CSC 386 Midterm Exam Review

# Introduction to Machine Learning

* How long have some of the older ML algorithms been around?
  + Around 1950s
* Why has ML’s popularity exploded after 2010?
  + Computational power, available data
* What is the primary difference between ML and traditional programming?
  + In traditional programming the programmer makes the rules, in ML the program develops the ‘rules’ itself by tuning itself over several iterations.
* List the steps in the ML pipeline.

1. Data Collection - Cleaning + Exploring Data
2. Feed model Data
3. Model evaluation – how well its performing
4. Iterate back to correct, or deploy

* Compare supervised learning and unsupervised learning.
  + Supervised learning we know the answer to the problem, the prediction the program should predict. In unsupervised they don’t know the true prediction, and are instead looking for patterns in the data.
  + With what we’ve done so far with our neural nets, is it supervised or unsupervised?
    - supervised

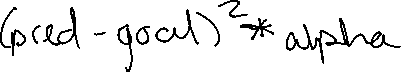
# Neural Predictions

* What is the name of the process of feeding input into a neural network eventually calculate a prediction?
  + Forward propagation
  + When we look at weights, it can be helpful to interpret the importance of the input it’s scaling by looking at the weight value itself. How do we interpret inputs that have weights that are: greater than 1, equal to 1, between 0 and 1, equal to 0, negative?
    - Greater than 1:
    - Equal to 1: directly related to output
    - Between 0 and 1: less or more important
    - Equal to 0: nullifies value
    - Negative: negative correlation
  + ~~Be able to interpret the author’s language of using Big/Small logical ORs, ANDs and NOTs when referring to weight values.~~
* What is the name of the process of multiplying the inputs by their corresponding weights and adding these terms together (there are actually names we can use)?
  + Dot product, weighted sum
* What field studies the mathematical operations of matrices and vectors?
  + Linear algebra
* Be able to perform the dot product of two vectors.

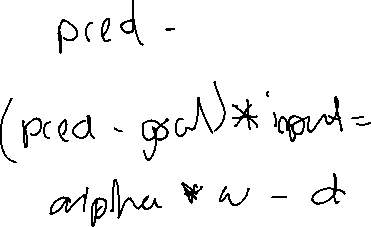
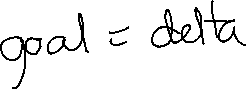


# Neural Learning

* Learning is where we autotune the weights of a neural network to give outputs representative of known true outputs given the input data. In this class, what is the method we use to determine how to properly adjust the weights?
  + Gradient descent
* Calculate and describe the difference between the pure error and the mean squared error (MSE).



* Give at least two reasons why we look at the MSE given that it’s not the pure error.
  + Magnifies big errors, diminishes small errors
  + Always positive – makes comparing the difference in errors easier
* Although simple, what are the drawbacks of the hot & cold method of adjusting weights?
  + Takes too long or overshooting, not computationally efficient
* How do we calculate *weight* *delta*?



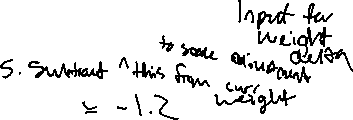
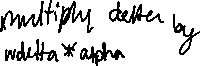
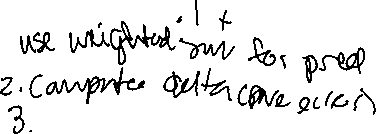
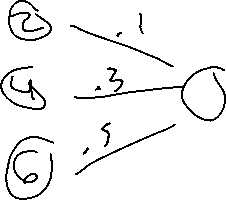
* When calculating delta, one of the terms is taking the prediction and subtracting the goal prediction. What property does this give us?
  + direction
* When calculating delta, we multiply by the input. What three properties does this give us?
  + Scaling, negative reversal, and stopping
* Why do we have alpha in the calculation of our weight delta?
  + Control size of changes, prevent overshooting
* Given a specific goal prediction and neural prediction pair, what does the graph of the MSE as the y-axis and a given weight as the x-axis look like?
  + As mse is closer to 1, the weight will be either left or right from its correct value. When mse is 0 weight will be at its correct value
  + What is our goal when adjusting the weight?
    - To lower the mse
* How does the weight delta relate to the MSE?
  + The derivative of the mse by weight
* What do we call the phenomenon where we overcorrect the weight and the adjustments lead to exploding weights and predictions?
  + overshooting



* + How can we adjust for this?
    - Decrease alpha
* Do a forward propagation to make a prediction and do a backpropagation using gradient descent for a single layer neural network.



* + Know the steps necessary to make this calculation.



# Learning Multiple Weights at a Time

* Is it a guarantee that all weights are in their optimal positions when we determine to stop training our neural network?
  + no
* When starting our weights with random values, are we guaranteed to get the same resulting weights after a given number of iterations? Why or why not?
  + No- they will adjust differently at different rates
* In order for a neural network to learn to make good predictions, is there a specific number of nodes and/or layers that we can determine ahead of time to get good results? (hyper parameters)
  + No – which is one of the biggest problems

# Deep Neural Networks and the Training Cycle

* In application, we’ll use an entire dataset with many observations (rows) during training. What is the term we use to describe this phenomenon? (not learning).
  + Generalization (generalizing the neural net to the whole data set)
* By adding multiple layers to a neural network, what is added to that helps make better predictions?
  + Cross correlation – connects inputs more
* What is the process of adjusting weights in a deep neural network by assigning long -distance error?
  + Back propagation
* What does ReLU stand for, what does it do, and what is its purpose?  
  Rectified Learning Unit
* Ensures the input is outputted only as positive or 0. Makes negative numbers not affect anything negatively  
    
  ReLu(x) = x if x > 0  
   0 if x <= 0