ASSIGNMENT - 2

1. Explain convolutional neural network, and how does it work?

Ans: CNNs are a type of deep learning architecture specifically designed for processing data with a grid-like structure, such as images. They excel at tasks like image classification, object detection, and image segmentation.

How CNNs Work:

Convolutional Layers:

* Apply filters (kernels) that slide across the input data (image) to extract features like edges, lines, and shapes.
* Each filter learns specific features at different locations in the image.
* The output of a convolutional layer is called a feature map.

Pooling Layers:

* Downsample the feature maps by summarizing the information in a local region.
* This reduces the spatial dimensionality of the data while preserving important features.
* Common pooling operations include max pooling and average pooling.

Fully-Connected Layers:

* Similar to regular neural networks, these layers perform classification or regression tasks based on the extracted features.
* They connect all neurons from previous layers, allowing for more complex feature combination.

2. How does refactoring parts of your neural network definition favor you?

Ans: Improved Readability and Maintainability: Breaking down the network into well-defined functions or classes makes the code easier to understand, modify, and debug.

Reusability: You can reuse network components for different tasks or create modular architectures.

Experimentation: Refactoring allows for easier exploration of different network configurations by swapping modules or changing parameters.

3. What does it mean to flatten? Is it necessary to include it in the MNIST CNN? What is the reason for this?

Ans: Flattening transforms a multi-dimensional tensor (e.g., image feature maps) into a one-dimensional vector.

It's typically used before fully-connected layers, where each neuron needs a single input value.

In MNIST CNN:

* Flattening is necessary because fully-connected layers require a 1D vector as input.
* It reshapes the output of convolutional and pooling layers (usually 3D tensors with height, width, and channels) into a vector representing all the activations from the feature maps.

4. What exactly does NCHW stand for?

Ans: NCHW (Number of channels, Channel index, Height, Width) is a common tensor format used in deep learning frameworks like PyTorch.

It represents the order of dimensions in a tensor:

* The first dimension (N) indicates the number of samples in a batch (e.g., multiple images).
* The second dimension (C) specifies the number of channels in the data (e.g., 1 for grayscale, 3 for RGB).
* The third and fourth dimensions (H and W) represent the height and width of the image, respectively.

5. Why are there 7\*7\*(1168-16) multiplications in the MNIST CNN’s third layer?

Ans: 7 \* 7 \* (16 \* (Oc - 1))

6.Explain definition of receptive field?

Ans: The receptive field of a neuron in a CNN defines the region of the input that contributes to its activation.

In the first convolutional layer, the receptive field is the size of the filter.

As you go deeper with stacked convolutional layers, the receptive field grows.

7. What is the scale of an activation’s receptive field after two stride-2 convolutions? What is the reason for this?

Ans: With two stride-2 convolutions, the receptive field quadruples in size compared to a single convolution with stride 1.

Reason:

* Stride 2 means the filter jumps two pixels at a time during the convolution operation.
* This effectively doubles the receptive field size in both height and width dimensions in each convolution.
* Two stride-2 convolutions lead to a 2x2 = 4 times larger receptive field.

8. What is the tensor representation of a color image?

Ans: In NCHW format, a color image is represented as a 4D tensor with dimensions:

* N (batch size)
* C (channels) = 3 (RGB)
* H (height)
* W (width)

9. How does a color input interact with a convolution?

Ans: During convolution in a color image, the filter interacts with all three color channels (R, G, B) simultaneously.

The filter learns to extract