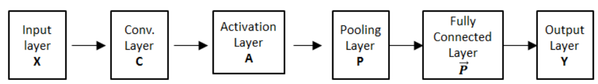
ECSE 4850/6850
Introduction to Deep Learning
Assignment #4
Due date: 2 pm, March 16th

The structure of a Convolutional Neural Network for binary classification $y \in \{0, 1\}$ is given below.



The input X a 7x7 image. The convolution layer C is produced by convolving X with a 3x3 filter W^x with stride 1, plus bias matrix W_0^x. The activation layer A is produced by applying the ReLU activation function to C. A max pooling of 3x3 with a stride 2 is then applied to C to produce the pooling layer P. A fully connected vector layer $\vec{P}$ is then produced by concatenating rows of P. The output layer consists of one node y. It is produced by $y = \sigma((W^o)^t \vec{P} + W_0^o))$, where W^o and W_0^o are output layer weight matrix and bias vector respectively.

Given an input X

$$X = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

, the desired output value of $t = 1$, the initial weight and bias matrix for the convolution layer as

$$W^x = \begin{bmatrix} 0.2 & 0.1 & 0.3 \\ 0.1 & 0.2 & 0.4 \\ 0.3 & 0.4 & 0.1 \end{bmatrix} \tag{1}$$

$$W_0^x = 0 \tag{2}$$

and the fully connected layer bias matrix and bias vector as,

$$W^o = \begin{bmatrix} 0.2 \\ 0.1 \\ 0.3 \\ 0.4 \end{bmatrix} \tag{3}$$

$W_0^o = -0.2$.

Perform the following tasks:

- Determine the dimension for each layer
- Perform forward propagation layer by layer to compute the values for each layer, the estimated output value $\hat{y}$, and the gradient of the output $\Delta\hat{y}$, given y=1, using negative log conditional likelihood (cross-entropy) loss function.
- Given $\Delta\hat{y}$, perform back-propagation to obtain the weight matrix gradient (ΔW^o) and bias vector gradient (ΔW_0^o) for the output layer and the weight matrix gradient (ΔW^x) and bias matrix gradient (ΔW_0^x) respectively for the convolution layer.
- Update the weights for the convolution and the output layers with their estimated gradients, using a learning rate of 0.5, and then compute the new output value $\hat{y}$ using the updated weight matrices. Verify that the new $\hat{y}$ has reduced the output loss function, as compared to the previous $\hat{y}$.