**How many nodes/layers?**

Theoretically only need 1 hidden layer for any non-linear problem, but in practice you need more

Look at existing work and start from that

More for more complex problems

Output neurons same as - number of outputs

Number of inputs – number of features? Or is that the parameters?

Each node in network is performing logistic regression, modifying weights of parameters it has as inputs (which are outputs from previous nodes, or features) which form an equation, which provides an output from the parameter values given

**Terminology**

Precision: accuracy of data

= (TruePositives) / ((TruePositive + FalsePositives)

Recall: ratio of true positives to all that were classified correctly

=

F1 score: 2 \* (Precision \* Recall) / (Precision + Recall)

-Better metric than accuracy: if positive occur at a low rate, and model marks everything as negative, it may have a high accuracy despite having a 100% false negative rate. Accuracy not really a meaningful number by itself

Specificity = true negative rate = TrueNegative / (TrueNegative + FalsePositive)

**ROC Curves**

Measure areas under curve: 1 is a perfect prediction, going down to 0.5 which is just random predictions

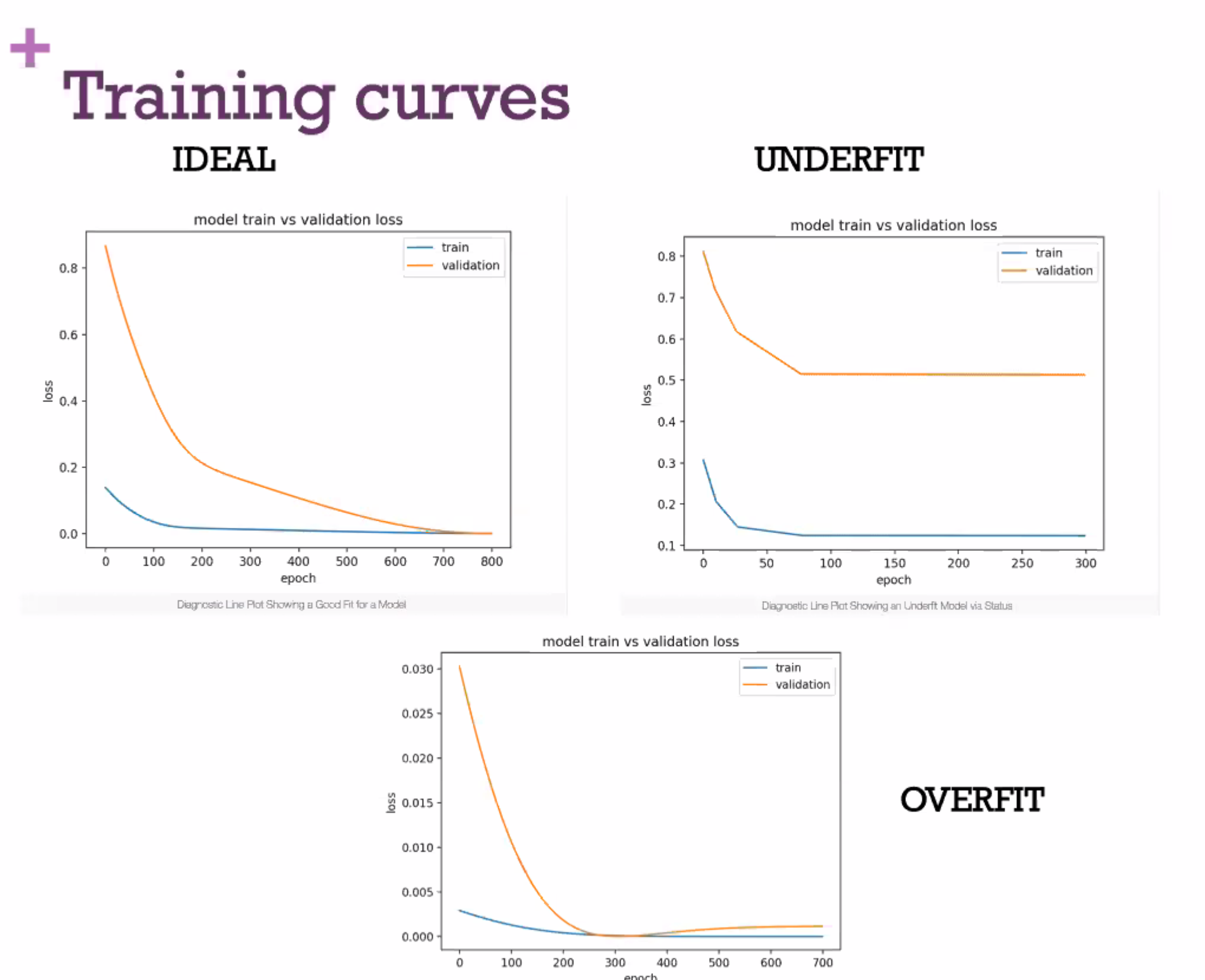
**Underfitting**

Inaccurate model tht diesn’t capture underlying trend of data

Occurs when model shows low variance (all data points close to mean) but high bias

**Overfitting**

Fits data too well, not a general model that works with new data



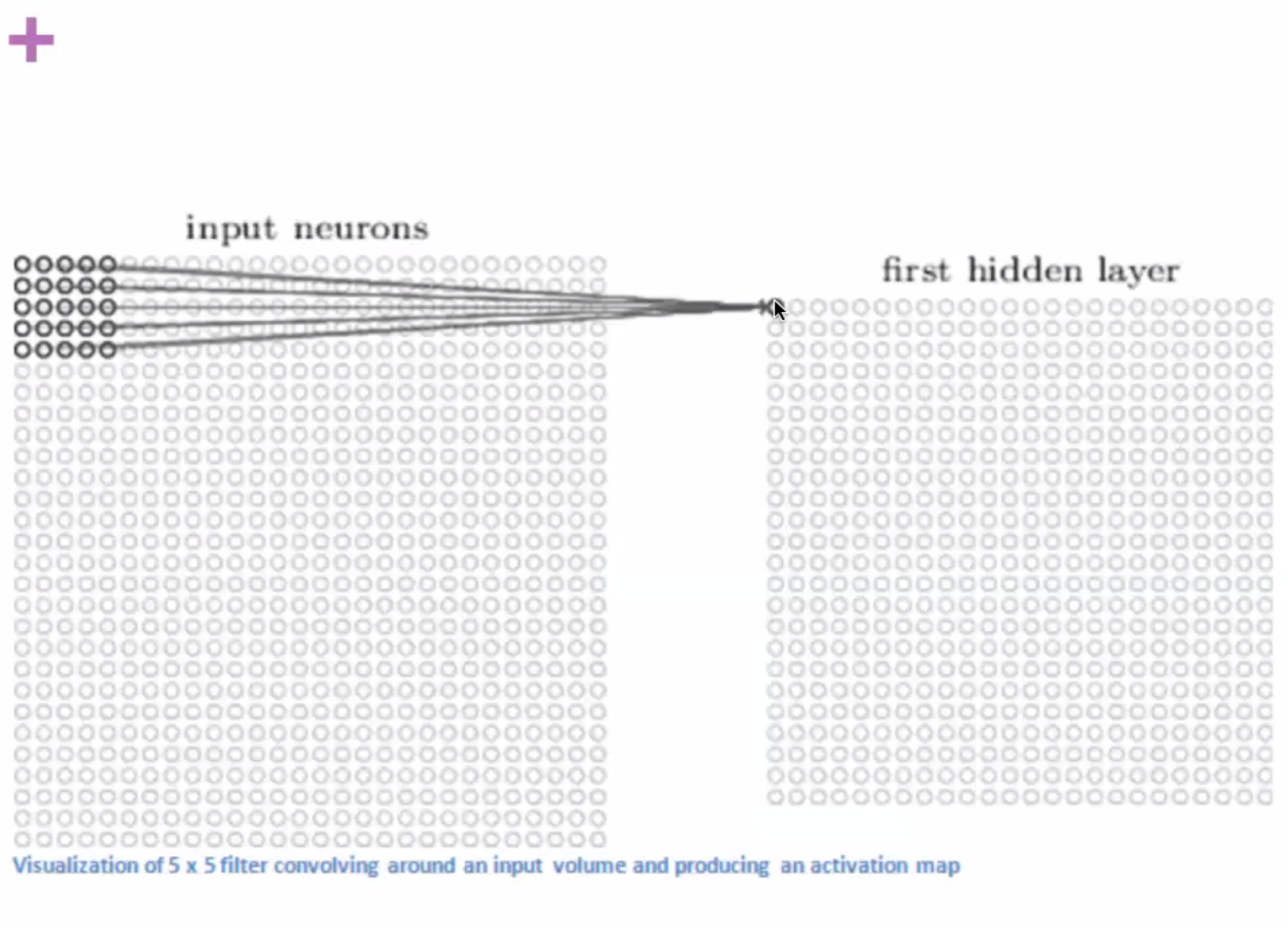
**Advanced Types of Neural Networks**

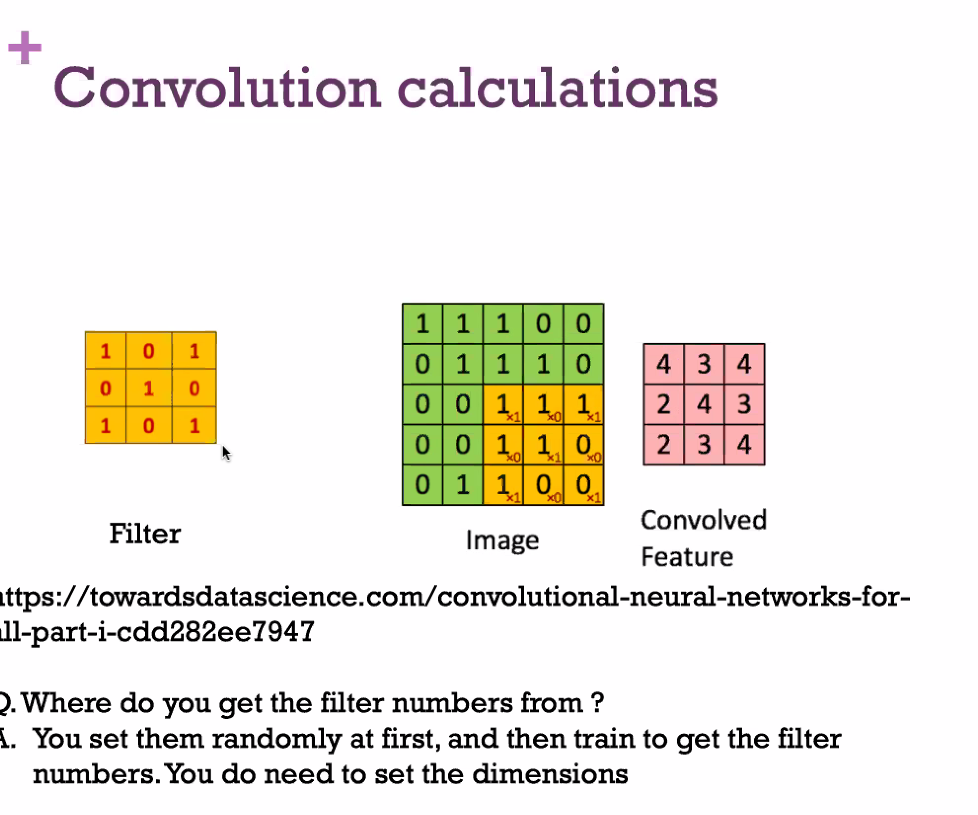
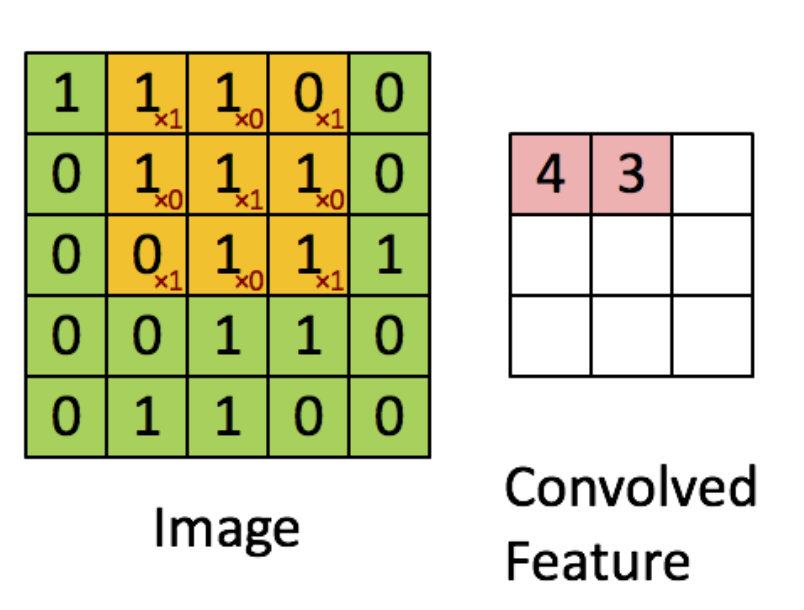
**CNNs - Convolutional Neural Networks**

Convolves matrix of data with random filters, and then pools them

Filter starts random, and are trained along with weights with gradient descent

-why filters/kernels 3x3? Not sure, maybe haven’t tried other sizes of filters. Lots of things in ai, like sigmoid function, that people just use for ages because that’s what always been used and it’s a young field? Some evidence that smaller is better, and odd numbers better, but 1 doesn’t group information from neighbouring elements, so 3x3 is a popular choice: <https://towardsdatascience.com/deciding-optimal-filter-size-for-cnns-d6f7b56f9363>

****

****

Use convolution: slide filter over original matrix of data, multiply original numbers by filter numbers and sum.

Next layer is an abstraction of previous layer

Dropout

The process of turning off a percentage of the weights in a layer during training. This prevents the network from overfitting.

Increase reliance on others, improving other connections

Pruning-relates to real brains, which may start with too many connections, or at least become more efficient trough pruning

**MaxPooling2D**: A method of shrinking 2D arrays inside of the network. It can half the size of an array by looking at nearby values and only keeping larger values.

**MaxPooling1D**: The same process as Max Pooling in 2D, except in 1D (think applied to a list of numbers).

**Flatten**: The process of flattening a 2D array into a 1D list. This is used if we have an array that we want to feed into a dense layer.

Finally goes to normal network for classification

Convolutional layers good for images and feature maps

Extremely powerful-starting to get better than human performance. Sometimes people shift their issue into a way that would allow usage of a convolution network

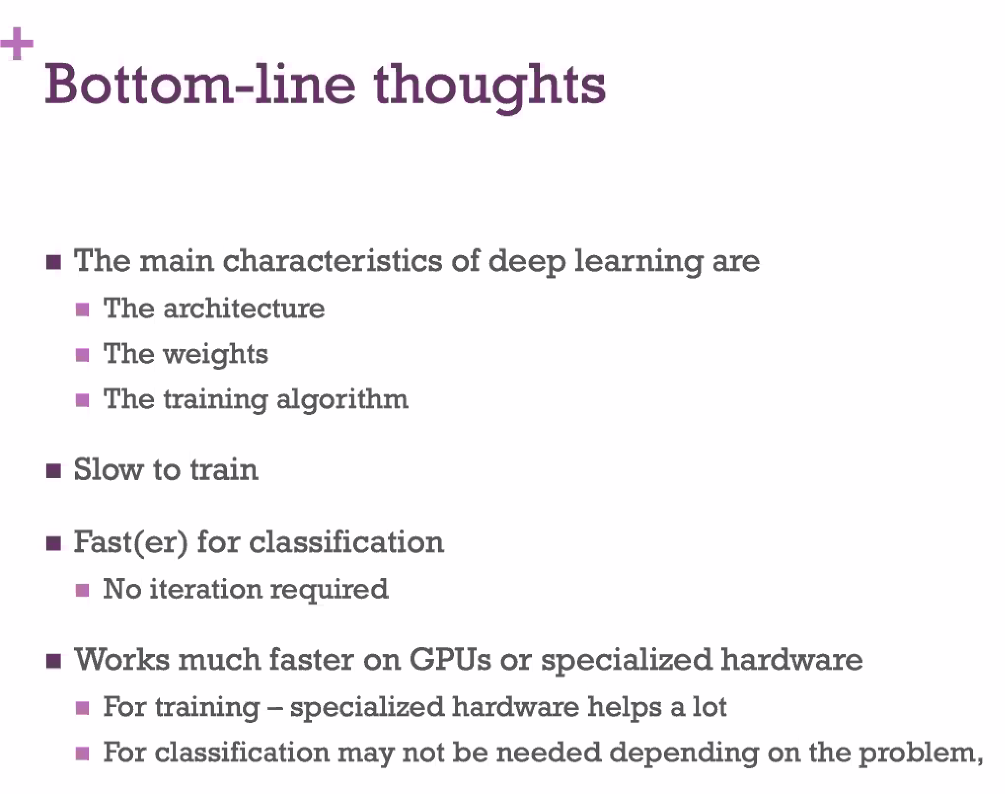
Start with random filter numbers, which are adjusted for better performance

Often good idea (and with networks general) to start with pretrained model and retrain and modify with new data, especially if lacking data

Gabor filter: filter often used for edge or shape detection, used in computer vision

-random filters started with often converge on classically used filters like this, but rather than tell computer to use a filter like this, let computer find out if there are better/more optimal filters. Could use these filters as a starting point, but often easier and more effective to just leave everything to the algorithm anyway.

Could do steps like find where a house is, then find the door, then find the number, each as a distinct process-takes more thinking time and may be less effective. Can just provide whole image and spend more time training, but simplify process, at least on human side



Figuring out architecture a parameters-no clear way of creating networks, more of an art than a science

**LSTMS**

**RNNs**

Amazon Mechanical Turk: crowdsources data labelling

-reliability issues? How do you verify the accuracy of labels?