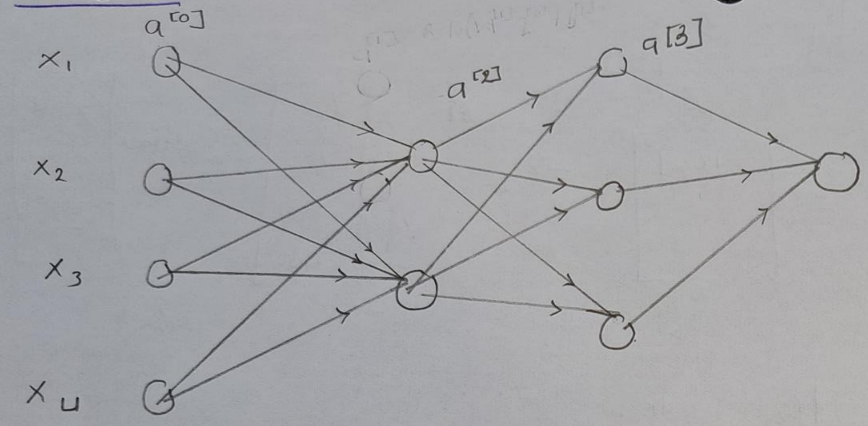
Qn 1



1 B) $Z^{[1]} = W^{[1]} a^{[0]} + b^{[1]}$

$$= \begin{bmatrix} W_1^{[1]} & W_1^{[1]} & W_1^{[1]} & W_1^{[1]} \\ W_2^{[1]} & W_2^{[1]} & W_2^{[1]} & W_1^{[1]} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \end{bmatrix} = \begin{bmatrix} Z_1^{[1]} \\ Z_2^{[1]} \end{bmatrix}$$

$$[2 \times 4] \times [4 \times 1] + [2 \times 1]$$

$$\begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \end{bmatrix} = Relu \begin{pmatrix} Z_1^{[1]} \\ Z_2^{[1]} \end{pmatrix}$$ to the next layer.

$$\begin{bmatrix} a_1^2 \\ a_2^2 \\ a_3^2 \end{bmatrix} = Relu \left[\underset{3 \times 2}{\begin{pmatrix} W_1^{[2]} & W_1^{[2]} \\ W_2^{[2]} & W_2^{[2]} \\ W_3^{[2]} & W_3^{[2]} \end{pmatrix}} \underset{2 \times 1}{\begin{pmatrix} a_1^{[1]} \\ a_2^{[1]} \end{pmatrix}} + \underset{3 \times 1}{\begin{pmatrix} b_1^{[2]} \\ b_2^{[2]} \\ b_3^{[2]} \end{pmatrix}} \right]$$ to next layer.

In the last layer $W^{[3]} \cdot a^{[2]} + b_1^{[3]}$

$$\begin{bmatrix} W_1^{[3]} & W_2^{[3]} & W_3^{[3]} \end{bmatrix} \begin{bmatrix} a_1^{[2]} \\ a_2^{[2]} \\ a_3^{[2]} \end{bmatrix} \; b_1^{[3]}$$

$$= Z_1^{[3]} \rightarrow Relu(Z_1^{[3]}) = a_1^{[3]}$$

$$a_1^{[3]} = \hat{Y} = f(x)$$

$f(x)$ is a functiont that tove in $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_n \end{bmatrix}$ and give out Y

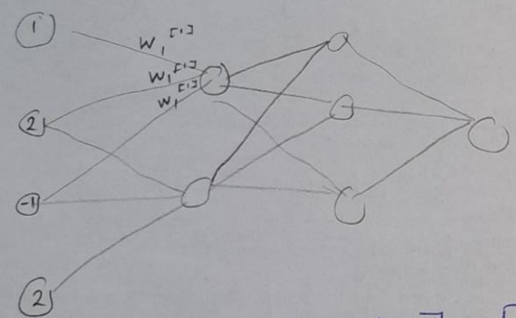
1 c) Now let $x_1 = 1,\ x_2 = 0,\ x_3 = -1,\ x_4 = 2.$

and $W_1^{[1]} = 1 \quad W_2^{[1]} = 1$

$W_1^{[2]} = 0 \quad W_1^{[2]} = -1 \quad W_3^{[2]} = 1$

$W_{[1]}^{3} = 2$

let $b_{[1]}^{[1]} = b_2^{[1]} = b_1^{[2]} = b_2^{[2]} \dots b_1^{[3]} = 0.$



$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ -1 \\ 2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$$

$\text{Relu}\begin{pmatrix} 4 \\ 4 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \end{pmatrix}$ to next layer

$$\begin{pmatrix} 0 & 0 \\ -1 & -1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 4 \\ 4 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ -8 \\ 8 \end{pmatrix},$$

$\text{Relu}\begin{pmatrix} 0 \\ -8 \\ 8 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 8 \end{pmatrix}$ to next layer

$$\begin{bmatrix} 2 & 2 & 2 \end{bmatrix} \begin{pmatrix} 0 \\ 0 \\ 8 \end{pmatrix} + (0) = 16$$

$\text{Relu}(16) = 16.$

d) The number of parameters = All the weights and all the biases to be learned.

weights $= 7$

bias $= 7$ =

Total parameter $= 14.$

Question 2

A) We Comopute Softmox $(z+c)$ and Compare with Softmox z

$$S_k(x+c) = \frac{e^{x_k+c}}{\sum_{k=0}^{K} e^{x_k+c}} = \frac{e^{x_k} \cdot e^{c}}{\sum_{k=0}^{K} e^{x_k} \cdot e^{c}} = e^{x}$$

$$= \frac{e^{x_k}}{\sum_{k=0}^{K} e^{x_k}} = S_k(x).$$

b) $$S_k(cx) = \frac{e^{cx_k}}{\sum_{k=1}^{K} e^{cx_k}} = \frac{e^{cx_k}}{\sum_{k=0}^{\infty} e^{cx_k} * e^{c(c'-1)x_k}}.$$

where I let c' be any Constant factor Introduced by multiplication, $c' > 0$.

Since $e^{c(c'-1)}$ is a Constond that appor on both numerator and denominator it may Cancel out but presence of c in e^{c} Change the magnitude of the terms resulting to different out put from the original.

$\Rightarrow$ Softmox is not mutiplication invariant.

c) $P_k(x) = \dfrac{x_k}{\sum_{i=1}^{k} x_i}$: $k = 1, 2, 3 \ldots k$.

To show that the population probability function is equvolent to multiplying all element by c, we consider a scaled vector cx.

we show. $P_k(cx) = P_k(x) \quad \forall k = 1, 2 \ldots k$.

$$P_k(cx) = \frac{(cx)_k}{\sum_{i=1}^{k} (cx)_i} = \frac{c x_k}{c \sum_{i=1}^{k} x_i}$$

$$= \frac{x_k}{\sum_{i=1}^{k} x_i}$$

$$= P_k(x)$$

hence shown.

$P_k(x)$ is multiplication invariant.

H/W 4 Qn 3

A) Drawing.

Layer 1.

$32 \times 32 \times 1$

Filtering 1

3, (5×5) kenels F1 F=1 F2 F3

$f_h \times f_w \times f_d$.

$5 \times 5 \times 1$

$f^{[1]} = 5$

$p^{[1]} = 0$

3 filters

$28 \times 28 \times 3$

↑H ↑W ↑chonels

$$n_H^{[1]} = \frac{n_H^{[0]} + 2p^{[1]} - f^{[1]}}{S^{[1]}} + 1$$

$$n_H = \frac{32 + 0 - 5}{1} + 1 = 28$$

Max pooling 1

1, (3×3)

3 3

3×3

The Same pool applied to each Channel Idependently

$26 \times 26 \times 3$

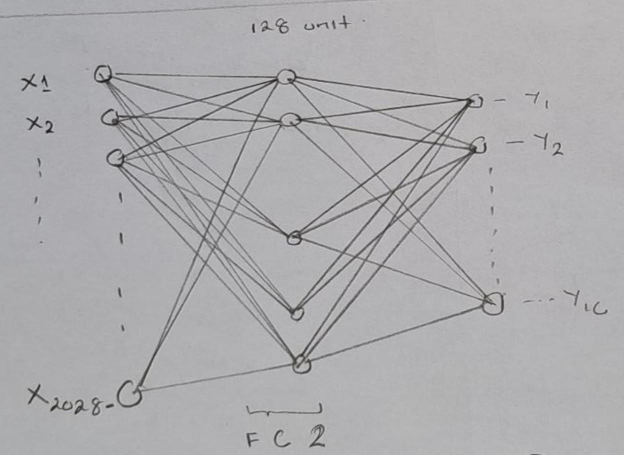
$$n_H = \frac{28 + 0 - 3}{2} + 1 = 26$$

Flattening

Flottening

$26 \times 26 \times 3 = 2028$

hence 2028×1 vector



Now in the Fully Connectd layer 2 (FC 2)

$W^{[2]} * a^{[1]} + b^{[2]} = Z^{[2]}$

$$\begin{bmatrix} 128 \times 2028 \end{bmatrix} \begin{bmatrix} 2028 \times 1 \end{bmatrix} + \begin{bmatrix} 128 \times 1 \end{bmatrix} = \begin{bmatrix} 128 \times 1 \end{bmatrix}$$

to get $a^{[2]}$ we pass through activation

$$a^{[2]} = \text{Relu}(Z^{[2]}) = \begin{bmatrix} a_1^{[2]} \\ \vdots \\ a_{128}^{[2]} \end{bmatrix}.$$

Porometers

$128 \times 2028 + 128$.

Qn 3

Now in the fully Connect layer 3

$$W^{[3]} \times a^{[2]} + b^{[3]} = Z^{[3]}$$

$$\begin{bmatrix} 10\times128 \end{bmatrix} \begin{bmatrix} 128\times1 \end{bmatrix} + \begin{bmatrix} \cancel{128\times1} \\ 10\times1 \end{bmatrix} = \begin{bmatrix} 10\times1 \end{bmatrix} \Rightarrow a^{[3]} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_{10} \end{bmatrix} = \text{Softmax} \begin{bmatrix} Z_1^{[3]} \\ Z_2^{[3]} \\ \vdots \\ Z_{10}^{3} \end{bmatrix}$$

B) Number of porometer in the mode.

In the Convolution layer

(filter porometer + bias) × No of filters.

$(5 \times 5 \times 1 + 1) \times 3 = 75$

In the ~~Conv~~ FC1 layer.

$128 \times 2028 + 128 = 259712$

In the FC2 layer

$10 \times 128 + 10 = 1290$

Total $\underline{\overline{261077}}$.

3C)A Convolutional Neural Network (CNN) can indeed be thought of as a special kind of feedforward neural network, where the individual pixel values of the input image are the inputs to the network. The key difference lies in the structure and constraints of the hidden layers.

In a CNN, the hidden layers are composed of convolutional layers and pooling layers, which impose certain constraints on the weights and connections in the network:

1. **Local Connectivity**: In a fully connected layer, each neuron is connected to every neuron in the previous layer. However, in a convolutional layer, each neuron is only connected to a small region of neurons in the previous layer. This region is defined by the size of the filter (in this case, 5x5). This constraint reflects the idea that only nearby pixels are relevant for feature detection.
2. **Shared Weights**: In a fully connected layer, the weights are independent and learned separately. In a convolutional layer, the same set of weights (the filter) is used for all neurons in the layer. This constraint reflects the idea that if a feature is useful to compute at one location, it should also be useful at a different location.
3. **Pooling**: The pooling layer (in this case, 3x3) reduces the spatial dimensions of the input by applying a downsampling operation along the spatial dimensions. This introduces a form of translation invariance and reduces the computational complexity.
4. **Fully Connected Layer**: The fully connected layer with 128 units can be seen as a standard layer in a feedforward neural network, where each unit is connected to all outputs from the previous layer (which would be the flattened output of the pooling layer).

So, while a CNN can be seen as a feedforward neural network with the pixel values of the image as inputs, the constraints imposed by the convolutional and pooling layers make it particularly suited for tasks involving images. These constraints help to reduce the number of parameters, provide translation invariance, and allow the network to learn hierarchical features, which are important for many visual tasks.