoría 1 corresponde a una zona de enriquecimiento, rica en sulfuros, mientras que las categorías 2 y 3 corresponden a zonas más superficiales, con menor predominancia de los sulfuros y mayor presencia de óxidos de hierro.

3.3.3.3. Implementación

El estudio realizado en esta aplicación se llevó a cabo en tres etapas, las cuales están diferenciadas por la metodología usada para tratar los datos, usando mecanismos de clasificación directa en las dos primeras y una aproximación probabilística para la tercera.

3.3.3.3.1. Clasificación directa

En un primer punto, se planeó usar una metodología de clasificación directa de las UGM en función de las leyes de alimentación. Esto significa que se espera que el output entregado por el sistema sea directamente la categoría asignada. Para realizar esto, los algoritmos por presentar deben presentar un comportamiento categórico, o uno que pueda modelarse para ser usado de esta manera.

Para este caso, al tratarse de solo tres posibles unidades geometalúrgicas, cada una de las categorías fue asignada a un número entero entre -1, 0 y 1. La función de activación a la salida de la red fue la tangente hiperbólica, y las capas ocultas fueron variadas durante el estudio. Para realizar el proceso de asignación, se impusieron los valores umbrales de -0.33 y 0.33, es decir, si es menor que el primer valor se considera de la primera categoría, si es mayor que el segundo valor es de tercera categoría, y se esta entre ambos es de segunda categoría..

3.3.3.3.2. Clasificación por árbol binario

Una de las formas más comunes y simples de realizar clasificación es por medio del uso de variables binarias. Esta metodología, al ser usada de manera directa, está limitada a solo poder clasificar un set de datos en dos categorías previamente diferenciadas. La metodología base para su uso es la siguiente:

1. Se debe contar con un set de datos para entrenamiento o comparación, el cual incluya en forma de vector todos los datos con los que se pueda contar a la hora de realizar la clasificación, y la categoría asignada a ese vector en específico. Por lo general, se suele asignar categorías de 0 y 1.
2. Se crea el algoritmo de clasificación. La mayoría de estos no produce directamente la categoría. Lo que se debe realizar es fijar un valor de umbral, bajo el cual, todos los números generados son equivalentes a 0, mientras que los valores complementarios son equivalentes a 1. Este valor por lo general se fija en 0.5.

Si se desea realizar clasificación de más de dos variables, existen varias opciones que pueden ser utilizadas con este fin. Por ejemplo, existe un método que permite usar algoritmos de clasificación binaria para clasificar más de dos clases, denominado árbol de búsqueda binario. Este consiste en agrupar las clases que más se parezcan entre ellas, con el fin de reducir a la decisión a una categorización binaria. Cuando esta sea realizada, se debe ingresar a un segundo árbol binario, el cual debe separar entre las clases que ya habían sido agrupadas previamente y así repetir el proceso la cantidad de veces necesarias hasta lograr un árbol que sea capaz de diferenciar todas las variables por su cuenta.

En este caso, se uso una lógica similar al del clasificador directo, pero usando una categorización binaria. Al estudiarse las tres categorías existentes, se puede ver que la categoría 1 es la más diferente entre las tres, por lo que el primer paso es identificar si pertenece o no a este grupo específico. Las redes programadas con este fin tienen como principales parámetros diferenciadores que la función a su salida debe ser un valor binario, siendo 1 el valor que indica la pertenencia, y 0 la no pertenencia. Para esto se uso como función de activación en la única neurona de salida a la función sigmoidea y como función de pérdida la Binary cross entropy.

En caso que el valor no sea categorizado como de categoría 1, ingresa en un segundo árbol, el cual diferencia entre categorías 2 y 3. El esquema general de diseño de este árbol es idéntico al del primario. Finalmente para crear el modelo de clasificación, ambos deben acoplarse para trabajar en conjunto y entregar la categoría predicha.

3.3.3.3.3. Clasificación probabilística

Tras probarse el sistema de red directa y ver sus beneficios y problemas, se recurrió a una segunda opción, basada en una metodología de clasificación difusa o probabilística. Este tipo de algoritmos entrega un vector que informa sobre la probabilidad que un set de datos en específico pertenezca a cada una de las categorías posibles. De esta forma, los resultados con los que se cuenta son más dinámicos e interpretables y permiten una mejor toma de decisiones, ya que no solo indican la posible categoría de pertenencia, sino que también indican el grado de certeza con el que se cuenta.

La arquitectura base que se usó para desarrollar este método de clasificación es la siguiente:

1. Para este tipo de metodología, se debe entrenar haciendo uso de un vector de largo igual al de las posibles opciones a presentarse, el cual debe contener valores de 1 en las posiciones que son iguales al valor de la categoría a la que perteneces, y 0 en las demás categorías. Por otro lado, los valores de entrada pueden ser los mismos que los que se usan en el método de predicción directa. En este caso se hicieron pruebas para añadir el algoritmo de normalización a la entrada, pero se vio que generaba una serie de errores a nivel de programación, por lo que se decantó por descartar su uso.
2. Se hicieron pruebas con diferentes configuraciones en cuanto a la estructura. Al no utilizarse un algoritmo de normalización, se descartó el uso de las funciones de activación tangente hiperbólica, dado que están pueden generar números negativos, los cuales no deberían presentarse en este caso.
3. La Capa de salida debe ser obligatoriamente de tipo softmax, ya que este es el único algoritmo de los estudiados que entrega una distribución de probabilidades. El número de neuronas en la salida debe ser equivalente al número de categorías existentes.

La salida de este algoritmo es un vector de probabilidades, por lo que se puede obtener diversa información desde éste. En primer lugar, se puede observar la probabilidad más alta, la que puede ser interpretada como la categoría a la que pertenece el set de datos entregado. Por otro lado, se pueden estudiar los órdenes de magnitud de las probabilidades y sus diferencias entre sí, pudiendo hacerse un análisis de sensibilidad en cuanto a la certeza del resultado entregado. Por un lado, si se tiene que una opción presenta una probabilidad mucho más elevada que las otras dos, se puede interpretar que el algoritmo está haciendo una clasificación con mucha seguridad. Por otro lado, si se tiene que dos o más opciones tienen una diferencia muy baja, se puede interpretar como que el algoritmo no tiene seguridad en la respuesta que está dando. Ante esta situación podría obrarse señalando al operador que no se puede identificar la información con la que se cuenta, o hacer uso de otro algoritmo de manera complementaria.

Capítulo 4

Resultados

4.1. Modelo fenomenológico puro

Los primeros resultados obtenidos están relacionados con la capacidad y desempeño del modelo fenomenológico puro a la hora de entregar resultados relativos a la recuperación de un banco de celdas, en cuanto a precisión y tiempo de calculo. Para esta etapa solo se considera el trabajo de predicción de recuperación, por lo que el método de clasificación aun no es pertinente. Por otro lado, para hacer predicción de leyes de concentrado, ante el desempeño en ensayos previos, se decantó el uso del modelo regresión, que es explicado con posterioridad.

El primer punto a estudiar es el grado de sensibilidad que presenta el modelo fenomenológico frente a cambios en el método de Montecarlo utilizado para modelar las distribuciones de tiempos de residencia $E(t)$. Específicamente, la variable a estudiar es la cantidad de números aleatorios a generar, y su efecto tanto en el tiempo de calculo de recuperaciones, como el grado de suavidad presente en la curva.

Para corroborar como la cantidad de números aleatorios afecta en el modelo, se recurrió al diseño de un experimento donde se calcularon las DTRs correspondientes a un banco de dos celdas, con tiempos de residencia medios de 3 y 3.5 minutos respectivamente y con el uso de datos según muestra el gráfico en la Fig. 4.1.

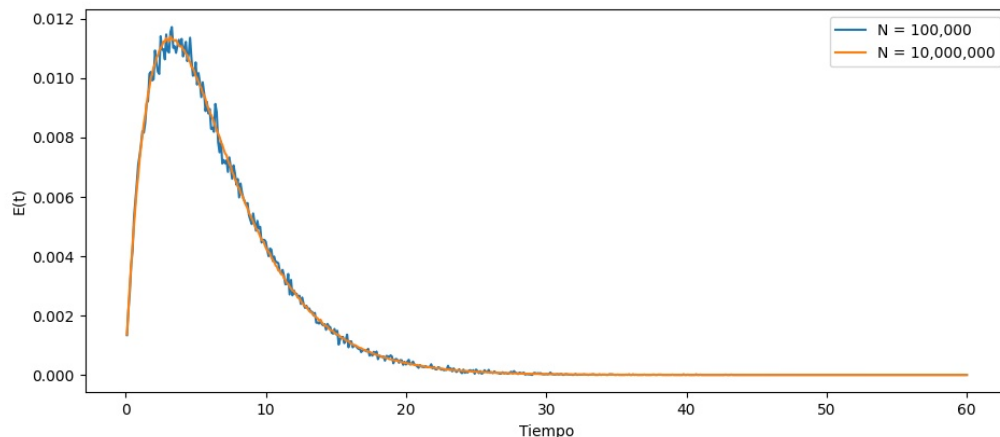


Figura 4.1: Comparación de DTRs según número de datos

Por otro lado, también se busco estudiar como asciende el tiempo de cálculo de la DTR en función de los números aleatorios, con el objetivo de hacer un *trade off* entre la suavidad de la curva generada y el tiempo que le toma calcularla. Para esta etapa también se calcularon las DTRs correspondientes a un banco de dos celdas, con tiempos de residencia medios de 3 y 3.5 minutos y números de datos que variaron entre los 10,000 y los 100,000,000 de datos a generar. En la Fig. 4.2 se aprecia como se ve la curva con escala logarítmica con respecto al número de datos.

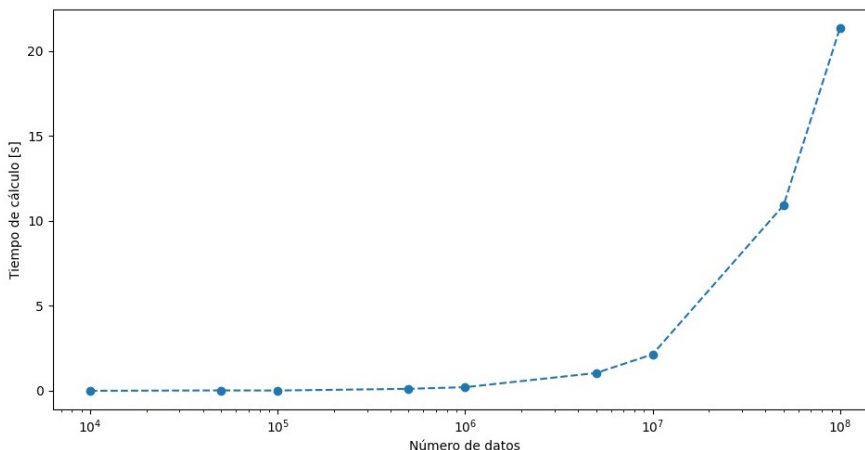


Figura 4.2: Aumento de tiempo de calculo de DTR frente a número de datos aleatorios

Por otro lado, se sabe que estos valores de DTRs son utilizados para el calculo de variables internas dentro de la función de recuperación, donde pueden ser llamadas en múltiples ocasiones dependiendo de el número de iteraciones que requiera el sistema. Considerando esto, se realizo un experimento igual al anterior, pero ya no considerando el calculo directo de la DTR, sino que el calculo de la recuperación de un sistema de tres celdas, con el fin de ilustrar como aumenta el tiempo de computo en la función objetivo al variar el número de datos aleatorios. Estos resultados se presentan en la Fig. 4.3.

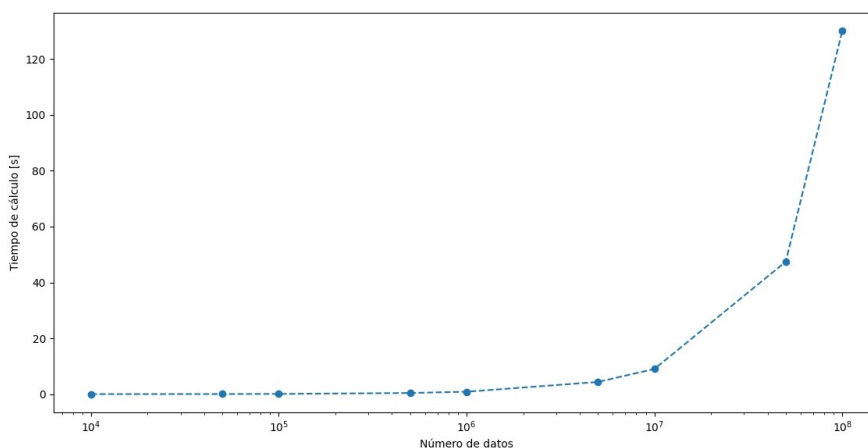


Figura 4.3: Aumento de tiempo de calculo de recuperación frente a número de datos aleatorios

4.2. Modelos aplicados a cálculo de DTRs

Tal como se planteo anteriormente, se buscó estudiar la posibilidad de calcular las distribuciones de tiempos de residencia por métodos alternativos al originalmente propuesto me todo de Montecarlo. En primer lugar, se trabajó con la posibilidad de hacer reemplazo con un método de redes neuronales. Se realizaron estudios en diversas etapas. A continuación, se presentan los mejores resultados obtenidos en cada una de estas etapas, con la estructura presente en la Tab. 4.1 y los parámetros de precisión y efectividad en la Tab. 4.2. El modelamiento se hizo solo considerando un banco de hasta tres celdas. Los resultados de más ensayos realizados se pueden encontrar en la sección de Anexos.

Tabla 4.1: Estructuras de modelos de predicción de DTR según etapa en estudio

Etapa 1: Predicción directa	Etapa 2: Modelos diferenciados
Estructura	Estructura
$n_h: 5 - n_{ne}: 30 - AF_{LR}$	$n_h: 5 - n_{ne}: 50 - AF_{LR}$
$n_h: 1 - n_{ne}: 600 - AF_{Sig}$	$n_h: 1 - n_{ne}: 600 - AF_{Sig}$
Etapa 3: Modelos normalizados y escalados	Etapa 4: Vectores de Latencia
Estructura	Estructura
$n_h: 5 - n_{ne}: 50 - AF_{LR}$	$n_h: 5 - n_{ne}: 50 - AF_{LR}$
$n_h: 1 - n_{ne}: 600 - AF_{LR}$	$n_h: 1 - n_{ne}: 600 - AF_{Sig}$

Tabla 4.2: Resultados de precisión en estudio de predicción de DTRs con redes neuronales

Etapa	Nº Celdas	MAE Promedio ($\times 10^{-2}$)	DKL ($\times 10^{-4}$)	Total variation ($\times 10^{-4}$)
Etapa 1	Global	4,8	122,1	7,7
Etapa 2	1	2,0	25,3	4,7
	2	2,2	49,1	2,9
	3	2,5	59,9	2,7
Etapa 3	1	0,7	1,1	2,8
	2	1,0	1,5	2,0
	3	1,0	1,5	1,7
Etapa 4	1	1,0	1,9	4,4
	2	1,1	2,0	2,3
	3	1,1	1,7	1,8

Por otro lado, se procedió también a realizar ensayos y estudios con el método de estimación de densidad con Kernel. Para este etapa se probó con cambios en el ancho de banda del métodos, con el Kernel utilizado y con el número de datos aleatorios por ser generados para predecir la distribución. En la Tab. 4.3 se muestran las estructuras usadas en los principales estudios, mientras que la Tab. 4.4 muestra los resultados obtenidos en cuanto a precisión. Para este método no es necesario generar separación por número de celdas, por lo que todos los modelos son globales.

Tabla 4.3: Estructuras usadas en estudio de estimación de densidad con Kernel para predicción de DTRs

Nº Ensayo	Ancho de Banda	Kernel	Nº de datos aleatorios
1	0,1	Gaussiano	300.000
2	0,1	Exponencial	300.000
3	0,1	Gaussiano	50.000
4	5	Gaussiano	50.000
5	0,1	Gaussiano	10.000

Tabla 4.4: Resultados de precisión en estudio de predicción de DTRs con estimación de densidad con Kernel

Nº Ensayo	EAM promedio ($\times 10^{-2}$)	DKL ($\times 10^{-4}$)	Total Variation ($\times 10^{-4}$)
1	2,7	24,8	50,1
2	2,8	25,4	50,1
3	4,0	36,5	51,7
4	38,5	1404,7	90,5
5	7,4	91,3	55,2

Para mostrar visualmente el desempeño del modelo, se graficaron las DTRs mediante los tres métodos en dos casos específicos: (1) El tiempo de residencia medio usado para generar la distribución esta dentro del rango de entrenamiento de las redes neuronales, y (2) Con un tiempo de residencia fuera del área de entrenamiento, esto con el fin de estudiar la capacidad de extensión de predicción de las redes. En las Fig. 4.4 y 4.5 se muestran respectivamente. Este caso específico es para 1 celda. Los casos para 2 y 3 celdas están incluidos en Anexos. Para el caso del calculo con uso de redes neuronales, se considero el uso de los modelos obtenidos en la etapa 3 del estudio.

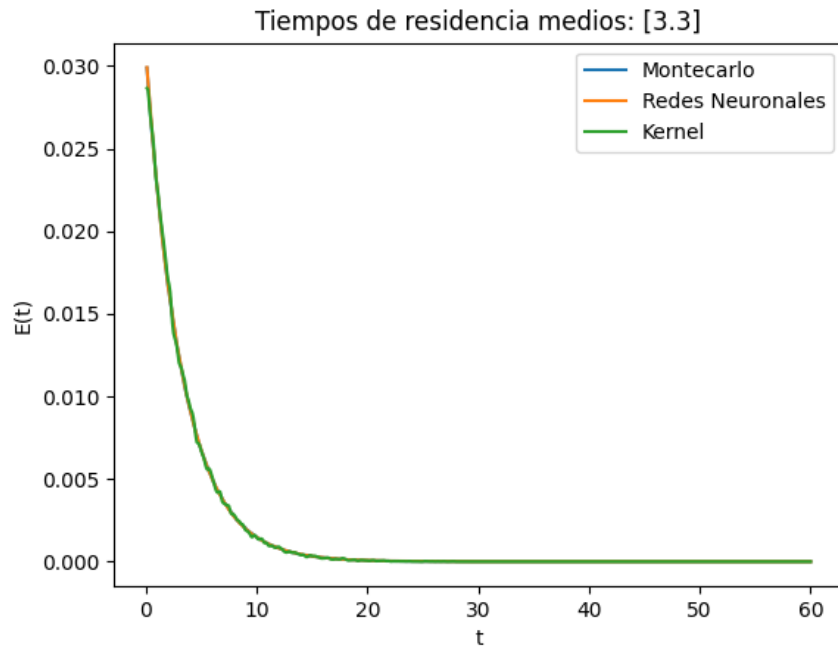


Figura 4.4: Comparación de distribuciones de tiempos de residencia con tiempo de residencia medio dentro del rango de entrenamiento

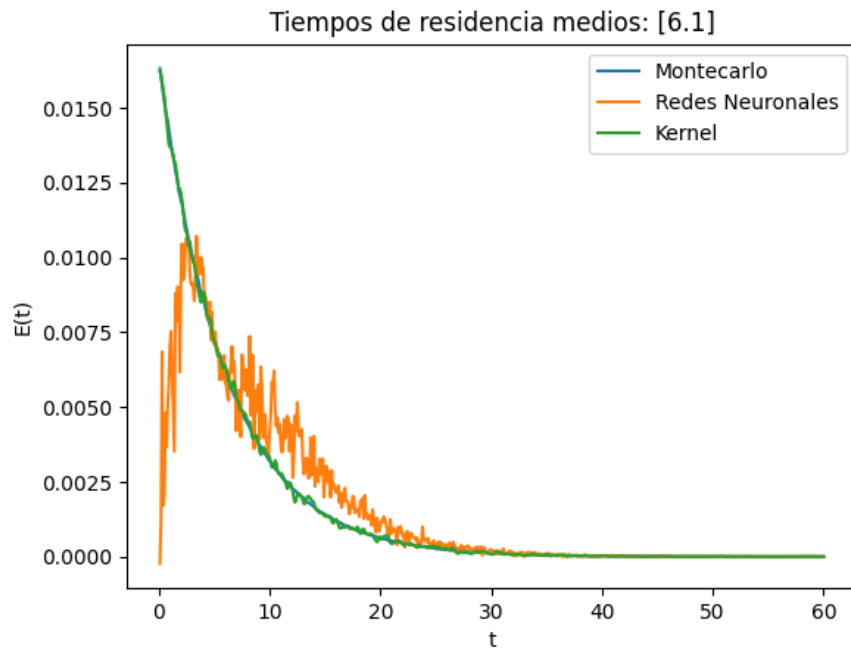


Figura 4.5: Comparación de distribuciones de tiempos de residencia con tiempo de residencia medio fuera del rango de entrenamiento

Una vez obtenidos los resultados de precisión entre las mismas DTRs obtenidas con distintos métodos, se procedió a estudiar como estas diferencias afectaban al momento de simular

la recuperación. Para esto, se creo un simulador temporal, el cual funciona ingresando valores fijos en las ley de alimentación, tipo de mineralogía y parámetros operacionales, mientras que el flujo de alimentación es impuesto como variable, mostrándose gráficamente tanto los flujos de alimentación como las recuperaciones esperadas.

Para este caso específico, se decidió estudiar un banco de dos celdas en serie. Se estudio un primer caso, donde los flujos se mantenían dentro de los límites en los cuales se entrenaron los sistemas de redes neuronales, mientras que el segundo caso se entreno con flujos de alimentación con límites mucho más amplios en los valores conocidos posibles. En las Fig. 4.6 y 4.7 se muestran ambos casos respectivamente.

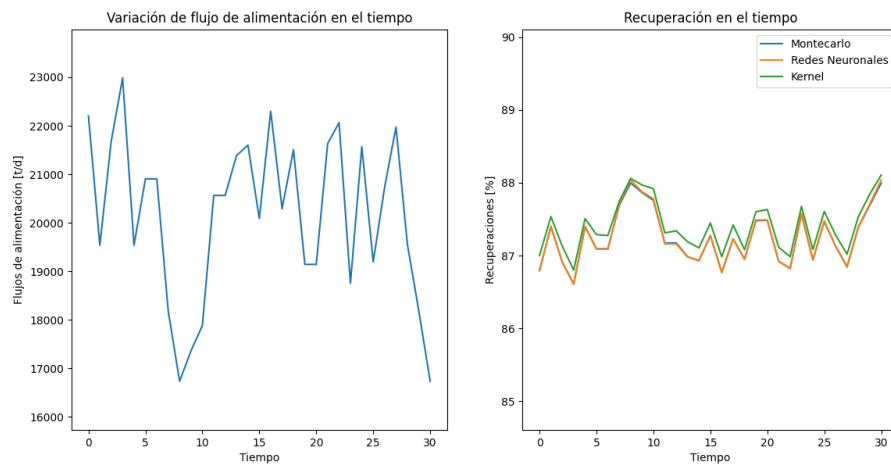


Figura 4.6: Curva de recuperación con flujo dentro de límites de entrenamiento

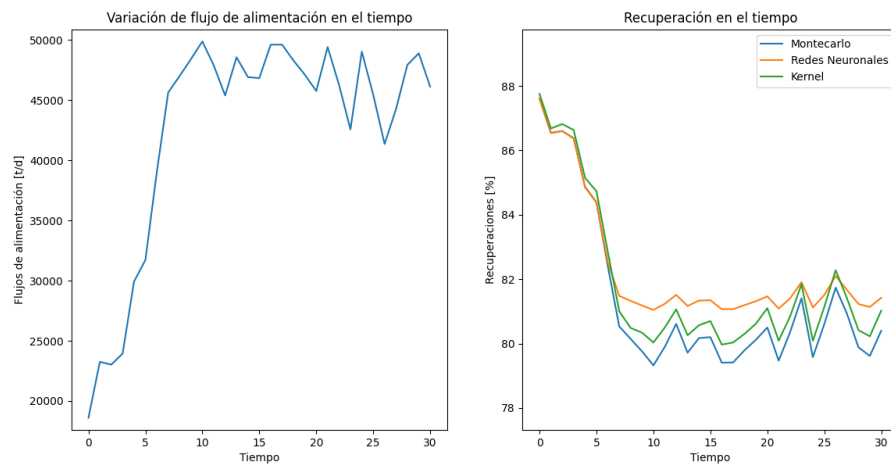


Figura 4.7: Curva de recuperación con flujo fuera de límites de entrenamiento

Por otro lado, además de estudiar las precisión que se obtiene con cada método, se busco

estudiar el tiempo de calculo asociado a cada método. En los casos de Montecarlo y estimación de densidad con Kernel, se realizo una simulación por cada número de celdas en banco, desde 1 hasta 8 celdas. Por otro lado, para el caso de las redes neuronales solo se pudo simular hasta el caso de 3 celdas, dado que solo esos modelos fueron creados. Sin embargo, con estos tres valores se pudo ver que seguían una relación bastante lineal en cuanto a la relación entre el tiempo de calculo y el número de celdas. Con esto, se generó un modelo lineal para interpolar los valores hasta el total de 8 celdas. La comparación realizada se muestra en la Fig. 4.8.

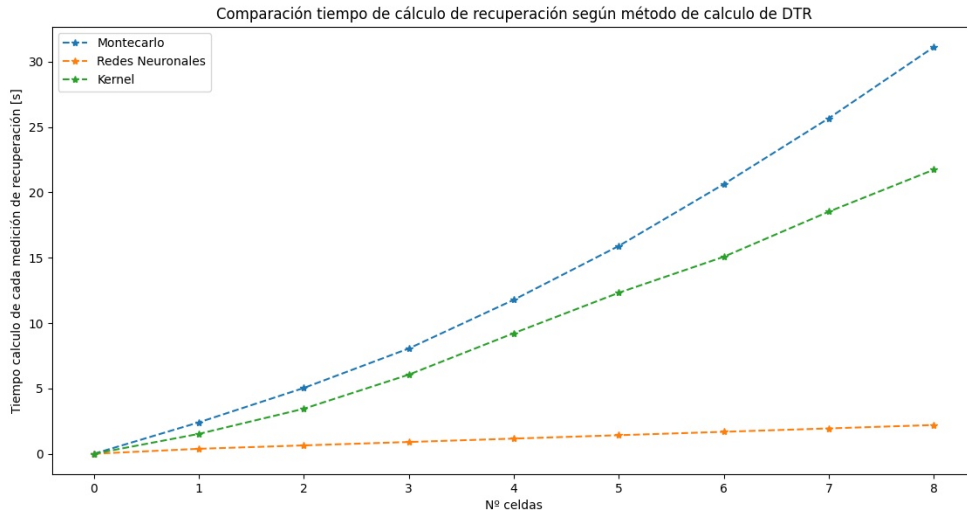


Figura 4.8: Comparación de tiempos de calculo de recuperación según número de celdas en banco y método

4.3. Predicción de leyes de concentrado

En este caso se hicieron estudios que justificasen tanto el uso de un modelo de regresión simple, obtenido con Excel, como de modelos de redes neuronales Cabe recordar que la base de datos desde la cual se trabajó no contaba con información relativa a unidades geometa-lúrgicas, por lo que se debió crear un modelo de características globales.

A continuación, en la Tab. 4.5 se presentan los resultados obtenidos de desde los modelos de regresión:

Tabla 4.5: Modelos de regresión para predicción de leyes de concentrado

Modelo de Regresión	Ecuación	Coficiente	ECM
Lineal	$L_{Cu,C} = 5.8181 * L_{Cu,A} + 2.1791$	0.77	7,95
Logarítmico	$L_{Cu,C} = 5.7015 * \ln(L_{Cu,A}) + 9.049$	0.74	8,68
Potencial	$L_{Cu,C} = 7.7622 * L_{Cu,A}^{0.8189}$	0.78	7,95
Polinomial grado 2	$L_{Cu,C} = -0.3087 * L_{Cu,A}^2 + 6.8597 * L_{Cu,A} + 1.617$	0.78	8,39

Por otro lado, también se realizo el estudio para observar la capacidad de predicción de esta variable mediante la aplicación de redes neuronales. Se realizaron diversos estudios,

combinando diferentes estructuras y funciones de activación. A continuación, en la Tab. 4.6 se pueden apreciar estos modelos y su desempeño.

Tabla 4.6: Modelos de redes neuronales para predicción de leyes de concentrado

Nº Ensayo	Estructura	Coficiente	ECM
1	$n_h: 10 - n_{ne}: 150 - AF_{LR}$	0.74	9.79
2	$n_h: 11 - n_{ne}: 150 - AF_{ReLU}$	0.75	9.87
3	$n_h: 11 - n_{ne}: 150 - AF_{TanH}$	0.69	12.44
4	$n_h: 1 - n_{ne}: 50 - AF_{ReLU}$	0.78	11.24
5	$n_h: 2 - n_{ne}: 50 - AF_{ReLU}$	0.78	11.34
6	$n_h: 2 - n_{ne}: 50 - AF_{ReLU}$	0.78	11.41

Dados los resultados obtenidos, y considerando que la creación de este modelo es crucial para el modelo de flotación global, se procedió a elegir el que mejor desempeño mostró de entre todos. En este caso, se decantó por el uso del modelo de regresión de carácter potencial.

A continuación, en la Fig. 4.9 se compara la curva de densidades de los datos reales contra ambos métodos de predicción. Estas curvas se obtuvieron con el método de estimación de densidad con un Kernel exponencial y un ancho de banda de 1.2.

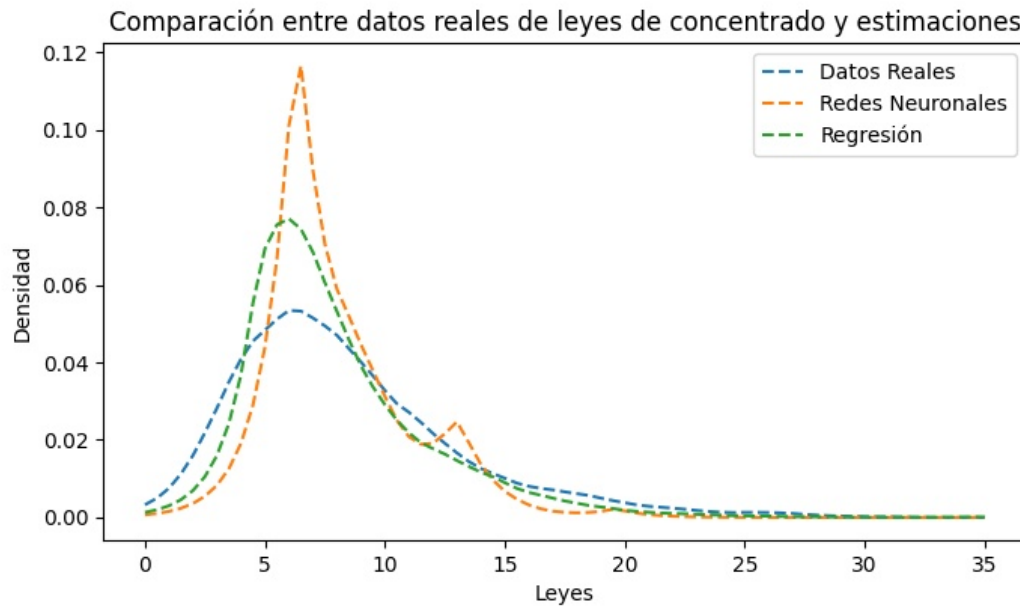


Figura 4.9: Comparación de densidades entre datos de leyes de concentrado

4.4. Identificación de Unidad Geometalúrgica

Se realizaron estudios para analizar la capacidad de clasificación de los métodos señalados en esta tesis. A diferencia de los modelos anteriores, la precisión se puede medir directamente como el cociente de datos que han sido clasificados correctamente y el número total de datos.

Además, se puede usar una herramienta denominada matriz de confusión para analizar la tendencia de errores que posee cada tipo de dato a clasificar.

La primera etapa era la de clasificación de variables de forma directa. En esta se hizo uso de la función de pérdida MSE. A continuación, en la Tab. 4.7 se muestran las estructuras utilizadas y en la Tab. 4.8 se muestran los resultados obtenidos.

Tabla 4.7: Estructura de Redes neuronales para clasificación directa de UGMs

Ensayo 1	Ensayo 2
$n_h: 5 - n_{ne}: 150 - AF_{LR}$	$n_h: 5 - n_{ne}: 150 - AF_{LR}$
$n_h: 1 - n_{ne}: 1 - AF_{TanH}$	$n_h: 1 - n_{ne}: 1 - AF_{LR}$
Ensayo 3	Ensayo 4
$n_h: 5 - n_{ne}: 150 - AF_{LR}$	$n_h: 10 - n_{ne}: 20 - AF_{TanH}$
$n_h: 1 - n_{ne}: 1 - AF_{Sig}$	$n_h: 1 - n_{ne}: 1 - AF_{TanH}$

Tabla 4.8: Resultados de clasificación directa de UGMs

Nº Ensayo	Precisión global	Precisión Cat 1	Precisión Cat 2	Precisión Cat 3
1	74.8 %	87.6 %	49.2 %	86.8 %
2	74.4 %	83.2 %	67.3 %	72.3 %
3	57.6 %	95.5 %	3.4 %	71.7 %
4	66.5 %	73.8 %	49.3 %	75.8 %

Para la segunda etapa, se decidió utilizar la metodología de árboles binarios, agrupando inicialmente las categorías 2 y 3. El árbol primero identifica si es que pertenece a la categoría 1, y si no, identifica si es 2 o 3. Además, se modificó la proporción de la base de datos y se usó la función Binary Cross Entropy. Para cuadrar con la elección de la función de pérdida, cada árbol tiene a la salida una capa con función de activación sigmoidea. Las estructuras usadas en los ensayos de ambos arboles se muestran en la Tab. 4.9, mientras que sus resultados estan en la Tab. 4.10

Tabla 4.9: Estructura de Redes neuronales para clasificación de UGMs con árbol binario

Nº Ensayo	Estructuras
1	$n_h: 10 - n_{ne}: 20 - AF_{LR}$
2	$n_h: 5 - n_{ne}: 200 - AF_{LR}$
3	$n_h: 5 - n_{ne}: 200 - AF_{Sig}$
4	$n_h: 10 - n_{ne}: 120 - AF_{LR}$

Para la tercera etapa, se decidió utilizar un método de clasificación probabilística. La función de pérdida elegida para poder optimizar el modelo fue la función Categorical Cross Entropy. Para cuadrar con la elección de la función de pérdida, se asigno a la salida una

Tabla 4.10: Resultados de clasificación de UGMs por árbol binario

Árbol primario			
Nº Ensayo	Precisión global	Precisión Cat 1	Precisión Cat 2/3
1	80.4 %	86.1 %	69.6 %
2	84.4 %	86.5 %	80.4 %
3	83.6 %	86.3 %	78.4 %
4	85.1 %	87.6 %	80.4 %

Árbol secundario			
Nº Ensayo	Precisión global	Precisión Cat 1	Precisión Cat 2/3
1	82.8 %	79.8 %	85.8 %
2	86.8 %	88.1 %	85.5 %
3	83.8 %	84.4 %	83.1 %
4	84.0 %	88.7 %	79.2 %

Arbol Combinado			
Precisión global	Precisión Cat 1	Precisión Cat 2	Precisión Cat 3
76.7 %	84.1 %	64.0 %	81.5 %

capa de 3 neuronas con funciones de activación softmax. A continuación, en la Tab. 4.11 se muestran las estructuras usadas en estas etapas, mientras que en la Tab. 4.12 se muestran los resultados correspondientes.

Tabla 4.11: Estructura de Redes neuronales para clasificación de UGMs con método probabilístico

Nº Ensayo	Estructuras
1	$n_h: 5 - n_{ne}: 200 - AF_{Sig}$
2	$n_h: 5 - n_{ne}: 200 - AF_{LR}$
3	$n_h: 8 - n_{ne}: 60 - AF_{Sig}$
4	$n_h: 10 - n_{ne}: 20 - AF_{LR}$

Finalmente, para comparar las tendencias al error que presenta cada método a la hora de clasificar, se muestran las matrices de confusión obtenidas en la mejor simulación de cada uno de ellos en la Tab. 4.13.

Tabla 4.12: Resultados de clasificación de UGMs por método difuso

Nº Ensayo	Precisión global	Precisión Cat 1	Precisión Cat 2	Precisión Cat 3
1	71.8 %	79.7 %	66.2 %	69 %
2	78 %	83.2 %	68.1 %	82.6 %
3	76.3 %	84.4 %	62.8 %	81.1 %
4	65.8 %	65.2 %	65.4 %	66.7 %

Tabla 4.13: Matrices de confusión por método

Clasificación directa			
	1	2	3
1	87.6 %	9.0 %	3.3 %
2	29.8 %	49.2 %	21.0 %
3	4.8 %	8.4 %	86.8 %

Clasificación por árbol binario			
	1	2	3
1	84.1 %	11.8 %	4.1 %
2	20.2 %	64.0 %	15.9 %
3	5.1 %	13.4 %	81.5 %

Clasificación por método difuso			
	1	2	3
1	83.2 %	12.9 %	3.9 %
2	16.8 %	68.1 %	15.1 %
3	3.4 %	14.0 %	82.6 %

Capítulo 5

Analisis de resultados

El objetivo de esta tesis es estudiar la eficacia o factibilidad de aplicar sistemas de Deep Learning como complementos a modelos fenomenológicos de flotación, con la visión de poder integrar alguno de estos algoritmos a una planta de flotación real. Sin embargo, frente a esto, se hace necesario validar la real efectividad que puede tener el peso del modelo fenomenológico elegido para este caso, que es el modelo de Polat y Chandler, dado que, si la base no presenta robustez suficiente, no tiene sentido aplicar técnicas más avanzadas dentro para dar soporte. Con esto en mente, se pueden hacer los siguientes análisis sobre el modelo de flotación escogido:

1. El modelo de Polat y Chandler proviene de una deducción matemática compleja, que involucra la utilización de una serie de supuestos sobre el proceso de flotación en sí mismo, que incluye la continuidad de las variables de alimentación al modelo. Esto quiere decir, que se espera que el comportamiento que se observe en la planta sea parejo, sin sobresaltos o depresiones fuertes en puntos específicos.

Frente a estos requerimientos, se encuentran los supuestos usados para construir bases de datos para trabajar, como que la alimentación fluctúa entre un 60% y un 120% del valor nominal de flujo en la planta, o que los cambios en los valores de leyes no pueden ser cambiados abruptamente frente a un cambio en la UGM de alimentación. Si se analiza cómo se relacionan estos parámetros, se puede ver como la utilidad del modelo se ve respaldada dado el hecho que se cumple el requisito mínimo de continuidad entre variables.

2. Las componentes de la ecuación de Polat y Chandler dependen de las características mineralógicas y operacionales del proceso, por lo que se debe saber la correlación que se puede esperar de los datos con los que se maneja una planta de flotación. El modelo requiere conocer las leyes de alimentación de la planta y el flujo tratado, información que en la realidad puede obtenerse de los instrumentos que son utilizados para realizar gestión y control de procesos. También requiere conocer los parámetros operacionales de altura de espuma y flujos de aire, que son variables controladas por el propio operador. Por último, también requiere conocer información sobre la UGM que está siendo tratada, para gestionar correctamente las constantes cinéticas y datos operacionales asociados a esos minerales en especial. Esta información puede ser compleja de obtener el tiempo real, por lo que lo más común es que se estime en función de los datos de manejo de materiales desde los acopios.

Con esto en mente, se deduce que el modelo de flotación presenta soporte real para

poder ser utilizado en una planta, dado que gran parte de la información que requiere su funcionamiento puede ser obtenida directamente desde la operación, y la que no, puede ser deducida con ayuda los modelos de soporte que se han desarrollado.

Teniendo en cuenta las consideraciones anteriores, se puede ver que el modelo fenomenológico planteado de base presenta el suficiente soporte para ser usado en el contexto industrial. Si bien, no es un modelo optimizado para usarse en situaciones de cambios abruptos constantes, se espera que un ambiente industrial en el cual podría verse inmerso no presente ese comportamiento. Por otro lado, el modelo planteado presenta mejoras y avances en relación a los ya existentes en literatura e industria, al observar el proceso por cada celda para calcular el tiempo de residencia medio de cada una, y también como un banco completo, al estudiar la DTR del conjunto de celdas.

Ahora bien, si ya se ha llegado a la conclusión que el modelo es útil para estos usos, también es importante cuestionarse si el añadir soporte con sistemas de Deep Learning conserva la representatividad que presenta, o si, por el contrario, entorpece la labor que se realiza. Si se identifica esta última situación, se llega a la conclusión que la utilización de redes neuronales u otros métodos de Machine Learning no se justifica en este ámbito. Si se estudian los tres ámbitos en los que se hecho foco para aplicar redes neuronales, se pueden obtener los siguientes análisis principales:

1. Las redes neuronales para el cálculo de la distribución de tiempos de residencia son un sistema de reemplazo, dado que sí existe una metodología real para calcularlas, que es un método de Montecarlo. Sin embargo, este último es un algoritmo que toma mucho tiempo de cálculo y conlleva un alto costo computacional. Dado esto, un sistema que emule su función con un tiempo de cálculo menor es una opción que se debe tener en cuenta. Bajo esta concepción, este primer sistema de Deep Learning no modifica directamente ningún aspecto general del modelo fenomenológico.

Por otro lado, se pudo ver en diversas etapas del estudio, que los modelos de redes neuronales no presentan una precisión exacta a la hora de reproducir los resultados entregados por el método de Montecarlo. Esto es relevante, ya que puede producir cambios en los resultados esperados, situación que sí se presentó durante el desarrollo de esta tesis, aunque fue en etapas de estudios en condiciones de variabilidad extrema para el sistema. Si se quiere tener confianza en la solidez de este modelo, es necesario contar con una extensa base de datos para entrenar esta red neuronal de salida. Ya se ha demostrado durante el desarrollo de esta tesis que los sistemas de Machine y Deep Learning pueden ser confiables y útiles, siempre y cuando se cuente con una base de datos lo suficientemente grande y se hagan el número suficiente de ensayos para identificar la mejor estructura posible.

Razonando de forma similar a la anterior, el algoritmo de estimación de densidad con Kernel es una alternativa al cálculo de las distribuciones de tiempos de residencia que presenta una serie de ventajas frente al uso de redes, como lo son la simpleza en su ejecución y programación, además de su versatilidad.

Sin embargo, los estudios preliminares demuestran que el tiempo de cálculo si se ve reducido con respecto al calculado por el método de Montecarlo, pero no de una forma tan sustancial como la que presenta el modelo de redes neuronales. Frente a esto, se debe tener en consideración ambas posturas al momento de elegir la más conveniente.

2. Las redes neuronales aplicadas para predicción de leyes de concentrado son usadas para

llenar un vacío en el modelo fenomenológico, generado por el poco conocimiento con el que generalmente se cuenta alrededor de la ganga. Similar al punto anterior, existe un modelo base que podría usarse para calcular las recuperaciones de ganga dentro del sistema, que funciona de manera equivalente al del mineral de interés. Sin embargo, este se basa en datos de los que no se tiene certeza, como lo es la constante cinética de la ganga.

Frente a esta falencia que presenta el modelo fenomenológico en representar la realidad, la opción de aplicar un sistema basado en metodologías de Machine y Deep Learning surge de manera natural. La ventaja que se tiene al aplicar este tipo de metodologías es que se basan en información con la que, si es posible contar en la realidad, como lo son las leyes de alimentación a una planta, por lo que acerca este sistema a la realidad. Con esto en mente, se tiene que ambos tipos de sistemas estudiados, regresiones y redes, son un soporte que aporte pragmatismo al modelo general, aunque, tal como ya se mencionó anteriormente, se requiere de la presencia de una base de datos lo suficientemente grande como para que los resultados obtenidos tengan la robustez que da tener diversas situaciones como entrenamiento.

3. Las redes neuronales para identificación de UGMs se presentan como un soporte a la ecuación de Polat y Chandler, en una labor que, a diferencia de los puntos anteriores, no tiene un modelo propio creado o un sistema que se ha demostrado su completa validez. En una vista global, el trabajo que realiza este algoritmo no está situado directamente dentro del modelo fenomenológico, sino que entrega información a este para que pueda calcular la recuperación. Teniendo esto en cuenta, se llega a la conclusión que un algoritmo de esta naturaleza no solo aporta al sistema creado, sino que realmente es necesario para su funcionamiento.

En una mirada global, se puede ver como los tres puntos de aplicación de redes neuronales que se han estudiado durante esta tesis presentan concordancia y acoplamiento con el modelo fenomenológico creado, por lo que se puede confirmar que, si están bien estructurados y entrenados, los sistemas de Deep Learning sí presentan una oportunidad de mejora en el modelamiento de este proceso metalúrgico.

5.1. Predicciones de distribuciones de tiempos de residencia

Analizando los modelos por separado, se puede partir con las redes de predicción de distribuciones de tiempos de residencia. En un principio, se creyó que un sistema único era factible en su armado, considerando el uso de funciones de activación sigmoideas a la salida, dadas las características de los resultados de una DTR, que, al ser una distribución de probabilidades, todos sus componentes están entre 0 y 1. Al no conocerse en un principio los algoritmos de normalización y desnormalización de variables, se pensó que era posible que esta metodología fuese capaz de simular correctamente los valores y la forma de la distribución.

Sin embargo, pudo verse que esta primera aproximación no resultó como se esperaba, presentándose en casi todos los casos coeficientes de correlación bajos para lo que se busca. Esto implica que los datos obtenidos desde estas redes no se aproximan a los reales ni en la forma ni en los valores en sí mismos. Existen varios motivos que podrían explicar estos

resultados. El primero es el uso de un modelo único sin importar el número de celdas. Como se explicó anteriormente, el uso de un modelo único implica dejar valores equivalentes a 0 cuando se quiere modelar un número de celdas menor al tope que permite el modelo. Esto trae consigo la dificultad que ese valor pasa a ser al mismo tiempo un número y una categorización, ya que en la realidad no tiene sentido un tiempo de residencia igual a 0, pero la red no hace esa distinción, haciendo que cuando se deba ingresar el valor de 0, este elimine un notable número de conexiones neuronales, lo que hace que el sistema pierda información que es relevante. Por otro lado, el no uso de los algoritmos de normalización y desnormalización afectan negativamente dado que la predicción se hará con variables directas, que presentan mayor variabilidad que las variables normalizadas.

Al notarse todos estos puntos de ineficiencia se realizaron los cambios planteados, los cuales consiguieron aumentar enormemente la capacidad de predicción del nuevo sistema. De los modelos presentados, se ve que el que presenta un mejor rendimiento, genera resultados con una divergencia de Kullback-Leibler muy cercana a 0, lo que implica que los resultados están casi perfectamente relacionados con los datos reales. Si se analiza cómo los cambios afectaron al sistema de predicción de redes, se pueden concluir los siguientes puntos:

1. El uso de modelos separados genera el principal cambio en los resultados. Esto debido a que se elimina la aparición de variables que se anulan según ciertas condiciones, lo que permite mantener todas las conexiones neuronales permanentemente activas.
2. Escalar los datos de salida ayuda a que el entrenamiento de las neuronas no se haga en dominios matemáticos tan reducidos. Además, permite mejorar los ajustes ya que mientras más se aumenta la escala, un error entre la predicción y los valores reales se verá más notorio.
3. Los algoritmos de normalización y desnormalización permiten que la predicción se realice en un dominio más acotado, además de permitir integrar la aparición de variables negativas que aportan más a la variedad de valores de ingreso. Es importante destacar que la red que mejor funcionamiento mostró es la que usa funciones de activación tangente hiperbólica, que justamente es la que permite el uso de variables negativas.

Teniendo en cuenta el resumen general que se ha realizado del cálculo de DTRs con redes neuronales, se puede ver una clara evolución en la calidad de precisión de los modelos, lo que denota tanto el aprendizaje que se ha realizado durante el desarrollo de esta tesis, como el número de pruebas que se debieron realizar.

Como segunda opción para el cálculo de las distribuciones de tiempos de residencia, se presentó la opción de la estimación de densidad con Kernel. Es importante destacar que este algoritmo está pensado y diseñado precisamente para cumplir la labor dada en este punto, que es la estimación de distribuciones de probabilidad, y que, como se mencionó anteriormente, puede ser interpretado como un equivalente matemático a la creación de un histograma porcentual. Teniendo esta primera consideración, se podría especular inicialmente que este debería ser una opción tan o más precisa que las redes, lo cual se terminó confirmando durante el desarrollo de esta tesis.

Se pudo ver en la etapa de estudios, que la elección de variables estructurales no genera una gran variabilidad entre los resultados obtenidos. El número de datos aleatorios a generar

para hacer el estudio demostró no tener una gran injerencia en la precisión. Esto es una buena noticia desde el punto de vista del gasto computacional de cálculo. El Kernel utilizado tampoco entrega una gran variabilidad a los resultados finales. Si bien se eligió trabajar con el exponencial ya que comparte distribución con el algoritmo de generación de números aleatorios, el gaussiano también demostró buenos resultados.

Sin embargo, en el lado de las desventajas, se puede ver que el tiempo de cálculo si bien el calculo es disminuido con respecto al método original, este sigue siendo superior al dado por las redes neuronales. Por ejemplo, si se quisiese hacer la comparación para un banco de 8 celdas, el método de estimación de densidad con Kernel tardaría unas 10 veces más que las redes en calcular la recuperación del banco.

Si se quisiese usar alguno de estos algoritmos en un ámbito industrial, se puede ver que se tendría un umbral de 20 segundos entre cada actualización de los parámetros operacionales al usar la estimación de densidad con Kernel, mientras que las redes alcanzarían un valor cercano a 2 segundo. Sin embargo, hay que considerar también el tiempo que tarden los equipos de medición en planta en entregar las señales que son necesarias para ingresar al modelo. Por ejemplo, los flujómetros pueden tener tiempos de respuesta y actualizaciones de menos de 1 segundo [36], los medidores de nivel de entre 1 y 2 segundos [37] y los analizadores elemental XRF en torno a los 20 segundos [38].

Si bien esto pareciese dar como ventaja definitiva el uso de las redes, dado que ambos métodos mostraron comportamientos similares en cuanto a precisión, hay que tener en cuenta dos consideraciones extra:

- Si bien las redes demuestras ser más rápidas, también demuestran un grado de complejidad más elevado en cuanto a su implementación. Se puede ver cómo es necesario incluir varios algoritmos auxiliares a su cálculo, además de requerir separación de variables y un reentrenamiento constante, lo que solo añade dificultades al proceso.
- Las redes demuestran ser muy poco flexibles en cuanto a capacidad de adaptación frente a cambios importantes en las celdas. Se pudo ver en varias etapas de desarrollo, como al presentarse casos con los que no se han entrenado, las redes pierden en gran medida su capacidad de predicción del fenómeno. Por otro lado, el algoritmo de estimación de densidad si demuestra flexibilidad frente a posibles cambios, ya que no requiere entrenamiento, si no que solo realiza adaptación a un modelo predeterminado.
- Si bien la ventaja en cuanto al tiempo de cálculo es notable, esta puede significar nada si es que los tiempos de actualización de las variables industriales son aún mayores. Por ejemplo, se puede ver que la actualización en cuanto a las leyes de alimentación puede ocurrir aproximadamente cada 20 segundos, haciendo que tener la capacidad de calcular recuperaciones cada 2 segundos no tenga utilidad si se quiere una actualización en la medición de recuperación que sea paralela a la medición de variables.

Considerando lo anterior, se puede plantear que es necesario hacer un Trade off entre las ventajas que entregan ambos métodos a la hora de decantarse por uno. Por ejemplo, si se decidiese usar el modelo de redes neuronales, considerando que la ley de alimentación no es una variable que se espere tenga una variabilidad temporal en cortos plazos y que se le podría asignar la ultima medición conocida, se podría tener un modelo que, en tiempo real,

indique la recuperación esperada, sabiendo que es susceptible a cambios en la alimentación que se salga de los parámetros de trabajo típicos. Por otro lado, si se escogiese el método de estimación de densidad con Kernel, se tendría un modelo que produciría menos data, pero que también es menos susceptible a perder representatividad.

5.2. Predicciones de ley de concentrado

Siguiendo con los modelos de predicción de ley de concentrado, se puede ver que los modelos de regresión simples presentan un comportamiento muy similar en cuanto a la precisión otorgada. Esto es señal que en los datos no existe una clara tendencia, por lo que cualquier modelo podría ajustar una serie de puntos según una ecuación base dada. Sin embargo, tampoco es despreciable el grado de cercanía que alcanzan. Se puede ver que el mejor modelo, el potencial, tiene un coeficiente de correlación de 0.78, muy cercano a 0.8, que suele ser un valor de umbral para definir la utilidad de una regresión. Con esto en mente, se decidió usar el modelo potencial para ser incluido dentro del modelo fenomenológico, a la espera de los resultados que fuese capaz de entregar las redes neuronales.

Una vez estudiados los sistemas de redes aplicados en este caso, se pudo ver que se presenta un comportamiento similar al que entregan los modelos de regresión más simples. Los diversos estudios aplicados a número de capas, número de neuronas y funciones de activación no presentaron mucha variabilidad, lo que apoya la hipótesis previamente planteada que no existe una clara tendencia entre los datos de leyes de alimentación y leyes de concentrado, por lo que cualquier configuración podría ser capaz de entregar resultados cercanos.

Como vista general de la labor realizada en este punto, se puede ver que los modelos sin algoritmos de normalización y el modelo potencial presentan un comportamiento similar. Frente a esto, se ha elegido quedarse con el más simple, que vendría siendo el potencial, ya que es más fácil de implementar y de entrenar, además que no requiere la creación de archivos ni variables extras. Por último, se prevé que los modestos resultados obtenidos se podrían mejorar si se implementase sobre una base de datos que cuente con diferenciación de acuerdo con la UGM, ya que permitiría agrupar los datos según el comportamiento metalúrgico dadas las diferencias en las leyes.

5.3. Clasificación de unidades geometalúrgicas

Por último, al estudiarse el algoritmo de identificación de UGMs, se hace importante destacar en un comienzo que este es el único algoritmo de clasificación que se programó, por lo que inicialmente se quería estudiar como era su rendimiento si se entrenaba como si fuese un algoritmo de predicción. Es por esto por lo que, en las primeras redes creadas, se decidió usar la función de pérdida MSE. Al estudiarlas, se puede ver como el rendimiento mostrado era aceptable pero perfectible, obteniéndose un máximo de precisión de 75 aproximadamente %, el cual se obtuvo al usar funciones de activación tipo Leaky ReLU, lo cual tiene sentido, ya que el uso de funciones sigmoideas acota demasiado los posibles valores que salen de cada neurona, perdiéndose información en las conexiones neuronales. Al estudiarse las matrices de confusión, se puede ver como en todos los casos, son los datos categoría 1 y 3 los que mejor precisión presentan, mientras que con los datos tipo 2 es donde más se tiende a equivocar. Esto puede deberse a que el dominio de datos que se le asignó a las clasificaciones del tipo 2

son los más acotados, dado que solo son los datos entre 1.5 y 2.5 los que pueden ser clasificados como 2, quedando todo el resto de los números reales como posibles dominios para las categorías 1 y 3. Esto indica que este tipo de restricciones puede perjudicar la clasificación de una categoría en específico, lo cual se tuvo en cuenta para la siguiente etapa de desarrollo.

Por otro lado, la metodología del árbol binario demostró tener un rendimiento ligeramente mayor en cuanto a precisión. Esto debido a que usa una función de pérdida diseñada específicamente para algoritmos de clasificación binaria. Además, se suman todos los cambios realizados, como el igualar los datos de cada categoría en la base de datos de entrenamiento y agrupar las categorías 2 y 3 en la identificación del primer árbol. Con respecto a esto, se puede ver que el segundo árbol, o sea, el que separa entre categorías 2 y 3 presenta un rendimiento mejor que el primero, llegando a valores en torno al 87% de precisión, contra el árbol primario que logra grados de precisión de aproximadamente 85%. Sin embargo, al momento de aplicar ambos arboles combinados, la precisión en la clasificación global baja hasta un valor aproximadamente 77%. Esto se debe probablemente a que ahora ya no se trata de un solo problema de clasificación, sino que de dos, por lo que los errores que cometen ahora están concatenados. De hecho, si se interpretan los grados de precisión como probabilidades de acierto para cada uno de los arboles y se multiplican para obtener la del método completo, se tiene que este valdría aproximadamente un 74%, muy cercano al valor real de la clasificación del árbol completo.

La aplicación del algoritmo de clasificación probabilística se presentó como una alternativa a la clasificación directa que contaba con la ventaja de entregar más información al usuario, lo que, en teoría, podría llevar a tomar decisiones más informadas. Se buscó con este método que la clasificación no solo dijese a que categoría pertenece un mineral en específico, sino que también se buscó poder señalar el grado de seguridad o certeza que se tiene al momento de señalar la posible UGM. Se pensó originalmente que este método podría presentar mejores resultados, ya que la lógica probabilística genera un producto más interpretable. Esto último se vio reflejado en los estudios, aunque no con tanta diferencia como se esperaba, obteniéndose que la mayor precisión obtenida con este método es de 78% aproximadamente.

Se puede ver que en reglas generales, los modelos presentan comportamientos similares en cuanto a la clasificación individual de cada categoría. La posible fuente de este comportamiento es la base de datos en sí misma. Por ejemplo, se pudo observar que existen dos categorías muy diferentes entre sí, la 1 y la 3, mientras que la categoría restante tiene un comportamiento intermedio y similar a ambas dependiendo del parámetro que se está estudiando. Esto hace que sea más complejo intentar categorizarla, ya que tiene dos posibles fuentes de confusión. Esto queda reflejado al momento de estudiar la matriz de confusión correspondiente, ya que se observa que presenta el mayor grado de error en la categoría 2, lo que genera dudas sobre su utilidad en este punto.

5.4. Propuesta de un modelo de predicción de recuperación

Teniendo en cuenta todos los resultados y análisis realizados, a continuación, se presenta el ensamblaje una propuesta de modelo completo, considerando los modelos que fueron escogidos en cada uno de los ambientes necesarios y sus posibles variaciones. Se muestra, además, un esquema que representa como se conectan los distintos algoritmos previamente mencionados:

1. Cálculo de DTR: Tal como se mencionó anteriormente, la elección del modelo a usar debe quedar en manos de la persona que decida implementar este modelo. Si desea tener la opción de contar con información constante en tiempo real, con el riesgo que el modelo presente fallos en casos anormales para la planta, se debería escoger el método de redes neuronales. Por otro lado, si se desea flexibilidad, se debería decantar el método de estimación de densidad con Kernel.
2. Predicción de leyes de cobre en concentrado: Se decanta por el uso de la ecuación potencial que relaciona en forma directa la ley de alimentación con la ley del concentrado. Se ha elegido este método ya que es más simple que implementar redes neuronales, además de presentar una precisión casi igual al de las redes.
3. Clasificación de UGMs: Existe similitud entre los métodos estudiados. Si se debiese escoger uno es esta etapa, podría ser el de clasificación probabilística, ya que es el que presenta un mayor grado de precisión y que tiene el error distribuido más equitativamente entre sus 3 categorías.

Finalmente, en la Fig. 5.1 se muestra un esquema que muestra como se complementan y encajan todos los modelos estudiados y desarrollados.

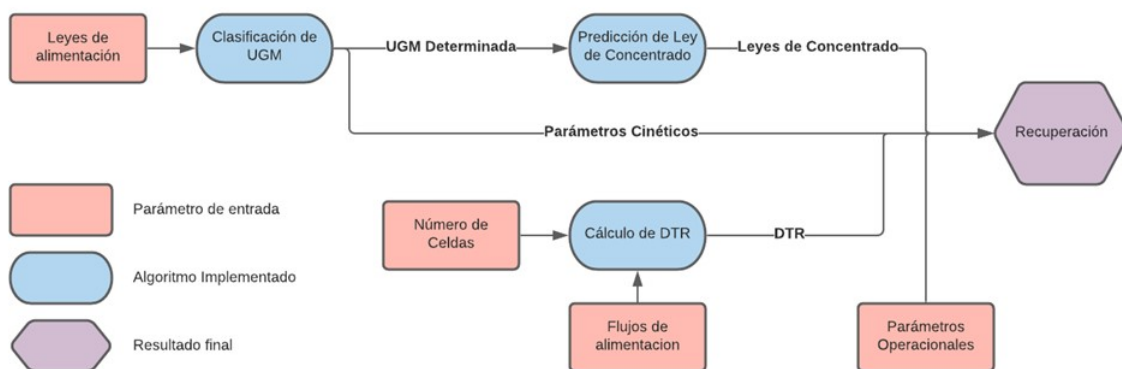


Figura 5.1: Esquema de funcionamiento de modelo creado

Capítulo 6

Conclusiones

Se puede ver cómo, en la construcción de todos los modelos que se han creado, se ha recurrido al uso de supuestos que, en mayor o menor medida, alejan la labor realizada de la realidad industrial. Es por esto por lo que se puede concluir que la utilización de metodologías provenientes de mundos como lo son Machine y Deep Learning pueden dar un gran soporte al modelamiento fenomenológico. Esto se puede observar dentro del desarrollo de la propia tesis, ya que, de las tres principales áreas que se han estudiado, se puede ver como dos llegan a cubrir áreas que no eran consideradas, como la de la identificación de UGMs y la de predicción de leyes.

Si se estudian los modelos implementados para la predicción de distribuciones de tiempos de residencia, se puede ver que ambas opciones presentadas como alternativa al cálculo con Montecarlo presentan ventajas y desventajas. Por el lado de las redes neuronales, estas tienen un grado de precisión muy elevado, al punto de poder predecir casi con total exactitud los valores de recuperación siempre y cuando se este dentro de los parámetros de entrenamiento. Por otro lado, el método de estimación de densidad con Kernel demuestra tener una mayor flexibilidad a cambio de tener un mayor tiempo de cálculo, lo cual tiene sentido al ser un equivalente matemático el cálculo del histograma acumulado.

El modelo de predicción de leyes es el que menos precisión demostró, con un coeficiente de correlación de 0.78, ligeramente por debajo del umbral de calidad o utilidad de un modelo que suele estar en torno a 0.8. Aun así, se puede ver que el acercamiento a la realidad no es despreciable y aun presenta oportunidades de mejora, dado que se podría estudiar este mismo problema en un caso donde se cuente con más información, como la UGM, con lo que se podrían crear modelos diferenciados.

Por último, el modelo de identificación de UGMs también presenta resultados prometedores, con precisiones globales en torno al 78 % usando el método de clasificación probabilística. Dada esta técnica que se ha identificado como la más útil hasta el momento, se abre la puerta para poder seguir mejorando estos resultados, con bases de datos que cuenten con una mejor diferenciación y un estudio en mayor profundidad de la estructura que deberían tener las redes que se usen con este fin.

En cuanto al ensamblaje final del modelo de recuperación completo, se puede ver como existe interconexión entre todos los métodos que fueron preparados, lo que demuestra que, si bien, todos parecen ser de áreas de trabajo distintas, en realidad, todo coexiste dentro de

un mismo modelo. Esto es prueba de la real factibilidad que posee la creación de un modelo de caja gris para flotación.

Existen puntos

Finalmente, se puede concluir que se ha cumplido el objetivo general, que era estudiar la factibilidad de dar soporte al modelamiento fenomenológico mediante el uso de algoritmo de Deep Learning, obteniéndose como respuesta general que sí es posible y que existen una amplia gama de posibilidades en esta área. Además, las áreas que sí se han explorado dentro del desarrollo de esta tesis aún presentan posibilidades de mejora, lo que abre las puertas a desarrollos futuros.

Bibliografía

- [1] “Chile afianza recuperación económica en 2021 con alza histórica del pib tras restricciones por pandemia,” El Mostrador, 2022, <https://www.elmostrador.cl/mercados/2022/03/18/chile-afianza-recuperacion-economica-en-2021-con-alza-historica-del-pib-tras-restricciones-por-pandemia/>.
- [2] “Copper mining and processing: Processing copper ores,” 2021, <https://superfund.arizona.edu/resources/learning-modules-english/copper-mining-and-processing/processing-copper-ores>.
- [3] Gani, R. y Pistikopoulos, S., “Property modelling and simulation for product and process design,” *Fluid Phase Equilibria*, vol. 194-197, pp. 43–59, 2002.
- [4] WILLS, B., “Chapter 12 - froth flotation,” en *Mineral Processing Technology (Fourth Edition)* (WILLS, B., ed.), International Series on Materials Science and Technology, pp. 457–595, Amsterdam: Pergamon, fourth edition ed., 1988, doi:<https://doi.org/10.1016/B978-0-08-034937-4.50021-1>.
- [5] Bradshaw, B. y O’Connor, C., “Flotation | bubble – particle adherence: Synergistic effect of reagents,” en *Encyclopedia of Separation Science* (Wilson, I. D., ed.), pp. 1455–1464, Oxford: Academic Press, 2000, doi:<https://doi.org/10.1016/B0-12-226770-2/01311-9>.
- [6] Michaud, 2016, <https://www.911metallurgist.com/blog/porphyry-copper-flotation>.
- [7] Runge, K., “Laboratory flotation testing - an essential tool for ore characterisation,” *Flotation Plant Optimisation: A Metallurgical Guide to Identifying and Solving Problems in Flotation Plants*, vol. 16, pp. 155–173, 2010.
- [8] Taber, K. S., *Models and Modelling in Science and Science Education*, pp. 263–278. Rotterdam: SensePublishers, 2017, doi:[10.1007/978-94-6300-749-8_20](https://doi.org/10.1007/978-94-6300-749-8_20).
- [9] Ljung, L., “System Identification in a MIC perspective,” *Modeling, Identification and Control*, vol. 15, no. 3, pp. 153–159, 1994, doi:[10.4173/mic.1994.3.4](https://doi.org/10.4173/mic.1994.3.4).
- [10] Oussar, Y. y Dreyfus, G., “How to Be a Gray Box: The Art of Dynamic Semi-Physical Modeling,” *Neural Networks*, vol. 14, no. 1, pp. 1161–1172, 2001, <https://hal.archives-ouvertes.fr/hal-00922197>.
- [11] García Zúñiga, H., “La recuperación por flotación es una función exponencial del tiempo,” *Boletín Minero*, vol. 148, p. 83–86, 1935.
- [12] Laplante, A., Toguri, J., y Smith, H., “The effect of air flow rate on the kinetics of flotation. part 1: The transfer of material from the slurry to the froth,” *International Journal of Mineral Processing*, vol. 11, no. 3, pp. 203–219, 1983, doi:[https://doi.org/10.1016/0301-7516\(83\)90026-1](https://doi.org/10.1016/0301-7516(83)90026-1).
- [13] Magne, L., Jofré, J., Barria, J., y Menacho, J., “Escalamiento de resultados de flotación

- de minerales a través de la función dtr,” *Revista de Metalurgia*, vol. 31, 1995, doi:[10.3989/revmetalm.1995.v31.i1.981](https://doi.org/10.3989/revmetalm.1995.v31.i1.981).
- [14] Bu, X., Xie, G., Peng, Y., Ge, L., y Ni, C., “Kinetics of flotation. order of process, rate constant distribution and ultimate recovery,” *Physicochemical Problems of Mineral Processing*, vol. 53, no. 1, pp. 342–365, 2017, doi:[10.5277/ppmp170128](https://doi.org/10.5277/ppmp170128).
- [15] Polat, M. y Chander, S., “First-order flotation kinetics models and methods for estimation of the true distribution of flotation rate constants,” *International Journal of Mineral Processing*, vol. 58, no. 1, pp. 145–166, 2000, doi:[https://doi.org/10.1016/S0301-7516\(99\)00069-1](https://doi.org/10.1016/S0301-7516(99)00069-1).
- [16] Dowling, E. C., Klimpel, R. R., y Aplan, F. F., “Model discrimination in the flotation of a porphyry copper ore,” *Mining, Metallurgy & Exploration*, vol. 2, pp. 87–101, 1985, doi:[10.1007/BF03402602](https://doi.org/10.1007/BF03402602).
- [17] Díaz, F. y Yianatos, J., “Residence time distribution in large industrial flotation cells,” *An International Journal*, vol. 3, pp. 2–10, 2010, doi:[10.1504/AFP.2010.031015](https://doi.org/10.1504/AFP.2010.031015).
- [18] “Software,” 2022, <https://jktech.com.au/products/software>.
- [19] Schwarz, S., Alexander, D., Whiten, W., Franzidis, J.-P., y Harris, M., “Jksimfloat v6: Improving flotation circuit performance and understanding,” *IMPC 2006 - Proceedings of 23rd International Mineral Processing Congress*, 2006.
- [20] Zhang, P., “Chapter 19 - industrial control system simulation routines,” en *Advanced Industrial Control Technology* (Zhang, P., ed.), pp. 781–810, Oxford: William Andrew Publishing, 2010, doi:<https://doi.org/10.1016/B978-1-4377-7807-6.10019-1>.
- [21] Guyon, I., “Introduction to machine learning,” *Slides and Videolecture*, 2007.
- [22] Brownlee, J., “What is the difference between a batch and an epoch in a neural network,” *Machine Learning Mastery*, vol. 20, 2018.
- [23] Shen, K., “Effect of batch size on training dynamics,” 2018, <https://medium.com/min-i-distill/effect-of-batch-size-on-training-dynamics-21c14f7a716e>.
- [24] Nicholson, C., “A beginner’s guide to neural networks and deep learning,” 2020,
- [25] Learning, D., “Deep learning,” *High-Dimensional Fuzzy Clustering*, 2020.
- [26] Emmert-Streib, F., Yang, Z., Feng, H., Tripathi, S., y Dehmer, M., “An introductory review of deep learning for prediction models with big data,” *Frontiers in Artificial Intelligence*, vol. 3, 2020, doi:[10.3389/frai.2020.00004](https://doi.org/10.3389/frai.2020.00004).
- [27] Avalos, S. A., *Advanced predictive methods applied to geometallurgical modelling*. PhD thesis, 2021.
- [28] Kingma, D. P. y Ba, J., “Adam: A method for stochastic optimization,” 2014, doi:[10.48550/ARXIV.1412.6980](https://doi.org/10.48550/ARXIV.1412.6980).
- [29] Rojas, R., *Neural networks: a systematic introduction*. Springer Science & Business Media, 2013.
- [30] Sharma, S., Sharma, S., y Athaiya, A., “Activation functions in neural networks,” *towards data science*, vol. 6, no. 12, pp. 310–316, 2017.
- [31] Contreras-Reyes, J. E. y Arellano-Valle, R. B., “Kullback–leibler divergence measure for multivariate skew-normal distributions,” *Entropy*, vol. 14, no. 9, pp. 1606–1626, 2012,

[doi:10.3390/e14091606](https://doi.org/10.3390/e14091606).

- [32] Smith, L. N., “A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay,” 2018, [doi:10.48550/ARXIV.1803.09820](https://doi.org/10.48550/ARXIV.1803.09820).
- [33] Fustos Toribio, R. M., Descubrimiento de unidades geometalúrgicas por medio de análisis de conglomerados geoestadístico. PhD thesis, 2017.
- [34] Wang, L., Peng, Y., Runge, K., y Bradshaw, D., “A review of entrainment: Mechanisms, contributing factors and modelling in flotation,” *Minerals Engineering*, vol. 70, pp. 77–91, 2015, [doi:https://doi.org/10.1016/j.mineng.2014.09.003](https://doi.org/10.1016/j.mineng.2014.09.003).
- [35] Concha, M. Grandes mineras advierten caída de casi 50 % en leyes de mineral de cobre en 20 años, 2012, <http://www.economiaynegocios.cl/noticias/noticias.asp?id=97921>.
- [36] Lynnworth, L. y Liu, Y., “Ultrasonic flowmeters: Half-century progress report, 1955–2005,” *Ultrasonics*, vol. 44, pp. e1371–e1378, 2006.
- [37] Kesavan, P., M P, R., Divakaran, A., Sahoo, P., y Murali, N., “Continuous type liquid level measurement system using pulsating sensor,” 2015.
- [38] , <https://www.xore.se/#:~:text=XRF%20Instrument%20Benefits&text=With%20measurement%20times%20as%20short,XRF%20analyzers%20on%20the%20market>.

Anexo

Etapa 1: Directa				
Nº Ensayo	Estructura	EAM promedio	DKL	Total Variation
1	5 * 150 LR - 601 SIG	7,17,E-01	5,25,E-01	1,57,E-02
2	5 * 30 LR - 601 SIG	4,80,E-02	1,22,E-02	7,74,E-04
3	21 * 15 LR - 601 SIG	1,74,E-01	6,95,E-02	2,08,E-03
4	5 * 30 SIG - 601 SIG	6,67,E-01	6,01,E-01	1,32,E-02

Etapa 2: Modelos diferenciados				
1 Celda				
Nº Ensayo	Estructura	EAM promedio	DKL	Total Variation
1	550 LR - 550 LR - 600 SIG	6,52,E-02	1,17,E-02	1,04,E-03
2	5 * 50 LR - 600 SIG	1,97,E-02	2,53,E-03	4,67,E-04
3	5 * 15 LR - 600 SIG	3,24,E-02	2,88,E-03	9,23,E-04
4	21 * 15 LR - 600 SIG	1,50,E-01	2,43,E-02	5,59,E-03
2 Celdas				
Nº Ensayo	Estructura	EAM promedio	DKL	Total Variation
1	550 LR - 550 LR - 600 SIG	6,17,E-02	1,69,E-02	5,81,E-04
2	5 * 50 LR - 600 SIG	2,25,E-02	4,91,E-03	2,89,E-04
3	5 * 15 LR - 600 SIG	2,80,E-02	5,93,E-03	3,43,E-04
4	21 * 15 LR - 600 SIG	2,06,E-01	4,96,E-02	2,54,E-03
3 Celdas				
Nº Ensayo	Estructura	EAM promedio	DKL	Total Variation
1	550 LR - 550 LR - 600 SIG	4,35,E-02	1,48,E-02	3,34,E-04
2	5 * 50 LR - 600 SIG	2,47,E-02	5,99,E-03	2,67,E-04
3	5 * 15 LR - 600 SIG	2,98,E-02	7,18,E-03	2,77,E-04
4	21 * 15 LR - 600 SIG	2,56,E-01	7,71,E-02	2,17,E-03

Etapa 3: Modelos normalizados y escalados				
1 Celda				
Nº Ensayo	Estructura	EAM promedio	DKL	Total Variation
1	550 TANH - 550 TANH - 600 LR	7,04,E-03	1,13,E-04	2,73,E-04
2	5 * 50 TANH - 600 LR	6,80,E-03	1,05,E-04	2,57,E-04
3	5 * 50 LR - 600 LR	7,14,E-03	1,09,E-04	2,81,E-04
4	21 * 15 TANH - 600 LR	7,27,E-03	1,09,E-04	2,95,E-04
2 Celdas				
Nº Ensayo	Estructura	EAM promedio	DKL	Total Variation
1	550 TANH - 550 TANH - 600 LR	9,37,E-03	1,49,E-04	2,04,E-04
2	5 * 50 TANH - 600 LR	9,04,E-03	1,39,E-04	1,96,E-04
3	5 * 50 LR - 600 LR	9,59,E-03	1,52,E-04	2,05,E-04
4	21 * 15 TANH - 600 LR	9,45,E-03	1,45,E-04	2,02,E-04
3 Celdas				
Nº Ensayo	Estructura	EAM promedio	DKL	Total Variation
1	550 TANH - 550 TANH - 600 LR	1,04,E-02	1,63,E-04	1,74,E-04
2	5 * 50 TANH - 600 LR	1,01,E-02	1,51,E-04	1,69,E-04
3	5 * 50 LR - 600 LR	1,00,E-02	1,51,E-04	1,67,E-04
4	21 * 15 TANH - 600 LR	1,12,E-02	1,74,E-04	1,82,E-04

Etapa 4: Modelos con vectores de latencia				
1 Celda				
Nº Ensayo	Estructura	EAM promedio	DKL	Total Variation
1	550 TANH - 550 TANH - 600 LR	1,46,E-02	3,81,E-04	6,02,E-04
2	5 * 50 TANH - 600 LR	1,08,E-02	2,10,E-04	4,58,E-04
3	5 * 50 LR - 600 LR	1,04,E-02	1,95,E-04	4,44,E-04
4	21 * 15 TANH - 600 LR	7,40,E-03	1,13,E-04	3,00,E-04
2 Celdas				
Nº Ensayo	Estructura	EAM promedio	DKL	Total Variation
1	550 TANH - 550 TANH - 600 LR	1,45,E-02	3,47,E-04	3,10,E-04
2	5 * 50 TANH - 600 LR	1,12,E-02	2,00,E-04	2,30,E-04
3	5 * 50 LR - 600 LR	1,10,E-02	1,98,E-04	2,30,E-04
4	21 * 15 TANH - 600 LR	9,34,E-03	1,42,E-04	2,03,E-04
3 Celdas				
Nº Ensayo	Estructura	EAM promedio	DKL	Total Variation
1	550 TANH - 550 TANH - 600 LR	1,36,E-02	2,67,E-04	2,40,E-04
2	5 * 50 TANH - 600 LR	1,03,E-02	1,56,E-04	1,73,E-04
3	5 * 50 LR - 600 LR	1,07,E-02	1,72,E-04	1,76,E-04
4	21 * 15 TANH - 600 LR	1,58,E-02	3,19,E-04	2,34,E-04