CS 344 Guide 10 – Neural Networks

a. Neural Networks
   i. Terms
      - *Neurons*
         a. *A node in a neural network, typically taking in multiple input values and generating one output value.*
         b. *Calculates output value by applying activation function (nonlinear transformation) to weighted sum of input values.*
      - *Hidden layers*
         a. *A synthetic layer in a neural network between the input layer (features) and output layer (prediction)*
         b. *Typically contains an activation function for training.*
      - *Activation function*
         a. *A function that takes in the weighted sum of all of the inputs from the previous layer and then generates and passes an output value (typically nonlinear) to the next layer.*
   ii. Compare and contrast handling non-linearities using feature crosses vs. neural networks.
      - Feature crosses:
         a. Takes the cross product of 2 or more features to create a synthetic feature.
         b. Includes one-hot vector encoding.
         c. Uses bucketing, binning, etc.
         d. Results in adding more dimensions to the feature set and could create sparse feature vectors.
         e. Limited in the non-linear problems it can solve.
      - Neural networks:
         a. Adds a non-linear transformation layer where non-linearity is encoded via the use of an activation function
            a. Sigmoid, ReLU, tanh, etc.
            b. Any mathematical function can be used.
         b. Hidden layers are added to the model, which could each have a different activation function.
         c. Uses back propagation.
         d. Can solve complex non-linear problems.
   iii. How does a neural network model non-linearities?
      - By directly introducing non-linearity by piping each hidden layer node through a non-linear function (activation function)
b. Training Neural Networks
   i. Terms
      - *Vanishing/exploding gradients*
         a. *Vanishing gradient problem:*
            a. *Tendency for gradients of early hidden layers of deep neural network to become flat (low).*
            b. *Increasingly lower gradients result in increasingly smaller changes to weights on nodes leading to little or no learning.*
         b. *Exploding gradient problem:*

a. *Tendency for gradients in deep neural networks to become steep (high).*
b. *Increasingly higher gradients result in increasingly larger changes to weights on nodes leading to NaN issue and non-convergence.*

- *Dead RELUs*
  a. Failure of gradients to propagate due to weights in a ReLU unit being below zero and the ReLU activation function normalizing them all to 0.
  b. Contributes nothing to network output and gradients cannot flow through during back propagation.
     a. With gradient source cut off, ReLU may not change enough to bring weighted sum back above 0.
- *Dropout*
  a. *A form of regularization in training deep neural networks.*
  b. *Removes a random selection of a fixed # of units in a network layer for a single gradient step.*
  c. *0.0 = no dropout regularization*
  d. *1.0 = drop everything*
  e. *Use intermediate values to be useful.*

ii. Give a general explanation of how *backpropagation* works.
- URL: https://hmkcode.github.io/ai/backpropagation-step-by-step/
- Construct a simple neural network consisting of input nodes, hidden layers with activation functions, and an output node.
- Connect nodes in neighboring layers with weights, forming the network parameters.
- Each node should have a non-linear activation function to model non-linearity.
- Calculate the forward propagation to obtain the output.
- Calculate the error function, which is the deviation between the predicted output value and actual target values.
- Calculate the back propagation – update weights using gradient descent.
- Rinse and repeat recursively using updated weight values each iteration.

iii. Of what value are *normalized* feature vectors?
- Helps speed the convergence of neural networks.
- Helps gradient descent converge.
- Avoids the NaN trap.
- Roughly zero-centered [-1,1] range often works well.
- Avoid outlier values too.

c. Multi-Class Neural Networks

i. Terms
- *One-vs-All*
  a. *Given classification problem with N possible solutions, then solution consists of N separate binary classifiers (one for each outcome)*
     a. *Example: animal vs. not animal.*
- *Softmax*
  a. *A function that provides probabilities for each possible class in a multi-class classification model.*
  b. *Probabilities add to 1.0*
- *Logits*

a. *Vector of raw (non-normalized) predictions that a classification model generates.*
b. *Typically becomes an input to the softmax function in a multi-class classification problem.*

ii. Does the softmax layer have to have the same number of nodes as the output layer? If so, why; if not, why not?

- It must have the same # of nodes as the output layer
- Because it must assign a probability for each class and altogether they must add to 1. If it doesn't assign a probability for each class, the combined probability for all classes may not add to 1.
- Ask Professor VanderLinden.
- URL: https://stats.stackexchange.com/questions/281887/if-softmax-is-used-as-an-activation-function-for-output-layer-must-the-number-o