Artificial Neural Networks

Prof. Dr. Sen Cheng

Nov 25, 2019

Problem Set 8: Perceptron

Tutors: Eloy Parra Barrero (eloy.parrabarrero@rub.de), Olya Hakobyan (olya.hakobyan@rub.de)

1. Load the Iris dataset from *sklearn.datasets* (at sci-kit learn's website you can find an example for how to load and display the data). The dataset contains measures of the flowers of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured for each sample: the length and the width of the sepals and petals respectively, in centimeters. Extract the first two features (septal length and width) and make a scatter plot from them. Use two different colors to distinguish Iris satosa samples from the rest. Is the data linearly separable?

2. Implement the perceptron algorithm from scratch. Use your algorithm to train a perceptron to classify Iris setosa based on septal length and width. Plot the resulting decision boundary together with the data points.

3. Plot the training error after each iteration of the algorithm. Think about what would happen if your design employed a limited number of iterations.

4. Explore the effect of the learning rate on the speed of convergence when you start with $w = 0$. Draw a conclusion.

5. Extract the first and fourth features (septal length and petal width) and make a scatter plot showing Iris virginica vs the rest. What would happen if a perceptron tried to classify Iris virginica?

6. Implement the pocket algorithm, which returns the best set of weights encountered during training instead of the last solution. For each iteration, plot the training error as well as the error for the best solution seen so far. In what way is this modification of the algorithm helpful?

7. (Bonus Question) In the dataset, the samples are ordered by species. However, convergence can be achieved faster if the samples are shuffled. Try running the algorithm after shuffling the data points once in the beginning, and/or shuffling at the beginning of each epoch. Compare the number of iterations it takes for the basic perceptron algorithm to find a solution in each of these cases, and contrast them to the results you got before. You can also look at the effect of shuffling on the pocket perceptron by plotting the evolution of the best classification error over iterations. In order to get more meaningful results, train the classifiers a number of times and plot the mean and standard deviations of the indicated variables.