Andriy Sheptunov

1. Forward Propagation and General Definitions

Take a multi-layer neural net, and consider its forward propagation procedure between any two arbitrary adjacent layers. Let m be the dimension of any output layer, n be the dimension of its preceding layer, and w_{nm} be the weight from neuron of index n from the input layer onto neuron indexed m in the output layer. Let $x_{1\dots n}$ represent neurons in the input layer, and $y_{1\dots m}$ represent neurons in the output layer. Finally, define a conventional sigmoid function ϕ as follows:

$$\phi(z) = \frac{1}{1 + e^{-z}}$$

The sigmoid function is chosen as it is continuous and easily differentiable:

$$\frac{\partial \phi}{\partial z} = \phi(z)\big(1 - \phi(z)\big)$$

We will use this differential later in back propagation.
Using these definitions, the input layer will be of the form

$$\text{Input layer} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Then, the subsequent computed output layer, representing a single step of forward propagation, is as follows:

$$\text{Output layer} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} \phi(\sum w_{i1}x_i) \\ \phi(\sum w_{i2}x_i) \\ \vdots \\ \phi(\sum w_{im}x_i) \end{bmatrix} = \begin{bmatrix} \phi(w_{11}x_1 + w_{21}x_2 + \dots + w_{n1}x_n) \\ \phi(w_{12}x_1 + w_{22}x_2 + \dots + w_{n2}x_n) \\ \vdots \\ \phi(w_{1m}x_1 + w_{2m}x_2 + \dots + w_{nm}x_n) \end{bmatrix}$$

2. Back Propagation

Consider the process of supervised training for a neural net. Define a training vector with dimension m as the set of desired outputs:

$$\text{Training set} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

and the actual data as the set of calculated outputs:

$$\text{Calculated output} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_m \end{bmatrix}$$

Then, we can use a weighted squared difference loss function to calculate the loss (error) from our output:

$$\begin{aligned} \text{Loss} = L &= \frac{1}{2}(y_1 - \hat{y}_1)^2 + \frac{1}{2}(y_2 - \hat{y}_2)^2 + \ldots + \frac{1}{2}(y_m - \hat{y}_m)^2 \\ &= \sum_{i=1}^{m} \frac{1}{2}(y_i - \hat{y}_i)^2 \end{aligned}$$

Note that $\frac{1}{2}$ is present for ease of differentiation (as is customary) and have no substantial effect.

We can unwrap this using our forward propagation definition on $\hat{y}_i$ to get a better understanding of the motivation of back propagation.

$$L = \sum_{i=1}^{m} \frac{1}{2}\left(y_i - \phi(w_{1i}x_1 + w_{2i}x_2 + \ldots + w_{ni}x_n)\right)^2$$

Recall that w_{nm} refers to edge weights between a neuron indexed n in the previous layer and a neuron indexed m in the current (output) layer, and that $x_{1\ldots n}$ refers to activations of neurons indexed $1\ldots n$ in the previous layer.

This exposes the underlying idea that the loss, L, is dependent on each of the weights w in the net. Since $x_{1\ldots n}$ are also values in a layer calculated by applying a sigmoid function to a sum, this loss function is recursive, with a base case at a layer where $x_{1\ldots n}$ represent the global input values to the neural net.

The goal of back propagation is to find $\frac{\partial L}{\partial w}$, or the rate of change of the loss with respect to any given weight in the neural net. This differential effectively tells us how much (denoted by magnitude), and in which direction (denoted by sign) to change any given weight to change the loss in some specific way. The ultimate goal is to minimize loss by changing weights in the direction opposite to $\frac{\partial L}{\partial w}$ for each respective weight.

The next task is to compute $\frac{\partial L}{\partial w}$. Let's choose an arbitrary w, such as w_{nm} to represent the calculation of $\frac{\partial L}{\partial w}$ for some weight in the last (output) layer of the neural net. Recall that w_{nm} is the edge weight on the connection between the last neuron in the previous layer and the last neuron in the final layer. Start by writing out the general expression for $\frac{\partial L}{\partial w_{nm}}$:

$$\frac{\partial L}{\partial w_{nm}} = \frac{\partial}{\partial w_{nm}} \sum_{i=1}^{m} \frac{1}{2}(y_i - \hat{y}_i)^2$$

We apply the sum rule for differentiation in order to move the differential inside the sum:

$$= \sum_{i=1}^{m} \frac{\partial}{\partial w_{nm}} \frac{1}{2}(y_i - \hat{y}_i)^2$$

We can now apply product rule and chain rule:

$$= \sum_{i=1}^{m} (y_i - \hat{y}_i) \frac{\partial}{\partial w_{nm}} (y_i - \hat{y}_i)$$

We can apply the sum rule for differentiation again to move the differential inside the difference between the training and calculated data, using the fact that y_i is constant to eliminate it.

$$= \sum_{i=1}^{m} (y_i - \hat{y}_i)(-\frac{\partial}{\partial w_{nm}} \hat{y}_i)$$

$$= \sum_{i=1}^{m} (\hat{y}_i - y_i)(\frac{\partial}{\partial w_{nm}} \hat{y}_i)$$

We can expand $\hat{y}_i$ using its definition:

$$= \sum_{i=1}^{m} (\hat{y}_i - y_i)\Big(\frac{\partial}{\partial w_{nm}} \phi(w_{1i}x_1 + w_{2i}x_2 + \ldots + w_{ni}x_n)\Big)$$

For clarity, let us focus specifically on

$$\frac{\partial}{\partial w_{nm}} \phi(w_{1i}x_1 + w_{2i}x_2 + \ldots + w_{ni}x_n)$$

To continue, we must perform chain rule on ϕ (recall that $\phi'(z) = \phi(z)\big(1 - \phi(z)\big)$):

$$= \phi(w_{1i}x_1 + w_{2i}x_2 + \ldots + w_{ni}x_n)\big(1 - \phi(w_{1i}x_1 + w_{2i}x_2 + \ldots + w_{ni}x_n)\big) \frac{\partial}{\partial w_{nm}} (w_{1i}x_1 + w_{2i}x_2 + \ldots + w_{ni}x_n)$$

This collapses simply to

$$= \hat{y}_i(1 - \hat{y}_i)\frac{\partial}{\partial w_{nm}}(w_{1i}x_1 + w_{2i}x_2 + \ldots + w_{ni}x_n)$$

Once again, for clarity, let us focus specifically on

$$\frac{\partial}{\partial w_{nm}}(w_{1i}x_1 + w_{2i}x_2 + \ldots + w_{ni}x_n)$$

We can apply sum rule once again to distribute in the differential, noting that $w_{ai}x_a$ are constant under differentiation unless $a = n$, in order to eliminate unnecessary terms:

$$= \frac{\partial}{\partial w_{nm}} w_{ni}x_n$$

We can now compose the entire function back together to proceed:

$$\sum_{i=1}^{m} (\hat{y}_i - y_i) \cdot \hat{y}_i \cdot (1 - \hat{y}_i) \cdot \frac{\partial}{\partial w_{nm}} w_{ni}x_n$$

This tells us $\frac{\partial L}{\partial w}$ with respect to any weight on an edge directed into the output layer of the net.

More generally, we have

$$\boxed{\frac{\partial L}{\partial w} = \sum_{i=1}^{m} (\hat{y}_i - y_i) \cdot \hat{y}_i \cdot (1 - \hat{y}_i) \cdot \frac{\partial}{\partial w} w_{ni}x_n}$$