Learning of Past Tense using the Connectionist Approach

CS784
LANGUAGE ACQUISTION

Instructors: Dr. Achla M. Raina
Dr. Harish Karnick

Submitted by: Abhishek Kumar Mall
Varun Sharma

Acknowledgement

We take this opportunity to thank our instructors Dr. Achla M. Raina and Dr. Harish Karnick. Without their help and guidance it would be impossible to finish the project under the given constraints of time. The project required enough background on verbs and their past tense, phonemes and phonemic representation of the verbs in the simulation. Due to their continuous help, text and reference material were easily available to us and we did not waste time searching for the correct reference.

Contents

Introduction

Our simulation is based on modeling the verb and past tense mapping through the connectionist approach.
The connectionist approach is based on neural network theory which says that networks are made up of nodes and connections.
Nodes: The nodes can have any property or no property at all and are activated on the basis of threshold values. If the value is greater than the threshold then the node is activated otherwise it is dormant.
Connections: The connections are assigned random weights in the starting but as they repeatedly experience the pattern, their weights are adjusted to give correct output next time.

In a network there are various numbers of input, hidden and output nodes. Here we have used a feed-foreword network according to which in the same layer 2 nodes have no connection. All other nodes are connected to each other. The input nodes are connected to output nodes via hidden nodes. Depending on the active nodes in the input layer and the weights of the connections, we get the desired output. Various algorithms have been developed to adjust the weights to give the correct output.
The connectionist approach is contrary to the dominant symbolic approach to learning in the sense that there are no rules and exceptions posited in the model to account for an output for a given input. This approach has its drawbacks but the biggest advantage it has is that it is said to model the brain more closely than symbolic model approach.

The Simulation

Rumelhart and McClelland's Model

Various simulations have been done which show same learning behavior as humans do. One of the most famous simulation was done by J. L. Rumelhart and D. E. McClelland where they used the Wickelgren feature to represent the verbs. Wickelgren feature consists of a set of trigrams called Wickelphones and which are further reduced to phonemic representation called Wickelfeatures. Wickelgren features almost permits a differentiation of all root form of English and their past tense.
They used in total 460 nodes for the representation of each word. Their success ratio were close to 85% and were able to show the same U-shaped pattern of learning that the children show.

Our Model

Model 1: Initially we started with a model which consisted of 26 nodes for representing each verb. For the first utterance each letter was assigned a value of 0.5 and for the next utterance the value was increased by 0.5.
Since the values of the nodes could not go beyond 1, verbs having utterances of letters more than twice could not be represented by this model. Also this model lacked order and precedence of the verbs in the final presentation. You cannot infer from the output what the verb is unless you know the expected pattern.
This model also showed some results. When it was trained over only regular verbs or only irregular verbs, it showed some kind of learning. But when the patter was mixed up the results were not good.

Model 2: The second model we opted for consisted of 260 nodes for each verb. Among each 26 nodes, from the starting, only one node was active in accordance with the letter, in precedence order of the verbs. That is to say, that the first letter of the verb was showed active in the first set of 26 nodes and the rest 25 were inactive, the second letter was active in the next set of 26 verbs and so on.
The model had the advantage of order over the first model and words having at most 10 alphabets could be represented using this model.
The only problem with that model was the final matrix of 260Xword size was very scarce and the pattern recognition won't

be easy. But nothing could be concluded as the simulation requirements for the model were much more that the computational capacity of our computer.

Model 3 and the final model: In all the simulations for each of the model we have used, we used the neural network tools of MATLAB. We basically used the feed forward network along with the Bayesian back propagation algorithm. The reason for choosing the Bayesian algorithm was that it learns the pattern in only a few epochs, which is actually the case with child learning.
Coming to final model, we decided to have a phonemic representation of the verbs and their past tenses. The classification of phonemes was used exactly as it was done by Rumelhart and McClleland. The classification was as follows: -

		Front		Middle		Back	
		V/L	U/S	V/L	U/S	V/L	U/S
Interrupted Cons.	stop	b	p	d	t	g	k
	nasal	m	-	n	-	N	-
Continuous Cons.	fric.	v/D	f/T	z	s	Z/j	S/C
	liq/SV	w/l	-	r	-	y	-
Vowel	high	E	i	O	^	U	u
	low	A	e	I	a/@	W	*/o

KEY: N= ng in sing; D= th in the; T= th in with; Z= z in azure; S= sh in ship; C=ch in chip; E= ee in beer; i= i in bit; O= oa in boat; ^=u in but or schwa; U= oo in boot; u= oo in book; A= ai in bait; e= e in bet; a= a in bat; @= a in father; W= ow in cow; *= aw in saw; o= o in hot.

Here we have 2 categories of phonemes and each category has 3 more classification and each classification has 2 types. This makes it 6 on each side of the 2 major categories.
For each of the 6 types we decide to use 3 bits to represent it. Thus each phoneme can be represented in 6 bits.
Next for each verb, we took each of its phonemes and concatenated the phonemes in the order of precedence. We put a restriction on the word size to 8 letters keeping into account our capacity of our personal computers. That would make up to 48 nodes, i.e. maximum of 8 phonemes. The bits for each of the phoneme are as follows: -

stop	001
nasal	010
fric.	011
Liq/SV	100
high	101
low	110

F v/l	001
F u/s	010
M v/l	011
M v/l	100
B v/l	100
B u/s	110

Data Set

Our data set consisted of 110 verbs both regular and irregular.
Initial training was done on a set of 100 verbs which has all the regular verbs (around 75) and around 25 irregular verbs.
Then we made two test files one which had the 10 irregular verbs already shown to the network.
The other file had 10 remaining new verbs that both regular and irregular.
For the whole data set we manually calculated the coded format for each of the verbs, and using Inp_Prin.java we created the required format for the input to the simulation.
The input file was named Input.txt, target file (which had the past tense of verbs in input file), was named Output.txt. The test file was named as TestData.txt, its target file as TestOp.txt. The file which had 10 irregular verbs was named irregular.txt and its past tense file was named irregularout.txt.

Final Simulation

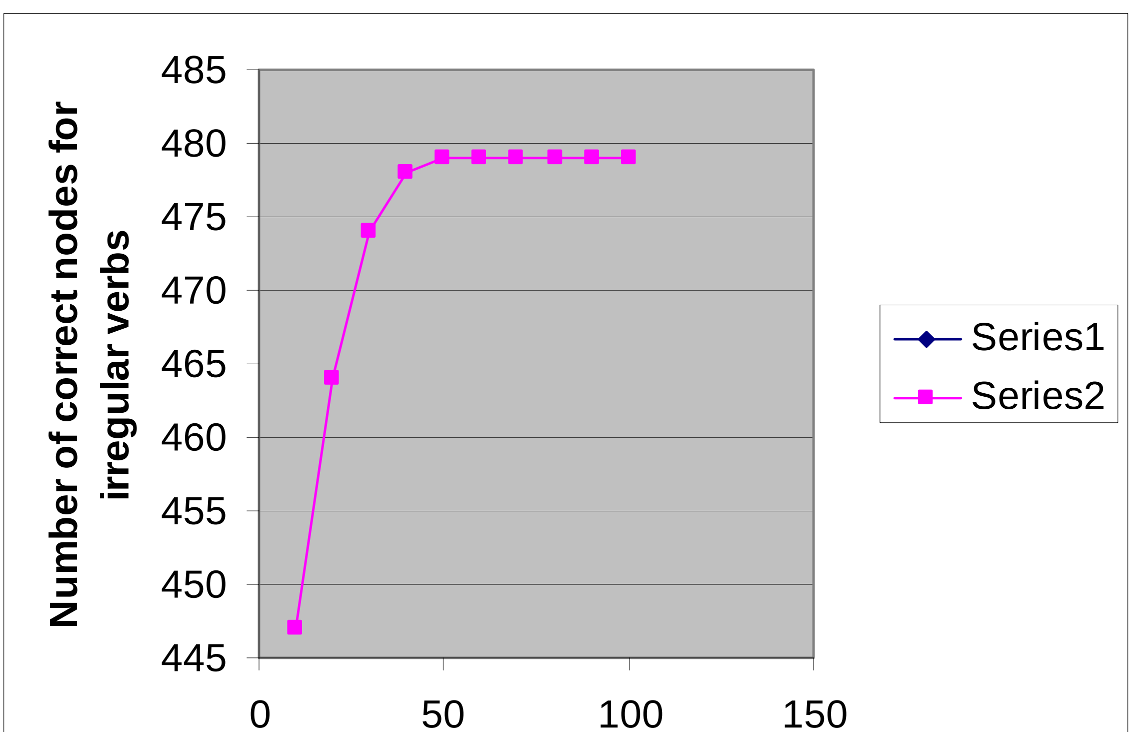
In the simulation we trained the network for 100 epochs. After every 10 epoch, we tested the data for the irregular verbs and the new set of verbs.

For testing the correctness, we made two programs, namely, Convert.java and test1.java. The first one converted the output file such that for values less than 0.5 the node value was made 0 and for values greater than equal to 0.5 it was made 1.

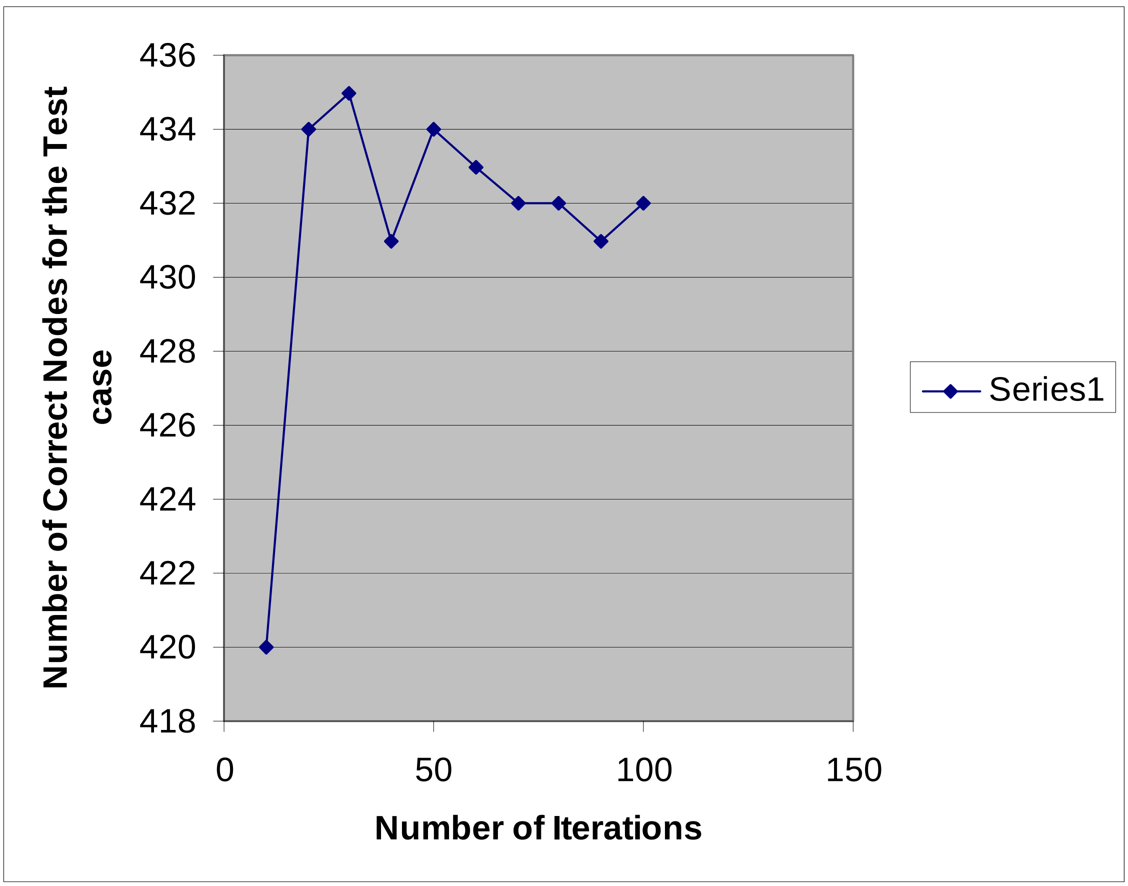
The second one, test1.java, compared the two output files we got after converting was compared to TestOp.txt and irregularout.txt.

The output files were named oui (for ith iteration) and iri.

The results we got have been presented in the form of two graphs. The first graph showed the learning of the irregular verbs and is as follows:

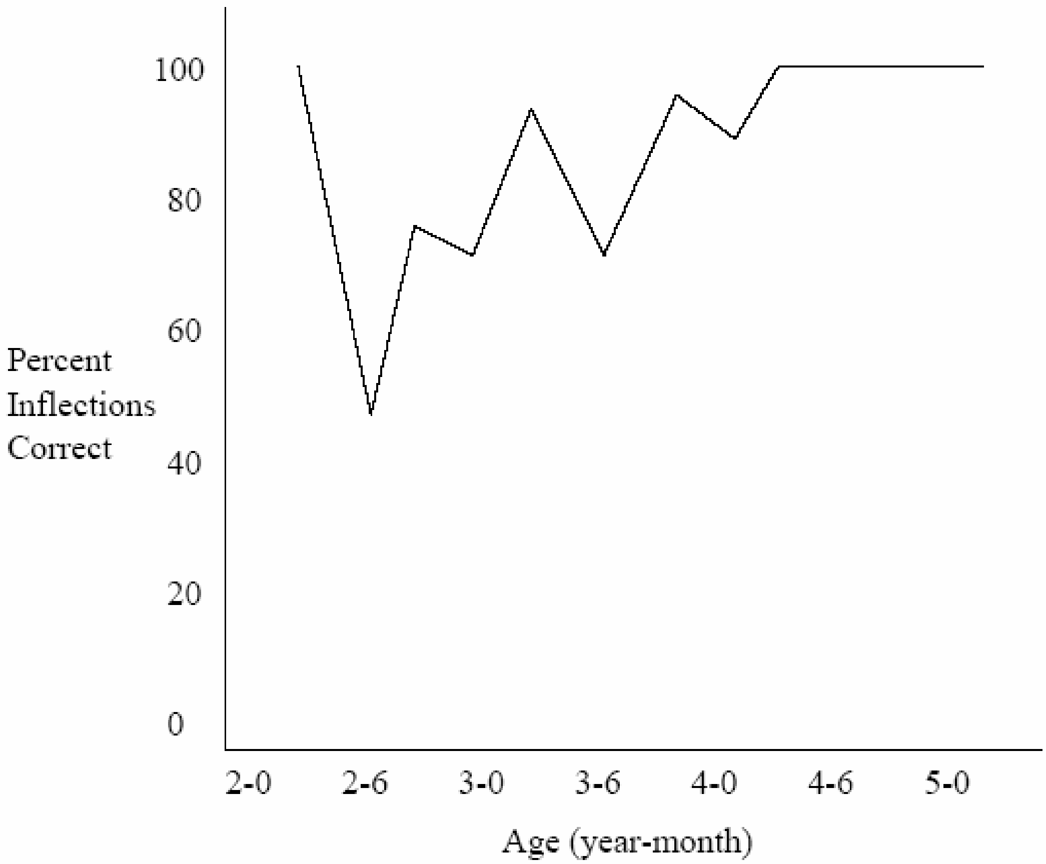


The second graph showed the results on new verbs and here is the graph that we got:



To compare the above graph we used a reference graph that we got from a link of a thesis done at MIT. The link is as follows:- http://genesis.csail.mit.edu/papers/Molnar.pdf

And the graph that they have given for overgeneralization was:


100
80
60
Percent
Inflections
Correct
40
20
0
2-0
2-6
3-0
3-6
4-0
4-6
5-0
Age (year-month)

Further Testing

We did further testing on our simulation to check the robustness of the system. For the created another program Randomize.java which randomly picked up 10 verbs out of the data set of 110 verbs and created 4 following files:

1) 100 remaining verbs
2) Past tense of those 100 verbs
3) 10 randomly picked up files
4) Past tense of those 10 verbs

We created this kind of data set 10 times and tested on each of them starting from a fresh network each time.
The output file was named wed.txt and its converted format (using Convert.java) was named tue.txt in each of the data format. The results we got have been tabulated in here:

S.No.	Name	Number of Nodes correct	Total number of nodes	Percentage correctness
1	TEST0	429	480	89.4
2	TEST1	444	480	92.5
3	TEST2	437	480	91
4	TEST3	384	480	80
5	TEST4	426	480	88.8
6	TEST5	437	480	91
7	TEST6	436	480	90.8
8	TEST7	438	480	91.2

9	TEST8	427	480	89
10	TEST9	424	480	88.3

The average percentage of node that were correct are: 89

Conclusion

- ☞ Our simulation has correctly learned the past tense of the given input.
- ☞ For the case when the input had new verbs the results were found to be correct up to 89%.
- ☞ From the 2nd graph in the final simulation we can see that it first over generalizes and then it is partially recovering which approximates the characteristic of child learning.
- ☞ On further testing our simulation shows a consistent character and average number of correct nodes in the testing was 89% again.

Disadvantages of our Model

- ☞ The word length is limited which deviates from the fact that children learn words of even larger lengths.
- ☞ Our error (SSE) reaches to a critical point (1-3%) after sometime and does not tend to minimize the error further.
- ☞ There may be possibility that two verbs may have same phonemic representation.

Bibliography

- McClelland, J.E. & Rumelhart, D.E. Learning the Past Tense. *Parallel Distributed Processing (Vol. 2),* 216-271.
- Pinker, S. & Prince, A. Analysis of Parallel Distribute Processing Model of Language Acquisition