CS-584 – Assignment 3 (5%)

Discriminative learning

Due by: March 29, 2016

Assignment Specifications

In this assignment you will implement algorithms for discriminative learning. You have to use two or more external data sets. Links to data sets are available on the course web-page (e.g. the UCI Machine Learning Repository http://archive.ics.uci.edu/ml/). At least one dataset should be of images (e.g. digits or faces). It is essential that you evaluate the performance of each algorithm you implement and the effects of varying different parameters on the performance of the learning algorithm. Use cross validation to test performance. The grade for this assignment will be based in part on the performance of your implementation and on the thoroughness of your evaluation. Make sure to explain the results you obtain and do not unnecessarily repeat similar results. The code you write should be modular and well documented. The program needs to be written in Python.

1. Logistic regression:
   - (a) Implement the logistic regression algorithm for two-class discrimination.
   - (b) Use non-linear combinations of inputs to increase the capacity of the classifier.
   - (c) Implement the logistic regression algorithm for K-class discrimination.
   - (d) Evaluate the performance of the algorithms you implemented.
2. Multilayer perceptron:
   - (a) Assume a two layer feedforward MLP with a single output where all the elements use sigmoid activation (including the output unit). Assume a training dataset $\{x^{(i)}, y^{(i)}\}_{i=1}^{m}$ where $x^{(i)} \in \mathbb{R}^n$ and $y^{(i)} \in \{0, 1\}$. Derive the backpropagation update equations for the weights of the output by minimizing the following error function: $\frac{1}{2}\sum_{i=1}^{m}(y^{(i)} - \hat{y}^{(i)})^2$. Compare the result you obtain to that obtained in class using maximum likelihood estimation.
   - (b) Implement a two layer feedforward MLP for three class-classification. Evaluate the performance of the algorithm you implemented as function of the number of elements in the hidden layer and as function of the learning rate parameter. Evaluate the effect of other parameters in your algorithm. Make sure to use momentum in your implementation. Compare your implementation to the scikit-learn implementation.

Follow the submission instructions of assignment 1.

Backpropagation equations

- Update equations:

$$\begin{aligned} v_j &\leftarrow v_j - \eta \Delta v_j && (1) \\ w_j &\leftarrow w_j - \eta \Delta w_j && (2) \end{aligned}$$

- Single output:

$$\begin{aligned} \Delta v &= \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) z^{(i)} && (3) \\ \Delta w_j &= \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) v_j z_j^{(i)} (1 - z_j^{(i)}) x^{(i)} && (4) \end{aligned}$$

- Multiple outputs:

$$\begin{aligned} \Delta v_j &= \sum_{i=1}^{m} (\hat{y}_j^{(i)} - y_j^{(i)}) z^{(i)} && (5) \\ \Delta w_j &= \sum_{i=1}^{m} \left(\sum_{l=1}^{k} (\hat{y}_l^{(i)} - y_l^{(i)}) v_{lj} \right) z_j^{(i)} (1 - z_j^{(i)}) x^{(i)} && (6) \end{aligned}$$