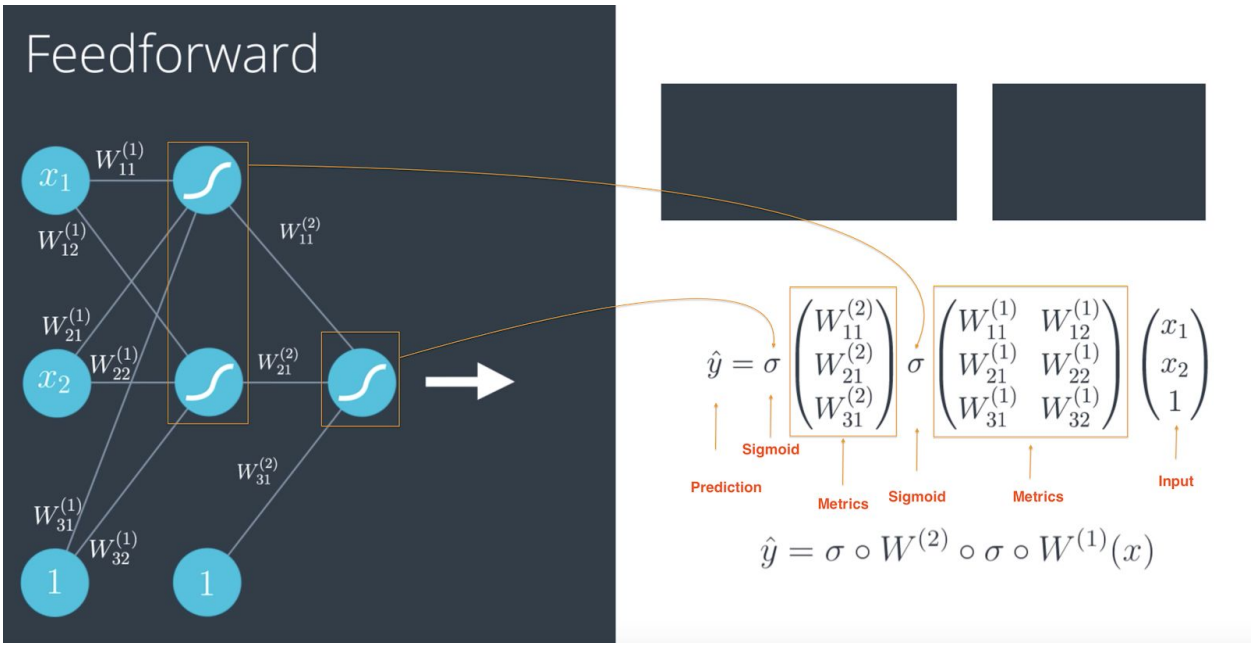
Note: This is the summary note from Udacity Introduction to Deep Learning with PyTorch

Feedforward

- is the process neural networks use to turn the input into an output.
- First step of backpropagation (next lesson)
- **CNNs** and **RNNs** are just some special cases of Feedforward networks.
    - https://towardsdatascience.com/feed-forward-neural-networks-c503faa46620



- What is the error function for multi-layer perceptron ?
    - Error Function:

$$E(W) = -\frac{1}{m} \sum_{i=1}^{m} y_i ln(\hat{y}_i) + (1 - y_i) ln(1 - \hat{y}_i)$$

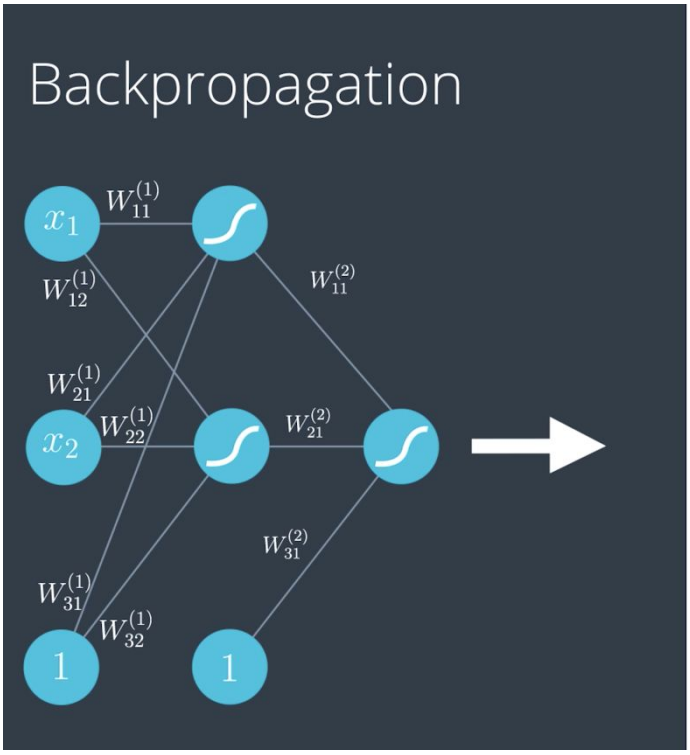
    - For Binary Error Function:
      We have $\hat{y} = \sigma(Wx + b)$

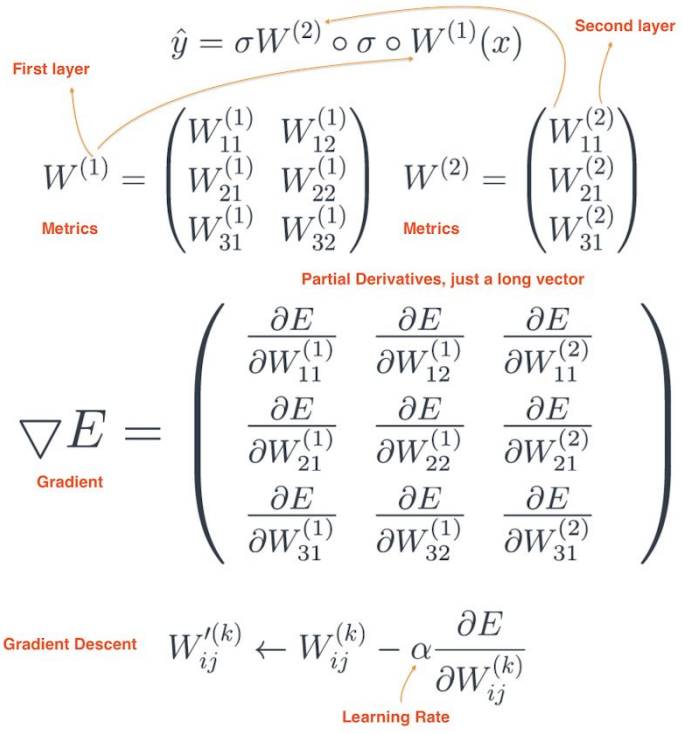
    - For MLP:
      We have $\hat{y} = \sigma \circ W^{(n)} \circ \sigma \circ W^{(n-1)} \ldots \circ \sigma \circ W^{(1)}(x)$

Backpropagation

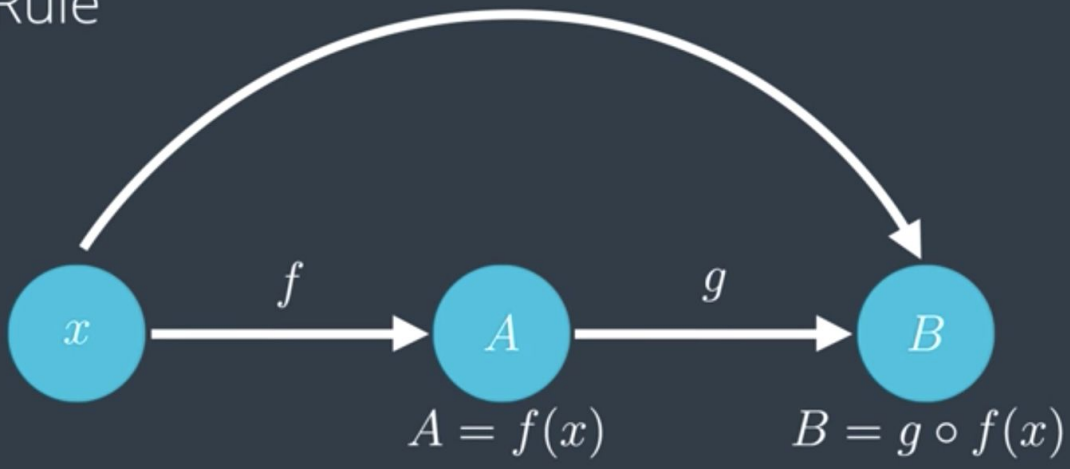
- It's a method of training the neural network
- Consists of:
    - Doing a feedforward operation.
    - Comparing the output of the model with the desired output.
    - Calculating the error.
    - Running the feedforward operation backwards (backpropagation) to spread the error to each of the weights.
    - Use this to update the weights, and get a better model.
    - Continue this until we have a model that is good.
- The method is handled well by Keras, a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano
- Good article : https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a

- Backpropagation explained in Math
    - First it's good to understand what is composition function: https://www.onlinemathlearning.com/composite-functions.html
        - f(g(x)) is the composite function that is formed when g(x) is substituted for x in f(x)
        - f(g(x)) is read as "f of g of *x*"
        - f(g(x)) can also be written as (f o g)(*x*) or fg(*x*)
        - In the composition (f o g)(*x*), the domain of f becomes g(*x*)
    - Formulas (prediction, gradient, gradient descent)





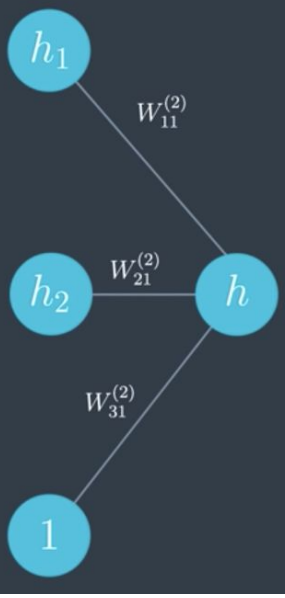
Chain Rule



Partial derivative of B with respective to x

$$\frac{\partial B}{\partial x} = \frac{\partial B}{\partial A}\frac{\partial A}{\partial x}$$

Backpropagation



$$h = W_{11}^{(2)}\sigma(h_1) + W_{21}^{(2)}\sigma(h_2) + W_{31}^{(2)}$$

$$\frac{\partial h}{\partial h_1} = W_{11}^{(2)}\sigma(h_1)[1 - \sigma(h_1)]$$

Derivate of sigmoid function

$$\begin{aligned}\sigma'(x) &= \frac{\partial}{\partial x}\frac{1}{1+e^{-x}} \\ &= \frac{e^{-x}}{(1+e^{-x})^2} \\ &= \frac{1}{1+e^{-x}} \cdot \frac{e^{-x}}{1+e^{-x}} \\ &= \sigma(x)(1-\sigma(x))\end{aligned}$$

Mini Summary

- Feedforward is the process neural networks use to turn the input into an output.
- Backpropagation is a method of training the neural network.