Modularized Internal Attention Network: first pass draft

Morgan Bryant
Department of Psychology
Stanford University, Stanford, CA, 94305 USA

January 27, 2017

Abstract

I outline a first draft at a novel neural network that uses a hidden attention mechanism to selectively process information within the network. This contrasts with Recurrent Attention Models [cite] by managing attention not in visual glimpses trained with reinforcement learning signals but instead on its internal layers. The *modules* unique to this architecture are differentiable and allow the network to be trained with standard gradient methods, without need for a reinforcement learning signal.

1 Introduction

This kind of modularized internal attention network can, in various versions:

- Make networks faster at learning;
- Speed up computation time;
- More easily transfer;
- shed intuitive light on the processes of the network (but less than might be initially expected);
- Somewhat reconcile formalism and connectionism;

- Potentially indicate a style of model structuring and learning that can match current neural networks as well as provide a model that more realistically mimics neurobiology.

The components that yield these results are the in-network attention selector and the module table. The attention selector is a portion of a layer output within the network that is treated as a command for processing as opposed to a set of parameters for a fixed computation. The module table is a collection of *modules* or small fully-connected layers that act like mini networks by encoding often-repeated operations or simple programs. But, critically, these programs are *hidden*: unlike the programs in [NPI cite, etc], these programs are not explicit about the action that they perform. The hypothesis of this report is that the modularity of the network will encourage a more collaborative and specialized utilization of its resources in parallel, while still preserving the extreme function learning that made neural networks famous by promoting hidden units as opposed to formal symbolic units.

Besides the above primary hypothesis, it might seem intuitive that this kind of network might also transfer better to new tasks or be wholly transplanted into newly initialized networks. Such a claim may be supported by the ability to freeze module layers, zero-out module weights, or simply transfer the module layers.

Note to draft: Another claim might be that this kind of architecture may lend well to recurrent networks, where data can be passed between known modules here and there.

2 Formal Model

The model can be conceived as a collection of pairs of layers, where one layer is a module and the other is a fully connected layer. Call the output of the previous layer the rank-2 matrix $X \in \mathbb{R}^{N \times (B+\theta)}$ for any whole constants N, B, θ with the following characterizations: θ is small and represents the parameters to modules; B is arbitrary and relatively large and is the number of modules; and N is somewhat small and indicates the number of modules that the subsequent layer will use.

Then, there are two approach that may work:

- ***Fully Distributed Approach*** For each $1 <= i <= N$, take $X_i \in \mathbb{R}^{\theta+\mathbb{B}}$ and let $\underline{x}_\theta = X_{i,1:\theta}$, or the first θ entries of X_i, and let $\underline{x}_B =$

$X_{i,\theta+1:\theta+B}$ be the last B entries of X_i. Make $Z_i = \underline{x}_\theta \cdot f(\underline{x}_B))^T$ such that $Z_i \in \mathbb{R}^{B \times \theta}$ and where f is a function that ensures that the vector is normalized and nonnegative, such as a normalize $\circ$ relu or a softmax, and preserves the dimensionality B. The outer product between the $\underline{x}_B$ and $\underline{x}_\theta$ are, intuitively, the weighted selections of where to send the given parameters. Then, make $Y_i = MZ_i$, where M is a rank-3 tensor of the B modules that each sends $\mathbb{R}^\theta \to \mathbb{R}^\phi$ and constitutes the "first" layer of the pair of layers.

The nonlinearity is useful since $\underline{x}_B$ represents the network's choice for which module to channel the parameters $\underline{x}_\theta$, and as such, may be best represented as if they were probabilities.

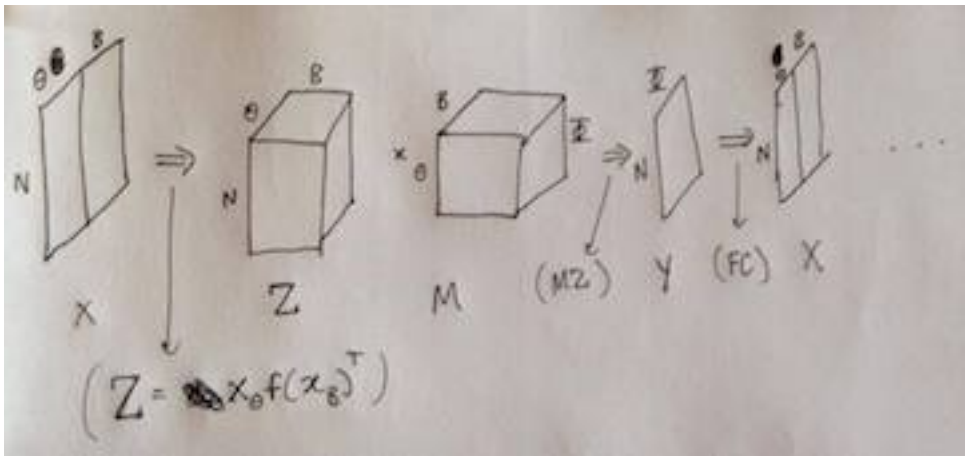

Figure 1: A visualization of the flow of computation used in the distributed approach

- ***Selective Approach*** The approach outlined above takes a distributed approach to selecting attention modules to direct computation flow. Alternatively, if instead modules are uniquely selected, these modules would be [both intuitively inclined towards being less hidden? ... and] more optimized for computation.

  For each $1 <= i <= N$, similarly take $X_i \in \mathbb{R}^{\theta+\mathbb{B}}$ and let $\underline{x}_\theta = X_{i,1:\theta}$ and $\underline{x}_B = X_{i,\theta+1:\theta+B}$. Make $Z_i = \underline{z}_i = f(\underline{x}_\theta)$, where f is a function that normalizes but is not necessarily make nonnegative. Then, make $Y_i = M_j \underline{z}_i$, where $j = argmin_{1<=j<=B}(\underline{x}_B)$.

Notice that this model replaces a rank-3 and rank-2 calculation with a rank-2 and rank-1 product and eliminates an outer product calculation. However, it sacrifices total distributivity, a property that tends to benefit neural networks, as well as complicating the training of unselected modules, since [unsure, but seems perhaps right:] it becomes statistically less likely for the least-selected modules to receive much selection or training after the primary selected modules are trained. [<< − treat as a hypothesis!]

Finally, combine (and flatten) the Y_i into a vector $\underline{y}$ in $\mathbb{R}^{N\times\phi}$, and feed $\underline{y}$ to a fully connected layer $\mathbb{R}^{N\times\phi} \to \mathbb{R}^{N'\times(B'+\theta')}$ (the "second" layer of the pair) with a nonlinearity, the output of which is treated as the input to the next module.

3 Discussion

This model effectively generates (in X) N module computations, where $\underline{x}_\theta$ are the parameters (of fixed size) and $\underline{x}_B$ indicate which module(s) each set of parameters should be fed to. $\underline{x}_\theta$ correspond to standard neural network parameters passed from layer to layer, so these easily fall in the backpropagation paradigm. But unlike other internal attention-selective models as in [REFERENCE2,REFERENCE3], the $\underline{x}_B$ module vector is simply another vector output by the system, orthogonal to the weights they manage, and thus can be trained "independently" of each of the parameters $\underline{x}_\theta$ that are fed to it, but remain sensitive to the types of parameters fed to it. These vectors are easily differentiable and thus can be backpropagated through – and without any special machinery.

Another point of note is that the modules, as the word implies, are building blocks that can be somewhat independently considered. Benefits to this aspect include the following:

- If module learning was a desirable task, a multi-layer network could utilize the same module matrix/tensor for each of the computations and thus provide much more signal to the modules.
- The modules could be incorporated into a recurrent neural network rather naturally, since it implicitly changes the computation it performs based on context, a problem that naive recurrent networks struggle with.

- If certain properties were going to be examined as bottlenecks in a network, any module set could be hand-designed to test certain characteristics (eg, a quick test to determine borderline expressivity could be zeroing out elements of modules by their index) or even wholly transplanted with an alternative system (ie, something more involved than a simple linear operation M such as another neural network). The module is only conditioned on being a function that does $\mathbb{R}^\theta \to \mathbb{R}^\phi$.

- The contents of trained modules can be examined in isolation for what they do: modules can be subjected to tests to determine the function they have – though, since they are presumably hidden unless they are hand-designed, these would probably not yield comprehensible results in isolation except coincidentally.

4 References and Related Literature

Note from Morgan: I'm still working on understanding how latex handles bibliographies... In the meantime:

1. Michael I Jordan and Robert A Jacobs, "A Competitive Modular Connectionist Architecture", MIT, MA, USA, https://papers.nips.cc/paper/430-a-competitive-modular-connectionist-architecture.pdf

2. Marijn F. Stollenga, Jonathan Masci, Faustino Gomez, Juergen Schmidhuber, "Deep Networks with Internal Selective Attention through Feedback Connections", IDSIA, Manno-Lugano, Switzerland

3. Volodymyr Mnih, Nicolas Heess, Alex Graves, Koray Kavukcuoglu, "Recurrent Models of Visual Attention", Google Deepmind, Mountain View, CA, USA

4. TODO: neural program interpreter..?

5. TODO: dram

Possibly helpful todo-read papers:

6. Duda, R.O. & Hart, P.E. (1973) Pattern Classification and Scene Analysis. New York: John Wiley & Sons. Found from (1) above, which justifies a softmax-probabilistic approach to the gating networks.

7. McLachlan, G.J. & Basford, K.E. (1988) Mixture Models: Inference and Applications to Clustering. New York: Marcel Dekker. Found from (1) above, which justifies a softmax-probabilistic approach to the gating networks.

8. Jacobs, R.A., Jordan, M.I., & Barto, A.G. (1991) Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks. Cognitive Science, in press. Found from (1) above; this is an alternative modular architecture on the same what/where tasks.

5 Notes to self - draft!

1/27/17: concerning andrew's recommendations based on the first presentation of this paper on 1/25, he suggested to reconsider the selection gate mechanism, suggesting the concrete distribution, a continuous analogue of discrete variables, may be effective. While I intend to familiarize myself with that distribution, in citation (1) above, a softmax gating mechanism was used as the intuitive method effectively.