Heuristics in Neural Networks: A Short Note[1]

This short note supplements the discussion of neural networks that we had at the Machine Learning and Big Data Workshop run by Professor Fatih Guvenen (the discussion led by Cristián Aguilera Arellano). We used the 2nd edition of *"The Elements of Statistical Learning"* (henceforth, ESL) by Trevor Hastie, Robert Tibshirani, and Jerome Friedman and *"Machine Learning: A Probabilistic Perspective"* by Kevin Murphy as main sources. While we were going over the potential issues in training neural networks, we asked if any heuristics could help to choose the starting values of weights and the number of hidden units and layers. In what follows, I briefly address these questions. I build on highly-cited sources published after ESL. To give an idea: Glorot and Bengio (2010) has 14036 citations in Google Scholar, He et al. (2015) — 13438 citations, Goodfellow, Bengio, and Courville (2016)[2] — 32128 citations.[3] For an extended discussion of challenges in neural network optimization, I refer an interested reader to Chapter 8.2 of Goodfellow et al. (2016).

Initial Values of Weights

In ESL, the authors suggest choosing the starting values of weights that are close to zero. Using exact zero weights leads to perfect symmetry (derivatives are zero), and hence the algorithm does not move. On the other hand, using large weights often leads to poor solutions.

Goodfellow et al. (2016) point out that modern initialization strategies are simple and heuristic primarily because neural network optimization is still not yet well understood. Furthermore,

Most initialization strategies are based on achieving some nice properties when the network is initialized. However, we do not have a good understanding of which of these properties are preserved under which circumstances after learning begins to proceed. A further difficulty is that some initial points may be beneficial from the viewpoint of optimization but detrimental from the viewpoint of generalization. Our understanding of how the initial point affects generalization is especially primitive, offering little to no guidance for how to select the initial point. (Goodfellow et al., 2016)

The only property that should be satisfied is the asymmetry between different units. In particular, if two hidden units with a similar activation function are connected to similar inputs, then these units must have different starting parameter values. They suggest randomly drawing initial weights from a high-entropy distribution. More specifically,

[1] Egor Malkov. E-mail: malko017@umn.edu. All errors are my own.

[2] All three authors belong to the team that developed the generative adversarial networks (GAN).

[3] As of October 20, 2021.

We almost always initialize all the weights in the model to values drawn randomly from a Gaussian or uniform distribution. The choice of Gaussian or uniform distribution does not seem to matter much but has not been exhaustively studied. The scale of the initial distribution, however, does have a large effect on both the outcome of the optimization procedure and the ability of the network to generalize. (Goodfellow et al., 2016)

What about heuristics? Crucially, weight initialization procedures depend on the activation function we use. First, consider neural networks with the sigmoid activation function

$$\sigma(v) = \frac{1}{1+\exp(-v)} \tag{1}$$

and the TanH activation function

$$\sigma(v) = \tanh(v) = \frac{2}{1+\exp(-2v)} - 1 \tag{2}$$

One widely used approach is to initialize the weights of a fully connected layer with m inputs to a node (e.g., the number of nodes in the previous layer) and n outputs from a layer (e.g., the number of nodes in the current layer) by sampling each weight from a uniform distribution

$$\omega_{ij} \sim U\left(-\frac{1}{\sqrt{m}}, \frac{1}{\sqrt{m}}\right) \tag{3}$$

Glorot and Bengio (2010) suggest using normalized initialization, which is a standard in neural network applications nowadays (recall the number of citations)

$$\omega_{ij} \sim U\left(-\sqrt{\frac{6}{m+n}}, \sqrt{\frac{6}{m+n}}\right) \tag{4}$$

This initialization procedure is derived under the assumption that the activation function is linear. Therefore, despite many strategies designed for the linear model perform reasonably well on its nonlinear counterparts (Goodfellow et al., 2016), one should still be careful while using it.

Now I turn to the initialization techniques with the ReLU (Rectified Linear Unit) activation function

$$\sigma(v) = \max\{0, v\} \tag{5}$$

for which the linearity assumption certainly does not hold. In fact, the initialization (4) was shown to have troubles with this activation function. He et al. (2015) address this problem and develop a procedure specifically for the ReLU activation function (and its more general form, Parametric Rectified Linear Unit or PReLU).[4] They suggest to draw initial weights from the normal distribution

$$\omega_{ij} \sim \mathcal{N}\left(0, \sqrt{\frac{2}{m}}\right)$$

where m is the number of inputs to a node.

Number of Hidden Units and Layers

In ESL, the authors suggest that more hidden units are better than fewer. A researcher can start with too many hidden units and then use regularization. Typically, the number of hidden units increases with the number of inputs and the size of the training sample. The number of hidden layers is selected based on background knowledge and experimentation.

Despite using the “background knowledge and experimentation” is still the most widely used way to select the number of hidden units and layers, I refer an interested reader to Stathakis (2009) who discusses various methods including trial and error, heuristic search, exhaustive search, and pruning and constructive algorithms. Furthermore, he refers to a theoretical result about the “optimal” number of hidden units in a neural network with two hidden layers.

References

Glorot, Xavier and Yoshua Bengio (2010). “Understanding the Difficulty of Training Deep Feedforward Neural Networks.” in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, 249-256.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). “Deep Learning.” *MIT Press.*

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification.” *arXiv:1502.01852.*

Murphy, Kevin (2012). “Machine Learning: A Probabilistic Perspective.” *MIT Press.*

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). “The Elements of Statistical Learning.” *Springer.*

Stathakis, Demetris (2009). “How Many Hidden Layers and Nodes?” *International Journal of Remote Sensing*, 30(8), 2133-2147.

[4] In this paper, the authors achieve 4.94% top-5 test error on the ImageNet 2012 classification dataset, and were the first to surpass human-level performance (5.1%) on this visual recognition challenge.