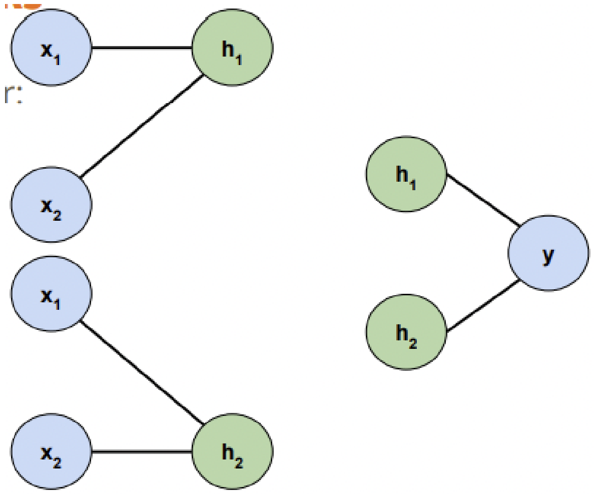
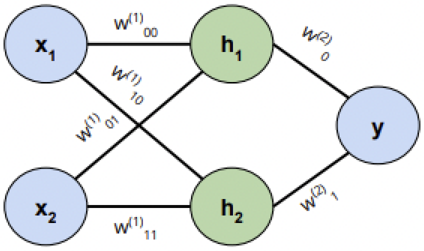
CAS CS 506
Lec 21

Neural Networks

1. Neural Networks
   a. Putting it all together





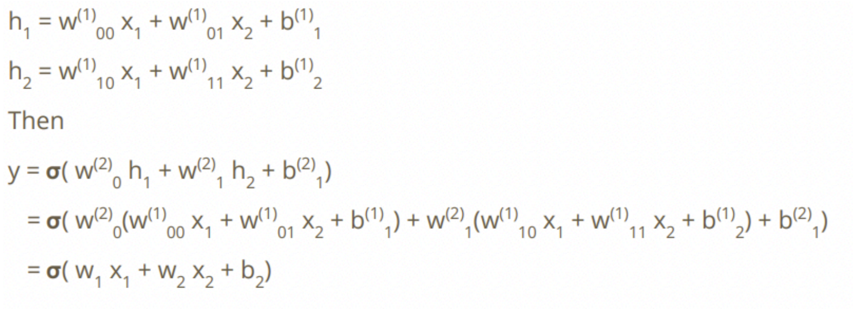
   b.
   c. Divided into input layer, hidden layer, and output layer
   d. It is all about learning features (created in the hidden layer(s) automatically
   e. We need to define
      i. How input flows through the network to get the output (forward propagation)
      ii. How the weights and biases gets updated (Backpropagation)
2. Neural Networks - Forward Propagation

a. Using matrix notation:

$$\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \sigma\left(\begin{bmatrix} w_{00}^{(1)} & w_{01}^{(1)} \\ w_{10}^{(1)} & w_{11}^{(1)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1^{(1)} \\ b_2^{(1)} \end{bmatrix}\right)$$

$$y = \sigma(\begin{bmatrix} w_{00}^{(2)} \\ w_{01}^{(2)} \end{bmatrix}^T \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + b^{(2)})$$

b. If we don't have o in the hidden layer, we just end up with normal logistic regression on x1 and x2



$h_1 = w^{(1)}_{00}\, x_1 + w^{(1)}_{01}\, x_2 + b^{(1)}_1$

$h_2 = w^{(1)}_{10}\, x_1 + w^{(1)}_{11}\, x_2 + b^{(1)}_2$

Then

$y = \boldsymbol{\sigma}(\, w^{(2)}_0\, h_1 + w^{(2)}_1\, h_2 + b^{(2)}_1)$

$= \boldsymbol{\sigma}(\, w^{(2)}_0(w^{(1)}_{00}\, x_1 + w^{(1)}_{01}\, x_2 + b^{(1)}_1) + w^{(2)}_1(w^{(1)}_{10}\, x_1 + w^{(1)}_{11}\, x_2 + b^{(1)}_2) + b^{(2)}_1)$

$= \boldsymbol{\sigma}(\, w_1\, x_1 + w_2\, x_2 + b_2)$

3. Neural Networks - BackPropagation

a. Weights and biases gets updated from logistic regression except relative to the learned features h

$\mathrm{Cost}(w, b)$

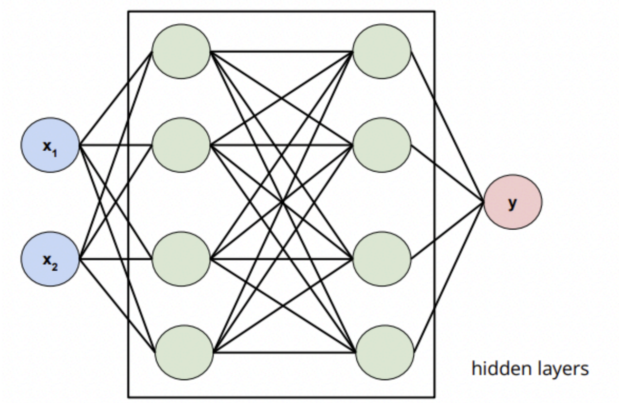
$$= \; -\frac{1}{n}\sum_{i=1}^{n}\left[yi\log(\sigma(-w^T h_i + b)) + (1 - y_i)\log(1 - \sigma(-w^T h_i + b))\right]$$

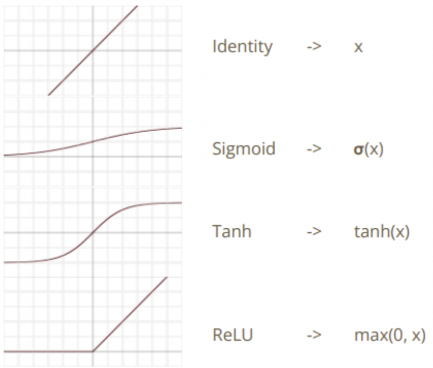
$$\nabla \mathrm{Cost}(w, b) = \left[\frac{\partial}{\partial w}\mathrm{Cost}, \frac{\partial}{\partial b}\mathrm{Cost}\right]$$

$$\frac{\partial}{\partial w}\mathrm{Cost} = \frac{1}{n}\sum_{i=1}^{n} h_i(y_i - \sigma(-w^T h_i + b))$$

$$\frac{\partial}{\partial b}\mathrm{Cost} = \frac{1}{n}\sum_{i=1}^{n} \sigma(-w^T h_i + b) - y_i$$

b.

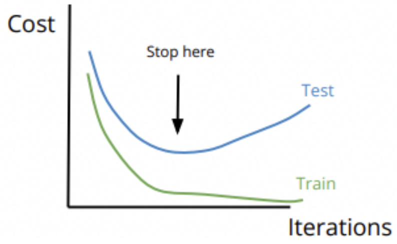
4. Feedforward Neural Networks
    a. 

5. Neural Networks
    a. Can do both classification and regression
    b. Tuning Parameters
        i. Step size a
        ii. Number of BackPropagation iterations
        iii. Batch size
        iv. Number of hidden layers
        v. Size of each hidden layer
        vi. Activation function used in each layer
        vii. Cost function
        viii. Regularization (to avoid overfitting)
6. Activation Functions
    a. 

    b. Note: can use any function you want in order to introduce non-linearity. These are just popular ones that have been shown to work in practice
    c. Tuning the activation function is equivalent to feature engineering
7. Neural Networks - Challenges
    a. High risk of overfitting as you're optimizing on the training set
    b. As the dimensionality of the input increases
        i. So does the number of weights
        ii. The gradients typically get smaller. Vanishing gradient problem

        iii. Doesn’t do well for computer vision where object of detection can be anywhere in the image
        iv. Doesn’t handle sequences of inputs (i.e. providing context for data)

8. Neural Networks - Regularization
    a. Two main ways:
        i. Early termination of weight / bias updates



        ii. Dropout - kill neurons (by setting them to 0) randomly