ELE 364: Assignment #5

1. (10 pts)

   (a) (3 pts) Show a perceptron that implements a two-input NAND (AND followed by NOT) function.

   (b) (7 pts) Show a perceptron network that implements the Boolean expression $f = abc + de$ with an interconnection of only three NAND perceptrons.

2. (10 pts) Derive the derivative of the $\tanh(z)$ activation as a function of the derivative of $\text{logistic}(2z)$.

3. (10 pts) Consider a two-layer feed-forward neural network with an input layer with two inputs, 1 and 2, a hidden layer with one neuron, 3, and an output layer with one neuron, 4. This neural network has five weights. Suppose the activation function is ReLU for the hidden layer and linear for the output layer. Initialize all weights to 0.1, then determine their values after one training iteration of the backpropagation algorithm. Assume a learning rate of 0.2 and the following training data instance: (1,1;0), where the first two values denote the inputs and the third value the desired output.

4. (10 pts) Suppose we use dropout to avoid overfitting. Consider a hidden layer with five neurons. Suppose it uses the logistic activation function. Let the weighted sums fed to the five activations be -0.18, -0.30, 0.52, -0.10, 0.74, respectively, for some input pattern. A DropMask vector (1,0,1,0,1) has been obtained. If the probability that a neuron will not be dropped is 0.25, what are the activation values after dropout has been implemented?

5. (20 pts) **Coding project**

   In this project, you will train multi-layer neural networks to determine whether transactions made by credit cards are fraudulent or not. The dataset consists of 30 different descriptive features. The target feature is a binary variable, where 1 indicates that the transaction is fraudulent and 0 that it is not. This dataset is highly imbalanced, the positive class (fraud) accounts for just 0.172% of all transactions.

   When you train multi-layer neural networks, you will investigate the impact of different hyperparameters, including layer width, network depth, and the choice of the hidden activation function. You will also experiment with early stopping as a regularization strategy.

   See the Jupyter notebook for more details.