National University of Computer and Emerging Sciences

Generative AI (AI-4009)
Date: February 26, 2024
Course Instructor
Akhtar Jamil

Sessional-I
Total Time: 1 Hours
Total Marks: 100
Total Questions: 04

Semester: SP-2024
Campus: Islamabad

Student Name	Roll No	Section	Signature

Q. No 1. Write short answers to the following questions [2 x 10=20]

I. In the DiagonalBiLSTM, a 2x1 convolution kernel is used. Is it useful to increase its size like 3x1 or 5x1? Explain your reason.

In the case of DiagonalBiLSTM, the receptive field is already designed to capture global dependencies across the data, increasing the kernel size beyond 2x1 does not significantly contribute to broadening the receptive field. Therefore, larger kernels might not provide additional benefits and could unnecessarily increase computational complexity without enhancing model performance.

II. What are Skip connections? Why they are useful? How can we select the number of skip connections?

Skip connections are used in neural network where outputs from earlier layers are fed directly to later layers, bypassing one or more intermediate layers. They are useful for preventing the vanishing gradient problem, allowing for deeper networks, and preserving information across layers. The number of skip connections is usually determined based on network depth, task complexity, and empirical testing, balancing performance improvements with computational cost.

III. Consider an RGB image of size 3x3. Each channel is shown below. Apply a 1x1 convolution where each weight is equal to 0.5. Select a suitable number of channels for the kernel.

2	8	6
4	2	8
6	4	2

R

4	2	8
6	4	2
8	6	4

G

6	4	2
8	6	4
2	8	6

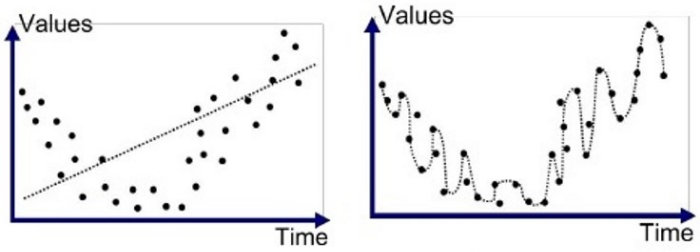
B

6	7	8
9	6	7
8	9	6

IV. Why do we need a validation dataset?

A validation dataset is needed to tune model parameters, prevent overfitting, and provide an unbiased evaluation of model performance, separate from the final test data evaluation.

V. The following diagram shows the results of fitting two different models to the same data. Identify the issues that occurred in both and propose your solutions.



1. Left diagram shows that the model is underfitting. The model is too simple. Solution: Use a more complex model.

2. Right diagram shows that model is Overfitting. The model is too complex and captures noise. Solution: Simplify the model and use techniques like regularization.

VI. What is the difference between discriminative and generative models? Write at least two examples for each.

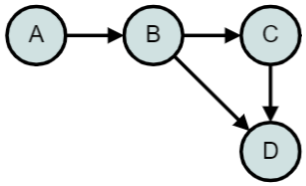
Discriminative models learn the boundary between classes; they directly map inputs to outputs. Examples: Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs).

Generative models learn the underlying distribution of classes; they can generate new data points. Examples: Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs).

VII. What is Batch Normalization? How does it impact the learning process?

Batch Normalization is a technique in deep learning that normalizes the inputs to a layer for each mini-batch, stabilizing the mean and variance of those inputs. This normalization accelerates the training process by allowing higher learning rates and reducing the problem known as internal covariate shift, where the distribution of layer inputs changes during training, making the training process less stable. As a result, networks train faster and more effectively. Additionally, batch normalization has a regularization effect, which can help reduce overfitting, often diminishing the need for other regularization techniques.

VIII. Consider the following graph. Is this a valid Bayesian Network? Write the formula to calculate the joint probability of D given all its dependencies.



$$P(D \mid A,B,C) = P(D \mid B,C) \cdot P(C \mid B) \cdot P(B \mid A) \cdot P(A)$$

IX. What are autoregressive models with finite memory? What are their drawbacks?

Autoregressive models with finite memory predict future values in a series based on a fixed, limited number of prior inputs. This limitation, known as "finite memory," means they might not capture long-term dependencies if relevant historical context exceeds their memory span. This limited context can lead to undesirable results which is a significant drawback of these models.

X. How to introduce long-range dependency in the autoregressive models?

To introduce long-range dependency in autoregressive models, one can incorporate techniques like Recurrent Neural Networks (RNNs) for introducing long-range dependency. RNNs are designed to handle sequential data and can theoretically maintain information from any arbitrary point in the past due to their recurrent structure. These approaches allow the model to remember and utilize information from further back in the series, helping to capture the underlying dynamics over longer time periods and improving the model's ability to forecast based on long-range historical data.

Q. No 2. [2+2+6]

Consider the following transition probabilities derived for a 3-bit image. Based on this table answer the next both questions a) and b).

	P(next=0)	P(next=1)	P(next=2)	P(next=3)	P(next=4)	P(next=5)	P(next=6)
P(current=0)	0.1429	0.1224	0.1020	0.0816	0.0612	0.0408	0.0204
P(current=1)	0.4286	0.1429	0.1224	0.1020	0.0816	0.0612	0.0408
P(current=2)	0.0204	0.4286	0.1429	0.1224	0.1020	0.0816	0.0612
P(current=3)	0.0408	0.0204	0.4286	0.1429	0.1224	0.1020	0.0816
P(current=4)	0.0612	0.0408	0.0204	0.4286	0.1429	0.1224	0.1020
P(current=5)	0.0816	0.0612	0.0408	0.0204	0.4286	0.1429	0.1224
P(current=6)	0.1020	0.0816	0.0612	0.0408	0.0204	0.4286	0.1429
P(current=7)	0.1224	0.1020	0.0816	0.0612	0.0408	0.0204	0.4286

a) If the current pixel sequence is […,2,3,4]. Calculate the joint probability of the next two pixel values being 5 and 6 respectively. Assume that you know the probability of the current sequence up to pixel value 4.

Here are the steps with their respective probabilities:

$$P(\text{next} = 5|\text{current} = 4) = 0.1224$$
$$P(\text{next} = 6|\text{current} = 5) = 0.1224$$

The joint probability is calculated as:

$P(\text{next} = 5,6 \mid \text{current} = 4) = P(\text{next} = 5| \text{current} = 4) \times P(\text{next} = 6|\text{current} = 5)$
$= 0.1224 \times 0.1224$
$= 0.01498$
Thus, the joint probability of transitioning from a pixel value of 4 to 5 and then to 6, based on the given transition probabilities, is approximately 0.01498.

b) If the last observed pixel value is 5, what is the most likely value of the next pixel?

$P(current = 5)$ then the column with the highest probability value is 4.

c) Consider a gradient vector $\mathbf{g} = [4, 3, 2]$ and a threshold value $\theta = 5$. Perform gradient clipping. Show all the steps.

Calculate the norm of the gradient vector:

$$||g|| = \sqrt{g_1^2 + g_2^2 + g_3^2}$$

$$||g|| = \sqrt{g_1^2 + g_2^2 + g_3^2}]$$

$$||g|| = \sqrt{4^2 + 3^2 + 2^2}$$

$$||g|| = 5.385$$

As $||g|| > \theta$ i.e. , 5.385 > 5 so, we clip the gradient as:

$$g = g \times \frac{\theta}{||g||}$$

$$g = 4\,x\,\frac{5}{5.385},\ 3\,x\,\frac{5}{5.385},\ 2\,x\,\frac{5}{5.385}$$

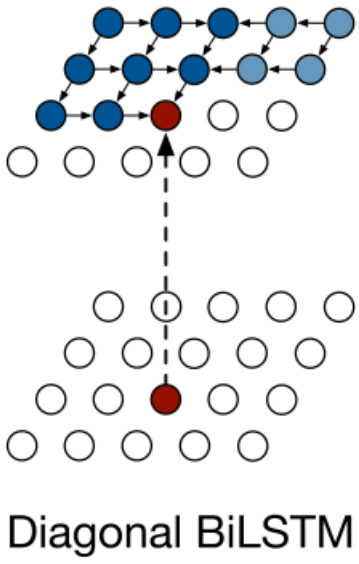
$$g = [3.71, 2.79, 1.86]$$

Q. No 3. [5+5]

a. Explain with the help of a diagram working of the Diagonal BiLSTM.

- First skew the input map into a space that makes it easy to apply convolutions along diagonals. The skewing operation offsets each row of the input map by one position with respect to the previous
- For each of the two directions, the input-to-state component is simply a 1×1 convolution K is that contributes to the four gates in the LSTM core

The state-to-state recurrent component is then computed with a column-wise convolution Kss that has a kernel of size 2 × 1.
The step takes the previous hidden and cell states, combines the con- tribution of the input-to-state component and produces the next hidden and cell states



Diagonal BiLSTM

b. Explain with the help of diagram working of the RowLSTM.

The Row LSTM is a unidirectional layer that processes the image row by row from top to bottom computing features for a whole row at once; the computation is per- formed with a one-dimensional convolution. An LSTM layer has an input-to-state component and a recurrent state-to-state component that together determine the four gates inside the LSTM core. To enhance parallelization in the Row LSTM the input-to-state component is first computed for the entire two-dimensional input map; for this a k × 1 convolution is used to follow the row-wise orientation of the LSTM itself.

To compute one step of the state-to-state component of the LSTM layer, one is given the previous hidden and cell states $\mathbf{h}_{i-1}$ and $\mathbf{c}_{i-1}$, each of size $h \times n \times 1$. The new hidden and cell states $\mathbf{h}_i$, $\mathbf{c}_i$ are obtained as follows:

$$
\begin{aligned}
[\mathbf{o}_i, \mathbf{f}_i, \mathbf{i}_i, \mathbf{g}_i] &= \sigma(\mathbf{K}^{ss} \circledast \mathbf{h}_{i-1} + \mathbf{K}^{is} \circledast \mathbf{x}_i) \\
\mathbf{c}_i &= \mathbf{f}_i \odot \mathbf{c}_{i-1} + \mathbf{i}_i \odot \mathbf{g}_i \qquad (3) \\
\mathbf{h}_i &= \mathbf{o}_i \odot \tanh(\mathbf{c}_i)
\end{aligned}
$$

Q. No 4. [10]

Consider an LSTM unit. The current input and is represented as x_0 and corresponding weights and biases are given for all the gates. The symbols are as follows: Forget gate (f), input gate(i), cell state (c) and output gate (o). The calculation for the cell are done according to the formulas shown. For x_0 the h_0 is 0. (Note: make any assumption if you think something is missing).

$x_0 = \begin{vmatrix} 1 \\ 2 \end{vmatrix}$

$W_{xc} = \begin{vmatrix} 0.45 \\ 0.25 \end{vmatrix}$ $W_{hc} = |0.15|$ $b_c = |0.20|$

$W_{xi} = \begin{vmatrix} 0.95 \\ 0.80 \end{vmatrix}$ $W_{hi} = |0.80|$ $b_i = |0.65|$

$W_{xf} = \begin{vmatrix} 0.70 \\ 0.45 \end{vmatrix}$ $W_{hf} = |0.10|$ $b_f = |0.15|$

$W_{xo} = \begin{vmatrix} 0.60 \\ 0.40 \end{vmatrix}$ $W_{ho} = |0.25|$ $b_o = |0.10|$

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$h_t = o_t * \tanh\left(C_t\right)$$

$$f_0 = \sigma\,(W_{xf} \,.\, x_0 + W_{hf} \,.\, h_{-1} + b_f)$$

$$f_0 = \sigma([0.70 \quad 0.45] \begin{bmatrix}1\\2\end{bmatrix} + [0.10] \times [0] + [0.15])$$

$$f_0 = 0.85195$$

$$\tilde{C}_0 = \tanh(W_{xc} \,.\, x_0 + W_{hc} \,.\, h_{-1} + b_c)$$

$$\tilde{C}_0 = \tanh([0.45 \quad 0.25] \begin{bmatrix}1\\2\end{bmatrix} + [0.15] \times [0] + [0.2])$$

$$\tilde{C}_0 = 0.81775$$

$$i_0 = \sigma\,(W_{xi} \,.\, x_0 + W_{hi} \,.\, h_{-1} + b_i)$$

$$i_0 = \sigma([0.95 \quad 0.80] \begin{bmatrix}1\\2\end{bmatrix} + [0.80] \times [0] + [0.65])$$

$$i_0 = 0.96083$$

$$o_0 = \sigma\,(W_{xo} \,.\, x_0 + W_{ho} \,.\, h_{-1} + b_o)$$

$$o_0 = \sigma([0.60 \quad 0.40] \begin{bmatrix}1\\2\end{bmatrix} + [0.25] \times [0] + [0.10])$$

$$o_0 = 0.81757$$

$$C_0 = \tilde{C}_0 \circ i_0 + f_0 \circ C_{-1}$$

$$C_0 = 0.81775 \times 0.96083 + 0.85195 \times 0$$

$$C_0 = 0.78572$$

$$h_0 = \tanh(C_0) \circ o_0$$

$$h_0 = \tanh(0.78572) \times 0.81757$$

$$h_0 = 0.53631$$