Question Paper

Exam Date & Time: 24-May-2023 (10:00 AM - 01:00 PM)

MANIPAL ACADEMY OF HIGHER EDUCATION

Manipal School of Information Sciences (MSIS), Manipal
Second Semester Master of Engineering - ME (Artificial Intelligence and Machine Learning) Degree Examination - May 2023

Deep Learning [AML 5202]

Marks: 100 **Duration: 180 mins.**

Wednesday, May 24, 2023

Answer all the questions.

1) [10 points] [TLO 1.1, CO 1] Suppose we have 10^3 samples corresponding to 3 output labels and that each sample is a $3 \times 28 \times 28$ color image. If we apply the softmax algorithm to this data, what are the dimensions of the following quantities assuming that the bias-trick pre-processing has been performed: (10)

- Data matrix;
- Weight matrix;
- Probability matrix;
- Adjusted probability matrix;
- Total average data loss;
- Regularization loss;
- Gradient of total loss w.r.t. the weight matrix?

2) [10 points] [TLO 1.2, CO 1] Consider the following dataset for binary classification: (10)

$$x^{(1)} = \begin{bmatrix} 2 \\ 5 \end{bmatrix}, \; x^{(2)} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \; x^{(3)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$
$$y^{(1)} = 1, \; y^{(2)} = 0, \; y^{(3)} = 0.$$

Calculate the SVM loss using bias-trick for the following weights and bias values:

$$W = \begin{bmatrix} 0.1 & -0.4 \\ 0.5 & -0.3 \end{bmatrix}, \; b = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

3) [10 points] [TLO 3.2, CO 3] Consider the same dataset, weights, and bias values from the previous problem. Calculate the Softmax loss using a regularization strength of 0.3. (10)

4) [10 points] [TLO 2.2, CO 2] Using the same setup from the previous problem, perform one step of gradient descent with learning rate $= 10^{-2}$. Round all numbers to 2 decimal places. (10)

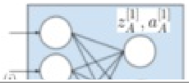
5) (10)

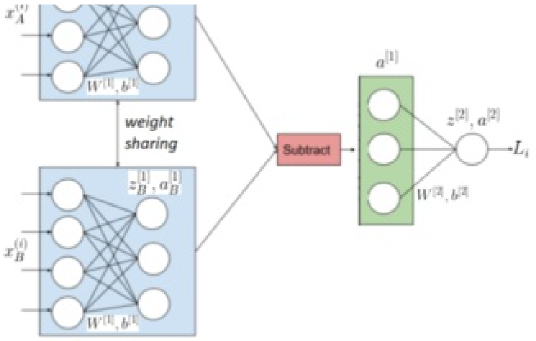
[10 points] [TLO 1.2, CO 1] For a particular binary classification task, your friend modifies the logistic regression loss function as follows:

$$L = \boldsymbol{\alpha} \left[-y \log \left(a^{[1]} \right) \right] - \boldsymbol{\beta} \left[(1-y) \log \left(1 - a^{[1]} \right) \right],$$

where the model has one input layer and one output layer that is sigmoid-activated. Note that your friend introduces the (unknown) coefficients $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ that have to be fine-tuned like any hyperparameter. What kind of a binary classification task might your friend be solving? Give a simple problem setup and reasonable values for the two parameters.

6) [10 points] [TLO 3.1, CO 2] A Siamese neural network consists of twin networks which accepts distinct inputs but share the same weights. The outputs of the twin networks are usually joined later on by one or more layers. The following image shows such a network with a pair of distinct input samples $x_A^{(i)}$ and $x_B^{(i)}$ both of which are of size 4×1: (10)





Note the following:

- $x_A^{(i)}$ and $x_B^{(i)}$ together constitute the ith input sample for the network;
- the weights for the first hidden layer is the same ($W^{[1]}$) as shown in the blue regions;
- *ReLU* activation is used in the hidden layer and sigmoid activation is used in the output layer.

Fill in the question marks below for the forward propagation for the ith sample leading to the loss L_i:

$$\begin{aligned}
? &= W^{[1]}x_A^{(i)} + b^{[1]}\\
a_A^{[1]} &= ?,\\
z^{[2]} &= W^{[1]} \times ? + ?,\\
? &= ReLU\left(z^{[2]}\right)\\
? &= a_A^{[1]} - a_B^{[1]},\\
z^{[2]} &= ? \times a^{[1]} + b^{[2]},\\
a^{[2]} &= ?,\\
L_i &= -\left(y^{(i)} \times ? + (1-?) \times \log\left(1 - a^{[2]}\right)\right).
\end{aligned}$$

7) [10 points] [TLO 3.1, CO 2] For the Siamese neural network in the previous problem, draw a tree diagram starting with L_i on the top and ending with the weights $W^{[1]}$ and bias $b^{[1]}$ by clearly showing the intermediate variables. (10)

8) (10)

[10 points] [TLO 3.2, CO 3] Using the tree diagram for the Siamese neural network in the previous problem, we want to compute the gradient $\nabla_{W^{[1]}}(L_i)$.

- The gradient $\nabla_{W^{[1]}}(L_i)$ ends up being a combination of quantities as follows:

$$\left[\underbrace{T}_{3\times4\times3\text{-tensor}} \times \underbrace{D_A}_{3\times3\text{-matrix}} \times \underbrace{I}_{3\times3\text{-matrix}} - \underbrace{T}_{3\times4\times3\text{-tensor}} \times \underbrace{D_B}_{3\times3\text{-matrix}} \times \underbrace{I}_{3\times3\text{-matrix}}\right] \times \underbrace{A}_{3\times1\text{-matrix}} \times \text{constant}.$$

Match the following with the quantities in the expression above:

$$(1)\,\nabla_{W^{[1]}}\left(z_A^{[1]}\right)\;(2)\,\nabla_{W^{[1]}}\left(z_B^{[1]}\right)\;(3)\,\nabla_{a^{[1]}}\left(z^{[2]}\right)\;(4)\,\nabla_{z^{[2]}}\left(L_i\right)\;(5)\,\nabla_{a_A^{[1]}}\left(a^{[1]}\right)\;(6)\,\nabla_{a_B^{[1]}}\left(a^{[1]}\right)\;(7)\,\nabla_{z_A^{[1]}}\left(a_A^{[1]}\right)\;(8)\,\nabla_{z_B^{[1]}}\left(a_B^{[1]}\right).$$

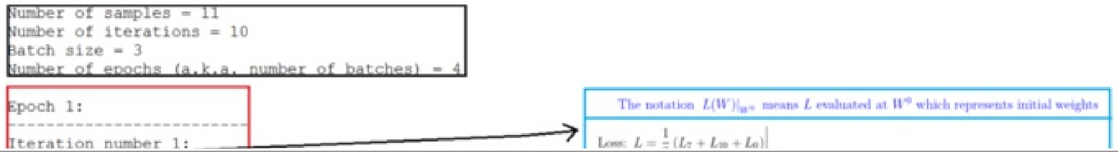
- Compute all the gradients *except* $\nabla_{W^{[1]}}\left(z_A^{[1]}\right)$ and $\nabla_{W^{[1]}}\left(z_B^{[1]}\right)$, and use fact that

$$\nabla_{W^{[1]}}\left(z^{[1]}_{A \text{ or } B}\right) \times \text{vector} = \nabla_{W^{[1]}}\left(W^{[1]}x^{(i)}_{A \text{ or } B} + b^{[1]}\right) \times \text{vector} = \text{vector} \times \left(x^{(i)}_{A \text{ or } B}\right)^{\mathrm{T}}$$

to fill-in the missing entries below:

$$\nabla_{W^{[1]}}(L_i) = \begin{bmatrix} ? & ? & ? \\ 0 & ? & ? \\ ? & ? & I\left(z_{A_3}^{[1]}\right) - I\left(z_{B_3}^{[1]}\right) \end{bmatrix} (?)^{\mathrm{T}}\left(a^{[2]} - ?\right)\left(? - \left(x_B^{(i)}\right)^{\mathrm{T}}\right).$$

9) [10 points] [TLO 4.1, CO 3] In the schematic given below for batch processing, fill in the question marks in the red boxes: (10)

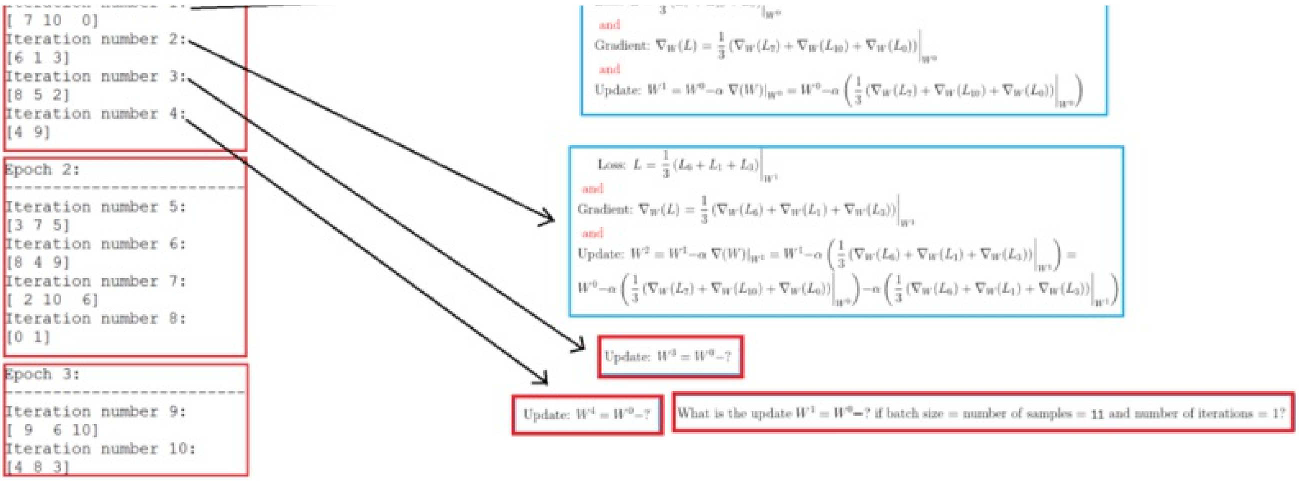


```
Number of samples = 11
Number of iterations = 10
Batch size = 3
Number of epochs (a.k.a. number of batches) = 4
```

```
Epoch 1:
-----------------------------
Iteration number 1:
```

The notation $L(W)|_{W^0}$ means L evaluated at W^0 which represents initial weights

Loss: $L = \frac{1}{3}(L_7 + L_{10} + L_6)$



```
[ 7 10  0]
Iteration number 2:
[6 1 3]
Iteration number 3:
[8 5 2]
Iteration number 4:
[4 9]
Epoch 2:
--------------------------
Iteration number 5:
[3 7 5]
Iteration number 6:
[8 4 9]
Iteration number 7:
[ 2 10  6]
Iteration number 8:
[0 1]
Epoch 3:
--------------------------
Iteration number 9:
[ 9  6 10]
Iteration number 10:
[4 8 3]
```

and

Gradient: $\nabla_W(L) = \frac{1}{3}\left(\nabla_W(L_7) + \nabla_W(L_{10}) + \nabla_W(L_0)\right)\Big|_{W^0}$

and

Update: $W^1 = W^0 - \alpha\ \nabla(W)|_{W^0} = W^0 - \alpha\left(\frac{1}{3}\left(\nabla_W(L_7) + \nabla_W(L_{10}) + \nabla_W(L_0)\right)\Big|_{W^0}\right)$

Loss: $L = \frac{1}{3}\left(L_6 + L_1 + L_3\right)\Big|_{W^1}$

and

Gradient: $\nabla_W(L) = \frac{1}{3}\left(\nabla_W(L_6) + \nabla_W(L_1) + \nabla_W(L_3)\right)\Big|_{W^1}$

and

Update: $W^2 = W^1 - \alpha\ \nabla(W)|_{W^1} = W^1 - \alpha\left(\frac{1}{3}\left(\nabla_W(L_6) + \nabla_W(L_1) + \nabla_W(L_3)\right)\Big|_{W^1}\right) =$
$W^0 - \alpha\left(\frac{1}{3}\left(\nabla_W(L_7) + \nabla_W(L_{10}) + \nabla_W(L_0)\right)\Big|_{W^0}\right) - \alpha\left(\frac{1}{3}\left(\nabla_W(L_6) + \nabla_W(L_1) + \nabla_W(L_3)\right)\Big|_{W^1}\right)$

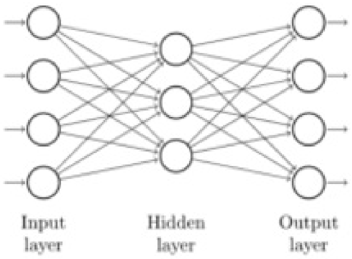
Update: $W^3 = W^0 - ?$

Update: $W^4 = W^0 - ?$

What is the update $W^1 = W^0 -?$ if batch size = number of samples = 11 and number of iterations = 1?

10) (10)

[10 points] [TLO 4.2, CO 4] An autoencoder is a neural network designed to learn feature representations in an unsupervised manner. Unlike a standard multi-layer network, an autoencoder has the same number of nodes in its output layer as its input layer. An autoencoder is trained to reconstruct its own input, that is, to minimize the reconstruction error. A simple autoencoder architecture is shown below:



- Assuming that the input sample is $x^{(i)}$, write down the forward propagation equations starting with $z^{[1]}$ and ending with the loss L_i. Since the loss measures how well the sample $x^{(i)}$ can be reconstructed, find a simple expression for it involving $a^{[2]}$ and $x^{(i)}$.
- The gradient descent update for $W^{[1]}$ can be written as:

$$\begin{aligned} W^{[1]} &= W^{[1]} + \alpha\left(-\nabla_{W^{[1]}}\left(L_i\right)\right) \\ &= W^{[1]} - \alpha\left[D_1 \times \left(W^{[2]}\right)^{\mathrm{T}} \times D_2 \times 2\left(a^{[2]} - x^{(i)}\right)\right]\left(x^{(i)}\right)^{\mathrm{T}}, \end{aligned}$$

$$\text{where } D_1 = \begin{bmatrix} \sigma\left(z_1^{[1]}\right)\left(1-\sigma\left(z_1^{[1]}\right)\right) & 0 & 0 \\ 0 & \sigma\left(z_2^{[1]}\right)\left(1-\sigma\left(z_2^{[1]}\right)\right) & 0 \\ 0 & 0 & \sigma\left(z_3^{[1]}\right)\left(1-\sigma\left(z_3^{[1]}\right)\right) \end{bmatrix},$$

$$\text{and } D_2 = \begin{bmatrix} \sigma\left(z_1^{[2]}\right)\left(1-\sigma\left(z_1^{[2]}\right)\right) & 0 & 0 & 0 \\ 0 & \sigma\left(z_2^{[2]}\right)\left(1-\sigma\left(z_2^{[2]}\right)\right) & 0 & 0 \\ 0 & 0 & \sigma\left(z_3^{[2]}\right)\left(1-\sigma\left(z_3^{[2]}\right)\right) & 0 \\ 0 & 0 & 0 & \sigma\left(z_4^{[2]}\right)\left(1-\sigma\left(z_4^{[2]}\right)\right) \end{bmatrix}.$$

Using this, answer when can slow learning happen? That is, when does $W^{[1]}$ get updated very slowly?

-----End-----