01.112 MACHINE LEARNING (2017)
HOMEWORK 1

SHAOWEI LIN

The problems which require your solutions will be marked (a), (b), (c), ..., at the end of every section. Each problem is worth one point. This assignment has a total of 6 points.

1. Linear Algebra and Probability Review

(a) Let $\theta \in \mathbb{R}^d$ be a vector and $\theta_0 \in \mathbb{R}$ be a constant. Let $x = [x_1, \ldots, x_d]$ be a vector of unknowns. Consider the hyperplane in $\mathbb{R}^d$ whose equation is given by $\theta \cdot x + \theta_0 = 0$. Given a point $y \in \mathbb{R}^d$, find its distance to the hyperplane.

(b) Let X and Y be independent Poisson random variables, i.e.

$$\mathbb{P}(X = x) = \frac{\alpha^x e^{-\alpha}}{x!}, \quad \mathbb{P}(Y = y) = \frac{\beta^y e^{-\beta}}{y!}, \quad \text{for all } x, y \geq 0.$$

for some rates $\alpha, \beta > 0$. Show that $Z = X + Y$ is also Poisson, and find its rate γ.

2. Python and Theano

We will be doing our Python programming in Jupyter Notebook, which provides a great environment for scientific computing, and for documenting machine learning experiments. To install Jupyter Notebook, please check the course information for instructions. First, create a Python notebook in Jupyter, and run the following code to check the version of Python you are running. Make sure that it is 3.5 or greater.

```
import sys
print (sys.version)
```

We will now install Theano,

`http://deeplearning.net/software/theano/`,

a Python library that allows you to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently. For Anaconda users, you may want to ensure that Anaconda is up-to-date by running the following at your command prompt. In Windows, you will need to run Command Prompt as an Administrator.

Date: September 17, 2017.

```
conda update conda
conda update anaconda
```

To install Theano, run the following at your command prompt.

```
conda install theano pygpu
```

The website below lists some other ways to install Theano and GCC (optional). Please note that while GCC and CUDA will speed up Theano significantly, getting all the dependencies to work in Windows could be challenging. If you want to use distributed machine learning in future, do consider installing Ubuntu on a separate partion of your hard disk or running Docker environments to install a full version of Theano on your Windows machine.

`http://deeplearning.net/software/theano/install.html`

Incidentally, to change the working directory for Jupyter Notebook, please read

`https://stackoverflow.com/questions/35254852`.

Note that for Windows, you have to specify the address using the following format.

```
c.NotebookApp.notebook_dir = u'C:\\Users\\TanAhBeng\\Dropbox'
```

To find the Jupyter install directory, you may have to search for the Anaconda directory on your hard disk. After you have created the config file, its location will be output to the screen. The file may reside in a directory other than your Jupyter install directory. To edit the file in Windows, you could use Notepad. Remember to also change the ”Target” and ”Start in” properties of the Jupyter Notebook shortcut in your Windows Start Menu.

Remark. I would have preferred to use TensorFlow instead of Theano, but installing it for Windows is tricky. The programming style in Theano is similar to TensorFlow.

(a) Write down the version of Python and the version of Theano you are using.

3. Linear Regression

Theano is a powerful Python library that is well-suited for optimization problems, especially the ones that we see in machine learning. It contains a symbolic engine that computes the gradients of many kinds of objective functions.

In this problem, we will apply gradient descent using Theano to perform linear regression on the following data set.

`https://www.dropbox.com/s/oqoyy9p849ewzt2/linear.csv?dl=1`.

The data set contains a 50×5 matrix of real numbers where the first column is the response Y and the remaining four columns form the feature matrix X. There are fifty samples in the training data, and the last column of X is a column of ones. First, we import the data.

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
csv = 'https://www.dropbox.com/s/oqoyy9p849ewzt2/linear.csv?dl=1'
data = np.genfromtxt(csv,delimiter=',')
X = data[:,1:]
Y = data[:,0]
```

Next, we import Theano and note down some basic information. Do not worry if you get a warning that `g++` is not detected. It is not needed for Theano to run.

```
import theano
import theano.tensor as T
d = X.shape[1]  # dimension of feature vectors
n = X.shape[0]  # number of training samples
learn_rate = 0.5  # learning rate for gradient descent
```

We now declare some symbolic variables in Theano. Here, the model parameter vector `w` is defined as a shared variable in Theano, which is a special kind of variable that keeps track of its state. We will be updating the state of `w` during the gradient descent.

```
x = T.matrix(name='x')  # feature matrix
y = T.vector(name='y')  # response vector
w = theano.shared(np.zeros((d,1)),name='w')  # model parameters
```

We will minimize the empirical risk to find the best parameter vector w. The good news is that Theano is able to help us in computing the gradient of the risk.

```
risk = T.sum((T.dot(x,w).T - y)**2)/2/n  # empirical risk
grad_risk = T.grad(risk, wrt=w)  # gradient of the risk
```

The next statement creates a function which computes one step of gradient descent and updates the state of the shared variable `w`. The training data X, Y are passed in as `givens`, and the function outputs the current value of `risk` each time it is called.

```
train_model = theano.function(inputs=[],
                          outputs=risk,
                          updates=[(w, w-learn_rate*grad_risk)],
                          givens={x:X, y:Y})
```

Finally, we run the gradient descent algorithm, and print the value of the risk after each iteration. At the end of the loop, we show the final state of the parameter vector `w`.

```
n_steps = 50
for i in range(n_steps):
    print(train_model())
print(w.get_value())
```

(a) Write down the vector `w` of coefficients computed by gradient descent.

(b) Write a program which calculates the exact solution for the optimal coefficients in the linear regression problem. Compare your solution with the answer in (a).

(c) Find a Python library that computes the optimal coefficients in the linear regression problem, and compare its solution with the answer in (a).

(d) **Bonus (0 points).** Write a program which computes the solution using stochastic gradient descent. You may use a minibatch size of 5 data points. For convergence, remember to decrease the learning rate over time.

HOMEWORK HINTS

Q1a. Find a vector φ that is orthogonal to the hyperplane. Compute the distance of the hyperplane from the origin. Project y onto φ, and compute the distance of this projection from the origin.

Q1b. For each integer $z \geq 0$, $\mathbb{P}(Z = z)$ is the sum of $\mathbb{P}(X = x)\mathbb{P}(Y = y)$ over all x, y such that $x + y = z$. Can you simplify the sum using the binomial theorem?