COMP 6721 - Artificial Intelligence
Introduction to Deep Learning

Solutions

Question 1 Assume that the 3×3 matrix below represents a gray scale image. Further assume that we will use X to train an autoencoder.

$$X = \begin{array}{|c|c|c|}\hline 0.2 & 0.3 & 0.2 \\ \hline 0.4 & 0.1 & 0.3 \\ \hline 0.1 & 0.9 & 0.5 \\ \hline \end{array}$$

(a) What will be the input and the output of an autoencoder that processes X.

Both the input and the output of the autoencoder will be X.

X is also known as the ground-truth of the network.

(b) What will be the size of the input and the output layers of such an autoencoder?

The size of the image is 3×3, so the input will be a vector of size 10 (9 for the input image + 1 for the bias).

The output of the network will be a vector of size 9 which is exactly the size of input image, since we want to reproduce the input image.

(c) What would be a suitable size for the number of hidden units?

Typically the number of hidden units should be smaller than the number of input units, since we aim to find an interesting representation of the input.

(d) What would happen if the size of the hidden layer is equal to the size of the input? How would you achieve the desired feature in this case?

If the size of the hidden layer is equal to the size of the input then it cannot learn a compressed and suitable representation of the input, and it will only copy the input to the output. More specifically the training result will be good, however the learned representation will not be useful for any other tasks.

In this case, a sparse autoencoder can be useful. In a sparse autoencoder, each hidden unit can be deactivated.

(e) Assume that we use an autoencoder with the following hyperparameters: the activation function is *sigmoid*, the loss/error function is the mean squared error (MSE), the hidden layer has a size of 5, and the learning rate η is 1. Perform a single iteration of forward

pass and backward pass through the autoencoder. You can assume an input value of 1 for the biases, and all the weights (including the biases) are initialized to 0.5.

The weight matrix from the input to the hidden layer is a matrix of size (5×9), the input layer is (1×9), and the bias from the input to the hidden layer is (1×5). The weight matrix from the hidden to the output layer is a matrix of size (9×5), and the bias of the output layer is (1×9).

The input vector is:

$$X = [0.2, 0.3, 0.2, 0.4, 0.1, 0.3, 0.1, 0.9, 0.5]$$

Forward pass

For each node in the hidden layer, we have to compute the pre-activation function (the *net*), and then apply the activation function (*sigmoid*). We can compute this for all nodes in the form of matrix multiplication.

(a) **Pre-activation (*net*):**

The pre-activation step is when we multiply the input and weights and add the result to the bias.

$$A = XW_{ih}^T + b_{ih} \tag{1}$$

where here A is a matrix of size (1×5).

$$A = [2.00, 2.00, 2.00, 2.00, 2.00]$$

(b) **Activation:**

Now to compute the result for h, the sigmoid function is applied to the pre-activation result (eq. **??**).

$$h = \sigma(A) = \sigma(XW_{ih}^T + b_{ih}) \tag{2}$$

where h is a matrix of size (1×5) and $\sigma(x) = \frac{1}{1+e^{-x}}$:

$$h = [0.88, 0.88, 0.88, 0.88, 0.88]$$

(c) **Output:**

To compute the output, O, the result of the hidden layer (eq. **??**) should be multiplied with the weights of the output layer, W_{ho}, then we apply sigmoid.

$$O = \sigma(hW_{ho}^T + b_{ho}) \tag{3}$$

where O is a matrix of size (1×9):

$$O = [0.937, 0.937, 0.937, 0.937, 0.937, 0.937, 0.937, 0.937, 0.937]$$

Backward Pass

First we calculate the error of each neuron.

(a) δ_O

$$\delta_O = (1 - O) \times O \times (O - T) \tag{4}$$

$$\delta_O = [0.043, 0.038, 0.043, 0.032, 0.049, 0.038, 0.049, 0.002, 0.026]$$

remember that because this is an autoencoder, we have:

$$T = X = [0.2, 0.3, 0.2, 0.4, 0.1, 0.3, 0.1, 0.9, 0.5]$$

(b) δ_{hidden}

$$\delta_h = h \times (1 - h) \times \sum_{k \in output} w_{hk}\delta_k \tag{5}$$

$$\delta_h = [0.017, 0.017, 0.017, 0.017, 0.017]$$

Now we compute the delta for the weights.
Recall that the learning rate η is assumed to be 1.

(a) ΔW_{ho}

$$\Delta W_{ho} = -\eta \times \delta_O \times h \tag{6}$$

The result is a matrix of size (9×5).

$$\Delta W_{ho} = \begin{bmatrix} -0.038, -0.038, -0.038, -0.038, -0.038 \\ -0.033, -0.033, -0.033, -0.033, -0.033 \\ -0.038, -0.038, -0.038, -0.038, -0.038 \\ -0.028, -0.028, -0.028, -0.028, -0.028 \\ -0.043, -0.043, -0.043, -0.043, -0.043 \\ -0.033, -0.033, -0.033, -0.033, -0.033 \\ -0.043, -0.043, -0.043, -0.043, -0.043 \\ -0.002, -0.002, -0.002, -0.002, -0.002 \\ -0.023, -0.023, -0.023, -0.023, -0.023 \end{bmatrix}$$

(b) Δb_{ho}

$$\Delta b_{ho} = -\eta \times \delta_O \times 1 \tag{7}$$

The result is a matrix of size (1×9).

$$\Delta b_{ho} = [-0.043, -0.038, -0.043, -0.032, -0.049, -0.038, -0.049, -0.002, -0.026]$$

(c) ΔW_{ih}

$$\Delta W_{ih} = -\eta \times \delta_h \times X \tag{8}$$

The result is a matrix of size (5×9).

$$\Delta W_{ih} =$$
$$\begin{bmatrix} -0.003, -0.005, -0.003, -0.007, -0.002, -0.005, -0.002, -0.015, -0.008 \\ -0.003, -0.005, -0.003, -0.007, -0.002, -0.005, -0.002, -0.015, -0.008 \\ -0.003, -0.005, -0.003, -0.007, -0.002, -0.005, -0.002, -0.015, -0.008 \\ -0.003, -0.005, -0.003, -0.007, -0.002, -0.005, -0.002, -0.015, -0.008 \\ -0.003, -0.005, -0.003, -0.007, -0.002, -0.005, -0.002, -0.015, -0.008 \end{bmatrix}$$

(d) Δb_{ih}

$$\Delta b_{ih} = -\eta \times \delta_h \times 1 \tag{9}$$

The result is a matrix of size (1×5).

$$\Delta b_{ih} = [-0.017, -0.017, -0.017, -0.017, -0.017]$$

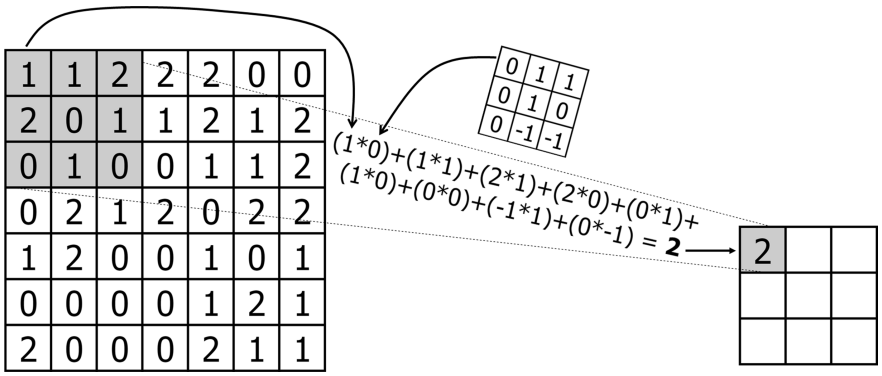
Question 2 Assume the following matrix that represents an image. This image will be fed to a convolutional neural network.

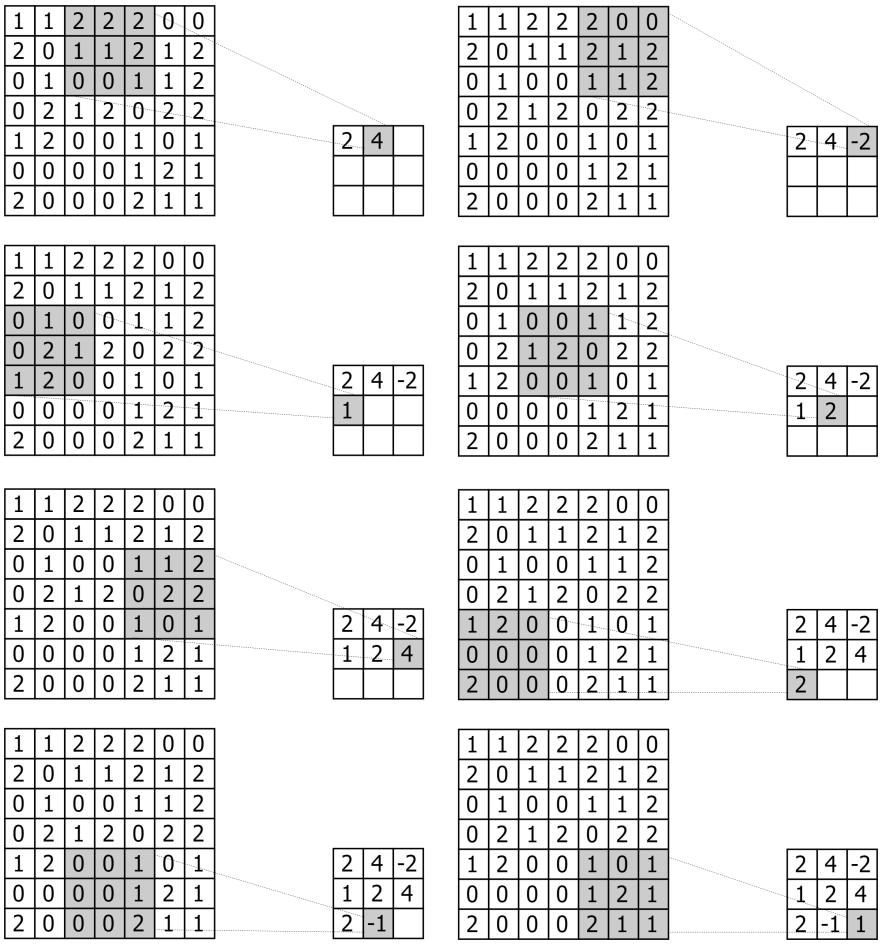
1	1	2	2	2	0	0
2	0	1	1	2	1	2
0	1	0	0	1	1	2
0	2	1	2	0	2	2
1	2	0	0	1	0	1
0	0	0	0	1	2	1
2	0	0	0	2	1	1

(a) Assume that we use the following convolution filter with a stride of 2.

0	1	1
0	1	0
0	-1	-1

What will be the size of the activation map? What will be the activation map?
The size of the activation map will be 3*3, and the output of the activation map will be calculated as follows:



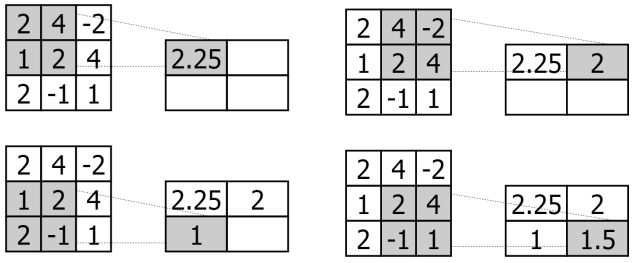


Based on the computations shown above, the activation map is as follows:

2	4	-2
1	2	4
2	-1	1

(b) What will be the output of the pooling layer with, a size of 2*2 and a stride of 1, on the activation map of part (a) if we use the following strategies?

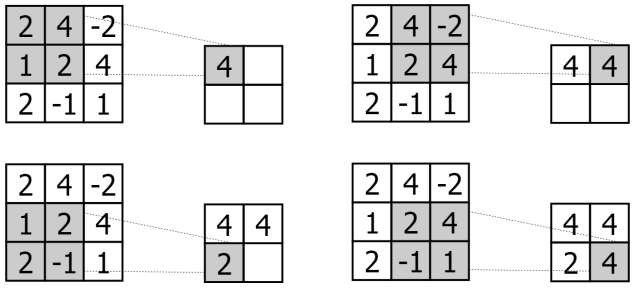
i. Average pooling



Based on the computation steps shown above, the output of the average pooling layer is as follows:

2.25	2
1	1.5

ii. Max pooling



Based on the computation steps shown above, the output of the max pooling layer is as follows:

4	4
2	4

Question 3 Consider the following sentence:

"the cat drinks the milk"

We will use this sentence to train a CBOW Word2Vec model. Assume that:

- you want to produce word embeddings of dimension 2,
- you use a context window of size 2 (1 word before and 1 word after the target word), and
- your vocabulary only contains the words in the sentence above

(a) Using only the sentence above, how many instances will be generated as training set?

3 instances

Instance	Context Word -1	Context Word +1	To Predict
1	the	drinks	cat
2	cat	the	drinks
3	drinks	milk	the

(b) List the one-hot vectors that correspond to each word in the vocabulary. (Assume alphabetical ordering)

Word	Hot Vector			
cat	1	0	0	0
drinks	0	1	0	0
milk	0	0	1	0
the	0	0	0	1

(c) List the one-hot vectors that correspond to each training instance in the input layer.

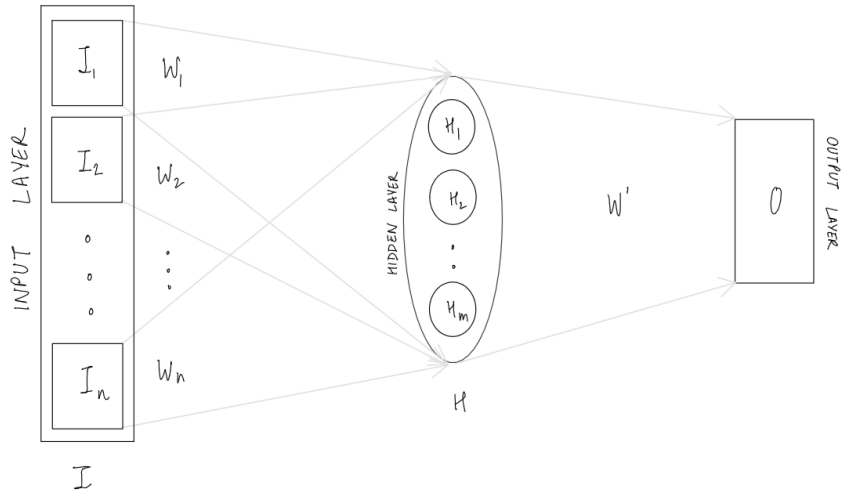
Instance	Context	Word	Hot Vector			
1	Context Word -1	the	0	0	0	1
	Context Word +1	drinks	0	1	0	0
2	Context Word -1	cat	1	0	0	0
	Context Word +1	the	0	0	0	1
3	Context Word -1	drinks	0	1	0	0
	Context Word +1	milk	0	0	1	0

(d) How many nodes will the hidden layer contain?
Number of nodes in hidden layer = 2

(e) What is the target hot vector for each training instance?

Instance	To Predict	Hot Vector			
1	cat	1	0	0	0
2	drinks	0	1	0	0
3	the	0	0	0	1

(f) Assume that the Word2Vec model is trained with the standard network depicted below:



i. What will be the values of n and m?
$n = 2$, $m = 2$

ii. What will be the sizes of I, W_i (for each $1 \leq i \leq n$), W' and O?
Size of $I_1 = I_2 = 1 \times 4$, $I = 2 \times 4$
Size of $W_1 = W_2 = 4 \times 2$
Size of $W' = 2 \times 4$
Size of $O = 1 \times 4$

(g) Assume that we have these weight vectors:

$$W = \begin{bmatrix} 2 & 6 \\ 4 & 3 \\ 1 & 4 \\ 5 & 2 \end{bmatrix}$$

$$W' = \begin{bmatrix} 6 & 2 & 8 & 3 \\ 4 & 5 & 9 & 7 \end{bmatrix}$$

To compute the final probabilities at the output layer, we use the softmax function as shown in class. Recall that for a given vector of size k, the softmax function is defined as:

$$p_i = \frac{e^{x_i}}{\sum_{i=1}^{k} e^{x_i}}, \text{ where } 1 \leq i \leq k$$

i. Trace the first feed forward pass in the network and show the values propagated all the way to the output layer.

Instance 1 -

Instance	Context	Word	Hot Vector				To Predict	Target			
1	Context Word -1 Context Word +1	the drinks	0 0	0 1	0 0	1 0	cat	1	0	0	0

Calculate the output of each hidden node for each context word

$$H = I \times W = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 2 & 6 \\ 4 & 3 \\ 1 & 4 \\ 5 & 2 \end{bmatrix} = \begin{bmatrix} 5 & 2 \\ 4 & 3 \end{bmatrix}$$

Take the average

$$H_{AVG} = \begin{bmatrix} 4.5 & 2.5 \end{bmatrix}$$

Calculate output

$$O = H_{AVG} \times W' = \begin{bmatrix} 4.5 & 2.5 \end{bmatrix} \times \begin{bmatrix} 6 & 2 & 8 & 3 \\ 4 & 5 & 9 & 7 \end{bmatrix} = \begin{bmatrix} 37 & 21.5 & 58.5 & 31 \end{bmatrix}$$

Calculate softmax probabilities for the output

$$softmax(O) = softmax(\begin{bmatrix} 37 & 21.5 & 58.5 & 31 \end{bmatrix})$$
$$= \begin{bmatrix} 4.6 \times 10^{-10} & 8.53 \times 10^{-17} & 0.99 & 1.14 \times 10^{-12} \end{bmatrix}$$

ii. What is the error after the first pass?

Calculate error

$$E = O - T = \begin{bmatrix} 4.6 \times 10^{-10} & 8.53 \times 10^{-17} & 0.99 & 1.14 \times 10^{-12} \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$
$$= \begin{bmatrix} \approx -1 & 8.53 \times 10^{-17} & 0.99 & 1.14 \times 10^{-12} \end{bmatrix}$$