Homework 1
601.482/682 Deep Learning
Spring 2022
Ting He

Feb 1, 2022

Due Wednesday, Feb 2, 11:59 pm EST
Please type your answers inline of the LaTeX file
Submit PDF to Gradescope (Entry Code: WYPD4D)

Important: Please use the natural logarithm (base e) for all logarithm calculations below. Please show each step of your calculations; points will be deducted if only final results are present.

1. In this exercise you will derive the well-known sigmoid expression for a Bernoulli distributed (binary) problem. The probability of the "positive" event occurring is p. The probability of the "negative" event occurring is $q = 1 - p$.

   (a) What are the odds o of the "positive" event occurring? Please express the result using p only.

   A: odds(positive event) $= \frac{p}{1-p}$

   (b) Given $\text{logit}(p) = x$, please derive the inverse function $\text{logit}^{-1}(x)$. Please express the result using x only. *Hint:* In statistics, the logit of the probability is the logarithm of the corresponding odds, i.e. $\text{logit}(p) = \log(o)$.

   A: Given $logit(p) = log(odds) = log(\frac{p}{1-p}) = x$
   We can derive that $e^x = \frac{p}{1-p}$
   such that $e^x = p(1 + e^x)$
   then $p = \frac{e^x}{1+e^x}$
   $logit^{-1}(x) = \frac{e^x}{1+e^x}$

   (c) The inverse function of the logit in (b) is actually the sigmoid function $S(x)$. You may already have noticed that the probability $p = \text{logit}^{-1}(x) = S(x)$. This means that the range of the sigmoid function is the same as the range of a probability, i.e. $(0, 1)$. The domain of the sigmoid function is $(-\infty, \infty)$. Therefore, the sigmoid function maps all real numbers to the interval $(0, 1)$.
   Now we look into the saturation problem of the sigmoid function. Calculate the values of the sigmoid function $S(x)$ for $x = \pm 100, \pm 10$, and 0. List your results (rounded to two decimal places) with corresponding x values.

   A: Since $S(x) = \frac{e^x}{1+e^x}$
   (rounded to 2 decimal places)
   given $x = 0, S(x) \approx 0.50$
   given $x = 10, S(x) \approx 1.00$
   given $x = -10, S(x) \approx 0.00$
   given $x = 100, S(x) \approx 1.00$

given $x = -100, S(x) \approx 0.00$

(d) Calculate the derivatives of the sigmoid function $S'(x)$ and the values of $S'(x)$ for $x = \pm 100, \pm 10$, and 0. List your results (rounded to two decimal places) with corresponding x values.

You may have noticed that $S(\pm 100)$ is very close to $S(\pm 10)$; the derivatives at $x = \pm 100$ and $x = \pm 10$ are very close to zero. This is the saturation of the sigmoid function when $|x|$ is large. The saturation brings great difficulty in training deep neural networks. This will reappear in later lectures.

A: The derivative is

$$S^{'} = (\frac{e^x}{1+e^x})^{'} = \frac{e^x(1+e^x) - e^{2x}}{(1+e^x)^2} = \frac{e^x}{1+e^{2x}+2e^x} = S(1-S)$$

when $x = 0, S(x) = 0.25$
when $x = \pm 10, S(x) = 0.00$
when $x = \pm 100, S(x) = 0.00$

2. Recall in class, we learned the form of a linear classifier as $f(\boldsymbol{x}; \boldsymbol{W}) = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$. We will soon learn, that iteratively updating the weights in negative gradient direction will allow us to slowly move towards an optimal solution. We will call this technique backpropagation. Obviously, computing gradients is an important component of this technique. We will investigate the first derivative of a commonly used loss function: the softmax loss. Here, we consider a multinomial (multi-class) problem.
We first define the following notation:

$$\begin{aligned}
\text{input features}: &\quad \boldsymbol{x} \in \mathbb{R}^D. \\
\text{target labels (one-hot encoded)}: &\quad \boldsymbol{y} \in \{0,1\}^K. \\
\text{multinomial linear classifier}: &\quad \boldsymbol{f} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}, \quad \boldsymbol{W} \in \mathbb{R}^{K \times D} \text{ and } \boldsymbol{f}, \boldsymbol{b} \in \mathbb{R}^K \\
\text{e.g., for the k-th classification}: &\quad f_k = \boldsymbol{w}_k^T \boldsymbol{x} + b_k, \text{ corresponding to } y_k, \\
&\quad \text{where } \boldsymbol{w}_k^T \text{ is the k-th row of } \boldsymbol{W}, k \in \{1...K\}
\end{aligned}$$

(a) Please express the softmax loss of logistic regression, $L(\boldsymbol{x}, \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{y})$, using the above notation.

(b) Please calculate its gradient derivative $\frac{\partial L}{\partial \boldsymbol{w}_k}$.

A: softmax loss:

$$\begin{aligned}
L &= -\log p(Y = truelabel | X = x) \\
&= -\log \sum_i yi P(Y = i | X = x) \\
&= -\log \sum_i yi \frac{e^{f_i}}{\sum_j e^{f_j}} \\
&= -\log \sum_i yi \frac{e^{(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b})_i}}{\sum_j e^{(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b})_j}} \\
&= -\log \frac{\boldsymbol{y}^T e^{\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b}}}{\boldsymbol{1}^T e^{\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b}}}
\end{aligned}$$

A: to calculate the gradient derivative $\frac{\partial L}{\partial \boldsymbol{w}_k}$, we only need to calculate every $\frac{\partial L}{\partial \boldsymbol{w}_{ki}}$. Denote true label of $\boldsymbol{y}$ as l. We have

$$\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{w}_{ki}} &= -\frac{1}{P(Y = l | X = \boldsymbol{x})} \frac{\partial P(Y = l | x = \boldsymbol{x})}{\partial f_k} \frac{\partial f_k}{\partial w_{ki}} \\
\frac{\partial f_k}{\partial w_{ki}} &= \boldsymbol{x}_i
\end{aligned}$$

$$\frac{\partial P(Y=l|x=\boldsymbol{x})}{\partial f_k} = \frac{\partial \frac{\sum_i y_i e^{f_i}}{\sum_j e^{f_j}}}{\partial f_k} = \frac{y_k e^{f_k}\sum_i e^{f_i} - e^{f_{f_k}}\sum_i y_i e^{f_i}}{\left(\sum_i e^{f_i}\right)^2}$$
$$= P(Y=k|X=\boldsymbol{x})(y_k - p(Y=l|X=\boldsymbol{x}))$$

Therefore, we have $\frac{\partial L}{\partial \boldsymbol{w}_k} = -[y_k - p(Y=k|X=\boldsymbol{x})]\boldsymbol{x}$

3. In class, we briefly touched upon the Kullback-Leibler (KL) divergence as another loss function to quantify agreement between two distributions p and q. In machine learning scenarios, one of these two distributions will be determined by our training data, while the other one is generated as an output of our model. The goal of training our model is to match these two distributions as well as possible. KL divergence is asymmetric, so assigning these distributions to p and q will matter. Here, you will investigate this difference by calculating the gradient. Recall that the KL divergence is defined as

$$\mathrm{KL}(p||q) = \sum_d p(d)\log\left(\frac{p(d)}{q(d)}\right)$$

(a) Show that the KL divergence is asymmetric using the following example. We define a discrete random variable X. Now consider the case that we have two discrete distributions, $P(x)$ and $Q(x)$, which we present as two vectors that express the frequency of an event x:

$$P(x) = [1,\ 6,\ 12,\ 5,\ 2,\ 8,\ 12,\ 4]$$
$$Q(x) = [1,\ 3,\ 6,\ 8,\ 15,\ 10,\ 5,\ 2]$$

Please compute the following:

i. The probability distributions, $p(x)$ and $q(x)$ (*Hint*: Normalize the distributions)

ii. Both directions of KL divergence, $\mathbf{KL}(p||q)$ and $\mathbf{KL}(q||p)$.

A. by adding each items at frequency table, we can get the total frequency for both $P(x)$ and $Q(x)$ are 50, then divided the each frequency with the total frequency of 50, we can get the probability distributions as following:

$$p(x) = [0.02, 0.12, 0.24, 0.10, 0.04, 0.16, 0.24, 0.08]$$

$$q(x) = [0.02, 0.06, 0.12, 0.16, 0.30, 0.20, 0.10, 0.04]$$

A: As we know the KL divergence as

$$\mathrm{KL}(p||q) = \sum_d p(d)\log\left(\frac{p(d)}{q(d)}\right)$$

Fill in the probability distribution from previous sub-question, we can get:

$$\begin{aligned} KL(p||q) &= \sum_d p(d)\log\left(\frac{p(d)}{q(d)}\right) \\ &= 0 + 0.12\log 2 + 0.24\log 2 + 0.1\log 0.625 + 0.04\log 0.133 \\ &\quad + 0.16\log 8 + 0.24\log 24 + 0.08\log 2 \\ &= 1.273 \end{aligned}$$

Similarly,

$$\begin{aligned} KL(q||p) &= \sum_d q(d)\log\left(\frac{q(d)}{p(d)}\right) \\ &= 0.02\log 1 + 0.06\log 0.5 + 0.12\log 0.5 + 0.16\log 0.16 + 0.3\log 7.5 \\ &\quad + 0.2\log 1.25 + 0.1\log 0.417 + 0.04\log 0.5 \\ &= -0.258 \end{aligned}$$

(b) Next, we try to optimize the weights $\boldsymbol{W}$ of an arbitrary model in an attempt to minimize KL divergence. As a consequence, $q = q_{\boldsymbol{W}}$ now depends on the weights.

i. Please express $\mathbf{KL}(q_{\boldsymbol{W}}||p)$ and $\mathbf{KL}(p||q_{\boldsymbol{W}})$ as optimization objective functions. You need to specify your optimization variables and express the objective as a min / max function. For example: $\max_x f(x, a, b)$. Please further derive and simplify your optimization objective functions from the original expression so that constant terms (not relevant for optimization procedures) are grouped.

ii. Can you tell which direction is easier to compute? Please state your choice of direction ($\mathbf{KL}(q_{\boldsymbol{W}}||p)$ or $\mathbf{KL}(p||q_{\boldsymbol{W}})$) and explain your reasoning in detail. Points will be deducted if you fail to explain your answer, regardless of correctness.
Hint: Please look into your derived optimization objective functions and think about which term is difficult to compute in practice. You may use 3(a)'s discrete distribution as an example to think about processing data in real world problems.

iii. Please calculate the gradient of $\mathbf{KL}(q_{\boldsymbol{W}}||p)$ and $\mathbf{KL}(p||q_{\boldsymbol{W}})$ w.r.t. $q_{\boldsymbol{W}}(d)$, the d-th element of $q_{\boldsymbol{W}}$. Please show your steps in detail.

A: We know that:

$$KL(q_{\boldsymbol{W}}||p) = \sum_d q_{\boldsymbol{W}}(d) \log\left(\frac{q_{\boldsymbol{W}}(d)}{p(d)}\right) \tag{1}$$

$$= \sum_d q_{\boldsymbol{W}}(d) \log\left(q_{\boldsymbol{W}}(d)\right) - \sum_d q_{\boldsymbol{W}}(d) \log\left(p(d)\right) \tag{2}$$

Similarly,

$$KL(p||q_{\boldsymbol{W}}) = \sum_d p(d) \log\left(\frac{p(d)}{q_{\boldsymbol{W}}(d)}\right) \tag{3}$$

$$= \sum_d p(d) \log\left(p(d)\right) - \sum_d p(d) \log\left(q_{\boldsymbol{W}}(d)\right) \tag{4}$$

We try to optimize the weights $\boldsymbol{W}$ by minimizing the KL divergence,

$$\min_{\boldsymbol{W}} KL(p||q_{\boldsymbol{W}}) = \min_{\boldsymbol{W}}(\sum_d p(d) \log\left(p(d)\right) - \sum_d p(d) \log\left(q_{\boldsymbol{W}}(d)\right))$$

Since in the left of the equation (4), it is not relevant to $\boldsymbol{W}$, I will use C_1 to replace the item.

$$\begin{aligned}\min_{\boldsymbol{W}} KL(p||q_{\boldsymbol{W}}) &\Leftrightarrow \min_{\boldsymbol{W}}(C_1 - \sum_d p(d) \log\left(q_{\boldsymbol{W}}(d)\right)) \\ &\Leftrightarrow \max_{\boldsymbol{W}}(\sum_d p(d) \log\left(q_{\boldsymbol{W}}(d)\right)) \\ &\Leftrightarrow \max_{\boldsymbol{W}}(E_p \log q_{\boldsymbol{W}})\end{aligned}$$

Similarly,

$$\min_{\boldsymbol{W}}(KL(q_{\boldsymbol{W}}||p)) \Leftrightarrow \min_{\boldsymbol{W}}(\sum_d q_{\boldsymbol{W}}(d) \log\left(q_{\boldsymbol{W}}(d)\right) - \sum_d q_{\boldsymbol{W}}(d) \log\left(p(d)\right))$$

A: $KL(p||q_{\boldsymbol{W}})$ is easier compute than $KL(q_{\boldsymbol{W}}||p)$.
First of all, the $KL(p||q_{\boldsymbol{W}})$ only has one term needs to maximize, $E_p \log q_{\boldsymbol{W}}$, while $KL(q_{\boldsymbol{W}}||p)$ has two separate term need to minimize, $\sum_d q_{\boldsymbol{W}}(d) \log\left(q_{\boldsymbol{W}}(d)\right) - \sum_d q_{\boldsymbol{W}}(d) \log\left(p(d)\right)$. Secondly, the $E_p \log q_{\boldsymbol{W}}$ can be easily estimate by sample mean, while the others can't.

A:

$$\frac{\partial KL(q_{\boldsymbol{W}}||p)}{\partial q_{\boldsymbol{W}}(d')} = \frac{\sum_d q_{\boldsymbol{W}}(d') \log\left(q_{\boldsymbol{W}}(d')\right) - \sum_d q_{\boldsymbol{W}}(d') \log\left(p(d')\right)}{\partial q_{\boldsymbol{W}}(d')}$$

$$= \frac{\sum_d q_{\boldsymbol{W}}(d^{'}) \log\left(q_{\boldsymbol{W}}(d^{'})\right)}{\partial q_{\boldsymbol{W}}(d^{'})} - \frac{\sum_d q_{\boldsymbol{W}}(d^{'}) \log\left(p(d^{'})\right)}{\partial q_{\boldsymbol{W}}(d^{'})}$$
$$= \log\left(q_{\boldsymbol{W}}(d^{'})\right) + 1 - \log\left(p(d^{'})\right)$$

Similarly,

$$\frac{\partial KL(p||q_{\boldsymbol{W}})}{\partial q_{\boldsymbol{W}}(d^{'})} = \frac{\partial \sum_d p(d^{'}) \log\left(p(d^{'})\right) - \sum_d p(d^{'}) \log\left(q_{\boldsymbol{W}}(d^{'})\right)}{\partial q_{\boldsymbol{W}}(d^{'})}$$
$$= \frac{\partial \sum_d p(d^{'}) \log\left(p(d^{'})\right)}{\partial q_{\boldsymbol{W}}(d^{'})} - \frac{\sum_d p(d^{'}) \log\left(q_{\boldsymbol{W}}(d^{'})\right)}{\partial q_{\boldsymbol{W}}(d^{'})}$$
$$= 0 - \frac{p(d^{'})}{q_{\boldsymbol{W}}(d^{'})}$$
$$= -\frac{p(d^{'})}{q_{\boldsymbol{W}}(d^{'})}$$

4. In this problem, you are provided an opportunity to perform a hands-on calculation of the SVM loss and softmax loss we learned in class.
We define a linear classifier:
$$f(\boldsymbol{x}, \boldsymbol{W}) = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$$
and provide a data sample:
$$\boldsymbol{x}_i = \begin{bmatrix} -15 \\ 22 \\ -44 \\ 56 \end{bmatrix}, \; y_i = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$
Assume that the weights of our model are given by
$$\boldsymbol{W} = \begin{bmatrix} 0.01, & -0.05, & 0.1, & 0.05 \\ 0.7, & 0.2, & 0.05, & 0.16 \\ 0.0, & -0.45, & -0.2, & 0.03 \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} 0.0 \\ 0.2 \\ -0.3 \end{bmatrix}.$$

(a) Please calculate the following loss values for this sample:
   i. SVM loss (hinge loss)
   ii. Softmax loss (cross-entropy loss)

A: Given the data sample and the linear classifier, we have:

$$f(\boldsymbol{x}, \boldsymbol{W}) = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b} = \begin{bmatrix} 0.01 & -0.05 & 0.1 & 0.05 \\ 0.7 & 0.2 & 0.05 & 0.16 \\ 0 & -0.45 & -0.2 & 0.03 \end{bmatrix} \begin{bmatrix} -15 \\ 22 \\ -44 \\ 56 \end{bmatrix} + \begin{bmatrix} 0.0 \\ 0.2 \\ -0.3 \end{bmatrix} = \begin{bmatrix} -2.85 \\ 0.86 \\ 0.88 \end{bmatrix}$$

Let's compare with $y_i = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

SVM loss

$$L_I = \sum_{j \neq y_i} max(0, S_j - S_{yi} + 1) = max(0, -2.85 - 0.88 + 1) + max(0, 0.86 - 0.86 + 1) = 0.98$$

In order to get Softmax function $\frac{e^{S_k}}{\sum_j e^{S_j}}$, we firstly compute e^{Sk} by exponential the $\begin{bmatrix} -2.85 \\ 0.86 \\ 0.88 \end{bmatrix}$, which we can get $\begin{bmatrix} -0.06 \\ 2.26 \\ 2.41 \end{bmatrix}$. After that we normalize the term, we can get $\begin{bmatrix} 0.02 \\ 0.10 \\ 0.88 \end{bmatrix}$. Then the Softmax loss

$$L = -\log(0.883) = 0.18$$

(b) What is one advantage and one disadvantage of using the SVM loss? What about the softmax loss?

A: SVM loss: one advantage: when x gets larger, it won't saturate. It runs fast finding the minimum loss when x gets larger; one disadvantage: the function is not balanced. softmax loss: one advantage: it is balanced. one disadvantage: it will saturate when x gets larger or smaller.