Homework 3

Aleck Zhao

April 6, 2018

1 Analytical (50 points)

1) Deep Neural Networks (12 points)

(a) Consider a 2-layer neural network, with M input nodes, Z nodes in the hidden layer and K nodes in the output layer. The network is fully connected, i.e. every node in the $n-1$th layer is connected to every node in the nth layer. However, for your application of interest, you suspect that only some of the nodes in the input are relevant. How would you modify the objective function to reflect this belief?

(b) Consider a N layer neural network. We could (a) train the entire network at once using back-propagation or (b) pre-train each layer individually, and then tune the final network with back-propagation. Will (a) and (b) converge to the same solution? Why would we favor strategy (a) vs. strategy (b)?

(c) Consider a $N \geq 2$ layer neural network with a single node in the output layer. We wish to train this network for binary classification. Rather than use a cross entropy objective, we want to take a max-margin approach and ensure a margin of $\gamma = 1$. Describe the structure of the last layer of the network, including the final activation function, and the training objective function that implements a max-margin neural network. What are the benefits of this network compared to one trained with cross entropy? Will a max-margin trained neural network learn the same decision boundary as an SVM?

2) Adaboost (12 points) There is one good example at $x = 0$ and two negative examples at $x = \pm 1$. There are three weak classifiers are

$$\begin{aligned} h_1(x) &= 1 \cdot \mathbf{1}(x > 1/2) - 1 \cdot \mathbf{1}(x \leq 1/2), \\ h_2(x) &= 1 \cdot \mathbf{1}(x > -1/2) - 1 \cdot \mathbf{1}(x \leq -1/2) \\ h_3(x) &= 1. \end{aligned}$$

Show that this data can be classified correctly by a strong classifier which uses only three weak classifiers. Calculate the first two iterations of AdaBoost for this problem. Are they sufficient to classify the data correctly?

Solution. We have

$$\mathcal{D} = \{(0, 1), (-1, -1), (1, -1)\}$$

We wish to find constants $\alpha_1, \alpha_2, \alpha_3$ such that

$$h(x) = \alpha_1 h_1(x) + \alpha_2 h_2(x) + \alpha_3 h_3(x) = y$$

for our three training examples. Evaluating h_1, h_2, h_3 at each of x_1, x_2, x_3, we have the equations

$$\begin{aligned} -\alpha_1 + \alpha_2 + \alpha_3 &= 1 \\ -\alpha_1 - \alpha_2 + \alpha_3 &= -1 \\ \alpha_1 + \alpha_2 + \alpha_3 &= -1 \end{aligned}$$

which has solution $(\alpha_1, \alpha_2, \alpha_3) = (-1, 1, -1)$, so the data is correctly classified by the strong classifier

$$h(x) = -h_1(x) + h_2(x) - h_3(x)$$

□

¡++¿

3) Ensemble Methods (12 points) Consider the following binary classification Boosting algorithm.

1. Given $\{\mathbf{x}_i, y_i\}_{i=1}^N$, number of iterations T, weak learner f.
2. Initialize $\mathcal{D}_0$ to be a uniform distribution over examples.
3. For each iteration $t = 1 \dots T$:
   (a) Train a weak learner f on the data given $\mathcal{D}_t$ to produce hypothesis h_t.
   (b) Compute the error of h_t as $\epsilon_t = P_{\mathcal{D}_t}[h_t(\mathbf{x}_i) \neq y_i]$
   (c) Compute $\alpha_t = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$
   (d) Update $\mathcal{D}$ as:
   $$\mathcal{D}_{t+1}(i) = \frac{\mathcal{D}_t(i)}{Z_t} \times \begin{cases} \exp(-\alpha_t + (T-t)/T) & \text{if } h_t(\mathbf{x}_i) = y_i \\ \exp(\alpha_t + (T-t)/T) & \text{otherwise} \end{cases}$$
4. Output final hypothesis $H(\mathbf{x}) = \text{sign}\left\{\sum_{t=1}^T \alpha_t h_t(\mathbf{x})\right\}$

Z_t is a normalization constant so that $\mathcal{D}$ is a valid probability distribution.

Describe the difference between this algorithm and the AdaBoost algorithm we learned about in class. What problem of AdaBoost is this change designed to fix? How does changing the algorithm's user provided parameter affect this behavior?

4. Overfitting in Clustering (14 points) Given the data set $x_1, ..., x_n$, we want cluster the data using the K-means algorithm. The K-means algorithm aims to partition the n observations into k sets ($k < n$) $S = \{S_1, S_2, \dots, S_k\}$ so as to minimize the within-cluster sum of squares

$$\underset{S=\{S_1,\dots,S_k\}}{\text{argmin}} \sum_{j=1}^{k} \sum_{x_i \in S_j} \|x_j - \mu_j\|_2^2 \tag{1}$$

where μ_j is the mean of points in S_j.

(a) Let γ_k denote the optimal value of the objective function, prove γ_k is non-increasing in k.

Proof. Suppose $S = \{S_1, \cdots, S_k\}$ and $T = \{T_1, \cdots, T_{k+1}\}$ are the partitions that give the optimal objective function values for k and $k+1$. □

¡++¿

(b) Suppose we modified the objective function as follows:

$$\underset{S=\{S_1,\dots,S_k\}}{\text{argmin}} \sum_{j=1}^{k} \sum_{x_i \in S_j} \max(\|x_j - \mu_j\|_2^2, \tau) \tag{2}$$

where τ is some (given) constant and γ'_k is the optimal value of this new objective function. Compare the values of γ_k and γ'_k ($<, \leq, =, \geq, >$) and prove this relation.

(c) K-medoids is an algorithm similar to K-means. Both K-means and K-medoids attempt to minimize the squared error but unlike K-means, K-medoids chooses a provided example as a cluster center (medoids) rather than the mean of a subset of the examples. For a given data set $\mathbf{X}$, compare the optimal clusterings produced by K-means and K-medoids ($<, \leq, =, \geq, >$) and prove this relation.

(d) Suppose you wanted to select k (the number of clusters) to minimize the objective function. Should you work with objective 1 or 2? If 2, how does your choice of τ effect your choice of k?