UNIVERSIDAD DE CHILE

FACULTAD DE FILOSOFÍA Y HUMANIDADES

ESCUELA DE POSTGRADO

DESCRIPTIVE ANALYSIS OF THE ACQUISITION OF THE BASE FORM, THIRD PERSON SINGULAR, PRESENT PARTICIPLE REGULAR PAST, IRREGULAR PAST, AND PAST PARTICIPLE IN A SUPERVISED ARTIFICIAL NEURAL NETWORK AND AN UNSUPERVISED ARTIFICIAL NEURAL NETWORK

Tesis para optar al grado de Magíster en Lingüística con mención en Lengua Inglesa

Martín Flores Quiroz

Profesor Guía: Saeid Atoofi

Santiago de Chile, año 2013

DEDICATORIA

Le dedico esta tesis a Natalia que ha estado acompañándome en este proceso y ha sabido apoyarme durante todo momento. Es bastante probable que sin ella el desarrollar esta investigación y escribirla hubiese sido insoportable.

También quisiera dedicar este trabajo a mi familia: a mis padres que siempre nos instaron a dar lo mejor; a mis hermanas, cuyo amor por las ciencias fue parte de la inspiración para esta investigación, en especial a Ángela quien siempre estuvo dispuesta a contestar mis preguntas y a ayudarme; y a mis cuadrúpedos amigos que aunque aún no puedan leer esto, siempre han sabido prestar apoyo.

A mi Mami Tencha, que nos enseñó a todos a soñar en grande.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Saeid Atoofi, for giving me the freedom to pursue this research, while providing critical and valuable suggestions to improve my work.

Also, I would like to thank the whole department of linguistics of Universidad de Chile. All the knowledge and tools they provided me during my BA and my MA have molded me into a professional and better person. I would like to especially thank Carlos Zenteno, whose unrelenting effort as a professor and researcher has inspired me to pursue linguistic research on the academic level.

Many thanks to Mario Peralta, the interesting discussions on artificial intelligence (among all the intellectual discussions we have had) kindled my interest on interdisciplinary approaches to linguistic research.

All my gratefulness to Professor Brian MacWhinney for providing me with the dataset necessary to carry out this research.

Finally, I would like to thank the academics from the departments of Computer Science and Psychology from the University of Birmingham, who provided me computational knowledge needed in this research.

RESUMEN

NOMBRE DEL AUTOR: MARTIN ANDRES FLORES QUIROZ

PROFESOR GUIA: SAEID ATOOFI

GRADO ACADEMICO OBTENIDO: MAGISTER EN LINGÜÍSTICA CON MENCION EN LENGUA INGLESA

FECHA DE GRADUACION:

DATOS PERSONALES DEL AUTOR:

DIRECCION: ESCOCIA 5458, LO PRADO, SANTIAGO.

TELEFONO: 28394950

DIRECCION CORREO ELECTRONICO: FQ.MARTIN @GMAIL.COM

Contents

LIST OF FIGURES

LIST OF TABLES

1. RESEARCH PROBLEM

Studying children's language acquisition in natural settings is not cost and time effective. Therefore, language acquisition may be studied in an artificial setting reducing the costs related to this type of research. By artificial, I do not mean that children will be placed in an artificial setting, first because this would not be ethical and second because the problem of the time needed for this research would still be present. Thus, by artificial I mean that the tools of simulation found in artificial intelligence can be used. Simulators as artificial neural networks (ANNs) possess the capacity to simulate different human cognitive skills, as pattern or speech recognition, and can also be implemented in personal computers with software such as MATLAB, a numerical computing software. ANNs are computer simulation models that try to resemble the neural processes behind several human cognitive skills. There are two main types of ANNs: supervised and unsupervised. The learning processes in the first are guided by the computer programmer, while the learning processes of the latter are random.

The purpose of this research was to simulate the acquisition of six English grammatical structures in affirmative mood, the base form, the third person singular, present participle, regular past, irregular past and past participle, in two ANNs with different learning strategies. The contrast among these two ANNs may help us to discern which of these might be more useful to simulate and study the process of language acquisition based on the simulated human cognitive skills.

The above mentioned grammatical structures were implemented because the language learning potential of ANNs is limited when compared to human learning. Elman (1993) suggests that when it comes to language related skills, ANNs have to start small in order to be successful. If they are presented with all the input children receive, the results are disastrous. The acquisition of English past tense has been studied before with contrasting results but always with one critique made by innatists: connectionist models are not able to have the exact same performance as children do.

Rumelhart and McClelland (1986) and MacWhinney and Leinbach (1991) devised connectionist models that tested the acquisition of the English past tense based on the phonological production of past tense forms from a base form. Their results were contrasted with the performance of children achieving similar results as human subjects, especially in the case of regular past tense formation in training, but very dissimilar results on irregular past tense formation in training and tests that involved regular past tense production.

Following Elman's suggestions, the connectionist model studied here will not involve production of past tense. Instead, it will focus on the ability of ANNs to perceive a verb form and classify it into one distinctive class.

2. RESEARCH QUESTIONS:

1. Which is the learning potential of a supervised ANN when it has to learn to classify the base form, third person singular, present participle, past participle, regular past, and irregular past in the indicative mood of English?

2. Which is the learning potential of an unsupervised ANN when it has to learn to classify the base form, third person singular, present participle, past participle, regular past, and irregular past in the indicative mood of English?

3. Is there a significant difference in the classification potential of the base form, third person singular, present participle, past participle, regular past, and irregular past in the indicative mood of English in this two ANNs?

4. Is there a significant difference in the learning potential of a supervised ANN and an unsupervised ANN when the amount of verb forms is reduced from six, to four and then to three?

5. Which of these two ANNs assimilate a learning potential similar to that of a human?

3 OBJECTIVES

3.1 GENERAL OBJECTIVE:

1. To determine which type of ANN, supervised or unsupervised, is more suited for the task of simulating the process of learning the base form, third person singular, present participle, past participle, regular past, and irregular past in the indicative mood of English.

3.2 SPECIFIC OBJECTIVES:

1.1 To identify the learning potential of the base form, third person singular, present participle, past participle, regular past, and irregular past in English by simulating these instances in a supervised ANN.

1.2 To identify the learning potential of the base form, third person singular, present participle, past participle, regular past, and irregular past in English by simulating these instances in an unsupervised ANN.

1.3 To identify the limits of the learning potential of verb forms in English by simulating these instances in a supervised ANN and an unsupervised ANN.

1.4 To contrast the classification potential of the base form, third person singular, present participle, past participle, regular past, and irregular past in English of a supervised ANN with an unsupervised ANN.

4. THEORETICAL FRAMEWORK

4.1 THE HISTORY OF ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks (ANNs) now belong to the field of Artificial Intelligence (AI), which can be roughly defined as “the study and design of intelligent agents”[1]. AI is immersed in the broad field of computer sciences which means that ANNs are also part of this field. This is not entirely true because ANNs did not always belong to the field of computer science. At first, they were created as a hypothesis of what happened in the brain when different processes were occurring. In this sense, the first approaches to ANNs came from psychology with the work of McCulloch and Pitts (1943), Hebb (1949) and Rosenblatt (1957). As a theory of mind, ANNs can be useful if they are used to test and analyze the theories that try to explain how the different processes in our brain occur. As a means for creating intelligent systems, ANNs have improved the systems of voice recognition, pattern recognition, finance and weather forecast.

In the following subsections, the first steps of ANNs as a theory of mind will be reviewed. Then, a brief summary of ANNs’ computer-related applications will be provided. Finally, ANNs will be related to the philosophical theories of connectionism and emergentism.

4.1.1 EARLY YEARS.

McCulloch and Pitts (1943) were the first to apply the concept of artificial neuron to understand the processes that occurred in the brain. In their paper ‘A logical calculus of the ideas immanent in nervous activity”, they stated the foundations to a field that would flourish for three decades before Minsky and Papert (1969) almost destroyed the field of artificial neural networks by making explicit its main flaws.

---

[1] Poole, Mackworth and Goebel 1998, 1.

In order to arrive to the idea of ANNs, McCulloch and Pitts spent several years working in psychology and biophysics respectively. After meeting, they realized that they shared some ideas on how biological neural networks work in the brain based on inputs gathered from the environment to produce outputs. Knowing that they were working on a first theory of mind and brain, they defined a set of simple conjectures about the conduct and performance of biological neural networks.

Before giving a review of the way an ANN works it is necessary to introduce some of the technical terminology used by the first researchers in this field. First, we need to understand the notion of input which can be defined as the information provided by the researchers or the environment. ANNs can learn by interacting with the input provided by the researchers or the environment. This interaction produces output, an outcome, which may or may not be right depending on the expected results. The output is produced if the stimulus created reaches a certain threshold; if the output is produced, the stimulus is deemed as excitatory; if no output is produced or if it does not correspond to the expected outcome, the stimulus is deemed as inhibitory. Finally, as ANNs resemble biological neural networks, we can also talk about synapse. Synapse is a junction by which electrical or chemical signals are passed from one neuron to another neuron or cell. The synaptic strength between an axon A and an axon B is called weight.

McCulloch and Pitts article is framed in the field of mathematical or predicate logic. In order to understand their work, it is necessary to introduce some of the notations used by them. First there is the notion of predicate. In simple terms, a predicate is any statement that can be assigned the values true or false. Two or more predicates can be connected by means of logical functions such as AND, OR or NOT. The logical function AND is also known as logical conjunction. This function is an operation that holds the value true if all its predicates are true. The logical function OR is also known as logical disjunction. In this operation the true value can be hold if one or more of its predicates are true. Finally, we have the logical function NOT which is also known as logical negation. This logical operation holds the value true if its predicate is false.

McCulloch and Pitts argued that their ANN could compute any logical expression by using the logical functions mentioned above. In order to do so, they defined a predicate or predicates that by means of logical functions would held the values true or false. If the value was true the artificial neuron would fire a signal to produce an output. The following figure shows the ANNs that accomplished the functions mentioned above.

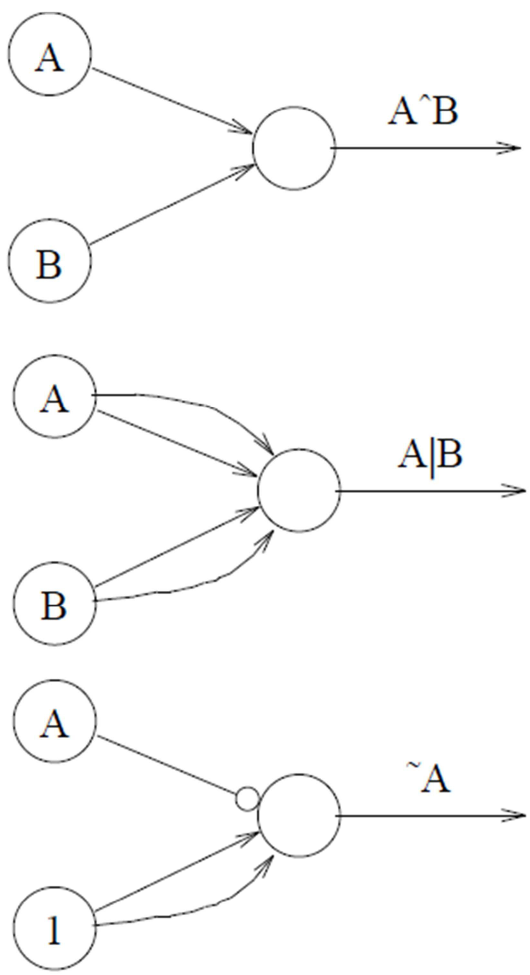

Figure 1. Graphic representation of the Boolean functions AND, OR and NOT in an ANN.

The basic ANN described by McCulloch and Pitts had a finite threshold and the input, based on its weight, could generate an excitatory or inhibitory stimuli. This first approach to artificial neural networks did not take into account the notions of memory and learning which would be introduced later by Donald Hebb (1949).

Although they claimed that in their paper they were presenting results for the research based on their theory, their conclusions were not reached by the researchers in that period of time because part of them was deemed as obscure:

> *"The present article is partly an exposition of their results; but we found the part of their [McCulloch and Pitts'] paper dealing with arbitrary nerve nets obscure, so we have proceeded independently there."* (Kleene, 1956:4)

Nonetheless, the importance of McCulloch and Pitts' work has to be acknowledged. Their simplification of neural networks to compute different phenomena allowed researchers from different fields to theorize and study a theory of mind and brain.

As stated above, their model was highly simplified, particularly because they did not have the technological resources that are available to us today. However, their theoretical work prepared the ground for the implementation of ANNs, such as Hebb's notions of memory and learning (1949) and Rosenblatt's Perceptron (1957) which was the first prototype of ANN's pattern recognition.

Noticing how simplified the first model of artificial neural networks was, Hebb theorized that ANNs could learn by means of the connections they established among themselves. These connections were not meant to be fixed because the constant firing of an excitatory stimulus would improve of the 'efficiency' of this network:

> *"When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process*

or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased." (Hebb, 1949: 62)

This learning process through modification can be considered as memory because the axons of the ANNs recall the frequency of firing between ANNs. Hebb's work has had consequences that influenced further research on this subject.

Although Hebb's theory of learning and memory was not described completely to be analyzed, it inspired a new of line of research in ANNs. One of the people inspired by Hebb's theory was Frank Rosenblatt who in 1957 invented an ANN named perceptron. Rosenblatt tried to implement a type of intelligent system that would emulate the way in which the human eye works to classify different classes of objects. At first, the perceptron would classify very simple objects as part of its learning process. After the learning process had started, the perceptron would learn to classify complex objects by readjusting the weights that would trigger an inhibitory or excitatory stimulus. The readjustment was a random process based on the laws of probability. It is said that the learning process found in Rosenblatt perceptron is founded on Hebb's notions because the perceptron learnt by reinforcing its connections and by modifying its weights.

Figure 2 shows the resemblance between a perceptron and human eye. Each artificial neuron finds the objects which are similar from a set of objects and classifies them.

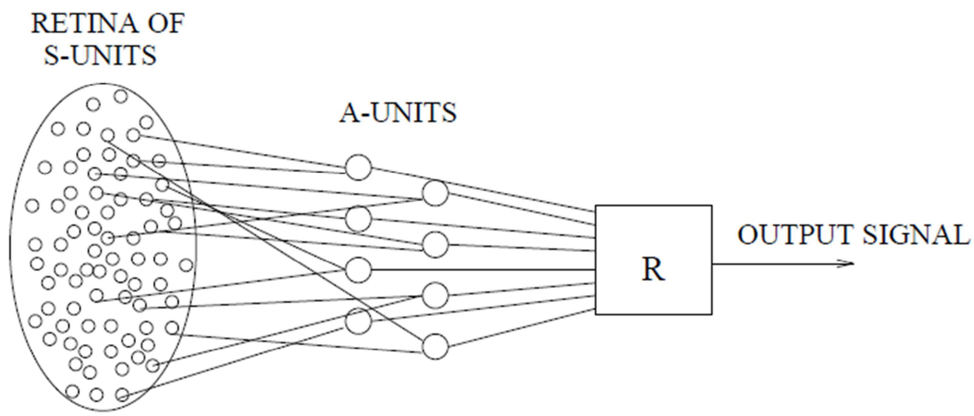

Figure 2. Graphic representation of the resemblance of a human eye and a perceptron.

As noted above, the perceptron can classify simple classes of objects, which does not necessarily mean that it could perceive and classify reality as human do. This shortcoming and the ones found on previous research were not noticed by the researchers in the first years of ANNs. In fact, McCulloch and Pitts, Hebb and Rosenblatt overestimated the potential of ANNs. One of these claims boldly states that the perceptron reached a potential similar to the human brain:

> *"It seems clear that the Class C' perceptron introduces a new kind of information processing automaton: for the first time we have a machine which is capable of having original ideas"* (Rosenblatt, 1959: 449)

These types of claims and the lack of evidence to support them would eventually lead to the downfall of the research in ANNs. Minsky and Papert published their book 'Perceptrons' after an exhaustive analysis of the potential of perceptrons. As stated above, researchers at that time had high expectations for this type of ANNs. Unfortunately, Minsky and Papert's work would deter research in the field of ANNs for more than a decade.

Their analysis began by stating the different orders of the functions ANNs could compute. AND, OR and NOT are functions that compute first-order predicates which do not allow the use of predicates or functions as arguments. Rosenblatt's perceptron was supposed to classify objects into sets using a different function called exclusive-or (XOR). This function holds true when one of the predicates is true while the other is false. This differs from the first-order disjunction because OR holds true if at least one of the predicates is true. Trying to compute more than two inputs with a perceptron and an XOR function would be doomed to failure because at least one of the inputs would be too large for the perceptron to compute. Thus, Minsky and Papert's work showed that perceptron could classify if the amount of the objects or the size of the area was small. This clearly crushed the previous ideas about perceptrons and ANNs in general, showing that they could not process the amount of information the researchers claimed they could.

Not only did Minsky and Papert provided solid arguments to doubt the classification potential perceptrons were claimed to have; they also questioned the claims about their learning memory. As the number of predicates raised, the number of artificial neurons and connections raised as well. This meant that for perceptrons to compute the classification problem of n-objects, they would have to have an n-number of artificial neurons eventually making the automaton crash.

After Minsky and Papert's book was published, research in the field of ANNs became unpopular and funding was hard

Since McCulloch and Pitt's and Rosenblatt's ANNs, and the debacle brought by Minsky and Papert the manner in which ANNs learn has been improved and tested by the new technologies that have been developed.

4.1.2 THE REBIRTH OF ANNS.

In 1982, John Hopfield published a series of papers on what he called ‘Hopfield networks’. These are networks of ANNs with binary thresholds whose values can be updated after each iteration. Units in these networks are symmetrically connected to other neurons, i.e. the same number of connections for every unit. Also, no unit can be connected with itself. Hopfield (1982) found that this avoids chaotic behavior in his networks. The connection among units allows the network to have memory. In this sense, this ANN can be tested after being trained by giving it faulty data, and if the ANN learnt properly, it will be able to reconstruct the data from its memory. These two characteristics, updating and memory, were crucial to the rebirth of this field.

In 1982 as well, Teuvo Kohonen developed a self-organizing map (SOM), a type of ANN that uses an unsupervised learning algorithm. SOMs consist of neurons which are related to a weight vector with the same dimension as the input. Each neuron is given a position in a two-dimensional map space. The training data is presented iteratively to ensure that only one neuron of the map is associated to one weight vector. This first step, account for the training of the ANN. The testing phase is called mapping (Kohonen, 1982). In this phase, new data is used as input and the SOM classifies each new input based on the distance, i.e. how similar or different they are, between its weight vector and the one found in the map. As in the case of Hopfield networks, SOMs can be updated and have the ability to match new input to a position in the map space based on its resemblance. This means that SOM can generalize, i.e. classify input that has not been used before properly.

Finally, the rebirth of ANNs is mostly associated to the use of the back propagation algorithm. Back propagation means that the error, i.e. the difference between the desired output and the output produced by the ANN, is propagated backwards to be taken into account in the next iteration of training. This is a type of supervised learning network because there is a desired output. Training in back propagation networks is divided in two parts. First, the input is propagated through the network to produce an output. Then, this output is compared to the desired output to produce the difference between them. Second, this difference is used to update the initial weights of the neurons to finds ones that minimize the difference between the produced output and the desired output. ANNs that use back propagation are able to solve nonlinearly separable problems as the ones presented by Minsky and Papert (1969) which overwhelmed perceptrons at their time. In order to minimize the difference between outputs, an extra layer of neurons, a hidden layer, has to be programmed as well. In the XOR problem, the hidden neuron is in charge of computing the inputs to solve this problem that once defeated ANNs.

4.1.3 SOME USEFUL NOTATIONS USED IN ANNS.

This section will introduce some of the notation used in this field which will be put into used in section 4.1.5 where two examples of ANNs are given.

Activation function: It is a mathematical algorithm that describes the relationship or mapping from input to output. There are different types of functions being some of them the following:

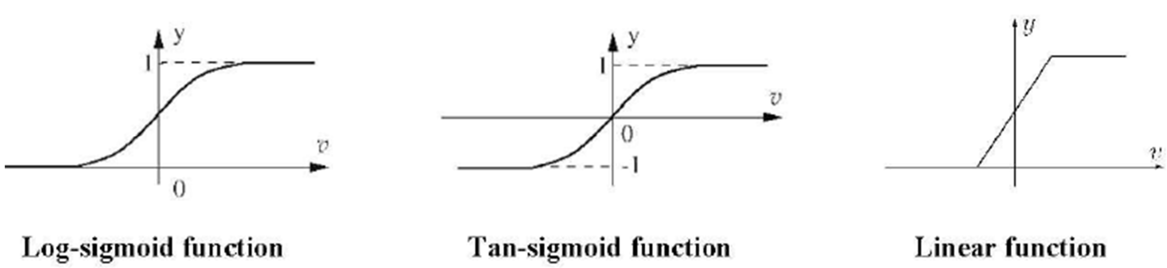

Figure 3. Examples of activation functions.[1]

[1] http://www5.in.tum.de/wiki/index.php/File:Activation_functions.PNG

Cross entropy: it is a way to measure the error in which the distance between the target output and the output is measured. In this particular way of measuring the error, the output represents the probability of a particular class of being true for an input.

Epoch: it is when all the training patterns have been run once.

Underfitting: it is when the ANN does not learn the training patterns and cannot classify properly.

Overfitting: it is when the ANN learns the training patterns in such a way that if new data does not fit the training patterns completely, it is not able to classify them correctly.

Generalization: the capability of the network to correctly classify patterns it has not seen before.

Overshoot: it is when the output exceeds its target.

Learning rate (eta): it is a parameter that states the speed at which an ANN learns.

For-loop: a statement that permits a code to be executed iteratively.

4.1.4 CONSIDERATIONS ABOUT THE TRAINING.

Ideally, an ANN after training has to be able to match its inputs into the desired output. Unfortunately, sometimes that is not the case or perfectly matching inputs into outputs may produce overfitting which has to be avoided. Therefore, the training has to be stopped at some point. This is a decision that has to be taken beforehand and has to be encoded in the code that trains the ANN.

First, the amount of epochs needed to successfully train the network. This decision, in the same manner as the ones which are going to be discussed later, depends on the type of ANN, the activation function being used and experience. A small amount of epochs may produce underfitting, whereas a large one might be unnecessary because the ANN has already learnt or it has not learnt yet but no further improvement can be done. Furthermore, depending on the amount of computations in the code, a certain amount of epochs might take too long to be reached.

Second, if the error in the validation sets starts to rise, that is a good indicator that no further improvement in training can be done. Specially, because there is a correlation between the validation error rising and a decrease in the generalization capability of the network.

Third, a small amount of error can be allowed so as to have good generalization. As stated above, the decision on how small an error can be depends on the type of network and the experience of the programmer.

4.1.5 TWO IMPLEMENTATIONS OF ANNS.

In this section, two implementations of ANNs are going to be thoroughly presented and discussed to exemplify their real world and psychological applications, and their coding and architecture. Both of these implementations were assessments in two modules of the MA in Philosophy of Mind and Cognitive Science I studied at the University of Birmingham during 2012 and 2013.

The first ANN is related to a psychological application. Its task is to simulate how biased competition works in visual search. The aim of this assessment was to test the notion that visual search is biased in nature. This notion is based on the work of Desimone and Duncan (1995). First, humans have a “limited capacity for processing information” (1995:193). This means that if we pay attention to an object, the available attention to notice others is reduced. Second, we have “the ability to filter out unwanted information” (1995:193). This process is known as selectivity. This means that when we notice attended stimuli, we are mostly unaware of unattended stimuli. Based on these two premises, they state that inputs have to compete for attention and that this competition is biased. The bias favors stimuli that are pertinent for the current situation. The mechanisms underlying the biased competition theory are based on bottom-up and top-down biases. On the former, as an example, it is easy to find a target in a group of homogenous distractors. On the latter, stimuli can be biased by our previous knowledge of what is relevant for the situation.

4.1.5.1 PSYCHOLOGICAL IMPLEMENTATION OF VISUAL SEARCH.

Hickey and Theeuwes (2011) consider that biased competition is carried out by inhibition produced by neurons. Furthermore, they state that competition through inhibition can be deemed as “visual attention” (2011:2054).

Wolfe (1998) also presents a model in which bottom-up processes work side-by-side with top-down processes. For him, there is always a first pre-attentive stage in which the different stimuli are activated based on the features they have. After this, a top-down mechanism based on the knowledge of the target is activated. Finally, each stimulus receives a ranking from the highest to the lowest based on the combination of both processes. The ranking serves to signal the places that require attention. In this model, visual search would go from the highest salience stimulus in descending order until the target is found or stop if the remaining stimuli do not reach a certain threshold. If the features of the target become closer with the ones of the distractors, the target might lose its ranking and serial searches would have to be done. On this matter, Trappenberg (2010) considers that serial search is apparent in nature and that its role is active once there is an “intense conflict-resolution demand in the recognition process.” (2010:295). For him, parallel search is the main mechanism involved in visual search (Trappenberg, 2010).

Some of the shortcomings of Wolfe's model are noted by Heinke and Humpreys (2005). The first is that the "serial search is not implemented in a manner consistent with the connectionist architecture of the earlier stages of the model" (2005:9). The second is the model's inability to group visual features. This means that in order to distinguish stimuli from distractors they have to be considerably different according to the relevant features of that specific task. Moreover, in the case that a target is grouped with distractors, it can become difficult to detect it and a distractor may be perceived as having a higher ranking.

Wolfe (1998) also notes that there is not a clear distinction between these parallel and serial search processes because different tasks of visual search can be explained by both processes depending on the degree of complexity of the task. For example, a feature search, which is in principle a parallel mechanism, can become a serial mechanism if the distractor is more similar to the target or if distractors are heterogeneous (1998:20). In this sense, he describes searches based on their reaction time (RT). Based on the visual task, a search can be efficient if the search slopes, which relate the RT to the number of distractors present (Heinke and Humpreys, 2005:2), are near zero msec/item whereas a search would be inefficient if the slopes are near 20 msec/item.

Trappenberg (2010) noted that the reaction time (RT) of visual search is not affected by the amount of distractors when they are very different from the target. On the other hand, the RT of visual search is indeed affected by the number of distractors when they are similar to the target.

In order to support the biased competition theory, a computer simulation of visual theory was carried out in MATLAB. First, a winner-take-all (WTA) network was set up. In this type of network, inputs compete to be activated (Trappenberg, 2010). To do so, all the inputs are connected to a strong inhibitory node that feedbacks and update the inputs. This is a recursive process that eventually activates only one neuron, the one with the strongest input. The number of iterations or time steps necessary to reach a winner depends on how strong the competing inputs are. This is similar to Desimone's and Duncan's (1998) theory because stimuli compete for visual attention. In order to calculate how efficient visual search is in this simulation, the number of time steps necessary to reach a clear winner is used as input for a function that calculates the reaction time slopes. The number of time steps spent on each item decides if the search was efficient or not.

In this particular simulation, two matrixes were used as input. On the one hand, the first matrix had low saliency (LS) distractors, this means that the target popped out easily. On the other hand, the second matrix had high saliency (HS) distractors; this means that the distractors were similar to the target making the visual search more difficult. Each matrix was composed of four items. In order to set the RTs to calculate the slopes for each matrix, thresholds were predetermined. For both matrixes the threshold was 0.7. This means that at the moment one neuron reached that threshold, it would be declared the winner. As noted by Wolfe (1998), we can set the threshold based on the needs of the experiment: a conservative threshold would minimize errors whereas a liberal one would minimize RTs. After the RTs for both matrixes were found, a new distractor was added to each matrix and the WTA network was run again. After this, the process repeated one more time. The purpose of adding distractors was to test if the RT of the network increased or not.

Having the RTs for the sets of matrixes, a function that calculates the slopes was run. This function needs at least two RTs and two numbers of items. The results for the LS and HS matrixes are presented as follows.

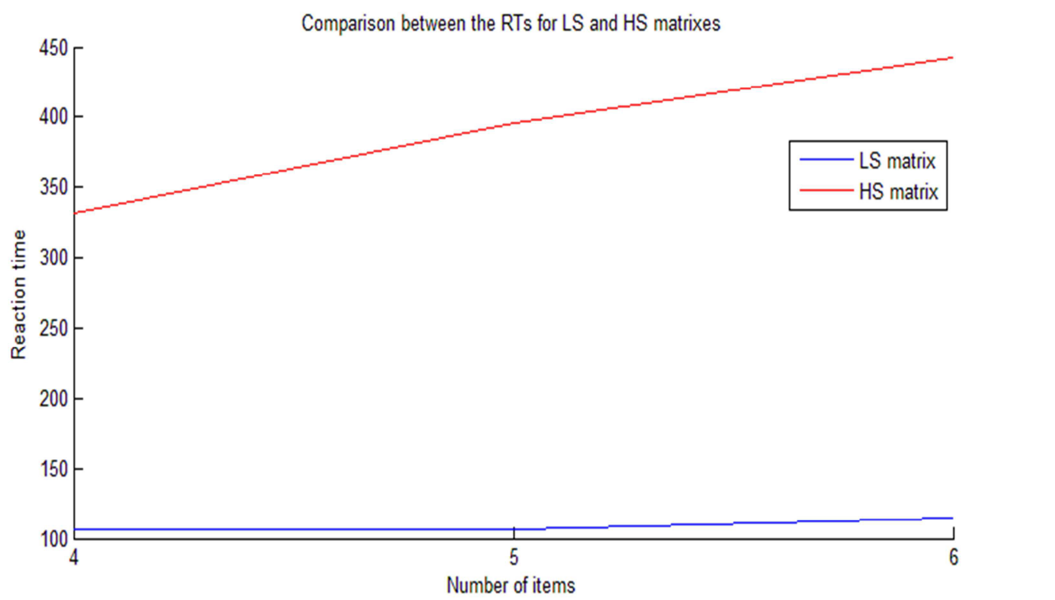

Figure 4. Comparison between the Reaction Time and the number of items for the Low Saliency matrix and the High Saliency matrix.

As it was expected, the RTs and slopes were lower when visual search was performed in the LS distractors matrix. The RT for the four item LS matrix was 106; 107 when an extra item was added; and 115 when two extra items were added. The search slope was 4.5 time steps per item.

In contrast, the RTs and slopes were higher when visual search was performed in the HS distractors matrix. The RT for the four item HS matrix was 332; 396 when an extra item was added; and 443 when two extra items were added. The search slope was 55.5 time steps per item.

The results of these simulations support the biased-competition model because the distractors found in the matrixes are not paid attention to, i.e. they are not activated, by the inhibitory feedback mechanism in the WTA network. This mechanism, after a series of time steps, picks out the most salient item. After each time step, less activation is given to the irrelevant items until the target is the only item being activated. This process of activation is quite efficient when the matrix has LS distractors with a search slope of 4.5 time steps per item. Although search becomes inefficient when the matrix has HS distractors, the winner item is picked out by a biased process since the distractors are inhibited after each time step.

The inhibition process found in the WTA network acts as the bias found in the biased competition model. This inhibition is founded on experimental data as stated by Trappenberg (2010) and Hickey and Theeweus (2011). This network cannot be used as a simulation of the guided search model since no ranking is created and only one location is picked out to be paid attention.

One of the shortcomings of this simulation is that it was predicted that after each time we added a LS distractor, the RT should not increase (Trappenberg, 2010:294). Unfortunately, this was the case. Nonetheless, the increase of the RTs after adding LS distractors was not dramatic. If instead of having an extra LS distractor, the matrix had an empty location, a zero in this case, the RT did not increase. Another shortcoming is the fact that the bias of the simulation is directed towards the less salient items having a strong bottom-up component. This would leave aside top-down processes that can also bias visual search.

4.1.5.2 OPTICAL CHARACTER RECOGNITION.

The second ANN is related to a real world application. Its task was to perform optical recognition of handwritten digits. The aim of this assessment was to compare the performance of an ANN that underwent two different types of training: online and batch training. In online training, the weights are updated immediately after each instance is used for training. Whereas, in batch training, the weight update is performed after all the weight changes are added up once all the examples from the training set have been presented.

The data set used in this implementation comes from the UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. The inputs came from forms filled by 43 individuals. These images were pre-processed and the result for each image was a 1 by 65 matrix, in which the first 64 elements were integers, numbers that can be written without a decimal component, ranging from 0 to 16 and 65th element was the class that the matrix belonged to.

This data set is composed of 3823 instances for the training set and 1797 instances for the testing set. Each instance has 64 input attributes and 1 class attribute. Input attributes are integers that range from 0 to 16. The class attribute range from 0 to 9.The class distribution for the training set is presented in Table 1 and the class distribution for the testing set is presented in Table 2.

Table 1. Class distribution of training set

Class	Instances
0	376
1	389
2	380
3	389
4	387
5	376
6	377
7	387
8	380
9	382

Table 2. Class distribution of testing set

Class	Instances
0	178
1	182
2	177
3	183
4	181
5	182
6	181
7	179
8	174
9	180

The architectural design of this ANN is described as follows. It composed of 64 input neurons, 30 hidden layer neurons and 10 output neurons. The activation function we used for the hidden layer was the sigmoid function and the activation function for the output neurons was the softmax function. Based on (Gurney, 1997; Haykin, 1999) a suitable architecture for this kind of network would include the sigmoid function in the hidden layer and the softmax function in the output layer. Namely, the reason why I used the softmax function is because it is best suited to deal with more than two classes. In this case, the task involved classifying the inputs into one of ten classes.

As discussed in section 4.1.4, the training of an ANN has to be stopped based on several considerations. These considerations are not completely theoretical because they also depend on experience and the type of ANN. The first consideration is the computation time required to successfully train a network. In this case, after some experimentation it could be seen that the base network performed well after 30 epochs, therefore 100 epochs seemed a reasonable time. The second consideration was to avoid overfitting. It was decided that if the validation error was higher than the minimum validation error computed to that epoch, for ten consecutives epochs, the training algorithm should stop. The third consideration was the performance goal. In case the total cross entropy error of the training set is below a certain number, the training should stop because the algorithm has converged and there is no need to continue with the training.

For building the neural network with online training, the design was based on John Bullinaria's Step by Step Guide to Implementing a Neural Network in C. He is a senior lecturer at the University of Birmingham and has implemented several ANNs to model brain damage and language processing tasks.

The main difference is that the network for the current simulation was built in MATLAB. Firstly, I created a for-loop to train the network for 100 epochs. Secondly, I uploaded both training and testing sets. The training set was split in a 7:3 ratio;

the latter was used for validation. Thirdly, the parameters for the network were set: J, the number of neurons in the hidden layer was set to 30; m, the shape of the slope for the sigmoid, was 0.01; s, the bias for the sigmoid, was 0; eta, the learning rate, was 0.7; the maximum amount of epochs was 100; the maximum amount of epochs that the net could fail was 10; two matrixes were set for the errors of the training and validation sets; two random weight matrixes were set for the connections from the input units to the hidden layer and from the hidden layer to the output layer; a counter was set up in 0; and both the total error and the minimum error were set to infinity. Next step was to initialize the weight as two random matrixes, one for the connections from the input units to the hidden layer and other from the hidden layer to the output layer. Then a for-loop was created to train the network for 100 epochs. Another for-loop was added to present all of the training examples. Then the output, the error and the weight updates were computed for each layer. After the weight updates, the validation check was implemented as the computation of the network output for each validation examples and the validation error check. Finally, when the training stopped, the performance was calculated for the testing set.

The pseudo code I used is presented in Appendix 1. It is slightly different for both types of training. The difference will be explained below.

The first network had 30 weight connections with the hidden layer and the hidden layer had 10 connections to the output units. The number of hidden units and therefore the number of connections for online training was later changed for one of the experiments.

There are two main differences in architecture between the network with online training and the one with batch training. The first difference was already explained previously. Namely, in batch training all the weight changes are summed once all the examples from the training set have been presented. After that, the weight update is performed. The second main difference is a consequence of the previous one. The learning rate for batch training has to be lower than online training (Wilson and Martinez, 2003). Otherwise it may overshoot. That is why the learning rate of batch training is divided by the amount of training examples. For the hidden layer a smaller learning rate was used. This is based on the following. After several attempts with one eta, the learning process was slow. Therefore, a second eta was introduced and adjusted to avoid overshooting and to have an efficient training process.

After the weights were updated and the stopping criterions were met, the testing set was run. In order to decide if the classification was carried out well, a 0.5 error was tolerated. This means that if the output was 0.5 or bigger, then it would be considered as 1. Otherwise, it would be considered as 0.

For this assessment, I decided to experiment with online training and batch training. The idea behind this is to compare the performance of both types of training for this task. Then, I also experimented with 3 different learning rates: 0.0.7, 0.7 and 7. To conclude with the experiments, I changed the amount of neurons of the hidden layer 4 times: 5, 15, 30 and 45.

The results of these three experiments are presented below. Each experiment is accompanied by graph showing the performance of the network.

The graph below shows the training error for both trainings.

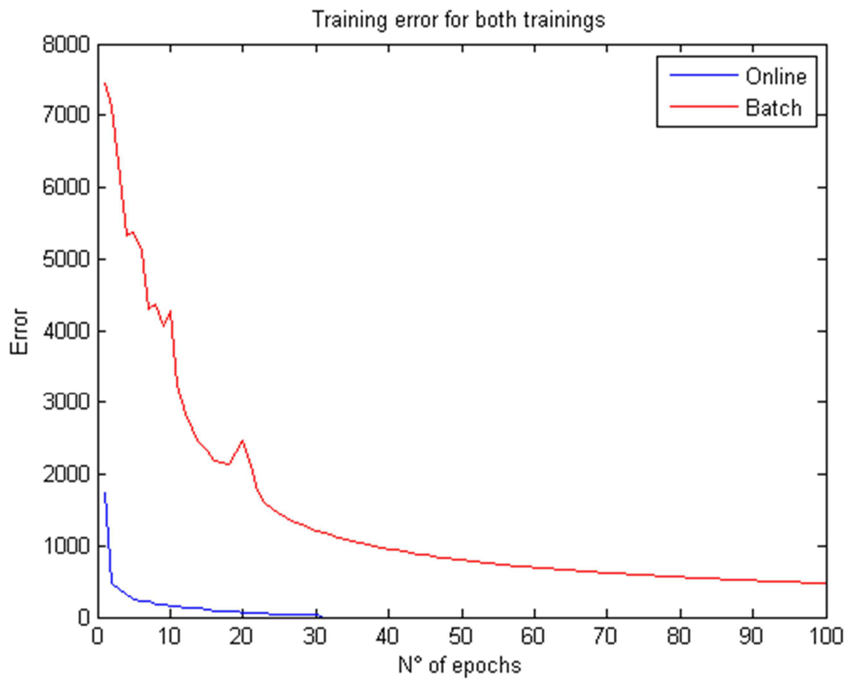

Figure 5. Training error for online training and batch training.

This graph shows that the training error decreases quite fast for the online training. Although, there is a considerable decrease for batch training, we can see that more epochs are needed to keep improving the training of the network.

The graph below shows the validation error for both trainings.

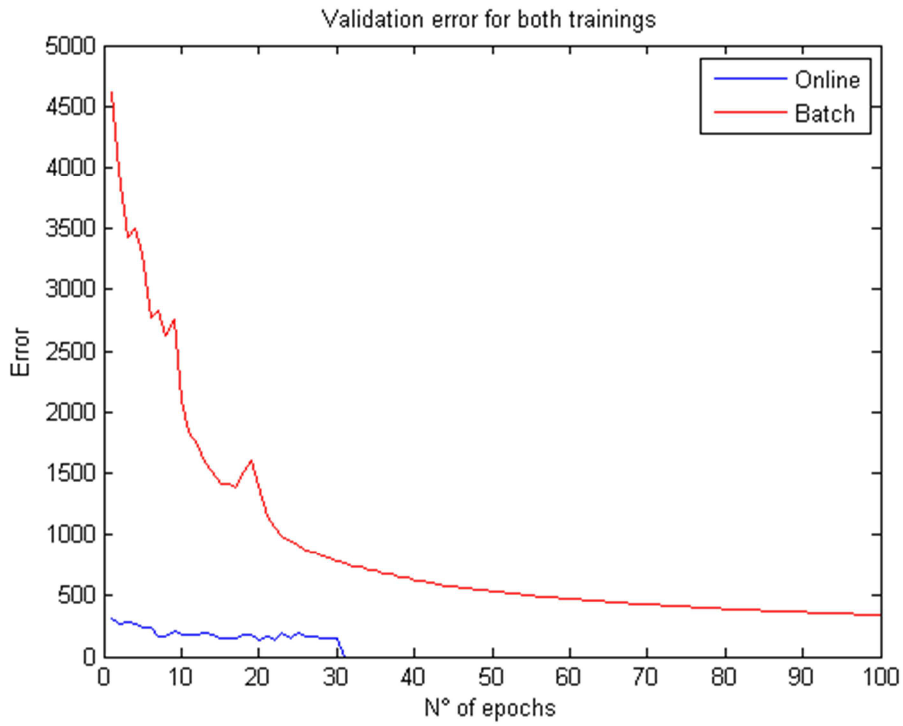

Figure 6. Validation error for online training and batch training.

In this graph we can see that few adjustments were done to the weights of online training after validation took place. In contrast, a great amount of adjustments were made for batch training, and as in the training, more epochs would be needed to improve the network.

The graph below shows the training error for online training with different etas.

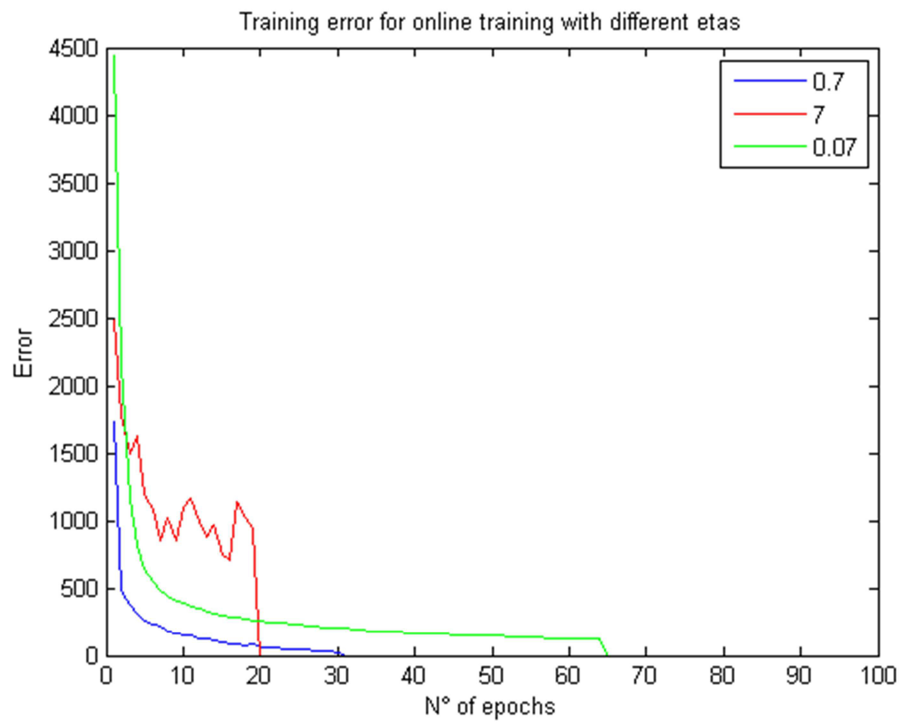

Figure 7. Training error for online training with different learning rates.

We can see that the training error for an eta of 7 is abruptly stopped to avoid overfitting. Meanwhile, more epochs are needed to train the network with an eta of 0.07.

The graph below shows the validation error for online training with different etas.

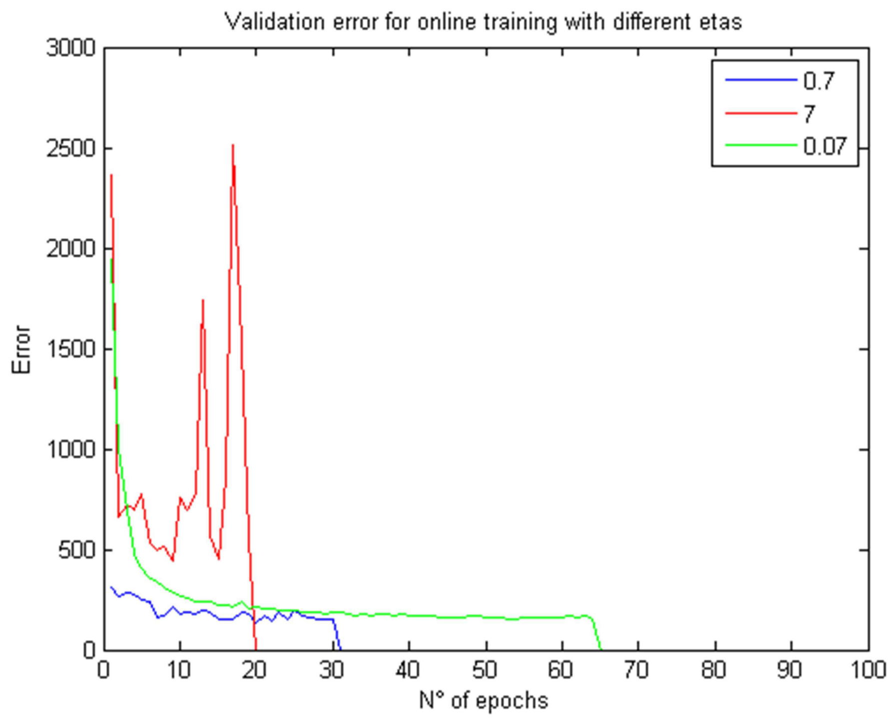

Figure 8. Validation error for online training with different learning rates.

As we can see, the weight changes for an eta of 7 oscillate drastically until they are stopped abruptly. This correlates with the situation described previously. For an eta of 0.07, more epochs are necessary to improve training.

Comparing these three variations, we can see quite different results. With an eta of 0.07, more epochs are needed to train the network. Whereas with an eta of 7, less epochs were done to avoid overfitting. A lower eta reduces standard deviation, while a higher eta increases it.

The graph below shows the training error for batch training with different eta.

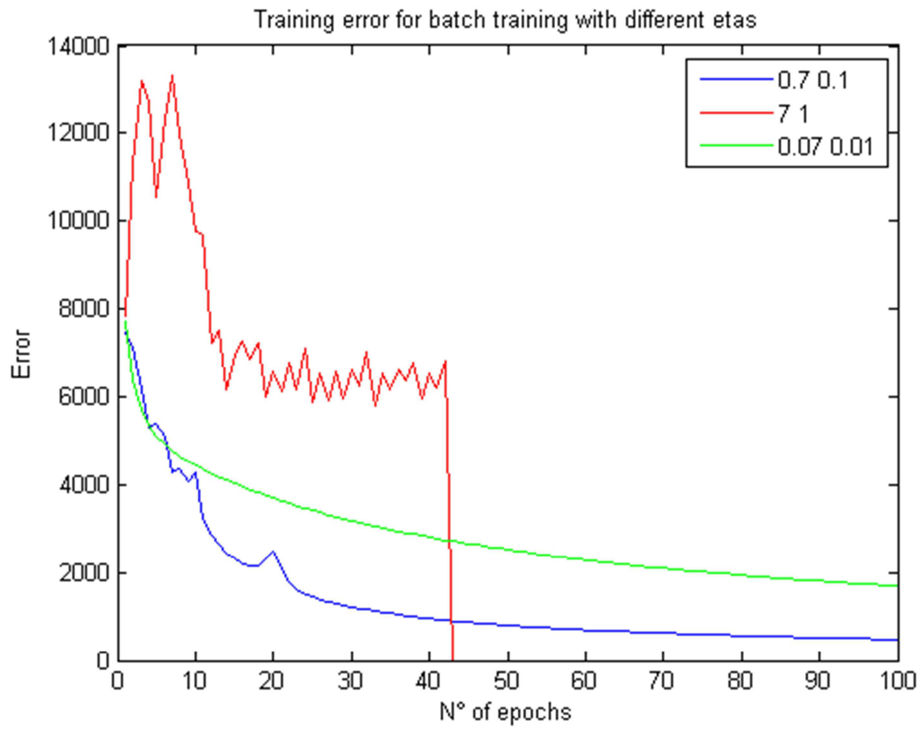

Figure 9. Training error for batch training with different learning rates.

From this graph we can immediately tell that the lower etas need more epochs to train the network. On the other hand, the highest eta overshoots the network giving it a poor performance rate.

The graph below shows the validation error for batch training with different eta.

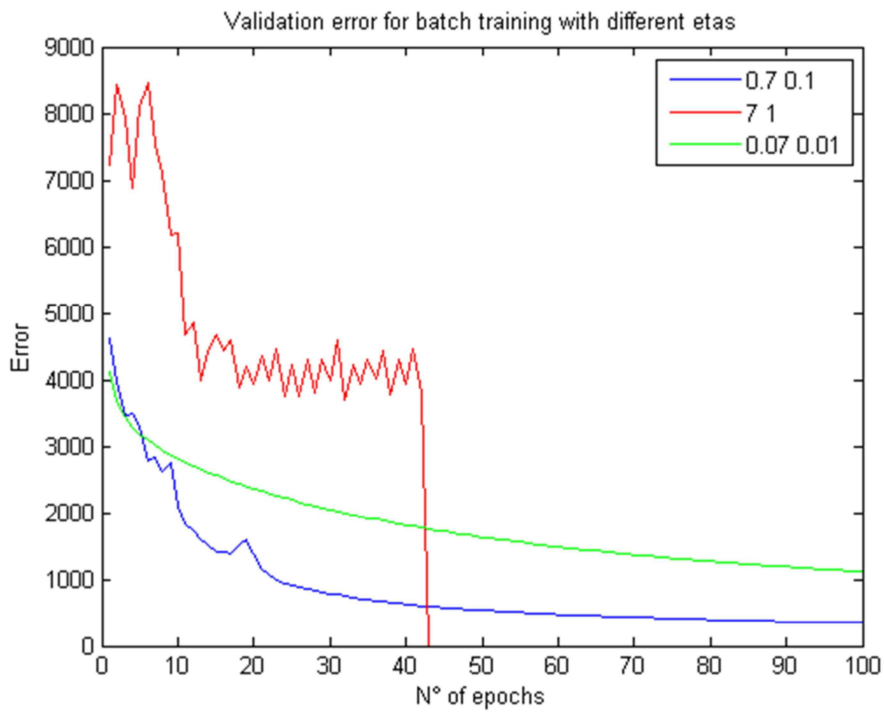

Figure 10. Validation error for batch training with different learning rates.

The same situation described above occurs for the validation error. The lower etas need more epochs to improve the training of the network whereas the highest eta overshoots and stops abruptly.

The comparison of these three variations shows the following results. The best mean performance rate is the one of eta1 0.7 and eta2 0.1. If we reduce both etas, more epochs are needed to reach optimal training. This is explained because more epochs are needed to train the network. If we increase both etas to 7 and 1, respectively, the results are disastrous. This is explained by overshooting, since the sigmoid was saturated before the learning had even started. Reducing and increasing the eta had a negative effect for standard deviation because it increased. This can be explained because the lowest eta needed more training epochs and the highest eta because its validation failed.

Thus, increasing the eta for both online and batch training does not have positive results, especially for batch training.

When changing etas for both trainings, there is a tendency for the mean epoch to drop, especially when trained with a high eta. This can be explained by the abrupt stop provoked by overshooting.

The graph below shows the training error for online training with different neurons in the hidden layer.

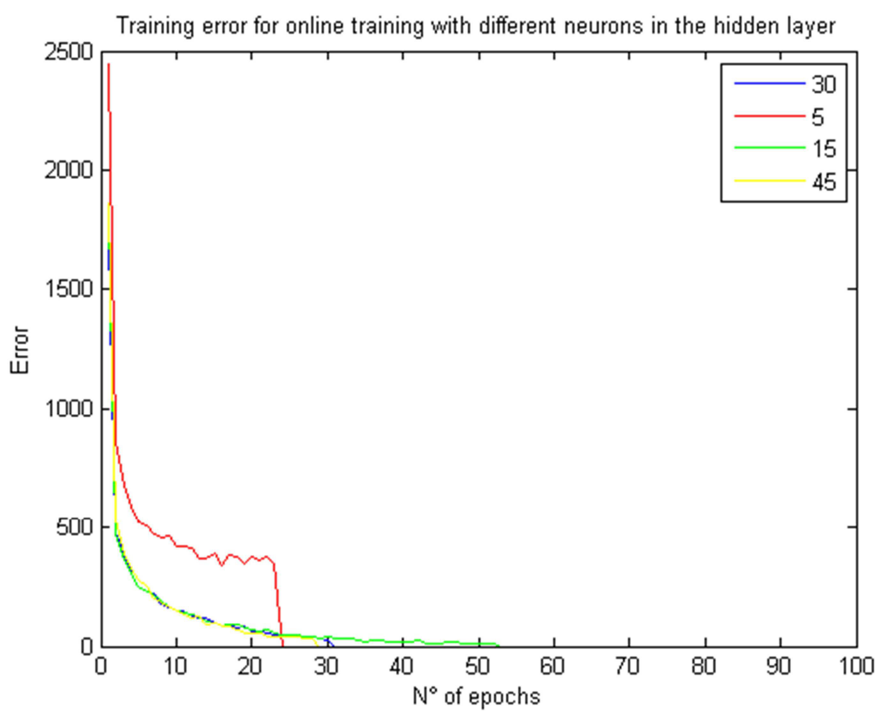

Figure 11. Training error for online training with different amounts of hidden units in the hidden layer.

As we can see, with 5 neurons the training error stops decreasing abruptly because the training stops. With 15 neurons, more epochs are needed to train the network. Finally, with 30 and 45 neurons similar training results are achieved.

The graph below shows the validation error for online training with different neurons.

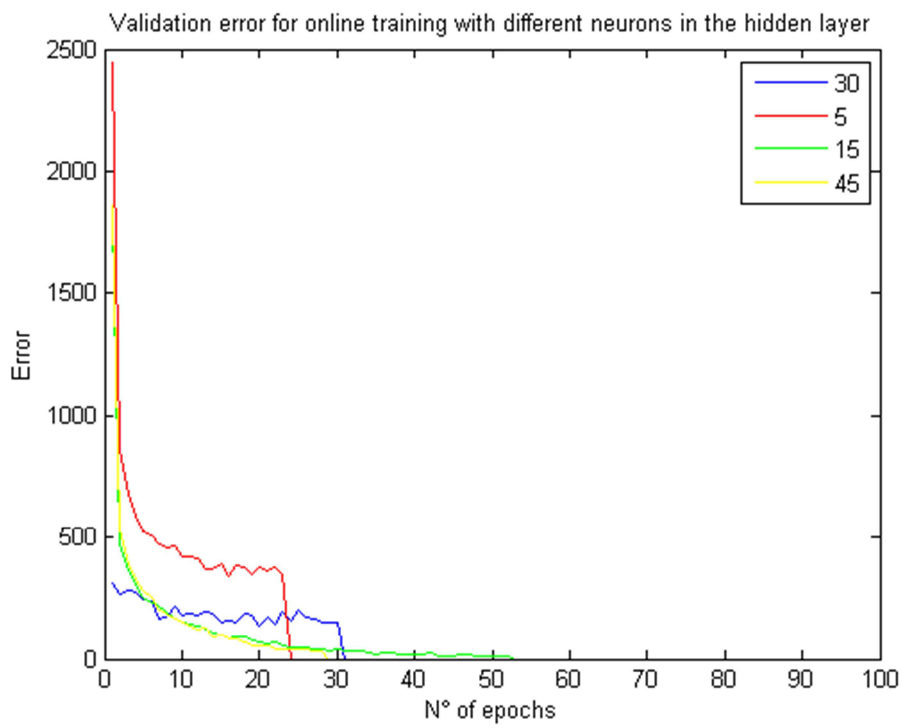

Figure 12. Validation error for online training with different amounts of hidden units in the hidden layer.

On this graph, we can see once again that with 5 neurons, weight changes end abruptly, the validation error increases and the validation fail criterion is met before than in the other configurations. For 15, more epochs are needed to improve training and for 30 and 45 neurons, a similar amount of epochs is needed.

After comparing these four variations, we can see that the mean performance rate and the overall classification rate improve by adding more neurons. Nonetheless, this improvement does not increase steeply and adding more neurons may put extra strain on the computing.

Based on the three experiments, it can be concluded that varying the type of training, the learning rate and the amount of neurons in the hidden layer does have an effect on the performance of the classification of digits.

Online training has a better overall performance than batch training and it also has a better performance when the eta is increased or decreased. The increasing of eta in batch training has terrible results for classification drastically impoverishing its performance rate.

Increasing the amount of neurons on the hidden layer does show an improvement in performance but it is not a sharp increase. It is worth keeping this in mind since increasing the amount of neurons in the hidden layer increases the amount of computations (Heaton, 2008), which can be time consuming, increasing the time needed to train the network.

Interestingly, 0 was the digit that was best classified in all except one of the simulations and 8 was the most poorly classified digit regardless the type of training, the eta and the amount of neurons.

4.1.6 A PHILOSOPHICAL SIDE NOTE OF BOTH IMPLEMENTATIONS.

At a first glance it is quite obvious how different both implementations are. Firstly, both of them are related to distinct fields, the first implementation to psychology and the second one to a task that has practical yet not psychological applications. Secondly, the degree of description of their architectural design and complexity was different as well. The visual search ANN needed few algorithms, parameters and lines of code to perform its task, whereas the optical recognition

ANN needed a considerable amount of these. Finally, how to interpret the results of each ANN has to be considered as well. The results of the optical recognition ANN are interpretable under the light of a real world image that was encoded into a matrix that was then classified into a number. There is little room to interpretation here: if the training process was successful, then the output number should correspond to the image, in this case coded data, used as input. In contrast, the simulation for visual search serves its purpose as long as it is interpreted in a certain way. As such, it is not a real simulation of visual search, it is only a simplification of what would happen in the real world if instead of high and low values we had, let us say, bright and dark colors. Nonetheless, its usefulness to support different theoretical approaches to a phenomenon cannot be discarded.

4.2 A BRIEF ACCOUNT OF THE ACQUISITION OF THE ENGLISH PAST TENSE.

Research concerning the stages children go through to acquire the English past tense has pinpointed three stages of acquisition with continuous and stable transitions between them (Berko, 1958; Brown, 1973; Kuczaj, 1977; Marcus et al., 1992). In the first stage, the production of past tense forms by children is rare. The past tense forms that are produced are the ones that are most frequently found in adult speech (Westermann, 2000) and the majority of them are irregular. It is to be expected that children correctly use the forms ‘came’, ‘knew’, ‘looked’, ‘took’, and ‘went’, not marking past tense for other verbs. Considering the fact that most verbs that children use at this stage are irregular, there is no evidence to describe past tense inflection based on a rule.

In the second stage, which starts at around 29 months of age, children produce an increasing number of past tense forms. New past tense forms are produced, being most of them examples of the regular past tense.

At this stage, indications of a linguistic rule for past tense formation emerge. The first piece of evidence to consider the emergence of a linguistic rule comes from experiments in which children were able to produce the past tense for pseudo-words, as in the case of ‘rick’, in which the children used the past tense ‘ricked’ (Berko, 1958). The second piece of evidence is related to the phenomenon known as overregularization, which starts occurring at this stage. This means that the irregular past tense forms which were produced correctly in the first stage are corrected in order to comply with the regular past tense form. Utterances such as ‘comed’, or ‘camed’ might be produced. These utterances occur simultaneously with the correct past tense form and none of them is predominant.

This stage can last as far as the school age, with a decreasing rate of overregularizations. Highly frequent irregular past tense forms in parental speech are less overregularized that low frequency irregular past tense forms (Marcus et al., 1992)

In the third stage, children produce both regular and irregular forms correctly. Nonetheless, regularizations and overregularizations can occur sometimes. An instance of regularization would be the case of verbs with two past tense forms such as ‘burn’, in which the regular past tense form ‘burned’ and the irregular past tense form ‘burnt’ are both acceptable.

These three developmental stages are referred to as U-shaped learning. This name is given because of the learning curve found in the acquisition of past tense in which irregular forms are produced correctly, then are overregularized and finally are produced correctly once more.

4.3 MODELLING THE ACQUISITION OF THE ENGLISH PAST TENSE.

Most models of English past tense acquisition have focused on this distinct learning curve. This section presents a survey of different models that have aimed to represent U-shaped learning. Namely, the focus will be on two neural networks models that have claimed to represent the experimental data discussed above. The first model has historical importance because it was the first model that claimed to have learnt the English past tense without resorting to implicit rules. The second model is of importance because it was the first one to implement the learning process based on the frequency of words, test them against experimental data beyond U-shaped learning and the dataset used on this model has been used by different ANN models both in psychological and real world applications (Bybee, 1995; Anderson, 2009; Smith, 2011).

4.3.1 RUMELHART AND MCCLELLAND'S MODEL.

In 1986, Rumelhart and McClelland presented a model that, in their opinion, acquired the English past tense based on the psychological data at hand. This meant that they were able to program an ANN that in the process of learning the English past tense showed the same U-shaped learning development as children. This model was composed of a simple pattern associator network similar to Kohonen's map (Section 4.1.2), which learnt the relationship between the base form of the verb and the past tense form (Rumelhart and McClelland, 1986).

Unfortunately, this model had to endure strong criticisms regarding its implementation and its real capability to account for experimental data. The most relevant criticism is related to U-shaped learning. Pinker and Prince (1988) after thoroughly analyzing this model reached the conclusion that U-shaped learning was reached due to the manipulation of input data. The input used to train this ANN was uneven. First, only the ten most frequent verbs were used as input, being eight of them irregular. On the second phase of training, 410 new verbs were added as input. 80% of these verbs were regular. This abrupt increase of regular verbs as inputs explains why this ANN started to overregularize the irregular verbs that learnt on the first stage of training. Considering that this model tried to fit experimental data, it also had to take into account the fact that regular verbs account for around 45-60% of verbs produced in children’s speech. Furthermore, this proportion is stable and does not undergo sudden changes (Marcus et al., 1992).

Also, the generalization capabilities of this model had a rather high error rate for regular verbs. Around 30% of the regular test verbs were produced incorrectly, which is implausible when compared to the experimental data (Pinker and Prince, 1988).

Regardless these criticisms, Rumelhart and McClelland’s model is important because it paved the way for ANNs to be used as means to model experimental data on linguistics settings.

4.3.2 MACWHINNEY AND LEINBACH’S MODEL.

In 1991, MacWhinney and Leinbach advanced on Rumelhart and McClelland’s model and presented an ANN which relied on the backpropagation algorithm to learn the English past tense (Section 4.1.5.2). The advantage of using such algorithm as learning mechanism is that the use of hidden units improved the representations that the ANN could make of input (Hinton, 2007; Huynh and Reggia, 2011). Moreover, MacWhinney and Leinbach used as input a more realistic training corpus which coincided with the actual frequencies of English verbs. This corpus was presented in the training stage according to their frequency of occurrence: the most frequent words were presented at each epoch, while the less frequent were presented at every 700th epoch. By doing so, the abrupt increase of input without a methodological and experimental reason, as in the previous model, was avoided.

This model, during training, produced the past tense for all the regular verbs in the training set, which meant a considerable improvement when compared to Rumerlhart and McClelland's model. However, it failed, first, to learn 9.7% of the irregular verbs in the training set and, second, to account for learning curve present in children's acquisition of the English past tense.

This ANN was tested using 13 untrained irregular verbs, which in Westermann's (2000) opinion is misleading because irregular verbs past tense cannot be predicted. He suggested that this kind of model should be tested against pseudo-words, which also had been tested on humans, and then compare both performances. At the time this model was presented, such pseudo-words and experimental data did not exist. MacWhinney (1993) presented a revised and simplified version of this model which was tested against a corpus of pseudo-words created and tested by Prasada and Pinker (1993).

Prasada and Pinker (1993) created a set of pseudo-words to investigate the inflection of novel words. In the case of pseudo-verbs, they were divided in irregular and regular-like, each composed of three classes: prototypical, intermediate and distant. Pseudo verbs were similar to existing verbs in different degrees, being prototypical the most similar and distant the least similar. The experimental data gathered after presenting the pseudo verbs to human subjects showed, on the one hand, that there is a tendency to inflect pseudo-regulars as regular, which is independent of their similarity to existing regulars. On the other hand, the tendency to inflect pseudo-irregular verbs as irregular is reduced as the similarity with existing irregular verbs decreases.

In order to test the generalization skills of acquisition models, the inflection of distant regular-like pseudo verbs has to be tested. In this case, a regular inflection has to be produced regardless the similarity to existing verbs. This would prove that the regular case has been learnt as the default inflection of novel cases.

The revised model performance was deemed as humanlike for its capacity to produce regular forms for distant pseudo-regulars (MacWhinney, 1993). Unfortunately, these results are based on an optimistic projection of the results that the network was producing at 4,200 epochs, although it was trained for 24,000 epochs. Furthermore, the revised model produced 90% of the irregular verbs in the training stage. Ideally, the model should be able to produce all the irregular verbs in the training set correcting and at the same time be able to produce the regular inflection for novel words.

4.4 THE INNATENESS OF LANGUAGE.

Chomsky on his critique of Skinner's Verbal Behaviour (1959) argued that conditioning was not the appropriate approach to explain language acquisition. First, acquiring a language does not depend entirely on environmental phenomena. Second, conditioning would not be able to produce all the possible expressions of a person's linguistic behaviour because that would mean that a person should be exposed to and trained in a vast if not infinite number of sentences to be able to use words and then sentences properly. Such situation is highly unlikely because of the input children receive and because we can produce and understand new utterances almost effortlessly. This ability of being able to acquire a language regardless the poverty of the stimulus humans are exposed to is central to the notion that language is innate. For Chomsky, mastery of a language involves knowing its grammar.

On Aspects of the Theory of Syntax (1965), Chomsky argues that the data children are exposed is highly impoverished, first, because they compose only a small sample of the infinite number of sentences that can be produced by natural language and, second, because they do not comprise the ill-formed sentences needed to create prototypes of ungrammatical sentences. Chomsky suggests that the Universal Grammar (UG) supplies constraints that help children avoid the production of ill-formed sentences and allow native speakers to recognize grammatical sentences from ungrammatical ones. As such, humans are born with a built-in language acquisition device (LAD) which prevents children from wandering around an infinite number of grammars before reaching the specific grammar of their language.

Putnam (1967) presents a set of empirical and alleged facts that support the notion of innateness in language. First, children learn their native language with remarkable ease in a short period of time. Being exposed to a language is all they need to acquire a language and no explicit instruction is needed. Second, as Chomsky (1959) noted on his critique of Skinner, there is no need for reinforcement to learn a language. Third, language acquisition does not depend on the IQ of children. Fourth, the existence of linguistic universals, common features to all languages, is a consequence of innateness and UG. Finally, and this would count as a pseudo argument, how could we account for a task as difficult as language acquisition without the help of an innate component?

Further arguments for the innateness of language are the following. First, the existence of Broca's and Wernicke's areas in which language is produced and understood respectively. The specialization of both areas and the language impairment due to damage to those areas (Radanovic and Mansur, 2011) would lead us to think that the language faculty or the LAD is located around these areas. Chomsky (2000) prefers to remain sceptical about these statements and consider that more investigation is needed to assert that the language faculty is pinpointed in Broca's or Wernicke's areas or in a different area. If conclusive evidence for a specialized area of the brain exclusively dedicated to language appears, the area could be deemed as the organ of language (Chomsky, 2000). Although Chomsky's scepticism, the existence, evolution and specialization of such areas highlight the importance of language as a human skill and may serve as a counterargument for people who consider language a by-product or emergent behaviour of neural activity.

Second, there is evidence of genetically determined systems in mammals, such as the visual system that require external appropriate stimuli to be developed and prevent deterioration. An example of this would be the visual system of cats (Chomsky, 2000). If kittens do not receive certain light stimulus, the structures in the striate cortex start to deteriorate. Moreover, if they receive one type of visual stimulus, only vertical or horizontal lines, the cells in the striate cortex would distribute in different manners depending on the stimulus. Therefore, there are systems that are innate but need stimulation to function. This would be the case of language acquisition: all of us are born with the innate skill to acquire a language without instruction but we need to be exposed to it. The time frame for exposure is not an idiosyncratic matter: language acquisition can occur up to puberty in which our acquisition skills deteriorate. This is known as the critical period of acquisition.

Third, the innateness of language can be built specifically or as a combination of different aspects of the brain. An example of a property specific to language which is not found elsewhere in the natural world is the property of discrete infinity. This property presupposes that we can have sentences with an infinite and not determined number of complete words, this means, we have sentences of six or seven words but not sentences with six and a half words (Chomsky, 2000). Nonetheless, it may be the case that a language property is built on other systems which have a different primary function. An example of this would be to ask if our tongue and teeth evolved in such way so that we could produce language or if it evolved first to eat certain foods and then the range of sounds that we could produce adapted to the existent architecture of the human body.

Chomsky et al. (2002) revisited the idea of innateness in language and contrasted it to two approaches to relate it to the brain. The first approach is related to emergentism and posits the idea that language, as mental things, is an emergent property of the brain. These emergent properties are produced by interactions between lower level events, synapses or neural networks. Unfortunately, the interactions have not been understood yet. Chomsky et al. (2002) consider that language is not an emergent property of the brain, and that once the right physical properties, the right synapses or neural networks, are found, the need to rely on emergent properties with obscure principles would disappear.

The second approach presents four perspectives to study animal and human language. These four perspectives are: mechanistic, search the mechanisms, psychological and/or physiological, that implement language; ontogenetic, find the genetic and environmental factors related to language; functional, find the evolutionary advantages that language presents for humans; and phylogenetic, find the evolutionary history of humans as species and compare language to different past features. Chomsky et al. (2002) consider that Hauser (1996) presents this approach clearly and evaluate his arguments. Regarding the first perspective, Hauser does not cover mechanisms at all, being the same case for the functional perspective. Furthermore, and this relates to the first two perspectives, Hauser consider that studying the psychological and physiological aspects of language are irrelevant to the formal study of language. Therefore, the first two perspectives are abandoned for human language. Language as an adaptive function is not consider for functions such as mating, survival and so on because Hauser regards language not as a system of communication but as a means to express thought.

Finally, the third approach is related to innateness and involves the notion that there are specialized mechanisms in the brain to learn in specific ways. In this sense, for language acquisition to happen, the existence of a language organ is needed. Chomsky et al. (2002) consider that this approach is sound and further argue that even behaviourists assume the existence of an innate mechanism to distinguish linguistic material from the rest of stimuli.

All in all, Chomsky is reluctant to consider that language is not innate, a by-product of our evolution or an emergent property of lower level phenomena. His views are challenged by what is known as connectionism and emergentism and the following sections present the arguments of these two approaches to learning and, particularly, language acquisition.

4.5 CONNECTIONISM AND EMERGENTISM.

Connectionism is a philosophical theory which tries to explain some of the mental process which occur in the brain by means of interconnected neurons or networks of neurons. Fodor and Pylyshyn (1988) in a critical analysis of connectionism stated that a natural or artificial system can evolve and self-organize itself without having to resort to memorization. Connectionist models are seen as "networks consisting of very large numbers of simple but highly interconnected "units" (1988: 4).

The most common way to represent these networks and their underlying processes are through ANNs. The connections of the artificial neurons resemble synapse. Synapse is a junction by which electrical or chemical signals are passed from one neuron to another neuron or cell. For ANNs to process an input, an activation threshold has to be reached which triggers the connections in the artificial neurons producing an output. Programmers train ANN to process the input to obtain certain outputs. The training may differ depending on the algorithm and the mathematical instructions that are used to solve a problem. The main idea of supervised training is that exemplar inputs, typical or best examples of a phenomenon, are introduced into the ANN and then exemplar outputs are introduced as well. This way, the ANN recognizes a pattern that is reinforced. As Bates et al (1998) described it:

> "Networks would be exposed to examples of a target behavior (for example, the appropriate responses to a set of varied stimuli). Through learning, the network would learn to adjust the weights in small incremental steps in such a way that over time, the network's response accuracy would improve." (1998: 5-6)

As seen in section 4.1.5, connectionist models may vary on the algorithm they use as an activation function, the number of input and output nodes, the nature of the input (inhibitory or excitatory) and the connections among them. Nonetheless, they remain analogous to simple neurons. Elman (2001) notices that there are three key features in the way ANNs operate. First, the activation function is nonlinear. This means that ANNs have a probabilistic nature in which the units may fire under certain conditions whereas in other they would remain in their resting state. Second, the knowledge of the network is represented in the connections between units and the weights the connections have. Third, there are no symbolic representations. The representation of stimuli presented as input would depend on the pattern of activation between units. This means that a word can be represented by only one unit or a complete set of them.

He also notices that at first the weight of the connections was changed manually, but now there are different algorithms to solve different tasks. This means that the weights needed for a specific ANN could be self-programmed. Moreover, learning is carried out by induction, as stated above, in which exemplar inputs and outputs are presented to the network.

Connectionism presents itself as an alternative to innatist symbolic models that completely rely on knowledge hardwired in the brain. The main reasoning behind this proposal is that innateness is confused with domain specificity, species specificity, localization and learnability (Bates et al. 1998). The first claim is that language is so particular, when compared to other abilities that we share with different animals, that it must be innate. The second claim is that we are the only species that has this particular kind of language, so it must be part of our genes. The third claim assumes that since particular parts of brain, such as Broca's and Wernicke's area, are involved in language, then it must be innate. Finally, the fourth claim states that learning a language is so difficult and nonetheless children are able to do it easily without instruction, then again, language must be innate.

Bates et al. (1998) breakdown each of these claims to prove that in some cases a nativist and a connectionist/emergentist explanation could produce the same result. First, on the subject of domain specificity, they consider four levels related to it: behavioral, representational, mental/neural processes and genetic. Regarding behavioral specificity, language is different from other cognitive systems because of the type of task it has to solve: "mapping a hyperdimensional meaning space onto a low-dimensional channel" (MacWhinney and Bates, 1989). This meaning space includes experiences shared by all humans and human language is constrained by information processing, such as memory. If we consider such factors, the possible solutions to this problem are limited. Therefore, behavioral specificity can serve as an argument for UG and innateness, but also for emergent properties. Concerning representational specificity, all knowledge representations are stored in our brain whether they are innate or not. Thus, representational specificity does not help any of the two positions. Regarding the specificity of mental/neural processes, the question arises whether language can be learned by a system which is not designed for learning it. This question has not been settled but evidence from plastic reorganization in children with focal or left hemisphere brain injuries shows that in the absence of the areas commonly involved in language acquisition, the brain is able to reorganize and learn it as normal children do (Stiles, 2000; Jonhston, 2009). Concerning genetic specificity, genetic disorders as specific language impairment (SLI) are used as an argument for the innateness of language. However, research on SLI shows that it is often accompanied by other disorders that could explain the impairment of language. As such, domain specificity does not serve as a strong argument for the innateness of language.

Second, on the subject of species specificity, there are more domains which are specific to humans, such as basketball and online dating, but that does not mean we are hardwired for such domains. Furthermore, we share neural circuitry with species very different from ours, such as rats, which are disposed before becoming functional. Therefore, the fact we are the only species that has skill does not mean it is innate.

Third, on the subject of localization, all knowledge, innate or learned, assumes that it is localized in the brain. It does not matter if the localization is universal or variable. This is proven by arguments as brain plasticity named above in which the brain reorganizes itself when there are focal lesions in children.

Fourth, on the subject of learnability, it is quite remarkable how children are able to learn a language when the data they receive might be incomplete, it does not cover the whole range of linguistic possibilities and lacks negative examples. This serves as a proof that language acquisition is innate otherwise this task could not be carried out. Nonetheless, simulations have been able to learn grammars without resorting to hardwired symbolic knowledge and only relying in a learning algorithm and different sets of conditions such as increasing memory at certain periods of time to learn complex structures.

O'Grady (2008) states an interesting remark: "the purpose of emergentism is not to refute nativism; it is to devise a better version of the innateness hypothesis." (O'Grady, 2008:630). This is true because connectionism and emergentism rely on innate structures such as neural networks to posit their theories. The difference is they argue for emergent properties of the interaction between different neural networks, and the environment. Therefore, they are not searching for the specific language organ; rather they are working on finding the interactions of low level events that can produce emergent properties such as language.

5. METHODOLOGY.

In this section the methodology to simulate the acquisition of simple present, and regular and irregular past will be described. First, a description of the datasets used in each simulation is provided. Second, a succinct description of the architecture of each network is given. Finally, the experiments to test the generalization skills of each network are described.

5.1 DATASET.

The dataset used for these simulations were provided by Professor MacWhinney in a private communication. According to the information provided in the readme files, this dataset is the same he and Leinbach used in 1991. Therefore, the description of the dataset is provided according to their 1991 article. In case the dataset differs, this will be stated.

MacWhinney's and Leinbach's dataset (1991) was built based on Francis and Kucera (1982) corpus of English. This corpus included the frequency of English words. The base set MacWhinney and Leinbach used included the 6949 most frequent verb forms, which included present, third person singular, present participle, past and past participle of both regular and irregular verbs. These forms were inflected from 2161 verbs. Taking into consideration the problems Rumelhart and McClelland (1986) encountered when simulating homophones and multiple forms, such as the past tense forms 'dreamed' and 'dreamt', MacWhinney and Leinbach eliminated the less common forms. Furthermore, to reduce the stress on the network, "all the forms that had more than three syllables, consonantal phonemes in a row, or more than two vocalic phonemes in a row were also removed" (MacWhinney and Leinbach, 1991). After these modifications to the original base set, 6090 forms, inflected from 2062 verbs were used as corpus for simulation. This corpus was divided once more to create a testing corpus. This testing corpus was formed of the least frequent 10% of the regular verbs, and the least frequent 10% of the irregular verbs. The training corpus was formed of 5481 forms. The training set comprised 118 irregular past tense forms, which is not the complete set of irregular verbs in English. Nonetheless, these are the most frequent irregular verbs. The list of irregular verbs is in Appendix 2.

The dataset that Professor MacWhinney provided me included 5772 verbs forms in the training set, 614 regular forms and 27 irregular forms, and 60 pseudo words for testing. The training set is in Appendix 3, the regular form testing set is in Appendix 4, the irregular form testing set is in Appendix 5, and the pseudo words testing set is in Appendix 6. These three sets were used without any modifications to test the acquisition and classification potential of eight verb forms: base form, present participle, third person singular, regular past, irregular past, regular past participle, irregular past participle and irregular third person singular. The last form had only one instance that appeared in the training set. It is the word ‘says’ and no reason is provided for such classification.

These datasets were modified to test the acquisition and classification potential of four and three verb forms. In the case of four form classification, the focuses were the base form, the third person singular, regular past and irregular past. Therefore, the present participle, the regular past participle and the irregular past participle were eliminated from the original dataset. The rationale behind this is that reducing the number of grammatical structures to be learned could reduce the amount of computations and the stress of the task. After the modifications were carried out, the training set was conformed of 3283 verb forms, the regular testing set was conformed of 478 verb forms, and the irregular testing set was conformed 12 verb forms.

In the case of three form classification, the training and testing sets were conformed of the same amount of verbs as the training and testing sets for four form classification. The only difference between these two forms of classification is the output produced. For three form classification, the base form and the third person singular are treated as present. This means that every time the network is provided with a base form or the third person singular as input, it will produce, ideally, the same code as output. For four form classification, the base form and the third person singular are treated as distinct instances and are produced as output with different codes.

5.1.1 CODING OF THE INPUT.

The training files provided by Professor MacWhinney are coded in the following manner. First, 433 numbers represent the coded form of the input. These numbers are instances of 1 or 9. Then, the written form of the verb form is provided followed by a letter and a binary number. The letters represented the verb form of the written form and the binary number if the form was regular (0) or irregular (1). The letter 'b' represents the base form; the letter 'z', third person singular; 'g', the present participle; 'd', the past; and 'n', the past participle. The following is an example of the coding of the verb 'upset':

11111111111111111111111111111111111111111111111111111111111111111111111111
11111111191119111111111111191111111111911111911111111191111991111111111111
11111111119111111111111111111111111111111111111111111191111111111
11111111111111111111111111111111111111111111111111111111111111111111111111
11191119111111111111191111111111911111911111111191111991111111111111111111
11119111111111111111111111111111111111111111111111111 upset-b-0

Figure 1. Coding for the base form of the verb 'upset'.

5.1.2 CODING OF THE OUTPUT.

The coding of the output varied depending on the amount of classes to be classified. For the eight class classification task, each output was composed of eight binary numbers. The network used for this task forced the output vector to be only activated in one element. Therefore, only one element of the vector could be 1 whereas the rest of the elements had to be 0. For example, the verb form 'upset' is the base form of the verb and it would be coded as 1,0,0,0,0,0,0,0 for the eight class classification task; 1,0,0,0 for four classes; and 1,0,0 for three classes. The tables below provide the coding of the output depending on the class they classify.

Table 1. Output coding for eight classes

Class	Ouput Code
Base form (-d-0)	1,0,0,0,0,0,0,0
Present participle (-g-0)	0,1,0,0,0,0,0,0
Third person singular (-z-0)	0,0,1,0,0,0,0,0
Regular past participle (-n-0)	0,0,0,1,0,0,0,0
Regular past (-d-0)	0,0,0,0,1,0,0,0
Irregular past (-d-1)	0,0,0,0,0,1,0,0
Irregular past participle (-n-1)	0,0,0,0,0,0,1,0
Irregular third person singular (-z-1)	0,0,0,0,0,0,0,1

Table 2. Output coding for four classes

Class	Ouput Code
Base form (-d-0)	1,0,0,0
Third person singular (-z-0)	0,1,0,0
Regular past (-d-0)	0,0,1,0
Irregular past (-d-1)	0,0,0,1

Table 3. Output coding for three classes

Class	Ouput Code
Present (-d-0) (-z-0)	1,0,0
Regular past (-d-0)	0,1,0
Irregular past (-d-1)	0,0,1

5.2 ARCHITECTURE OF THE NETWORKS.

5.2.1 ARCHITECTURE OF THE BACK PROPAGATION NETWORK.

In order to find the best architecture for the back propagation network that would carry the supervised learning classification task, different configurations were tested. First, a multi-layered network with one hidden layer was built. It had 433 input units in the input layer, 200 hidden units in the hidden layer, and the output units in the output layer varied according to the classes: 3, 4 or 8.

MacWhinney and Leinbach (1991) used a similar architecture, with two hidden layers and 200 units in each of them.

This network was built using the Neural Network toolbox of Matlab. This toolbox is a package of built-in functions and applications used for modeling complex nonlinear systems. It can be started by typing ‘nnstart’ on the command window in the workspace of Matlab. A new window opens in which we have options for the task we want to accomplish. Once the task has been selected, in the case of pattern recognition, the input and output datasets for training have to be uploaded. In the case of clustering, only the input dataset for training has to be uploaded. Next, the input is divided into training, validation and testing set. The percentages for each set can be changed directly here. Then, the amount of hidden neurons in the hidden layer is defined. When using this toolbox from ‘nnstart’, it is not possible to add more hidden layers. Nonetheless, this can be done manually. Finally, the network can be trained and plots related to the efficiency of the network are created.

The results for this network were below the expectations for each of the classification tasks (Pinker and Prince, 1988; Westermann, 2000). For the eight class classification task, this first network was able to train correctly but had an 8% of classification error when it was tested with the regular test set. For the three class classification task and the four class classification task, the network performed perfectly in training and testing.

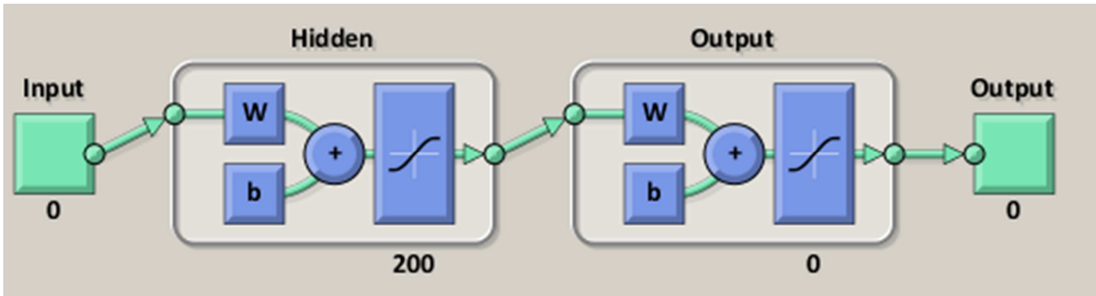

Figure 2. Representation of the architecture of the back propagation network with one hidden layer.

Considering that the performance of the network is expected to be 100% in training and regular forms testing, a second hidden layer was added in an attempt to improve the classification potential of this network.

MacWhinney and Leinbach (1991) implemented a two-hidden layer back propagation network with 200 hidden units in each hidden layer. Considering that his model was built more than 20 years ago using a different simulator (PlaNet, which was not available online), the decision was made to start with less units in the hidden layers. The first tests were done using 25 units in each hidden layer achieving results below the expectations in both training and testing. The amount of units was doubled, 50 units in each hidden layer, and that produced the expected results.

This pattern recognition network was initialized using the function 'patternnet'. In this case, the following was typed into the command window:

```
Net=patternnet([50 50]);
```

By typing this, a pattern recognition net is initialized. The numbers inside the square brackets define the amount of hidden units and the number of hidden layers. In this case, this network has 50 hidden units and two hidden layers. Then, the input and output are uploaded using the function 'dlmread', which imports data from text into a matrix. Next, the network was trained using the function ‘train’, as follows:

Net=train(net, input’,output’);

This produces a window similar to the one in the toolbox in which the plots can be visualized.

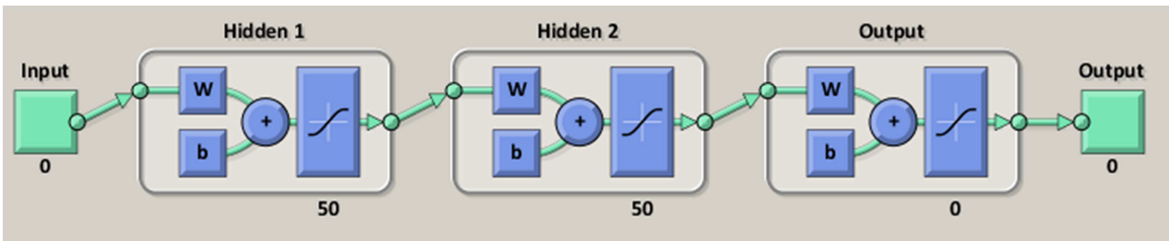

Figure 3. Representation of the architecture of the back propagation network with two hidden layers.

The codes to run the four simulations related to the back propagation network are in Appendix 7.

5.2.2 ARCHITECTURE OF THE SELF-ORGANIZING MAP (SOM) NETWORK.

A SOM comprises a competitive layer, similar to the one presented in section 4.1.5.1. This type of network can classify multidimensional vectors based on the number of neurons in the competitive layer (Trappenberg, 2010). The output of this network is an approximate two dimensional topology of the classes present in the data set. According to the Mathworks webpage, the SOM network has “a preference (but not a guarantee) of assigning the same number of instances to each class” (Mathworks, 2013).

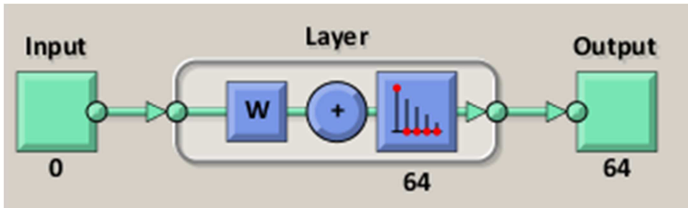

Figure 4. The Architecture of a SOM network.

The code to run the SOM network is in Appendix 8.

5.3 THE EXPERIMENTS.

The experiments in this section are related to the back-propagation network and not to the SOM network. The reason for this is that the SOM network cannot be tested on the same parameters as a back propagation network (Gurney, 1997; Haykin, 1999). SOMs are networks that group data based on their similarity and, as such, they can give an idea on how the different verb forms are grouped but these results cannot be compared with expected output and therefore may not represent the distribution of inputs. Nonetheless, plots related to the SOM network are provided to visualize the clustering it achieved.

5.3.1 FIRST EXPERIMENT: TRAINING OF THE NETWORK.

The training of the three different back propagation networks is the first experiment. The networks have to be able to correctly classify the verb forms given in training. A 100% of accuracy is a must according to Pinker and Prince (1988). The network stops training once generalization stops improving based on the validation checks performed with the validation set or if a certain threshold is reached (Gurney, 1997; Haykin, 1999). The fact that the network has stopped training and reduced the distance between the expected output and the produced output does not entail that, first, it can classify correctly the training, validation and testing set, and, second, that it will generalize well to new inputs. Therefore, the confusion plot produced once training has stopped is a visualization of the accuracy of the network.

Figure 5. Example of a confusion plot.

5.3.2 SECOND EXPERIMENT: U-SHAPED LEARNING

ANNs have to be able to replicate the experimental data in order to be of any interest as a psychology modeling tool. The classification task can be accomplished by a Python, a computer language, program in which the programmer explicitly states the rule for regular past in English. This would certainly serve as an argument for innateness and symbolic rules but it would completely defeat the purpose of connectionist models. Therefore, connectionist models have to replicate the u-shaped learning discussed in section 4.2. The training state plot produced after the network has stopped training can serve as a measure of the learning curve of the network. This plot shows how the error, the difference between the desired output and the actual output, is reduced along a gradient. In some instances the error can increase as in the case of overregularization.

5.3.3 THIRD EXPERIMENT: GENERALIZATION OF NEW INPUT.

Pinker and Prince (1988) criticized Rumelhart's and McClelland's model (1986) because it was not able to generalize instances of regular and irregular verbs that were not introduced as input for training. They state that a model of past tense acquisition has to be able to at least generalize the regular verbs as regular verbs. Therefore, after the training of the network stopped, regular and irregular past tense forms were introduced as input to run some extra tests on the classification potential of the network.

5.3.4 FOURTH EXPERIMENT: GENERATIVE SIMPLIFIED MODEL.

The task performed by MacWhinney's and Leinbach's network (1991) cannot be compared, in terms of difficulty, to the task performed here. They codified phonemes and expected their network to produce different forms based on the base form. The network presented above classified codified words into classes. These are two completely different tasks with a different degree of difficulty. Therefore, a fourth experiment was set up. In this experiment, only base forms were presented to the network. The goal was to produce as output a binary code that stated if the verb used as input would have a regular past tense form [1,0] or an irregular past tense form [0,1]. To test the generalization potential of this network, two extra tests were implemented. First, a dataset composed of regular base forms was provided and the network was expected to produce [1,0] as output for each base form. Second, a set of pseudo words from the Prasada and Pinker corpus (1993) was presented as input to the network. This corpus was composed of pseudo base forms, both regular and irregular, and the goal was for the network to produce the correct output depending on the input.

1308 base forms with a regular past tense form and 148 base forms with an irregular past tense form were used as input for training. For the regular past tense form test, 336 base forms were used as input. For the pseudo words test, 60 pseudo words were used as input.

This is a back propagation network with five hidden layers and 200 hidden units in each layer.

This experiment is conceptually similar to the one devised by MacWhinney and Leinbach. Nonetheless, it is important to keep in mind that the network studied in this dissertation is not a generative model that produces aural output or provides the past tense form in a graphic manner. This is a simplified model that provides a theoretical approach to language acquisition.

6. RESULTS.

In this section the results of each experiment are presented. They are presented following the order of the experiments stated above.

6.1 RESULTS OF THE FIRST EXPERIMENT: TRAINING OF THE NETWORK.

As stated on the previous section, the confusion plot presents itself as a good manner to measure the effectiveness of training based on percentages. Therefore, for each classification task, the confusion plot and its results are presented.

6.1.1 EIGHT CLASS CLASSIFICATION TASK.

Figure 18 presents the confusion plot for the eight class classification task after training finished.

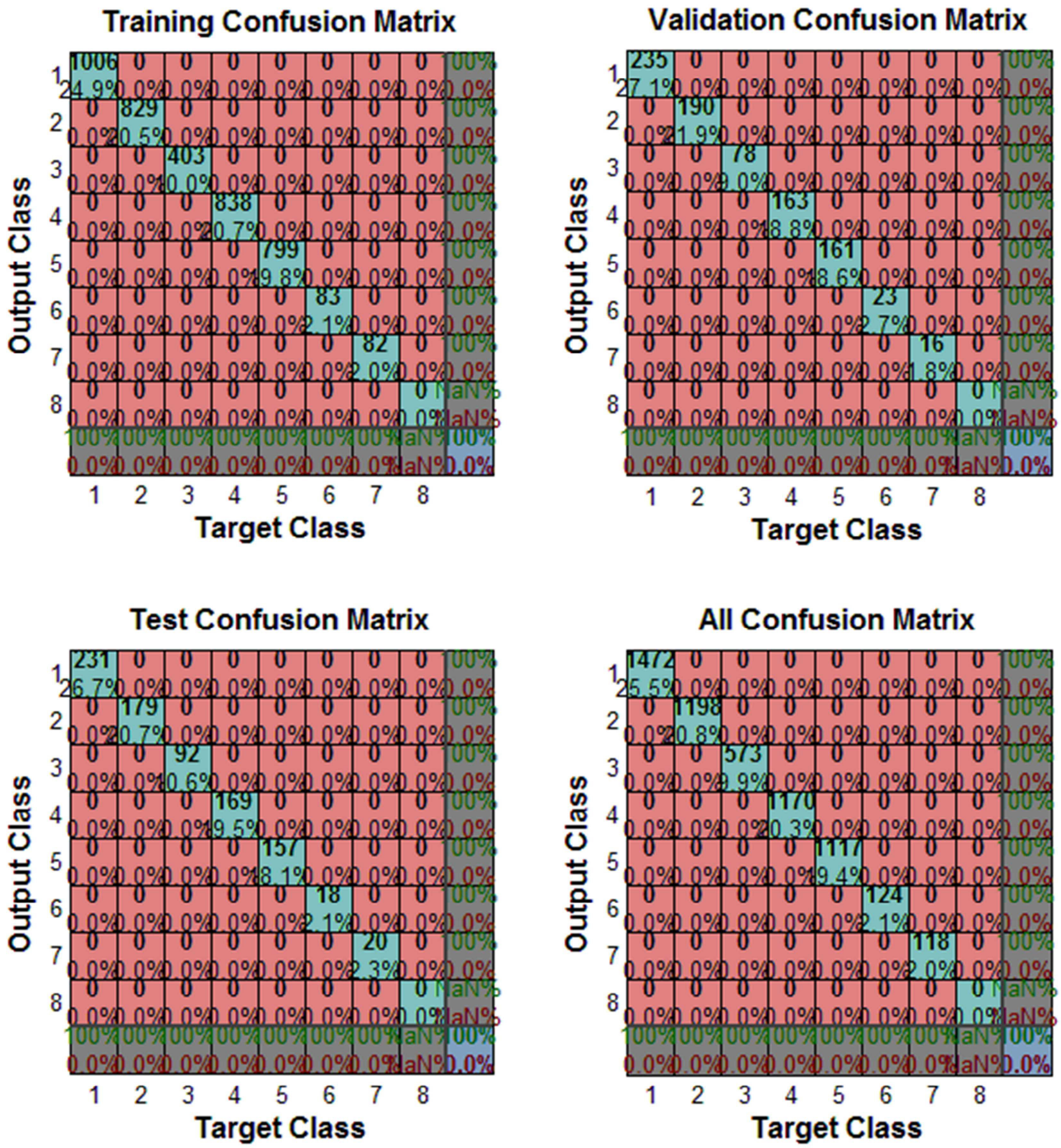

Figure 1. Confusion plot for eight class classification.

In this plot, the green percentage represents the percentage of correct classifications and the red percentage the percentage of incorrect classifications. Because there are eight classes, the confusion matrices are crowded and not all the numbers are easily perceived. Nonetheless, what is relevant about this plot is the blue percentage presented at the lower right corner of each confusion matrix.

In this particular plot, it can be seen that in the training, validation and testing of this network, the classification was a 100% correct.

According to Pinker and Price (1988), these results are a prerequisite to even consider that a connectionist model can be an accurate model of language acquisition.

6.1.2 FOUR CLASS CLASSIFICATION TASK.

Figure 19 presents the confusion plot for the four class classification task after training finished.

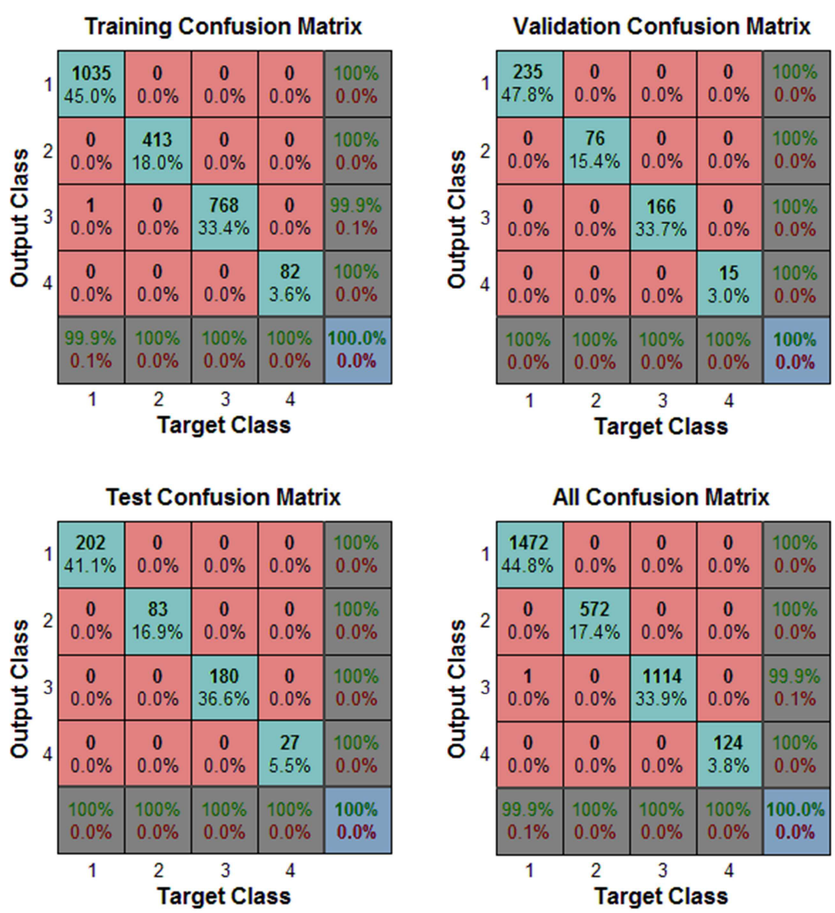

Figure 2. Confusion plot for four class classification.

In this particular plot, all the percentages and amount of instances correctly and incorrectly classified can be easily seen. As an overall consideration, the network classified all of the inputs correctly during training. Although there is one instance which was misclassified, this does not affect the overall percentage of the network.

6.1.3 THREE CLASS CLASSIFICATION TASK.

Figure 20 presents the confusion plot for the three class classification task after training finished.

Figure 3. Confusion plot for three class classification.

As in the case of four class classification, only one instance was misclassified but this did not affect the overall score of this network, which was a 100% of correct classifications. If it had been the case that the number of misclassifications were larger, and therefore relevant, it would have been interesting to see at which stage the misclassification occurred and between which classes. In both cases, both misclassifications occurred at the training stage and both of them were classified as present when they should have been classified as regular past.

In conclusion, the three types of classification complied with the prerequisite imposed by Pinker and Prince (1988) that stated that as a first stage, the network had to classify all of the inputs correctly.

6.2 RESULTS OF THE SECOND EXPERIMENT: U-SHAPED LEARNING.

In this section the training state plots and their results for each classification task are presented.

6.2.1 EIGHT CLASS CLASSIFICATION TASK.

Figure 21 presents the training state plot for the eight class classification task after training had stopped.

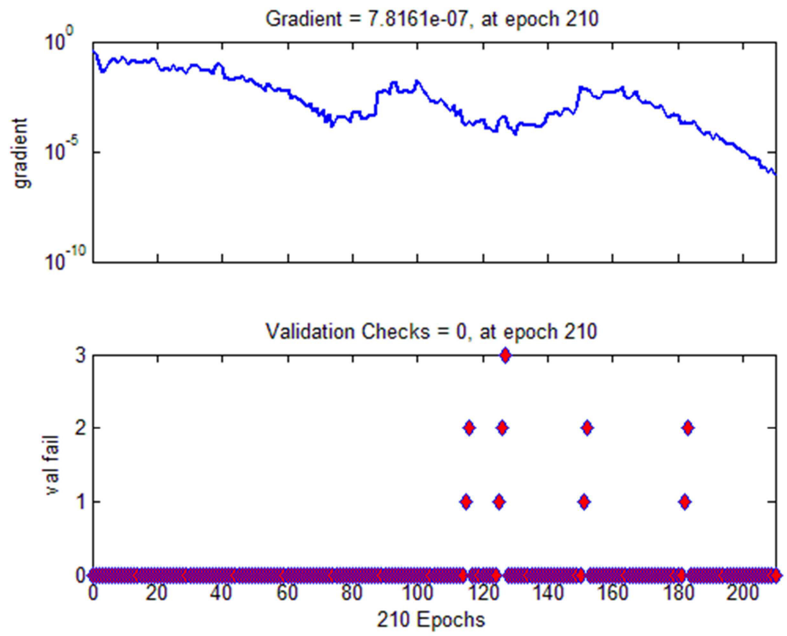

Figure 4. Training state plot for the eight class classification task.

There are two graphs in this figure. The first one shows how the error decreased along the gradient and the second one shows the validation checks, which check if any new improvements can be done to the training and when to stop training. For the purposes of U-shaped learning discussion, the first graph is very relevant. It shows how the error decreases or increases as training goes by. Ideally, the error should only decrease but, as the connection weights are shifting to find the best possible configuration, error may increase. In the graph we can see that the error began at roughly 10^{0} and by the 210th epoch, it has decreased to roughly 10^{-6}. To reach that point the error decreased and increased in small steps until there was no new improvement to make. This shows that at an n epoch, the network was doing better than at an n+1 epoch, showing that mistakes can be done although at a previous stage that mistake was non-existent.

6.2.2 FOUR CLASS CLASSIFICATION TASK.

Figure 22 presents the training state plot for the four class classification task after training had stopped.

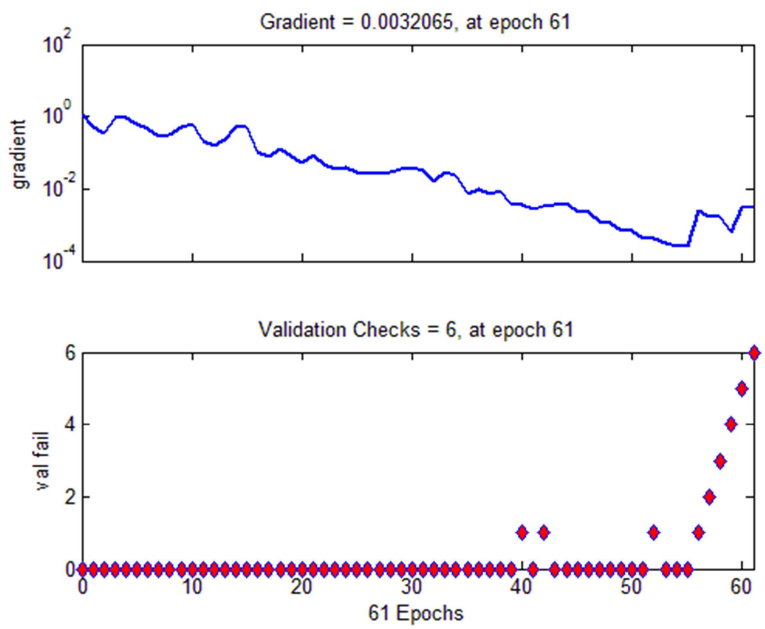

Figure 5. Training state plot for the four class classification task.

In this graph, the error is decreased in the same manner as described above, this means that to reach the lowest error possible, the connection weights had to be rearranged to different values and this made the error decrease and increase at different epochs. This resembles the stages at which children go through to acquire past tense as seen in section 4.2. In this case the error started at 10^{0} and it was decreased to roughly 10^{-2} at the 61st epoch. For the four class classification task, the training stopped because there were six validation checks, which means that the error increased repeatedly instead of decreasing.

6.2.3 THREE CLASS CLASSIFICATION TASK.

Figure 23 presents the training state plot for the three class classification task after training had stopped.

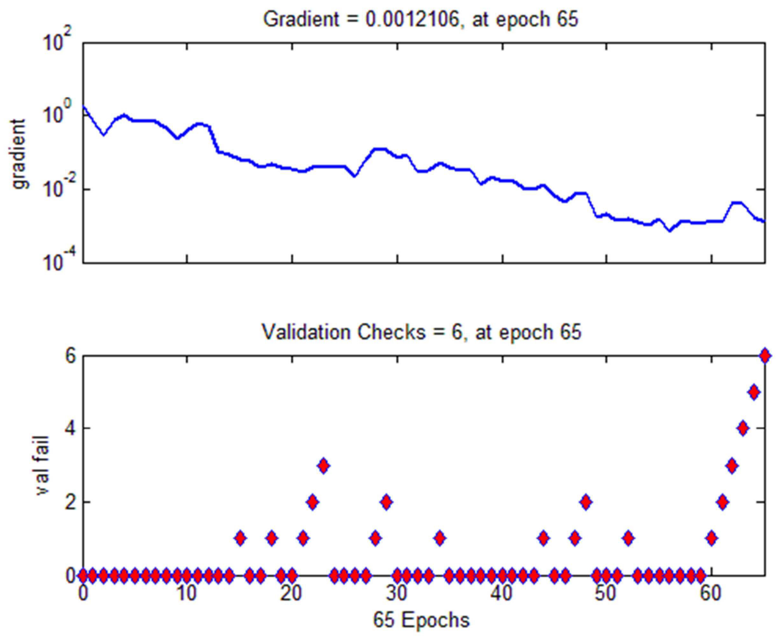

Figure 6. Training state plot for the three class classification task.

The three class classification task performs in a manner similar to the four class classification task. The error was decreased from 10^0 to roughly 10^{-3} at the 65th epoch. Also, the training was stopped because there were six validation checks, showing that the error was increasing repeatedly instead of decreasing.

All in all, depending on the amount of classes to be classified, the error decreases reaching different values. What is common to all networks is the fact that on the way to reaching the lowest possible error by shifting connections weights, the error increases and decreases at different epochs regardless of the epoch in which the training is. This means that it can be the case that at time 1 the network performs better than at time 2 or 3.

6.3 THIRD EXPERIMENT: GENERALIZATION OF NEW INPUT.

As stated in section 4.2, Pinker and Prince (1988) and Prasada and Pinker (1993) consider that there are two tests that connectionist models have to pass after training. The first one is to be able to recognize new input as regular or irregular forms. The second one is see their behaviour regarding pseudo words. The third experiment is related to the former and the fourth experiment to the latter. Therefore, in this section the results related to the generalization of new input are presented.

In order to test the new input, the function 'train' is used as follows:

Sim(net, testinput)

Where 'net' stands for the trained network and 'testinput' for the new input. The results of that simulation are compared to the output expected for the new input. Before doing that, the results are normalized by subtracting an amount and then being rounded with the 'ceil' function to the closest integer, number that can be expressed without a fractional component. The reason behind this is that the produced output matrix is conformed of elements which are not integers. By doing this, the highest element is the only element that becomes relevant for the comparison.

After this, a matrix is created as a result. If the elements of the matrix are zero, it means that there is no difference between the produced output and the expected output and that the new inputs were correctly classified. An if-statement is included to transform the elements that are 0 into 1 and the elements that are 1 into 0. These new elements are summed to produce the amount of correct classifications.

6.3.1 EIGHT CLASS CLASSIFICATION TASK.

In this classification task the amount of regular inputs was 614 and the amount of irregular inputs 27. After the new simulation was carried out both regular and irregular inputs were classified correctly. These results show that this network is able to perfectly generalize to new inputs.

6.3.2 FOUR CLASS CLASSIFICATION TASK.

In this classification task the amount of regular inputs was 478 and the amount of irregular inputs 12. As in previous classification task, this network was able to correctly classify both regular and irregular inputs.

6.3.3 THREE CLASS CLASSIFICATION TASK.

In this classification task the amount of regular inputs was 478 and the amount of irregular inputs 12. As in the previous classification tasks, this network was able to correctly classify both regular and irregular inputs.

In conclusion, the three networks were able to classify new input, complying with the conditions stated by Pinker and Prince (1988).

6.4 FOURTH EXPERIMENT: GENERATIVE SIMPLIFIED MODEL.

As stated in in section 6.3.4, the networks above are not generative in the sense that they do not produce output as Rumelhart's and McClelland's (1986) and MacWhinney's and Leinbach's (1991) networks did. However, the back propagation network can be used to produce a very simplified version of the previous models. Furthermore, with this kind of network pseudo words could be tested as well.

In this experiment, the tests performed after training are related to the production of regular past tense form from new inputs and the correct production of regular or irregular past tense forms from a corpus of pseudo words. Ideally, this network would have to be able to produce the correct matrix depending if the input would have a regular past tense form [1,0] or an irregular paste tense form [0,1]. Then, for the new inputs, all of them regular base forms, the network should produce the regular past tense matrix. Finally, the pseudo words should receive an output according to their resemblance to regular base forms or irregular base forms.

As stated on section 6.3.4, the initial training input was composed of 1456 base forms, 148 of them have an irregular past tense form and 1308 of them have a regular past tense form. The first network was composed of 200 hidden units and 2 hidden layers. The training results were quite disastrous for irregular past tense: 0% of correct outputs. Increasing the number of hidden units and layers only improved the percentage of correct outputs related to regular past tense but not the percentage of correct outputs related to irregular past tense. Considering that the amount of base forms that have an irregular past tense form was 10% of the total input dataset, there is one possible cause for the results the network was producing: the amount of irregulars is too small. To solve this problem, more input had to be supplied to the network. Considering that the amount of irregular forms was small compared to the regular forms, MacWhinney and Leinbach (1991) used a procedure in which the most frequent verbs were presented repeatedly during training. Therefore, the decision was made to increase the number of base forms with an irregular past tense form. The procedure was to copy the 148 irregular base forms two times, increasing the presence of irregular base forms to 444. After doing this, the results for irregular base forms increased considerably.

6.4.1 TRAINING OF THE NETWORK.

The task of knowing the past tense form based on the base form did not have the same results as the classification tasks. Figure 24 presents the confusion plot for this network after training stopped.

Figure 7. Confusion plot for the present_past network.

As we can see on this figure, the network had an overall performance of 95.5%, being the highest percentage of correct instances for base forms with regular past tense (98.1%). The base forms with an irregular past tense were given the correct output more than 90% of the time. These results are below the expectations set by Pinker and Prince (1988) discussed in past sections.

6.4.2 GENERALIZATION OF NEW INPUT.

Out of the 336 regular inputs, 32 were produced with incorrect output. This is to be expected considering that at training the network was not able to produce the expected output for all the inputs. These results as well are below the expectations set by Pinker and Prince (1988).

6.4.3 PSEUDO WORDS TEST.

Out of the 60 pseudo words, 27 were produced with incorrect output. This means that almost 50% of outputs differed with the expected outputs. 17 of the pseudo words with incorrect output resembled irregular base forms and 10 resembled regular base forms. According to Westermann (2000), these results are below the human subject performance.

6.5 THE SOM SIMULATION.

As stated on section 5.2.2, SOM networks are perfect for clustering data based on their similarities. These networks learn in an unsupervised manner, meaning that there is no expected output. The following figures show how the input for the eight class classification task clustered

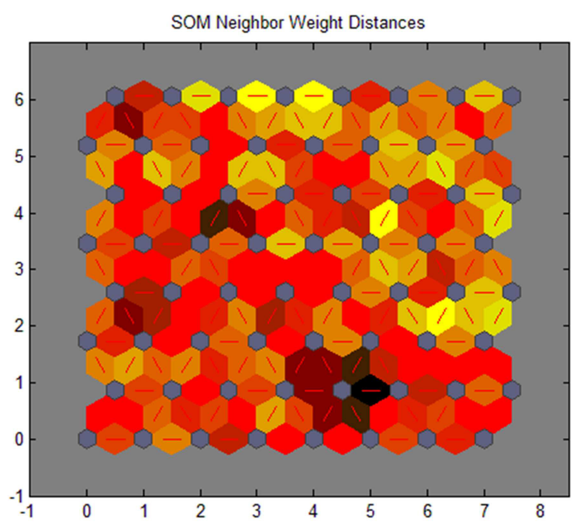

Figure 8. SOM neighbour weight distances for the eight class input.

The figure above shows the connections between neighbouring neurons. The grey blue patches are the neurons and the red lines the connections to their neighboring neurons. The color of each connection represents how close the neuron's weight is to its neighbors. Red is the strongest positive connection, blue, the strongest negative connection and black, no connection at all.

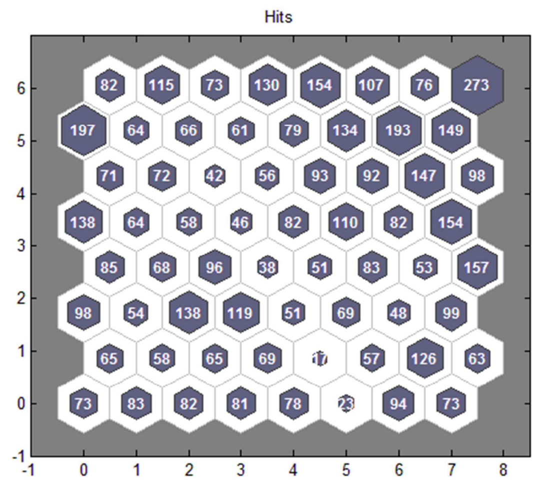

Figure 9. Sample hits for the eight class input.

This figure shows the amount of inputs that were clustered in each neuron.

Unfortunately, from the clustering of the inputs, there is no way to differentiate between discrete classes as in the case of the back propagation network. Especially, because the clustering can be done in different manners (Haykin, 1999, Trappenberg, 2010). It might be the case that the SOM network focused on the stems of the words to cluster them or in the prefixes or affixes. What is positive about these networks is the fact that different inputs, in this case, words that may or may not have related features, can be grouped together without an explicit rule.

7. DISCUSSION

In this section, the results presented in section 6. will be discussed in detail, focusing on the implications such results have for connectionist models, innatism and language acquisition.

7.1 TRAINING OF THE NETWORKS.

Regarding the back propagation classification models, the training of the network was successful in the sense that they were able to classify the training inputs correctly. The amount of classes involved in the task did not posit any trouble to the successful training of each network. The only improvement that could be stated is that the amount of time needed to train each network was proportional to the amount of classes but this is not relevant, especially because the amount of time needed to train the networks was less than 20 seconds.

This suggests something interesting. Classification of forms might not be a demanding task and, therefore, the difficulty of it does not increase as the amount of classes increases.

These results show that the ability to classify verb forms into different classes can be easily acquired by simple back propagation networks.

7.2 U-SHAPED LEARNING.

The training of the networks does not show a smooth and linear reduction of the error in order to achieve the expected output. The error increases and decreases as epochs go by and the weights are shifted in order to find the best possible configuration. If we are to consider U-shaped learning as presented in section 4.2, the networks are not able to produce this curve of learning because there should have been no error at all at first, then, the error should have increased sharply to then sharply decrease. According to Westermann (2000), the inability to produce u-shaped learning as described above is a proof that previous connectionist models failed to acquire the English past tense as children do. Therefore, that would mean that the back propagation networks used in this research failed. Nonetheless, when MacWhinney and Leinbach (1991) addressed the critiques made by Pinker and Prince (1988), they noted that there is no experimental evidence to state that U-shaped learning affects all the verbs in the same manner at the same time (MacWhinney, 1974; Kuczaj, 1977); Derwing, 1979; Derwing and Baker, 1979). Marcus et al. (1992) concluded after using the data from three different corpora that there is little evidence for across-the-board U-shaped learning in past tense acquisition. Alternatively, it is the case that different verbs have different U-shaped learning curves. Some of them have a strong u-shaped learning, other a weaker version and others none at all.

The evidence above can serve as an explanation why the error increases and decreases in training. As different verbs have different learning curves, the weights are increased and decreased at different rates producing errors as new verbs are in the process of being learnt.

This evidence, which was not considered by critiques to connectionist models, also serves as a proof that the back propagation networks are able to perform similarly as human subjects.

.

7.3 GENERALIZATION TO NEW INPUT.

The fact that the network was successful at the training process does not mean that it has learnt correctly. It may be the case that the network memorized the training dataset. The manner to test this is by presenting new input to the network. The inputs that were presented to test the generalization potential of the network were one set of infrequent regulars and one set of infrequent irregulars.

The three back propagation networks generalized correctly the two datasets. There were no misclassifications and, as in the same case with training, the amount of classes did not increase the difficulty of the task at all. Westermann (2000) suggests that it is misleading to test using irregular verbs because they are not predictable at all. The fact that the three networks are able to classify irregulars correctly further suggests that classification is not a very demanding task. This works as an argument for the implementation of the last experiments, whose results are discussed in 7.4.

Although the ability to classify irregulars cannot be accounted for because of their unpredictable and idiosyncratic nature, the correct classification of regulars is to be expected. The three back propagations networks discussed here are able to produce such classification for all the regulars without a problem.

In conclusion, the three networks were able to perform perfectly regardless of the amount of verb forms even in tests in which error may have been expected. Compared to previous results with generative connectionist models, these results suggest that classification is a less demanding and easier task than production of language. The experiments with the generative simplified model which uses the same learning algorithm as the back propagation networks of the classification task can shed light on this matter.

7.4 GENERATIVE SIMPLIFIED MODEL.

The last back propagation network received as input base forms that had a regular past tense form and base forms what had an irregular past tense form. As stated in section 6.3.4, this is a very simplified version of the generative models devised by Rumelhart and McClelland (1986) and MacWhinney and Leinbach (1991). Also, the amount of base forms with irregular past tense was increased as stated in section 6.4, in order to provide the network with enough inputs.

7.4.1 TRAINING OF THE GENERATIVE SIMPLIFIED MODEL.

After training stopped, almost a 100% of the regulars and 90% of the irregulars were produced correctly. These results are similar to the ones reached by MacWhinney and Leinbach. Unfortunately, Westermann (2000) following Pinker's and Prince's critique (1988), considers that the results reached by MacWhinney and Leinbach and the fact that they could not account for the traditional view of u-shaped learning 'do not make it a realistic model of child past tense acquisition'(Westermann, 2000:98).

This network was trained with different possible configurations in an attempt to improve the training results and reached a 100% of correct outputs for both types of base forms but these efforts resulted fruitless being the best possible results the ones being discussed here. Therefore, according to the critiques against connectionist models of language acquisition, the generative simplified model is not able to completely model English past tense acquisition in the training stage.

7.4.2 U-SHAPED LEARNING IN THE GENERATIVE SIMPLIFIED MODEL.

Regarding u-shaped learning, as it was done on the results section, the training state plot of the network provides us with a graphic means to see how the error in performance increased or decreased as training occurred.

The following graph shows the training state for the generative simplified model discussed here.

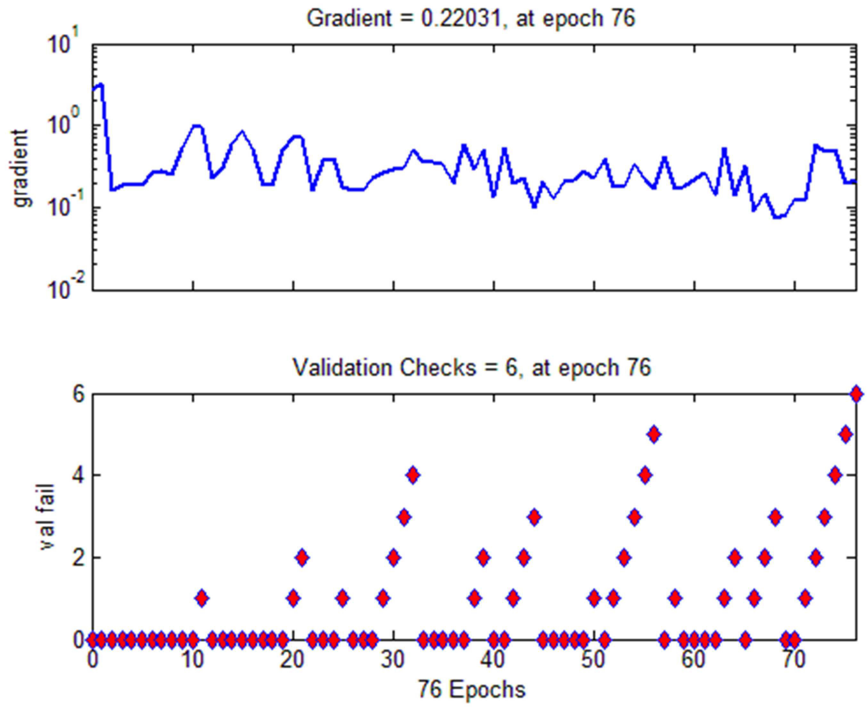

Figure 1. Training state of the generative simplified model.

The learning curve in this graph shows a dramatic reduction of the error of the network on the first epochs of training. Then the error increases to decrease again. If this had been the only curve, then this would have complied with the traditional perspective of u-shaped learning for past tense acquisition. Nonetheless, as MacWhinney and Leinbach (1991) noted, literature on the subject of U-shaped learning in English past tense acquisition shows that there is no across-the-board U-shaped learning for verbs. That can explain why in the generative simplified model there are sharp increases and decreases in error, small increases and decreases and stable states during training.

According to this, the learning curves found in the generative simplified model show a performance similar to human subjects, successfully modeling u-shaped learning.

7.4.3 REGULAR PAST TENSE TEST.

Following Westermann (2000) considerations, this model was tested against regular base forms and pseudo words. There was no irregular test because irregular past tense forms cannot be predicted.

In the regular test, for 90% of the verbs, the outputs were produced correctly.

These results suggest that a generative simplified model is not able to emulate human performance thoroughly. At least, in 90% of the cases it will produce the output a human subject is expected to produce. It may be that the cases in which the produced output differed with the expected outputs are instances of verbs that fall on the category of distant regular or irregular verbs. In fact, this seems to be the situation. Especially since the regular test was composed of the least frequent regular verbs. For example, the vowel elements in the base form ‘blind’ are similar to the vowel elements of the base form ‘grind’, which has an irregular past tense. Therefore, the network might treat it as an irregular. The same can be said about ‘wing’, ‘stink’, ‘sight’ to name a few. Prasada and Pinker (1993) note that this is in fact true for human subjects, being the case that novel verbs can be treated as irregular verbs based on the global similarity between them.

It can also be the case that verbs with same spelling but different past tense form. This is the case of the verb ‘bind’ that appears first as a base form with an irregular past tense form in the training dataset but also as base form in the regular test dataset. The same can be said about ‘string’, ‘grind’ to name a few.

7.4.4 PSEUDO WORDS TEST.

In the pseudo words test the results show there is a tendency for overregularization because 17 of the 30 irregular pseudo verbs were treated as regular. Nonetheless, 27 out of 60 forms presented as input were not given the expected output.

Prasada and Pinker (1993) on their study of the generalization of irregular and regular verbs prepared and used the set of pseudo words used in this simulation. They found varying degrees of responses which depended on the similarity the pseudo words had with existing verbs, as discussed in section 4.3.2. Furthermore, the responses were not the same for all the subjects. For example, for one distant regular pseudo verb there were combinations of regular and irregular inflexions such as adding the suffix –ed and also changing a vowel in the verb. In the case of distant irregular pseudo verbs, the subjects tended to regularize the verbs. These two findings can explain why there were almost 50% of wrong outputs and why most of them should have had an irregular past tense form.

All in all, the generative simplified model is able to comply with the experimental data of human subjects for u-shaped learning, regular generalization and the inflection of pseudo words. The only problem is that in the training stage it should have provided the correct output in all instances of regular and irregular verbs. Nonetheless, that problem could be solved with a larger training dataset or with a longer span of training, especially because human subjects overregularize verbs both in childhood and adulthood (Marcus et al., 1992).

8. CONCLUSIONS

The purpose of this thesis was to study the acquisition of the English past tense in order to provide new arguments for the discussion of nature v/s nurture. So far, connectionist models have been disregarded because they have not been able to mimic human performance entirely as in the case of MacWhinney's and Leinbach's model or because there were methodological issues and inaccurateness when compared to human performance as in the case of Rumelhart's and McClelland's model (Pinker and Prince, 1988; Prasada and Pinker, 1993; Westermann, 2000).

In order to study the acquisition of the English past tense by connectionist models, two ANNs were built. The first one was a back propagation network with supervised learning. This meant that the expected output was provided so that the network could compare how different was the produced output from the expected output to decrease the difference and learn. The second one was a self-organizing map with unsupervised learning. This meant that the network learnt from the similarities between the input and learnt how to cluster them.

Unfortunately, the self-organizing map was not designed for the task of classification or simplified production. Although, the network was able to cluster the input into different classes, these could not be compared to the classes provided by the MacWhinney and Leinbach (1991) corpus. As a suggestion, a SOM could cluster the initial input and feed it to a back propagation network with the intention to test how that kind of network could work.

The task of classifying according to predefined classes immediately excluded the learning potential of the SOM network. Because of that, the experiments and the discussion focused on the back propagation network and the generative simplified model. This immediately led to the conclusion that a supervised network was better suited for the task of classification and simplified production for the type of inputs and outputs used in this research.

The results of both the classification and generative networks suggest the following: classification is an easier task than production. This conclusion is based on the easiness with which the classification model was able to pass all tests and the difficulties that the generative simplified model had in the training stage and regular test to provide the expected outputs. The classification model, regardless the amount of classes, was able to pass all tests as the critiques of connectionist models required, except for U-shaped learning.

U-shaped learning as considered traditionally involves a first stage of correct usage, followed by a sharp decrease in performance, particularly for irregular verbs. Finally, the proper production of both regular and irregular forms is reached. Both models did not show this across-the-board U-shaped learning, making them not realistic models of English past tense acquisition. This would be true only if the traditional perspective of U-shaped learning is taken. As Marcus et al. (1992) concluded after an extensive research of English past tense acquisition in children, there is no evidence that across-the-board U shaped learning occurs. Rather, it is the case that different verbs have different stages of acquisition: some of them show a strong version of u-shaped learning, others a weak version and other none at all. This evidence suggests that the learning curves showed by both the classification and generative model are plausible in human performance.

Moreover, the problems the generative simplified model had in the regular test particularly can be accounted by the nature of the input. Some of the infrequent regular verbs used in this test had a frequent irregular counterpart. While others were distant regulars, and, as Prasada and Pinker (1993) have shown, could be treated as irregular verbs. This could also work the other way around: distant irregular verbs could be treated as regular verbs. This was the case for several verbs in the pseudo words tests. Consequently, even the generative simplified model could serve as model of English past tense acquisition. The only problem the network has to overcome is the correct production of expected outputs in the training stage. Unfortunately, the amount of base forms with an irregular past tense form was not enough to provide the network with all the examples needed. Nonetheless, as noted in previous sections, the performance in training was 98% for base forms with a regular past tense form and around 90% for base forms with an irregular past tense form. It could be the case for irregulars that they were overregularized which is a phenomenon that occurs in both children and adults (Marcus et al., 1992).

This conclusion leaves an open question and room for future research: Can it be that classification develops first in the brain in order to produce the correct instances? Modularity in the brain has been discussed by Fodor (1983) and the evidence of Broca's and Wernicke's areas shows that there are specialized areas in the brain that deal with linguistic knowledge in different ways. A simulation can be built in which the output of a classification network or a SOM network is given to a generative network to produce outputs to test this statement. However, these results have to be compared to the performance of human subjects; otherwise, they would not represent language acquisition.

Likewise, custom made algorithms and implementations for this task could be made in order to provide a completely realistic network which is provided with the same language that children receive and produce vocalizations as output.

8.1 LIMITATIONS OF THIS RESEARCH

The models studied here provide evidence for the notion that a UG is not hard wired into our brains because these models were able to perceive the classes and produce the relevant output without the need for rules. Certainly, this statement has to be considered cautiously because of the following reasons. First, not all the input children receive is presented to the networks. As Elman (1993) has shown, when ANNs are presented with an approximation of the whole range of sentences children are exposed to, their learning potential sharply decreases due to memory constraints.

Second, these models do not produce outputs by means of vocalizations.

A stronger learning algorithm and more advanced implementations would be needed to achieve this task, completely emulating human behavior. For the time being, the simplified models presented here provide us with tools to study linguistic knowledge theoretically, using ANNs as a psychological tool.

REFERENCES.

Anderson, J. (2009). *Cognitive psychology and its implications.* New York: Worth Publishers.

Archive.ics.uci.edu (1994) UCI Machine Learning Repository: Optical Recognition of Handwritten Digits Data Set. [online] Available at: http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits [Accessed: 18 Dec 2012].

Bates, E., Elman, J., Johnson, M., Karmiloff-Smith, A., Parisi, D., and Plunkett, K. (1998). Innateness and emergentism. In W. Bechtel and G. Graham (Eds.) *A Companion to Cognitive Science*. Oxford: Basil Blackwood.

Berko, J. (1958). The child's learning of English morphology. *Word*, 14, 150–177.

Brown, R. (1973). *A First Language. The Early Stages.* Cambridge: Harvard University Press.

Bybee, J. (1995) Regular morphology and the lexicon. *Language and Cognitive Processes*, 10 (5), p.425-455.

Chomsky, N. (1965) *Aspects of the theory of syntax.* Cambridge: M.I.T. Press.

Chomsky, N. (2000) *The architecture of language.* Oxford: Oxford University Press.

Chomsky, N. and Belletti, A., et al. (2002) *On nature and language.* Cambridge: Cambridge University Press.

Cs.bham.ac.uk (2009) Implementing a Neural Network in C. [online] Available at: http://www.cs.bham.ac.uk/~jxb/INC/nn.html [Accessed: 18 Dec 2012].

Derwing, B. (1979). English pluralization: a testing ground for rule evaluation. In G.Prideaux, B. Derwing, and W. Baker (Eds.), *Experimental linguistics: Part 1*. Ghent: E.Story-Scientia.

Derwing, B., and Baker, W. (1979). Rule learning and the English inflections (with emphasis on the plural). In G. Prideaux, B. Derwing, & W. Baker (Eds.), *Experimental linguistics: Part 2*. Ghent: E. Story-Scientia.

Desimone R. and Duncan, J. (1995). Neural Mechanisms of selective visual attention. *Annual Review of Neuroscience*, 18: 193-222.

Elman, J. (1993) Learning and development in neural networks: The importance of starting small. *Cognition*, 48, 71-99.

Elman, J. (2001). Connectionism and language acquisition. In Tomasello, M. and Bates, E. (Eds.), *Essential readings in language development*. Oxford: Basil Blackwell.

Francis, W. and Kucera, H. (1982) *Frequency analysis of English usage*. Boston: Houghton Mifflin.

Fodor, J. (1983) *The modularity of mind*. Cambridge, Mass.: MIT Press.

Fodor, A. and Pylyshyn, W. (1988) Connectionism and Cognitive Architecture: A Critical Analysis. *Cognition* 28, 3–71.

Gurney, K. (1997). *An introduction to neural networks*. London: UCL Press.

Hauser, M. (1996) *The evolution of communication*. Cambridge, Mass.: MIT Press.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall.

Heatonresearch.com (2008) The Number of Hidden Layers | Heaton Research. [online] Available at: http://www.heatonresearch.com/node/707 [Accessed: 5 Jan 2013].

Hebb, D. (1949) *The organization of behavior*. New York: Wiley.

Heinke, D., and Humphreys, G. W. (2005). Computational Models of Visual Selective Attention: A Review. [online] Available from: <http://www.comp-psych.bham.ac.uk/ publications/review.pdf> [Accessed 22 November 2012]

Hickey, C and Theeuwes, J. (2011) Context and Competition in the capture of visual attention. *Attention, Perception and Psychophysics*, 73, 2053-2064.

Hinton, G. (2007) Learning multiple layers of representation. *Trends in Cognitive Science*, 11 (10), 428-434

Hopfield, J (1982) Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the USA*, vol. 79 (8), 2554–2558.

Huynh, T.Q.; Reggia, J.A. (2011) Guiding Hidden Layer Representations for Improved Rule Extraction From Neural Networks. *Neural Networks, IEEE Transactions on*, 22 (2), 264 - 275

Johnston, M. (2009) Plasticity in the developing brain: implications for rehabilitation. *Developmental Disabilities Research Reviews*, 15 (2), 94-101.

Kleene, S. (1956). Representation of events in nerve nets and finite automata. In Shannon, C. and McCarthy, J. (eds), *Automata Studies*, 3-42. Princeton, N.J: Princeton University Press.

Kohonen, T. (1982). Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics* 43 (1), 59–69.

Kuczaj, S. A. (1977). The acquisition of regular and irregular past tense forms. *Journal of Verbal Learning and Verbal Behavior*, 16, 589–600.

Macwhinney, B. (1974) *How Hungarian children learn to speak.* University of California, Berkeley.

Macwhinney, B. and Bates, E. (1989) *The Crosslinguistic study of sentence processing.* Cambridge: Cambridge University Press.

MacWhinney, B. and Leinbach, J. (1991). Implementations are not conceptualizations: Revising the verb learning model. *Cognition*, 40, 121–157.

MacWhinney, B. (1993). Connections and symbols: Closing the gap. *Cognition*, 49, 291–296.

Marcus, G. F., Pinker, S., Ullman, M., Hollander, M., Rosen, T. J., and Xu, F. (1992). Overregularization in language acquisition. Monographs of the Society for Research in Child Development, 57 (4), 1:182.

Mathworks.co.uk (2013) Self-organizing map - MATLAB selforgmap - MathWorks United Kingdom. [online] Available at: http://www.mathworks.co.uk/help/nnet/ref/selforgmap.html [Accessed: 5 May 2013].

McCullough, W.S., and Pitts, W.H. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 115-133.

Minsky, M. and Papert, S. (1969) *Perceptrons*. Cambridge, Mass.: MIT Press.

O'grady, W. (2008) Innateness, universal grammar, and emergentism. *Lingua*, 118, 620-631.

Pendulum Project - Sccswiki (2009) File:Activation functions.PNG - Sccswiki.

[online] Available at: http://www5.in.tum.de/wiki/index.php/ File:Activation_functions.PNG [Accessed: 10 Feb 2013].

Pinker, S. and Prince, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition*, 28, 73–193.

Prasada, S. and Pinker, S. (1993). Generalization of regular and irregular morphological patterns. *Language and Cognitive Processes*, 8(1), 1–56.

Prechelt, L. (1998) Early Stopping-| but when?. *Neural Networks: Tricks of the trade*, 55-69.

Putnam, H. (1967) The ‘innateness hypothesis’ and explanatory models in linguistics. *Synthese*, 17 (1), 12-22.

Radanovic, M. and Mansur, L. (2011) *Language disturbances in adulthood*. United Arab Emirates: Bentham eBooks.

Rosenblatt, F. (1957) The Perceptron--a perceiving and recognizing automaton. Report No 85-460-1, Cornell Aeronautical Laboratory.

Rumelhart, D. E. and McClelland, J. L. (1986). On learning the past tenses of English verbs. In J. L. McClelland, D. E. Rumelhart, & The PDP Research Group, *Parallel distributed processing: Explorations in the microstructure of cognition. Volume 2: Psychological and biological models.* Cambridge, MA: Bradford Books/MIT Press.

Rumelhart, D. E. and McClelland, J. L. (1987). Learning the past tenses of English verbs: Implicit rules or parallel distributed processing? In B. MacWhinney (Ed.), *Mechanisms of language acquisition*. Hillsdale, NJ: Erlbaum.

Skinner, B. (1957) *Verbal behaviour*. New York: Appleton-Century-Crofts.

Smith, F. (2011) *Understanding Reading*. Hoboken: Taylor & Francis.

Stiles, J. (2000) Neural Plasticity and Cognitive Development. *Developmental Neuropsychology*, 18 (2), 237-272.

Trappenberg, T. (2010). *Fundamentals of Computational Neuroscience* 2nd Ed. Oxford: Oxford University Press.

Westermann, G. (2000) Constructivist Neural Network Models of Cognitive Development. Ph D. University of Edinburgh.

Wilson, R. and Martinez, T. (2003) The general inefficiency of batch training for gradient descent learning. *Neural Networks*, 16, 1429-1451

Wolfe, J. M. (1998). Visual Search. In H. Pashler (Ed.) *Attention*, 13-74. East Sussex: Psychology Press Ltd.

A. APPENDIX 1: CODE FOR THE BACK PROPAGATION NETWORK IN SECTION 5.1.5.2

Begin Batch Back-Propagation
    initialize wij, wjk
    for $epoch = 1$ **to** T
        $\Delta wij = zero$
        $\Delta wjk = zero$
        for $p = 1$ **to** N(number on training examples)
            $\vec{x} :=$ p input vector
            $\vec{t} :=$ p target vector
            Compute $\vec{y}$ hidden layer output vector
            Compute $\vec{o}$ network output vector
            for $i = 1$ **to** I(number of inputs)
                for $j = 1$ **to** J(number of neurons on hidden layer)
                    $\Delta wij_{i,j} = \Delta wij_{i,j} + \eta\delta_j x_i$
                end
            end
            for $j = 1$ **to** J(number of neurons on hidden layer)
                for $k = 1$ **to** K(number of neurons on output layer)
                    $\Delta wjk_{j,k} = \Delta wjk_{j,k} + \eta\delta_k y_j$
                end
            end
        end
                    $wij = wij + \Delta wij$
                    $wjk = wjk + \Delta wjk$
    if termination criterion
        Break
    end
end

B. APPENDIX 2: LIST OF IRREGULAR VERBS

arose
ate
awoke
beat
became
bent
bled
blew
bore
bought
bound
broke
brought
built
burst
came
cast
caught
chose
clung
cost
crept
cut
dealt
drank
drew
drove
dug
fed
fell
felt
fled
flew
flung
forgave
forgot
fought
found
froze
gave
got
grew
ground
heard
held
hid
hit
kept
knew
laid
led
left
lent
let
lost
made
meant
met
overcame
put
quit
ran
rang
read
rode
rose
said
sang
sank
sat
saw
sent
set
shed
shone
shook
shot
shut
slept
slid
sold
sought
spat
spent
split
spoke
sprang
spread
spun
stole
stood
strode
strove
struck
stuck
swam
swept
swore
swung
taught
thought
thrust
told
took
tore
understood
underwent

upheld
upset
went
wept
wet
withdrew
woke
wore
wound
wove
wrote

B. APPENDIX 3: TRAINING DATASET

abandon
abandoned
abandoned
abandoning
abetted
abetted
abide
abides
abiding
abolish
abolished
abounded
absent
absorb
absorbed
absorbed
absorbing
abuse
abused
accept
accepted
accepted
accepting
accepts
accompany
accompanied
accompanied
accompanying
accomplish
accomplished
accomplished
accomplishing
accorded
according
account
accounted
accounting
accounts
accrued
accruing
accuse
accused
accused
accuses
accusing
ached
aching
achieve
achieved
achieved
achieves
achieving
acknowledge
acknowledged
acknowledged
acquaint
acquainted
acquiesce
acquire
acquired
acquired
acquiring
act
acts
acted
acted
acting
activated
adapt
adapted
adapting
add
added
added
adds
adding
address
addressed
addressed
addresses
addressing
adhere
adhered
adjoined
adjoining
adjoins
adjust
adjusted
adjusted
adjusting
adjusts
admire
admired
admired
admiring
admit
admits
admitted
admitted
admitting
admonished
adopt
adopted
adopted
adopting
adopts
adore
advance
advanced

advanced
advancing
advertise
advertised
advertised
advertising
advise
advised
advised
advises
advising
advocate
advocated
advocating
affect
affected
affected
affecting
affects
affirm
affirmed
afford
afforded
afforded
affording
affords
age
aged
aging
agonizes
agree
agreed
agreed
agreeing
agrees
aid
aided
aiding
aids
aim
aimed
aimed
aiming
aims
alarm
alarmed
alert
alerting
alienated
align
aligned
alleged
alleging
alleviate
allocate
allocated
allocated
allotted
allotted
allow
allowed
allowed
allowing
allows
alter
altered
altered
altering
amass
amaze
amazed
ambush
ambushed
amended
amended
amortize
amount
amounted
amounted
amounts
amuse
amused
amused
analyze
analyzed
analyzing
announce
announced
announced
announces
announcing
annoyed
answer
answered
answered
answering
answers
appeal
appealed
appealed
appealing
appear
appeared
appeared
appearing
appears
appease
applaud
applauded
apply
applied
applied
applies
applying
appoint
appointed
appointed

appraise
appreciate
appreciated
appreciated
approach
approached
approached
approaches
approaching
appropriated
appropriating
approve
approved
approved
arbitrate
arched
arches
argue
argued
argued
argues
arguing
arise
arises
arising
arose
arisen
arouse
aroused
aroused
arouses
arousing
arrange
arranged
arranged
arranging
arrest
arrested
arrested
arrive
arrived
arrived
arrives
arriving
ascending
ascertain
ascertained
ascribed
ask
asked
asked
asking
asks
aspire
assail
assailed
assaulted
assaulted
assemble
assembled
assembled
assembling
assented
assert
asserted
asserted
asserting
asserts
assess
assessed
assessing
assign
assigned
assigned
assigning
assigns
assist
assisted
assisted
assisting
associate
associated
associated
associating
assume
assumed
assumed
assumes
assuming
assure
assured
assured
assures
assuring
astonished
astounded
attach
attached
attached
attaches
attaching
attack
attacked
attacked
attacking
attacks
attain
attained
attained
attaining
attempt
attempted
attempted
attempting
attempts
attend
attended

attended
attending
attends
attest
attested
attract
attracted
attracted
attracting
attracts
attributed
attributed
attributing
augment
augmented
augmented
authorize
authorized
authorized
authorizes
authorizing
avail
availed
avenging
average
averaged
averaging
averted
averting
avoid
avoided
avoided
avoiding
avoids
await
awaited
awaiting
awaits
awake
awoke
awaken
awakened
award
awarded
awarded
awarding
back
backed
backed
backing
backs
backstitch
bake
baked
baking
balance
balanced
balancing
balked
bandaged
bang
banged
banging
banish
banished
banked
banking
bankrupt
banned
bar
barred
barred
barring
bars
barge
barging
base
based
based
bases
basing
bat
batting
bathe
bathed
bathing
battered
battered
battle
battling
bawled
bayed
bear
bearing
bears
bore
born
beat
beating
beats
beat
beaten
beckoned
beckons
become
becomes
becoming
became
become
befall
beg
begged
begging
begin
beginning
begins
begrudge

behave
behaved
behaved
behaves
behaving
behold
belched
belied
believe
believed
believed
believes
believing
bellowed
bellowing
belong
belonged
belonged
belonging
belongs
belted
bend
bending
bent
bent
benefit
benefited
bestow
bestowed
bestowed
bet
betting
betide
betray
betrayed
betrays
better
bevel
beveled
beware
bewitched
bickering
bid
binding
binds
bound
bound
birdied
birdied
blame
blamed
blaming
blast
blasted
blaze
blazed
blazing
bleached
bleed
bleeding
bled
blend
blended
blended
bless
blessed
blinded
blinding
blindfolded
blink
blinked
blinking
blistered
block
blocked
blocked
blocking
blockading
bloom
bloomed
blooming
blossom
blotted
blotting
blow
blowing
blows
blew
blown
blundered
blunt
blush
blushed
blushing
board
boarded
boarded
boarding
boast
boasted
boasting
bobbed
bobbing
boil
boiled
boiling
bolster
bolt
bolted
bolted
boost
boosted
boosting
boot
bordered
bordering
bore

bored
bored
boring
borrow
borrowed
borrowed
borrowing
bother
bothered
bothered
bothering
bothers
bottled
bounce
bounced
bounced
bouncing
bound
bounded
bounded
bounding
bow
bowed
bowing
brace
braced
branch
branched
branded
break
breaking
breaks
broke
broken
breakfasted
breathe
breathed
breathing
breed
breeding
bribed
bridge
bring
bringing
brings
brought
brought
bristled
bristling
broaden
broadened
broadened
broadening
broiled
bruised
bruising
brush
brushed
brushed
brushing
buckle
bud
budge
budget
budgeting
build
building
builds
built
built
bulged
bulging
bump
bumped
bury
buried
buried
burn
burned
burned
burning
burst
burst
burst
bursting
butt
butted
buy
buying
buys
bought
bought
buzzed
buzzing
cackled
calculate
calculated
calculated
calculating
call
called
called
calling
calls
calm
calmed
calmed
calving
campaign
campaigned
campaigning
canned
canning
cancel
canceled
cap
capture

captured
capturing
care
cared
cared
cares
caring
caressed
caressing
carry
carried
carried
carries
carrying
carve
carved
carving
cash
cast
casting
casts
cast
cast
catalogued
catch
catching
caught
caught
cater
catering
cause
caused
caused
causes
causing
cautioned
cease
ceased
ceased
ceases
celebrate
celebrated
celebrated
celebrates
celebrating
center
centered
centered
centering
centers
centralized
centralizing
certify
certified
challenge
challenged
challenged
challenges
challenging
chanced
change
changed
changed
changes
changing
channeled
chanted
chanted
chanting
charge
charged
charged
charging
charm
charted
charting
chase
chasing
chatted
chattering
cheat
cheated
check
checked
checked
checking
cheer
cherish
cherished
cherished
chew
chewed
chewing
chide
chilled
chilled
chilling
chipped
chipping
choke
choked
choked
choking
choose
chooses
choosing
chose
chosen
chopped
chopping
chortled
chuck
chuckle
chuckled
circled
circling
circulate

circulated
circulating
cite
cited
cited
cites
citing
claim
claimed
claimed
claiming
claims
clambered
clamped
clamped
clamping
clapped
clapping
clarify
clarified
clarifying
clasped
clasping
classify
classified
clattered
clean
cleaned
cleaned
cleaning
clear
cleared
cleared
clearing
clenched
clicked
clicked
climb
climbed
climbed
climbing
cling
clinging
clings
clung
clog
close
closed
closed
closes
closing
cluster
clustered
clutched
clutched
clutching
cluttered
cooperate
coasted
cocked
cocked
coincide
coincided
coincides
collapsed
collapsed
collapsing
collar
collect
collected
collected
collecting
collects
color
colored
coloring
colors
combat
combed
combine
combined
combined
combines
combining
come
comes
coming
came
come
comforting
command
commanded
commanded
commanding
commands
commenced
commencing
commend
commended
commending
comment
commented
commented
commenting
commit
commits
committed
committed
committing
commute
commuting
compare
compared
compared
compares
comparing
compel

compelled
compels
compensate
compensated
compete
competing
compiled
compiled
compiling
complain
complained
complaining
complains
complete
completed
completed
completes
completing
complicate
complicated
comply
complied
complied
complying
compose
composed
composed
composes
composing
compounded
comprehend
comprehending
compressed
comprise
comprised
comprised
comprises
comprising
compromise
compromising
compute
computed
computes
computing
conceal
concealed
conceals
concede
conceded
conceded
conceding
conceive
conceived
conceived
conceives
concentrate
concentrated
concentrates
concentrating
concern
concerned
concerned
concerns
conclude
concluded
concluded
concludes
concluding
concur
concurs
condemn
condemned
condemned
condemning
condemns
condensed
conditioned
conditioning
conduct
conducted
conducted
conducting
conducts
confer
conferred
confess
confessed
confesses
confessing
confide
confided
confiding
confine
confined
confining
confirm
confirmed
confirmed
confirming
confirms
confiscated
conflict
conflicting
conform
conformed
conforms
confront
confronted
confronted
confronting
confronts
confuse
confused
confused
congealed
conjures
connect

connected
connected
connecting
connects
conquer
conquered
conquering
consent
consented
conserve
conserving
consider
considered
considered
considering
considers
consign
consist
consisted
consisting
consists
conspired
constitute
constituted
constituted
constitutes
constituting
consult
consulted
consulted
consulting
consume
consumed
consuming
contact
contain
contained
contained
containing
contains
contemplate
contemplated
contemplating
contend
contended
contended
contends
content
contented
contest
contested
continue
continued
continued
continues
continuing
contract
contracted
contracted
contracts
contradict
contrast
contrasted
contrasting
contrasts
contribute
contributed
contributed
contributes
contributing
contrived
control
controlled
controlled
controlling
controls
convened
converge
converse
convert
converted
converting
convey
conveyed
conveyed
conveys
convicted
convince
convinced
convinced
convincing
cook
cooked
cooking
cool
cooled
cooled
cooling
cools
cooperate
cooperated
cooperating
coordinate
coordinated
coordinating
cope
coping
copied
core
corked
correct
corrected
corrected
correlate
correlated
correspond
corresponded

corresponding
corresponds
corrupting
cost
costing
costs
cost
cost
cough
coughed
coughing
counseled
counseling
count
counted
counted
counting
counts
counter
counteract
counteracting
coupled
coupled
coupling
court
courting
cover
covered
covered
covering
covers
coveted
crack
cracked
cracked
cracking
crash
crashed
crashed
crashing
crawl
crawled
crawled
crawling
crazed
creaked
creaking
creased
create
created
created
creates
creating
credit
credited
credits
creep
creeping
crept
crept
cringing
crippled
crippling
criticize
criticized
criticized
criticizing
cross
crossed
crossed
crosses
crossing
crouch
crouched
crouched
crouching
crowd
crowded
crowded
crowding
crowed
crowing
crowned
crowning
crumble
crumbling
crushed
crushed
crushing
cry
cried
cried
crying
culminated
culminates
cultivate
cultivated
cultivated
cultivating
cupped
cure
cured
curl
curled
curled
curse
cursed
cursed
cursing
curtail
curved
curving
cut
cuts
cutting
cut
cut

damage	decanting	deemed
damaged	decay	defeat
damages	decayed	defeated
damaging	decaying	defeated
damn	deceived	defeating
damned	decide	defend
dampen	decided	defended
dampened	decided	defended
dance	decides	defending
danced	deciding	defends
danced	declaimed	defy
dancing	declare	defying
dangling	declared	defied
dare	declared	defied
dared	declares	define
dared	declaring	defined
dares	decline	defines
daring	declined	defining
darkened	declined	defraud
darkening	declines	defray
darn	declining	delay
darned	decomposes	delayed
darted	decorate	delayed
dashed	decorated	delegate
dashed	decorating	delegated
dashing	decrease	delight
date	decreased	delighted
dated	decreases	deliver
dates	decreasing	delivered
dating	decry	delivered
dazzling	decried	delivering
deal	dedicated	delivers
dealing	dedicated	delude
deals	dedicates	demand
dealt	deduce	demanded
dealt	deduced	demanded
debated	deduct	demanding
debating	deducted	demands
decanted	deemed	deny

denying
denied
denied
denies
denote
denoted
denotes
denoting
denounce
denounced
denounced
denouncing
depart
departed
departed
departing
depend
depended
depending
depends
depict
depicted
depicted
depicting
deplores
deprive
deprived
depriving
derive
derived
derives
deriving
descend
descended
descended
descending
descends
describe
described

described
describes
describing
desert
deserted
deserts
deserve
deserved
deserved
deserves
design
designed
designed
designing
designs
designate
designated
designated
designating
desire
desired
desired
desires
desiring
despise
despised
destroy
destroyed
destroyed
destroying
detached
detect
detected
detecting
determine
determined
determined
determines
determining

detested
devastated
devastating
develop
developed
developed
developing
develops
devise
devised
devised
devote
devoted
devoted
devoting
devour
diagnose
dialed
dictate
dictated
dictates
dictating
die
died
died
dies
differ
differed
differs
diffused
diffusing
dig
digging
dug
dug
digesting
dilated
diluted
diluting

diminish
diminished
diminishes
diminishing
dine
dining
dip
dipped
direct
directed
directed
directing
directs
disabled
disabling
disabuse
disagree
disagreed
disagrees
disappear
disappeared
disappeared
disappearing
disappears
disapprove
disapproved
disarmed
disarming
discern
discerned
discerning
discharge
discharged
discharging
disciplined
disciplined
disclose
disclosed
disclosed
disconnected
discount
discounted
discourage
discouraged
discover
discovered
discovered
discovering
discovers
discuss
discussed
discussed
discusses
discussing
disdaining
disfigured
disfigured
disguise
disguised
disguised
disgusted
dishearten
dislike
disliked
dislikes
dislodge
dismembered
dismiss
dismissed
dismissed
dismissing
dismounted
dismounted
disobeyed
dispatched
dispatched
dispatching
dispel
dispelled
dispense
disperse
dispersed
displace
displaced
display
displayed
displayed
displaying
displays
dispose
disposed
disposed
disprove
dispute
disputed
disregard
disregarded
disregarding
disrupt
disrupted
disrupting
dissolve
dissolved
dissolving
dissuade
distinguish
distinguished
distinguishes
distinguishing
distort
distorted
distracted
distracted
distribute
distributed
distributing
distrust

disturb
disturbed
disturbed
disturbing
dive
dived
diving
diverted
diverting
divide
divided
divided
divides
dividing
divorce
divorced
dock
document
documented
dodge
dodging
doing
done
dominate
dominated
dominated
dominates
donate
donated
donned
doomed
doomed
doting
dotted
double
doubled
doubled
doubles
doubling
doubt
doubted
doubting
down
downed
downed
dozed
dozing
drafted
drafting
drag
dragged
dragged
dragging
drain
drained
drained
dramatize
dramatizes
draped
draped
draw
drawing
drew
drawn
draws
drawled
dream
dreamed
dreamed
dreaming
dreams
dress
dressed
dressed
dressing
dry
drying
dried
dried
drift
drifted
drifted
drifting
drifts
drill
drilled
drilling
drink
drinking
drinks
drank
dripped
dripping
drive
drives
driving
driven
drove
drop
dropped
dropped
dropping
drops
drown
drowned
drowned
drowning
drummed
drumming
duck
ducked
ducking
dump
dumped
dumping
duplicate
duplicated

dwarf
dwell
dwelling
dwindle
dwindled
dwindling
dyed
dying
earn
earned
earned
earning
earns
ease
eased
eased
easing
eat
eating
ate
eaten
echo
echoed
echoing
edged
edging
edit
edited
editing
educate
educated
educating
effect
effected
effecting
elect
elected
elected
elicit
elicited
elicited
eluded
eluding
embark
embarrassed
embarrassing
embodied
embodies
embodying
embrace
embraced
embraces
embracing
emerge
emerged
emerged
emerges
emerging
emitted
emphasize
emphasized
emphasized
emphasizes
emphasizing
employ
employed
employed
employing
employs
emptied
emptied
empties
emulate
enable
enabled
enabled
enables
enabling
enact
enacted
enacting
enclosed
enclosed
encompass
encompassed
encounter
encountered
encountered
encounters
encourage
encouraged
encouraged
encourages
encouraging
end
ended
ended
ending
ends
endeared
endorse
endorsed
endow
endowed
endure
endured
endured
endures
enduring
enforce
enforced
enforcing
engage
engaged
engaged
engaging
engender

engendered
engulfed
enhance
enhanced
enjoy
enjoyed
enjoyed
enjoying
enjoys
enlarge
enlarged
enlarging
enlist
enlisted
enlisted
enrich
enriched
enroll
enrolled
enrolled
ensue
ensued
ensues
ensuing
ensure
entail
entails
enter
entered
entered
entering
enters
entertain
entertained
entertained
entertaining
entitle
entitled
entitled
entitles
entrusted
envy
envied
equal
equals
equate
equated
equated
erasing
erect
erected
erected
erecting
eroded
erupt
erupted
erupted
escape
escaped
escaped
escapes
escaping
escort
escorted
establish
established
established
establishes
establishing
estimate
estimated
estimated
estranged
even
evoke
evoked
evoked
evokes
evolve
evolved
exacting
exacts
examine
examined
examined
examining
exceed
exceeded
exceeded
exceeding
exceeds
excite
excited
exciting
execute
executed
exercise
exercised
exercised
exercises
exercising
exert
exerted
exerted
exerting
exerts
exhaled
exhaust
exhausted
exhausted
exhausting
exhibit
exhibited
exhibited
exhibiting
exhibits
exist

existed
existed
existing
exists
expand
expanded
expanded
expanding
expands
expect
expected
expected
expecting
expects
expelled
expended
experience
experienced
experienced
experiences
experiencing
expired
export
exported
expose
exposed
exposed
exposes
exposing
extend
extended
extended
extending
extends
exuded
eyed
eyeing
face
faced
faced
faces
facing
faded
faded
fading
fail
failed
failed
failing
fails
fall
falling
falls
fell
fallen
falsify
falter
faltered
fan
fanned
fanning
fancy
fancied
farm
farming
fascinate
fascinated
fascinated
fashion
fashioned
fashioned
fasten
fastened
fastened
fathered
fathom
favor
favored
favored
favoring
favors
fear
feared
feared
fearing
fears
feature
featured
featured
features
featuring
feed
feeding
feeds
fed
fed
feel
feeling
feels
felt
felt
fell
felling
fetch
field
fielding
fight
fighting
fights
fought
fought
figure
figured
figured
figures
figuring
file

filed
filed
filing
fill
filled
filled
filling
fills
filtered
filtering
finance
financed
financing
find
finding
finds
found
found
fingered
finish
finished
finished
finishing
fire
fired
fired
firing
fit
fits
fitting
fitted
fitted
fix
fixed
fixed
fixing
flags
flame
flaming
flanked
flapped
flapping
flared
flared
flaring
flash
flashed
flashed
flashing
flattened
flattened
flattered
fleeing
fled
fled
flexed
flicked
flung
flung
flip
flipped
flipping
float
floated
floating
flogged
flood
flooded
flooded
flooding
flopped
flourish
flourished
flourishes
flow
flowed
flowed
flowing
flows
flowering
fluttered
fluttering
fly
flies
flying
flew
flown
foamed
focus
focused
focused
focuses
focusing
foil
fold
folded
folded
folding
follow
followed
followed
following
follows
fool
fooled
fooling
forbid
forbidding
forbids
forbidden
force
forced
forced
forces
forcing
forecast
forecast

forecasting
forego
foregoing
foresee
foreseeing
forestall
forget
forgetting
forgot
forgotten
forgive
forgave
forgiven
forked
form
formed
formed
forming
forms
formalize
formalized
formulate
formulated
formulated
formulating
fort
fortify
fortified
foster
fostered
fosters
fouled
founded
founded
founding
frame
framed
framed
framing

free
freed
freeing
frees
freeze
freezing
froze
frozen
frighten
frightened
frightened
frightening
frowned
frowning
frustrate
frustrated
fry
fried
fulfill
fulfilled
fulfilled
fulfilling
fulfills
fumbled
fumbling
function
functioned
functioning
functions
furnish
furnished
furnished
furnishes
further
furthering
fuse
fused
fussing
gain

gained
gained
gaining
gaped
gaping
gasped
gasping
gather
gathered
gathered
gathering
gaze
gazed
gazing
generalize
generalized
generate
generated
generated
generates
generating
germinate
gestured
get
gets
getting
got
got
giggled
give
gives
giving
gave
given
glance
glanced
glancing
glared
glaring

glaze
glazed
gleamed
gleaming
glimpsed
glimpsed
glinted
glinting
glistened
glistening
gloated
glorify
glorified
glow
glowed
glowing
glowered
glowering
glued
glued
go
goes
going
gone
went
gouged
gouging
govern
governed
governing
governs
grab
grabbed
grabbing
graduate
graduated
graduated
graduates
graduating
grant
granted
granted
granting
grasp
grasped
grasped
grasping
greet
greeted
greeted
grimace
grinned
grinning
grinding
ground
ground
gripped
gripped
gripping
groaned
grok
grokked
groped
groping
grounded
grounded
group
grouped
grouping
grow
growing
grows
grew
grown
growled
grumble
grunted
guarantee
guaranteed
guaranteed
guarantees
guard
guarded
guarding
guess
guessed
guessed
guessing
guide
guided
guided
guides
guiding
gulped
gushed
hail
hailed
halt
halted
hammered
hamper
hampered
hand
handed
handed
handing
handle
handled
handled
handles
handling
hang
hanging
hangs
hung
hung
happen

happened
happened
happening
happens
harassed
harassed
harassing
harbor
harbored
hark
harvesting
hasten
hastened
hastened
hastening
hate
hated
hated
hates
hating
haul
hauled
hauled
hauling
haunt
haunted
haunted
haunting
head
headed
headed
heading
heal
healed
healed
healing
hear
hearing
hears
heard
heard
heat
heated
heating
heaved
heaving
heed
help
helped
helped
helping
helps
herd
hesitate
hesitated
hibernate
hide
hiding
hid
hidden
hinted
hints
hire
hired
hired
hiring
hissing
hit
hits
hitting
hit
hit
hitched
hitching
hold
holding
holds
held
held
hollered
hollering
honor
honored
honored
honoring
honors
hooked
hooked
hope
hoped
hoped
hopes
hoping
hopped
hopping
house
housed
housed
houses
housing
hover
huddled
huddled
huddling
hug
hugged
hugging
hummed
humming
hunt
hunted
hunting
hurl
hurled
hurling
hurry
hurried

hurried
hurrying
hurt
hurting
hurts
hurt
ignite
ignore
ignored
ignored
ignores
ignoring
illumined
illustrate
illustrated
illustrates
illustrating
imagine
imagined
imagined
imagines
imitate
imitated
imitates
imitating
impair
impaired
impart
imparted
impinge
impinging
implement
implementing
imply
implied
implied
implies
implying
import
imported
imported
impose
imposed
imposed
imposes
imposing
impress
impressed
impressed
improve
improved
improved
improves
improving
improvise
improvised
incite
include
included
included
includes
including
increase
increased
increased
increases
increasing
incur
incurred
indicate
indicated
indicated
indicates
indicating
induce
induced
induces
inducing
indulge
indulged
inferred
inflict
inflicted
inflicting
influence
influenced
influences
inform
informed
informed
informing
informs
infuriated
ingested
ingested
inherit
inherited
inherited
inhibit
inhibited
inhibited
inhibiting
inhibits
initiate
initiated
initiated
initiating
inject
injecting
injured
injured
inquire
inquired
inquired
inquiring
insert
inserted

inserted
insist
insisted
insisted
insisting
insists
inspect
inspecting
inspire
inspired
inspired
inspiring
install
installed
installed
installing
instituted
instituted
insulate
insulated
insulating
insult
insure
insured
insuring
integrate
integrated
integrates
integrating
intend
intended
intended
intends
intercept
intercepted
interest
interested
interested
interfere
interfered
interfering
interpret
interpreted
interpreted
interpreting
interprets
interrupt
interrupted
interrupted
intersect
intersecting
intertwined
intervened
interview
interviewed
interviewed
interviewing
intimated
intoned
intrigued
introduce
introduced
introduced
introduces
introducing
inure
inured
invade
invaded
invading
invent
invented
invented
invest
invested
invite
invited
invited
invites
inviting
invoke
invoked
invoking
involve
involved
involved
involves
involving
iodinated
iodinating
ionized
ionizing
ironed
ironing
isolate
isolated
isolating
issue
issued
issued
issues
issuing
itch
itching
itemized
itemizing
jabbing
jam
jammed
jeopardize
jerked
jerking
jingled
join
joined
joined
joining

joke
joking
jolt
journeyed
judge
judged
judged
judging
jump
jumped
jumped
jumping
justify
justified
justifying
keep
keeping
keeps
kept
kept
keynote
kick
kicked
kicked
kicking
kill
killed
killed
killing
kills
kiss
kissed
kissing
kneel
kneeling
knelt
knit
knitted
knock
knocked
knocked
knocking
know
knowing
knows
knew
known
labeled
labeled
labeling
labor
labored
labored
lack
lacked
lacked
lacking
lacks
lag
land
landed
landed
landing
lapse
lash
lashed
last
lasted
lasting
laugh
laughed
laughed
laughing
launch
launched
launched
lay
laying
lays
laid
laid
lead
leading
leads
led
led
leaked
lean
leaned
leaning
leap
leaped
leaped
leaping
learn
learned
learned
learning
learns
lease
leased
leasing
leave
leaves
leaving
left
left
lecturing
leered
leering
lend
lending
lends
lent
lent
lengthen
lengthened

lengthening
lessen
lessened
lessened
lessening
let
lets
letting
let
let
leveled
leveled
leveling
levy
liberate
liberated
licensed
licensing
lick
licked
licked
lie
lying
lied
lies
lift
lifted
lifted
lifting
light
lighted
lighting
lit
lightened
like
liked
liked
likes
liking

limit
limited
limited
limiting
limits
line
lined
lined
linger
lingered
lingering
lingers
link
linked
linking
list
listed
listed
listing
lists
listen
listened
listening
live
lived
lived
lives
living
load
loaded
loading
loathed
locate
located
located
locating
lock
locked
locked

locking
lodging
logged
logging
long
longed
longed
longing
look
looked
looked
looking
looks
loomed
looming
loose
loosen
loosened
looted
looting
lose
loses
losing
lost
lost
lounged
lounging
love
loved
loved
loves
loving
lower
lowered
lowered
lowering
lugged
lugged
lunged

lurched
lurching
lure
lured
lurked
lurking
magnified
magnifying
mail
mailed
mailed
mailing
maintain
maintained
maintained
maintaining
maintains
make
makes
making
made
made
man
manned
manage
managed
managed
manages
managing
maneuvered
maneuvering
manifest
manifested
manifested
mar
marred
mars
march
marched

marched
marches
marching
mark
marked
marked
marking
marks
marketed
marketing
marry
married
married
marries
marrying
marshal
marvel
masquerades
mass
master
mastered
match
matched
matched
matches
matching
mate
mated
mating
matter
mattered
matters
mature
maturing
mean
meaning
means
meant
meant

measure
measured
measured
measures
measuring
meet
meeting
meets
met
met
melt
melted
melted
melting
memorize
menacing
mending
mention
mentioned
mentioned
mentioning
mentions
merge
merged
merging
merit
merited
merits
mesh
mess
messing
metered
mind
mingle
mingled
mingled
minimize
minimized
minimized

minimizing
ministered
ministering
mirrors
misleading
misled
miss
missed
missed
misses
missing
mistrusted
misuse
mitigates
mitigating
mix
mixed
mixing
mobilized
mobilizing
mock
mocking
modernizing
modify
modified
modifying
moisten
mold
molded
molding
mollify
mop
mopped
mopping
motivated
motivates
motivating
mount
mounted
mounted
mounting
mounts
mourn
mourned
mourning
move
moved
moved
moves
moving
multiply
multiplied
multiplied
multiplies
multiplying
mumbled
murder
murdered
murdering
murmured
murmuring
mused
muster
muttered
muttering
nailed
nailed
name
named
named
naming
narrowed
narrowed
narrowing
narrows
near
neared
nearing
need
needed
needed
needing
needs
negate
neglect
neglected
neglected
neglecting
negotiate
negotiated
negotiated
negotiating
nested
nested
nesting
nod
nodded
nodding
nominate
nominated
note
noted
noted
noting
notes
notice
noticed
noticed
noticing
notify
nudged
numbered
numbered
numbering
nurture
obey
obeyed

obeyed
obeying
object
objected
objects
obliged
obliged
observe
observed
observed
observes
observing
obsessed
obtain
obtained
obtained
obtaining
obtrudes
occupy
occupied
occupied
occupies
occupying
occur
occurred
occurred
occurring
occurs
offend
offended
offer
offered
offered
offering
offers
officiated
offset
offset
omits
omitted
omitting
oozed
open
opened
opened
opening
opens
operate
operated
operated
operates
operating
oppose
opposed
opposed
opposes
opposing
opted
ordain
ordained
order
ordered
ordered
organize
organized
organized
organizing
oriented
orienting
oust
outdistanced
outdo
outface
outgrow
outlawed
outline
outlined
outraged
outrun
outweigh
outweighed
outweighed
overcome
overcomes
overcoming
overcame
overcome
overflowed
overheard
overheard
overlap
overlapped
overlapping
overload
overlook
overlooked
overlooking
overlooks
overreach
overreached
overtake
overthrow
overthrown
owe
owed
owed
owes
owing
own
owned
owned
owns
pace
paced
pacing
pacify
pack

packed
packed
packing
packaged
packaging
padded
padding
paint
painted
painted
painting
paints
panic
parallel
paralleled
pardon
pardoned
pare
parked
parked
parking
part
parted
parting
partakes
pass
passed
passed
passes
passing
patrol
patrolling
patronized
patronizing
patted
patting
pause
paused
paused
pausing
pave
paved
paving
pay
paying
pays
paid
paid
pecked
peel
peeled
peeled
peeling
peered
peering
penetrate
penetrated
penetrated
people
perceive
perceived
perceived
perceives
perfected
perfecting
perform
performed
performed
performing
performs
perish
permeated
permeates
permit
permits
permitted
permitted
permitting
persist
persisted
persisted
persists
persuade
persuaded
persuaded
persuading
pertain
pertaining
pertains
pervades
pervading
petitioned
petting
phone
phoned
photograph
photographed
photographing
photographs
phrased
phrasing
pick
picked
picked
picking
picks
picture
pictured
pierced
piercing
pile
piled
piled
piling
pillage
pin
pinned

pinch
pinched
pinched
pinching
pinpoint
pioneered
pioneering
pitched
pitched
pitching
pitied
place
placed
placed
places
placing
plague
plagued
plagued
plan
planned
planned
planning
plans
plant
planted
planted
planting
play
played
played
playing
plays
plead
pleaded
pleading
please
pleased
pleased
pleases
pledged
plot
plotted
plow
plowed
plowing
plucked
plug
plugged
plumped
plunged
plunged
plunging
point
pointed
pointed
pointing
points
poised
poised
poisoned
poisoned
poked
pokes
poking
polish
polished
polished
polishing
pondered
pondering
pool
pooling
pop
popped
popping
portray
portrayed
portrays
pose
posed
posed
posing
position
possess
possessed
possessed
possesses
possessing
posted
posted
postpone
postponed
postponing
postulate
postulated
pound
pounded
pounding
pour
poured
poured
pouring
pours
power
powered
practice
practiced
practicing
praise
praised
praised
praises
praising
pray
prayed
prayed

praying
preach
preached
preaching
precede
preceded
preceded
preceding
preclude
predict
predicted
predicted
predicting
predicts
prefer
preferred
preferred
prefers
prejudiced
prepare
prepared
prepared
prepares
preparing
prescribe
prescribed
prescribed
present
presented
presented
presenting
presents
preserve
preserved
preserves
preserving
preside
presiding
press
pressed
pressed
presses
pressing
presume
presumed
presumes
presuming
presupposes
pretend
pretended
pretended
pretending
pretends
prevail
prevailed
prevailing
prevails
prevent
prevented
prevented
preventing
prevents
priced
pricing
prides
print
printed
printed
printing
probe
probed
probing
proceed
proceeded
proceeded
proceeding
proceeds
processed
processing
proclaim
proclaimed
proclaimed
proclaiming
proclaims
procure
procured
produce
produced
produced
produces
producing
profess
professed
professing
proffered
profit
profited
program
programed
programing
progress
progressed
progressed
progresses
prohibited
prohibited
prohibiting
project
projected
projecting
projects
prolonged
promise
promised
promised
promises
promising

promote
promoted
promotes
promoting
prompt
prompted
prompted
prompts
pronounce
pronounced
pronounced
propped
propel
propose
proposed
proposed
proposes
proposing
prosecute
prosecuted
prosecuting
prosper
protect
protected
protected
protecting
protects
protest
protested
protested
protesting
protests
protruded
protruding
prove
proved
proved
proves
proving
provide
provided
provided
provides
providing
provoke
provoked
provokes
pry
publicized
publicizing
publish
published
published
publishes
publishing
puffed
puffing
pull
pulled
pulled
pulling
pulls
pumped
pumping
punish
punished
purchase
purchased
purchased
purchases
purchasing
purged
purify
purified
purport
purported
purporting
purports
pursed
pursue
pursued
pursued
pursues
pursuing
push
pushed
pushed
pushes
pushing
put
puts
putting
put
put
puzzle
puzzled
puzzled
qualify
qualified
qualified
qualifies
quarrel
quarreled
quarreled
quarreling
quell
question
questioned
questioned
questioning
questions
quiet
quieted
quit
quit
quitting
quote

quoted
quoted
quotes
quoting
race
raced
racing
radiated
radiated
raged
raging
rained
raining
raise
raised
raised
raises
raising
raked
rally
ramble
rammed
range
ranged
ranged
ranges
ranging
ranked
ranking
ransack
ransacked
rape
raped
rapped
rapped
rapping
rate
rated
rates
rating
rationed
rationalize
rattling
reach
reached
reached
reaches
reaching
react
reacted
reacting
read
reading
reads
read
read
readjust
ready
realize
realized
realized
realizes
realizing
reap
rear
reared
reared
rearrange
reason
reasoned
reasoning
reassemble
reassured
reassured
reassuring
rebel
rebelled
rebelled
rebelling
rebuild
rebuilding
rebuilt
rebuke
rebut
recall
recalled
recalled
recalling
recalls
recapture
recede
receding
receive
received
received
receives
receiving
recite
reckon
reckoned
reclaim
reclaimed
recognize
recognized
recognized
recognizes
recognizing
recommend
recommended
recommended
recommending
recommends
reconcile
reconciled
record
recorded
recorded

recording
records
recount
recounted
recounting
recounts
recover
recovered
recovered
recovering
recruit
recruited
recruiting
recur
recurring
redeem
reduce
reduced
reduced
reduces
reducing
refer
referred
referred
referring
refers
refill
refine
refined
refining
reflect
reflected
reflected
reflecting
reflects
reform
reformed
refrain
refuse
refused
refused
refuses
refusing
refuted
regained
regaining
regard
regarded
regarded
regarding
regards
register
registered
registered
registering
regret
regrets
regretted
regretted
regulate
regulated
regulated
regulating
reinforce
reinforced
reinforces
reject
rejected
rejected
rejecting
rejects
relate
related
related
relates
relating
relax
relaxed
relaxed
relaxes
relaxing
release
released
released
releases
relieve
relieved
relieved
relinquish
relinquished
relinquishing
relish
relive
rely
relying
relied
relied
relies
remain
remained
remained
remaining
remains
remake
remark
remarked
remarked
remarking
remarks
remedy
remember
remembered
remembered
remembering
remembers
remind
reminded

reminded
reminding
reminds
remove
removed
removed
removes
removing
renamed
render
rendered
rendered
rendering
renders
renew
renewed
renewed
rent
rented
renting
repay
repaid
repair
repaired
repeat
repeated
repeated
repeating
repeats
repel
repelled
repent
replace
replaced
replaced
replaces
replacing
replenish
replenished
reply
replied
replied
replies
report
reported
reported
reporting
reports
represent
represented
represented
representing
represents
reproduce
reproduced
reproduces
reputed
request
requested
requested
requesting
requests
require
required
required
requires
requiring
rescind
rescue
rescued
resemble
resembled
resembles
resembling
resent
resented
resented
reserve
reserved
reserved
reserving
reside
resided
resides
residing
resign
resigned
resigned
resist
resisted
resisted
resisting
resolve
resolved
resolved
resolves
resolving
resort
resorted
resorting
respect
respected
respected
respecting
respects
respond
responded
responded
responding
responds
rest
rested
rested
resting
rests
restore
restored

restored
restoring
restrain
restrained
restraining
restrict
restricted
restricting
restricts
result
resulted
resulted
resulting
results
resume
resumed
resumed
resuming
retain
retained
retained
retaining
retains
retard
retarded
retire
retired
retired
retiring
retort
retorted
retreat
retreated
retreated
retreating
retrieve
retrieved
return
returned
returned
returning
returns
reveal
revealed
revealed
revealing
reveals
reverse
reversed
reverses
reversing
revert
reverted
review
reviewed
reviewed
reviewing
reviews
revise
revised
revised
revive
revived
revived
reviving
revolved
revolving
reward
rewarded
rid
ridding
ride
ride
rides
riding
rode
ridden
ridicule
ridiculed
ring
ringing
rang
ring
ringed
rip
ripped
ripping
rise
rises
risen
rose
rising
risk
risked
risked
rivaled
roam
roaming
roared
roaring
roast
roasted
robbed
robbed
rock
rocked
rocking
roll
rolled
rolled
rolling
root
rooted
rot
rots
rotting
rotated

rotates
rotating
round
rounded
rounded
rounding
rouse
roused
row
rowed
row
rub
rubbed
rubbed
rubbing
ruin
ruined
rule
ruled
ruled
rules
ruling
rumbling
run
running
runs
run
ran
rush
rushed
rushed
rushes
rushing
rustle
rustling
sacrifice
sacrificed
sacrificing
safeguard
sag
sagged
sagging
sail
sailed
sailed
sailing
salted
salting
salvage
salvaging
sampled
sampling
sanction
sanctioned
sanctions
satisfy
satisfied
satisfied
satisfies
satisfying
save
saved
saved
saves
saving
savored
savoring
saw
say
saying
says
said
said
scan
scanned
scanning
scandalized
scared
scared
scattered
scattered
schedule
scheduled
scheduled
school
scooped
scooted
score
scored
scored
scoring
scoured
scouring
scowled
scrambled
scrape
scraped
scraped
scraping
scratch
scratched
scratched
scratching
scrawled
scrawled
scream
screamed
screamed
screaming
screeched
screeching
screen
scrub
scrubbing
scrutinizing
scurried
seal

sealed
sealing
searing
search
searched
searched
searches
searching
seated
seating
secede
seceded
secure
secured
securing
see
seeing
sees
seen
saw
seek
seeking
seeks
sought
sought
seem
seemed
seemed
seeming
seems
seep
seeping
seize
seized
seized
select
selected
selected
selecting
selects
sell
selling
sells
sold
sold
send
sending
sends
sent
sent
sense
sensed
senses
sensing
separate
separated
separated
separates
separating
serve
served
served
serves
serving
service
servicing
set
sets
setting
set
set
settle
settled
settled
settles
settling
sever
severed
sew
sewing
shade
shaded
shading
shadow
shadowed
shadowing
shake
shakes
shaking
shaken
shook
shape
shaped
shaped
shapes
shaping
share
shared
shared
shares
sharing
shattered
shattered
shattering
shave
shaved
shaved
shaving
shearing
shedding
sheds
shed
shed
shield
shielded
shift
shifted

shifted
shifting
shifts
shimmy
shines
shining
shone
ship
shipped
shipping
shivered
shivering
shocked
shocked
shoot
shooting
shot
shot
shop
shopping
shorten
shortened
shouldered
shout
shouted
shouted
shouting
shouts
shoved
shoving
show
showing
shows
showed
showed
shower
showered
shred
shrieked
shrilled
shrink
shrinking
shrugged
shudder
shuddered
shuddering
shuffled
shuffling
shuns
shut
shut
shut
shy
shied
sickened
sighed
sighing
sighted
sighting
sign
signed
signed
signing
signs
signal
signaled
signaling
signify
silenced
silenced
simmer
simplify
simplified
simplifies
simulate
simulated
sin
sinned
sing
sing
singing
sings
sang
sung
singled
singled
sink
sinking
sank
sunk
sipped
sipping
sit
sits
sitting
sat
sat
size
sized
sketched
sketching
skidded
skidding
skimmed
skimmed
skimming
skip
skipped
skipped
skipping
skirt
slackened
slackening
slammed
slammed
slamming
slanting

slap
slapped
slapped
slapping
slash
slashed
slashed
slashing
sleep
sleeping
slept
slept
slice
sliced
slide
sliding
slid
slide
slip
slipped
slipped
slipping
slit
slow
slowed
slowed
slowing
slug
slugged
slugging
slump
slumped
slumped
smack
smacked
smash
smashed
smashed
smell
smelled
smelled
smelling
smells
smile
smiled
smiled
smiles
smiling
smoke
smoked
smoked
smoking
smoldering
smooth
smoothed
smoothed
smoothing
smothered
smothered
snag
snaked
snap
snapped
snapped
snapping
snarled
snarling
snatch
snatched
snatched
sneaked
sneaking
snickered
sniff
sniffed
snorted
snowed
snowing
snows
snuggled
soak
soaked
soaking
soared
soaring
sobered
sobering
soften
softened
softened
softening
soil
soiled
solder
solve
solved
solves
solving
soothed
soothing
sort
sorted
sound
sounded
sounded
sounds
sow
sown
spaced
spacing
span
spanned
spans
spare
spared
spared
sparked

sparkling
sparks
speak
speaking
speaks
spoke
spoken
spear
specialize
specialized
specialized
specializing
specify
specified
specifies
specifying
speculate
speculated
speculating
speed
speeding
sped
spell
spelled
spend
spending
spends
spent
spent
spice
spiced
spilled
spilling
spills
spin
spinning
spun
spit
spitting
spat
splashed
splashing
splitting
split
split
spoil
spoiled
sponged
sponsor
sponsored
sponsored
sponsoring
sponsors
spot
spotted
spotted
spotting
spouted
sprawled
sprawled
sprawling
spray
sprayed
sprayed
spraying
spread
spreading
spreads
spread
spread
spring
sprang
sprung
sprinkle
sprinkled
sprinkling
sprinted
sprouted
sprouting
spur
spurred
spurred
square
squared
squatted
squeaked
squeeze
squeezed
squeezed
squeezing
squinted
squinting
stabilize
stabilizing
stacked
stacking
staffed
stage
staged
staged
staging
stagger
staggered
staggered
staggering
stain
stained
stained
staining
stake
stalked
stalking
stalled
stammered
stamp
stamped
stamped

stamping
stampede
stand
standing
stands
stood
stood
stare
stared
stared
staring
starred
starring
start
started
started
starts
starting
startled
startled
starved
starving
state
stated
stated
states
stating
stave
stay
stayed
stayed
staying
stays
steal
stealing
stole
stolen
steamed
steaming
steer
steered
steering
stem
stemmed
stems
step
stepped
stepped
stepping
stick
sticking
sticks
stuck
stuck
stiffened
stiffens
stifle
stifling
stimulate
stimulated
stimulated
stimulates
stimulating
sting
stipulate
stir
stirred
stirred
stirring
stirs
stock
stoop
stooped
stooping
stop
stopped
stopped
stopping
stops
store
stored
storing
stormed
straggle
straighten
straightened
straightened
straightening
strained
strained
straining
stray
streaked
streaming
strengthen
strengthened
strengthening
strengthens
stress
stressed
stressed
stresses
stressing
stretch
stretched
stretched
stretches
stretching
stride
strode
strike
strikes
striking
stricken
struck
strung
strip

stripped
stripped
strive
strives
striving
strove
stroked
strolled
strolling
struggle
struggled
struggling
study
studying
studied
studied
stuff
stuffed
stumbled
stumbled
stumbling
stunned
subdue
subdued
submit
submits
submitted
submitted
submitting
subsided
subsidize
subsidized
substitute
substituted
substituted
substituting
subtracted
subtracting
succeed

succeeded
succeeded
succeeding
succeeds
suck
sucked
sucking
sue
sued
sued
suffer
suffered
suffered
suffering
suffers
suffice
suffused
suggest
suggested
suggested
suggesting
suggests
suit
suited
suited
sulked
sums
summed
summed
summarize
summarized
summarizing
summate
summon
summoned
summoned
supervise
supervised
supervising

supplant
supplement
supplemented
supplemented
supplementing
supply
supplying
supplied
supplied
supplies
support
supported
supported
supporting
supports
suppose
supposed
supposed
suppress
suppressed
surged
surging
surprise
surprised
surprised
surprising
surrender
surrendered
surrendering
surround
surrounded
surrounded
surrounding
survey
surveyed
surveying
survive
survived
survived

surviving
suspect
suspected
suspected
suspecting
suspects
suspend
suspended
sustain
sustained
swaggered
swallow
swallowed
swallowed
swallowing
swarmed
swarming
swayed
swayed
swaying
swear
swore
sworn
sweep
sweeping
swell
swelled
swelling
swollen
sweep
swept
swept
swerved
swerving
swim
swimming
swam
swing
swinging
swung
swung
swirled
swirling
swished
switch
switched
switched
switching
swooped
swooping
symbolize
symbolized
symbolized
symbolizes
sympathize
tackle
take
takes
taking
taken
took
talk
talked
talked
talking
talks
tally
tap
tapped
tapped
tapping
tapered
tapering
taste
tasted
tasted
tastes
tasting
tax
taxed
taxing
teach
teaches
teaching
taught
taught
teamed
tear
tearing
tore
torn
tease
telegraphed
telephone
telephoned
telephoned
telephoning
tell
telling
tells
told
told
tempted
tempting
tend
tended
tended
tending
tends
term
termed
termed
terms
terminate
terminated
terrified
terrifies

terrifying
test
tested
tested
testing
testify
testified
testified
testifies
thank
thanked
thanking
thaw
thawed
thawing
theorize
thickened
thickened
thin
think
thinking
thinks
thought
thought
threaten
threatened
threatened
threatening
threatens
thrilled
thrived
throbbed
throbbing
throw
throwing
throws
threw
thrown
thrust
thrusting
thrust
thrust
thunder
thwart
thwarted
tick
ticked
tie
tied
tied
ties
tying
tighten
tightened
tightening
tilt
tilted
tilted
tilts
timed
timing
tip
tipped
tired
tired
tiring
toast
toe
tolerate
tolerated
top
topped
topped
tormented
tormenting
toss
tossed
tossed
tossing
total
totaled
totaling
totals
touch
touched
touched
touches
touching
toured
touring
trace
traced
tracing
track
tracked
tracking
trade
traded
traded
trading
trail
trailed
trailed
trailing
train
trained
trained
training
tramped
trample
transact
transcending
transcends
transfer
transferred
transferred
transform

transformed
transformed
transforming
transforms
translate
translated
translating
transmit
transmitted
transpired
transpiring
transport
transported
transporting
trapped
trapping
travel
traveled
traveled
traveling
traverse
traversed
traversed
tread
treat
treated
treated
treating
treats
tremble
trembled
trembling
trim
trimmed
tripped
trot
trotted
trotted
trouble
troubled
troubled
troubling
trudged
trust
trusted
trusted
trusting
trusts
try
trying
tried
tried
tries
tucked
tucked
tucking
tumbled
tumbled
tumbling
tune
tuned
turn
turned
turned
turning
turns
twined
twist
twisted
twisted
twisting
twitched
twitching
type
typed
typing
uncover
uncovered
undergo
undergoes
undergoing
underwent
underlie
underlying
underline
underlined
undermine
underscored
understand
understanding
understands
understood
understood
undertake
undertakes
undertaken
underwrite
undo
undone
undressed
undressing
unfolded
unfolded
unfolding
unfolds
unify
unified
unifies
unifying
unite
united
united
uniting
unload
unloaded
unloading
unlock

unlocked
unlocked
unlocks
untie
upgrade
uphold
upholding
upheld
upsets
upset
upset
urge
urged
urged
urges
urging
use
used
used
uses
using
ushered
utilize
utilized
utilizes
utilizing
utter
uttered
valued
values
vanish
vanished
vanished
vanishing
vary
varied
varied
varies
varying

veered
vent
venture
ventured
ventured
verify
verified
vexed
vexing
view
viewed
viewing
views
violate
violated
violates
violating
visit
visited
visited
visiting
visits
visualize
voiced
volunteer
volunteered
volunteering
vote
voted
voted
votes
voting
vowed
vowing
wadded
wade
waded
wagged
wagging

wage
waged
wager
wailed
wailing
wait
waited
waited
waiting
waits
wake
wakes
waking
woke
walk
walked
walked
walking
walks
wall
wander
wandered
wandering
want
wanted
wanted
wanting
wants
warm
warmed
warmed
warming
warn
warned
warned
warning
warns
warped
warping

warrant
wash
washed
washed
washing
waste
wasted
wasted
wasting
watch
watched
watched
watching
watered
watered
watering
wave
waved
waving
waxed
weaken
weakened
wear
wearing
wears
wore
worn
weave
weaving
wove
woven
weep
weeping
wept
weigh
weighed
weighed
weighing
weighs
welcome
welcomed
welcomed
welcoming
well
wet
wetting
wet
whacked
wheeled
wheeled
while
whine
whining
whip
whipped
whipped
whipping
whirled
whirling
whisper
whispered
whispered
whispering
whistled
whistling
whizzed
widen
widened
widened
wiggled
wiggling
willed
willed
win
wins
winning
winced
wind
winding
winds
wound
wound
wind
winded
wink
winked
winking
wipe
wiped
wiped
wiping
wired
wired
wish
wished
wished
wishes
wishing
withdraw
withdrawing
withdrawn
withdrew
wither
withhold
withheld
withholding
withstand
withstood
witness
witnessed
witnessed
witnessing
wobble
wobbled
wonder
wondered
wondered

wondering
wonders
work
worked
worked
working
works
worry
worried
worried
worries
worrying
worship
worshiping
wound
wounded
wrangled
wrap
wrapped
wrapped
wrapping
wreck
wrecked
wrecked
wrecking
wrenched
wrestle
wring
wrinkled
wrinkled
write
writes
writing
written
wrote
writhe
writhing
yanked
yearned
yell
yelled
yelling
yield
yielded
yielded
yielding
yields

B. APPENDIX 4: REGULAR TEST DATASET

abound
accord
ache
acknowledges
acquires
activate
adjourned
advocated
affirmed
affirming
aided
alerted
alienate
allege
alleged
allot
amend
analyzes
annoy
appeased
applauding
appropriate
arch
arched
arraigned
arresting
ascend
ascended
ascribe
astound
attested
attribute
avenge
avert
babbled
ban
battering
beckon
betrayed
bind
birdie
blamed
blasting
blind
blister
blot
blurted
blushed
boasts
bogeyed
borders
box
boxed
brag
bragged
breathes
bribe
brightened
bristle
broadens
broil
bruise
budgeted
bumping
burns
buzz
bypass
cable
calming
can
captured
caress
case
caution
ceasing
chance
channel
chant
charmed
chart
chat
cherishing
chin
chinning
chop
circle
circulated
clatter
clench
clinch
clipped
clogging
clutch
coddled
coerce
coin
coined
collapse
comfort
commence
commutes
compensating
competed
compile
complement
compound
compress
conceiving
concurred
condense
condition

congealed
congested
congregate
conjure
consented
consisted
conspire
contacted
contacted
contacting
contracting
contradicts
contrive
converted
convict
cooked
copy
correlating
corrupt
corrupted
counsel
countered
covet
crave
credited
cringed
crooned
crown
crumbled
crush
culminate
culminating
cup
cupped
curling
curtailed
cushion
cushioning
damaged
dances
dangle
dangled
darkened
dash
dazzle
dazzled
debate
deceive
decompose
decomposing
decreased
deduced
deem
delegating
deluded
denoted
departs
deplore
deplored
detach
detest
devastate
diffuse
dilate
dined
disable
disarm
discipline
disdain
dismounting
dispatch
dispensed
disrupted
distract
distributed
distributes
divert
divorced
dodged
dominating
dot
draft
drafted
dumped
eats
edge
embody
emit
enacted
engulfed
enslave
ensuring
entreat
entreated
entrust
equating
erase
erased
escorting
esteemed
estimating
evolved
evolving
excited
expel
expended
expire
fade
fans
father
filter
flag
flanked
flatten
flattening
flatter
flattered

flaunted
flee
flex
flicker
flickered
fling
flock
flocked
flog
flower
flowered
foam
foaming
foretell
forfeit
fork
forsake
fortified
foul
found
freed
frequent
frown
furthered
fuss
gang
gasp
glare
glazing
gleam
glide
glimpse
glisten
glue
gobbled
gouge
grimaced
grin
grind
grip
grope
ground
grunt
hailed
halted
halting
hammer
harass
harvest
heads
heave
hindered
hint
hissed
hook
hop
hum
hunted
hustle
illumine
imitated
impressing
incited
infer
influencing
infuriate
institute
interact
interests
interferes
intervene
iodinate
iron
jabbed
jerk
joins
journey
knitted
label
lapsed
lashing
launches
launching
lecture
level
license
lightened
lining
listens
lodge
loom
lounge
lug
lunge
lurk
manning
maneuver
masquerade
memorized
menaced
mend
merges
metering
milks
minded
mirror
mitigate
moan
moaned
mobilize
mocked
modernize
modernized
modifies
mopped
motivate
murmur

mutter
neck
nest
nodded
notified
notified
nudge
number
obsesses
officiate
omit
ooze
ordering
outlining
outnumber
outrage
overflowing
overlooked
package
pad
paralyze
paralyzed
paralyzes
park
parodied
parted
partake
patronize
pat
peck
peer
penetrating
permeate
persisting
petition
petted
phoned
phones
phrase
pictured
picturing
pilot
pioneer
pitch
pity
pledged
plied
plotted
plotting
pluck
plunge
poison
poisoning
poke
ponder
pops
portraying
post
practiced
preached
prejudice
preserved
presumed
presuppose
price
pride
process
progressing
prohibit
projected
prolong
prolonging
prop
protrude
provoked
puff
puffed
pump
purge
rack
radiate
rain
rained
rake
ram
ranked
rattled
reassure
reciting
recurred
reeled
refute
regain
reinforcing
rejoin
releasing
relieves
repaired
reproducing
rescuing
rethink
retires
revolve
ridiculing
rival
roar
rob
robbing
rolls
rooting
rotate
rotated
rumbled
salute
saluted
sample
scare

schooling
scoop
scour
scowling
scrutinized
sear
seat
seated
seceding
shatter
shear
shed
shine
shiver
shock
shoulder
shove
shriek
shrinks
shuffle
shun
sickening
sidle
sidled
sifted
sigh
sight
sighted
silence
single
sizzle
sizzled
sizzling
skate
skating
sketch
sketched
sketches
slant
slanted
smoldered
sneak
sneaked
sniffing
snort
snow
soothe
sounding
space
spark
spells
spill
split
sponge
sprawl
springing
sprout
staff
stalling
startle
starve
steady
steadied
still
stinging
stink
stipulates
storm
straggling
strain
stream
streamed
string
stroke
strut
strutted
stuffing
stumble
subside
substantiate
subtract
suffuse
sum
supervises
surge
surveyed
sustaining
swap
swarm
sway
swearing
swears
swerve
symbolizing
taper
teased
teasing
telegraph
tempt
thicken
thrash
thrashed
thrill
thrive
thundered
thundering
time
timed
tire
toasting
topping
tour
traced
transcend
trip
tripping
tuck

tug
tugged
tumble
underlining
undermined
underscore
unfold
upset
usher
uttered
uttering
validate
value
veer
veering
vex
viewed
violated
volunteered
wanders
warranted
water
waver
wax
whinnied
whirl
whistle
wield
wielded
will
wing
winged
withered
yank
yearn

B. APPENDIX 5: IRREGULAR TEST DATASET

befell
bet
bid
bid
bled
clung
forbade
knelt
misled
overtaken
overtook
rung
shrunken
slung
slung
spun
stung
stung
strung
striven
swum
undid
upheld
wept
wet
withheld
withstood

F. APPENDIX 6: PSEUDO WORDS TEST

blafe
blip
brilth
brip
cleed
cleef
cloe
flape
foa
frilg
fring
frink
froe
gleef
glinth
glip
gloke
goav
grare
greem
jare
joam
keeb
krilg
meep
nace
ning
nist
plaonth
plare
pleem
plimph
plip
ploab
ploag
ploamph
preed
preek
proke
quare
queed
queef
skring
slace
smaib
smaig
smairg
smairph
smeeb
smeeg
smeej
smeelth
smeenth
smeerg
spling
spring
treem
trilb
trisp
voa

G. APPENDIX 7: CODE TO RUN THE BACK PROPAGATION SIMULATIONS.

7.1 Code to run the eight class classification task

```
input=dlmread('class_8_input.txt');
output=dlmread('class_8_ouputoutput.txt');
net=patternnet([50 50]);
net=train(net,input',output');
testinput=dlmread('class_8_inputreg_test.txt');
testinput=testinput(1:614,1:433);
y=sim(net,testinput');
y=y-0.5;
y=ceil(y);
testoutput=dlmread('class_8_outputreg_test.txt');
outputmatrix=zeros(614,1);
for t=1:614
        error_t=abs(y'-testoutput);
        if error_t==0;
                outputmatrix(t)=1;
        else
                outputmatrix(t)=0;
        end
end
sum(outputmatrix)


testinput_irreg=dlmread('class_8_inputirreg_test.txt');
yireg=sim(net,testinput_irreg');
yireg=yireg-0.5;
yireg=ceil(yireg);
testoutput_irreg=dlmread('class_8_outputirreg_test.txt');
outputmatrix_irreg=zeros(27,1);
for t=1:27
        error_t1=abs(yireg'-testoutput_irreg);
        if error_t1==0;
                outputmatrix_irreg(t)=1;
        else
                outputmatrix_irreg(t)=0;
        end
end
sum(outputmatrix_irreg)
```

7.2 Code to run the four class classification task

```
input=dlmread('class_4_input.txt');
output=dlmread('class_4_output.txt');
net=patternnet([50 50]);
net=train(net,input',output');
testinput=dlmread('class_4_inputreg_test.txt');
testinput=testinput(1:478,1:433);
y=sim(net,testinput');
y=y-0.5;
y=ceil(y);
testoutput=dlmread('class_4_outputreg_test.txt');
outputmatrix=zeros(478,1);
for t=1:478
        error_t=abs(y'-testoutput);
        if error_t==0;
                outputmatrix(t)=1;
        else
                outputmatrix(t)=0;
        end
end
sum(outputmatrix)


testinput_irreg=dlmread('class_4_inputirreg_test.txt');
yirreg=sim(net,testinput_irreg');
yirreg=yirreg-0.5;
yirreg=ceil(yirreg);
testoutput_irreg=dlmread('class_4_outputirreg_test.txt');
outputmatrix_irreg=zeros(12,1);
for t=1:12
        error_t1=abs(yirreg'-testoutput_irreg);
        if error_t1==0;
                outputmatrix_irreg(t)=1;
        else
                outputmatrix_irreg(t)=0;
        end
end
sum(outputmatrix_irreg)
```

7.3 Code to run the three class classification task

```
input=dlmread('class_3_input.txt');
output=dlmread('class_3_output.txt');
net=patternnet([50 50]);
net=train(net,input',output');
testinput=dlmread('class_3_inputreg_test.txt');
testinput=testinput(1:478,1:433);
y=sim(net,testinput');
y=y-0.5;
y=ceil(y);
testoutput=dlmread('class_3_outputreg_test.txt');
outputmatrix=zeros(478,1);
for t=1:478
        error_t=abs(y'-testoutput);
        if error_t==0;
                outputmatrix(t)=1;
        else
                outputmatrix(t)=0;
        end
end
sum(outputmatrix)


testinput_irreg=dlmread('class_3_inputirreg_test.txt');
yirreg=sim(net,testinput_irreg');
yirreg=yirreg-0.5;
yirreg=ceil(yirreg);
testoutput_irreg=dlmread('class_3_outputirreg_test.txt');
outputmatrix_irreg=zeros(12,1);
for t=1:12
        error_t1=abs(yirreg'-testoutput_irreg);
        if error_t1==0;
                outputmatrix_irreg(t)=1;
        else
                outputmatrix_irreg(t)=0;
        end
end
sum(outputmatrix_irreg)
```

7.4 Code to run the generative simplified model

```
input=dlmread('present_input.txt');
output=dlmread('past_output.txt');
net=patternnet([200 200 200 200 200]);
net=train(net,input',output');
testinput=dlmread('present_inputreg_test.txt');
y=sim(net,testinput');
y=y-0.5;
y=ceil(y);
testoutput=dlmread('past_output_regtest.txt');
outputmatrix=zeros(336,1);
for t=1:336
	error_t=abs(y'-testoutput);
	if error_t==0;
		outputmatrix(t)=1;
	else
		outputmatrix(t)=0;
	end
end
sum(outputmatrix)


pseudotest_input=dlmread('pseudotest_input.txt');
ytest=sim(net,pseudotest_input');
ytest=ytest-0.5;
ytest=ceil(ytest);
pseudotest_output=dlmread('pseudotest_output.txt');
outputmatrix_pseudo=zeros(60,1);
for t=1:60
	error_t1=abs(ytest'-pseudotest_output);
	if error_t1==0;
		outputmatrix_pseudo(t)=1;
	else
		outputmatrix_pseudo(t)=0;
	end
end
sum(outputmatrix_pseudo)
```

H. APPENDIX 8: CODE TO RUN THE SOM NETWORK.

```
input=dlmread('class_8_input.txt');
net=selforgmap([8 8]);
net=train(net,input);
```