Problems 2

1. Backpropagation (3 pts)

Let $J : \mathbb{R}^m \to \mathbb{R}$ be a loss function of an affine transformation $Wx + b$, where $W \in \mathbb{R}^{m\times d}$, $x \in \mathbb{R}^d$, and $b \in \mathbb{R}^m$. Our goal is to find the partial derivative of J with respect to each element of W, $\frac{\partial J}{\partial W_{ij}}$ as well as the $\frac{\partial J}{\partial b_i}$, for each element of b. For convenience, let $y = Wx + b$, so $J(y)$ is a function of y. Suppose we have already computed the partial derivative of J with respect to the entries of $y = (y_1, \dots, y_m)^\top$, $\frac{\partial J}{\partial y_i}$ for $i = 1, \dots, m$. Then by the chain rule, we have

$$\frac{\partial J}{\partial W_{ij}} = \sum_{k=1}^{m} \frac{\partial J}{\partial y_k} \frac{\partial y_k}{\partial W_{ij}}.$$

(a) (1 pts) Show that $\frac{\partial J}{\partial W_{ij}} = \frac{\partial J}{\partial y_i} x_j$, where $x = (x_1, \dots, x_d)^\top$. (not required, but you might want to use Dirac delta function, $\delta_{ij} = 1,$ if $i = j,$ otherwise 0)

(b) (1 pts) Give a vectorized expression for $\frac{\partial J}{\partial W}$ in terms of the column vectors $\frac{\partial J}{\partial y}$ and x.

(c) (1 pts) Show that

$$\frac{\partial J}{\partial x} = W^\top \frac{\partial J}{\partial y}.$$

2. [Coding problem] GMM (Gaussian Mixture Model) (7 pts)

(a) (2 pts) Write the code to generate the dataset. There are 3 clusters, $x^{(i)} \in \mathbb{R}^2$. The mean of each clusters are $\mu_1 = [7.0, -1.0], \mu_2 = [3.0, -1.5], \mu_3 = [5.5, 1.0]$., and standard deviation is same for all clusters, $\sigma = 0.7$. You generate dataset by sampling 100 data points for each cluster. Plot the generated dataset in 2D scatter plot (using different colors (or shapes) for different clusters).

(b) (2 pts) Using K-means algorithm to cluster the dataset you generated (you are not allowed to use any kmeans libraries, implement it by yourself). Try to run k-means algorithm for k=2, 3, 4 and 6. Provide training curve every iterations (x-axis: training iteration, y-axis: distortion, you can run multiple times with different initializations, and pick the best one to plot). Plot the final centroids (marker ='x', like cs229 lecture note in Figure 1) over the dataset plot you provided in (a) for each different k. So you would provide 4 different plots w/ 4 different k.

(c) (3 pts) Using GMM to dataset you generated. In this problem, you can use any GMM libraries if you want, but I strongly recommend you to implement by yourself. Try to run GMM algorithm for k=2, 3, 4 and 6. Plot the obtained μ (marker ='x') over the dataset plot you provided in (a) for each different k. So you would provide 4 different plots w/ 4 different k.