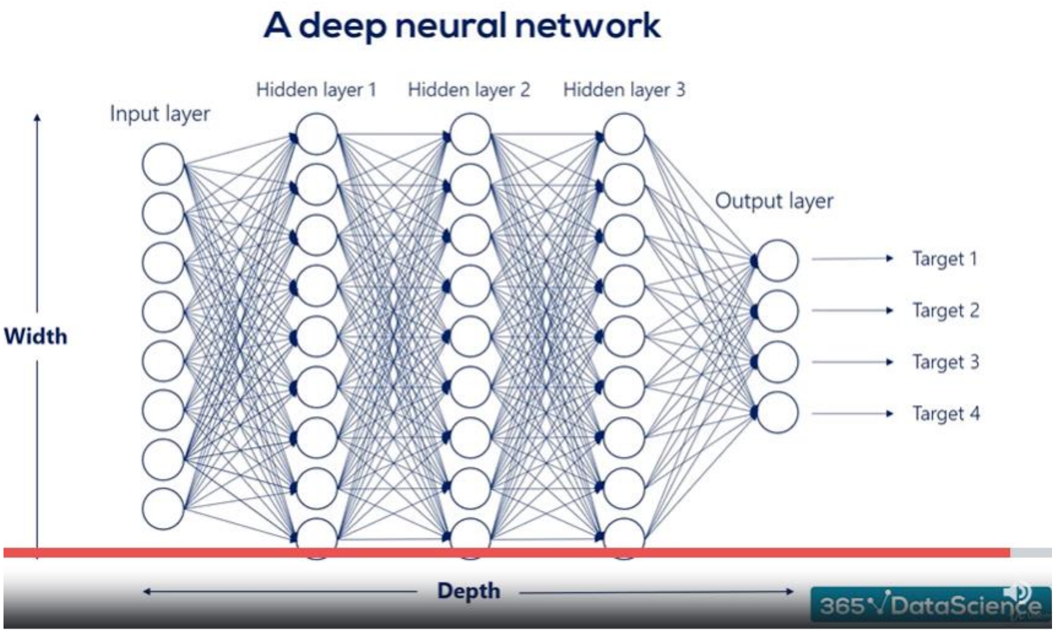
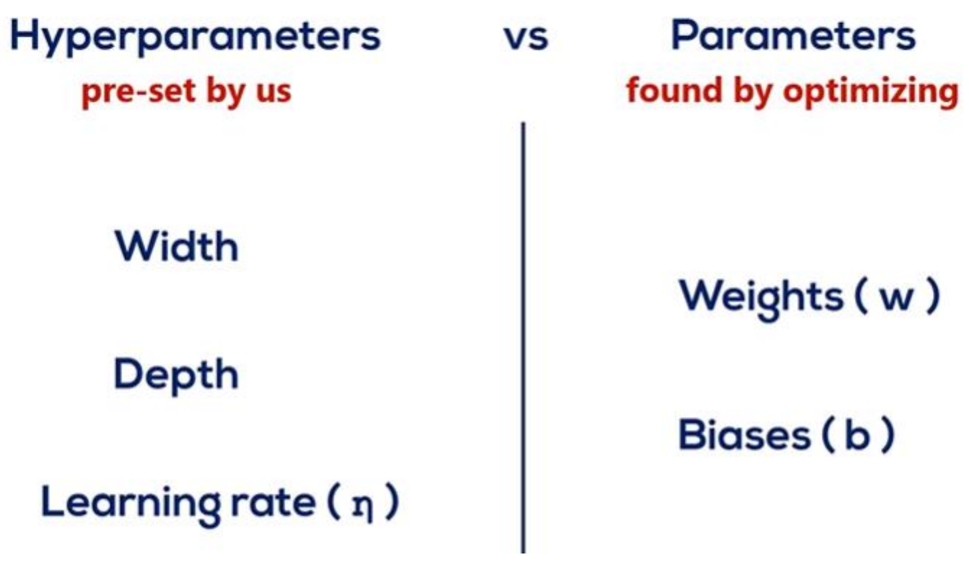
Neural Networks summary

Summary I made on the course deep neural networks with TensorFlow on Udemy

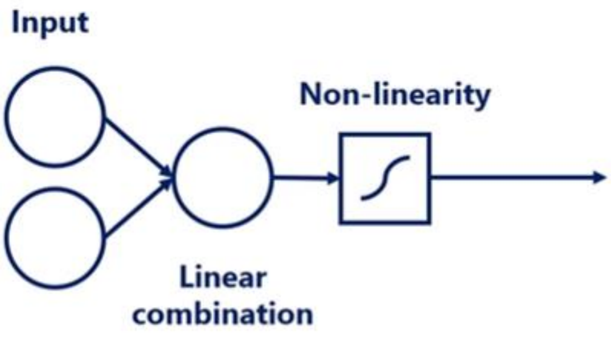
A deep neural network consist of an input layer which is the data that you want to feed into your neural network (images of cats and dogs) and the output layer is the target you want to achieve (let's say cats(as 1) and dogs(as 0)). A deep net also contains hidden layers. The number of hidden layers is called depth and the number of units in a hidden layer is the width of your network.



What is the difference between hyperparameters and parameters?



Hyperparameters	vs	Parameters
pre-set by us		found by optimizing
Width		Weights (w)
Depth		Biases (b)
Learning rate (η)		

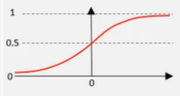
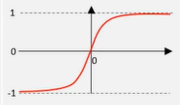
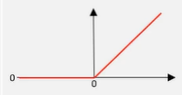


In order to have deep nets and find complex relationships through arbitrary functions, we need non-linearities.

Non linearities are also called activation functions: activations transform inputs (like the temperature which might be increasing or decreasing) into an output of a different kind (like put on a jacket or not):

Examples of different activation functions are given below:

Common activation functions

Name	Formula	Derivative	Graph	Range
sigmoid (logistic function)	$\sigma(a) = \frac{1}{1+e^{-a}}$	$\frac{\partial\sigma(a)}{\partial a} = \sigma(a)(1-\sigma(a))$		(0,1)
TanH (hyperbolic tangent)	$\tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}$	$\frac{\partial \tanh(a)}{\partial a} = \frac{4}{(e^a + e^{-a})^2}$		(-1,1)
ReLu (rectified linear unit)	relu(a) = max(0,a)	$\frac{\partial\,\mathrm{relu}(a)}{\partial a} = \begin{cases} 0, if\ a \le 0 \\ 1, if\ a > 0 \end{cases}$		(0,∞)
softmax	$\sigma_i(\boldsymbol{a}) = \frac{e^{a_i}}{\sum_j e^{a_j}}$	$\frac{\partial\sigma_i(\boldsymbol{a})}{\partial a_j} = \sigma_i(\boldsymbol{a})\left(\delta_{ij} - \sigma_j(\boldsymbol{a})\right)$ Where δ_{ij} is 1 if i=j, 0 otherwise	different every time	(0,1)

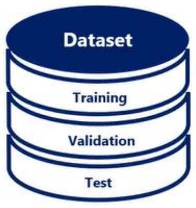
How the Backpropagation works:

The algorithm measures the contribution of each hidden unit to the respective errors. After finding the contribution of each hidden unit to the respective errors we can update the new weights. Then identifies which weight leads to which errors; it then adjusts:

- The weights that have a bigger contribution to the errors by more;

- The weights that have a smaller contribution to the errors by less

Training/validation and test data set:



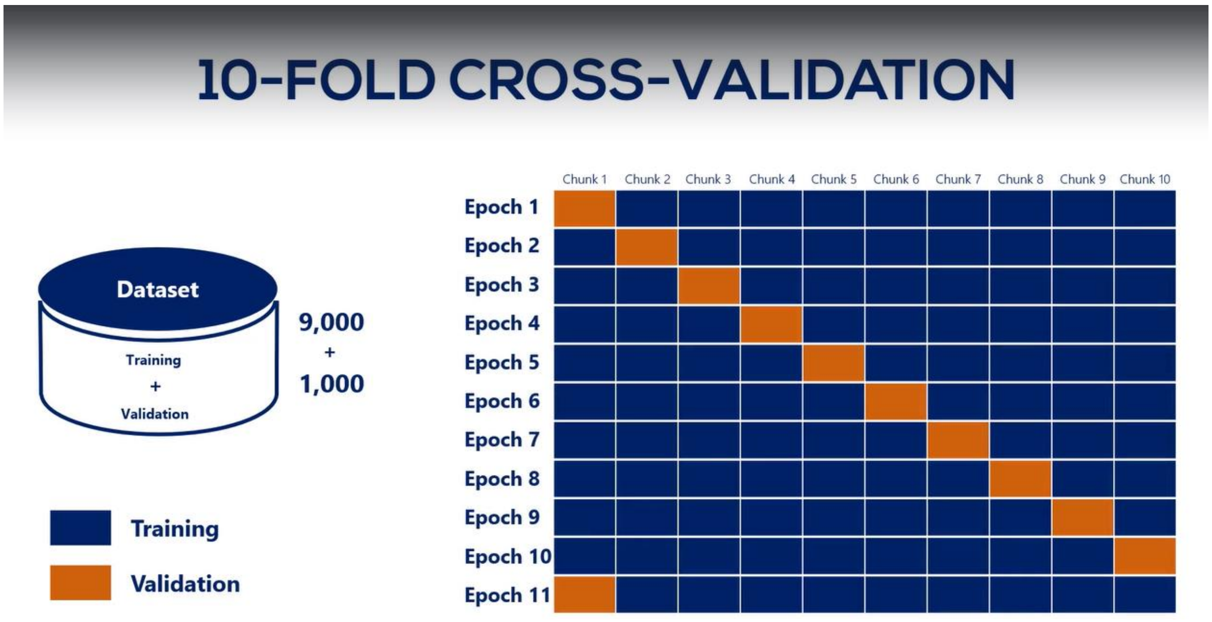
Train data sets is the data set that we train our model with, in here we do both the forward and the backward propagation updating the weights.

Validation data sets helps prevent overfitting by passing (forward pass) the data into the model created through the train data set and calculate what the loss is. When the validation loss starts increasing it means the model is overfitting.

With the **test** data set, the predictive power of the model is measured.

N-fold cross validation:

By few data we can use the N-fold cross validation to create a model that utilizes all the data. This helps us prevent underfitting and the model will probably overfit since we are training on the validation data.



Early stopping:

Early stopping is a technique that helps us stop overfitting by stopping early.