Generalization In Simple Recurrent Networks

Marius Vilcu (mvilcu@cs.sfu.ca)
School of Computing Science
Simon Fraser University, 8888 University Drive, Burnaby, Canada, V5A 1S6

Robert F. Hadley (hadley@cs.sfu.ca)
School of Computing Science
Simon Fraser University, 8888 University Drive, Burnaby, Canada, V5A 1S6

Abstract

In this paper we examine Elman's position (1999) on generalization in simple recurrent networks. Elman's simulation is a response to Marcus et al.'s (1999) experiment with infants; specifically their ability to differentiate between novel sequences of syllables of the form ABA and ABB. Elman contends that SRNs can learn to generalize to novel stimuli, just as Marcus et al's infants did. However, we believe that Elman's conclusions are overstated. Specifically, we performed large batch experiments involving simple recurrent networks with differing data sets. Our results showed that SRNs are much less successful than Elman asserted, although there is a weak tendency for networks to respond meaningfully, rather than randomly, to input stimuli.

Introduction

In a recent paper, Elman (1999) casts doubt upon the widely noted results of Marcus et al. (1999). In the Marcus et al.'s experiment, 7-month old infants were habituated to sequences of syllables of the form ABA or ABB (e.g., "we di we" or "le di di"). Marcus et al. found that infants showed an attentional preference for novel test sequences of syllables (which we call "sentences"), which differed from the habituation stimuli[1]. Marcus et al. argue that the reason for this behavior is the fact that infants extracted "algebra-like rules that represent relationships between placeholders (variables)" (1999). They also concluded that simple recurrent networks (and, in general, all networks whose training is based on backpropagation of error) were not able to display this kind of behavior because they could not generalize outside the training space.

The issue of generalization outside the training space was previously addressed in Niklasson and van Gelder (1994), and Marcus (1998). In essence, the training space represents the n-dimensional hyperplane delimited by the set of training vectors. We say that a connectionist model generalizes to novel stimuli when correct output is reliably produced for an input item that was not included in the training set (i.e., the network was never trained on that stimulus in any position within its input layer). Marcus maintains that a neural network trained with the backpropagation algorithm (or any variant of it) is not able to display such a behavior, because the innate structure of the backpropagation algorithm[2] precludes the network from generalizing to nodes that have not been specifically trained.

Elman agrees that the Marcus experiment does "indicate that infants discriminated the difference between the two types of sequences" (1999), but he believes that this result may be explained by the relationship between the last two syllables: infants were able to distinguish that in one case the last two syllables were identical (ABB), and in the other case the last two syllables were different (ABA). Moreover, Elman maintains that it is feasible for a simple recurrent network to perform this same task, provided the network is presented with the same background knowledge as infants have (in particular, an exposure to a wide range of syllables that infants have before participating in the experiment).

Having said that, Elman performs an experiment involving an SRN that aims to simulate the Marcus et al.'s experiment. There are three phases in Elman's simulation: 1) the pre-training period, corresponding to the prior experience of the infants in learning to recognize syllables; 2) a second phase corresponding to the habituation task that infants encountered (presenting ABA and ABB sentences); 3) a testing phase involving novel stimuli, as in the infants' experiment. At the end of his simulation, Elman concludes that his results "clearly indicate that the network learned to extend the ABA vs. ABB generalization to novel stimuli" (1999).

Granting Elman's basic assumptions, we constructed an experiment that mimics his simulation. We did not

[1] For example, after habituated to ABA sequences, the infants spent more time recognizing novel test sequences of the form ABB than did for ABA sequences, and vice versa.

[2] The weights connecting a given output node are trained independently of the weights connecting any other output node. Consequently, the set of weights connecting one output unit to its input units is entirely independent of the set of weights feeding all other output units. This is called input-output independence, and it is believed to be the major weak point of backpropagation neural networks. It is less clear that the problem arises for competitive learning networks, however. See Hadley et al (1998) for details.

have access to all Elman's data, but we used the same Plunkett and Marchman's (1993) distinctive feature notation of consonants and vowels that Elman employed in his experiment. However, since the results we obtained led us to a different conclusion than Elman's, in order to have a more complete picture of the performances of simple recurrent networks, we created a variety of data sets by changing the degree of overlapping units in the training/testing vectors. Also, to be sure that the results were not obtained by chance, we performed batch training, i.e., at least 64 different training-test sessions were carried out for each individual training corpus (i.e., 64 or 128 different weight initializations were assigned to the basic configuration, resulting in 64 or 128 separate networks trained on each data set).

Basic structure of Elman's and our experiments

A simple recurrent network architecture was used for all experiments. The input layer contains 12 or 24 units (depending on the experiment; see details below). The number of hidden/context units was varied between 10 and 40. The output layer contains two units; one was used only during the pre-training phase, while the other unit was only used during the sentence habituation and testing phases.

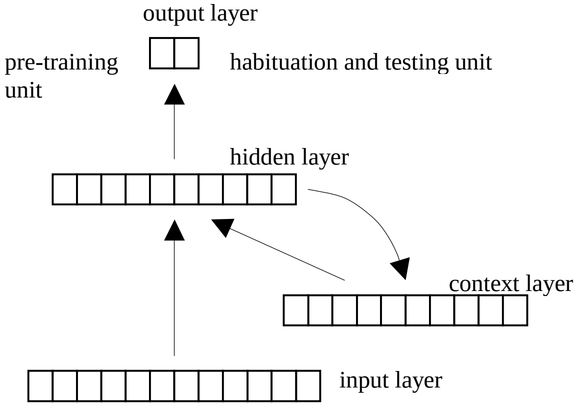

Fig. 1. Architecture of the network

The data set deployed in the pre-training phase contained 50,000 syllables (separate tokens) from the full set of 120 possible types. Each syllable was presented to the network, one at a time, and the SRN was trained to distinguish between the current syllable and the previous one (whether or not they are identical). Only one of the two output units was used during this supervised training.

The habituation phase followed the pre-training phase. During this phase the same network was presented with 32 distinct sentences formed with 8 different syllables (these 8 syllables also occurred in the set of 120 types of syllables employed in the pre-training set). The 32 sentences were generated from the ABA and ABB grammars (16 ABA sentences, and 16 ABB sentences). Each sentence was presented to the network, one syllable at a time. Following the last syllable of a sentence, the network was trained to output a 0 in the case of ABA sentences, and a 1 in the case of ABB sentences. During this training phase only the second output unit was used (the one not used during pre-training). Interestingly, the weights were modified only after the last syllable of the current sentence was presented (no training occurred following the first two syllables). This was done in order to ensure that the network would learn to make discriminations the same way as the infants presumably would, using similar stimuli.

For testing, four sentences were used, formed with 4 "relevantly novel" syllables (i.e., these syllables appeared in the 120-syllable pre-training set, but not in the training corpus). Two sentences had the form ABA, and the other two had the form ABB. Again, the second output unit was used to monitor the network's responses.

Before presenting our results, we would like to clarify the following issues:

1) Because we were not able to have *exactly* the data set that Elman used, we generated our patterns based on Plunkett & Marchman's (1993) feature representation of consonants and vowels. Since Elman encoded his stimuli using the same notation, we believe that the difference between our data set and Elman's is minimal and arguably insignificant to the outcome of the experiment.
2) Elman's main objective was to challenge Marcus' assertion that SRNs are not able to generalize outside the training space. However, we believe his claims are overstated. Although a minority of sessions in our batch jobs was as good as Elman's, *in general*, we found that the SRN did not perform as well as Elman maintains.

As noted above, all experiments were based on Elman's simulation. Between our experiments and Elman's there were a few differences, however. These consisted in the way in which the data sets were created and the way the results were computed. Our first data set very closely resembles Elman's representation of vectors, both corpora being based on the same distributed representation of syllables. Since the results we obtained for this first data set offered only little evidence to support Elman's position, we have created a second corpus of patterns, by changing Elman's original vectors in order to have a more uniform and more overlapping data set (see below the description of Experiment 2). Lastly, our third data set employs completely non-overlapping vectors, i.e. we used a localist representation of input patterns.

Experiment 1

The input corpus for this experiment was very close to the Elman's data set. We used distributed representations to create the patterns: each syllable had 12 phonetic features, each syllable being made up from a consonant followed by a vowel. All the syllables were generated randomly using the whole set of letters, and the patterns were created based on Plunkett & Marchman's (1993) notation of each letter[3]. We created 120 vectors this way. All of these patterns were used in the pre-training phase, while 8 of them were employed during training and other 4 vectors were used for testing.

For example, here are 2 of the 8 training syllables and 2 of 4 testing syllables:

```
   training
mo -1  1 -1 -1  1  1 -1  1 –1  1 –1  1
wu -1  1 -1  1  1  1  1  1  1  1 –1  1
   testing
za -1  1  1 -1 -1  1  1  1  1  1 -1 -1
fe -1 -1  1 -1  1  1  1  1 -1 -1  1 -1
```

We tried to generate as diverse and random data set as possible, like infants are presumably exposed to prior to participating in the Marcus et al's experiment. However, our results showed that these patterns were not very "friendly" to our SRNs, and the networks were not nearly as successful as infants in discriminating those sentences.

One of the characteristics of this data set was that, because of the randomness of patterns, many of the testing vectors were very different from the vectors employed in training. For instance, the average distance[4] among training vectors was about 3-4 bits, while the difference between training and testing patterns exceeded 6-7 bits. In our opinion, this contrast among patterns is responsible for making the testing session difficult.

Experiment 2

Since the results based on the first data set failed to prove Elman's strong claims, we generated a different corpus of stimuli, trying to make the training process successful. Consequently, we have manually created 12 vectors (8 vectors are used in training, the other 4 in testing). The remainder of 108 vectors have been generated randomly. All these patterns have been distributed uniformly between the two sets of stimuli (training and testing). In this way, the distance among vectors within the same set of stimuli (whether training or testing) was similar to the distance between vectors found in the different corpora.

For example, here are a few of the vectors used in this experiment:

```
 training
 -1 -1 -1  1  1 -1 -1  1 -1  1  1  1
  1 -1 -1  1  1 -1 -1  1  1 -1 -1  1
 testing
  1 -1  1 -1  1 -1  1 -1  1 -1  1 -1
 -1 -1  1  1 -1 -1  1  1 -1 -1  1  1
```

In this case, the average difference among the vectors within the same corpora is about 4-6 bits. This value is close to the difference between training stimuli and testing stimuli (6-7 bits). Because the patterns are uniformly spread across the training and testing sets, this represented a training advantage for networks. However, this tactic further reduces the novelty of the test "sentences".

Experiment 3

The third experiment involved a rather different data set. This employed completely *non-overlapping* vectors. As a result, the vectors were larger (24 bits, instead of 12).

For example, here are 4 of 12 training/testing vectors (the rest of the 108 vectors used during pre-training were generated randomly):

```
1) 1 –1  1 –1 –1 –1 –1 –1 –1 –1 –1 –1
–1  1 –1 –1 –1 –1 –1 –1 –1 –1 –1 –1
2)–1  1 –1  1 –1 –1 –1 –1 –1 –1 –1 –1
–1 –1 –1 –1 –1 –1 –1 –1 –1 –1 –1 –1
3)–1 –1 –1 –1  1 –1  1 –1 –1 –1 –1 –1
–1 –1 –1 –1 –1 –1 –1 –1 –1 –1 –1 –1
4)–1 –1 –1 –1 –1  1 –1  1 –1 –1 –1 –1
–1 –1 –1 –1 –1 –1 –1 –1 –1 –1 –1 –1
```

As it may be seen, the patterns were generated by moving a 4-bit frame [1 –1 1 –1] along the 24 bit vectors. In this way, the resulting vectors do not overlap, and the distance among all vectors is the same (4 bits).

Although this corpus of stimuli had little in common with Elman's data, we wanted to examine the performance of SRNs in the case of non-overlapping vectors and to address Marcus' issue about generalization to genuinely novel items.

Results

The performances of SRNs (and, in general, any network using the backpropagation algorithm) are influenced by several training parameters, such as initial weights, learning rate, etc. Usually, the initialization of weights is performed randomly and if training parameters are not chosen properly (especially the learning rate), the network may end up in a region of local minimum of the error function. One way to

[3] For example, the pattern for syllable "da" was a 12-bit vector created by concatenating the 6-bit feature representation of "d" with the 6-bit feature representation of "a".

[4] The distance between two vectors is given by the number of bits by which the two vectors differ.

reduce this liability is to perform a batch experiment, to test the network with a large number of different weight initializations and training parameters. Another advantage of this approach is that at the end of the batch sessions we will have a more precise picture of the behavior of the networks, and also know whether or not the results are generated accidentally.

Significantly, in a series of preliminary experiments, we found that, very often, the weight initializations we used determined poor results for networks, regardless of training parameters (including hidden layer size). Therefore, we decided to perform at least 64 different training-test sessions for each of our 3 experimental designs, each session using a different weight initialization (there were 64 sessions for the first two experiments, and 128 different sessions for the third experiment). In this way, we generated at least 64 separate trained networks for each batch experiment[5].

We chose to use two criteria in order to evaluate the results:

(1) Our first criterion was simply based on the percentage of "acceptable" results. We say that a network generates acceptable results when it grammatically categorizes each presented test sentence within 30% of its target value. Since there are 4 test sentences (2 ABA sentences, and 2 ABB sentences), we have 4 output values to record for each of the 64 networks (let's say, A, B, C, D are the network outputs for the 4 test sentences). Having 0 as the target value for A and B, and 1 as the target value for C and D, an acceptable result is an output value less than .3 for A and B, and greater than .7 for C and D.

(2) Although the above-mentioned criterion is very tolerant, Elman's results would not be counted as acceptable in conformity with this criterion (one of Elman's responses is about .6, outside the 30% error margin; see below for more details). However, not to reject Elman's approach on a-priori grounds, we adopted a second, more forgiving criterion: we consider a result as "acceptable" if only 3 of the 4 responses are within 30% of their target values, while the remaining response is within 45% of its target value.

Since our extensive set of training-test results were significantly different from Elman's isolated result, we tried to see whether, at least, they were better than mere chance. For the first criterion, the chance value is given by the probability that all 4 test sentences are fortuitously, correctly classified, i.e., the network outputs are within 30% of the target values. Clearly, the chance probability that the network outputs a value in the target range, for any of the 4 test sentences, is .3. Therefore, the probability that all 4 sentences are correctly recognized is .0081 (=.3 x .3 x .3 x .3), or .81%. This represents the "chance" value, which we compared all our results to. For the second criterion, the chance value is slightly different. Here, in order to correctly categorize purely by chance, networks should report output values within 30% of the target values for 3 sentences, and within 45% for the 4th sentence. Consequently, the probability for that happening is .01215 (=.3 x .3 x .3 x .45), or 1.215%.

Experiment 1

This experiment is closest to Elman's simulation. Results reported by Elman (1999) were as follows:

	response	target		response	target
A	0.004	0	C	0.853	1
B	0.008	0	D	0.622	1

Even though the network's response for the last sentence was very close to the chance value of .5, Elman asserted, "these responses clearly indicate that the network learned to extend the ABA vs. ABB generalization to novel stimuli" (1999). In our view, based only on these results, Elman overstates the facts. In accord with our first evaluation criterion, his result would not even have been considered acceptable. We devised the second criterion, even more lenient than the first one, in order to cover Elman's result. In any case, in our extended series of experiments, we found that the responses of our networks were highly dependent on weight initializations.

We performed numerous batch experiments, systematically varying, in all combinations, the available parameters values: learning rate (between 0.01 and 0.1), the number of hidden/context units (between 10 and 40), the momentum (0 and 0.5), weight initialization (within the interval [-1, 1], or [-0.1, 0.1]). The best results were obtained for 30 hidden/context units, a learning rate of 0.01, momentum 0 and weight initialization within [-1, 1].

Specifically, for this first experiment, our results were:

(1) of the 64 trained networks, 15 generated acceptable results in conformity to the first evaluation criterion; thus, the percentage of acceptable results is 15/64 x 100 = 23.43%;

(2) evaluating with the second, more lenient, criterion, the percentage of acceptable results is 23/64 x 100 = 35.93%;

We believe the results lend, at best, weak support to Elman's claims. A percentage of good results around 30% cannot lead us to the conclusion that "the network learned to extend the ABA vs. ABB generalization to novel stimuli", as Elman asserted (1999). Granted, the results are significantly larger than the chance values (.81% for the first criterion, and 1.215% for the second one), which means that there is a tendency for the networks to train in such a fashion that they give meaningful, rather than random results.

As noted earlier, these results might be partially explained by the randomness of patterns used in this

[5] One could metaphorically regard these trained networks as the infants involved in the Marcus et al 's experiment.

experiment. There were instances when the training vectors were very different from the vectors used for testing (up to 90% of the bits were different).

To prove that a different corpus of stimuli can generate better results, we performed a second set of tests, making the data set more uniform and decreasing the distance between training and testing patterns. Here are the details:

Experiment 2

As noted earlier, most part of the 120 patterns used in this second experiment were created randomly, except the 12 vectors employed in the training (habituation) and testing phases. These 12 vectors were generated manually and distributed uniformly over the training and testing sets in order to have a similar distance among all the vectors.

In this case, the average difference between training and testing vectors is about 6 bits, close to the distance among vectors within the same set (whether training or testing), which is about 7 bits.

We varied many training parameters in this case too, and we obtained the best result for 40 hidden/context units, a learning rate of 0.1 and momentum 0.5.

As expected, the results were substantially better:

(1) there were 40 trained networks (out of 64) whose responses were acceptable in conformity with the first evaluation criterion; thus, the percentage of acceptable results is 40/64 x 100 = 62.5%;

(2) there were 42 trained networks that responded acceptably in accord with the second criterion; the percentage of acceptable results is: 42/64 = 65.62%;

Noteworthy, these results were obtained for a number of 40 hidden/context units. When using 10 hidden/context units (as Elman presumably did), the results were worse: 29.68% in accord with the first criterion, and 34.37% evaluating with the second criterion.

Although the percentages of 62.5% or 65.62% of successfully trained networks are not impressive, in contrast with the chance value of .81% (and 1.215% respectively), they represent a significant result (the probability to respond acceptably, in conformity to our criteria, is 80 times greater than the probability by mere chance). Therefore, this experiment demonstrates more convincingly what we noted earlier: there clearly is a tendency for the networks to train in such a fashion that they give meaningful, rather than random results. However, we must bear in mind that the training regime now under consideration *does* ***not*** *satisfy the conditions for generalization outside the training space*.

Experiment 3

The third experiment differs from the first two with respect to the type of the vectors involved: here we used *completely non-overlapping vectors*, because we wanted to address Marcus' challenge of generalization outside the training space. Thus, we tried to discover whether simple recurrent networks are indeed able to generalize to novel stimuli.

Initially, it would seem that our testing patterns were not novel to the network (since they also appeared in the pre-training set). But, there are two arguments behind our assumption that the testing vectors are actually novel:

- the output unit used during pre-training is different from the output unit used during habituation (second training). Since the representation of patterns is localist and the training algorithm is backpropagation, these two output units are purportedly independent: the training of one unit should not influence the other unit, as Marcus argued (1998).
- the training regimes used during the pre-training and habituation phases are different (one algorithm teaches the network to determine whether or not consecutive syllables are identical, while the other one teaches the network to differentiate between ABA and ABB sentences). Since the testing vectors do not appear in the training data set used during *sentence* habituation, they are novel to the network in the relevant sense.

For this experiment we performed two sets of tests, both involving 128 separate training/testing sessions. Although 64 trained networks are presumably enough to form a complete picture of the behavior of networks, we wanted to see whether or not the general tendency noted previously was repeatable for a substantially larger batch experiment. The answer was affirmative.

The first set of experiments employed a test corpus of 4 sentences, exactly the same number of sentences used by Elman, and by us in the experiments 1 and 2. The results were as follows:

(1) there were 8 successfully trained networks (out of 128); thus, the percentage of well-trained networks was, in conformity with the first criterion, 8/128 = 6.25%;

(2) there were 14 trained networks which responded acceptably in accord with the second criterion; the percentage of acceptable results is: 14/128 = 10.93%;

Although these values are much less impressive than those of the previous experiment, they still are better than chance. Of course, the absolute percentage of successful networks (6.25, or even 10.93) is small, indicating that SRNs have problems dealing with novel stimuli. However, it is still substantially greater than .81 (or 1.215 for the second criterion), which would have been obtained by pure chance.

However, for the second set of tests we expanded the test corpus to 30 sentences. In this case, *none* of the 128 trained networks output good results in accord with *any* of the two criteria. This result was consistent for different training parameters, such as learning rate and number of hidden/context units.

Discussion

In the three experiments described herein, we have systematically varied a wide range of parameters. Indeed, in the case of Experiment 1, were Elman's data set is very closely approximated, we have parametrically varied not only the learning rate and weight initialization range, but also the hidden layer size (which Elman did not do). On the basis of all three experiments described above, we believe it is fair to say that Elman's case has been substantially overstated.

On the other hand, certain of our results may lend some modest confirmation to Elman's position, at least with respect to the very simple syntax employed in the Marcus et al experiment. To be sure, in the case of Experiment 1, which most closely approximated Elman's training data, the percentage of successfully trained networks was only 23.43%. However, this percentage is far above the purely chance values that we have cited. In addition, we have shown that even when all input vectors within a given training corpus are completely non-overlapping (Experiment 3), as many as 6% of trained networks satisfy our “least forgiving” criterion of success, at least when the test corpus contained just 4 sentences (as in Elman's case). Significantly, though, when the test corpus for Experiment 3 was expanded to contain 30 novel sentences, no positive results whatsoever were obtained even when our more lenient “success criterion” was used. This outcome lends clear support to Marcus' claims on “generalization outside the training space” -- at least with respect to the infant learning experiment described by Marcus et al (1999).

Finally, we must also emphasize that, except in Experiment 2 (where we modified the syllable vectors to ensure that training and test input vectors were much more similar), the *preponderance* of trained networks failed to satisfy even the most forgiving success-criterion adopted here. More importantly, we have replicated the design of Experiment 1 using two modestly more complex grammars, and have obtained only negative results. In particular, when the grammars (ABCA vs. ABCB) and ABCDA vs. ABCDB) were employed, we were unable to train *even a single* network successfully (from a batch of 64 networks). This strongly suggests that the SRN architecture deployed in Elman's "refutation" of Marcus is incapable of abstracting the underlying structure of anything but the very simplest of grammars. Our view is that the "grammar" deployed by Marcus et al (1999) is perhaps too simple to present a useful challenge to eliminative connectionist networks. A desirable step for future research would be to repeat the "human infant experiment" using the modestly more complex grammars just cited.

References

Elman, J. (1998). Generalization, simple recurrent networks, and the emergence of structure. M.A. Gernsbacher and S.J. Derry (Eds.) *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society*. Mahwah, NJ: Lawrence Erlbaum Associates

Elman, J. (1999). Generalization, rules, and neural networks: A simulation of Marcus et al., in press (http://crl.ucsd.edu/~elman/Papers/MVRVsimulation.html)

Hadley, R.F., Arnold, D., and Cardei, V. (1998). Syntactic Systematicity Arising from Semantic Predictions in a Hebbian-Competitive Network, *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society*, Madison, Wisconson: Lawrence Erlbaum Assoc., Publishers.

Marcus, G. F. (1998). Rethinking eliminative connectionism, *Cognitive Psychology*, vol. 37.

Marcus, G. F., Vijayan, S., Rao, S. B., Vishton, P. M. (1999). Rule learning in seven-month-old infants, *Science*, 283, 77-80.

Niklasson, L. F. and van Gelder, T. (1994). On being systematically connectionist. *Mind and Language*, 9:288-302

Plunkett, K., & Marchman, V. (1993). From role learning to system building: Acquiring verb morphology in children and connectionist nets. *Cognition*, 48, 21-69