CS 541-Deep Learning Week1 Homework

Abhiroop Ajith*, Youness Bani†
Worcester Polytechnic Institute (WPI)
Dept. of Robotics Engineering
Worcester MA, USA
Email: *aajith@wpi.edu, †ybani@wpi.edu

***Index Terms*—Linear Regression,Probability Mass Function,Deep Learning**

I. Introduction

This document solves the problem sets given in the Week 1 Assignment for Dr. Jacob Whitehall's CS541 Deep Learning course for Spring 2022. The assignment focuses on using python and its libraries like numpy and scipy to perform various matrix operations,linear regression and other operations without from scratch without the use of off-the shelf software.

II. Methodology

A. Problem 3

The 3rd Problem deals with using the numpy and scipy libraries for Probability distributions.

1) Problem 3a: The data given *PoissonX.npy* is loaded using np.load function. The plt.hist function is used to plot the given data, so it can be visualized in the form of a graph.This is shown in the figure 5 below:

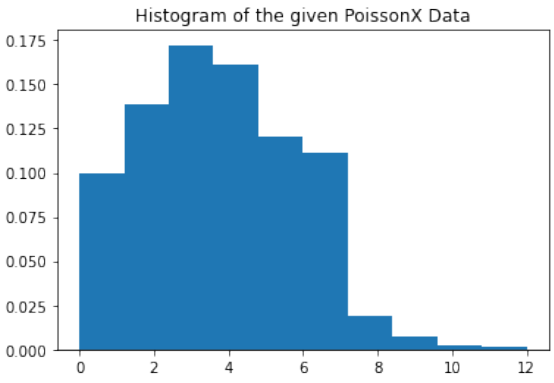

Fig. 1. Histogram of PoissonX Data

Using the scipy library and the scipy.stats.possion function and the online documentation [1] as reference, the probability distribution functions of a random variable with different rate parameters are plotted.The graphs are plotted using the poisson.pmf function and plt.vlines function.The graphs are plotted for the various values of Mu=2.5, 3.1, 3.7, and 4.3 and are shown in the figures below.

As you can see in the resulting graphs in Fig 2,3,4,5, the graph

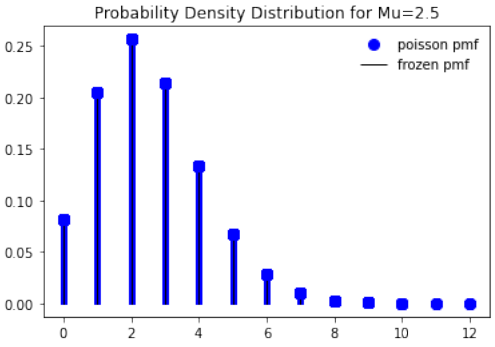

Fig. 2. Probability Density Distribution for Mu=2.5

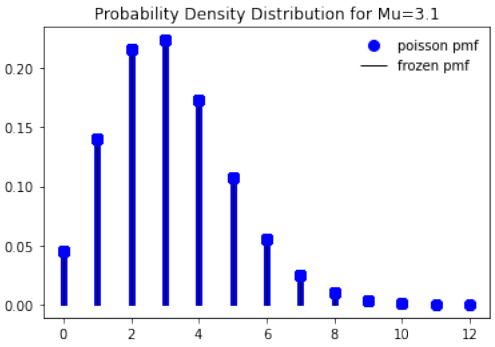

Fig. 3. Probability Density Distribution for Mu=3.1

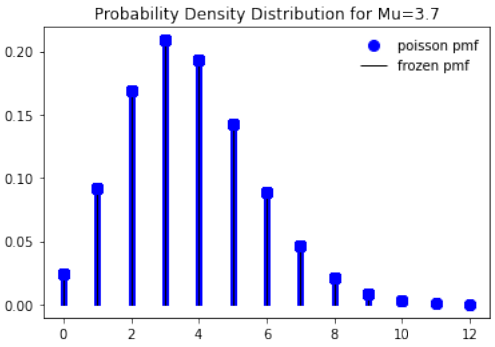

Fig. 4. Probability Density Distribution for Mu=3.7

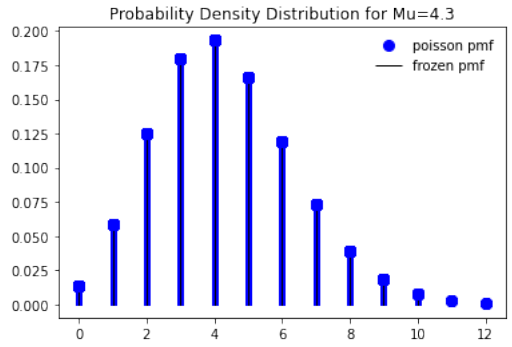

Fig. 5. Probability Density Distribution for Mu=4.3

that looks the most consistent and similar to the data given in the "PoissionX" file is when **Mu=3.7**. **Therefore the answer for 3a is when Mu=3.7 the data is most consistent with the given data/histogram and is the alternative rate parameter.**

2) Problem 3b: For Part(i): The Y value is compared to the mean, therefore $y = x^2$, **The value of y tends to be larger for all values of x from -inf to +inf except when x ranges from -1 to 1 (including 0)** it remains smaller or equal
Part(ii):The Y value is compared to the variance to find the uncertainty. The graph is plotted on MATLAB. From the graph

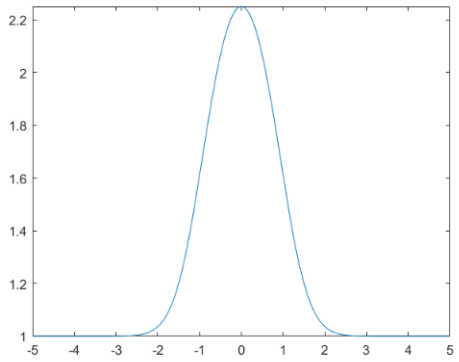

Fig. 6. Graph of the Variance function on MATLAB

we can compute that **when the value of x is, x =0 in magnitude, the value of y tends to be larger as shown in the figure 6**

B. Problem 2

The 2nd problem deals with linear regression using numpy on the given age regression dataset. The cost function used in this document is the *MSE Cost Function* which is given by the Formula:

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{y}^{(i)} - y^{(i)})^2 \tag{1}$$

The reported MSE Cost on the training set and the testing set are as follows:

Training Error:100.930991876925
Testing Error:538.3985786068

C. Problem 4

This Problem deals with Proofs and derivations

1) Problem 4a: Given that:

$$\bigtriangledown_x(x^T a) = \frac{\partial}{\partial x_k}(x^T a) = \frac{\partial}{\partial x_k}(a^T x) \tag{2}$$

for k ϵ (1,2,.....n)

$$\frac{\partial}{\partial x_k}(\sum_{i=1}^{n} a_i x_i) = \begin{bmatrix} \frac{\partial}{\partial x_1}(\sum_{i=1}^{n})a_i x_i \\ .. \\ .. \\ \frac{\partial}{\partial x_n}(\sum_{i=1}^{n})a_i x_i \end{bmatrix}$$

$$\bigtriangledown_x(x^T a) = \begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ a_n \end{bmatrix} = a$$

$$\bigtriangledown_x(x^T a) = \bigtriangledown_x(a^T x) = a \tag{3}$$

Hence Proved.

2) Problem 4b:

$$\frac{\partial}{\partial x}(x^T A x) = \frac{\partial}{\partial x}(\sum_{i=1}^{n} x_i A_{i1} + \sum_{i=1}^{n} x_i A_{i2} + ... + \sum_{i=1}^{n} x_i A_{in}) \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ x_n \end{bmatrix}$$

$$\frac{\partial}{\partial x}(x^T A x) = \sum_i x_i A_{ik} \sum_j x_j A_{kj} = \frac{\partial}{\partial x}(\sum_j \sum_i x_i A_{ij} x_j) = \frac{\partial}{\partial x}(\sum_i \sum_j x_i A_{ij} x_j)$$

$$= \frac{\partial}{\partial x_k}(\sum_{i \neq k}^{n} x_{ik} A_{ik} x_k + \sum_{j \neq k}^{n} x_j A_{kj} x_k + x_k A_{kk})$$

$$= \sum_{i \neq k}^{n} x_i A_{ik} + \sum_{j \neq k}^{n} x_j A_{kj} + 2 x_k A_{kk}$$

$$\frac{\partial}{\partial x_k}(x^T A x) = \sum_i x_i A_{ik} + \sum_j x_j A_{kj}$$

$$\frac{\partial}{\partial x_k}(x^T A x) = A_k^T x + A_k x$$

$$\bigtriangledown_x(x^T a) == \begin{bmatrix} A_1^T x + A_1 x \\ \cdots \\ \cdots \\ A_n x + A_n^T x \end{bmatrix} = (A + A^T)x \tag{4}$$

Hence Proved.

3) Problem 4c: From Equation (4) we know that:

$$\bigtriangledown_x(x^T a) = (A + A^T)x$$

If A is symmetric then $\rightarrow$ A^T=A Therefore Equation 4 becomes:

$$\bigtriangledown_x(x^T a) = 2Ax$$

4) Problem 4d: In Problem 4d, Multiplying

$$\begin{aligned}(Ax+b)^T(Ax+b) &= (b^T + x^T A^T)(Ax+b) \\ &= b^T Ax + b^T b + x^T A^T Ax + x^T A^T b\end{aligned}$$

$$\bigtriangledown_x[(Ax+b)^T(Ax+b)] = b^T A + 2A^T Ax + A^T b$$

Since $b^T A = A^T b$

$$\bigtriangledown_x[(Ax+b)^T(Ax+b)] = 2A^T Ax + 2A^T b$$

$$\bigtriangledown_x[(Ax+b)^T(Ax+b)] = 2A^T(Ax+b) \tag{5}$$

Hence Proved.

REFERENCES

[1] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.poisson.html