Exploring Gang Effects By Output Node Similarity In Neural Networks

Paul Rodriguez (prodrigu@cogsci.ucsd.edu)
Department of Cognitive Science, 0515; 9500 Gilman Drive
La Jolla, CA. 92093-0515

Introduction

If a network is trained to minimize mean squared error then it will also learn to maximize the posterior probability of the output given the input (Kuan, et al., 1994). In other words, a network learns to maximize expectations. However, some interesting and relevant psychological modeling are more often concerned with the nature of generalizations across classes of input. In the case of Simple Recurrent Networks (SRNs) these learning effects can occur in temporal context of the feedback connections. For example, Elman (1989) showed how a SRN can learn a simple grammar in a prediction task and the network will learn to produce expectations for all items in a class, even though not all combinations were seen in training. In this work I take a geometric viewpoint toward understanding how a feedforward networks have a "gang" effect due to class similarity in output, and a SRN can have "gang" effects in context. The problem is related to several connectionists models in language that have shown that a network can have generalizations for unseen input. (e.g. Hare, 1990). I will look at generalizations for unseen combinations of bigrams (input-to-output mapping) and trigrams (context and input-to-output mapping), and extend the findings of Bartell, et al.(1993).

Class Effects

If input vectors for a network are encoded with 1-of-n encoding then there is no structure in the input space. In such a case all generalizations that a network learns is based on similarity in the output vectors or similarity in context. In order to draw out the detail of this generalization, it is useful to compare the network to the actual expectations produced by a simple counting procedure. The counting program merely counts for each input the number of times it is trained to produce each output. This gives a forward probability vector of P(output|input), which can be interpreted as a vector in output space.

Furthermore, the program keeps track of the backward probability for each output node, which is the number of time an input is trained to produce that output. This gives a backward probability vector of P(input | output), which is the class conditional probability of the output. The geometric interpretation is that the backward probability is related to the decision planes for that output node.

If a network is trained with simple grammars it can learn to produce expectations for unseen input combinations. The actual bigram probability for unseen combinations is P(output|input)=0. However, by using the geometric interpretations one can account for class effect by comparing the backward probability across all output nodes. Output nodes with similar backward probability vectors will share similar output decision planes. The closer the vectors or the more output nodes in a class, the higher and longer a class effect.

For example, assume a grammar task that uses sequence of verbs and nouns, but during training some verbs are not trained on all possible nouns that can follow. A class effect can be surmised by producing output vectors from counting program and performing a hierarchical clustering. The cluster can be compared to a cluster of output weight vectors. The bigram counting produces output vectors that cluster similarly, but do not give an absolute indication of network performance. However, they give a nice way to formalize the class effect as depending on the distance of the following:

$$P(\text{input=verb}(i) \mid \text{output= noun}(j)) \sim$$
$$P(\text{input=verb}(k) \mid \text{output=noun}(l)),$$

where i,j,k,l index the verb and noun examples that are seen in training, but index combination i,l are the unseen examples, and there is a class effect on output node for noun(l).

An SRN will also produce class effects in which the network produces expectations for a verb-noun sequence that has not been seen in training. One can extend the above to include forward probabilities in context as follows:

$$P(\text{output =noun}(i) \mid \text{input= verb}(j)\ \text{noun}\ (k)) \sim$$
$$P(\text{output=noun}(l) \mid \text{input= verb}(m)\ \text{noun}(n))$$

where the unseen examples are indexed by i,m,n. The geometric interpretation is that similar contexts results in similar hidden unit activations. In conclusion, a SRN does more than learn to maximize expectations and adding a geometric interpretation to probabilities help explain the relevant properties in psychological modeling.

References

Bartell,B.T., Cottrell, G.W., Elman, J.L. (1993) The Role of Input and Target Similarity in Assimilation. *Proceedings of the Tenth Annual Conference of the Cognitive Science Society* (pp. 322-327). Hillsdale, NJ: Lawrence Erlbaum Associates.

Elman, J.L. (1991) Distributed Representations, Simple Recurrent Networks, and Grammatical Structure. *Machine Learning* 7. Kluwer Academic Publishers.

Hare, M., (1990) The Role of Similarity in Hungarian Vowel Harmony: A Connectionist Account. *Connection Science*, 2.

Kuan, C.-M.; Hornik, K.; White, H. (1994)
A convergence result for learning in recurrent neural networks. *Neural Computation*, Vol.6, (no.3).