Himesh Bhatia

Neural Networks

Neural networks consist of either neurons or perceptrons that are connected in a series of layers. These neural networks are models of the human brain, which consists of trillions of neurons that fire together. In order to understand neural networks, we first have to ask what is a perceptron and a neuron.

1 Perceptrons and Neurons

1.1 Perceptrons

Perceptrons have several inputs and produces a binary output. Let $\mathbb{X} \subseteq \mathbb{R}$ be the space of input signals, and $\mathbb{Y} \subseteq \mathbb{R}$ be the space of output signals. Suppose a perceptron has n inputs, $\{x_k\}_{k=1}^n$. Each input has an associated weight, $\{w_k\}_{k=1}^n \subset \mathbb{R}$. Then the output of the perceptron, y, is defined as

$$y = \begin{cases} 1 & \text{if } \sum_{k=1}^n x_k w_k > t, \\ 0 & \text{if } \sum_{k=1}^n x_k w_k \leq t, \end{cases}$$

where $t \in \mathbb{R}$ is the defined as the threshold of the perceptron. However, having the output as either a 0 or 1 is restrictive, so we can expand the model of the perceptron to a neuron.

1.2 Neurons

Neurons also take several inputs, but instead of producing a binary output, it produces a real number between $[0,1]$. Suppose a neuron has n inputs, $\{x_k\}_{k=1}^n$. Each input has an associated weight, $\{w_k\}_{k=1}^n \subset \mathbb{R}$. The neuron has an associated bias $b \in \mathbb{R}$. Define a sigmoid function $\sigma : \mathbb{R} \to [0,1]$ as a $C^1[0,1]$ function. Then the output or *activation* is

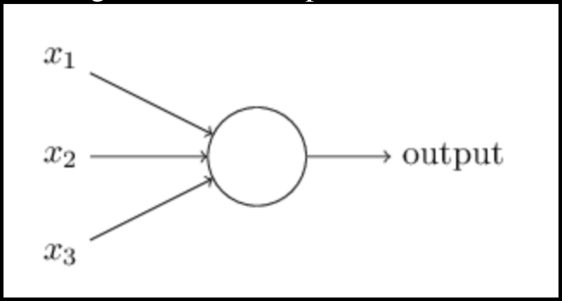
$$y = \sigma\left(\sum_{k=1}^n x_k w_k + b\right)$$

It is important to choose a $C^1[0,1]$ function as the sigmoid function because the derivative of the sigmoid function is used in the equations of backpropagation. Examples of sigmoid functions are $\sigma(x) = \tanh(x)$ or $\sigma(x) = \frac{1}{1+e^{-x}}$.

2 Neural networks

Neural networks have several layers of neurons whose inputs come from the previous layer. Let $\mathbb{X} \subseteq \mathbb{R}^n$ be the input space, $\mathbb{Y} \subseteq \mathbb{R}^m$ be the output space, and suppose we have k layers in our network. In each layer, we have $l_i \in \mathbb{Z}_{>0}$ neurons, $i \in \{1,\ldots,k\}$, where the 1^{st} layer is the input layer, and k^{th} layer is the output layer.

Figure 1: An example of a neuron



Let α_i^j be the activation of the i^{th} neuron in the $j^{th} \in \{2,\dots,k\}$ layer. Let us define $w_{h,i}^j \in \mathbb{R}$ be the weight of the connection from the h^{th} neuron from the $(j-1)^{th}$ layer to the i^{th} neuron in the j^{th} layer. Let $b_i^j \in \mathbb{R}$ be the bias associated with the i^{th} neuron in the j^{th} layer. Then the activation of the neuron is

$$\alpha_i^j = \sigma\left(\sum_{h=1}^{l_{j-1}} w_{h,i}^j \alpha_h^{j-1} + b_i^j\right) \tag{1}$$

Note that we can simplify these definitions by using vectors and matrices. Let $\underline{\alpha^j} = \left(\alpha_1^j,\dots,\alpha_{l_j}^j\right)^T$ be the activation vector of the j^{th} layer. Let $\underline{\underline{W^j}} = \left[w_{h,i}^j\right]_{h,i}^T$ be the matrix of the weights of the connections from the $(j-1)^{th}$ to the j^{th} layer. Let $\underline{b^j} = \left(b_1^j,\dots,b_{l_j}^j\right)^T$ be the bias vector of the j^{th} layer. Then equation 1 becomes

$$\underline{\alpha^j} = \sigma\left(\underline{\underline{W^j}}\,\underline{\alpha^{j-1}} + \underline{b^j}\right)$$

Remark that we set $\underline{\alpha^1} = \underline{x}$, where $\underline{x} \in \mathbb{R}^n$ is the input vector. We will also denote

$$z_i^j = \sum_{h=1}^{l_{j-1}} w_{h,i}^j \alpha_h^{j-1} + b_i^j$$

When constructing a neural network, the weights and the biases are the parameters that can be fine tuned to optimize the neural network. I have several ideas to find the optimal parameters, specifically the idea that we try to apply Bayesian analysis or control theory. We could also try to use Fourier analysis. However, to proceed, we need to define how well the neural network performs. For this, we need to define a cost function.

Let $\underline{a} \in \mathbb{Y}$ be the desired output for some given input $\underline{x}$. We have call $(\underline{x}\ , \underline{a})$ the *training data* for the neural network. Then the *cost function* is defined as

$$C : \mathbb{Y} \times \mathbb{Y} \to \mathbb{R}_{\geq 0}$$

Generally, we will use the squared error:

$$C(\underline{a}, \underline{y}) = \|\underline{a} - \underline{y}\|_2$$

or the cross-entropy function:

$$C(\underline{a}, \underline{y}) = -\sum_{j=1}^{m} a_m \ln(y_m)\ + (1-a_m)\ln(1-y_m)$$

2.1 Backpropagation equations

The aim of backpropagation equations is to find $\frac{\partial C}{\partial w^j_{h,i}}$ and $\frac{\partial C}{\partial b^j_j}$ so we can minimize the cost function with the parameters we can control. We can do this indirectly by first calculating $\delta^j_i = \frac{\partial C}{\partial z^j_i}$ and related to what we want. Let us first calculate this for the output layer (the k^{th} layer). Note that $y_j = \sigma\left(z^k_j\right)$ and hence

$$\frac{\partial y_j}{\partial z^k_i} = \begin{cases} \sigma'\left(z^k_j\right) & \text{if } j = i, \\ 0 & \text{otherwise} \end{cases}$$

Thus, we have

$$\delta^k_i = \frac{\partial C}{\partial z^k_i}$$
$$\delta^k_i = \sum_{j=1}^{m} \frac{\partial C}{\partial y_j} \frac{\partial y_j}{\partial z^k_i}$$
$$\delta^k_i = \frac{\partial C}{\partial y_i} \sigma'\left(z^k_i\right) \tag{2}$$

Using this, we can find $\delta^j_i = \frac{\partial C}{\partial z^j_i}$ for any $j \in \{1, \ldots, k-1\}$. Note that,

$$z^{j+1}_p = \sum_{h=1}^{l_j} w^{j+1}_{h,p} \alpha^j_h + b^{j+1}_p$$
$$z^{j+1}_p = \sum_{h=1}^{l_j} w^{j+1}_{h,p} \sigma\left(z^j_h\right) + b^{j+1}_p$$
$$\Rightarrow \frac{\partial z^{j+1}_p}{\partial z^j_i} = w^{j+1}_{i,p} \sigma'\left(z^j_i\right)$$

where $z^1_h = x_h$. Hence,

$$\delta^j_i = \frac{\partial C}{\partial z^j_i}$$
$$\delta^j_i = \sum_{p=1}^{m} \frac{\partial C}{\partial z^{j+1}_p} \frac{\partial z^{j+1}_p}{\partial z^j_i}$$
$$\delta^j_i = \sum_{p=1}^{m} \delta^{j+1}_p w^{j+1}_{i,p} \sigma'\left(z^j_i\right) \tag{3}$$

We can related equations 2 and 3 to find $\frac{\partial C}{\partial w^j_{h,i}}$ and $\frac{\partial C}{\partial b^j_i}$. That is

$$\delta^j_i = \frac{\partial C}{\partial z^j_i}$$
$$\delta^j_i = \frac{\partial C}{\partial b^j_i} \frac{\partial b^j_i}{\partial z^j_i}$$
$$\delta^j_i = \frac{\partial C}{\partial b^j_i} \tag{4}$$

and

$$
\begin{aligned}
\frac{\partial C}{\partial w^j_{h,i}} &= \sum_{p=1}^{l_j} \frac{\partial C}{\partial z^j_p}\frac{\partial z^j_p}{\partial w^j_{h,i}} \\
\frac{\partial C}{\partial w^j_{h,i}} &= \frac{\partial C}{\partial z^j_h}\frac{\partial z^j_h}{\partial w^j_{h,i}} \\
\frac{\partial C}{\partial w^j_{h,i}} &= \delta^j_h \alpha^{j-1}_i
\end{aligned} \tag{5}
$$