Mixture of Experts – Background

Mixture of experts is a way of solving machine learning problems by combining results from several models, which are referred to as experts. There can be many types of experts, such as decision trees, linear models or neural networks. The objective of each expert is typically to solve some subset of the problem space. Combining results from several experts using a combining strategy provides better results than only using a single expert.

The combining strategies are typically additive and often related to the strategies to learn individual experts. For example, in meta-learning algorithm bagging (Breiman, 1994), each expert is learned independently, and the combined result is the average of individual results, whereas in boosting (Schapire & Freund, 2012), experts are learned sequentially depending on the results previous experts produced, and the combined result is the weighted average with corresponding weighting strategy.

In the field of neural networks, the combining strategy is to compute the weighted sum by a gating network. With each expert being a neural network, the whole mixture of experts is also a neural network. The function of the mixture of experts can be formulated as following:

$$\acute{y}(x, w, v) = \sum_z g(x, v)_z \acute{y}_z(x, w_z) \tag{1}$$

where x and $\hat{y}$ are the input and output of mixture of experts, w_z and v are the model parameters of the expert z and the gating network, and $\hat{y}_z$ and g are the functions of the expert m and the gating network.

The gating network outputs a probabilistic distribution vector, so that $\sum_z g(x, v)_z = 1$ and $g(x, v)_z \geq 0$. This is often achieved by applying the so called softmax-function in the end of the gating network. The values to be operated by softmax-function are the unnormalized log probabilities, sometimes they are called logits, and are denoted as the vector $g_{logits}(x, v)$ in following equation:

$$g(x, v)_z = softmax\left(g_{logits}(x, v)\right)_z = \frac{\exp\left(g_{logits}(x, v)_z\right)}{\sum_{\acute{z}} \exp\left(g_{logits}(x, v)_{\acute{z}}\right)} \tag{2}$$

Typically, the softmax-function is used in mixture of experts (Waterhouse, 1998; Jordan & Jacobs, 1993; Shlomo E. Chazan, 2018; Ioannou, et al., 2016), but there are also some alternative functions instead of softmax. In the Google's large mixture of experts (Shazeer, et al., 2017), they used top-k-softmax, where only the top-k logits are turned into non-zero probabilities. In Xu's approach (Xu;Jordan;& Hinton, 1995),they computed the gate values by normalizing multiple parametrized probabilistic densities ϕ, and $g(x, v)_z = \frac{\alpha_z \phi_z(x)}{\sum_{\acute{z}} \alpha_{\acute{z}} \phi_{\acute{z}}(x)}$, where α_z and ϕ_z are parametrized by v_z and the desity ϕ_z is Gaussian with mean and variance as parameters. A function called sparsemax as an alternative of softmax has also been researched (Martins & Astudillo, 2016), but, instead of mixture of experts, they tested them on the final layer of classification and neural attention mechanisms.

Often, we want to estimate the probabilistic distribution $P(Y|X)$ of target data Y conditioned on input data X, regardless what model we are using. If we assume that there is such hidden data Z, that $P(Z|X)$

and $P(Y|X,Z)$ are simpler to estimate than $P(Y|X)$. Then we could estimate $P(Y|X)$ by this following Bayesian equation:

$$P(Y=y|X=x)=\sum_{z}P(Z=z|X=x)P(Y=y|X=x,Z=z) \tag{3}$$

The structure of mixture of experts formed in equation 1 could be treated as the estimators of the equation 3. Instead of estimating, it rather constructs hidden data Z with the probabilities $g(x,v)$ given by the gating network:

$$g(x,v)_z=P(Z=z|X=x) \tag{4}$$

Given gating network is computing the probability of hidden data, we have also experts learned to estimate the target data given hidden data:

$$\hat{y}_z(x,w_z)=P(Y=y|X=x,Z=z) \tag{5}$$

By the term neural network, we referred to the class of models, which are end-to-end differentiable. There are reasons behind experts and gating functions being neural networks. They are possible to be trained using the back-propagation algorithm (Werbos, 1994), jointly if needed. Moreover, while a neural network is often formed as a combination of layers, the mixture of experts could be a layer, only if the backpropagation algorithm is possible to be applied through it. The mixture of experts being neural networks allows computing partial derivatives of the loss with respect to the input of the mixture of experts, hence it allows the gradients to be backpropagated to the previous layers, to be showed later.

Consider a standard neural network, except with a hidden layer being mixture of experts. We denote the input and output of this layer as H_1 and H_2 instead of X and Y. Denote X and Y as the input and output of the neural network. This model assumes the following probabilistic equation about data generating process:

$$P(X,Y,Z,H_1,H_2)=P(Y|H_2)P(H_2|Z,H_1)P(Z|H_1)P(H_2|X)P(X) \tag{6}$$

That is, Y, H_2 and H_1 are modeled as functions of H_2, H_1 and X, where $Y(H_2)$ and $H_1(X)$ are modeled by parts of the standard neural network and $H_2(H_1)$ modeled by the layer of mixture of experts with some intermediate data Z.

We could derive the partial derivatives of the layer output with respect to parameters of experts, parameters of gating network and the layer input:

$$\frac{\partial H_2}{\partial w_z}=\frac{\partial H_2}{\partial H_{2,z}}\frac{\partial H_{2,z}}{\partial w_z}=\frac{\partial \sum_{\acute{z}} G_{\acute{z}} H_{2,\acute{z}}}{\partial H_{2,z}}\frac{\partial H_{2,z}}{\partial w_z}=G_z\frac{\partial H_{2,z}}{\partial w_z} \tag{7}$$

$$\frac{\partial H_2}{\partial v}=\frac{\partial \sum_z G_z H_{2,z}}{\partial v}=\sum\nolimits_z H_{2,z}\frac{\partial G_z}{\partial v} \tag{8}$$

$$\frac{\partial H_2}{\partial H_1} = \frac{\partial \sum_z G_z H_{2,z}}{\partial H_1} = \sum\nolimits_z G_z \frac{\partial H_{2,z}}{\partial H_1} + \sum\nolimits_z H_{2,z} \frac{\partial G_z}{\partial H_1} \quad (9)$$

where notations w_z, v and z remain the same, and G_z and $H_{2,z}$ are the weight and the output of the expert z. The partial derivatives remaining in the equations are from either gating network or experts, so they could be computed.

To apply the back-propagation algorithm, we should compute the partial derivatives of the loss with respect to parameters of experts, the gating network and previous layers. This could be done by the chain rule of derivatives, and all the partial derivatives on the right side of following equations are from either the equation 9 or parts of the standard neural network and could be also computed:

$$\frac{\partial L}{\partial w_z} = \frac{\partial L}{\partial H_2} \frac{\partial H_2}{\partial w_z} \quad (10)$$

$$\frac{\partial L}{\partial v} = \frac{\partial L}{\partial H_2} \frac{\partial H_2}{\partial v} \quad (11)$$

$$\frac{\partial L}{\partial U_{prev}} = \frac{\partial L}{\partial H_2} \frac{\partial H_2}{\partial H_1} \frac{\partial H_1}{\partial U_{prev}} \quad (12)$$

where L is the loss, and U_{prev} is the parameters before the mixture of experts.

The more advanced structure of mixture of experts is by stacking them hierarchically (Ioannou, et al., 2016; Waterhouse, 1998; Jordan & Jacobs, 1993). This often reminds the decision tree, where each data is separated recursively into subspaces by comparing some features against the thresholds. Similarly, in hierarchical mixture of experts, each expert represents a subspace, and each expert has multiple children-experts, which represent even smaller subspaces.

References

Breiman, L. (1994). *Bagging Predictors.*

Ioannou, Y., Robertson, D., Zikic, D., Kontschieder, P., Shotton, J., Brown, M., & Criminisi, A. (2016). Decision Forests, Convolutional Networks and the Models in-Between. *arxiv.*

Jordan, M. I., & Jacobs, R. A. (1993). Hierarchical mixtures of experts and the EM algorithm. *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan).*

Martins, A. F., & Astudillo, R. F. (2016). From Softmax to Sparsemax:A Sparse Model of Attention and Multi-Label Classification. *Proceedings of the 33 rd International Conference on Machine.*

Schapire, R. E., & Freund, Y. (2012). *Boosting: Foundation and Algorithms.* Cambridge, Massachusetts: The MIT Press.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., & Dean, J. (2017). Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. *ICLR.*

Shlomo E. Chazan, S. G. (2018). *Deep Clustering based on a Mixture of Autoencoders.*

Waterhouse, S. R. (1998). *Classification and Regression using Mixtures of Experts.*

Werbos, P. J. (1994). *The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting.*

Xu, L., Jordan, M. I., & Hinton, G. E. (1995). *An Alternative Model for Mixtures of Experts.*