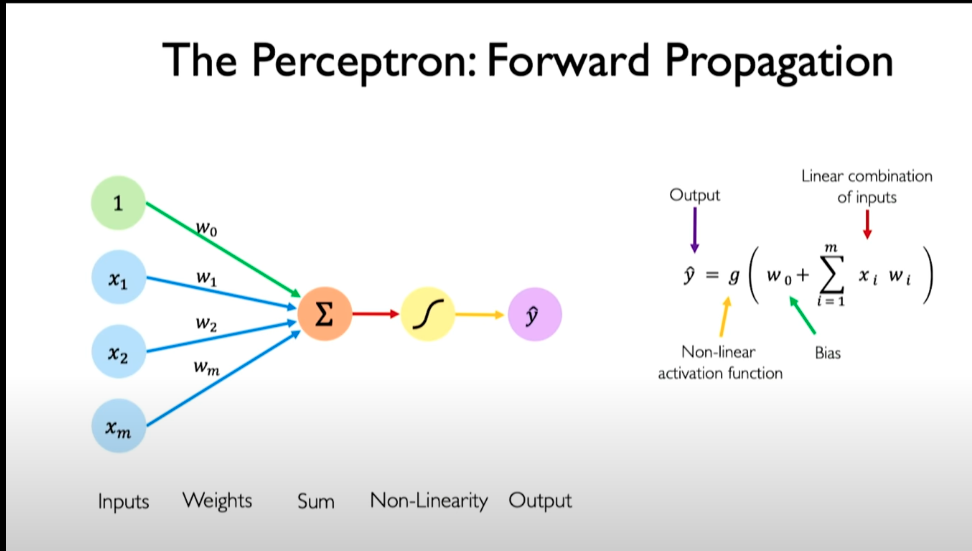
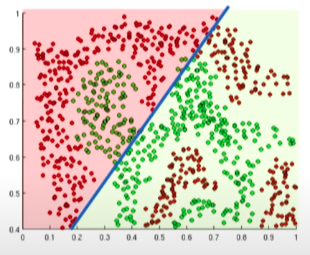
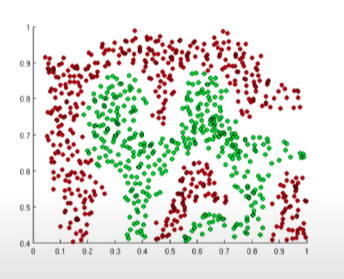
**Deep Learning:**

****

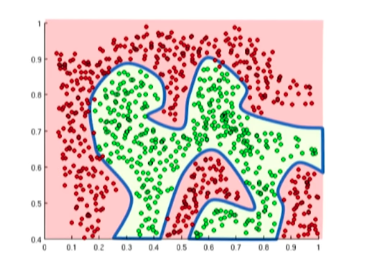
**Activation Function:**

The purpose of the activation function in a neural network is to introduce **NON - LINERITIES** into the network.



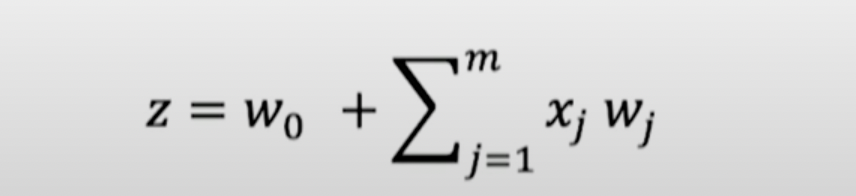
Now, imagine, if we want to distinguish the green and red points. If we have only one line to specify the division, WE Can not. So, this is why we need to introduce non-linearity in our function.

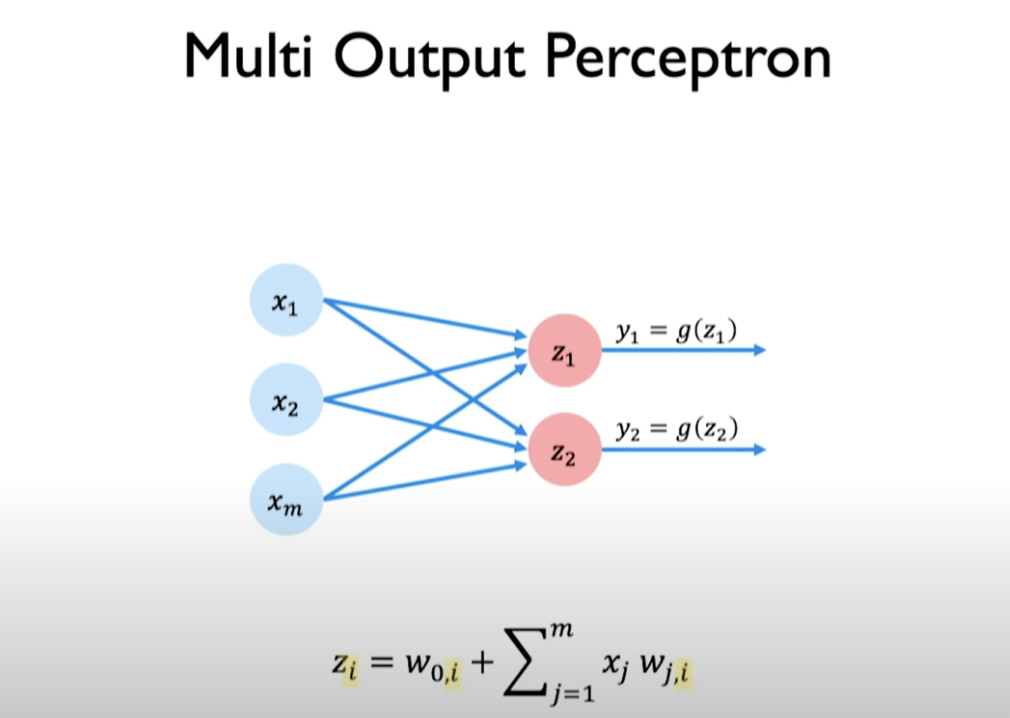
Non- linearities allow us to approximate arbitrarily complex functions and that's what makes NN extremely powerful.

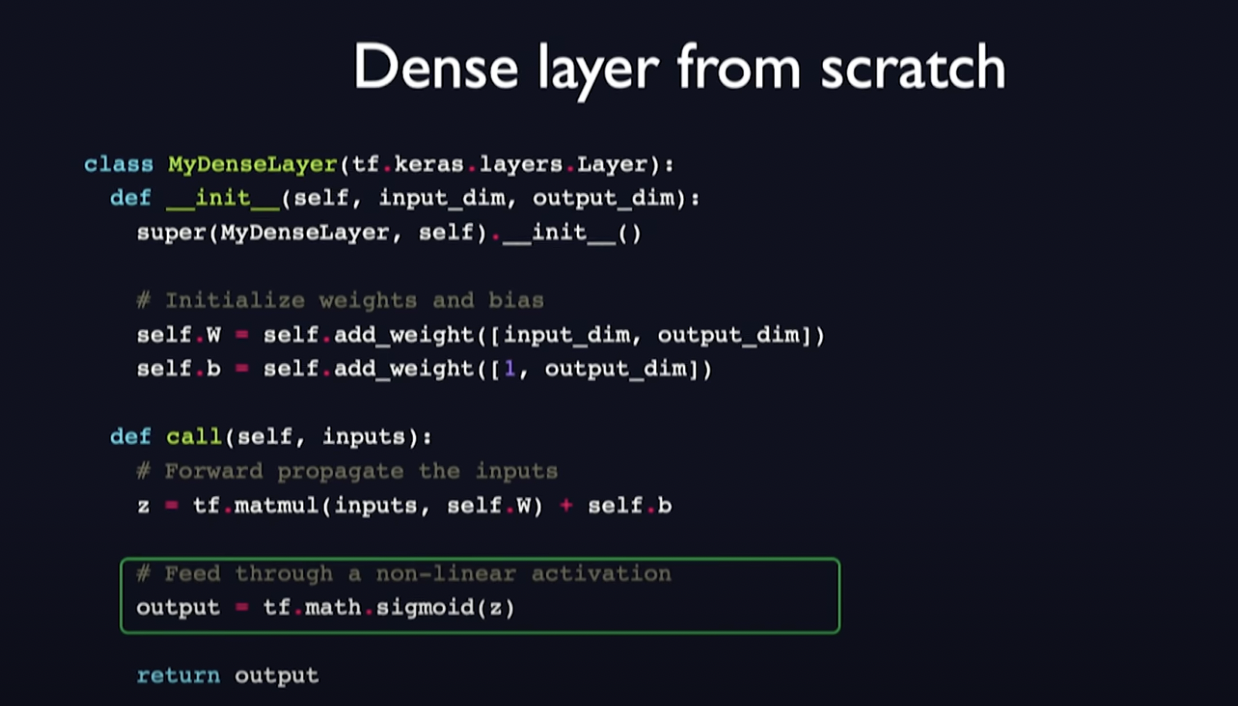


How a perceptron works:

* The dot product of inputs and weights.
* Add a bias
* Apply Non-linearity





Two different neurons would apply their own bias and weights on the inputs. In the above figure, we will get two outputs. Also, all inputs are densely connected to the neurons that why it is called the Dense layers.

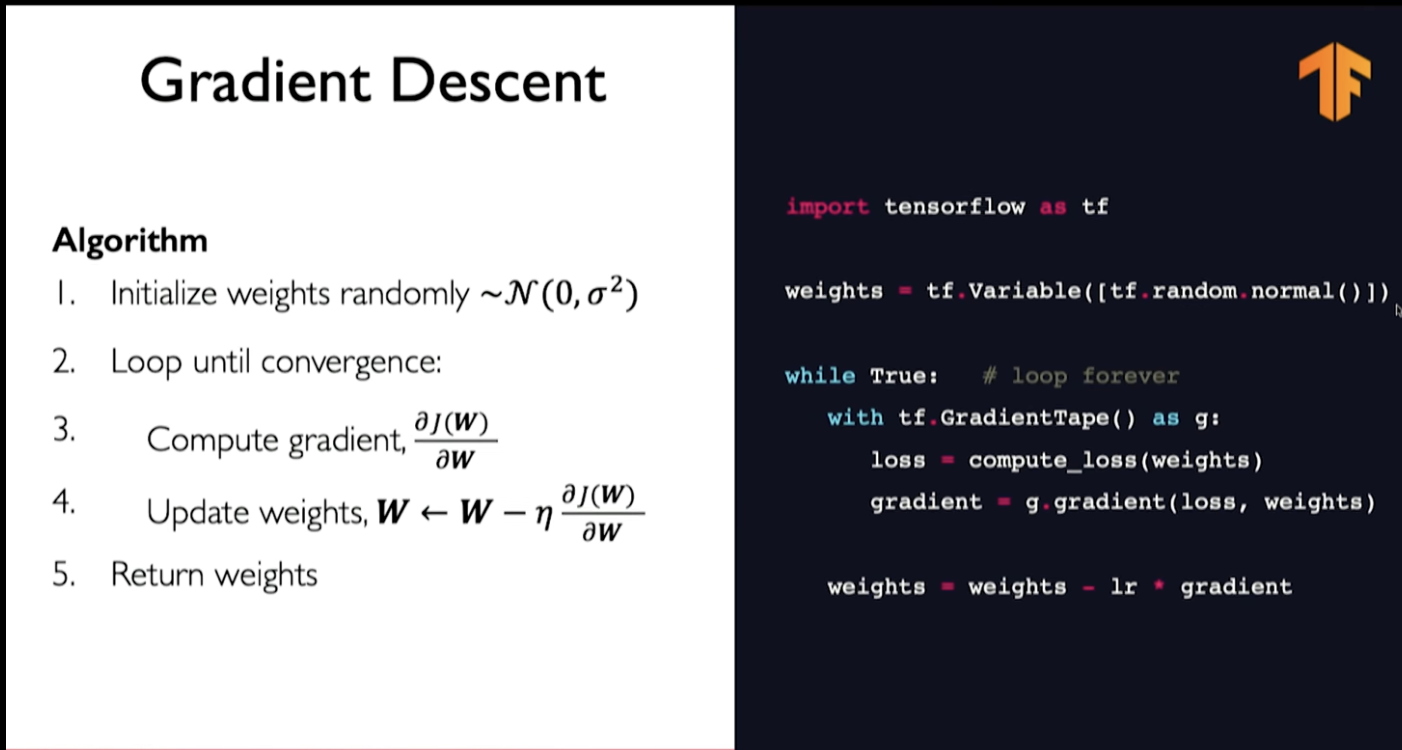
***Import tensorflow as tf***

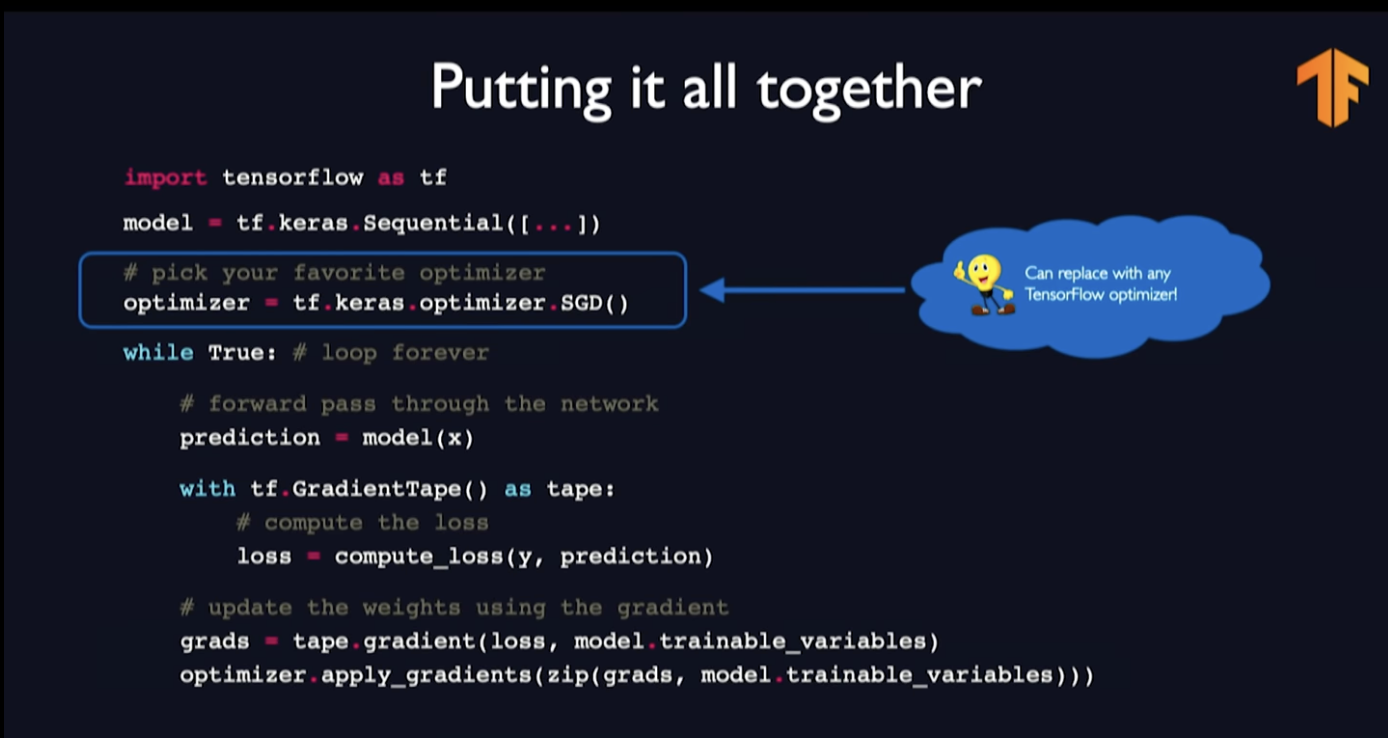
***Layer = tf.keras.layes.Dense(unites = 2)***

**Optimization**

We need to find the weights that achieves the lowest loss.

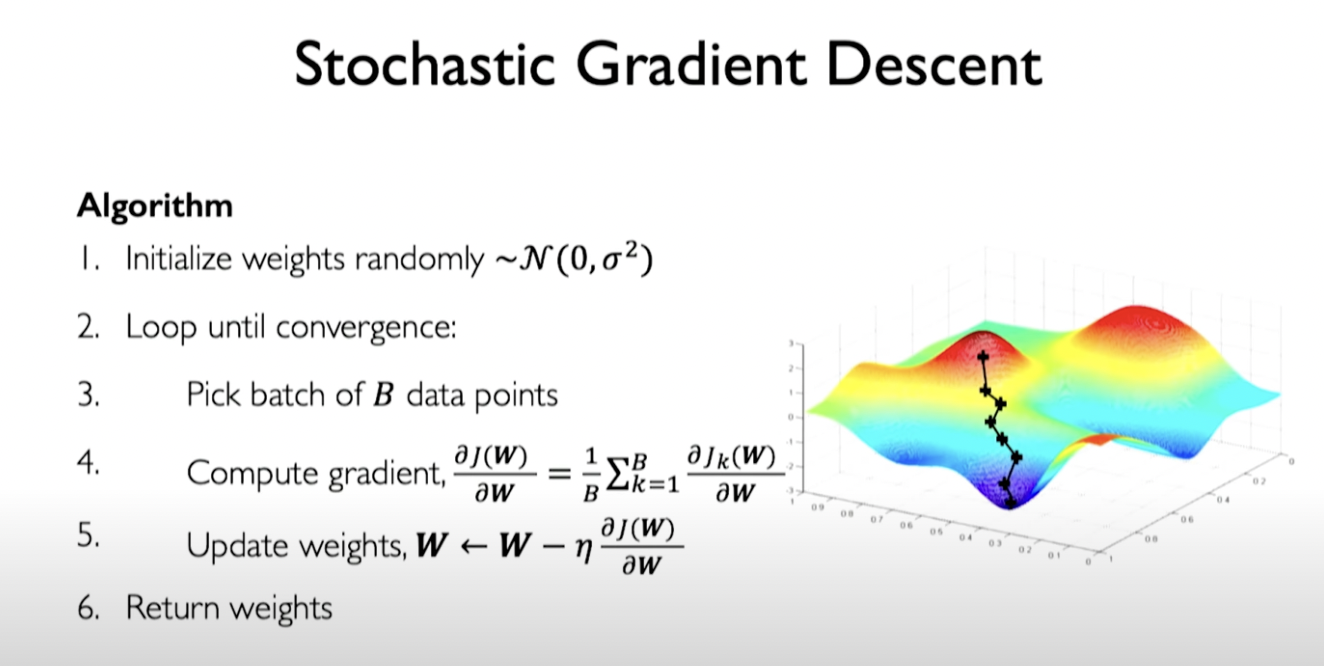
For the optimization, we need to fing the w’s that minimizes the j(w’s), our emi[pirical loss or average loss. Loss function is just a function with takes the inputs with the set of the weights and gives out a single value which can be refer as the error.

we compute the derivative or the gradient of the loss function to tell us the direction we need to go in order to maximize our loss and following that, we take a small step to the opposite direction of that gradient to find the lowest loss. 



The gradient can be very heavy computationally intensive to compute. Here comes the concept SGD.

**SGD**: instead of computing the gradient on entire dataset. We compute it only on a single data point of our whole dataset. It can be sometime noisy. Obviously a single datapoint whom’t be able to showcase the gradient of whole, but we would be able to estimate the grient from one.



Its not going to be the true gradient but an estimate. The major benefit of it is it is computationally less intensive. But we can refer it to be very noisy and stochastic.

So, whats the middle ground. Instead of computing whole dataset or just a single data point. We can compute a subset of the dataset or a mini batch.