CS 506
April 24, 2023
Advanced Neural Networks

Introduction
Neural Networks give the false idea that they can clear for themselves.

Logistic Regression
Using logistic regression, we can model the weights and biases and update the weights until we get to a local minimum. Learn the features as you learn to model y. Play around with the architecture such as the number of hidden neurons to try to get a better, more accurate model of y. These are not perfect tools.

BackPropagation
BackPropagation you go from the back layers and update the layers from the back. We need to extract the hidden weights and go one layer further into the cost function. Keep in mind that the gradient computation depends on and is a factor of h. If the gradient is 0 then we cannot update the weights.

Animation
The animation is us learning the three hidden features. Based on the three hidden features we are doing logistic regression to model y. In a 2d space it does not look linear.

Tuning parameters
To find out what tuning parameters affect each other is pretty nebulous and takes trial and error as well as some intuition. You can technically use any activation functions but the ones on the slides are just standard ones that are monotonically increasing.

Normalizing Data for Gradient (article)
The idea behind neural networks is divide and conquer to learn some features that better describe the data. It is very hard to find the weights that make the data positive and negative. Usually it will make the data all positive or all negative. Neuron either activates or does not. A feature shows if such and such is a case of some of the data and not of another part of data. We do not want a global transformation of x but a local transformation. It is not efficient if it is a global transformation. Divide and conquer. If the data is not normalized, we cannot divide and conquer because it is hard to find weights that will activate a neuron with just a part of the data.

Initialization Gotchas
The gradient of the weights and biases depend on the activation of the previous layer. If the activation is 0, then the weights will not be updated. If the feature is constant, you are not learning anything. That feature has no value as it updates the weights all by the same amount. Do not do 0 initialization or constant initialization. Some sort of centered distribution around 0 is best. The value of the sigmoid function should be between 0 to 1.

Convolutional neural networks
Diagonal patterns optimize to the largest number. We need to be more flexible with our model. This is another perspective of neural networks. Convolutional neural networks can apply filters such as that of finding diagonal patterns and apply it many times. It gets complex and complicated really fast.

Auto Encoders