Anna Bentler, Bernette Chan, Haley Tran, Zarina Yunis

Final Project - Neural Network on Spam Emails

What We Did

Final Class (our Main Class)

- Parsing and normalizing data & separating into features/targets
- Use randomly selected indices to split data into training and testing/validation datasets
- Train neural network over the given number of epochs, updating weights and biases through backpropagation
- Test neural network with training dataset
- Compute testing metrics (true/false positives/negatives, accuracy, precision, recall, f1 score)

Neuron Class

- Variables to define weights, biases, delta for each neuron (and getters and setters)
- Method to compute weighted sum and return result of sigmoid activation function
- Weight update methods (called in backPropagate)

Layer Class

- Variables and methods for storing and retrieving neurons in a network layer

Neural Network Class (imported contents from BackPropagation class inside)

- Constructor for neural network that includes number of layers and neurons per layer
- Compute iteration (pass inputs through each layer to produce an output)
- Backpropagate method that calculates gradients and updates weights and biases to minimize error function

Hyperparameter Tuning

- Implemented grid search (iterate through different combinations of hyperparameters) to determine best number of hidden layers, best learning rate, best number of epochs
- Model is re-initialized and computed with ideal hyperparameters

How It Works

The dataset is parsed and normalized to values between 0 and 1, and the features and targets are separated into ArrayLists. Then the neural network is initialized with the specified layers (three, in our case) where the input layer has 57 nodes, then 57 nodes for the first hidden layer and 8 nodes for the second hidden layer, and 1 node for the output layer. The entire neural network structure is adaptable to any number of layers, neurons, and hyperparameters. Then, the network trains across multiple epochs, and for each epoch it performs forward propagation to compute outputs and backpropagation to adjust weights based on the errors. We test the neural network's performance by evaluating the testing set on accuracy, precision, recall, and F1 score. We also tuned the hyperparameters on the validation set to classify spam emails more accurately.

Results

- The model has accuracy of 0.64, the proportion of correctly classified emails.
- We have a precision score of 0.7409, meaning the majority of predicted spam was spam.
- Our recall was 0.6060, indicating that the model correctly identified 0.6060 of all actual spam.
- The F1 score was 0.6667, is the overall performance balancing precision and recall.
- Experienced struggles with the model converging incorrectly and skewing results towards non-spam. We debugged extensively and made improvements but the issue still remains to some degree.
- Through debugging, we learned how the learning rate, epochs, and activation function can impact the end result. We are satisfied with the way we were able to apply neural networks to the real-world problem of spam email detection.