Potential website sources

<https://en.wikipedia.org/wiki/Linear_separability>

[www.faadooengineers.com/online-study/post/eee/neural-networks-and-fuzzy-logic/1478/linear-separability](http://www.faadooengineers.com/online-study/post/eee/neural-networks-and-fuzzy-logic/1478/linear-separability)

<https://www.cs.nott.ac.uk/~pszqiu/Teaching/G53MLE/ffnets-note.pdf>

<https://aishack.in/tutorials/linear-separability/>

<file:///C:/Users/jmcke/Downloads/Artificial_Intelligence_A_Modern_Approac.pdf>

<https://deepai.org/machine-learning-glossary-and-terms/activation-function>

<https://brilliant.org/wiki/backpropagation/#the-backpropagation-algorithm>

<https://www.guru99.com/unsupervised-machine-learning.html>

lit notes

Thing to talk about:

* The prominence of recommenser systems and algorithm designed to improve user experience like with social media and Netflix
* The Netflix coemption to create a better algorithm
* The use of reccomender system
* The danger of rellying too much on these kinds of algorithm based learning
* The great implementation of neural network
* The type of neural network and how they can be used.

Using nn for prediction

Metion the history of NN

Creation of perceptrons

Goal with testing is to minimise the error

Perceptron/delta rule

Momentum is used to speed learning up or to avoid the local minima

Look into sigmoidal activation

Mention liniear seperability

Mention the obkect orientated techniche used to make the program more efficient

encapsulation; information hiding; inheritance

mlp can solve non linear problem

layers of neuron in the middle are the hidden layer

can’t use delta rule for the hidden layer as you don’t know the target

werbos developing back propagation

bpg uses a generalised data rule

output deltas propgated back for errors

picton has a book on backpropgation

with BPG error for hidden nodes uses deltas of neuron in the next layer \* the connecting weights

simple adaptive momentum is a new layer for training multi layer perceptrons

adaptive uses

relevant under and over fitting issues considered

too many hidden layer nodes can lead to over fitting

one layer works for any continuous function

2 or more is fore deep learning

Might want to consider using a roullete wheel approach

https://brilliant.org/wiki/backpropagation/

<https://towardsdatascience.com/backpropagation-intuition-and-derivation-97851c87eece>

<https://www.ibm.com/cloud/learn/unsupervised-learning>

<https://www.analyticsvidhya.com/blog/2014/10/ann-work-simplified/>

https://developer.android.com/studio/intro