Homework 2
COMP 379
Brian Nguyen
10/1/2021

This program is meant to implement a Perceptron, Adaline, and baseline model to a variety of linearly and non-linearly separable datasets, comparing their performance and accuracy in analyzing training datasets and predicting upon testing datasets.

To implement and test the Perceptron model, I created two datasets, one linearly (LS) and one non-linearly (NLS) separable, that contained features regarding a basketball player's rebounds per game (*RPG*), height (*height*), and position grouping (*position*) as the class label. Each dataset was randomly split into training (70%) and testing (30%) datasets (see *Fig. 1*). The Perceptron was able to converge on a decision boundary for the LS training dataset after only 3 epochs with a learning rate of 0.1 (see *Fig. 2*). The model also had a prediction accuracy rate of 100% on the LS testing dataset (see *Fig. 3*). However, the Perceptron was not able to converge on a decision boundary for the LS training dataset and only stopped updating its weights because a maximum of 10 epochs was set with a learning rate of 0.1 (see *Fig. 2*). The model had a prediction accuracy rate of 33.33% on the NLS testing dataset (see *Fig. 3*).

To implement and test the Adaline model, the Titanic *train.csv* dataset was first preprocessed by removing non-categorical and non-quantifiable features such as *passengerId*, *Name*, *Ticket*, *Cabin*, *Embarked*, and all samples that were missing values for any of the unremoved features. The *Sex* feature's data was also converted from categorical values (*male*/*female*) to binary (*1*/*0*). The processed dataset was then randomly split into training (70%) and testing (30%) datasets. After some trial-and-error with learning rate, epoch maximum, and random state seed adjustments, the batch-version of Adaline was able to minimize its cost function to 96.32 in 15 epochs with a learning rate of 0.0000001 (see *Fig. 4*). The model also had a prediction accuracy rate of 68.37%, which is the highest accuracy rate of 42 different random state seeds. By analyzing Adaline model's weight vector after its final update, it was apparent that the most predictive feature was *Parch* with the heaviest weight of 0.024, followed by *Pclass* (0.011) and *Fare* (0.010). The least predictive feature was *SibSp* with the lightest weight of -0.023.

A baseline model was also implemented to evaluate whether the Perceptron and Adaline models were behaving in a methodological way. This baseline model multiplied the final weights vector of each Perceptron and Adaline model by a random state seed array of similar dimensions (effectively creating a vector of random weights), calculating the net input of each sample, and making a prediction based on if the net input was a negative number or not. Compared to the Perceptron and Adaline's accuracy rates of 100%, 33.33%, and 68.37% on the LS, NLS, and Titanic testing datasets, the baseline model had accuracy rates of 33.33%, 66.67%, and 50.70%, respectively. It was by pure chance that the baseline model was more accurate than the Perceptron's prediction upon the NLS testing dataset.

Figures Referenced

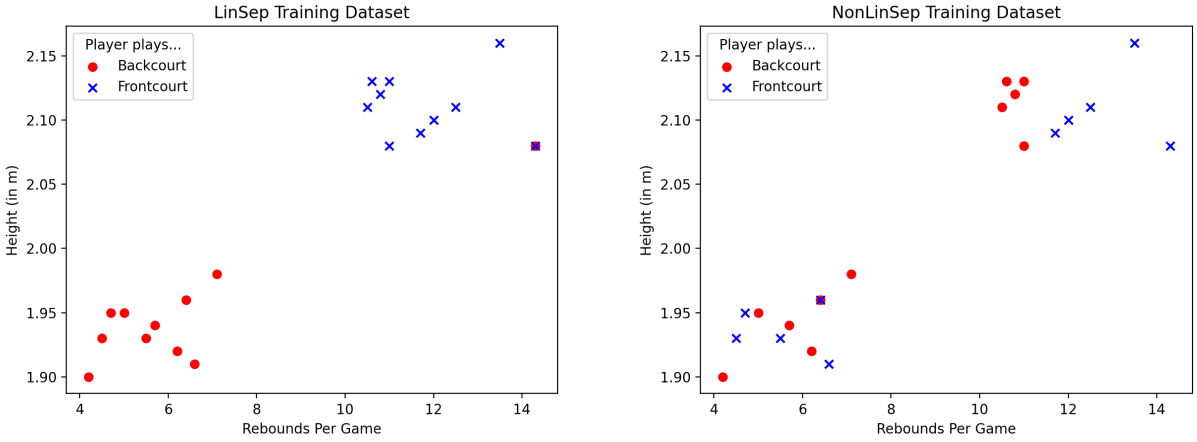

Fig. 1: Perceptron LS and NLS Training Dataset Scatterplots

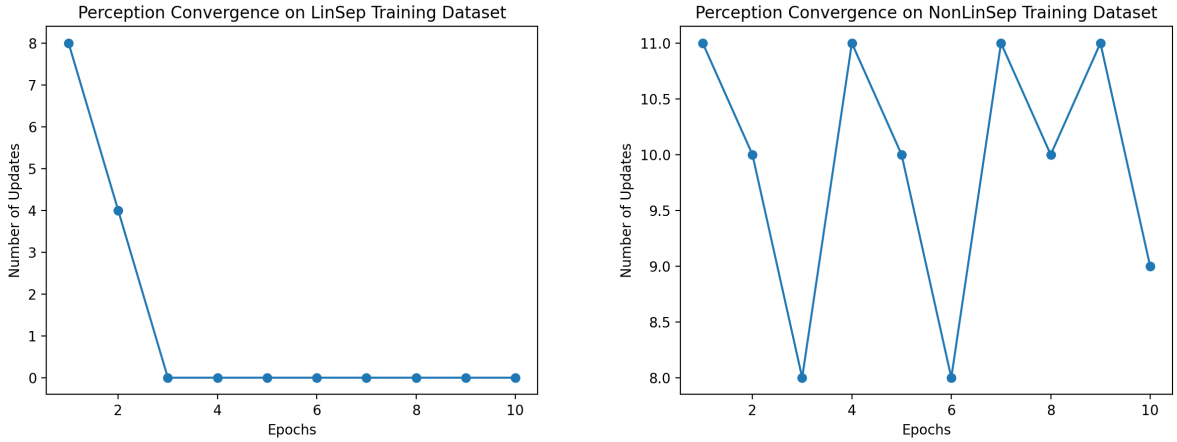

Fig. 2: Perceptron Convergence on LS and NLS Training Datasets

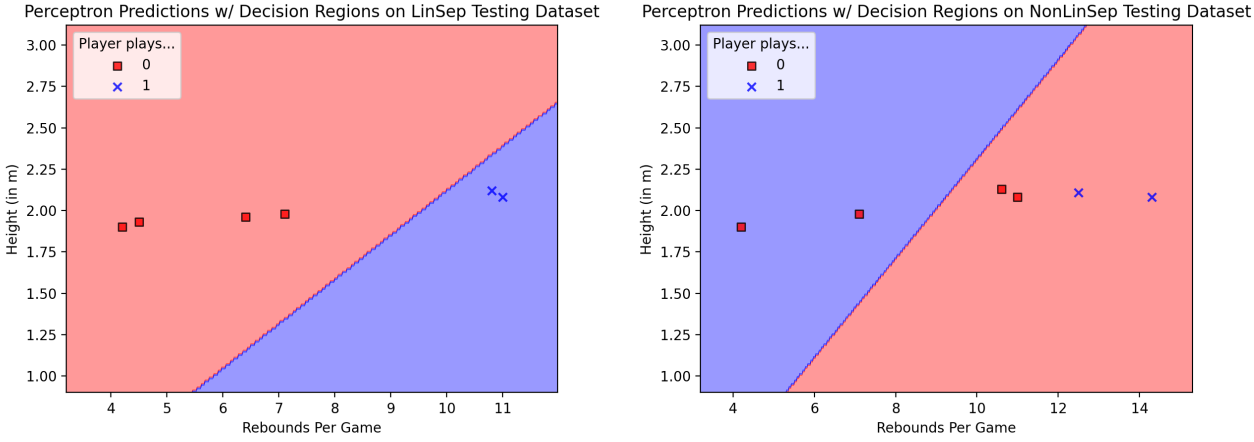

Fig. 3: Perceptron Predictions w/ Decision Regions on LS and NLS Testing Datasets

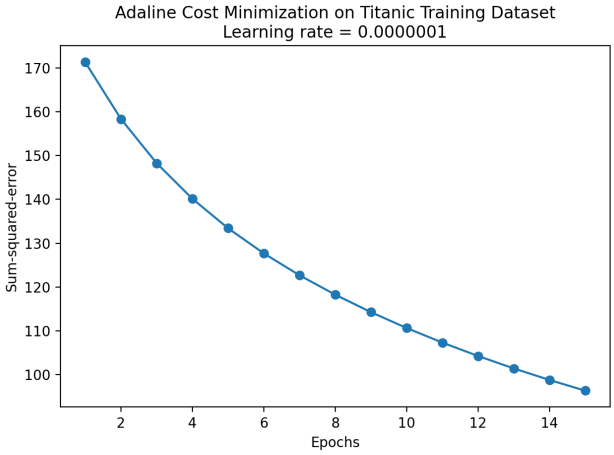

Fig. 4: Adaline Cost Minimization on Titanic Training Dataset