In [1]:

```
import numpy as np
import edf
from time import time
import sys
%matplotlib inline
import matplotlib.pyplot as plt
```

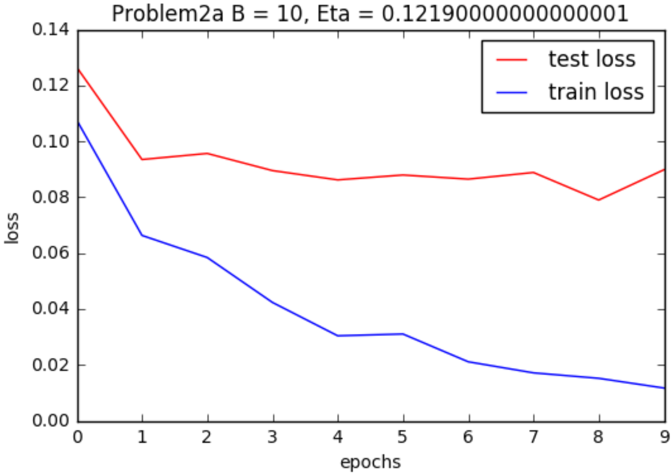
In [2]:

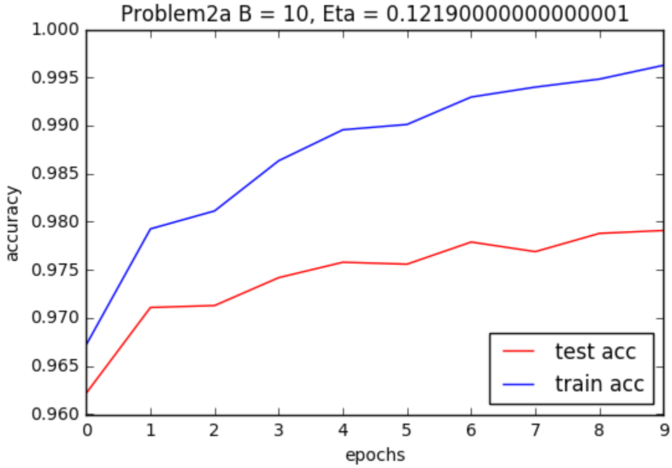
```
traindata = './mnist_data/train.npz'
valdata = './mnist_data/test.npz'

data = np.load(traindata)
t_imgs = np.float32(data['imgs'])/255.
t_labels = np.float32(data['labels'])

data = np.load(valdata)
v_imgs = np.float32(data['imgs'])/255.
v_labels = np.float32(data['labels'])
```

In [3]:

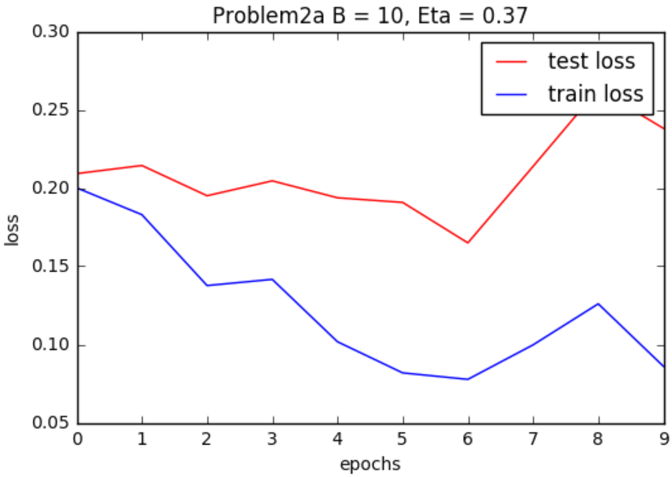
```
############################### B=10 eta=eta(B) ###########################################


############################################ please modify this cell to finish the problem 2.a ######################


# for repeatability
np.random.seed(0)

# batch size, please try 10, 50 and 100. For each run, you might need to reloading the kernel (edf.py)
# to clear the history information
batch_array = [10,100,50]
eta_array = [10 * 0.0056 + 0.0659, 0.37, 100 * 0.0056 + 0.0659, 0.37, 50 * 0.0056 + 0.0659, 0.37]
#batch = 10
# learning rate eta, measured by per-batch unit. If you change this batch size, you might also change eta
# accoridng to the equation given in the homework.
#eta = batch * 0.0056 + 0.0659


for i in range(0,3):
    batch = batch_array[i]
    for j in range(i*2,i*2+2):


        # Inputs and parameters
        inp = edf.Value()
        lab = edf.Value()

        W1 = edf.Param(edf.xavier((28*28,128)))
        B1 = edf.Param(np.zeros((128)))
        W2 = edf.Param(edf.xavier((128,10)))
        B2 = edf.Param(np.zeros((10)))

        # models
        hidden = edf.RELU(edf.Add(edf.VDot(inp,W1),B
1))
        pred = edf.SoftMax(edf.Add(edf.VDot(hidden,W
2),B2))
        loss = edf.LogLoss(edf.Aref(pred, lab))
        acc = edf.Accuracy(pred,lab)

        def eval(imgs, labels):
            batches = range(0, len(labels), batch)
            objective = 0
            accuracy = 0
            for k in batches:
                inp.set(imgs[k:k+batch])
                lab.set(labels[k:k+batch])
                edf.Forward()
                objective += np.mean(loss.value)
                accuracy += acc.value
            return accuracy/len(batches),
objective/len(batches)
        #print(i,j)
        eta = eta_array[j]
        accuracy, objective = eval(t_imgs, t_labels)
        print("Random accuracy = %.4f" % accuracy)

        train_loss = []
        train_acc = []
        test_loss = []
        test_acc = []
        ep = 0
        stime = time()
        epoch = 10
        batches = range(0, len(t_labels), batch)

        while ep < epoch:

            # randon shuffle the train data in each
 epoch
            perm = np.random.permutation(len(t_label
s))
            for k in batches:
                inp.set(t_imgs[perm[k:k+batch]])
                lab.set(t_labels[perm[k:k+batch]])
                edf.Forward()
                edf.Backward(loss)
                edf.SGD(eta)

            # evaluate on trainset
            t_acc, t_loss = eval(t_imgs, t_labels)
            print("Epoch %d: train loss = %.4f [%.3f secs]" % (ep, t_loss,time()-stime))
            train_loss.append(t_loss)
            train_acc.append(t_acc)

            # evaluate on testset
            v_acc, v_loss = eval(v_imgs, v_labels)
            print("test accuracy=%.4f" % v_acc)
            test_loss.append(v_loss)
            test_acc.append(v_acc)
            stime = time()
            ep += 1

        # plot
        plt.figure(1)
        plt.xlabel("epochs")
        plt.ylabel("loss")
        plt.plot(np.arange(len(test_loss)), test_loss, color='red')
        plt.plot(np.arange(len(train_loss)), train_loss, color='blue')
        plt.title("Problem2a B = {}, Eta = {}".format(batch,eta))
        plt.legend(['test loss', 'train loss'], loc='upper right')
        plt.show()

        plt.figure(2)
        plt.xlabel("epochs")
        plt.ylabel("accuracy")
        plt.plot(np.arange(len(test_acc)), test_acc, color='red')
        plt.plot(np.arange(len(train_acc)), train_acc, color='blue')
        plt.legend(['test acc', 'train acc'], loc='lower right')
        plt.title("Problem2a B = {}, Eta = {}".forma
t(batch,eta))
        plt.show()
```

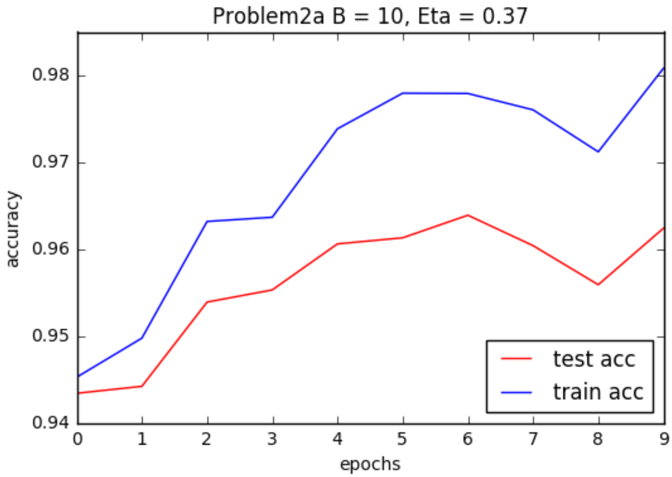
```
Random accuracy = 0.1329
Epoch 0: train loss = 0.1076 [16.849 sec
s]
test accuracy=0.9622
Epoch 1: train loss = 0.0663 [16.703 sec
s]
test accuracy=0.9711
Epoch 2: train loss = 0.0584 [16.238 sec
s]
test accuracy=0.9713
Epoch 3: train loss = 0.0423 [16.637 sec
s]
test accuracy=0.9742
Epoch 4: train loss = 0.0304 [16.803 sec
s]
test accuracy=0.9758
Epoch 5: train loss = 0.0310 [16.557 sec
s]
test accuracy=0.9756
Epoch 6: train loss = 0.0211 [16.654 sec
s]
test accuracy=0.9779
Epoch 7: train loss = 0.0171 [16.684 sec
s]
test accuracy=0.9769
Epoch 8: train loss = 0.0152 [15.355 sec
s]
test accuracy=0.9788
Epoch 9: train loss = 0.0117 [16.502 sec
s]
test accuracy=0.9791
```


Problem2a B = 10, Eta = 0.12190000000000001
loss
epochs
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
1
2
3
4
5
6
7
8
9
test loss
train loss

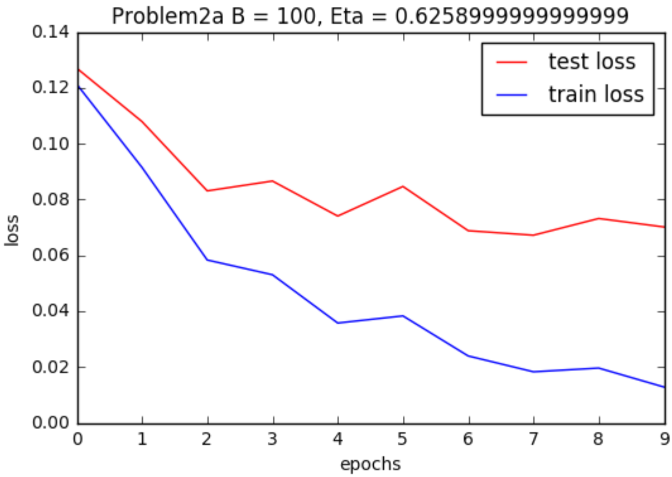

Problem2a B = 10, Eta = 0.12190000000000001
accuracy
epochs
1.000
0.995
0.990
0.985
0.980
0.975
0.970
0.965
0.960
0
1
2
3
4
5
6
7
8
9
test acc
train acc

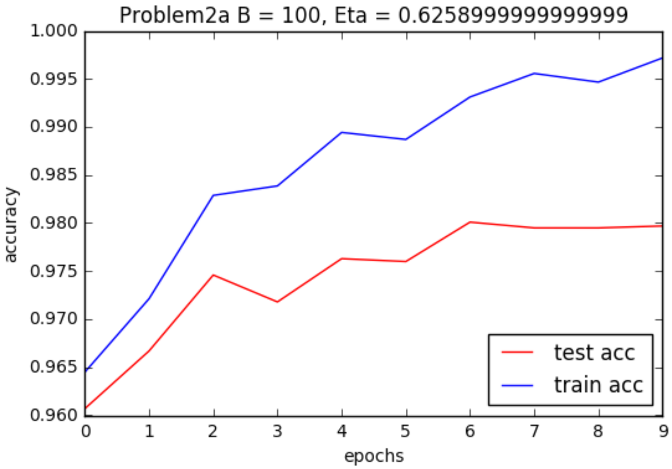
```
Random accuracy = 0.1498
Epoch 0: train loss = 0.2001 [32.488 sec
s]
test accuracy=0.9434
Epoch 1: train loss = 0.1830 [31.560 sec
s]
test accuracy=0.9442
Epoch 2: train loss = 0.1377 [31.651 sec
s]
test accuracy=0.9539
Epoch 3: train loss = 0.1416 [32.373 sec
s]
test accuracy=0.9553
Epoch 4: train loss = 0.1018 [32.319 sec
s]
test accuracy=0.9606
Epoch 5: train loss = 0.0819 [31.752 sec
s]
test accuracy=0.9613
Epoch 6: train loss = 0.0777 [31.687 sec
s]
test accuracy=0.9639
Epoch 7: train loss = 0.0998 [33.052 sec
s]
test accuracy=0.9604
Epoch 8: train loss = 0.1260 [31.836 sec
s]
test accuracy=0.9559
Epoch 9: train loss = 0.0860 [32.783 sec
s]
test accuracy=0.9624
```


Problem2a B = 10, Eta = 0.37

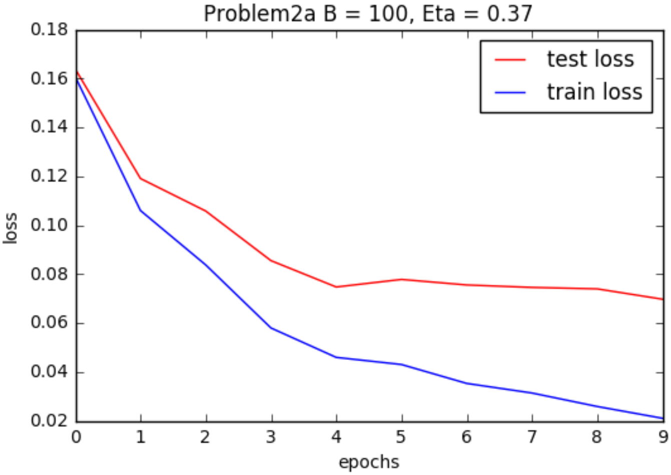

Problem2a B = 10, Eta = 0.37

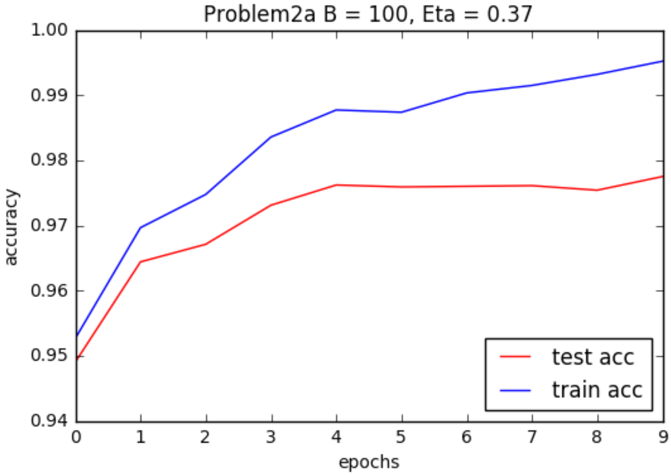
```
Random accuracy = 0.1110
Epoch 0: train loss = 0.1213 [9.023 secs]
test accuracy=0.9607
Epoch 1: train loss = 0.0915 [8.543 secs]
test accuracy=0.9667
Epoch 2: train loss = 0.0583 [9.110 secs]
test accuracy=0.9746
Epoch 3: train loss = 0.0530 [9.043 secs]
test accuracy=0.9718
Epoch 4: train loss = 0.0357 [9.297 secs]
test accuracy=0.9763
Epoch 5: train loss = 0.0383 [9.198 secs]
test accuracy=0.9760
Epoch 6: train loss = 0.0239 [8.947 secs]
test accuracy=0.9801
Epoch 7: train loss = 0.0183 [9.314 secs]
test accuracy=0.9795
Epoch 8: train loss = 0.0196 [9.305 secs]
test accuracy=0.9795
Epoch 9: train loss = 0.0128 [9.094 secs]
test accuracy=0.9797
```



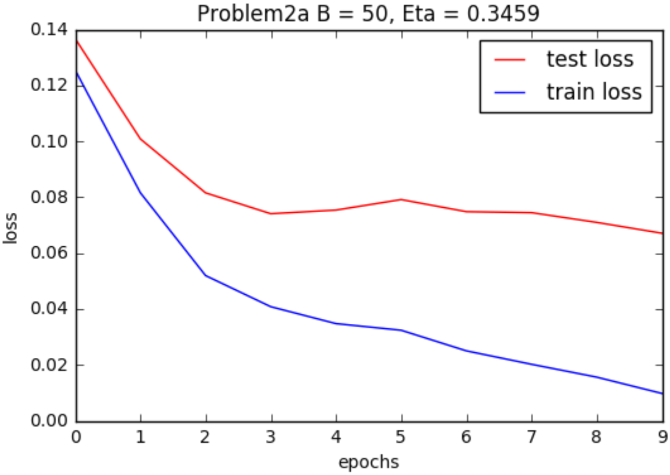

Problem2a B = 100, Eta = 0.6258999999999999
accuracy
1.000
0.995
0.990
0.985
0.980
0.975
0.970
0.965
0.960
0
1
2
3
4
5
6
7
8
9
epochs
test acc
train acc

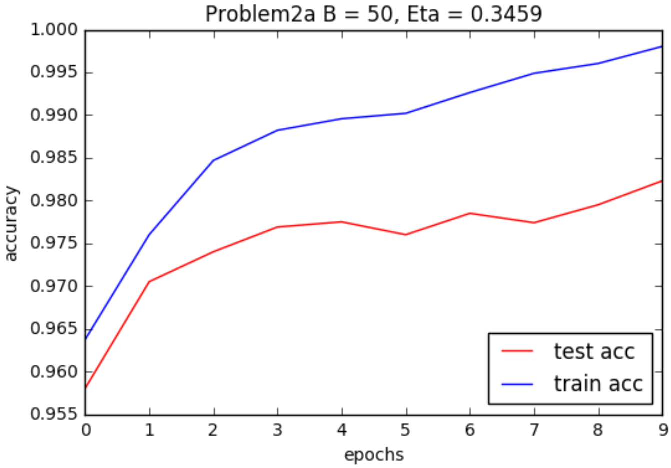
```
Random accuracy = 0.1019
Epoch 0: train loss = 0.1607 [14.574 sec
s]
test accuracy=0.9490
Epoch 1: train loss = 0.1060 [14.408 sec
s]
test accuracy=0.9644
Epoch 2: train loss = 0.0837 [14.101 sec
s]
test accuracy=0.9671
Epoch 3: train loss = 0.0579 [14.699 sec
s]
test accuracy=0.9731
Epoch 4: train loss = 0.0459 [14.572 sec
s]
test accuracy=0.9762
Epoch 5: train loss = 0.0430 [14.298 sec
s]
test accuracy=0.9759
Epoch 6: train loss = 0.0353 [14.525 sec
s]
test accuracy=0.9760
Epoch 7: train loss = 0.0313 [14.479 sec
s]
test accuracy=0.9761
Epoch 8: train loss = 0.0258 [14.528 sec
s]
test accuracy=0.9754
Epoch 9: train loss = 0.0210 [15.039 sec
s]
test accuracy=0.9775
```


Problem2a B = 100, Eta = 0.37
loss
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0
1
2
3
4
5
6
7
8
9
epochs
test loss
train loss

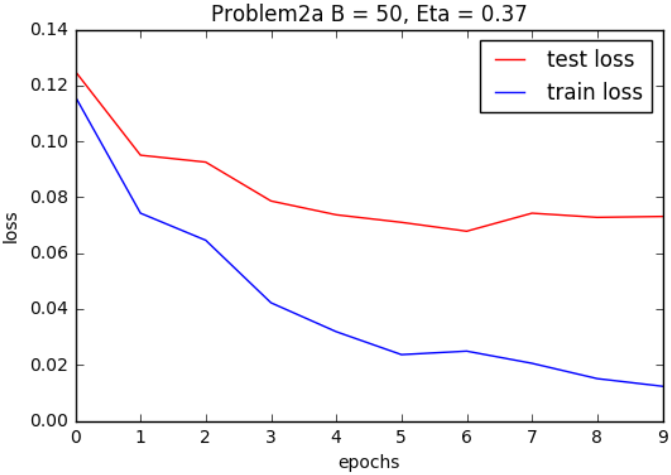

Problem2a B = 100, Eta = 0.37
accuracy
1.00
0.99
0.98
0.97
0.96
0.95
0.94
0
1
2
3
4
5
6
7
8
9
epochs
test acc
train acc

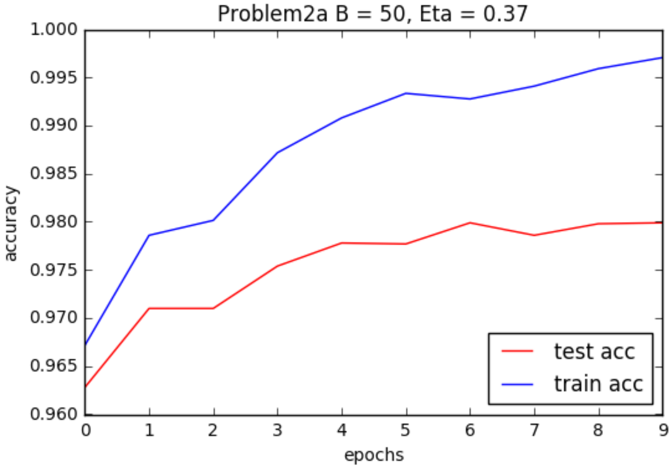
```
Random accuracy = 0.1153
Epoch 0: train loss = 0.1255 [35.189 sec
s]
test accuracy=0.9580
Epoch 1: train loss = 0.0816 [35.721 sec
s]
test accuracy=0.9705
Epoch 2: train loss = 0.0519 [35.576 sec
s]
test accuracy=0.9740
Epoch 3: train loss = 0.0408 [35.059 sec
s]
test accuracy=0.9769
Epoch 4: train loss = 0.0348 [35.489 sec
s]
test accuracy=0.9775
Epoch 5: train loss = 0.0324 [35.186 sec
s]
test accuracy=0.9760
Epoch 6: train loss = 0.0250 [31.097 sec
s]
test accuracy=0.9785
Epoch 7: train loss = 0.0202 [32.529 sec
s]
test accuracy=0.9774
Epoch 8: train loss = 0.0156 [30.174 sec
s]
test accuracy=0.9795
Epoch 9: train loss = 0.0098 [30.507 sec
s]
test accuracy=0.9823
```


Problem2a B = 50, Eta = 0.3459
test loss
train loss
loss
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
1
2
3
4
5
6
7
8
9
epochs


Problem2a B = 50, Eta = 0.3459
accuracy
1.000
0.995
0.990
0.985
0.980
0.975
0.970
0.965
0.960
0.955
test acc
train acc
0
1
2
3
4
5
6
7
8
9
epochs

```
Random accuracy = 0.1734
Epoch 0: train loss = 0.1160 [36.906 sec
s]
test accuracy=0.9628
Epoch 1: train loss = 0.0743 [35.551 sec
s]
test accuracy=0.9710
Epoch 2: train loss = 0.0646 [33.493 sec
s]
test accuracy=0.9710
Epoch 3: train loss = 0.0422 [34.386 sec
s]
test accuracy=0.9754
Epoch 4: train loss = 0.0319 [33.039 sec
s]
test accuracy=0.9778
Epoch 5: train loss = 0.0237 [33.223 sec
s]
test accuracy=0.9777
Epoch 6: train loss = 0.0249 [32.905 sec
s]
test accuracy=0.9799
Epoch 7: train loss = 0.0206 [34.677 sec
s]
test accuracy=0.9786
Epoch 8: train loss = 0.0151 [33.591 sec
s]
test accuracy=0.9798
Epoch 9: train loss = 0.0123 [35.684 sec
s]
test accuracy=0.9799
```


Problem2a B = 50, Eta = 0.37
test loss
train loss
loss
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
1
2
3
4
5
6
7
8
9
epochs


Problem2a B = 50, Eta = 0.37
test acc
train acc
accuracy
1.000
0.995
0.990
0.985
0.980
0.975
0.970
0.965
0.960
0
1
2
3
4
5
6
7
8
9
epochs

```
In [ ]:
```