



#

## Tiny Book of Neural Networks

####

*By : Amila Pathirana*

---

### ##Introduction

---

Neural networks is a pattern recognition concept inspired by the way our brain works. Yet, in order to understand the way the neural networks work, you need no knowledge of the intricate inner workings of the brain. Remember neural networks are inspired by the inner workings of the brain; it is not designed like a brain. Moreover, it is not at all as complicated as a human brain. Neural networks provide the best solutions to many problems where there is no direct mathematical relationships between the input and output and you have lots and lots of labeled data. Classification problems is a classic example that we can use neural networks. Apart from classification, speech recognition, and natural language processing is examples that we can use neural networks effectively. There are numerous other problems that we can use neural networks, but for a beginner, understanding it as the way of pattern recognition as we discussed in the example is a good starting point. Going through a simple example is the best way to understand neural networks. Here is one such example.

---

## Iris Guy

---

Imagine you have set of iris photos. Each photo has a label. That label says which species of iris is in the photo. You looks at all the photos and their labels for some time. After spending some time on it, you get good at identifying iris species by looking at their photos.

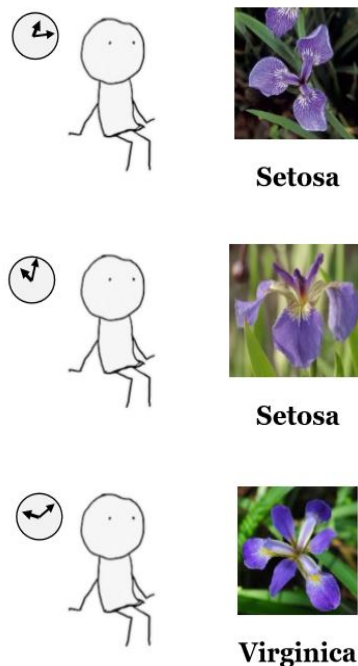


Fig. 1: Training the brain

One day, one of your friends come along, give you a photo of iris, and ask you what it is. Since you are an expert on identifying iris photos, you have no problem identifying what iris species it belongs to.



Fig. 2: Testing the brain

The way neural network solves the problem in the kind of way. In order to get something useful out of it, neural network needs to be trained, just like you trained to identify irises. After the training phase is over, when you see a photo of Setosa iris, you can say it is a photo of Setosa iris with confidence. You may not seen the exact image of a Setosa iris shown to you before. However, you trained your brain with other Setosa iris images in such a way that when you see certain features, you can say it is indeed a Setosa iris. As you can see from the example, neural networks help us when there is no direct mathematical relationships between the input and output and you have lots and lots of labeled data. In this case, the input to the neural network was the pixel values of the photo. And the output was the label like "Setosa", "Virginica", "Versicolor". Clearly, an analytical mathematical relationship between inputs and outputs is nearly an impossibility. Therefore, such problems are ideal problems to be solved using neural networks.

Imagine there is an iris flower garden. In this garden, there are three types of irises. Namely;

- Versicolor iris
- Setosa iris
- Virginica iris

These three types of iris flowers have their own unique characteristics. Figure 3 shows how these three iris species look like.



Fig. 3: Different species of iris in the garden

You received 150 iris flowers from the garden. Fifty from each species. Four features were measured from each flower: the length and the width of the sepals and petals, in centimeters. Those four measured parameters are shown in in Figure 4. In addition, each of these measured set of four parameters is given a label indicating the species of iris flower.

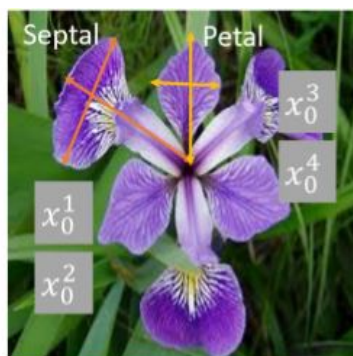


Fig. 4: Petal length, width and septal length, width

Now assume that, you do not have the access to the photo of the irises. You only have access to only the four measured parameters, petal length, petal width, septal length and septal width of iris flowers. Our goal is to develop a neural network that identifies species of any iris from our garden, given the four input parameters. Let's build our first neural network.

## Structure of a simple neural network

We start with the simple neural network shown in Figure 5. In the neural network shown in Figure 5, Layer 0 is the input layer. This is where we input measured parameters. In our case, we have four parameters. Input layer is connected to Layer 1 via set of weights( $w_1$ ). There are five nodes in layer-1. Nodes of Layer 1 is connected to nodes of Layer 2 via a another set of weights( $w_2$ ) , just like the previous layer. Apart from weight matrices, each node except the nodes in the input layer is connected to a variable  $b$ , which is called

bias. Finally, the outputs from the last layer (\$y\_2\$) is the output of the neural network. These outputs of the neural network along with actual observations (\$y\_2\$) is used to calculate the error (\$E\$) of the neural network response.

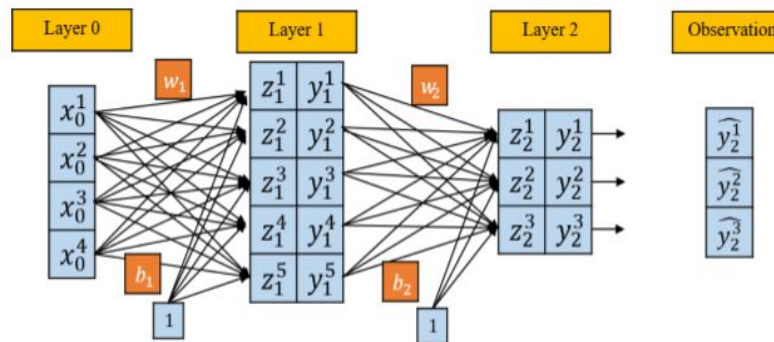


Fig. 5: Structure of a simple neural network

### ###Perceptron

*Perceptron* sounds complex, but it is not. Single node of a layer (except input layer) is called perceptron. What it does is add all the inputs coming into the perceptron to obtain  $z$ , which is an internal variable of the perceptron. Then sent the summation through a function. This function is called activation function. Output of the function is the output of the perceptron. For this example, we will use the sigmoid activation function shown in Equation 1. 
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

When we take whole layer, internal variable  $z$  is a vector. In that case, activation function is applied to each element of the  $z$  vector.

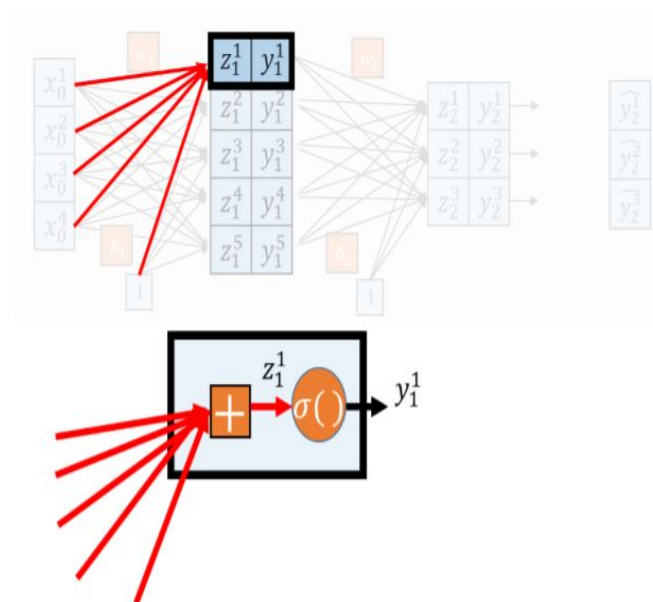


Fig. 6: Perceptron- Add all the inputs and apply activation function

### ##Training the neural network

Neural network to be useful in solving the problem, it needs to be well trained. Finding the best possible values for weight and bias matrices is called training. What I meant by "best possible" is, error of the final decision is minimal. Let us start training our neural network!

###Forward propagation Forward propagation is the first phase of the training process. Goal of the forward propagation step is to find the output  $y_2$ , given the input vector  $x$ , two weight matrices and two bias matrices. At the beginning of the training process, we can initialize the weight matrices with random values. When the neural network received an Input vector, it is multiplied by the weight matrix ( $w_1$ ) and added the biases to obtain the internal variable of Layer 1 ( $z_1$ ).