CSE353 Assignment 1
Semin Bae (114730530)

1. $E \ = \ (v,\ h;\ W,\ b,\ c\ ) \ = \ \ -b^T v - c^T h - b^T Wc$

A. What are the dimensions of parameters ‘W’, ‘b’, ‘c’?

answer: $W \ = \ n \times m,\ b = R^n,\ c = R^m$

reason: we know v and h are n, m-dimensional respectively. If we want to apply $b^T v$ (inner product), vector b and v should be in the same dimensional vector*. That’s why vectors b and c are n, m-dimensional respectively. If we apply the Matrix-Vector Product like ‘Wc’, their number of rows and columns are the same. we know $b^T\ is\ 1 \times n$ dimensional for the Transformation, and $c\ is\ m \times 1$ dimensional. That is why $W\ is\ n \times m$ dimensional matrix.

B. Express the first two terms $b^T v$ and $c^T h$ in summation notation

answer: $b^T v \ = \ \sum_i^n b_i v_i$ / $c^T h \ = \ \sum_i^m c_i h_i$ $*\ But\ those\ two\ i\ is\ not\ same\ so\ you\ can\ show\ as\ j\ for\ other\ i$

reason: Inner product $v^T u$ is scalar. v and u should have the same dimensions. and we can denote them as $v^T u \ \equiv \ < v, u > \ \equiv v \cdot u \equiv \sum_i^d v_i u_i$ *

C. Express the last term $b^T Wc$ in summation notation

answer: $\sum_i^n \sum_j^m b_i W_{ij} c_j$

reason: It's a vector-matrix-vector product*. we know Wc is n-dimensional vector so let say Wc as $x \ \in R^n$. Then $b^T x$ is an inner product so we can express as $\sum_i^n b_i x_i$. Then $\sum_i^n b_i x_i \ = \sum_i^n b_i (Wc)_i$. Now, $(Wc)_i$ is a Matrix-vector product. We can express them as $\sum_i^n b_i \sum_j^m W_{ij} c_j \ = \sum_i^n \sum_j^m b_i W_{ij} c_j$

D. Based on the answers to B and C($E = - \sum_i^n b_i v_i - \sum_i^m c_i h_i - \sum_i^n \sum_j^m b_i W_{ij} c_j$), write the expression for

1) $\partial E(v, h; W, b, c)/\partial W_{ij}$

answer: $-\ b_i c_j$

reason: $(- \sum_i^n b_i v_i - \sum_i^m c_i h_i)\partial/\partial W_{ij} = 0$, because those terms do not contain any W. And after derivatives of W is 1. so $(\sum_i^n \sum_j^m b_i W_{ij} c_j)\ \partial/\partial W_{ij} \ = \ b_i \cdot\ 1 \ \cdot\ c_j = b_i c_j$

2) $\partial E(v, h; W, b, c)/\partial b_j$

answer: $-\ v_j - \sum_j^m W_{ij} c_j$

* = referred from p.5 in Math_prelim.pdf in first week

reason: $\sum_i^m c_i h_i = 0$, because it does not contain any b. And $\sum_i^n b_i v_i$ has b_j component where only (other summation sequences do not contain b_j, derivative to zero) j sequence, so $b_j v_j \cdot \partial/\partial b_j = v_j$. And $\sum_i^n \sum_j^m b_i W_{ij} c_j$ contains b_j components, and we know b is n-dimensional. So, $\sum_i^n \sum_j^m b_i x \cdot \partial/\partial b_j = \sum_j^m x$. Then we know $\sum_i^n \sum_j^m b_i W_{ij} c_j = \sum_j^m W_{ij} c_j$.

3) $\partial E(v, h; W, b, c)/\partial c_i$

answer: $-h_i - \sum_i^n b_i W_{ij}$

reason: same reason with question 2) $\partial E(v, h; W, b, c)/\partial b_j$

* = referred from p.5 in Math_prelim.pdf in first week

2. Programming: Given a set D of n-dimensional points ($D = \{x \mid x \in \mathbb{R}n\}$) and a point $y \in \mathbb{R}n$, write a function that returns a point $x \in D$ such that it has the smallest Euclidean distance between it and y.

* My answer used the C# programming

answer:

```
public int[] GetSmallestDistance(int[][] D, int[] vector)
{
   if(D[0].Length != vector.Length)
   {
      Debug.LogError("# Set D and given vector are not in same dimension");
      return null;
   }

   float minDistance = int.MaxValue;
   int minPointer = 0;

   for(int i = 0; i < D.Length; i++)
   {
      float distnace = EuclideanDistance(D[i], vector);

      if (distnace < minDistance)
      {
         minDistance = distnace;
         minPointer = i;
      }
   }

   return D[minPointer];
}

private float EuclideanDistance(int[] v1, int[] v2)
{
   if(v1.Length != v2.Length)
   {
      Debug.LogError("# Those two vector are not in same dimension");
      return float.MaxValue; // Maybe there's better way for exception
   }

   int dimension = v1.Length;
   float returnValue = 0;

   for(int i = 0; i < dimension; i++)
   {
      returnValue += (v1[i] - v2[i]) * (v1[i] - v2[i]);
   }

   returnValue = Mathf.Sqrt(returnValue);

   return returnValue;
}
```

* = referred from p.5 in Math_prelim.pdf in first week

3.

a. Why is optimization important in machine learning?
answer:
Machine learning is the process where a computational entity analyzes massive data through a given algorithm to learn patterns, and creates a model. However, it is not enough to simply create a model. It is also important to evaluate and improve its performance. The optimization is necessary for this purpose. The optimization involves adjusting algorithms, parameters, and data to enhance the model's performance to learn from data. Therefore, optimizing both data and algorithms is essential to create a better machine learning model.

b. What's the significance of an inner product?
answer:
Most data units are vectors, so a good understanding of linear algebra is necessary to optimize and manipulate this data. In particular, the inner product is a significant concept in linear algebra. For example,to calculate the norm of a vector, $||v||$, which is given by $\sqrt{v \cdot v}$, you need to use the inner product to multiply the vector by itself and then take the square root. Similarly, to find the distance between two vectors what we used with 'Question 2', it is given by $d(v_1, v_2) = \sqrt{(v_2 - v_1) \cdot (v_1 - v_2)}$. It also uses the inner product to multiply vectors and take the square root. Additionally, the inner product plays several important roles when dealing with vectors, such as measuring the similarity between them and analyzing their components.

c. Describe two usage of probability/statistics in machine learning.
answer:
I chose the two techniques we covered this week, MLE and BPE. Those are the fundamentals in many machine learning applications, so I select these two probability and statistical techniques.
1. Maximum Likelihood Estimation: MLE is a statistical technique used to estimate parameters from given data. It works by finding the parameter values that maximize the likelihood of observing the data. This method is crucial for determining which parameters best fit the data and for optimizing model performance. As discussed in class, MLE is a fundamental technique used in many machine learning models.
2. Bayesian Parameter estimation: BPE is a prob./stat. technique for estimating parameters by combining data and prior knowledge. It uses Bayes' theorem to combine data and prior probabilities, and calculates posterior probabilities through this. This method is useful for model parameters to reflect uncertainty and increase the reliability of predictions. For example, in spam filtering, the spam probability can be updated based on the characteristics of the email and prior knowledge to perform more accurate spam classification.

* = referred from p.5 in Math_prelim.pdf in first week