Implementation 2

Group Members
Derek Jackson
Meredith Leung
Dylan Sanderson

Introduction

This report describes the algorithm optimization of online perceptron, average perceptron and kernel perceptron training methods to differentiate between handwritten 3s and 5s. For the given dataset, the results indicate that using a kernel perceptron delivers the most accurate predictions, particularly on iteration 4 with p = 3. Furthermore, the online and average perceptron training methods resulted in large accuracies.

Part 1: Online Perceptron

We ran through 15 iterations of an online perceptron to train and validate our machine learning algorithm. Our training and validation accuracies peaked on iteration 14, with values of 0.966 and 0.9525 respectively (figure 1). **(1b)** We therefore made predictions on our test data using the weights generated from iteration 14 and wrote the results into *oplabel.csv*.

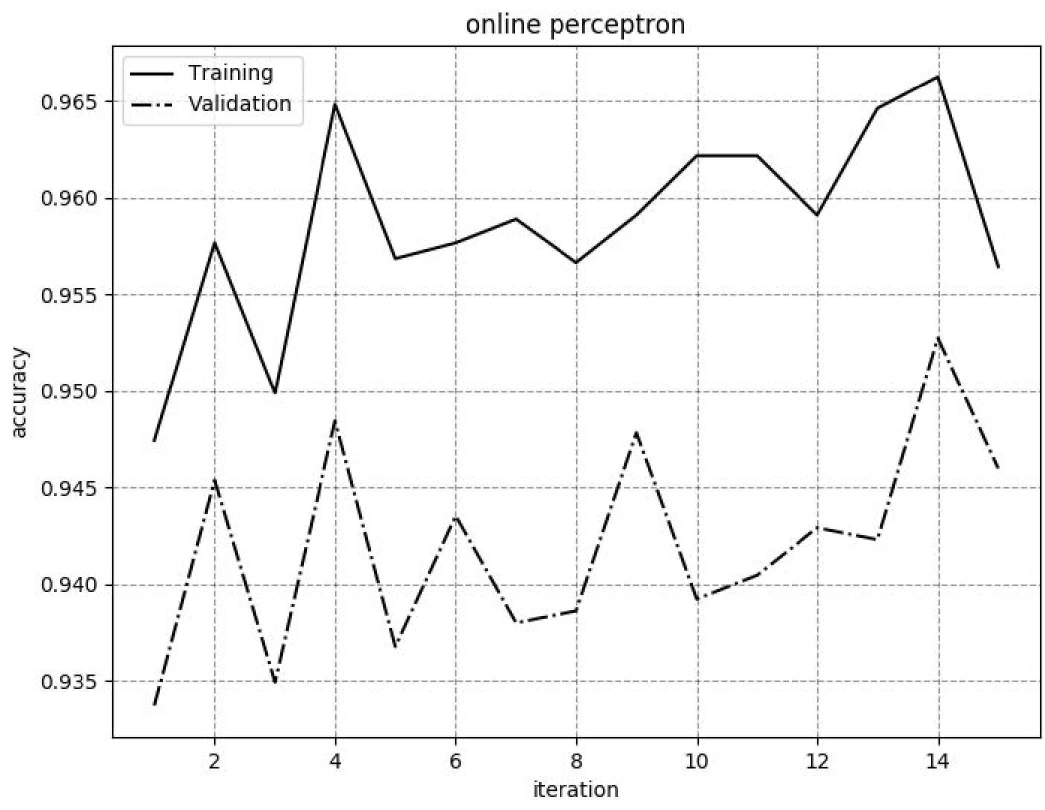

figure 1. Accuracies of training and validation predictions generated from an online perceptron used to differentiate between handwritten numbers 3 and 5.

(1a) Our training accuracy never reaches 100% because, while this method guarantees convergence, it does not guarantee that we converge to the max margin separator. The online perceptron also works best with randomly ordered training examples. So our accuracies may have improved if we had shuffled our data, but for the sake of grading we avoided doing this. More iterations could also affect the end accuracy, but may overfit to the training data. Typically, the online perceptron is not used without modifications (such as the voted or average perceptron) because simple updates can drastically improve the accuracy and generalizability of results.

Part 2: Average Perceptron

In part two, we similarly ran through 15 iterations to train and validate our algorithm, this time using the average perceptron method. (**2a**) Figure 2 shows that both the training and validation accuracies were best on iteration 15.

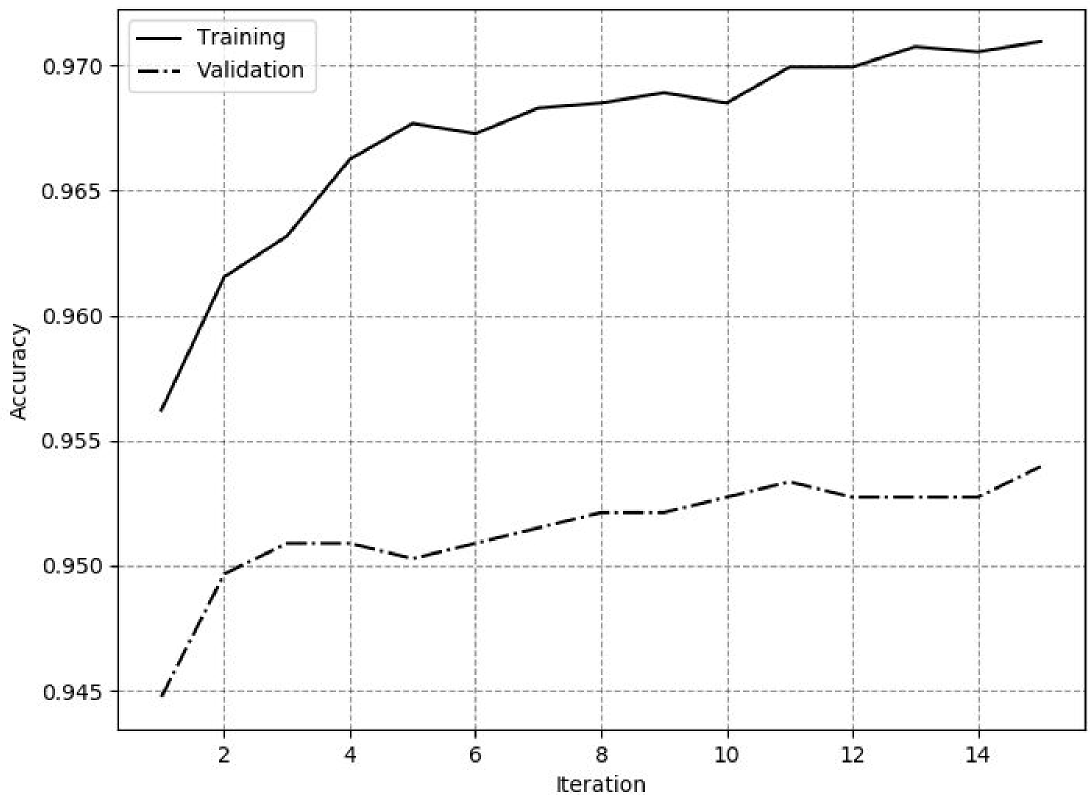

Figure 2: Accuracies of training and validation predictions generated from the average perceptron algorithm.

(**2b**) Comparing the online perceptron method, the accuracies generated from the average perceptron tend to be slightly higher. Furthermore, the results of the average perceptron tend to be smoother than the online perception, and increase nearly monotonically. This is due to the weights being updated based on a running average. In online perception, while the weights of the prior iteration do have an effect on the updated weights (*e.g.* the weight is updated if the prediction is wrong), they are slightly more "independent" then the weights in the average perceptron. In the average perceptron, the weights are updated regardless of the prediction.

(**2c**) Figure 2 shows that iteration 15 results in the largest validation accuracy. The resulting validation accuracy is approximately 0.954. The weights from this iteration were applied to the test data and the results written to *aplabel.csv.*

Part 3: Polynomial Kernel Perceptron

In this section we test how well the kernel perceptron does in predicting handwritten 3s and 5s using 15 iterations and p-values (degrees) in range one through five. The training and validation accuracies for each iteration can be seen in figure 3.

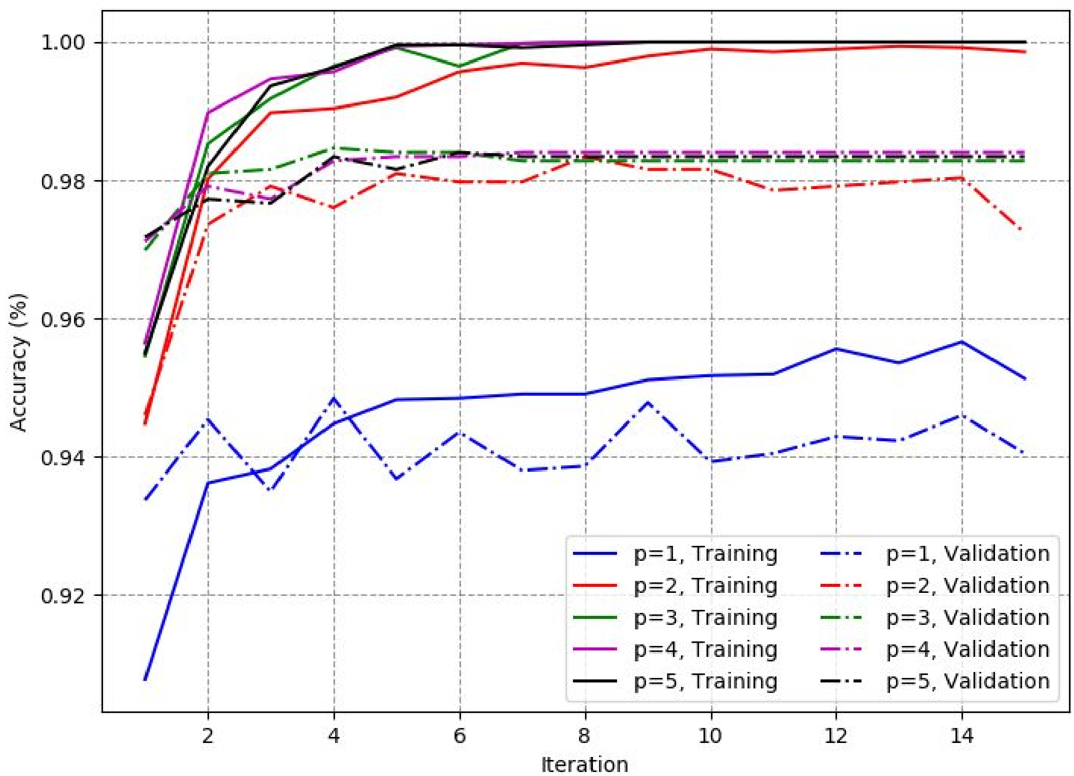

figure 3. Accuracies of training and validation predictions generated from a kernel perceptron used to differentiate between handwritten numbers 3 and 5.

Most polynomial degrees (p) caused the kernel perceptron to settle at a similar accuracy, with p = 1 and p = 2 having lower accuracies. The best performance came from p = 3 on iteration 4 (figure 3 and figure 4). Interestingly, the accuracy from p = 3 decreased after iteration 4. This could be caused from the model slowly overfitting the training data.

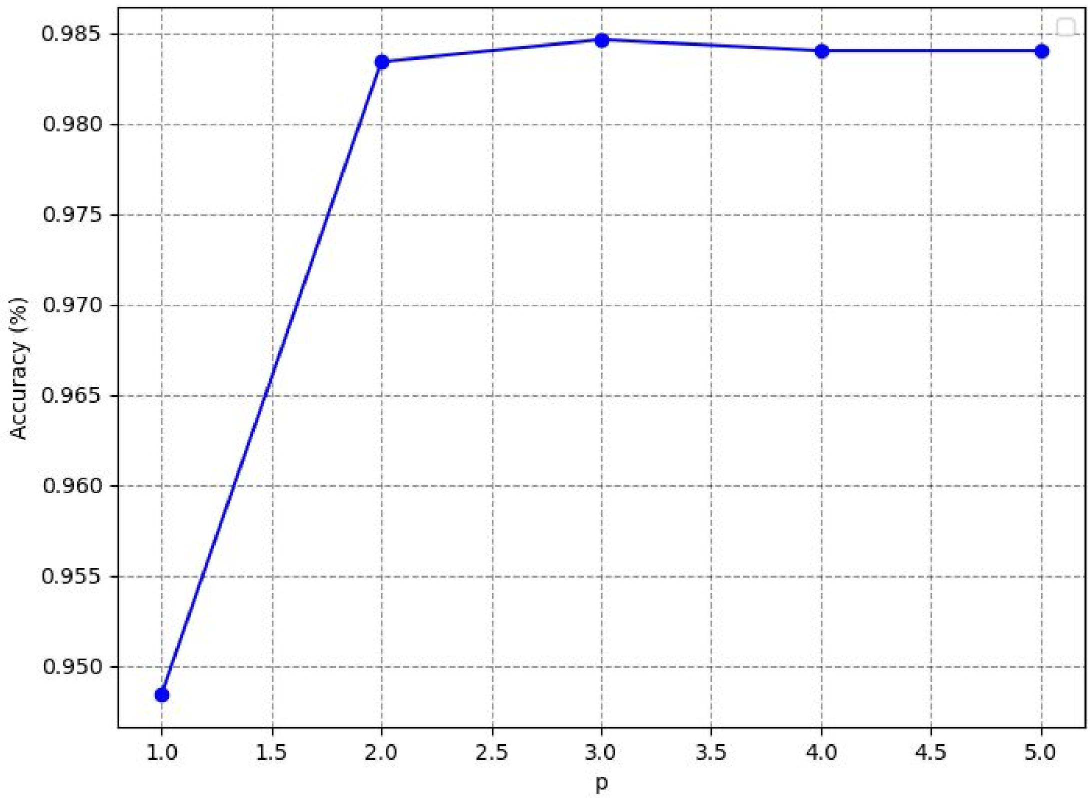

figure 4. Highest accuracies of each p value for the kernel perceptron. Highest accuracy occured with p = 3, on iteration 4.

(3a) A p value greater than 1 causes an increase the accuracy of the kernel perceptron. Higher order degrees allow the margin that separates our data to be more non-linear, where a p value of 1 is linear. This is because a linear plane in a higher dimension can have a non-linear shape in the original input dimension. A p of 3 fits our data best. When p = 4 and 5 it still captures the curve that separates our data fairly well, but these higher order degrees are likely overkill for our dataset.

Discussion of Results and Summary

Our experimentation with different perceptron variations indicates that the kernel perceptron performed best when trying to predict handwritten 3s and 5s. These results followed our expectations as the kernel perceptron allows for a non-linear decision boundary, leading to more flexibility for the perceptron algorithm to find the optimal model. We were surprised by the high performance of the online perceptron and average perceptron algorithm. The lowest accuracy across the online and average perceptron algorithms is approximately 0.935, indicating that these models still perform well.