A REVIEW OF SPARSE EXPERT MODELS IN DEEP LEARNING

**The Sparsely-Gated Mixture-of-Experts Layer**

## INTRODUCTION

Sparse expert models, of which Mixture-of-Experts (MoE) is the most popular variant, are a thirty-year old concept re-emerging as a popular architecture in deep learning. They are neural networks where a set of the parameters are partitioned into “experts”, each with a unique weight. During training and inference, the models route input examples to specific expert(s) weights. As a result, each example only interacts with a subset of the network parameters (contrasting the usual approach where the entire network is used for each input), keeping the amount of computation small relative to the total model size, and allowing for extremely large, but efficient models.

In early concepts (Jacobs et al. ,1991), the experts defined an entire neural network and the MoE was similar to ensemble methods. However, later and more efficient implementations relied on the idea of MoE as a component of a neural network.

## OUR APPROACH: THE SPARSELY-GATED MIXTURE-OF-EXPERTS LAYER

Our approach to increase model capacity without a proportional increase in computational costs is to introduce a general-purpose neural network component: the Sparsely-Gated Mixture-of-Experts Layer (MoE).

It consists of a set of n “expert networks", each a simple convolutional neural network, and a trainable “gating network”, which selects a sparse combination of the experts to process each input and whose output is a sparse n-dimensional vector.

Let us denote by G(x) and Ei(x) the output of the gating network and the output of the i-th expert network for a given input x, the output y of the MoE module can be written as:

y =∑G(x)iEi(x)

We save computation based on the sparsity of the output of G(x); wherever G(x)i = 0, we do not need to compute Ei(x).

## ROUTING ALGORITHMS

The routing algorithm is a key feature to all sparse expert architectures, determining where to send examples. Typically, the naive routing decision is non-differentiable, because it makes a discrete decision of which experts to select. Shazeer et al. (2017) proposed a differentiable heuristic: the output of the expert computation is weighted by the probability of choosing it and this produces a gradient to the router.

### GATING NETWORK

**Softmax Gating**: A simple choice of non-sparse gating function (Jordan & Jacobs, 1994) is to multiply the input by a trainable weight matrix Wg and then apply the Softmax function. G(x) = Softmax (x · Wg)

**Noisy Top-K Gating**: We add two components to the Softmax gating network: sparsity and noise. Before taking the softmax function, we add tunable Gaussian noise, then keep only the top k values, setting the rest to −∞ (which causes the corresponding gate values to equal 0). The noise term helps with load balancing. The amount of noise per component is controlled by a second trainable weight matrix.

**Balancing expert utilization:** it has been observed that the gating network tends to converge to a state where it always produces large weights for the same few experts. So, we defined the importance of an expert relative to a batch of training examples to be the batch-wise sum of the gate values for that expert, and an additional loss which is added to the overall loss function of the model. This loss is equal to the square of the coefficient of variation of the set of importance values, multiplied by a hand-tuned scaling factor, and encourages all experts to have equal importance. Nevertheless, experts may still receive very different numbers of examples. For example, one expert may receive a few examples with large weights, and another may receive many examples with small weights.

We trained the gating network by simple back-propagation, along with the rest of the model.

## CONCLUSIONS

Sparsity reduces the training and inference costs, resulting in massive models with a better accuracy than their dense counterparts.

Furthermore, sparse expert models more naturally lend themselves to interpretability studies because each input is processed by an identifiable, discrete subset of the model weights (the chosen experts).