Homework 1
600.482/682 Deep Learning
Fall 2019

Jason Zhang jzhan127

September 13, 2019

Due Fri 9/13 11:59pm.
Please type your answers inline of the LaTeX file
Submit PDF to Gradescope with entry code MKDPGK

1. In this exercise you are going to 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.

   In statistics, the logit of the probability is the logarithm of the corresponding odds, i.e. $\text{logit}(p) = \log(o)$.

   Answer:
   $o = \frac{p}{1-p}$

   (b) Given $\text{logit}(p) = x$, please derive the inverse function $\text{logit}^{-1}(x)$. Please express the result using x only.

   Answer: Let $y = logit(x) = log(\frac{x}{1-x})$. We swap x and y like so and solve for y again to get the inverse function: $x = log(\frac{y}{1-y})$.

$$
\begin{aligned}
x &= log(\frac{y}{1-y}) \\
e^x &= \frac{y}{1-y} \\
(e^x - e^x y) &= y && \text{Rearranging} \\
e^x &= y + e^x y && \text{Rearranging} \\
\frac{e^x}{1+e^x} &= y
\end{aligned}
$$

   Thus we get $\text{logit}^{-1}(x) = \frac{e^x}{1+e^x}$.

   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 $(0, 1)$.

   (c) Now we look into the saturation of the sigmoid function. Calculate the value of the sigmoid function $S(x)$ for $x = \pm 100, \pm 10$, and 0. Round the results to two decimal places.

   Answer: $S(x) = \frac{e^x}{1+e^x}$
   For $x = 100$:
   $S(x) = \frac{e^{100}}{1+e^{100}} = 0.99$
   For $x = -100$:

$S(x) = \frac{e^{-100}}{1+e^{-100}} = 0.00$
For $x = 10$:
$S(x) = \frac{e^{10}}{1+e^{10}} = 0.99$
For $x = -10$:
$S(x) = \frac{e^{-10}}{1+e^{-10}} = 0.00$
For $x = 0$:
$S(x) = \frac{e^{0}}{1+e^{0}} = 0.50$

(d) Calculate the derivatives of the sigmoid function $S'(x)$ and the value of $S'(x)$ for $x = \pm 100, \pm 10$, and 0. Round the results to two decimal places. **Answer:**

$$\begin{aligned} S'(x) &= \frac{(e^x)'(1+e^x) - e^x(1+e^x)'}{(1+e^x)^2} && \text{Using Product Rule} \\ &= \frac{e^x(1+e^x) - e^x e^x}{(1+e^x)^2} \\ &= \frac{e^x + e^{2x} - e^{2x}}{(1+e^x)^2} \\ &= \frac{e^x}{(1+e^x)^2} \end{aligned}$$

$S'(x) = \frac{e^x}{(1+e^x)^2}$
For $x = 100$:
$S'(x) = \frac{e^{100}}{(1+e^{100})^2} = 0.00$
For $x = -100$:
$S'(x) = \frac{e^{-100}}{(1+e^{-100})^2} = 0.00$
For $x = 10$:
$S'(x) = \frac{e^{10}}{(1+e^{10})^2} = 0.00$
For $x = -10$:
$S'(x) = \frac{e^{-10}}{(1+e^{-10})^2} = 0.00$
For $x = 0$:
$S'(x) = \frac{e^{0}}{(1+e^{0})^2} = 0.25$
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.

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 (multiple classes) problem.
Let's first define the notations:

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

(a) Please express the softmax loss of logistic regression, $L(\boldsymbol{x}, \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{y})$ using the above notations.
Answer: For some arbitrary input feature vector $\boldsymbol{x_i}$ and corresponding class label k:
$L_i = -log(Y = k | X = x_i) = -log(\frac{e^{s_k}}{\sum_j e^{s_j}})$ For $s_i = f(x_i, W)$.
Thus for our above problem:
$L(\boldsymbol{x}, \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{y}) = -log(\frac{e^{\boldsymbol{y}^\top (\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})}}{\sum_{j=1}^K e^{w_j{}^\top + b_j}})$

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

Answer: Let $P = \frac{e^{\boldsymbol{y}^\top(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b})}}{\sum_{j=1}^{K} e^{w_j{}^\top + b_j}}$.

$$\begin{aligned}\frac{\partial L}{\partial \boldsymbol{w}_k} &= \frac{\partial L}{\partial P}\frac{\partial P}{\partial \boldsymbol{w}_k}\\ &= -\frac{1}{P}\frac{\partial P}{\partial \boldsymbol{w}_k}\end{aligned}$$

We have two cases in which we need to consider for $\frac{\partial P}{\partial \boldsymbol{w}_k}$ which is when w_k corresponds to the weight vector in W that corresponds to the target label y_i and when w_k is does not correspond to the target label y_i. Note: $\boldsymbol{y}^\top W$ gives us the weight vector corresponding to the target label.

Let $\sum = \sum_{j=1}^{K} e^{w_j{}^\top + b_j}$

Case 1: $w_k = \boldsymbol{y}^\top \boldsymbol{W}$

$$\begin{aligned}\frac{\partial P}{\partial \boldsymbol{w}_k} &= \frac{(e^{\boldsymbol{y}^\top(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b})})'\sum - e^{\boldsymbol{y}^\top(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b})}\sum'}{\sum^2} && \text{Using quotient rule}\\ &= \frac{\boldsymbol{x}e^{\boldsymbol{y}^\top(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b})}\sum - e^{\boldsymbol{y}^\top(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b})}\boldsymbol{x}e^{\boldsymbol{w}_k^\top\boldsymbol{x}+\boldsymbol{b}_k}}{\sum^2}\\ &= \frac{\boldsymbol{x}e^{\boldsymbol{y}^\top(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b})}}{\sum}\frac{\sum - e^{\boldsymbol{w}_k^\top\boldsymbol{x}+\boldsymbol{b}_k}}{\sum}\end{aligned}$$

Now plugging it into $\frac{\partial L}{\partial \boldsymbol{w}_k}$ we get:

$$\begin{aligned}\frac{\partial L}{\partial \boldsymbol{w}_k} &= -\frac{\sum}{e^{\boldsymbol{y}^\top(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b})}}\frac{\boldsymbol{x}e^{\boldsymbol{y}^\top(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b})}}{\sum}\frac{\sum - e^{\boldsymbol{w}_k^\top\boldsymbol{x}+\boldsymbol{b}_k}}{\sum}\\ &= -\boldsymbol{x}\left(1 - \frac{e^{\boldsymbol{w}_k^\top\boldsymbol{x}+\boldsymbol{b}_k}}{\sum}\right)\end{aligned}$$

Case 2: $w_k \neq \boldsymbol{y}^\top \boldsymbol{W}$

$$\begin{aligned}\frac{\partial P}{\partial \boldsymbol{w}_k} &= \frac{(e^{\boldsymbol{y}^\top(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b})})'\sum - e^{\boldsymbol{y}^\top(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b})}\sum'}{\sum^2} && \text{Using quotient rule}\\ &= \frac{-e^{\boldsymbol{y}^\top(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b})}\boldsymbol{x}e^{\boldsymbol{w}_k^\top\boldsymbol{x}+\boldsymbol{b}_k}}{\sum^2}\\ &= -\frac{\boldsymbol{x}e^{\boldsymbol{y}^\top(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b})}}{\sum}\frac{e^{\boldsymbol{w}_k^\top\boldsymbol{x}+\boldsymbol{b}_k}}{\sum}\end{aligned}$$

Now plugging it into $\frac{\partial L}{\partial \boldsymbol{w}_k}$ we get:

$$\begin{aligned}\frac{\partial L}{\partial \boldsymbol{w}_k} &= -\frac{\sum}{e^{\boldsymbol{y}^\top(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b})}} - \frac{\boldsymbol{x}e^{\boldsymbol{y}^\top(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b})}}{\sum}\frac{e^{\boldsymbol{w}_k^\top\boldsymbol{x}+\boldsymbol{b}_k}}{\sum}\\ &= \boldsymbol{x}\left(\frac{e^{\boldsymbol{w}_k^\top\boldsymbol{x}+\boldsymbol{b}_k}}{\sum}\right)\end{aligned}$$

Thus we have the following piece-wise derivative:

$$\frac{\partial L}{\partial \boldsymbol{w}_k} = \begin{cases} -\boldsymbol{x}(1 - \frac{e^{\boldsymbol{w}_k^\top\boldsymbol{x}+\boldsymbol{b}_k}}{\sum}) & w_k = \boldsymbol{y}^\top\boldsymbol{W}\\ \boldsymbol{x}(\frac{e^{\boldsymbol{w}_k^\top\boldsymbol{x}+\boldsymbol{b}_k}}{\sum}) & w_k \neq \boldsymbol{y}^\top\boldsymbol{W}\end{cases}$$

3. In class, we briefly touch 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 determine by our training data, while the other is being generated as 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 that assigning these distributions to p and q will matter. Here, you will investigate this difference by calculating the gradient. 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 KL divergence is asymmetric using the following example. We define a discrete random variable X. Now consider the case that we have two sampling distributions $P(x)$ and $Q(x)$, which we present as two vectors that express the frequency of event x:

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

Please compute 1) the probability distribution, $p(x)$ and $q(x)$ (hint: calculate the normalization); and 2) both directions of KL divergence, $\mathbf{KL}(p||q)$ and $\mathbf{KL}(q||p)$.
Answer: To get the probability distributions we will calculate the normalization of each of the frequency distributions.
$\sum P(x) = 50, \sum Q(x) = 50$, divide each frequency distribution by the sums respectively.

$$p(x) = [\frac{1}{50}, \frac{6}{50}, \frac{12}{50}, \frac{5}{50}, \frac{2}{50}, \frac{8}{50}, \frac{12}{50}, \frac{4}{50}] = [0.02,\ 0.12,\ 0.24,\ 0.1,\ 0.04,\ 0.16,\ 0.24,\ 0.08]$$
$$q(x) = [\frac{1}{50}, \frac{3}{50}, \frac{6}{50}, \frac{8}{50}, \frac{15}{50}, \frac{10}{50}, \frac{5}{50}, \frac{2}{50}] = [0.02,\ 0.06,\ 0.12,\ 0.16,\ 0.3,\ 0.2,\ 0.1,\ 0.04]$$

Now we compute:
$\mathbf{KL}(p||q) = \sum_d p(d) \log\left(\frac{p(d)}{q(d)}\right) \approx 0.352$
$\mathbf{KL}(q||p) = \sum_d q(d) \log\left(\frac{q(d)}{p(d)}\right) \approx 0.484$

(b) Next, we try to optimize the weights $\boldsymbol{W}$ of a model in an attempt to minimize KL divergence. As a consequence, $q = q_{\boldsymbol{W}}$ now depends on the weights. Please express $\mathbf{KL}(q_{\boldsymbol{W}}||p)$ and $\mathbf{KL}(p||q_{\boldsymbol{W}})$ as optimization objective functions. Can you tell which direction is easier for computation? To find out, please look back at the original expression of $\mathbf{KL}(q_{\boldsymbol{W}}||p)$ and $\mathbf{KL}(p||q_{\boldsymbol{W}})$ and see which terms can be grouped to be a constant. This constant can be thus cancelled out when calculating the gradient. Then, please also 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 $q_{\boldsymbol{W}}$.
Answer:
Using $\mathbf{KL}(q_{\boldsymbol{W}}||p)$:

$$\begin{aligned}\mathbf{KL}(q_{\boldsymbol{W}}||p) &= \sum_d q_{\boldsymbol{W}}(d) log\left(\frac{q_{\boldsymbol{W}}(d)}{p(d)}\right) && \text{From the given equation for KL}\\ &= \sum_d q_{\boldsymbol{W}}(d)(log(q_{\boldsymbol{W}}(d)) - log(p(d))\\ &= \sum_d q_{\boldsymbol{W}}(d) log(q_{\boldsymbol{W}}(d)) - \sum_d q_{\boldsymbol{W}} log(p(d))\end{aligned}$$

We then use the substitution $p(d) = P(d)/\sigma_P$ where $P(d)$ is the true distribution and σ_P is the normalization constant associated with $P(d)$:
$\sum_d q_{\boldsymbol{W}}(d) log(q_{\boldsymbol{W}}(d)) - \sum_d q_{\boldsymbol{W}} log(p(d)) = \sum_d q_{\boldsymbol{W}}(d) log(q_{\boldsymbol{W}}(d)) - \sum_d q_{\boldsymbol{W}} log\left(\frac{P(d)}{\sigma_P}\right) =$
$\sum_d q_{\boldsymbol{W}}(d) log(q_{\boldsymbol{W}}(d)) - \sum_d q_{\boldsymbol{W}} log(P(d)) + \sum_d q_{\boldsymbol{W}} log(\sigma_P)$.
$\sum_d q_{\boldsymbol{W}} log(\sigma_P) = log(\sigma_p) \sum_d q_{\boldsymbol{W}} = log(\sigma_p)$. Substituting in this fact we get:
$\sum_d q_{\boldsymbol{W}}(d) log(q_{\boldsymbol{W}}(d)) - \sum_d q_{\boldsymbol{W}} log(P(d)) + log(\sigma_P)$.

Thus $\mathbf{KL}(q_{\boldsymbol{W}}||p) = \sum_d q_{\boldsymbol{W}}(d) log(q_{\boldsymbol{W}}(d)) - \sum_d q_{\boldsymbol{W}} log(P(d)) + log(\sigma_P)$.

Given that KL divergence(in information theory) will calculate the entropy introduced from using an encoding from a distribution that is not the true distribution, our optimization will attempt to minimize the KL divergence, which in turn will minimize the entropy.
Thus our optimization objective function will be:

$$\operatorname*{argmin}_{\boldsymbol{W}}(\mathbf{KL}(q_{\boldsymbol{W}}||p)) = \operatorname*{argmin}_{\boldsymbol{W}}(\sum_d q_{\boldsymbol{W}}(d)log(q_{\boldsymbol{W}}(d)) - \sum_d q_{\boldsymbol{W}}(d)log(P(d)) + log(\sigma_P))$$

Using $\mathbf{KL}(p||q_{\boldsymbol{W}})$:

$$\begin{aligned}\mathbf{KL}(p||q_{\boldsymbol{W}}) &= \sum_d p(d)log\left(\frac{p(d)}{q_{\boldsymbol{W}}(d)}\right) && \text{From the given equation for KL}\\ &= \sum_d p(d)(log(p(d)) - log(q_{\boldsymbol{W}}(d)) \\ &= \sum_d p(d)log(p(d)) - \sum_d p(d)log(q_{\boldsymbol{W}}(d))\end{aligned}$$

We then use the substitution $p(d) = P(d)/\sigma_P$ where $P(d)$ is the true distribution and $q_{\boldsymbol{W}}(d) = Q(d)/\sigma_Q$ where $Q(d)$ is the approximated distribution and σ_P, σ_Q are the normalization constants associated with $P(d)$ and $Q(d)$:
$\sum_d p(d)log(p(d)) - \sum_d p(d)log(q_{\boldsymbol{W}}(d)) = \sum_d p(d)log(p(d)) - \sum_d \frac{P(D)}{\sigma_P}log\left(\frac{Q(d)}{\sigma_Q}\right) = \sum_d p(d)log(p(d)) - \sum_d \frac{P(D)}{\sigma_P}log(Q(d)) + \sum_d \frac{P(D)}{\sigma_P}log(\sigma_Q))$.
$\sum_d \frac{P(D)}{\sigma_P}log(\sigma_Q)) = log(\sigma_Q)\sum_d \frac{P(D)}{\sigma_P} = log(\sigma_Q)$. (Probability distributions sum to 1)
Substituting in this fact we get:
$\sum_d p(d)log(p(d)) - \sum_d \frac{P(D)}{\sigma_P}log(Q(d)) + log(\sigma_Q)$.

Thus $\mathbf{KL}(p||q_{\boldsymbol{W}}) = \sum_d p(d)log(p(d)) - \sum_d \frac{P(D)}{\sigma_P}log(Q(d)) + log(\sigma_Q)$.

Given that KL divergence(in information theory) will calculate the entropy introduced from using an encoding from a distribution that is not the true distribution, our optimization will attempt to minimize the KL divergence, which in turn will minimize the entropy.
Thus our optimization objective function will be:

$$\operatorname*{argmin}_{\boldsymbol{W}}(\mathbf{KL}(p||q_{\boldsymbol{W}})) = \operatorname*{argmin}_{\boldsymbol{W}}(\sum_d p(d)log(p(d)) - \sum_d \frac{P(D)}{\sigma_P}log(Q(d)) + log(\sigma_Q))$$

Looking at both objective functions, we can simply further and discard the constant terms since taking the derivative with respect to $q_{\boldsymbol{W}(d)}$ will simply negate their contribution anyway.

Thus we get the following objective functions:

$$\operatorname*{argmin}_{\boldsymbol{W}}(\mathbf{KL}(q_{\boldsymbol{W}}||p)) = \operatorname*{argmin}_{\boldsymbol{W}}(\sum_d (q_{\boldsymbol{W}}(d)log(q_{\boldsymbol{W}}(d)) - q_{\boldsymbol{W}}(d)log(P(d))))$$

$$\operatorname*{argmin}_{\boldsymbol{W}}(\mathbf{KL}(p||q_{\boldsymbol{W}})) = \operatorname*{argmin}_{\boldsymbol{W}}(-\sum_d \frac{P(D)}{\sigma_P}log(Q(d)) + log(\sigma_Q))$$

Looking at both of these equations, the term that is most troubling is σ_P. This is because we will not necessarily deal with distributions that have easily calculable norms. Thus the easier direction will be to use $\mathbf{KL}(q_{\boldsymbol{W}}||p)$.

Calculating gradient of $\mathbf{KL}(q_{\boldsymbol{W}}||p)$ w.r.t $q_{\boldsymbol{W}(d)}$:

$$\begin{aligned}
\nabla_{q_{\boldsymbol{W}(d)}\mathbf{KL}(q_{\boldsymbol{W}}||p)} &= \sum_d (q_{\boldsymbol{W}}(d)log(q_{\boldsymbol{W}}(d)))' - (q_{\boldsymbol{W}}(d)log(P(d)))' \\
&= \sum_d q'_{\boldsymbol{W}}(d)log(q_{\boldsymbol{W}}(d)) + q_{\boldsymbol{W}}(d)\frac{1}{q_{\boldsymbol{W}}(d)}q'_{\boldsymbol{W}} - q'_{\boldsymbol{W}}log(P(d)) \\
&= \sum_d q'_{\boldsymbol{W}}(d)(log(q_{\boldsymbol{W}}(d)) - log(P(d)) + 1)
\end{aligned}$$

Calculating gradient of $\mathbf{KL}(p||q_{\boldsymbol{W}})$ w.r.t $q_{\boldsymbol{W}(d)}$:

$$\begin{aligned}
\nabla_{q_{\boldsymbol{W}(d)}\mathbf{KL}(p||q_{\boldsymbol{W}})} &= -\sum_d (\frac{P(D)}{\sigma_P}log(Q(d))' + log(\sigma_Q))' \\
&= -\sum_d \frac{P(D)}{\sigma_P}\frac{1}{(Q(d))}Q'(d) + \frac{1}{\sigma_Q}(\sigma_Q)'
\end{aligned}$$

4. In this problem, you are provided an opportunity to perform 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 are given 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

$$\boldsymbol{W} = \begin{bmatrix} 0.01, \; -0.05, \; 0.1, \; 0.05 \\ 0.7, \; 0.2, \; 0.05, \; 0.16 \\ 0.0, \; -0.45, \; -0.2, \; 0.03 \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} 0.0 \\ 0.2 \\ -0.3 \end{bmatrix}.$$

Please calculate 1) SVM loss (hinge loss) and 2) softmax loss (cross-entropy loss) of this sample. Use the natural log.

Answer: We can start by computing:

$$f(x_i, \boldsymbol{W}) = \boldsymbol{W}x_i + \boldsymbol{b} = \begin{bmatrix} 0.01, \; -0.05, \; 0.1, \; 0.05 \\ 0.7, \; 0.2, \; 0.05, \; 0.16 \\ 0.0, \; -0.45, \; -0.2, \; 0.03 \end{bmatrix} \begin{bmatrix} -15 \\ 22 \\ -44 \\ 56 \end{bmatrix} + \begin{bmatrix} 0.0 \\ 0.2 \\ -0.3 \end{bmatrix} = \begin{bmatrix} -2.85 \\ 0.86 \\ 0.28 \end{bmatrix}$$

We let $s = f(x_i, \boldsymbol{W})$.
From y_i we know that we will be looking at the 3rd label which is the 3rd element in s. Thus $s_{y_i} = 0.28$

Calculating SVM loss:
$L_i = \sum_{j \neq y_i} max(0, s_j - s_{y_i} + 1)$.
Now we solve for $L_i = \sum_{j \neq y_i} max(0, s_j - s_{y_i} + 1) = max(0, -2.85 - 0.28 + 1) + max(0, 0.86 - 0.28 + 1) = 0 + 1.58 = \mathbf{1.58}$

Calculating Softmax loss:
$L_i = -log(P(Y = y_i|X = x_i)) = -log(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}) = -log(\frac{e^{0.28}}{e^{-2.85} + e^{0.86} + e^{0.28}}) = \mathbf{1.04019}$.