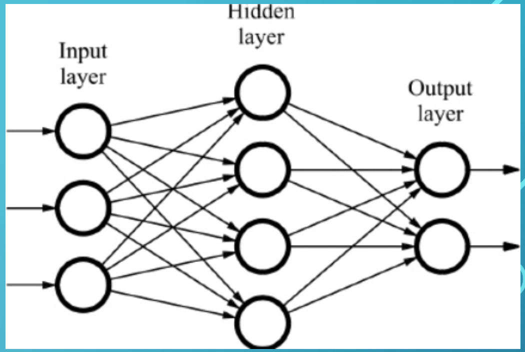
NEURAL NETWORKS CRASH COURSE

ALEX MEI | FALL 2020 | RESEARCH METHODS IN CS

OVERVIEW

- High-Level Introduction
- Mathematical Logic
- Review Questions

NEURAL NETWORKS… WHAT ARE THEY!?

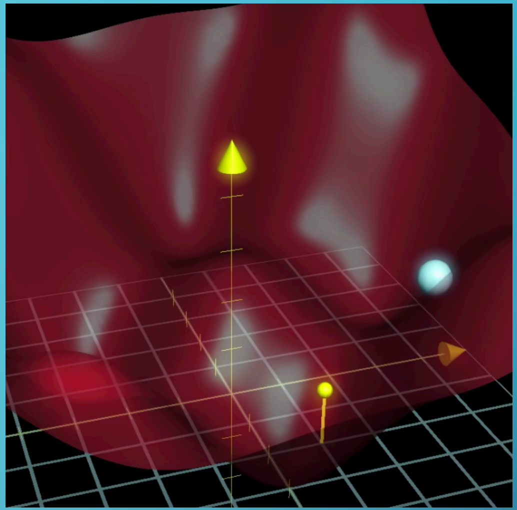


- Neuron: stores a value known as the activation within the unit interval
- Neural Network: complex connection of networks
  - First and Last layers corresponds to some factor about input and output
  - Example: brightness of pixel in image and resulting number
  - Middle Layers: black box with unknown connections to and from other layers
  - Goal: have each layer separating inputs by some determining characteristics

CONNECTIONS, CONNECTIONS, CONNECTIONS

- Weighted average of connections from previous layer’s neurons
- Sigmoid Function: transform weighted average to be within unit interval
- Bias: constant added to the weighted average (for each neuron)
- Goal: find weights and bias to allow the network to be successful

WEIGHING OUT YOUR OPTIONS

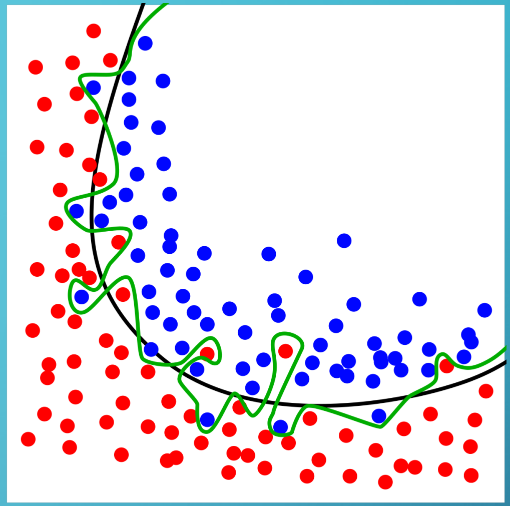


- Initialization: random weights, random biases
- Cost (error): square of differences in performance of weights/biases
- Goal: find weights and bias to minimize cost function
- Gradient: direction of steepest increase
  - Idea: take step toward (local) minimum and repeat until reached
  - May need to experiment to see which local minimum is smaller

PUT IT IN REVERSE

- Given the gradient, adjust based on error function
    - Three options: bias, weights, and activations (focus on stronger connections)
    - For activations, recursively apply to previous layers
- Sum each neuron of last layer to backwards propagate to previous layer

TRAIN, TRAIN, TRAIN



- First Iteration: bad performance
- Adjust based on results of cost function
- Rerunning test data should result in better performance
- Repeat process until performance plateaus
- Beware! Overfitting = model works for only one set of test data

REVIEW I

Neural Networks have many layers. What do each of the following layers symbolize?

- First Layer
- Middle Layer(s)
- Last Layer

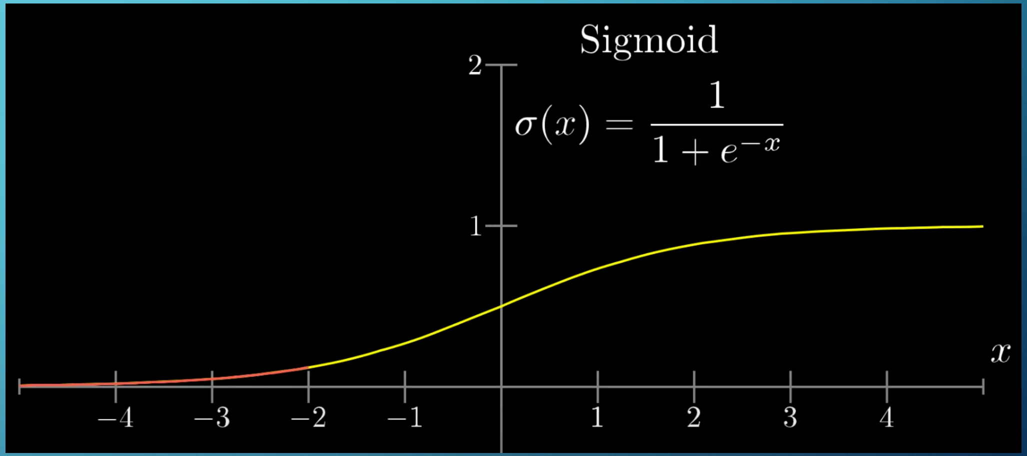
REVIEW II

- How do we minimize the cost function? Why is this important?
- What adjustments can we make to our neural network to make this minimization?

NEURONS, ACTIVATE!

$$a_0^{(1)} = \overset{\text{Sigmoid}}{\downarrow \atop \sigma}\left(w_{0,0}\, a_0^{(0)} + w_{0,1}\, a_1^{(0)} + \cdots + w_{0,n}\, a_n^{(0)} + \underset{\text{Bias}}{\underset{\uparrow}{b_0}} \right)$$

$$\sigma\left(\begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,n} \\ w_{1,0} & w_{1,1} & \cdots & w_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,0} & w_{k,1} & \cdots & w_{k,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix} \right)$$



AND THE COST IS…

$$\text{Cost} \longrightarrow C_0(\ldots) = (a^{(L)} - y)^2$$

$$z^{(L)} = w^{(L)} a^{(L-1)} + b^{(L)}$$

$$a^{(L)} = \sigma(z^{(L)})$$

$$\frac{\partial C_0}{\partial w^{(L)}} = \frac{\partial z^{(L)}}{\partial w^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C0}{\partial a^{(L)}}$$

$$\frac{\partial C0}{\partial a^{(L)}} = 2(a^{(L)} - y)$$

$$\frac{\partial a^{(L)}}{\partial z^{(L)}} = \sigma'(z^{(L)})$$

$$\frac{\partial z^{(L)}}{\partial w^{(L)}} = a^{(L-1)}$$

TIME TO BARGAIN!

$$\frac{\partial C_0}{\partial w^{(L)}} = \frac{\partial z^{(L)}}{\partial w^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C0}{\partial a^{(L)}} = a^{(L-1)} \sigma'(z^{(L)}) 2(a^{(L)} - y)$$

$$\frac{\partial C_0}{\partial b^{(L)}} = \frac{\partial z^{(L)}}{\partial b^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C0}{\partial a^{(L)}} = 1 \sigma'(z^{(L)}) 2(a^{(L)} - y)$$

$$\frac{\partial C_0}{\partial a^{(L-1)}} = \frac{\partial z^{(L)}}{\partial a^{(L-1)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C0}{\partial a^{(L)}} = w^{(L)} \sigma'(z^{(L)}) 2(a^{(L)} - y)$$

$$C(w_1, b_1, w_2, b_2, w_3, b_3)$$

$$\underbrace{\frac{\partial C}{\partial w^{(L)}}}_{\text{Derivative of full cost function}} = \overbrace{\frac{1}{n} \sum_{k=0}^{n-1} \frac{\partial C_k}{\partial w^{(L)}}}^{\text{Average of all training examples}}$$

$$\nabla C = \begin{bmatrix} \frac{\partial C}{\partial w^{(1)}} \\ \frac{\partial C}{\partial b^{(1)}} \\ \vdots \\ \frac{\partial C}{\partial w^{(L)}} \\ \frac{\partial C}{\partial b^{(L)}} \end{bmatrix}$$

PRACTICE I

Suppose there is a 2-layer neural network with 2 neurons per layer with the following weight matrix (and no bias):

$$weights = \begin{bmatrix} 0.2 & 0.4 \\ 0.6 & 0.8 \end{bmatrix}$$

Calculate the output with the given input:

$$input = \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix}$$

PRACTICE II

Calculate the cost given this expected output:

$$output = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Then, calculate the gradient descent.

SOURCES

- 3Blue1Brown (Information)
- Wikipedia, DataBricks (Images)