Mutual Information as a Segmentation Cue in Connectionist Learning Approcahes

Joshua Herring
jwherrin@indiana.edu

Abstract

Since the mid-1980s, neural networks have been increasingly important as a research tool in nearly all areas of Cognitive Science. They are not, however, without their problems, greatest among these being a lack of semantic transparency: it simply isn't known, in most cases, how a given network arrives at the outputs it produces. This paper addresses this problem as it applies to a specific domain of cognitive linguistic research: the segmentation problem. Specifically, it investigates the extent to which neural network outputs correspond to patterns in pointwise mutual information scores. Two neural networks were trained on approximately 10000 3- and 4-letter patterns, and the hits, misses and false alarms in the output were compared to mutual information scores at the point of spacing.

Introduction: The Segmentation Problem meets Connectionism

The first linguistic task faced by any child in the early stages of first language acquisition is the segmentation problem. Confronted with a jumble of sounds, the infant somehow comes to organize it into meaningful units. What biological methods it uses to accomolish this are a subject of intense debate. In many camps it is claimed, in effect, that the problem is uninteresting because the necessary tools are specified innately: language acquisition is a matter of "recognizing" basic units one is programed to find salient. There is almost certainly some truth to this - but it leaves a lot of interesting questions unanswered. Even assuming access to innate features (by no means an uncontroversial assumption), for example, the infant still has to organize given segments into meaningful units - attaching, as it were, the correct features to the correct "words."

Mutual Information

Computational Linguistics researchers have tended to see this as a statistical analysis problem. Given discrete units, the task is to identify patterns, which in the case of a string of seemingly random symbols would involve identifying which combinations were highly frequent, which were not so frequent, and which nonexistent. (It is not altogether clear what this would mean in the case of a completely continuous signal, but presumably it would again involve repeated patterns in the input, only with more involved mathematical ways of identifying them.) Many statistical methods for identifying significant collocations have been developed. The one that perhaps sees the most use in Computational Linguistics is the information theoretic measure Mutual Information, which is given by the following formula:

$$MI(x, y) = log_2 \frac{p(x, y)}{p(x)p(y)}$$

Many interpretations are available for this, but to some views this is an expression in terms of $bits$ of the amount of information contained in a sequence. It has long been established that highly frequent collocations need shorter bit sequences to be expressed under optimal coding methods. Thus, a higher Mutual Information score - indicating a highly significant collocation - could be seen as a measure of potential data compression - i.e. how many bits are saved (under a hypothetical optimal coding scheme) by knowing that y follows x. In theory, mutual information should be available to help with the segmentation problem. While no one believes that children are expressly calculating these scores, regions of high mutual information between segments in a sequence with low mutual information at the outer edges should correspond to patterns that learners are likely to identify as cohesive and available across contexts - i.e. should be linguistic units.

Connectionism

Connectionist methods have become increasingly popular as frustration with the limitations and perceived cognitive implausibility of purely symbolic approaches grows. Supporters of this trend argue that the phenomena under scruitiny are best understood as emergent properties of the interaction of relatively simple base processes and as such cannot be (adequately) captured though explicit, higher-level rules. Allowing a neural network to "learn" the solution and then studying its behavior is thought to bring one closer to a description of the actual problem.

Problems with this approach have been noted since the begining, however. For one thing, the architecture of a given network is known to have consequences for its ability to solve particular problems. Though subsequent research has identified a number of particular pitfalls, it is still not entirely clear which architectures are appropriate to which problems

and why. Another problem is that networks are not guaranteed to find optimal solutions. Networks are trained using gradient descent algorithms over the error function. That is, the network is given an input, produces an output, and then its weights are adjusted in directions implied by the error between the actual output and expected target. It proceeds stepwise in this fashion until the error can no longer be reduced. The danger is that the error function may be highly complex, involving several "solutions" in the form of *local minima* - regions in which no small change to the error function will result in improvement, but which are nevertheless suboptimal. Again, mildly successful solutions have been proposed (usually involving adjustments to a learning rate), but no solution that guarantees success has yet been found. Most importantly, however, neural networks lack what is often called "semantic transparency." The fact that a neural network appears to have found a solution to its problem says essentially nothing about *how* it solved it - what kind of generalization it used. This is of particular concern because it is sometimes possible that a network simply "memorizes" input patterns without reaching any kind of generalization at all. Trying to obtain transparency from networks is the largest problem that advocates of connectionist approaches face, and to date no adequate solution has been found. It had been proposed, for example, that hidden layer activations can be recorded and subjected to principal components analysis to detect any trends, but this has the feel of circularity. A neural network is itself a kind of principal components analysis machine; this solution is not unlike making one neural network responsible for explaining the output of another. The alternative - simply studying the inputs the network successfully classifies and trying to infer common characteristics - has the feel of doing the work the network has already done by hand, raising questions of usefulness.

Parallel Observation

The famous success of connectionist networks as industrial tools, however, makes it inadvisable to give up on them completely. Clearly they have something to offer - but more studies will need to be done before they are wholly adequate as research tools. One avenue that can and should be pursued might be termed *parallel observation*. This involves running traditional approaches along with connectionist approaches and comparing the results of the two - specifically observing where the two deviate. Such an approach should at the very least help to shed light on what kinds of tasks are appropriate for neural networks.

Proposal

The purpose of this study is to perform a kind of parallel observation on the performance of a neural network in terms of mutual information. Specifically, it aims to study how good a predictor pointwise mutual information scores are of the output of a network trained to predict spaces in unsegmented streams of (English) letters.

Methods

Chapter one of a children's story - A. A. Milne's "The Red House Mystery" (obtained from Project Gutenberg) - was segmented into three- and four- letter patterns paired with the a representation of the spaces observed for that pattern in the text. Each member of the pair was presented as a list. Punctuation was removed from the input and capital letters were converted to lower case, so the segmentation function operated over an alphabet of precisely 26 symbols. For example, for the sentence "this is a sentence," the first four patterns for the the four-letter patterns would be:

```
[t, h, i, s] -> [1, 0, 0, 0, 1]
[h, i, s, i] -> [0, 0, 0, 1, 0]
[i, s, i, s] -> [0, 0, 1, 0, 0]
[s, i, s, a] -> [0, 1, 0, 1, 1]
```

where 1 represents the presence of a space.[1] The letters were themselves represented as binary sequences of length 26 bits with each bit meant to specify a particular letter. This may seem excessive, but it was deemed necessary to avoid potential issues with crosstalk. All letters were represented with equal potential saliency. In all, approximately 10,000 such patterns were obtained. It will be noted that input patterns need not and generally do not have unique outputs. The same four-letter sequence of characters will have different spacing requirements depending on how it occurs in the text.

Using a `C++` program, a network was created to handle patterns of each input length. The networks were trained for 1000 epochs on the entire sequence of patterns presented in the same order in which they were taken from the text. This was meant to simulate actual incremental, linear processing of language. The networks themselves consisted of three layers - an input layer, output layer and hidden layer of 98(72), 4(3) and 5(4) nodes respectively. The learning algorithm was simple one-pass backpropagation. For reasons that will be explained in the discussion, no recurrent layer was included.

Outputs from the final epoch were separated according to performance. Patterns which generated hits more than false alarms or misses were grouped together, those that generated false alarms more often than hits or misses were grouped together, etc. Hits are understood as correct prediction of a space, false alarms as incorrect prediction of a space, and misses as failure to predict a space when one was required by the target. Patterns were converted into strings with spaces in the appropriate places, and pointwise mutual information scores were calculated between the two segments so separated based on their frequency of occurrence in the text.

Results

An overview of the performance of each network is as follows:

4-Letter Network

```
HITS: 2982.0
```

[1] In actual trials, the spacing patterns were lists of 1, to indicate the presence of a space, and -1 to indicate absence. 0 is used here for readability reasons.

FALSE ALARMS: 768.0
SIMILARITY: 0.79849939976
HIT PERCENTAGE: 0.242754802996
HITS TO FALSE ALARMS: 3.8828125

3-Letter Network

HITS: 5229.0
FALSE ALARMS: 4122.0
SIMILARITY: 0.776607982395
HIT PERCENTAGE: 0.532051282051
HITS TO FALSE ALARMS: 1.26855895197

Due to the inordinate number of input patterns, results are reported here only for the 20 highest scorers in each category - that is, the patterns which were most characteristically hits, false alarms, and misses. Complete results are included as appendices.

3-Letter Hits

PAT	HIT	FA	MISS	MI
he t	18	1	0	-0.6194
s th	18	3	0	0.0425
all	19	9	11	NA
he s	19	10	0	0.3747
e ha	21	0	0	0.5551
een	21	0	0	NA
he w	21	0	0	1.2458
for	25	4	24	NA
t th	25	1	0	-0.0334
e th	27	4	0	-0.2616
rey	28	2	0	NA
aud	29	0	1	NA
he h	29	0	0	0.4396
his	33	1	20	NA
ere	42	14	1	NA
aid	42	1	0	NA
hat	43	11	4	NA
was	46	0	43	NA
her	50	47	36	NA
and	52	1	47	NA

4-Letter Hits

PAT	HIT	FA	MISS	MI
d aud	11	0	0	3.1344
he ho	11	0	0	2.6618
he ma	11	0	0	2.7459
r the	11	0	0	-0.117
t and	11	0	0	1.2217
r mar	12	0	0	3.6713
s the	12	1	0	0.0048
d the	12	0	0	0.5481
mr ma	12	0	0	5.9811
the t	12	0	0	-0.495
aid a	13	0	0	1.8651
the m	13	1	0	1.5424
f the	14	0	0	1.7772
of th	14	0	0	3.0638
t was	14	0	0	1.7740
ing t	15	0	0	0.9377
e was	15	0	0	1.3916
in th	16	0	0	1.5170
t the	18	1	0	0.1683

4-Letter False Alarms

PAT	HIT	FA	MISS	MI
ent l	0	5	0	1.9914
hi ch	0	5	0	4.8541
y sel	0	5	0	4.2711
ny ea	0	6	0	4.7372
lit t	0	6	0	3.4902
nt ha	0	6	0	1.9293
eny e	0	6	0	3.0082
he nh	0	6	0	3.1229
th an	3	7	0	1.5529
in to	1	7	0	2.0405
of fi	0	7	0	5.1179
he ar	2	8	0	2.1229
a bou	0	8	0	3.5909
eof f	0	8	0	4.3697
cay l	0	10	0	4.6691
fif t	0	10	0	3.490
y ley	0	10	0	5.134
yle y	0	10	0	5.271
f rom	0	13	0	5.539
k now	0	13	0	5.872

3-Letter False Alarms

PAT	HIT	FA	MISS	MI
ou t	2	18	0	0.8009
ou l	1	18	0	1.9062
us t	0	18	0	2.0163
re a	6	20	0	1.0529
en t	12	20	0	1.548
ou s	3	21	0	1.4479
ve n	0	21	0	2.1227
e ar	2	22	0	1.4639
h im	0	22	0	3.2693
g ht	0	22	0	5.5113
t hi	7	28	0	2.1117
u dr	0	28	0	4.3380
s te	0	28	0	2.2222
d re	3	34	0	2.5040
s he	7	36	0	0.9429
nt h	2	39	0	2.1986
s ai	0	40	0	3.1962
t ha	4	45	0	2.2594
ot h	0	45	0	3.1831
in g	0	87	0	4.5493
t he	12	178	0	2.7025

4-Letter Misses

PAT	HIT	FA	MISS	MI
aunt	0	0	20	NA
know	0	0	20	NA
when	0	0	22	NA
well	0	0	22	NA
were	0	0	24	NA
othe	0	0	25	NA
been	0	0	26	NA
room	0	0	26	NA
from	0	0	26	NA
mark	0	0	27	NA
door	0	0	27	NA
this	0	0	27	NA
drey	0	0	28	NA
audr	0	0	28	NA
nthe	0	0	29	NA
with	0	0	29	NA
what	0	0	35	NA
here	0	0	39	NA
that	0	0	54	NA
ther	0	0	55	NA
said	0	0	80	NA

3-Letter Misses

PAT	HIT	FA	MISS	MI
bro	0	0	18	NA
ont	0	0	18	NA
wha	0	0	18	NA
o th	0	0	20	0.9935
to t	0	0	20	1.1115
thi	0	0	21	NA
say	13	11	22	NA
now	0	0	22	NA
out	0	0	22	NA
int	0	0	27	NA
hes	0	0	28	NA
n th	0	0	39	1.0927
sai	0	0	40	NA
tha	0	0	42	NA
him	0	0	44	NA
you	0	0	70	NA
she	0	0	70	NA
ing	0	0	79	NA
the	0	0	268	NA

Discussion

Overview

One of the most visible results is the clear difference in performance in terms of hits and false alarms between the 3- and 4-letter networks. The 4-letter network has extremely low rates of both, whereas the 3-letter network is relatively high on both counts. False alarms, in particular, decrease with the additional letter: the 4-letter hits-to-false-alarms ratio is three times higher than that for the 3-letter network, and this with a *significantly* lower hit rate. Without really detailed analysis of the hidden layers it is impossible to say what is responsible - but a good guess would be that it's a "crosstalk" effect. That is, the additional letter seems to be inhibiting the network from responding at all. It seems to have been almost co-opted as a node for forbidding spaces the network is not sure about.

Of course, it should be noted that"crosstalk" effects are generally the result of a network not having enough weight space to build an adequate internal representation of the solution function, loss of accuracy owing to conflicts over which weights should do what, so there is a real sense in which this result is the opposite of what might have been expected. There is also the possibility that this reflects some truth about distributional patterns in English. It might be that trends become clearer at the 4-letter window level that would not be available for a 3-letter window, but not sufficiently clear for generalization. More research is required.

Words as Patterns

Another highly visible result is the network's poor performance on patterns that are equal in size to the input window - that is, in recognizing spaces which are at the edges of the target pattern. Most of the misses for the 3-Letter network and *all* of the misses for the 4-Letter network involved such

patterns.[2]

Since the network is fully connected, it is unlikely that this is an archtectural problem. All nodes at each layer feed inputs to all nodes at the next layer, so spaces at the edges are just as informed by letters in the middle as letters at the edges.

It is, perhaps, not surprising that spaces at the right-hand edge of the input will be hard to detect. Given that the network processes the patterns in the original order, the effect is very much that of moving a window over the input string (1000 times in a row). At the moment of the appearance of a space on the right, the network has little information to warn of its approach. As the space moves toward the center of the pattern, the network accumulates information (in the form of error backpropagation) that it is there. This fails to explain, however, why the network suddenly "forgets" that the space is there at the left edge of the word! And yet, a quick glance at the patterns (especially for the 4-Letter case) will confirm that most of the spaces missed seem to be at either end of the word (most of the cases are stand-alone words in English).

It is worth noting that many of the words are highly frequent words in English (as is generally the case with shorter words cross-linguistically). The network should have had a number of examples of each over the course of the run. This is, therefore, reasonable evidnence that it has not adopted a "strategy" of memorizing words (that is, stable patterns with spaces on either side) as such and placing spaces around them. (Further evidence that this is true can be found by looking at the false alarms patterns. The network seems to miss several extremely frequent words - such as "the" - in the right contexts. It should be noted, however, that the most frequent 3-Letter hit patterns do indeed correspond to full English words.)

Mutual Information

Given the nature of the networks, it would be not unreasonable to expect them to use a kind of indirect mutual information in tackling the problem. One might expect mutual information to "fall out" of the distribution of the patterns. After all, the network will have seen more exemplars of given letters together if they, in fact, occur together frequently in the text, and it will have updated its weights accordingly.

The results do not entirely bear out this expectation, however. Mutual information is indeed lower, as might be expected, between "hit" segments. But it is high indeed between "false alarm" segments. The network, therefore, does not seem to have adopted a strategy of giving preference to areas of low mutual information when deciding to place a space. At the very least, it is fair to say that high mutual information does not *discourage* it from predicting a space.

It is unclear why this should be - but three explanations immediately suggest themselves. First, while mutual information *does* seem to be a reasonable predictor of segmentation for the patterns in question, there is no evidence that this is generally the case. Differences in mutual information between the segments divided by spaces and those not may not be as pronounced in other contexts (in fact, a glance at the full results tables seems to bear this out). This would tend to cast doubt on mutual information as a reliable predictor for segmentation in general - not simply in the case of a neural network's ability to learn segmentation. (However, the fact that mutual information *does* show a high correspondence with spaces (low MI) and absences of spaces (high MI) on the most frequent patterns also suggests that it is highly useful as a method for identifying anchor patterns. It may be that early stages of language acquisition exploit it to subdivide the input and that the learner moves on to other strategies once it has acquired a requisite number of stable patterns.) The second and more obvious explanation is that this is an effect of crosstalk. Other patterns that the network has stored in other contexts are interfereing with its ability to correctly identify the spaces in the false alarm cases.Patterns where this kind of effect seems to be obvious would include "ny ea," "k now,", and "a bou" - all of which contain segements that would be identifiable by MI as useful in other input patterns. A third and less obvious explanation may be that "mutual information," as such, is not being consistently applied. In traditional mutual information applications, mutual information scores are only compared across segments of consistent length. The neural network, however, is presented with patterns where there are (a) variable numbers of spaces, (b) variable segment lenghts that have to be compared. An interesting future study would be to look at cross effects in mutual information between segments of different lengths to attempt to quantify exactly how much of a problem this is (if, indeed, it turns out to be a problem at all).

Conclusion

Neural Networks

Neural networks do not seem to be particularly well suited to segmentation problems. This is evident in the (apparently huge) tradeoff between accuracy and volume of correct answers. Intuitively, this is almost certainly because there is no underlying "function" to learn: patterns of letter occurence are largely determined on higher cognitive levels. Statistics is not sufficient. The evidence from this study also suggests that this may be in part due to architectural considerations. Neural networks are, by their nature, prevented from exploiting certain otherwise available statistical information for reasons which, at this point, remain mysterious and the subject of future inquiry, but which can plausibly be assumed to be related to crosstalk and the fiercely incremental nature of the learning algorithm. Until such time as further studies have been done on what sorts of statistical patterns neural networks are sensitive to (assuming, of course, that they are not simply storing highly frequent patterns, which is also a very real possibility), claims about the distributional properties of texts made on the basis of neural networks should be viewed with an appropriate amount of caution.

[2] A score of NA on mutual information obviously corresponds to the lack of availability of a pattern against which to compare it. This situation arises when there is no segment internal to the pattern presented.

Mutual Information

A surprising conclusion of this study is that mutual information does not seem to be a particularly effective tool for exploitation by neural networks - and possibly by incremental learning algorithms in general. It is not clear why this is so. Some of the problem likely lies in the nature of neural networks themselves - subject, as they are, to crosstalk problems - i.e. architectural limitations on the number of meaningful patterns they can store and exploit. Another possibility lies in the nature of mutual information itself. It is not clear how consistently mutual information can be applied to linguistic distributions. It seems plausible to assume that mutual information is a good predictor of which initial patters a learner will use to "anchor" future learning, but it may not be very useful beyond the earliest stages of learning. It is, in other words, likely to be an effective initial bootstrap but will not be very useful as a predictor of linguistic boundaries in general. Probably learners exploit multiple cues and their interactions in considerably more complex ways than assumed by this study.

Future Directions

The results of this study are obviously incomplete: it would be a mistake to accept any of the conclusions without futher inquiry. That said, the results are strongly suggestive of what might be fruitful areas of such inquiry. First, the suggestion that mutual information applies inconsistently to segments of varying length - while highly plausible - should be thoroughly explored and quantified. Knowledge of specific interaction effects across levels would be useful for future investigations of connectionist performance and may reveal distributional properties of language that have not previously been clear. Second, it will obviously be useful to find out exactly what the effects of adding additional letters to the input patterns are and whether these effects can be usefully quantified cross-linguisticially. Such an experiment would be advised to start with patterns that are the same length as the longest gap between spaces in the original input and "build down" from there. Recurrent neural networks were not used in this study because it was felt that these would amount to simply expanding the window by a single letter, but there is no reason, of course, why this should not also be tested. It may be that building explicit memory into the system would affect the results - though there is no reason to outright expect this to be the case. Another potential problem not touched on until now is the small number of epochs used in the project. Error reports suggest that increasing the number of epochs will not make much difference (the error rate stabilizes soon after the 30th epoch), but again, there is no reason why this cannot be tested for the sake of completeness. Probably the most interesting avenue of future research, however, would be to perform the same experiment on automatically generated texts that can be conrolled for mutual information (and other statistical) effects. This would have the effect of rendering the relevant interactions more transparent than they may have been here. This is, in fact, the next avenue that I will explore.

A Note on Cognitive Plausibility

Connectionist claims of cognitive plausibility should be viewed with a healthy does of skepticism. Simply put, not enough is known about the architecture of the brain and the input signal for real learning to build congitively plausible networks. In addition, connectionist networks have already been shown to differ from the actual human brain in a number of crucial aspects. Most notable is the apparent absence of backpropogation processes in the human mind.

For this reason, no claims of correspondence to the actual language acquisition process are made by this paper. The concepts of learning and exploitation of statistical distributions here are to be thought of on a purely algorithmic, information-theoretic level - not on the level of human cognition. Conclusions reached in this manner make assertions about the distribution of information available for exploitation in natural language - but they do not make any (valid) claims about how such exploitation is actualy accomplished by humans.

Reference list included separately