homework HW1 Math6373 due date thursday feb 10th at midnight

Data set:
use four years of daily market data; download 7 daily closing prices of
Gold, Platinum, Silver, DowJones, Euro, Yen , Renminbi,
Note: Renminbi = Yuan
On day “t”: V(t) = line vector of 7 prices = [V1(t) ... V7(t)]
V1(t) = price of Gold on day t =Gold(t) ... V2(t) = price of platinum on day t =Platinum(t)
the four years data set contains N actual days
replace calendar dates by index t=1,2,3 ... N
N= total number of days in whole data set
X(t) = feature vector has dimension 5x7 =35
X(t) = long line vector [V(t), V(t-1), V(t-2), V(t-3), V(t-4)]
case # t is INITIALLY described by feature vector Xt

Goal: construct an MLP to predict (on day t) the future gold price Z(t) = V1(t+1) = Gold(t+1)
data set = { X(1) , X(2) , ... X(N) } cases observed over 4 years
true value Z(t) is known on the data set for all t <= N-1

Q0
for each j = 1... 7 compute Mj= mean over all t of the values Vj(t)
Mj = (1/N)*(Vj(1) + ... + Vj(N)
for j=1 .. 6 construct the graph displaying both Vj(t)/Mj and V7(t)/M7(t)
Visual interpretation?

Q1
replace each price Vj(t) by rate of return rVj(t) = [Vj(t) - Vj(t-1)] / Vj(t-1)]
replace Z(t) by rZ(t) = [Z(t) - Z(t-1)] / [Z(t-1) = [V1(t+1) - V1(t)] / V1(t)]
replace X(t) by rX(t) = [rV(t), rV(t-1), rV(t-2), rV(t-3), rV(t-4)]
for case # t, the **new feature vector is rX(t),**
the **true target variable to be predicted is rZ(t)**
compute mean.rZ = average of the N absolute values rZ(t) , namely
mean.rZ = (1/N) *(| rZ(1) | + ... + | rZ(N) |)
on one single graph display the 7 curves rV1(t) ... rV7(t)
Visual interpretation?
explain how a good prediction of rZ(t) on day t will easily provide
a good prediction on day t for Z(t) = Gold(t+1)

Q2
define the first attempted architecture of your MLP with 3 layers as follows
Input layer L1➔ hidden layer L2 ➔ Output layer L3
size L1 = 35 ; size L2 = h ; size L3 = 1;
The integer h will be finalized below
denote param(h) the total # of weights and thresholds in this MLP

give a formula for param(h)

Q3

randomly select 80% of all cases as your training set; display TRN = size of training set
the remaining 20% of cases will be the test set
apply the parsimony principle : impose param(h) < # informations brought by the training set
compute the maximum value h* of h , derived from this parsimony principle

Q4

fix 2 possible values for h namely h1= h* and h2 = 3 h*
note that h2 does not verify the parsimony principle
for each such value of h, launch the automatic learning of your MLP
you will need to select (and report your choices)
the type of response function(RELU is suggested)
the type of initialization of the weights and thresholds (default random choices in tensorflow}
the type of gradient descent optimizer (Adams is a good generic choice)
the Batch Size BATS (try 4 possible BATS values : TRN/40, TRN/20, TRN/10, TRN/2)
the type of loss function (MSE)
the **total number of epochs TOTEP** (suggestion : at least 100 or 150 epochs)
the criterion used to stop the automatic learning (explain the basic choices in tensorflow)

for each of the 8 choices of the pair (h , BATS)

display the computing time necessary for automatic learning
display the total number numBATS of batches
display the terminal value trainMSE of MSE on the whole training set
display the the curve MSE(m)

give a comparative interpretation of these results

Q5

Monitoring of EACH one of of the eight automatic learning
for m =1,2, ...,TOTEP after each epoch # m
compute trainMSE(m) on the whole training set and testMSE(m) on the whole test set
display the two curves trainMSE(m) and testMSE(m) versus m
compute and display the curve ||grad MSE(m) || / sqrt(param(h))

Q6

for each one of the eight automatic learning and for each epoch # **m =1,2, ...,TOTEP**
compute the two normalized accuracy curves
trainAcc(m) = sqrt(trainMSE(m))/ mean.rZ
testAcc(m) = sqrt(testMSE(m)) / mean.rZ
recall that mean.rZ is the mean of all absolute vales |rZ(t)|
on the same graph display the two curves { trainAcc(m) and testAcc(m) versus m}
interpret these results for each automatic learning ;
check if and when there is overfit;

comment the behaviour of the || gradMSE(m) || versus m
for each learning, determine an **optimal stopping epoch index m*** and the corresponding optimal values obtained for trainAcc(m*) , testAcc(m*)
you shoul try to find m* minimizing testAcc(m) but also
make sure that there is no overfit at m*

Q7
use your preceding analyzis to determine the best pair (h , BATS), and the corresponding best weights Wij + thresholds Bi reached at optimal stopping epoch m*
display the histogram of all |weights|= |Wij| linking neuron j of L1 to neuron i of L2
identify the 10 smallest and the 10 largest |Wij|
display the histogram of all |weights| = |m(i,1)| linking neuron i of L2 to neuron 1 of L3
rank the |m(i,1)| in increasing order and display this increasing curve

Q8
Most Influential Hidden Neurons
identify the neuron i*in L2 such that |m(i*,1)| > all |m(i,1)|
this neuron is strongly influential on the output

Q9
most influential explanatory variables
the neuron i* is connected to 35 inputs by weights W(i*,1) ...W(i*, 35)
for each neuron j in L1, compute average impact of input(j) on neuron i* by
impact(j on i*)) = | W(i*,j) | x mean| input(j) | where
mean | input(j) | = average value of | input(j) | over all cases
rank the impacts(j,i*) in increasing order and display these 35 ordered values
identify the 2 explanatory variables which have the highest influence on neuron i*
identify the 2 explanatory variables which have the lowest influence on neuron i*
conclusions?

Q10
your suggestions to improve the architecture of the MLP for better testAcc, trinAcc?
try at least one of your suggestions

1. Email 1

answer to questions asked by Thuy Le

1. What is the target Y for our prediction? Is it the rate of return of the Gold?

RA answer 1

the ultimate goal on day t is to predict the next day price of gold G(t+1)

but for the MLP output computed from the input data available on day t should be an estimate of the yet unknown rate of return for gold, namely

rG(t+1) = (G(t+1) -G(t)) / G(t)

obviously a good estimate of rG(t+1) will immediately

produce a good estimate of G(t+1)

2. about Question 5: The m in trainMSE(m) represents the index of the epoch?

For example: if we have 100 epochs, so the trainMSE(m) are\

[trainMSE(1) trainMSE(2) ..... trainMSE(100)]

RA answer 2

yes the m in trainMSE(m) represents the index of the epoch

each MSE(m) is computed at the end of epoch #m, on the whole training set

3. about HW1 Question 6: The trainAcc(k), should the k be the m (# of epoch)?

and the mean.rZ should be calculated from absolute values of the rate of return of Gold?

RA answer3

the k in trainAcc(k), testAcc(k) should be replaced by m = index of epoch #m

this was a typo

the sqrt(trainMSE(m)) and the sqrt(testMSE(m))should both be normalized by the average value of |rG(t)| over all days t

note that this is an average of the absolute values |rG(t)| of the rG(t)

On Mon, Feb 7, 2022 at 3:04 PM Thuy Le <thuthuy230193@gmail.com> wrote:

Good morning Professor Azencott,

We have some questions about HW1

1. What is the target Y for our prediction? Is it the rate of return of the Gold?

2. Question 5: The m in trainMSE(m) represents the index of the epoch?

For example: if we have 100 epochs, so the trainMSE(m) = [trainMSE(1)   trainMSE(2) ..... trainMSE(100)]

3. Question 6: The trainAcc(k), should the k should be m (# of epoch)?

and the mean.rZ should be calculated from absolute values of the ror of Gold?

Thank you,

--

Thuy Le

High School Math Teacher | Bachelor Degree in Mathematics| UNIVERSITY of HOUSTON

thuthuy230193@gmail.com | (832) 998 – 0179

2. Email 2

questions from Chia-Hung Chien

about Q5

I assume that k means the learning steps in each epoch (which is related to the batch size) and m means the epoch number. For example, I have 800 training data point and set batch size = 40 and # epochs = 10. I will have 10 trainMSE(m) where m = 1, 2, 3, …, 10.

Each trainMSE represents the training set MSE at the end of each epoch in each automatic learning..

RA answer:

k is the index of batch #k ; in the context you mention (batch size =40 , training set size =800, then k= 1 2 3... 20 because 20 = 800 /40, and each epoch will contain 20 batches. You have chosen 10 for the nuber of epochs: this is not large enough; i suggest a first try

with at least 100 epochs, and you may need many more epochs ;

about Q6, since each epoch will have 20 steps of learning under my setting, I will have 20*10 = 200 values trainMSE(k) in each automatic learning. However, the number of steps will be changed if we change the batch size which is required in the homework. Thus, each learning will have different length. Or do I just select the last epoch so that I will have trainMSE(k) with length of 20.

RA answer: if you choose # epochs= 100 and number of batches per epoch = 20 ; you will get a total of 2000 batches , and hence 2000 values for [batch #k trainMSE] indexed

by k=1 2 3 ... 2000; each [batch #k trainMSE] is computed only on batch#k, and hence uses only the 40 cases of batch # k. At the end of each epoch #m , with m= 1 2 3 ... 100, you should definitely compute trainMSE(m) on the whole fixed training set of 800 training cases as well as testMSE(m) on the whole fixed test set of 200 test cases

When you change the batch size to 10 for instance, the number of batches per epoch becomes equal to 800/10 = 80; if you keep the same number of epochs =100 , you will have a total of 8000 batches and hence 8000 values for [batch #k trainMSE]; however the number of values of trainMSE(m)and testMSE(m) will remain equal to 100, but their values will be different ;

to evaluate performance use essentially the normalized curves trainRMSE(m) and tesRMSE(m) as explained in the lectures ; you CANNOT make a judgment base on ONLY THE LAST EPOCH

about Q7 and Q8 : The weights that link neuron i of L2 to neuron 1 of L3 are denoted m(k,1). I

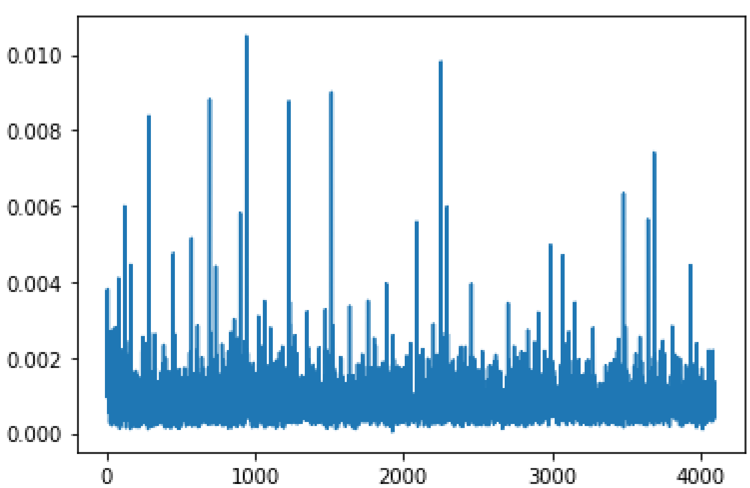
believe that this k has nothing to do with the k in Q5 and Q6.

RA answer : you are right, there is a typo in the text ! The weights linking neuron "i" of L2 to

neuron 1 of L3 SHOULD BE denoted m(i,1) for i = 1 2 3 ... h

3. Email 3

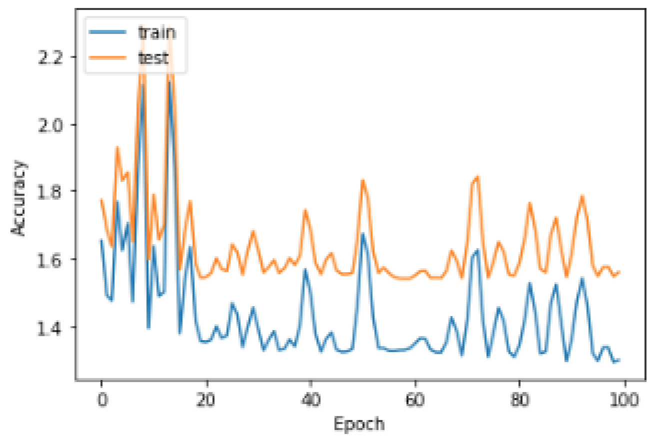
4. your ||gradcurve|| seems to have extremely low values practically from the beginning
5. and there is no stable improvement ; it looks like learning has no impact
6. this is rather unusual
7. repeat learning for same example with several different random intializations of the weights and thresholds
8. repeat learning with a different training sets
9. display also the behaviour of MSE batch by batch next time you send me similar gradient displays
10.
11. for question 6 please USE LAST VERSION OF HW1 SENT YESTERDAY
12. the quotation of Q6 you sent me is from the *old version* of HW1
13.
14. for the picture you extracted from the lecture notes the vertical coordinates represent trainMSE(m) in blue and testMSE(m) in yellow for each epoch "m"
15.
16.
17.
18. On Mon, Feb 7, 2022 at 4:58 PM Brian Le <le.bri000@gmail.com> wrote:
19. Hello Dr.Azencott,
20.
21. I just wanted to ask you about my group's gradient plots or curves after you had presented them in class today. I just wanted to present two example models that I have made. You had said the goal was for the gradient to be zero, which both of my curves do as training goes on. If you can check them out to see is that how they would perform that would be helpful.
22.
23. Below is the gradient curve for h =21 and batch size of 20. There are a total of 4000 batches total when the gradient of the MSE is found. Is this supposed to be what you expected as a result? Most of the MSE output is similar to the one shown below.
24.



25.
26. I also had a question regarding Q6 with plotting both the train and test MSE.

Q6
compute the two normalized accuracy curves
trainAcc(k) = sqrt(trainMSE(k))/ mean.rZ

27. testAcc(k) = sqrt(testMSE(k)) / mean.rZ
28. I believe that this output of the Accuracy Curves is correct in the format that you have given to us. Although, I do not understand directly the y-axis and the representation of the accuracy in this context. Can you give me context to an explanation of these values that are presented in the y-axis?
29.



30.
31.
32. Thanks,
33. Brian Le

4. Email 4

nothing wrong with the formula for trainAcc testAcc

but the names trainAcc testAcc   may be misleading !

you still want to minimize trainAcc and test Acc in spite of their names

On Tue, Feb 8, 2022 at 8:38 PM Sophie zou <sophiemz2016@gmail.com> wrote:

Hi Dr.Azencott,

I have a question about Q6.

Q6

compute the two normalized accuracy curves

trainAcc(k) = sqrt(trainMSE(k))/ mean.rZ

testAcc(k) = sqrt(testMSE(k)) / mean.rZ

With this formula, if we get a smaller MSE, we will get a lower Acc. But in fact, a smaller MSE should represent a better model with a higher Acc, it’s actually contradictory. Is there something wrong with the formula？

Thanks,

Man Zou

5. Email 5

you are using the old version of the text for HW1

please use the last version emailed last week to the class, which has many explanatory inserts highlighted in yellow

you ask why not use a graph in Q7

it is ok to use a graph but only if you reorder the neurons so that the graph becomes increasing

On Wed, Feb 9, 2022 at 12:37 AM Thuy Le <thuthuy230193@gmail.com> wrote:

Good morning Professor Azencott,

I have some questions for HW1

1. Question 1: "display separately the two curves". Is this a typo?

2. Question 7: 'display the histogram of all |weights|= |Wij| linking neuron j of L1 to neuron i of L2". Why do we choose histogram here? Is graph plotting better?

--

Thuy Le

High School Math Teacher | Bachelor Degree in Mathematics| UNIVERSITY of HOUSTON

thuthuy230193@gmail.com | (832) 998 – 0179

6. Email 6

Q0 in HW1

you should construct 6 graphs

each graph containing two normalized curved Vj(t)/Mj and V7(t)/M7

the goal is to visually understand the links between each one of the first 6 variables and the Renminbi which could (maybe) have a strong impact given the size of chinese economy

you should also add one single synthetic graph with all 7 curves

with a graphic emphasis on gold (target variable) to visually compare the possible impact of each explanatory variable