CSE 676: Deep Learning, Section B **Spring 2025**

Assignment 1 - Part V

Name	Student ID
Anchal Daga	50609480
Sharanya Nallapeddi	50593866

1 Part V.I) CNN

1.1 Task 1

We know,

The formula for calculating the output dimensions of a convolution layer,

$$Output = \frac{(I - K + 2P)}{S} + 1$$

where:

$$\begin{array}{lcl} \text{Input size, I} & = & 32 \\ \text{Kernel size, } K & = & 5 \\ \text{Padding, } P & = & 0 \\ \text{Stride, } S & = & 1 \end{array}$$

Therefore,

$$O = \frac{(32 - 5 + 2(0))}{1} + 1$$

$$= \frac{27}{1} + 1$$

$$= 28$$

Thus,

The output size after first convolution layer is $10 \times 28 \times 28$

1.2 Task 2

The number of learnable parameters in a convolution layer is calculated by:

$$\text{Total parameters} = (K \times K \times C) \times F + B$$

where:

$$\begin{array}{lcl} \text{Kernel Size, K} & = & 5 \\ \text{Number of channels, C} & = & 3 \\ \text{Number of filters, F} & = & 10 \\ \text{Bias, B} & = & 10 \end{array}$$

Thus,

$$(5 \times 5 \times 3) \times 10 + 10$$

$$= (75 \times 10) + 10$$

$$= 750 + 10$$

$$= 760$$

Therefore, the total number of learnable parameters is 760.

1.3 Task 3

We know,

The formula for calculating the output dimensions of a convolution layer,

$$Output = \frac{(I - K + 2P)}{S} + 1$$

where:

$$\begin{array}{lcl} \text{Input size, I} & = & 32 \\ \text{Kernel size, } K & = & 5 \\ \text{Padding, } P & = & 1 \\ \text{Stride, } S & = & 1 \end{array}$$

Therefore,

$$O = \frac{(32 - 5 + 2(1))}{1} + 1$$

$$= \frac{29}{1} + 1$$

$$= 30$$

Thus,

The output size after first convolution layer is $10 \times 30 \times 30$

1.4 Task 4

If the input was a greyscale image, the image channel would reduce from 3 (RGB) to 1 (Grey).

The number of learnable parameters in a convolution layer would be calculated as:

$$\text{Total parameters} = (K \times K \times C) \times F + B$$

where:

$$\begin{array}{lcl} \text{Kernel Size, K} & = & 5 \\ \text{Number of channels, C} & = & 1 \\ \text{Number of filters, F} & = & 10 \\ \text{Bias, B} & = & 10 \end{array}$$

Thus,

$$(5 \times 5 \times 1) \times 10 + 10$$

$$= (25 \times 10) + 10$$

$$= 250 + 10$$

$$= 260$$

Therefore, the total number of learnable parameters is 260.

1.5 Task 5

Given the tasks involving multi-class classification with 5 output classes, the most appropriate activation function is `softmax`.

Softmax:

- This function transforms raw logits into probabilities.
- It allows each output to be interpretable as a probability for a class

Why other activation functions would not work:

- `ReLU` is used in hidden layers but does not normalize outputs to probabilities.
- `Sigmoid` is better suited for binary classification
- `Tanh` does not provide valid probability distributions.

Thus, `softmax` is the best choice for multi-class classification.

1.6 Task 6

The softmax function is given by:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

if we add a constant c to all inputs,

$$\sigma(z_i + c) = \frac{e^{z_i+c}}{\sum_j e^{z_j+c}}$$

and simplify the expression as follows,

$$= \frac{e^c e^{z_i}}{e^c \sum_j e^{z_j}}$$

Since e^c is common in both numerator and denominator, it can be canceled out,

$$= \frac{e^{z_i}}{\sum_j e^{z_j}} = \sigma(z_i)$$

Therefore, the soft max function remains unchanged under constant shifts.

The significance of shift invariance is as follows:

- It helps in preventing numerical problems like overflow caused by excessively high values.
- It helps to ensure that models remain adaptable to changes or transformations in the input data.

2 Part V.II) LSTM Derivation

The standard LSTM equations are given as follows:

2.1 Gate Equations

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ g_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} \left(W \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} \right) \tag{1}$$

2.2 Cell State Update

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{2}$$

2.3 Hidden State Update

$$h_t = o_t \odot \tanh(c_t) \tag{3}$$

2.4 Gradient Computation

Given a loss function $\mathcal{L}$ at time step t, we compute the gradients of $\mathcal{L}$ with respect to the LSTM variables.

2.5 Gradient with respect to c_t

$$\frac{\partial \mathcal{L}}{\partial c_t} = \frac{\partial \mathcal{L}}{\partial h_t} \odot o_t \odot (1 - \tanh^2(c_t)) \tag{4}$$

2.6 Gradient with respect to o_t

$$\frac{\partial \mathcal{L}}{\partial o_t} = \frac{\partial \mathcal{L}}{\partial h_t} \odot \tanh(c_t) \tag{5}$$

2.7 Gradient with respect to i_t

$$\frac{\partial \mathcal{L}}{\partial i_t} = \frac{\partial \mathcal{L}}{\partial c_t} \odot g_t \tag{6}$$

2.8 Gradient with respect to f_t

$$\frac{\partial \mathcal{L}}{\partial f_t} = \frac{\partial \mathcal{L}}{\partial c_t} \odot c_{t-1} \tag{7}$$

2.9 Gradient with respect to g_t

$$\frac{\partial \mathcal{L}}{\partial g_t} = \frac{\partial \mathcal{L}}{\partial c_t} \odot i_t \tag{8}$$

2.10 Gradient with respect to h_{t-1}

Since the gates depend on the previous hidden state h_{t-1}:

$$\frac{\partial \mathcal{L}}{\partial h_{t-1}} = W_h^T \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial i_t} \\ \frac{\partial \mathcal{L}}{\partial f_t} \\ \frac{\partial \mathcal{L}}{\partial o_t} \\ \frac{\partial \mathcal{L}}{\partial g_t} \end{bmatrix} \tag{9}$$

3 References:-

3.1 Lecture Notes, CNN Architecture I, Alina Vereshchaka

3.2 Lecture Notes, RNN & Long Short-Term Memory (LSTM), Alina Vereshchaka

3.3 Video Explanation, Piazza, Alina Vereshchaka