Homework Assignment 3

COGS 181: Neural Networks and Deep Learning

Due: October 22, 2017, 11:59pm
Instructions: Please answer the questions below, attach your code, and insert figures to create a pdf file; submit your file to TED (ted.ucsd.edu) by 11:59pm, 10/22/2017. You may search information online but you will need to write code/find solutions to answer the questions yourself.

Late Policy: %5 of the total points will be deducted on the first day past due. Every 10% of the total points will be deducted for every extra day past due.

System Setup: You are free to choose either pip or anaconda as the package installer. After the installation of one of the installer, type pip/conda install $PACKAGE_NAME in the terminal to install python packages. For more information, see Piazza system setup post.

Grade: ____ out of 100 points

1 (35 points) Perceptron

We apply perceptron learning algorithm to learn a linear classifier. In Perceptron Learning Algorithm, the activation rule is defined as:

$$f(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{w}^T\mathbf{x} + b \geq 0 \\ -1, & \text{otherwise} \end{cases}$$

and the optimal solution of $\mathbf{w}^*$ is $\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_{i=1}^{n} \mathbf{1}(y_i \neq \text{sign}(\mathbf{w}^T\mathbf{x} + b))$, where $\mathbf{1}(\cdot)$ is the indicate function, and **sign**$(\cdot)$ is the sign function.

Download the data file **Q1_data.txt** from the course website. The dataset was originally downloaded from the *UC Irvine Machine Learning Repository*, and has been modified for this class.

Dataset Description

The dataset contains 2 classes of 50 instances each, where each class refers to a type of the iris plant. Each instance contains 5 attributes, which are sepal length in cm, sepal width in cm, petal length in cm, petal width in cm and class (*Iris-setosa*, *Iris-versicolor*). The first 4 attributes are numerical, while the last attribute is categorical. In this task, you are going to use the first 4 attributes to predict the 5th attribute by learning a binary classifier.

In data file, each line stands for an instance. The examples are shown below:

sepal length/cm	sepal width/cm	petal length/cm	petal width/cm	plant type
5.0	3.3	1.4	0.2	*Iris-setosa*
7.0	3.2	4.7	1.4	*Iris-versicolor*

Training/Test set

The dataset currently contains 100 instances. For this task, we split 100 instances into two sets, which are training set and test set. The classifier will be trained on the training set, and you will be asked to report your classification result on the test set. You need to combine the first 35 instances of each class into the training set, and leave the rest of them into the test set. We simply denote $\mathbf{X}_{train}$ as the set of all training data points, $\mathbf{Y}_{train}$ as the set which contains their corresponding class labels.

Pseudocode for the Perceptron Learning Algorithm

We provide a piece of pseucode for the perceptron learning algorithm, as is shown in Algorithm 1.

Algorithm 1 Perceptron Learning Algorithm

Data: training data points $\mathbf{X}$, and training labels $\mathbf{Y}$;
Randomly initialize parameters $\mathbf{w}$ and b; pick a constant $\lambda \in (0, 1]$, which is similar to the step size in the standard gradient descent algorithm (by default, you can set $\lambda = 1$.
while *not every data point is correctly classified* **do**
  randomly select a data point $\mathbf{x}_i$ and its label y_i;
  compute the model prediction $f(\mathbf{x}_i)$ for $\mathbf{x}_i$;
  if $y_i == f(\mathbf{x}_i)$ **then**
    continue;
  else
    update the parameters $\mathbf{w}$ and $\mathbf{b}$:
    $\mathbf{w}_{t+1} = \mathbf{w}_t + \lambda(y_i - f(\mathbf{x}_i))\mathbf{x}_i$
    $\mathbf{b}_{t+1} = \mathbf{b}_t + \lambda(y_i - f(\mathbf{x}_i))$
  end
end

1.1 Programming

Implement your own perceptron learning algorithm, and train a linear classifier on the training set. Plot the error rate on the training set during training. **HINT:** You might need to set the number of iterations to be large enough.

1.2 Decision Boundary

Derive the decision boundary of your trained classifier.

1.3 Test

Classify the data in the test set and report these error metrics shown below:

1. Accuracy (The percentage of correctly classified data points in the test set).

   (For the following 3 error metrics, you can think of one of the classes as positive, and the other on as negative.)

2. Precision ($= \frac{\sum \text{True positive}}{\sum \text{Test outcome positive}}$)
3. Recall ($= \frac{\sum \text{True positive}}{\sum \text{Condition positive}}$)
4. F-value ($= \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$)

2 (15 points) Logistic Regression (1)

For a logistic regression function with input $x \in \mathbb{R}$ and output $y \in \{0, 1\}$, the probability of $P(y = 1|x) = \dfrac{e^{\alpha+\beta x}}{1 + e^{\alpha+\beta x}}$.

1. Please write down the formulation of $P(y = 0|x)$.
2. Show that $[P(y = 1|x)]^y \times [P(y = 0|x)]^{1-y} = \dfrac{1}{1 + e^{-(2y-1)(\alpha+\beta x)}}$
3. What is the decision boundary for classifier:

$$y = \begin{cases} 1, & \alpha + \beta x \geq 0, \\ 0, & else \end{cases}$$

and how is it related to $P(y = 1|x)$.

3 (40 points) Logistic Regression (2)

In logistic regression algorithm, y is the label for each data point, and it can be either 0 or 1. Here, we define our approximate function to

$$h(\mathbf{x}; \mathbf{w}, b) = \sigma(f(\mathbf{x}; \mathbf{w}, b)) = \frac{1}{1 + e^{-f(\mathbf{x};\mathbf{w},b)}} = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x}+b)}}$$

where $\mathbf{x} \in \mathbb{R}^K$ is the feature, and b is the bias, and $\mathbf{w} \in \mathbb{R}^K$ contains the set of parameters $\{w_0, w_1, ..., w_K\}$. The output of $h(\mathbf{x}; \mathbf{w}, b)$ is called the confidence. When $h(\mathbf{x}; \mathbf{w}, b) >= 0.5$, the classifier outputs 1 for the given $\mathbf{x}$; otherwise, the classifier outputs 0.

Download the data file **Q3_data.txt** from the course website. This dataset is still modified from **Iris** dataset.

Dataset Description

The dataset contains 2 classes of 50 instances each, and the two classes are *Iris-versicolour*, *Iris-virginica*), respectively. The data format is exactly the same with the data we used in Q1.

Training/Test set

The dataset currently contains 100 instances. You need to combine first 15 instances of each class into test set, and leave the rest of them into the training set.

3.1 Gradient Descent

In this task, the first 4 attributes and the 5th attribute of i-th instance are considered as its feature $\mathbf{x}^{(i)}$ and its label $y^{(i)}$.

The goal is to train a classifier based on logistic regression to predict the correct label $y^{(i)}$ from the given feature $\mathbf{x}^{(i)}$. We define a loss function which measures the distance between the correct label and the prediction from the classifier, as is shown below:

$$\mathcal{L}(\mathbf{w}) = -\sum_i \left(y^{(i)} \ln p^{(i)} + (1 - y^{(i)}) \ln(1 - p^{(i)})\right)$$

where

$$p^{(i)} = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x}^{(i)}+b)}},$$

which is called the sigmoid function.

The training procedure is to minimize the loss function on the training set. Consider ***gradient descent*** method to find the optimal $\mathbf{w}^*$ in $h(\mathbf{x}; \mathbf{w}, b)$.

1. Derive $\dfrac{\partial \mathcal{L}(\mathbf{w})}{\partial \mathbf{w}}$.
2. Suppose that the learning rate is denoted as α. Write down the update rule for $\mathbf{w}$.

3.2 Training

Implement your own ***gradient descent*** algorithm, and train a binary classifier based on logistic regression. You might need to vectorize your algorithm in order to run efficiently. Not that b is also supposed to be learned in your gradient descent algorithm.

3.3 Decision Boundary

Derive the decision boundary of your trained classifier.

3.4 Test

Classify the data in the test set and report these error metrics shown below:

1. Accuracy

   (For the following 3 error metrics, you can think of 1 of the 2 classes as positive, and the other one as negative.)

2. Precision
3. Recall
4. F-value

4 (10 points) Logistic Regression (3)

For a logistic regression function with input $\boldsymbol{x} \in \mathbb{R}^m$ and output $y \in \{-1, +1\}$, the probability of $P(y = +1|\boldsymbol{x}) = \dfrac{1}{1 + e^{-(b+\boldsymbol{w}^T\boldsymbol{x})}}$.

1. Please show that $P(y|\boldsymbol{x}) = \dfrac{1}{1 + e^{-y(b+\boldsymbol{w}^T\boldsymbol{x})}}$
2. Please analyze the decision boundary for this logistic regression classifier. (On what condition of $\boldsymbol{x}$, y will be predicated as $+1$ or -1?)