Edinburgh University Technical Report – May 1994

Learning the Past Tense of English Verbs: Connectionism Fights Back

John A. Bullinaria

Neural Networks Research Group
Department of Psychology
University of Edinburgh
7 George Square
Edinburgh EH8 9JZ

Abstract

The ability to learn the past tense of English verbs has become a benchmark test for cognitive modelling. In a recent paper, Ling (1994) presented a detailed *head-to-head* comparison of the generalization abilities of a particular Artificial Neural Network (ANN) model and a general purpose Symbolic Pattern Associator (SPA). The conclusion was that 'the SPA generalizes the past tense of unseen verbs better than ANN models by a wide margin'. In this paper we show that this conclusion was based on comparisons with an uncharacteristically poorly performing ANN. A different ANN model is presented which not only out-performs the existing ANN models by a wide margin but also out-performs the SPA by a significant amount. We provide an explanation of how this happens and suggest several ways in which the model can be improved further.

This research was supported by the United Kingdom Joint Councils Initiative in Cognitive Science/HCI, Grant number SPG 9029590.

1. Introduction

Since Rumelhart & McClelland (1986) first suggested that the learning of language skills, such as the past tenses of English verbs, could be modelled better with Artificial Neural Networks (ANNs) than by systems involving symbolic processing, there have appeared numerous papers attempting to show that this is simply not true (e.g. Pinker & Prince, 1988; Lachter & Bever, 1988; Kim, Pinker, Prince & Prasada, 1991; Pinker, 1991; Ling & Marinov, 1993; Ling, 1994). There have also been numerous papers arguing that their ANN approach out-performs all previous approaches and answers the previous criticisms (e.g. Plunkett & Marchman, 1991; MacWhinney & Leinbach, 1991; MacWhinney, 1993). The latest in this string of claims and counter-claims (Ling, 1994) appears to demonstrate quite clearly that their symbolic approach performs significantly better than the best connectionist approach on the same past tense training and testing data.

The problem of learning the past tenses of English verbs is just one example of a whole class of mappings of the form:

input character string → *output character string*

in which the output character set can be the same or different to the input character set. Two well known problems of this class are reading aloud for which *letters* → *phonemes*, and spelling for which *phonemes* → *letters*. The past tense mapping may be either *letters* → *letters* or *phonemes* → *phonemes*, though for comparison with previous studies we shall be concentrating on the phoneme mapping here. Since neural networks are rather successful at reading and spelling (e.g. Bullinaria, 1994) and there is a close relationship between the various mappings it would be surprising if we were not able to find equally successful neural network systems for past tense learning.

Let us examine what is involved by considering a representative set of seven words from the past tense training data, namely:

1.	REG	bust	bustId	(bust)
2.	EXC	go	wEnt	(go)
3.	REG	bar	bard	(bar)
4.	REG	blak	blakt	(black)
5.	EXC	luz	lOst	(lose)
6.	EXC	tek	tUk	(take)
7.	EXC	st&nd	stUd	(stand)

in which we use the UNIBET phoneme representation system of MacWhinney (1990). First we note that there are two classes of words. The past tenses of regular words (denoted REG) are formed by adding the suffixes /Id/, /t/ or /d/ depending only on the final phoneme of the verb stem. Exception words (denoted EXC) do not follow these main rules, but may follow a sub-rule (e.g. 'take' and 'shake') or may be totally exceptional (e.g. 'go'). Neural networks are generally good at learning hierarchies of rules, sub-rules and exceptions so this is not a problem. This is why neural network models (e.g. Bullinaria, 1994) are so much better at reading aloud than statistical analogy models (e.g. Sullivan & Damper, 1992).

The next thing we note is that the mapping is highly redundant. For the regular words the mapping is a straightforward identity mapping for all the phonemes in the verb stem and this is also true for parts of many of the exception words as well. That the same phonemes in different word positions follow the same mapping is something that any system must recognise if it is to be efficient and generalize well. This is often referred to as the *recognition problem.* The second problem is that (even for the regular words) the lengths of the input strings do not match those of the corresponding output strings, nor is there any constant relation that holds for all words. This means that we have the problem of aligning the input strings with the output strings so that each input phoneme is trained to map to the right output phoneme. This is often referred to as the *alignment problem.*

The original connectionist approach of Rumelhart & McClelland (1986) solved these problems by splitting the input and output strings into triples of characters (known as Wickelfeatures). This representation alone attracted much criticism in the literature (e.g. Pinker & Prince, 1988) and the generalization performance was unacceptably poor. The same was true of the corresponding Wickelfeature model of reading (Seidenberg & McClelland, 1989). To avoid the limitations of the Wickelfeature approach, MacWhinney & Leinbach (1991) and MacWhinney (1993) used new input - output representations that employed a series of templates to overcome the alignment problem. It is these improved connectionist systems that were out-performed by the Symbolic Pattern Associator (SPA) of Ling & Marinov (1993) and Ling (1994).

Since the existing connectionist approaches with their templates have still to address the recognition problem, it is not surprising that their generalization performance is still rather poor. In this paper we shall show that, if we adopt the approach of a recent connectionist model of reading (Bullinaria, 1994), we can deal successfully with both the recognition and alignment problems and achieve a past tense learning system that generalizes better than any existing system.

2. The New Connectionist Model

The obvious way to solve the recognition problem is to have the system process the input string one character at a time and produce the output character that corresponds to the given input character in the context of the other characters in the input string. That is precisely how Sejnowski & Rosenberg (1987) proceeded for their NETtalk model of reading. The problem with their model, however, is that it requires the alignment problem to be solved by hand by pre-processing the training data and this is usually regarded as cheating. (The templates of MacWhinney & Leinbach might attract the same criticism.) Fortunately, it has recently been shown how NETtalk can be modified so that it can solve the alignment problem for itself (Bullinaria, 1993, 1994).

In Bullinaria (1994) numerous variations of the basic modified NETtalk reading model are discussed and the simulation results presented. In this paper we will describe the simplest corresponding past tense learning model and its results. A full analysis of all the possible variations will require many more simulations and will be presented elsewhere at a later date.

Both the original NETtalk and the modified version solve the alignment problem by inserting blanks into the output strings so that there is a one-to-one correspondence between the input and output characters. To make this work for the past tense training data we therefore need to add two suffix markers (arbitrarily ‘[]’) to the end of each word so that the length of the output string is never more than the corresponding input string. Then to get the alignment right we need to insert blank characters (i.e. phonemic nulls) into the output strings to give a one-to-one correspondence. It was the problem of having to insert these blanks by hand that hampered progress with this type of model in the past.

The new connectionist model consists of a standard fully connected feedforward network with one hidden layer arranged in the same way as the NETtalk model of Sejnowski & Rosenberg (1987). The input layer consists of a window of *nchar* sets of units with each set containing one unit for each different input character occurring in the training data (i.e. 36 phonemes plus two suffix markers). The output layer has a single set of units containing one unit for each different output character occurring in the training data (i.e. 36 phonemes plus the blank). The input strings slide through the *nchar* characters wide input window, starting with the first character of the string at the centre and ending with the final character at the centre. Each character that falls within the window activates a single unit in the appropriate set. The networks are then trained to activate the output unit that corresponds to the input character in the centre of the window using the standard back-propagation learning algorithm

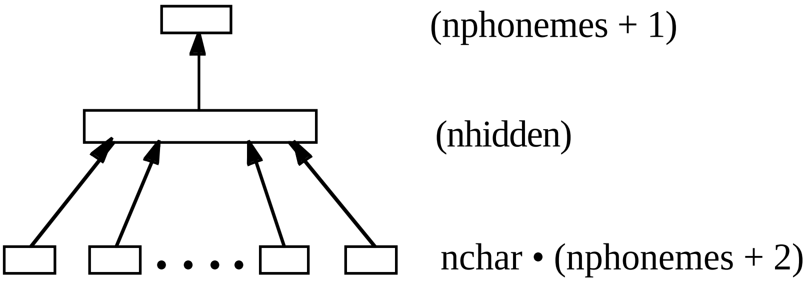

Figure 1. The NETtalk style ANN architecture.

of Rumelhart, Hinton & Williams (1986) with a sigmoid prime offset (Fahlman, 1988). The output of the network is currently simply taken to be the phoneme that corresponds to the output unit with the highest activation. No doubt more sophisticated versions of this model in the future will benefit from the introduction of basins of attraction in the output unit activation space (e.g. as in Hinton & Shallice, 1991).

If our input string contains *nin* characters (including the two suffix markers) and the output string contains *nout* characters (excluding any blanks) then there are:

$$ntarg = nin\,!\,/\,nout\,!\,(nin - nout)\,!$$

ways that the output string can be padded out with blanks to solve the alignment problem. Rather than doing this padding by hand as in the original NETtalk we will allow the network to choose between the *ntarg* possibilities itself. It has been shown (Bullinaria, 1993) that this can be achieved simply by comparing all *ntarg* output targets with the network′s actual output and using the target that already has the lowest output activation error to train the network. Given a sufficiently representative training set and a sufficiently small learning rate, the sensible alignments dominate the weight changes so that eventually the network learns to use the best set of alignments.

Note that our representation works for any word, we do not have to exclude any words that cannot be represented in terms of Wickelfeatures (as in Rumelhart & McClelland, 1986) or that do not fit a particular template (as in MacWhinney & Leinbach, 1991). The only problem we have is the finite length of our input window which places an artificial limitation on the capturing of long range dependencies that may occur for exception words. We shall discuss this important issue in our concluding section.

3. Simulation Results

For convenience, we used the same network and learning parameters as the corresponding reading model (Bullinaria, 1994), namely a window size of 13 characters, 300 hidden units, learning rate 0.05, momentum 0.9 and sigmoid prime offset 0.1. We have not yet investigated whether different parameters can give superior results. The network also automatically attaches a word separation character (namely ′|′) to the beginning and end of each input word since this was found to improve the reading model′s generalization performance. It has not yet been tested whether this helps or hinders our past tense learning performance. Each network was run until it achieved perfect performance on the training data. This typically required between 50 and 100 epochs of training with all the words in the training data set used in random order in each epoch. We have not yet attempted to model word frequency effects with these models. The reason for this is simply that, in this type of model, simulating realistically the wide range of word frequencies experienced by humans is computationally prohibitive (e.g. Bullinaria, 1994). MacWhinney & Leinbach (1991)

Old ANN: % correct			SPA: % correct			New ANN: % correct		
Reg	Irreg	Comb	Reg	Irreg	Comb	Reg	Irreg	Comb
63.3	18.8	59.2	83.0	29.2	77.8	89.9	13.6	83.2
58.8	10.3	53.2	83.3	22.4	76.2	87.7	21.7	81.6
58.7	16.0	54.4	80.9	20.0	74.8	91.0	13.3	84.0
60.3	15.0	55.6	82.4	23.9	76.3	89.5	16.2	82.9

Table 1: Comparison of generalization ability of the ANNs and the SPA.

attempted to use realistic word frequencies in their model and failed to learn all the low frequency irregular words in the training data even after 24000 epochs of training.

We used the same training data as Ling (1994) which is a noise free set of 1389 stem/past tense pairs of which 1253 are regular and 136 are irregular. Our first three runs each took 500 of these pairs at random for training and a non-overlapping set of 500 for testing the generalization ability. The generalization results are shown in Table 1 with the corresponding results of the old ANN and the SPA from Ling (1994, Table 4). Since we wanted to use the full set of training and testing data, which contains words that would not fit into the templates of MacWhinney & Leinbach (1991), we could not use exactly the same random training and testing sets as Ling (1994). Our results do not therefore constitute a direct head-to-head comparison, but the averages over three random runs should allow a reasonably fair comparison.

We see that our new ANN has not only improved considerably on the performance of the old ANN, but it also seems to have done better than the SPA. In fact our average combined performance of 82.9% is remarkably similar to the 82.8% recorded for the SPA with the improved right-justified and isolated suffix template (Ling, 1994, Table 7). It is likely that we are now near the optimal performance for this size of training data. Moreover, with our new ANN we have not needed to resort to templates or any other procedure to solve the alignment problem prior to training. It is difficult to say much more than this because we cannot expect any system to do well on unseen irregular words (if they could it would mean that the words were not really irregular) and it is well known (e.g. Kim et al., 1991; Prasada & Pinker, 1993) that even humans do not always give a regular past tense for unseen verbs. An unseen verb may be given a regular past tense or an irregular past tense derived by analogy with a phonologically related irregular word (in the same way that non-words are sometimes pronounced irregularly).

Of course, if the training data only contained regular words, then it is reasonable to expect correct regular responses for all unseen regular words. For this reason, Ling (1994) carried out a second set of runs, training on various sized sub-sets of the 1253 regular words. In each case the network was tested on all the regular words not used for training. Table 2 shows the generalization performance for our ANN compared with the Old ANN and the SPA from Ling (1994, Table 5). The performance of our ANN is averaged over two runs. The Ling (1994) results are of a single head to head comparison. We see that given a sufficiently representative set of training data our ANN can achieve perfect generalization performance. Of course, given the simplicity of the production rules for the past tenses of regular words, it is inevitable that a suitably constructed symbolic system will also be able to achieve a similar performance. However, now that we have achieved 100% performance with our ANN we can be sure that no symbolic approach can do better.

The reduction in performance as we decrease the training set size is due to the fact that some phonemes are very rare in the training data (e.g. /D/, /T/, /U/ and /2/ each occur in less than 1% of the words). This means that they may not occur at all in some of the smaller

Training	Percent correct on testing		
set size	Old ANN	SPA	New ANN
50	14.6	55.4	51.3
100	34.6	72.9	83.6
300	59.8	87.0	98.5
500	82.6	92.5	99.4
1000	92.0	93.5	100.0

Table 2: Generalization performance after training on regular verbs.

training sets and the networks have no way to generalize to phonemes they have not seen before. They also have problems generalizing when they have only seen a phoneme in a couple of different contexts. Using a distributed input-output representation may improve matters here, but since we can already get perfect performance on only 1000 words, this has not yet been tried.

Another problem with very small training sets is that the multi-target learning algorithm tends to have trouble learning the right alignments, e.g. resulting in learning the mapping /bar[]/ → /ba−rd/ rather than /bar−d/. This will obviously cause problems for the generalization, but is easily remedied by restricting the early stages of training to words with only one target (i.e. the regular /Id/ words), in the same way that we tend to teach children the easy words first. For realistic sized training sets with sufficiently low learning rates, the networks can manage without this interference.

If, as we suggested in the introduction, our networks′ improved performance is largely due to our addressing the recognition problem, we should expect the Old ANN performance on *N* training patterns to be similar to our New ANN performance on *N*/*mwl* patterns, where *mwl* is the mean word length in the training data inputs. (The precise ratio will, of course, depend on the details of the templates used and the distribution of phonemes.) For our past tense data *mwl* = 4.95, so our prediction is in reasonable agreement with the data of Table 2. It also suggests that the Old ANN will require at least of the order of 3000 training patterns to achieve perfect generalization performance even when trained only on regular words.

Since it is now clear that 500 patterns is not quite enough even for totally regular training data we repeated the mixed data simulations of Table 1 with random training data sets of 1000 patterns and the remaining 389 patterns used for testing generalization. The generalization results are shown in Table 3. At this stage it was discovered that the window size of 13 characters was not large enough to capture the long range dependencies necessary to deal with both the regular /IkspEnd[]/ → /IkspEndId/ and the irregular /spEnd[]/ → /spEnt/,

New ANN: % correct		
Reg	Irreg	Comb
91.2	17.9	83.9
92.6	21.4	84.9
92.2	15.6	83.4
92.0	18.3	84.1

Table 3. Generalization performance with 1000 training patterns.

a	a	0.006140	59	v	v	0.001837	75
b	b	0.001718	98	w	w	0.083130	60
d	d	0.000000	215	z	z	0.237004	73
e	e	0.000004	124	D	D	0.992059	7
f	f	0.000493	96	E	E	0.000002	139
g	g	0.013347	78	I	I	0.000000	336
h	h	0.432766	30	N	N	0.909739	28
i	i	0.005813	99	O	O	0.017590	50
j	j	0.464054	28	S	_	0.481740	86
k	k	0.000000	259	T	T	0.990170	11
l	l	0.000096	274	U	U	0.999823	5
m	m	0.000127	140	Z	Z	0.980407	39
n	n	0.000000	293	&	&	0.000115	110
o	o	0.086437	77	1	1	0.052860	30
p	p	0.000000	209	2	2	0.958364	10
r	r	0.000000	439	3	3	0.000188	99
s	s	0.000000	345	6	6	0.000000	456
t	t	0.000000	431	[	_	0.000175	1000
u	u	0.000847	55	]	d	0.963291	1000

Table 4: Default outputs. In each case we have the input, the output, the output activation error score and the number of instances in the input training data.

so the second and third runs here only achieve 99.9% on the training data.

The good generalization performance indicates that the ANN has managed to learn the identity mapping for the verb stems. We can check the default outputs quite easily by passing each phoneme through the network on its own without the suffix or word separation markers. The results for the first network of Table 3 are shown in Table 4. There is only one input phoneme that does not produce the correct dominant output (namely /S/). This is not simply a random error. There are two words in the training data for which an /S/ does map to a blank (namely /k&tS[]/ → /kOt/ and /titS[]/ → /tot/) and the network has clearly decided that keeping the /S/ and blank outputs nearly equally activated (at 0.67 and 0.81) is the best way to deal with them. There is only one other phoneme which has a close rival to the dominant output (namely /U/) but in this case the phoneme is very rare in the training data and all the output activations are very low. The second and third networks of Table 3 give the correct dominant output for all phonemes and there are no close rivals but still the error scores for the rare phonemes are quite high.

The error scores give an indication how strong each default is. The higher the error score, the more dependent it is on the context. This may simply be due to the relatively low occurrence of that phoneme in the training data, it may be because of a relatively high number of exceptional words that contain that phoneme or it may indicate particularly consistent context information for that phoneme. We see that there is a strong correlation between the number of times a phoneme occurs in the training data and the error scores. Even in the networks trained only on regular words we get some very high error scores for the rare phonemes. This again suggests that if we trained on a more representative set of training data we would get even better performance and clearer default outputs. Allowing direct input to output connections may also make things easier in this respect but since it is difficult to justify such an architecture and our model does well enough without them we have not yet tested this possibility.

Clearly, with the current input and output coding, we cannot expect the network to give sensible outputs for phonemes it has not seen before since it will not have built up the connection weights for those units, so default strategies in this sense have not been learnt. However, if a suitable distributed representation were used for the inputs and outputs (e.g. such that each phoneme activated a different sub-set of about half the input units in each set),

d[]|	dId	0.000028		n[]|	nd	0.000000	
t[]|	tId	0.000000		o[]|	od	0.269690	
				r[]|	rd	0.000000	
f[]|	ft	0.000007		u[]|	ud	0.000000	
k[]|	kt	0.000000		v[]|	vd	0.000005	
p[]|	pt	0.000000		w[]|	wd	0.004251	•
s[]|	st	0.000000		z[]|	zd	0.000010	
E[]|	Et	0.364781	•	D[]|	Dd	0.108734	
S[]|	St	0.000148		I[]|	Id	0.027781	•
				N[]|	Nd	0.033120	
a[]|	ad	0.001613	•	O[]|	Od	0.000239	
b[]|	bd	0.000010		T[]|	Td	0.866150	•
e[]|	ed	0.000036		U[]|	rd	0.999108	•
g[]|	gd	0.000031		Z[]|	Zd	0.247931	
h[]|	hd	0.186526	•	&[]|	&d	0.000174	•
i[]|	id	0.201244		1[]|	1d	0.000530	
j[]|	jd	0.084088	•	2[]|	2d	0.208950	
l[]|	ld	0.000000		3[]|	3d	0.000002	
m[]|	md	0.000001		6[]|	6d	0.000000	•

Table 5 Default suffixes and output error scores for the various word endings. A '•' indicates that the word ending does not occur in the training data.

then an identity mapping could be learnt that would generalize appropriately to novel sub-sets corresponding to new phonemes. When given the opportunity, neural networks (and presumably real brains) do tend to distribute their activations as widely as possible over the available hidden units. Given then, that the inputs and outputs of our model will correspond to hidden units in a more complete language acquisition system, it certainly makes sense to use a more distributed representation. Again, a thorough investigation of this possibility must be postponed to a later date.

The next thing we need consider is the extent to which the network has acquired the correct set of rules for the suffixes on regular words. Again we can examine this quite easily by testing the network on input strings with the suffix and word separation markers attached. The default output for just /[]|/ is /d/, which is what we would expect given that this is the most common past tense suffix in the training data. As we introduce more and more context information into the inputs, the network is able to over-rule the lower level defaults in order to provide the correct output suffixes for each word and also any necessary word body changes. The results of supplying each possible final input phoneme in turn are shown in Table 5 (again for the first network of Table 3). The network is seen to produce the correct regular suffix for all the word endings in the training data and also to generalize reasonably well for word endings not in the training data. There are only two cases with close output rivals. The first is for /U/ which is still producing very low output activations. The other is for the suffix of a final /E/ which has /t/ activated at 0.68 and /d/ at 0.51. Given that /E[]|/ never occurs in the training data, this is a reasonable response. For the second network of Table 3 we get the same outputs except that a final /E/ results in the suffix /d/ and the only close rivals occur for our problematic /U/. The third network of Table 3 differs in that it has a suffix /t/ for /E/ and the only close rivals occur with the low outputs for /U/ and /T/.

The context free outputs of tables 4 and 5 constitute the basis of the default set of implicit production rules that have been learnt by the network. That a single distributed system can accommodate such a system of rules and still be able to deal effectively with exceptions to those rules should be considered an advantage rather than a disadvantage (cf. Pinker & Prince, 1988; MacWhinney & Leinbach, 1991). Presumably, the poor generalization of the SPA when trained only on regular words (shown in Table 2) indicates that it has failed to acquire these default production rules.

REGULAR WORDS

no change

SEr[]	SErd	SEr
kEr[]	kErd	kEr
flo[]	flod	flo
End[]	EndId	End
kro[]	krod	kro
hEd[]	hEdId	hEd
rIsk[]	rIskt	rIsk
6t&k[]	6t&kt	6t&k
stOk[]	stOkt	stOk

corrupted Id

p&t[]	p&tId	p&td
kost[]	kostId	kostd
rIgrEt[]	rIgrEtId	rIgrEtd
bord[]	bordId	bordd
dIfit[]	dIfitId	dIfitI
IkspEnd[]	IkspEndId	IkspEndI
l&nd[]	l&ndId	l&ndI

vowel change

gIg6l[]	gIg6ld	g6g6ld
wil[]	wild	wEld
spIl[]	spIld	sp6ld
fri[]	frid	frEd
fr3t6n[]	fr3t6nd	frot6nd
fUlfIl[]	fUlfIld	f1lfIld
wild[]	wildId	wEldId
brOd6n[]	brOd6nd	brEd6nd
fitS6r[]	fitS6rd	fEtS6rd
sno[]	snod	snu

others

dIs3d[]	dIs3dId	dIs3tId
s6bs3d[]	s6bs3dId	s6bs3tId
abskjur[]	abskjurd	baskturd
skId[]	skIdId	skIt
lin[]	lind	lint

IRREGULAR WORDS

regularized

6ndu[]	6ndId	6ndud
swIm[]	sw&m	swImd
sIt[]	s&t	sItId
swEr[]	swor	swErd
drO[]	dru	drOd
wek[]	wok	wekt
dr3v[]	drov	dr3vd
kwIt[]	kwIt	kwItId
f3t[]	fOt	f3tId
krip[]	krEpt	kript
mek[]	med	mekt
bIk6m[]	bIkem	bIk6md
S3n[]	Son	S3nd
sik[]	sOt	sikt
luz[]	lOst	luzd
b6rst[]	b6rst	b6rstId
ov6rk6m[]	ov6rkem	ov6rk6md
wIDdrO[]	wIDdru	wIDdrOd

no change

bEr[]	bor	bEr
h3d[]	hId	h3d
sl3d[]	slId	sl3d
f6rbId[]	f6rb&d	f6rbId
brek[]	brok	brek
brIN[]	brOt	brIN

others

stil[]	stol	stold
6phold[]	6phEld	6phildId
f6rgIv[]	f6rgev	f6rgevd
spid[]	spEd	spot
fil[]	fElt	fEl
str3k[]	str6k	strokt
wIThold[]	wIThEld	wIThild
wiv[]	wov	wEvd

Table 6: A typical set of ANN generalization errors. In each case we list the input, the target output and the actual output.

Of course there is more to past tense learning than getting the best generalization score. We also have to get realistic errors on the past tenses that are not produced correctly. Table 6 lists the generalization errors for the first of the runs shown in Table 3. The other two runs give similar patterns of errors. We see that in general they are psychologically realistic. Most of the irregular word errors are regularizations, several follow the 'no change' sub-rule and the rest are a mixture of other sub-rule responses. The majority of the regular word errors can be seen to follow from the application of various sub-rules and analogies. Those

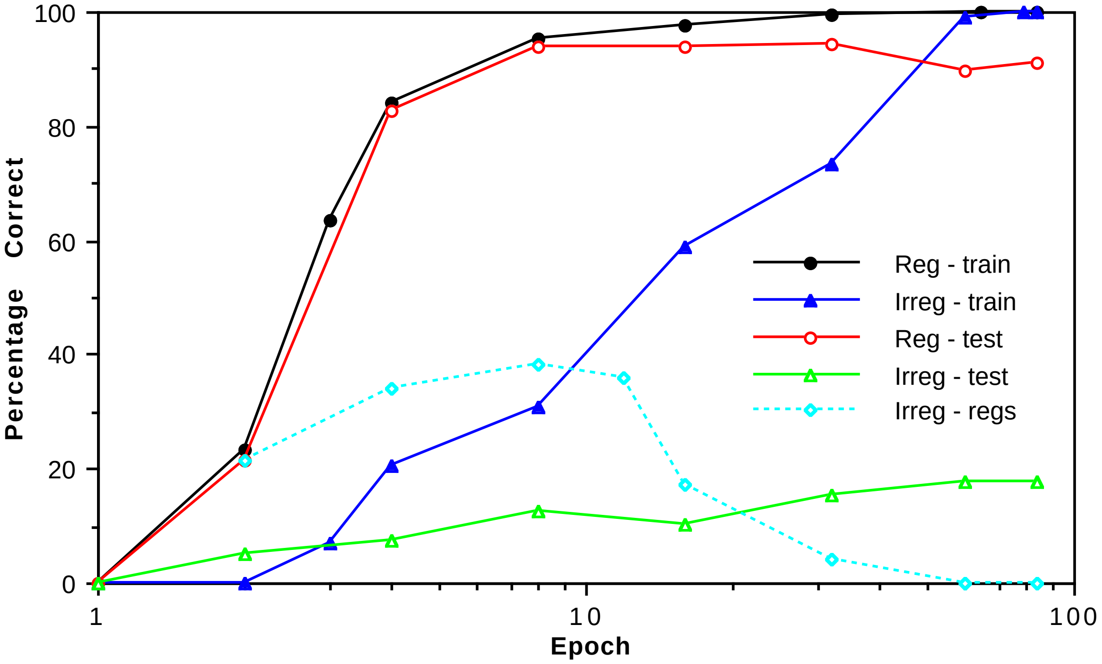

Figure 2. Typical learning curves for our ANN.

denoted 'corrupted Id' are failed attempts to replace the /Id/ suffix by the 'no-change' sub-rule. If we coded each suffix by a single output unit, this form of error would disappear. Our general conclusion seems to be that the networks are performing reasonably well, but their use of the various sub-rules and analogies is not yet sufficiently constrained. There is good reason to suppose that using a more representative set of training data will alleviate these problems.

4. Discussion and Conclusions

We have presented a prototype ANN model of past tense learning that gives near human level generalization performance without any need to pre-process the training data (e.g. by fitting it into templates). However, there is more to modelling the learning of past tenses than getting the best performance out of the trained network. There are various frequency and developmental effects that need to be modelled, performance after damage must be examined, etc.

Figure 2 shows a typical set of learning curves (again for the first network of Table 3). The fact that the network acquires the regular words before the irregulars is further evidence that the network is concentrating on learning the main rules and only accommodating the exceptions when forced to. Notice that there is no sign in the training data performance curves of any large scale U-shaped learning effects that are often observed in children (e.g. Kuczaj, 1977). A more detailed study with the networks trained on a more realistic word frequency distribution (which would result in a much larger proportion of irregular words in the training data) may result in the required effects as discussed in Plunkett & Marchman (1991). (As noted already, constraints on computational resources have so far prevented us from attempting to model any word frequency effects.)

A more traditional account of the U-shaped curves has the child begin by using a lexical/semantic system to memorize the relatively small number of past tenses in their vocabulary. Many of these past tenses will be irregular, there will be no evidence of rule usage and they will not show any regularity effects. Then at a later stage, as the number and regularity of the past tenses in their vocabulary increases, it becomes more efficient to make use of the regularities in the data and hence an increasingly complex hierarchy of rules and sub-rules are acquired. (This is presumably what our connectionist system is modelling.) Over application of these rules at an early stage of leaning (e.g. corresponding to the first forty epochs of Figure 2) will result in the over-generalization of irregular words commonly found in children. As learning proceeds (e.g. beyond epoch sixty of Figure 2) the rule based system will be able to deal with the irregular words as well as the regular words and the U will be complete. The dotted curve in Figure 2, which shows the number of regular outputs for the irregular words in the training data, confirms that the over-regularization of the irregular words in the training data does occur in this way in our network. That some homophonous verbs have different past tenses (e.g. 'lie' → 'lay' or 'lied') is further evidence that a semantic system must be involved in addition to the rule based system (e.g. Kim et al., 1991). There is good evidence that a similar dual route system of language acquisition is also necessary for reading and spelling (e.g. Bullinaria, 1994).

We have only dealt with the normal development and final performance of our model. Important constraints are placed on cognitive models by their ability to account for abnormal development in children and performance after various types of brain damage (e.g. Pinker, 1991). A preliminary investigation indicates that damage to our past tense model results in a graceful degradation of performance with a higher proportion of errors on the irregular words than on the regular words. (This is the same pattern of errors that we find for the corresponding reading and spelling models.) This is something else that will require further study in the future, both for the connectionist and the symbolic approaches.

The model presented here is very much a first attempt at a NETtalk style solution to the past tense learning problem. Assuming that real brains adopt similar strategies for a range of types of language acquisition it is encouraging that the existing successful reading and spelling models require virtually no changes to result in a successful past tense production system. In principle, the same system could also learn other verb tenses at the same time. This might be achieved by using different suffix markers for each of the different tenses or by using general suffix markers in conjunction with special input units (not in the moving window) to indicate which tense is required. Since the same concepts of verb stems and suffixes are involved, the larger training set should also result in improvements over our current performance.

The most serious limitation of our model is the use of the moving window which is psychologically implausible in may respects. In principle, however, the window of context information can be replaced by a system of recurrent connections (e.g. Jordan, 1986) that is able to learn any long range dependencies it might require. Such a system may also be able to output the suffixes at the end of words without the need for any explicit suffix markers on the input strings (which is another unsatisfactory feature of the current model). These recurrent connections must also be added to our already long list of possible future improvements to the model.

References

Bullinaria, J.A. (1993). Neural Network Learning from Ambiguous Training Data. Submitted to *Connection Science.*

Bullinaria, J.A. (1994). Representation, Learning, Generalization and Damage in Neural Network Models of Reading Aloud. Edinburgh University Technical Report.

Fahlman, S. E. (1988). Faster-Learning Variations on Back-Propagation: An Empirical Study. In *Proceedings of the 1988 Connectionist Models Summer School,* Morgan Kauffmann.

Hinton, G.E. & Shallice, T. (1991). Lesioning an Attractor Network: Investigations of Acquired Dyslexia. *Psychological Review*, **98**, 74-95.

Jordan, M.I. (1986). Attractor Dynamics and Parallelism in a Connectionist Sequential

Machine. *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, 531-536, Hillsdale, NJ: Erlbaum.

Kim, J.J., Pinker, S., Prince, A. & Prasada, S. (1991). Why no mere mortal has ever flown out to center field. *Cognitive Science*, **15**, 173-218.

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

Lachter, J. & Bever, T. (1988). The relation between linguistic structure and associative theories of language learning: A constructive critique of some connectionist learning models. *Cognition*, **28**, 195-247.

Ling, C. X. (1994). Learning the Past Tense of English Verbs: The Symbolic Pattern Associator vs. Connectionist Models. *Journal of Artificial Intelligence Research*, **1**, 209-299.

Ling, C.X & Marinov, M. (1993). Answering the connectionist challenge: a symbolic model of learning the past tense of English verbs. *Cognition*, **49**, 235-290.

MacWhinney, B. (1990). *The CHILDES Project: Tools for Analyzing Talk.* Hillsdale, NJ: Erlbaum.

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

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

Pinker, S. (1991). Rules of Language. *Science*, **253**, 530-535.

Pinker, S. & Prince, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition*, **29**, 73-193.

Plunkett, K. & Marchman, V. (1991). U-shaped learning and frequency effects in a multi-layered perceptron: Implications for child language acquisition. *Cognition*, **38**, 43-102.

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

Rumelhart, D.E., Hinton, G.E. & Williams, R.J. (1986). Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing*, Volume 2 (eds. D.E. Rumelhart & J.L. McClelland) Cambridge, Mass: MIT Press.

Rumelhart, D.E. & McClelland, J. (1986). On learning the past tense of English verbs. In *Parallel Distributed Processing*, Volume 2 (eds. D.E. Rumelhart & J.L. McClelland) Cambridge, Mass: MIT Press.

Seidenberg, M.S. & McClelland, J.L. (1989). A distributed, developmental model of word recognition and naming. *Psychological Review*, **96**, 523-568.

Sejnowski, T.J. & Rosenberg, C.R. (1987). Parallel Networks that Learn to Pronounce English Text. *Complex Systems.* **1**, 145-168.

Sullivan, K.P.H. & Damper, R.I. (1992). Novel-Word Pronunciation within a Text-to-Speech System. In G. Baily, C. Benoît & T.R. Sawallis (eds.),*Talking Machines: Theories, Models and Designs*, Elsevier, Amsterdam.