

CS 445/545: Machine Learning, Spring 2017

Homework 1: Perceptrons , Due Tuesday, April 18, 2017, 2pm

For this homework you will write code to implement perceptrons and the perceptron learning algorithm.

If you haven't already, please sign up for the class mailing list:

<https://mailhost.cecs.pdx.edu/mailman/listinfo/mlspring2017>

Perceptrons

You will train 10 perceptrons that will, as a group, learn to classify the handwritten digits in the MNIST dataset. See the class slides for details of the perceptron architecture and perceptron learning algorithm. Each perceptron will have 785 inputs and one output. Each perceptron's target is one of the 10 digits, 0–9.

Preprocessing: Scale each data value to be between 0 and 1 (i.e., divide each value by 255, which is the maximum value in the original data). This will help keep the weights from getting too large. Randomly shuffle the order of the training data.

Training: Train the perceptrons with three different learning rates: $\eta = 0.001, 0.01, \text{ and } 0.1$.

For each learning rate:

1. Choose small random initial weights, $w_i \in [-.05, .05]$. Recall that the bias unit is always set to 1, and the bias weight is treated like any other weight.
2. Compute the accuracy on the training and test sets for this initial set of weights, to include in your plot. (Call this “epoch 0”). [For instructions on how to compute accuracy, see below.]
3. Repeat for 50 epochs: cycle through the training data, changing the weights (according to the perceptron learning rule) after processing each training example \mathbf{x}^k by each perceptron as follows:
 - For each perceptron, compute $\mathbf{w} \cdot \mathbf{x}^k$, y^k , and t^k at each output unit.

Recall:

$$t^k = \begin{cases} 1 & \text{if the output unit is the correct one for this training example} \\ 0 & \text{otherwise} \end{cases}$$

$$y^k = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x}^k > 0 \\ 0 & \text{otherwise} \end{cases}$$

- Update all weights in each perceptron:

$$w_i \leftarrow w_i + \eta(t^k - y^k)x_i^k,$$

(Note that this means that for some output units $(t^k - y^k)$ could be zero, and thus the weights to that output unit would not be updated. That's okay!)

4. After each epoch (one cycle through training data), compute accuracy on training and test set (for plot), without changing weights. The output with the highest value of $\mathbf{w} \cdot \mathbf{x}^k$ is the prediction for this training example. For example, if the output corresponding to the digit '7' has the highest value, the prediction of the group of perceptrons is '7'. This prediction is used for computing accuracy. Accuracy on a set of examples (e.g., training or test) is the fraction of correct classifications (prediction = true class) on that set.

Report: Your report should include the following:

1. A one-paragraph description of the experiment.
2. **For each learning rate:**
 - Plot of accuracy (fraction of correct classifications) on the training and test set at each epoch (including epoch 0), along with comments as to whether you are seeing either oscillations or overfitting.
 - Confusion matrix on the test set, after training has been completed.
 - Short discussion of confusion matrix: which digits are classified most accurately, and which digits tend to be confused with one another?
3. Short discussion comparing results of different learning rates. Did you see any difference in the results when using different learning rates?

Here is what you need to turn in:

- Your report, with all the information requested above, in pdf format.
- Your well-commented code.

How to turn it in (read carefully!):

- Send these items in electronic format to mm@pdx.edu by 2pm on the due date. No hard copy please!
- Put "MACHINE LEARNING HW 1" in the subject line.

If there are any questions on this assignment, don't hesitate to ask me or e-mail the class mailing list.

Policy on late homework: If you are having trouble completing the assignment on time for any reason, please see me before the due date to find out if you can get an extension. Any homework turned in late without an extension from me will have 5% of the grade subtracted for each day the assignment is late, up to a maximum penalty of 25%.