Machine Learning October, 2022

Problem Set 2

Instructor: Prof. Jie Tang, Prof. Jun Zhu *Student Name*

Requirements:

- We recommend that you typeset your homework using appropriate software such as LaTeX. If you submit your handwritten version, please make sure it is cleanly written up and legible. The TAs will not invest undue effort to decrypt bad handwritings.
- We have programming tasks in each homework. Please submit the source code together with your homework. Please include experiment results using figures or tables in your homework, instead of asking TAs to run your code.
- Please finish your homework independently. In addition, you should write in your homework the set of people with whom you collaborated.

1 Collaborators and Sources

Please list your collaborators and sources here.

2 Back Propagation (3pts)

Problem 1 (3pts)**.** For a batch of training samples $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^n, \boldsymbol{x}_i = (\boldsymbol{x}_{i,1}, ..., \boldsymbol{x}_{i,n_x})^\top \in \mathbb{R}^{n_x}, \boldsymbol{y}_i = (\boldsymbol{y}_{i,1}, ..., \boldsymbol{y}_{i,n_y})^\top \in \mathbb{R}^{n_y}$, we implement a two-layer neural network, which defines a function $g : \mathbb{R}^{n_x \times n} \to \mathbb{R}^{n_y \times n}$, i.e. $g(\{\boldsymbol{x}_i\}_{i=1}^n) = \{\hat{\boldsymbol{y}}_i\}_{i=1}^n$, satisfying

$$
\begin{aligned}
\boldsymbol{z}_{1,i} &= \boldsymbol{W}^{(1)}\boldsymbol{x}_i + \boldsymbol{b}^{(1)} \in \mathbb{R}^{n_1}, \quad \boldsymbol{W}^{(1)} \in \mathbb{R}^{n_1 \times n_x}, \boldsymbol{b}^{(1)} \in \mathbb{R}^{n_1}, i = 1, 2, ..., n \\
\boldsymbol{h}_{1,i} &= \mathrm{ReLU}(\boldsymbol{z}_{1,i}) \in \mathbb{R}^{n_1} \\
\boldsymbol{\mu} &= \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{h}_{1,i} \in \mathbb{R}^{n_1} \\
\boldsymbol{\sigma}_k^2 &= \frac{1}{n}\sum_{i=1}^{n} (\boldsymbol{h}_{1,i,k} - \boldsymbol{\mu}_k)^2 \in \mathbb{R}, \quad k = 1, 2, ..., n_1 \\
\hat{\boldsymbol{h}}_{1,i,k} &= \gamma \frac{\boldsymbol{h}_{1,i,k} - \boldsymbol{\mu}_k}{\sqrt{\boldsymbol{\sigma}_k^2 + \epsilon}} + \beta \in \mathbb{R}, \quad k = 1, 2, ..., n_1, \quad \text{BN-layer} \\
\boldsymbol{z}_{2,i} &= \boldsymbol{W}^{(2)}\hat{\boldsymbol{h}}_{1,i} + \boldsymbol{b}^{(2)} \in \mathbb{R}^{n_y}, \quad \boldsymbol{W}^{(2)} \in \mathbb{R}^{n_y \times n_1}, \boldsymbol{b}^{(2)} \in \mathbb{R}^{n_y}, i = 1, 2, ..., n \\
\hat{\boldsymbol{y}}_i &= \mathrm{Softmax}(\boldsymbol{z}_{2,i}) \in \mathbb{R}^{n_y},
\end{aligned}
\tag{2.1}
$$

here $\boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)}, \gamma, \beta, \boldsymbol{W}^{(2)}, \boldsymbol{b}^{(2)}$ are parameters we can optimize and ϵ is constant. We define the loss function as

$$
f_{\mathrm{CE}}(\boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)}, \gamma, \beta, \boldsymbol{W}^{(2)}, \boldsymbol{b}^{(2)}) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{n_y} \boldsymbol{y}_{i,k} \log \hat{\boldsymbol{y}}_{i,k}. \tag{2.2}
$$

Calculate

$$
\frac{\partial f_{\mathrm{CE}}}{\partial \boldsymbol{b}^{(2)}}, \frac{\partial f_{\mathrm{CE}}}{\partial \boldsymbol{W}^{(2)}}, \frac{\partial f_{\mathrm{CE}}}{\partial \beta}, \frac{\partial f_{\mathrm{CE}}}{\partial \gamma}, \frac{\partial f_{\mathrm{CE}}}{\partial \boldsymbol{b}^{(1)}}, \frac{\partial f_{\mathrm{CE}}}{\partial \boldsymbol{W}^{(1)}}. \tag{2.3}
$$

Notice: The form like $\frac{\partial \cdot}{\partial \cdot}$ is not allowed to appear in the result.

3 Mixtures of Logistic Models (3pts)

In this part, we consider **mixtures of logistic models**.

Recall the sigmoid function

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \tag{3.1}$$

we consider K logistic models, each governed by its own weight parameter $\boldsymbol{w}_k$. If we denote the mixing coefficients by π_k satisfying $\sum_{k=1}^{K} \pi_k = 1$, then the mixture distribution can be written as

$$p(t|\phi, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k y_k^t [1 - y_k]^{1-t}, \tag{3.2}$$

where t is the target variable, ϕ is the feature vector, $y_k = \sigma(\boldsymbol{w}_k^\top \phi)$ is the output of component k, here $\phi, \boldsymbol{w}_k \in \mathbb{R}^d$, and $\boldsymbol{\theta}$ denotes the adjustable parameters namely $\{\pi_k\}_{k=1}^K$ and $\{\boldsymbol{w}_k\}_{k=1}^K$.

Now suppose we are given a data set $\{\phi_n, t_n\}_{n=1}^N$. The corresponding likelihood function is then given by

$$p(\boldsymbol{t}|\boldsymbol{\theta}) = \prod_{n=1}^{N} \left(\sum_{k=1}^{K} \pi_k y_{nk}^{t_n} [1 - y_{nk}]^{1-t_n} \right), \tag{3.3}$$

where $y_{nk} = \sigma(\boldsymbol{w}_k^\top \phi_n)$ and $\boldsymbol{t} = (t_1, t_2, ..., t_N)^\top$. We consider the log-likelihood function

$$L(\boldsymbol{\theta}) = \log p(\boldsymbol{t}|\boldsymbol{\theta}). \tag{3.4}$$

Problem 2 (3pts)**.** Please use the above definition to solve this problem

1. First, we fix $\{\boldsymbol{w}_k\}_{k=1}^K$ and consider

$$\max_{\boldsymbol{\pi}} L(\boldsymbol{\theta}) \quad s.t. \sum_{k=1}^{K} \pi_k = 1. \tag{3.5}$$

   By using Lagrange multiplier to prove that the optimal solution $\{\pi_k\}_{k=1}^K$ satisfying

$$\pi_k = \frac{\sum_{n=1}^{N} \gamma_{nk}}{N}, \tag{3.6}$$

   here

$$\gamma_{nk} = \frac{\pi_k y_{nk}^{t_n} [1 - y_{nk}]^{1-t_n}}{\sum_j \pi_j y_{nj}^{t_n} [1 - y_{nj}]^{1-t_n}}. \tag{3.7}$$

2. Now we fix $\{\pi_k\}_{k=1}^K$, prove that

$$\nabla_{\boldsymbol{w}_k} L = \sum_{n=1}^{N} \gamma_{nk}(t_n - y_{nk})\phi_n. \tag{3.8}$$

3. Further, calculate $\boldsymbol{H}_k = -\nabla_{\boldsymbol{w}_k} \nabla_{\boldsymbol{w}_k} L$.

4 Generative Adversarial Networks (1pt)

Problem 3. The minimax objective function in GAN is

$$\min_G \max_D (\mathbb{E}_{p_{data}(x)}[logD(x)] + \mathbb{E}_{p(z)}[log(1 - D(G(z)))]) \tag{4.1}$$

Prove that the optimal solution of D is

$$D(x) = \frac{p_{data}(x)}{p_{data}(x) + p_{model}(x)} \tag{4.2}$$

where p_{model} is the distribution of $G(z)$ when $z \sim p(z)$.

5 Clustering (3pts)

5.1 EM for mixture of multinomials

Recall a multinomial distribution with the parameter $\boldsymbol{\mu} = (\mu_i)_{i=1}^d$:

$$P(\boldsymbol{x} \mid \boldsymbol{\mu}) = \frac{n!}{\prod_i x_i!} \prod_i \mu_i^{x_i}, \quad i = 1, \cdots, d \tag{5.1}$$

where $x_i \in \mathbb{N}$, $\sum_i x_i = n$, and $0 < \mu_i < 1$, $\sum_i \mu_i = 1$.

Consider the following mixture-of-multinomials model to analyze a corpus of documents that are represented in the bag-of-words model.

Specifically, assume we have a corpus of D documents and a vocabulary of W words from which every word in the corpus is token. We are interested in counting how many times each word appears in each document, regardless of their positions and orderings. We denote by $T \in \mathbb{N}^{D \times W}$ the word occurrence matrix where the w-th word appears T_{dw} times in the d-th document. According to the mixture-of-multinomials model, each document is generated i.i.d. as follows. We first choose for each document d a latent "topic" c_d (analogous to choosing for each data point a component z_n in the mixture-of- Gaussians) with

$$P(c_d = k) = \pi_k, k = 1, 2, \cdots, K; \tag{5.2}$$

And then given this "topic" $\boldsymbol{\mu}_k = (\mu_{1k}, \ldots, \mu_{Wk})$ which now simply represents a categorical distribution over the entire vocabulary, we generate the word bag of the document from the corresponding multinomial distribution [1]

$$P(d \mid c_d = k) = \frac{n_d!}{\prod_w T_{dw}!} \prod_w \mu_{wk}^{T_{dw}}, \tag{5.3}$$

where $n_d = \sum_w T_{dw}$. Hence in summary

$$P(d) = \sum_{k=1}^K P(d \mid c_d = k) P(c_d = k) = \frac{n_d!}{\prod_w T_{dw}!} \sum_{k=1}^K \pi_k \prod_w \mu_{wk}^{T_{dw}}. \tag{5.4}$$

Problem 4 (1.5pts)**.** Given the corpus T, design and derive an EM algorithm to learn the parameters $\{\boldsymbol{\pi}, \boldsymbol{\mu}\}$ of this mixture model.

Problem 5 (1.5pts)**.** Implement the EM algorithm on the Newsgroups dataset.

Set the number of topics K to be 10, 20, 30, 50 respectively and show the most-frequent words in each topic for each case.

[1] Make sure you understand the difference between a categorical distribution and a multinomial distribution. You may think about a Bernoulli distribution and a binomial distribution for reference.