COMP 6721 Applied Artificial Intelligence (Fall 2021)

Worksheet #6: Introduction to Deep Learning

Autoencoder. Assume that the 3×3 matrix below represents a gray scale image:

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

Your job is to use X to train an *autoencoder*. So, the input to our network is going to be X. But what is the expected output (labels)? .....

Define the size of the input layer: ..... and output layer: ..... of your autoencoder.

How many hidden neurons would you use? (Assuming we use a single hidden layer): .....
(Hint: there is no single correct answer, but you can define a sensible range.)

Autoencoder Activation. Assume that we use an autoencoder with the following hyperparameters: the activation function is *sigmoid* and the hidden layer has a size of 5. Perform a single forward 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. Note, our input vector corresponding to the image above is

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

1. First, compute the pre-activation function (the *net*), by multiplying the input and weights, plus the bias. Rather than listing every weight as on the previous worksheet, we now use matrix notation:[1]

$$A = X \cdot W_{ih}^T + b_{ih} \tag{1}$$

   the result is a matrix of size (1×5), $A = [$ ..... $]$

2. Now to compute the result for h, the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ is applied to the pre-activation result (eq. 1):

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

   where h is a matrix of size (1×5), $h = [$ ..... $]$

3. To compute the output O, the result of the hidden layer (eq. 2) should be multiplied with the weights of the output layer, W_{ho}, then we apply sigmoid again:

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

   where O is a matrix of size (1×9), $O = [$ ..... $]$

[1] See https://medium.com/from-the-scratch/deep-learning-deep-guide-for-all-your-matrix-dimensions-and-calculations-415012de1568 for a review of matrix calculations for neural networks

CNN Activation Map. Assume the following matrix that represents an image. This image will be fed to a convolutional neural network (CNN).

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

Assume that we use the following convolution filter with a stride of 2 (no padding):

0	1	1
0	1	0
0	-1	-1

What will be the size of the activation map? .....

What will be the activation map?

Pooling Layer. 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 the question above, if we use the following strategies:

1. Average pooling:

2. Max pooling: