**How Deep Neural Networks work (03 July 2020)**

**Basics**

* Made up of input and output. Inner layers made up of neurons and nodes.
* Each neurons are assigned a value from -1 to 1.
* For second layers or neurons, they are connected to all first layers of neurons.
* Nodes are assigned weightage.
* By summing up all the weighted products, we get the values for neurons in the second layers.
* This is applied to all second layers of neurons.

**Step to keep network stable**

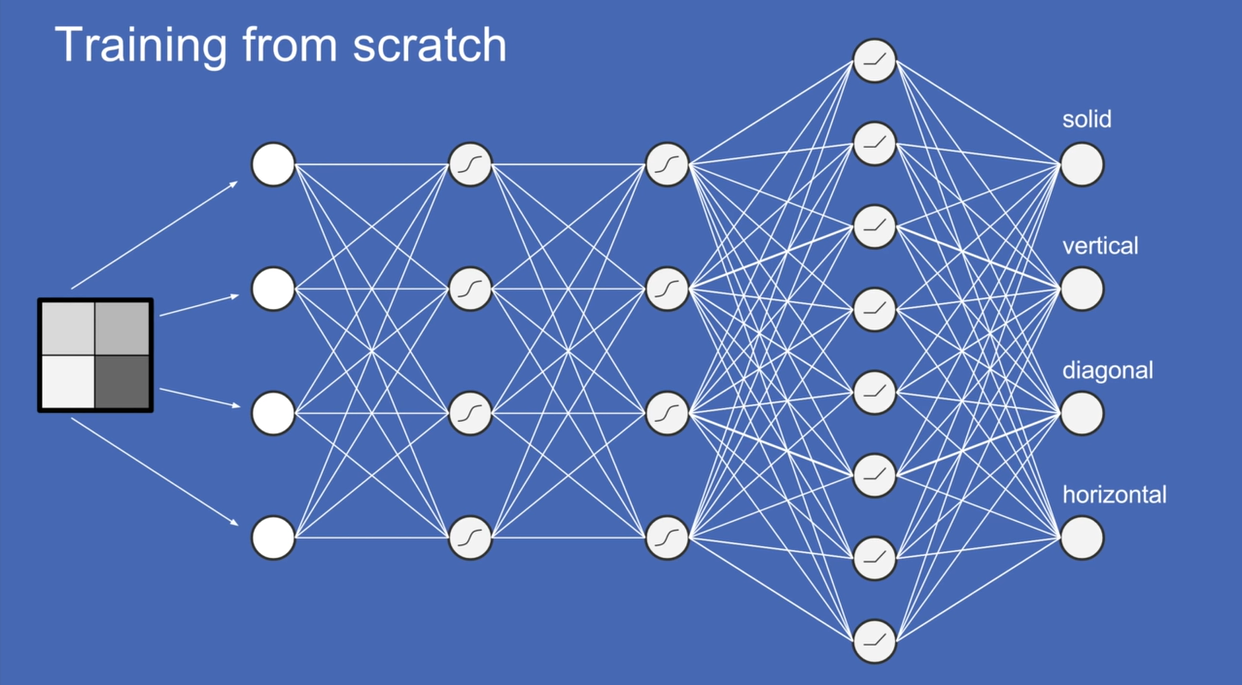
* To keep the neurons values in second layer within the range of input, we implement sigmoid squash functions, so the neural network is bounded and stable.
* All the second layers neurons are differed by weightages.
* Receptive field get more complex when they are combined together.
* Then we add multiple layers and repeat.
* We can add variation in subsequent layers by introducing new functions, rectified linear units. It converts all negative to 0 and keep only 0 to 1 values.
* Then we create an output layer and find the outcome that matches the correct input. We can then mapped out the nodes with correct output.

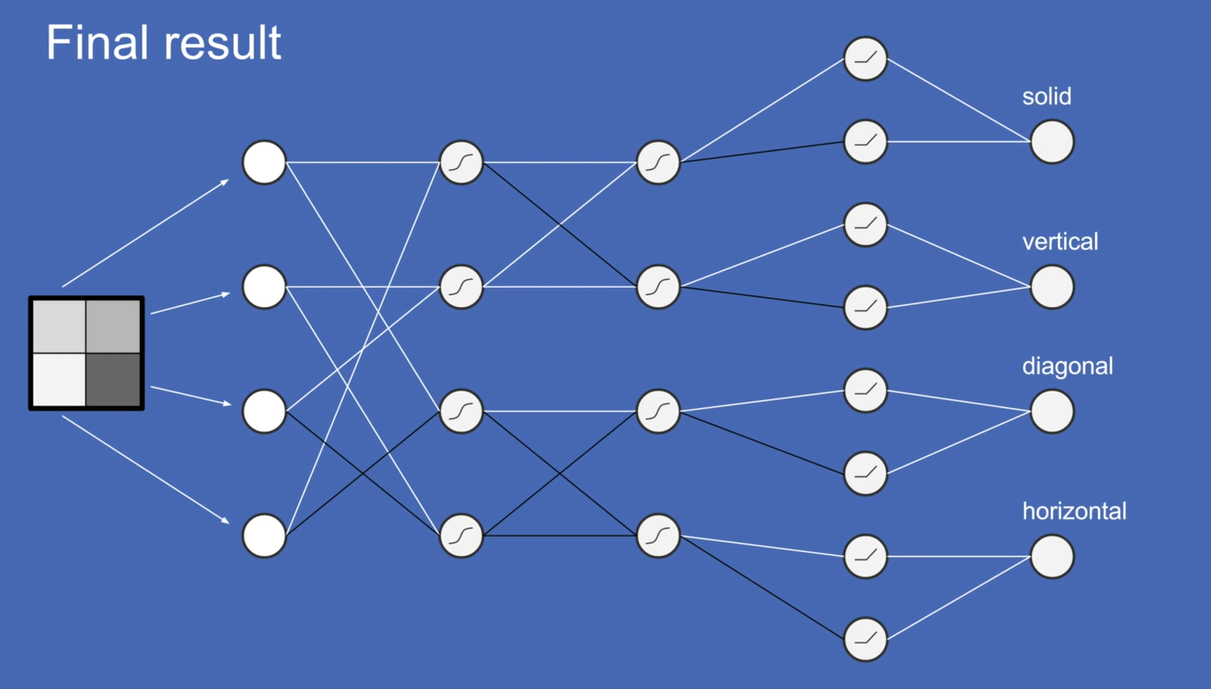
**Training the model = Minimizing the errors**

* Errors in neural network: we need to make the error as low as possible. Error = answer – truth. 0 0 0 1 vs -0.75 0.25 0.5 -0.25
* We can minimize the error by using approach above, however computation is expensive and for many nodes and outputs we need to do it many times.
* Use gradient descent algorithm. de/dW. We can know the direction to tune to move the error down, but have to know the error functions.
* Use chaining, de/dW1 and de/dW2, we can find the change in error when we tweak the weightage in different layers. Then we can get the full slope of the chain
* Back propagation we have to do, (i) weighted connection between two neurons (ii) sums: (iii) Sigmoid function (iii) relu function
* As the final result, many nodes weightage will be very small or zero and we can see the good representation of the network.(Not always the case)

**In Summary:**

* We start with fully connected network and assign random weights.
* Then put in the input with known output, so we can calculate the error.
* We do back propagation and tune the error bit by bit in the correct direction
* Until we get the sparse and receptive fields can be easily understood.





**More to learn:**

* Biased Neurons
* Dropout
* Backpropagation details