Week 8 Summary: Neural Networks

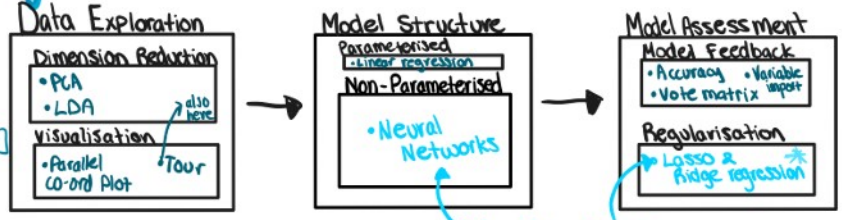
Sunday, 21 June 2020 6:02 PM

Week 8 : Neural Networks

(Regularisation isn't in the tutorial)

① The Big Picture

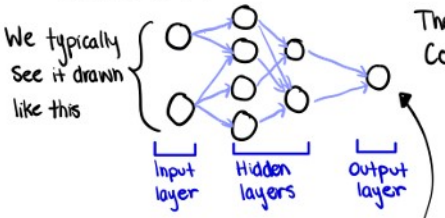
[I lost the old "Big Picture", so this week its a bit different]



* Note: Regularisation could also be considered to be model structure but because it fixes a high variance model I left it here.

② Neural Networks

? What is it ?



The in depth details of how this drawing is a model is Covered Covered in ETC3555 I'm just going over the "gist" of it.

This is our model "structure"

(as far as you are concerned it is only a set of hyperparameters that need to be optimised)

* Some online sources will give good advice on how to pick # of nodes & layers.

• How the nodes combine & interact is also beyond this unit.

! Then what DO we cover ?

① Model structure to mathematical formula

② "gist" intuitive sense of how they work.

NEURAL NETWORK

ARCHITECTURE

HIDDEN LAYER — INPUT LAYER

Function structure :

$$\hat{y} = g\left(\alpha_0 + \sum_{k=1}^{S} \left(\alpha_k f\left(\beta_0 + \sum_{j=1}^{P} \beta_{jk} x_j\right)\right)\right)$$

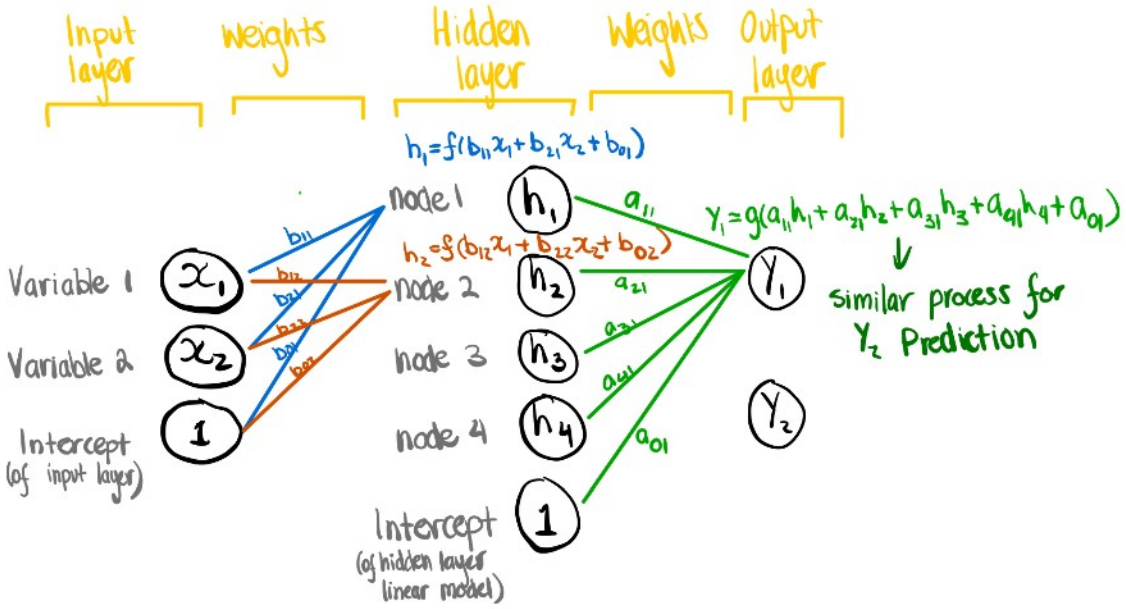
(Number of nodes in first hidden layer / second layer weights / second layer constant) (Inputs (number of predictors) / first layer's weights / First Layers constant)

* Activation functions

Network Architecture

Input layer — Weights — Hidden layer — Weights — Output layer

Network Architecture



Activation functions $(u = \beta_0 + \sum_{j=1}^{p} \beta_j x_j)$

Logistic

$$f(u) = \frac{1}{1 + e^{-u}}$$

1 output node

Gaussian Radial

$$f(u) = \frac{1}{\sqrt{2\pi}} e^{\frac{-u^2}{2}}$$

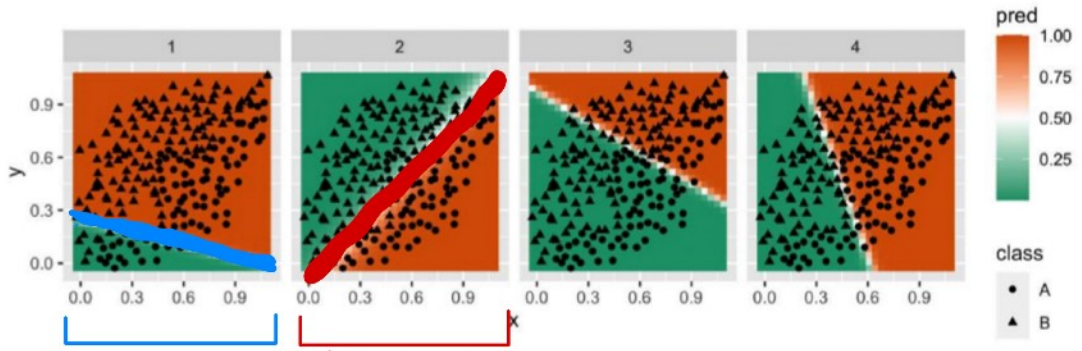
$(0, 1]$

Hyperbolic Tangent

$$f(u) = \frac{e^{u} - e^{-u}}{e^{u} + e^{-u}}$$

$(-1, 1)$

On the scatter plot



node 1 linear input

$y_1 = b_{11}x_1 + b_{21}x_2 + b_{01}$

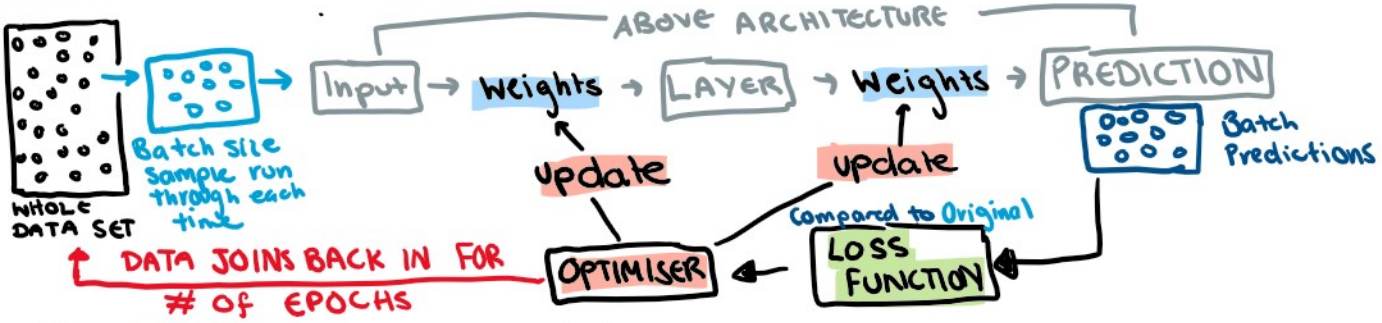
$h_1 = f(y_1)$

node 2 linear input

$y_2 = b_{12}x_1 + b_{22}x_2 + b_{02}$

$h_2 = f(y_2)$

FEEDBACK MECHANISM



The weights of the model are trained according to a

# of EPOCHS

The weights of the model are trained according to a feedback mechanism

The loss function (cross entropy/RSS/ect) measures model performance

The optimiser decides how the loss function is used to update the weights

Batch size is the number of samples fed into the model at a time

↑batch size ⇒ ↓ computation, ↑ stable estimates

Epoch refers to how many iterations through the entire training data.

PROS VS CONS

PROS	CONS
○ Capture complex relationship between outputs and inputs.	○ Black box type model ○ Dont work well when $p > n$ (or close to) & can potentially overfit ○ Computationally expensive