CS 165B – Machine Learning, Spring 2023

Assignment #4
Due Thursday, June 08 by 11:59pm PST

Notes:

- *This assignment is to be done individually. You may discuss the problems at a general level with others in the class (e.g., about the concepts underlying the question, or what lecture or reading material may be relevant), but the work you turn in must be solely your own.*
- Zoey Song zhenqiao@ucsb.edu is the head TA for this HW. Please contact her for questions.
- Be sure to re-read the "Policy on Academic Integrity" on the course syllabus.
- Be aware of the late policy in the course syllabus – i.e., *late submissions will be accepted but 20% deduction per day*.
- Justify every answer you give – **show the work** that achieves the answer or **explain** your response.
- Any updates or corrections will be posted on Piazza, so check there occasionally.
- **All assignments must be clear and legible.** It is recommended to type the solutions on this Microsoft Word file directly and save it as PDF. If you'll be submitting a handwritten assignment, please ensure that it's readable and neat. If your writing is not easily readable, your solution is not easy to follow on the page, or your PDF is not of very high quality, your assignment will not be graded. **DO NOT submit picture of your written work.** (If you must scan your written work in, use a high-quality scanner. Plan in advance.)
- To turn in your assignment:
  - Submit a PDF version to Gradescope

Problem 1 [10 points]

Given 4 data points in 3-d space, (1, 1, 0), (2, 2, 0), (0, -1, -1), (0, -2, -2).

(1) [2pts] Please explain what PCA can be used for and give a geometrical interpretation of first principal component of PCA.

(2) [4pts] Please show how to compute the first principal component given the data points. You only need to show how to compute it and do not need to give the exact number of the first principal component.

(3) [4pts] Please show how to compute the projected values for the data points using the first principal component. You can use w=(w1,w2,w3) as the first principal component.

(1) PCA can be used for dimensionality reduction and find the intrinsic linear structure in the data. The first principal component is the direction of maximum variance in the data.

(2) Let M=[[1/4,5/4,-3/4,-3/4];[1,2,-1,-2];[3/4,3/4,-1/4,-5/4]] be the zero mean 3x4 matrix. S=M*M^T be the 3x3 covariance matrix. Then we compute the eigenvectors of S, and the eigenvectors with largest eigenvalue is the first principal component.

(3) Let the data x = (x1, x2, x3), then the projected value is (x1-3/4)*w1 + (x2)*w2 + (x3+3/4)*w3. Note that you need to first minus the main of feature to be centralized.

Problem 2 [10 points]

There are four data points: x1 = (0, 2), x2 = (1, 1), x3 = (3, 2), x4 = (4, 0), and you are given two initial cluster centers: m1 = (0, 0), m2 = (3, 1). Please simulate one iteration of K-means clustering. In each iteration: (1) Assign data points to a cluster center. (2) Update the cluster centers.

(1) For each data point, compute the distance between it and the cluster center, and choose the closest cluster center.

$$x_1(2,\sqrt{10}), x_2(\sqrt{2},2), x_3(\sqrt{13},1), x_4(4,\sqrt{2})$$

x1 and x2 will be clustered to m1, and x3 and x4 will be clustered to m2.

(2) The new cluster centers are:

$$m_1' = (0.5,1.5), m_2' = (3.5,1)$$

Problem 3 [10 points]

Suppose you have trained three classifiers, each of which returns either 1 or −1, and their accuracies are listed as below:

Classifier	Accuracy
C_1	0.7
C_2	0.6
C_3	0.8

Let C be the classifier that returns a majority vote of the three classifiers. Assuming the errors of the C_i are independent, what is the probability that C(x) will be correct on a new test example x?

There are four cases that C will make correct prediction:
C1 and C2 are correct: 0.7*0.6*(1-0.8)
C1 and C3 are correct: 0.7*0.8*(1-0.6)
C2 and C3 are correct: 0.6*0.8*(1-0.7)
C1, C2 and C3 are correct: 0.7*0.6*0.8

Sum the above probabilities together: 0.788

Problem 4 [10points]

Suppose there are 12 samples in total, among which 9 samples are correctly classified and 3. samples are incorrectly classified The initial weights for each sample is 1/12. What is the updated weight for correctly classified samples and misclassified samples for the Boosting algorithm?

The error rate $\epsilon = \frac{3}{12} = 0.25$

Misclassified point: $w' = \frac{w}{2\epsilon} = \frac{1/12}{2\times 0.25} = 1/6$

Correctly classified point: $w' = \frac{w}{2(1-\epsilon)} = \frac{1/12}{2\times(1-0.25)} = 1/18$

Problem 5 [10 points]
Suppose you have run Adaboost on a training set for three boosting iterations. The results are classifiers h1, h2, and h3, with coefficients α1 = 0.3, α2 = 0.1, and α3 = 0.6. You find that the classifiers results on a test example x are h1(x) = 1, h2(x) = 1, and h3(x) = −1, What is the class returned by the Adaboost ensemble classifier H on test example x?

The combined score is: 0.3*1+0.1*1+0.6*(-1) = -0.2 < 0, thus the ensemble classifier will return -1 for the example x.

Problem 6 [10 points]
Let's consider a simple two-layer neural network (MLP). It has input size 2, one hidden layer size 3, and output size 1.

The input x=[$\begin{matrix}-2\\3\end{matrix}$], y = 1

Two sets of weights W1= $\begin{bmatrix}-1.6 & 0.8\\0.3 & 0.6\\1.6 & -0.2\end{bmatrix}$, W2=[0.2 0.8 -0.5]

Biases b1= $\begin{bmatrix}0.3\\0.5\\-0.7\end{bmatrix}$, b2=0.6

All the non-linear activation function is sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$, and loss function:

$$\mathcal{L} = -y \log \hat{y} - (1-y)\log(1-\hat{y}). \text{ (hint: } \sigma'(x) = \sigma(x) * (1-\sigma(x)))$$

For forward propagation, (base is e).

(a) Calculate z1=W1x+b1
Z1=[5.9 1.7 -4.5] ^T

(b) Calculate h1= $\sigma(z1)$
h1=[0.997 0.845 0.010]^T

(c) Calculate z2=W2h1+b2
Z2=1.470

(d) Calculate $\hat{y} = \sigma(z2)$
0.813

Problem 7 [10 points]
Also for the neural network in Problem 6. Now consider backwardpropagation.

(a) Calculate $\partial L/\partial \hat{y}$
-1.229

(b) Calculate $\partial \hat{y}/\partial z2$
0.1519

(c) Calculate $\partial z2/\partial W2$
[0.997 0.845 0.010]

(d) Calculate $\partial L/\partial W2$
[-0.186 -0.158 -0.002]

Problem 8 [10 points]

(1) Which assumption is word2vec based on?
(2) Give the math formulation of skip-gram and CBOW.
(3) What is the advantage of negative sampling in skip-gram?
(4) Suppose the semantic space is well learned, and we know the embedding of king, queen, man. How can we calculate the vector of woman?

Ans:

(1) Hypothesis assumption

(2) Skip-gram

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$$

CBOW

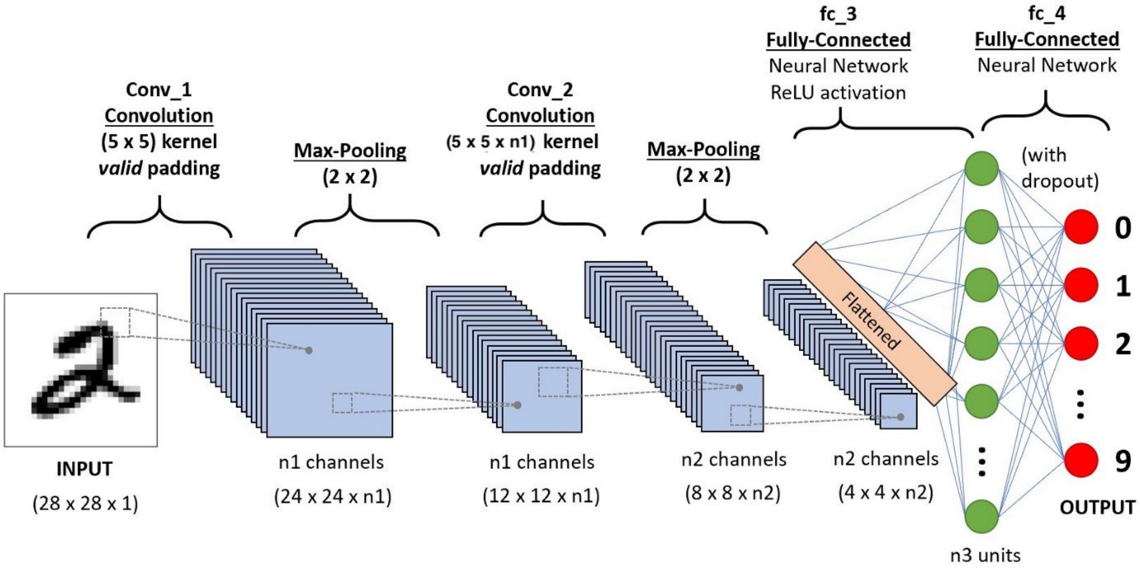
$$J_\theta = \frac{1}{T}\sum_{t=1}^{T}\log p(w_t \mid w_{t-n},\ldots,w_{t-1},w_{t+1},\ldots,w_{t+n})$$

(3) First, it can distinguish the target word from those drawn from the noise distribution to make similar words get closer in the semantic space. Second, it is more efficient since it just needs a set of negative samples for logistic regression calculation instead of using the all words in vocabulary to calculate the numerical probability distributions.

(4) Emb(woman) = Emb(queen) - Emb(king) + Emb(man)

Problem 10 [10 points]

The below CNN model is for digital recognition. The first layer is doing convolution where a kernel 5*5 is used. The second layer is doing pooling where the max-pooling is applied. The third layer is doing convolution where a kernel 5*5*n1 is used. The forth layer is doing pooling where the max-pooling is applied. The fifth layer is a fully connected hidden layer where the hidden size is n3. The sixth layer is a fully connected output layer where the number of output is 10.



(a) Please compute the number of parameters for this model when n1=6, n2=16, n3=128. Write out the number of weights and number of biases separately.

(b) Based on your calculation, what is one significant advantage of using convolutional layer over fully-connected layers when handling image data?

(a) Weights: (5*5)*6+(5*5*6)*16+(4*4*16)*128+128*10=36598
Biases: 6+16+128+10=160

(b) The first two convolutional layers requires only 2550 weight parameters while the last two FC layers requires 34048 weight parameters. Convolutional operation saves a lot of computation by sharing kernel weights across local image patches.