CS 273P: Machine Learning and Data Mining

Homework 2

Due date: **Tuesday 11:59 PM, Jan 30, 2024**

Instructor: Xiaohui Xie

The submission for this homework should be

a) `one stand-alone PDF file` submitted to Gradescope, containing all of the relevant code, figures, and any text explaining your results.

b) `one stand-alone zip file` submitted to Canvas, including all source codes.

When coding the answers, try to write functions to encapsulate and reuse the code, instead of copy pasting the same code multiple times. This will not only reduce your programming efforts, but also make it easier to debug, and for us to grade your work.

Write neatly (or type) and show all your work!

Points: This homework adds up to a total of **65 points**, as follows:

Problem 1: Linear Regression from Scratch	30 points
Problem 2: Getting familiar with PyTorch	30 points
Statement of Collaboration	5 points

Problem 1: Linear Regression from Scratch (30 points)

In this problem we will explore linear regression, gradient descent and regularization on a toy example. You'll be implementing your own linear regression model. Please fill in the corresponding parts in the provided jupyter notebook.

Problem 2: Getting familiar with PyTorch (30 points)

For this problem we will explore how to use PyTroch to solve the Linear Regression Problem, by using Gradient Descent Algorithm.

1. Load the "`data/curve80.txt`" data set, and split it into 75% / 25% training/test. We will use `degree=5` for all the polynomial features.

```
data = np.genfromtxt("data/curve80.txt")
X = data[:,0]
X = np.atleast_2d(X).T  # code expects shape (M,N) so make sure it's 2-dimensional
Y = data[:,1]           # doesn't matter for Y
Xtr,Xte,Ytr,Yte = ml.splitData(X,Y,0.75)  # split data set 75/25

degree = 5
XtrP = ml.transforms.fpoly(Xtr, degree=degree, bias=False)
XtrP,params = ml.transforms.rescale(XtrP)
```

   Transform numpy arrays to tensor. Make sure the `XtrP_tensor` has the shape of (60, 5) while `Ytr_tensor` has the shape of (60, 1). `(5 points)`

```
XtrP_tensor =  ... XtrP ...
Ytr_tensor = ... Ytr ...

XtrP_tensor = XtrP_tensor.float()
Ytr_tensor = Ytr_tensor.float()
```

2. Initialize our linear regressor. `(5 points)`

```
linear_regressor = torch.nn.Linear(in_features=..., out_features=...)
```

3. Set up the criterion and optimizer.

```
criterion = torch.nn.MSELoss()
optimizer = torch.optim.SGD(linear_regressor.parameters(), lr=0.1)
epochs = 100000
```

4. Training the regressor using gradient descent. `10 points`

```
loss_record = []
for _ in range(epochs):
    optimizer.some_function ...                # set gradient to zero
    pred_y = linear_regressor(XtrP_tensor)
    loss = criterion(pred_y, Ytr_tensor)       # calculate loss function
    ....                                       # backpropagate gradient
    loss_record.append(loss.item())
    optimizer.some_function ...                # update the parameters in the linear regressor
```

5. Plot the loss v.s. epochs. Show the plot here. `5 points.`

```
plt.plot(range(epochs), (loss_record))
```

6. Visualize the trained linear regressor. `5 points.`

```
xs = np.linspace(0,10,200)
xs = xs[:,np.newaxis]
xsP, _ = ml.transforms.rescale(ml.transforms.fpoly(xs,degree=degree,bias=False), params)
xsP_tensor = torch.from_numpy(xsP).float()
ys = linear_regressor(xsP_tensor)

plt.scatter(Xtr,Ytr,label="training data")
plt.plot(xs,ys.detach().numpy(),label="prediction function",color ='red')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
```

Statement of Collaboration (5 points)

It is **mandatory** to include a `Statement of Collaboration` in each submission, with respect to the guidelines below. Include the names of everyone involved in the discussions (especially in-person ones), and what was discussed.

All students are required to follow the academic honesty guidelines posted on the course website. For programming assignments, in particular, I encourage the students to organize (perhaps using Campuswire) to discuss the task descriptions, requirements, bugs in my code, and the relevant technical content `before` they start working on it. However, you should not discuss the specific solutions, and, as a guiding principle, you are not allowed to take anything written or drawn away from these discussions (i.e. no photographs of the blackboard, written notes, referring to Campuswire, etc.). Especially `after` you have started working on the assignment, try to restrict the discussion to Campuswire as much as possible, so that there is no doubt as to the extent of your collaboration.