## Deep Learning

1.

Least square assumptions

* You can transform the x not the weights
* Sin(w1 + w2) NOT LINEAR
* Anytime your weights are not linearlised it is not good
* Signal dependent noise (x1+x2)(noise) NOT LINEAR
* Tikhonov is used when the training set is bad, bias results

2.

Logistic Regression

* Supervised learning
* Specification
* Classification
* Model of likelihood = Sigmoid (binary) / Logistic
* Loss function = Cross Entropy / Binary Cross Entropy / Multinomial Cross Entropy
* High negative log-likelihood means poor fit of data
* Assumes linear relationship between variables and logit
* Logistic model is estimated by Maximum Likelihood Estimation

Logit

* Between – infinity and + infinity
* Probability - 1 and 1
* Infinity means +1 or negative infinity means -1
* If an observation is very farm away from the hyperplane and is in class 1, it has a high value for Logit. Deep in class 1

Linear Regression

* Error values are assumed to be normally distributed
* If you can assume normal distribution for errors its Probit regression

3.

Kernel Methods

* P – features, N – number of observations
* Best suited for large P small N
* Similarity function
* Complexity of the kernel function is independent of the corresponding mapped feature space

SVM

* Support Vector Machine
* Starts as linear classifier but has a different loss function

Machine Learning Methods

* Less effective when classes mix
  + Overlapping points
  + Can’t draw boundaries between classes

Boundary maps

* When there is no clear boundary in two areas, must transform data set to use Logistic regression.
* Use RBF kernel to get nice blobby parts
* When you get jagged boundaries, you are using Neighbours

4.

K nearest neighbours

* Jagged edges
* When K is increased, probability is increased from just being 0 or 1 can be 0.1 or 0.2…, when K isn’t 1 probabilities are never just 0 or 1
* Increases computational time when evaluation the prediction

Decision Trees

* ML methods (AdaBoost) also measure the probability of belonging to a class not just the most likely class
* More efficient than Kernel SVMs
* Decision trees are bad at non vertical or horizontal

ROC Curve / Thresholds

* Use ROC curve for thresholds on binary classifiers

5.

Unit

* Differentiable Function
* Neuron units take linear combination of inputs and gives it a non-linear function (activation function)
* Increasing DNN layers doesn’t necessarily mean things will be better

Universal Approximation Theorem

* Any continuous function can be arbitrarily well approximated GIVEN
* A single hidden layer neural network with a neural network with a linear neuron output, given enough neurons

Epoch

* Faster to compute a mini batch if the batch sizer is 1024
* Gradient descent tis number of times all the training vectors are used / batch size
* n/batch size

Training / Validation

* When validation is better than training at the start, dropout is at the start
* Dropout is only applied at the start of training
* When oscillations happen, learning rate is too high
* Adding Dropout and Gaussian noise layer simulates more data

6.

Feed Forward Neural network

* Everything you see is FNN, Convolutional neural layers are FNN

Convolutional Neural Network

* Mainly to process and analyze images, works with voice and natural language
* Softmax can be used
* Weight sharing occurs in CNN

Flattening

* Transforms a tensor into a vector

Convolution cannot have dense layers, dense means fully connected.

Dense layer doesn’t do weight sharing

Inception Modules are used in convolutional networks. Reduced dimensionality

GAN

* Input noise doesn’t not matter to ensure generated samples follow the target data distribution. It does for Autoencoders.

Cross-entropy for logistic losses not linear. L1 is for logistic losses.

LSTM – long short term memories. (Text)