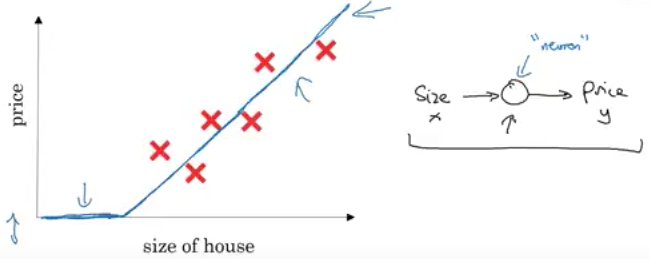
**Deep Learning by Andrew Ng**

**Contents:**

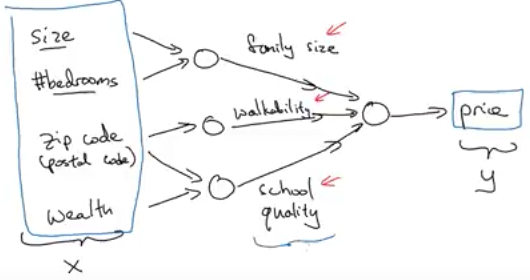
1. Neural Networks and Deep Learning
2. Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization
3. Structuring your ML project
4. Convolutional Neural Networks
5. NLP: Building sequence models (LSTM, RNN)

**Lec 2: What is a Neural Network?**

Consider the case of house price prediction. We fit a line in linear regression to determine price against size of the house. In neural network the neuron performs this function.



For predicting house price with more than one feature:

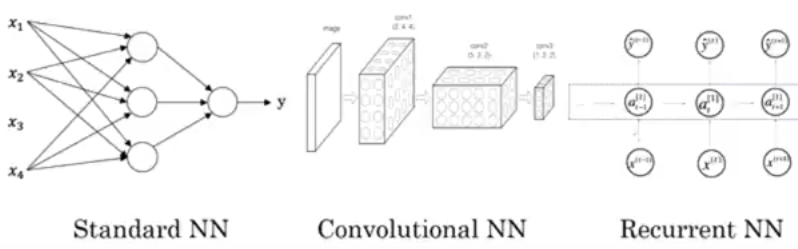


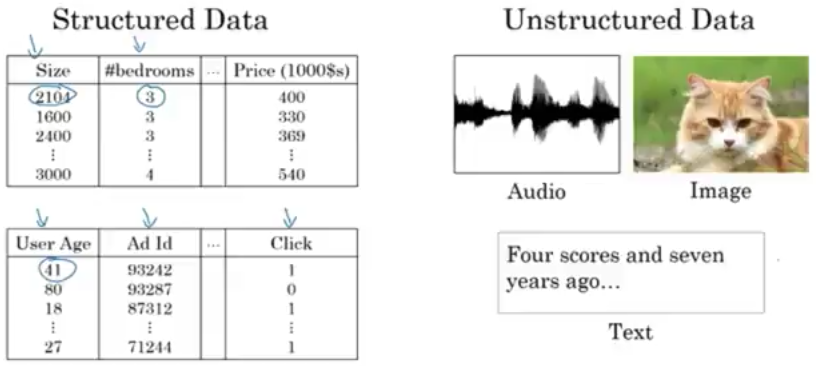
For neural networks to work, providing input and output is alone sufficient. It figures out the hidden layer by itself by taking all input features to one neuron. It performs really well given large number of training example.

**Lec 3: Supervised Learning with Neural Networks**

Most neural networks are based on supervised learning where both input and output are fed. CNNs for images, RNN for sequence data, standard neural networks for classification/regression and custom/hybrid neural networks for more complicated applications.

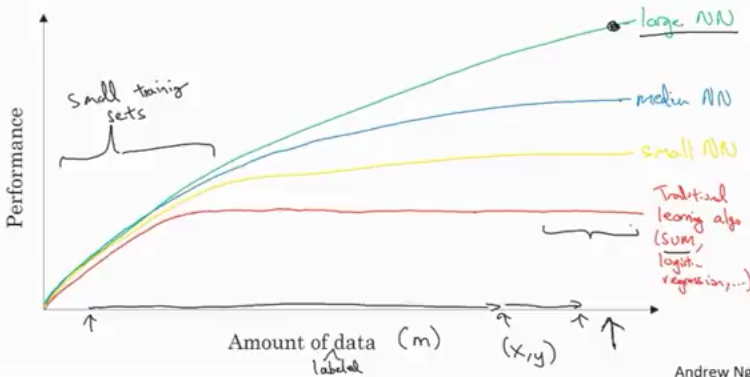
Some neural network examples:





**Lec 4: Drivers behind rise of Deep Learning**

Over recent times, deep learning has scaled up with larger neural nets (deeper layers and more neurons) and with lots of labeled data to train on.



Considering the portion of graph where there are lesser training sets, the performance of any algorithm depends heavily on feature engineering. So any algorithm can perform well.

Apart from data and computation, algorithms have played a major role as well. For example, switching over from sigmoid function to ReLu activation function helps gradient descent perform learning much faster

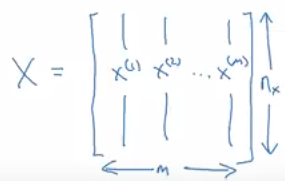
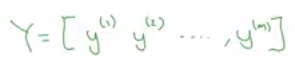
Computation plays a big role because mostly NNs are run by trial and error in the cycle of idea -> code -> experimentation and back to idea again.

**Lec 5: Binary Classification in Deep Learning**

We are looking at logistic regression.

For a 64 x 64 image with three channels, it must be converted to a feature vector (np.flatten) of size 64 x 64 x 3.

*Notations used:*

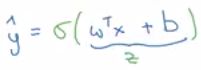
* (x, y) is the training example along with its label.
* Consider ‘m’ training examples: 
* For neural networks it is always advised to stack the ‘m’ input vectors and its labels separately:  

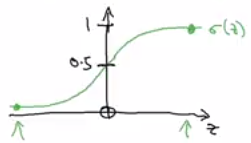
**Lec 6: Logistic Regression**

Given a training example X, we want to predict y’ (y-hat):  where x is .

Its parameters weights and bias are: 

The output cannot be a linear equation because what we want is a probability output, hence we fit a sigmoid function.



Sigmoid function:  

Note:

* If value is large then output is close to 1
* If value is small then output is close to 0

**Lec 7: Logistic Regression Cost Function**

Cost function is used to train the parameters ‘W’ and ‘b’.

The loss function determines how good the predicted output (y’) is against actual output (y). Squared error is an absolute choice but in logistic regression the gradient descent would have many minima (not convex), hence it is not used. 

The loss function used is: 



Loss function is for a single training example.

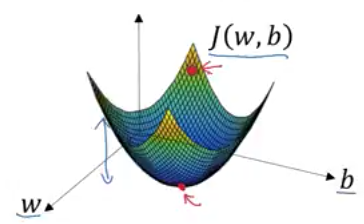
The cost function is used to check how the parameter W and b are doing for the entire training set: 

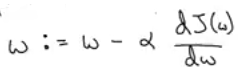


**Lec 8: Gradient Descent**

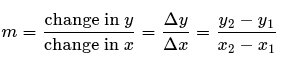
Gradient descent algorithm to learn parameters ‘W’ and ‘b’ on the training set.

Gradient descent finds best ‘W’ and ‘b’ that minimizes the cost function J(W, b).

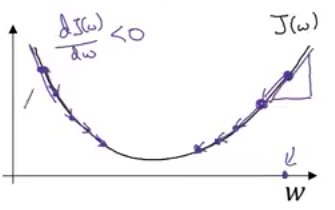
The cost function is a bowl curve (convex function) with a single minimum: 

To get minimum cost having a single parameter we have to repeat: 

The slope is the change made to alter ‘W’.

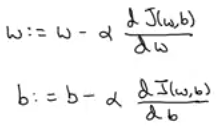
Slope of curve at a point: 

Two cases of positions of ‘W’:



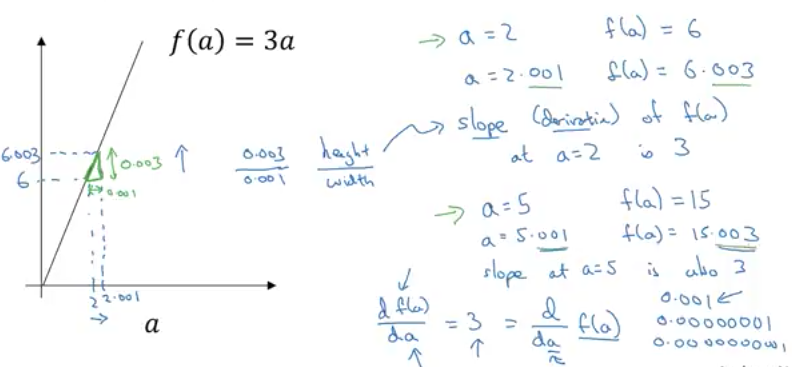
* If ‘W’ is high, then the slope at that point will be positive, hence the change must be subtracted.
* If ‘W’ is low, then the slop at that point will be negative, hence the change must be added.

Now if there are two parameters repeat the following:

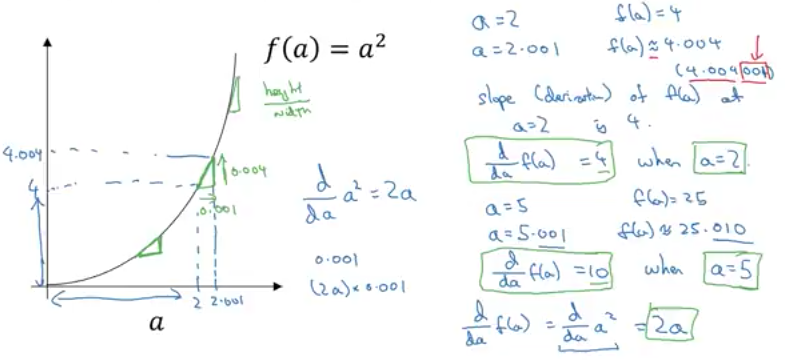


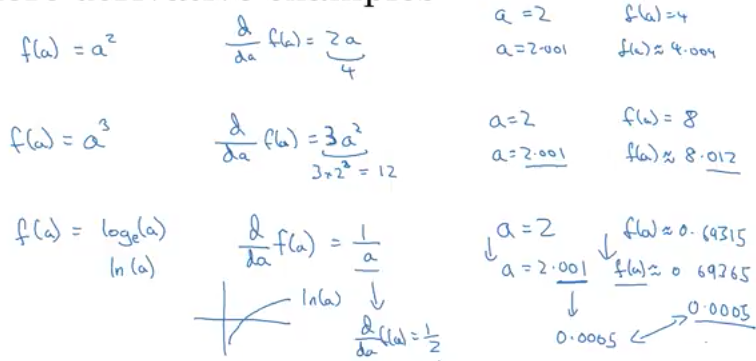
**Lec 9: Derivatives**

Intuition of derivatives (slope) of a line which is same throughout. So if a little nudge in variable ‘a’ causes function ‘f(a)’ to move by three times, then the slope of that function is 3.



**Lec 10: Derivatives Examples**





**Lec 11: Computation Graph**

Computation helps understand:

* Forward pass to obtain output
* Backward pass to obtain gradients/derivatives to optimize cost function

Example of forward pass:

