

CS280 Fall 2022 Assignment 1

Part A

Basics & MLP

February 18, 2023

Name: Bingnan Li

Student ID: 2020533092

1. Gradient descent for fitting GMM (10 points).

Consider the Gaussian mixture model

$$p(\mathbf{x}|\theta) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

where $\pi_j \geq 0, \sum_{j=1}^K \pi_j = 1$. (Assume $\mathbf{x}, \boldsymbol{\mu}_k \in \mathbb{R}^d, \boldsymbol{\Sigma}_k \in \mathbb{R}^{d \times d}$)

Define the log likelihood as

$$l(\theta) = \sum_{n=1}^N \log p(\mathbf{x}_n|\theta)$$

Denote the posterior responsibility that cluster k has for datapoint n as follows:

$$r_{nk} := p(z_n = k|\mathbf{x}_n, \theta) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})}$$

(a) Show that the gradient of the log-likelihood wrt $\boldsymbol{\mu}_k$ is

$$\frac{d}{d\boldsymbol{\mu}_k} l(\theta) = \sum_n r_{nk} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k)$$

Proof. By the *chain rule*, we have

$$\begin{aligned} \frac{\partial l(\theta)}{\partial \boldsymbol{\mu}_k} &= \sum_{n=1}^N \frac{1}{p(\mathbf{x}_n|\theta)} \frac{\partial p(\mathbf{x}_n|\theta)}{\partial \boldsymbol{\mu}_k} \\ &= \sum_{n=1}^N \frac{1}{p(\mathbf{x}_n|\theta)} \frac{\partial \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\partial \boldsymbol{\mu}_k} \\ &= \sum_{n=1}^N \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{p(\mathbf{x}_n|\theta)} \frac{\partial (-\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k))}{\partial \boldsymbol{\mu}_k} \\ &= \sum_{n=1}^N r_{nk} \boldsymbol{\Sigma}^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) \end{aligned}$$

■

(b) Derive the gradient of the log-likelihood wrt π_k without considering any constraint on π_k . (bonus 2 points: with constraint $\sum_k \pi_k = 1$.)

Proof. For the case without any constraint on π_k , by the *chain rule*, we have:

$$\begin{aligned} \frac{\partial l(\theta)}{\partial \pi_k} &= \sum_{n=1}^N \frac{1}{p(\mathbf{x}_n|\theta)} \frac{\partial p(\mathbf{x}_n|\theta)}{\partial \pi_k} \\ &= \sum_{n=1}^N \frac{1}{p(\mathbf{x}_n|\theta)} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ &= \sum_{n=1}^N \frac{r_{nk}}{\pi_k} \end{aligned}$$

■

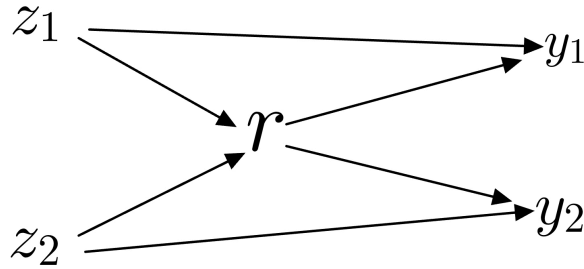
2. Softmax & Computation Graph (10 points).

Recall that the softmax function takes in a vector (z_1, \dots, z_D) and returns a vector (y_1, \dots, y_D) . We can express it in the following form:

$$r = \sum_j e^{z_j} \quad y = \frac{e^{z_j}}{r}$$

(a) Consider $D = 2$, i.e. just two inputs and outputs to the softmax. Draw the computation graph relating z_1 , z_2 , r , y_1 , and y_2 .

Solution:



(b) Determine the backprop updates for computing the \bar{z}_j when given the \bar{y}_i . You need to justify your answer. (You may give your answer either for $D = 2$ or for the more general case.)

Solution:

By the *chain rule* and the forward precess, we have the backprop updates as follows:

$$\begin{aligned} \bar{r} &= \sum_{i=1}^D \bar{y}_i \frac{\partial y_i}{\partial r} = - \sum_{i=1}^D \bar{y}_i \frac{e^{z_i}}{r^2} \\ \bar{z}_i &= \bar{y}_i \frac{\partial y_i}{\partial z_i} + \bar{r} \frac{\partial r}{\partial z_i} = \bar{y}_i \frac{e^{z_i}}{r} + \bar{r} e^{z_i} \end{aligned}$$

(c) Write a function to implement the vector-Jacobian product (VJP) for the softmax function based on your answer from part (b). For efficiency, it should operate on a mini-batch. The inputs are:

- a matrix \mathbf{Z} of size $N \times D$ giving a batch of input vectors. N is the batch size and D is the number of dimensions. Each row gives one input vector $z = (z_1, \dots, z_D)$.
- A matrix \mathbf{Y}_{bar} giving the output error signals. It is also $N \times D$

The output should be the error signal \mathbf{Z}_{bar} . Do not use a for loop.