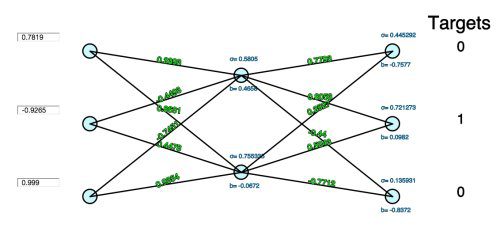
**Day 73 DIY Solution**

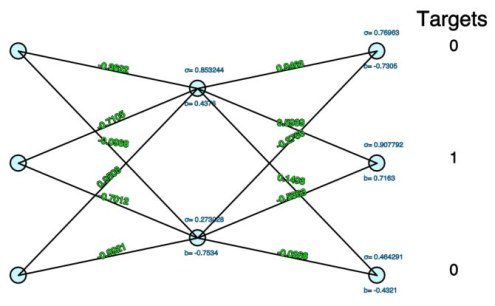
1. **What is backpropagation in neural network?**

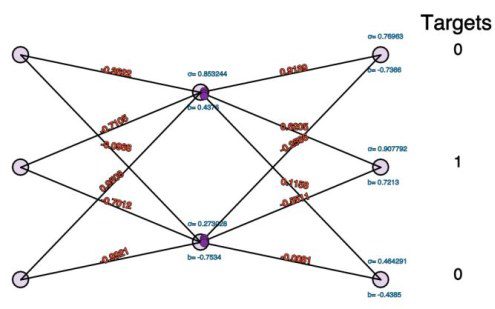
* **Backpropagation** is one of the important concepts of a neural network. Our task is to classify our data best. For this, we have to update the weights of parameter and bias, but how can we do that in a deep neural network? In the linear regression model, we use gradient descent to optimize the parameter. Similarly here we also use gradient descent algorithm using Backpropagation.
* For a single training example, **Backpropagation** algorithm calculates the gradient of the **error function**. Backpropagation can be written as a function of the neural network. Backpropagation algorithms are a set of methods used to efficiently train artificial neural networks following a gradient descent approach which exploits the chain rule.
* The main features of Backpropagation are the iterative, recursive and efficient method through which it calculates the updated weight to improve the network until it is not able to perform the task for which it is being trained. Derivatives of the activation function to be known at network design time is required to Backpropagation.
* The aim of backpropagation (backward pass) is to distribute the total error back to the network so as to update the weights in order to minimize the cost function (loss). The weights are updated in such as way that when the next forward pass utilizes the updated weights, the total error will be reduced by a certain margin (until the minima is reached).

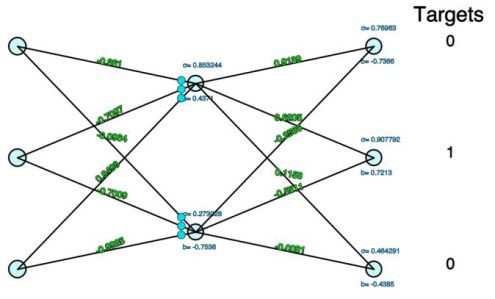
**2. How does a neural network learn?**

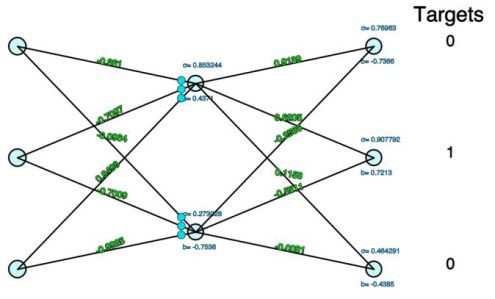
* Neural networks rely on training data to learn and improve their accuracy over time. However, once these learning algorithms are fine-tuned for accuracy, they are powerful tools in computer science and [artificial intelligence](https://www.ibm.com/in-en/cloud/learn/what-is-artificial-intelligence), allowing us to classify and cluster data at a high velocity. Tasks in speech recognition or image recognition can take minutes versus hours when compared to the manual identification by human experts. One of the most well-known neural networks is Google’s search algorithm.
* Neural networks are generating a lot of excitement, as they are quickly proving to be a promising and practical form of machine intelligence. At Fast Forward Labs, we just finished a project researching and building systems that use neural networks for image analysis, as shown in our toy application [Pictograph](http://pictograph.us/). Our companion deep learning report explains this technology in depth and explores applications and opportunities across industries.
* To help understand how neural networks learn, I built a visualization of a network at the neuron level, including animations that show how it learns. If you’re familiar with neural networks or want to follow the rest of the post with a visual cue, please see the interactive visualization [here](http://mwskirpan.com/NN_viz).





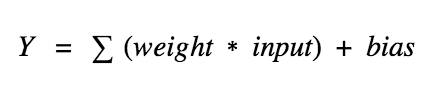




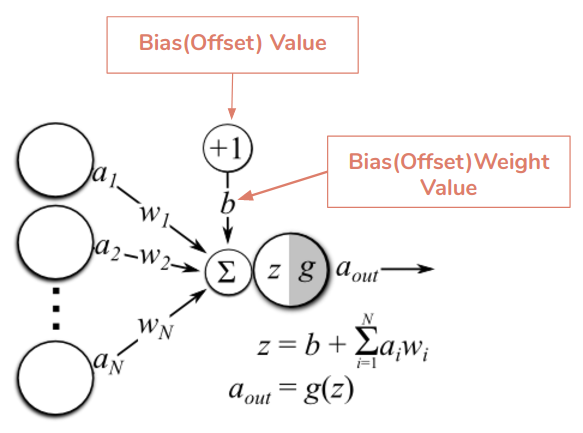


**3.What is weight and bias in artificial neural network?**

* Weights and biases (commonly referred to as *w and b*) are the learnable parameters of a some machine learning models, including neural networks.
* Neurons are the basic units of a neural network. In an ANN, each neuron in a layer is connected to some or all of the neurons in the next layer. When the inputs are transmitted between neurons, the weights are applied to the inputs along with the bias.



* **Weights** control the signal (or the strength of the connection) between two neurons. In other words, a weight decides how much influence the input will have on the output.
* **Biases**, which are constant, are an additional input into the next layer that will always have the value of 1.
* Bias units are not influenced by the previous layer (they do not have any incoming connections) but they do have outgoing connections with their own weights.
* The bias unit guarantees that even when all the inputs are zeros there will still be an activation in the neuron.



**4. How does weight work in neural network?**

* Weights enable the artificial neural network to dial up or dial down connections between neurons. For example, suppose you create an artificial neural network to distinguish among different dog breeds.
* Each neuron in one layer of the neural network may focus on a different characteristic — snout, ears, eyes, tail, size, shape, color, and so on.
* With weighted inputs, the network can increase or decrease the strength of the connection between each neuron in this layer and the neurons in the next layer to place less emphasis on the tail, for example, and more on the size and shape.
* Weights are numeric values that are multiplied by inputs. In backpropagation, they are modified to reduce the loss. In simple words, weights are machine learned values from Neural Networks. They self-adjust depending on the difference between predicted outputs vs training inputs.

**5. What will happen if the learning rate Is set too low or too high?**

* If your learning rate is set too low, **training will progress very slowly** as you are making very tiny updates to the weights in your network.
* However, if your learning rate is set too high, it can cause undesirable divergent behavior in your loss function
* Learning rate (λ) is one such **hyper-parameter** that defines the **adjustment in the weights of our network with respect to the loss gradient descent**. It determines how fast or slow we will move towards the optimal weights.

Here is the algorithm:

* Repeat until convergence {  
     
   Wj = Wj - λ θF(Wj)/θWj  
     
  }

**Learning rate explained through a child’s interaction**

* To understand this better let’s consider an example.
* If a child sees **ten dogs** and all of them are black in color, he might believe that all dogs are black and would consider this as a feature when trying to identify a dog.
* Imagine he’s shown a white dog, and his parents tell him that it’s a dog. With a **desirable learning rate**, he would quickly understand that black color is not an important feature of dogs and would look for another feature.
* But with **a low learning rate**, he would consider the white dog as an outlier and would continue to believe that all dogs are black.
* And if the **learning rate is too high**, he would instantly start to believe that all dogs are white even though he has seen more black dogs than white ones.
* The point is it’s’ really important to **achieve a desirable learning rate** because:
* both low and high learning rates results in wasted time and resources
* A lower learning rate means more training time
* more time results in increased cloud GPU costs
* a higher rate could result in a model that might not be able to predict anything accurately
* A desirable learning rate is one that’s low enough so that the network converges to something useful but high enough so that it can be trained in a reasonable amount of time.

# Tuning the learning rate

* The learning rate is the most important hyper-parameter for tuning neural networks. A good learning rate could be the difference between a model that doesn’t learn anything and a model that presents state-of-the-art results.
* The below diagram demonstrates the different scenarios one can fall into when configuring the learning rate.

