Sharif University of Technology
Electrical Engineering Department

Deep Learning
THW3

Amir Hossein Yari
99102507

December 5, 2023

Contents

Question 1

Both architectures have the same number of parameters and go through the same process, essentially treating each filter's output as an output neuron in the convolutional layer, then indeed both architectures are equivalent in terms of representational capacity and the ability to learn from the data.

Question 2

a

$$\text{Output Size} = \frac{\text{Input Size} - \text{Filter Size} + 2 \times \text{Padding}}{\text{Stride}} + 1$$
$$\Rightarrow \mathrm{W_O} = \frac{128 - 5 + 2 \times 2}{1} + 1 = \frac{128 - 1}{1} + 1 = 128$$

Since the image and filter are square, $\Rightarrow \mathrm{W_O} = \mathrm{H_O}$.
Since there are 16 filters, the final output volume has a depth of 16.

$$\text{Output Size} = 128 \times 128 \times 16$$

$$\text{Number of Parameters} = (\text{Filter Width} \times \text{Filter Height} \times \text{Number of Input Channels})$$
$$\times \text{Number of Filters} + \text{Number of Filters}$$
$$\Rightarrow \text{Number of Parameters} = (5 \times 5 \times 3) \times 16 + 16 = 1216$$

b

Layer 1 (Convolutional Layer):

$$\text{Output Size}_1 = \frac{128 - 5 + 2 \times 2}{1} + 1 = 128 \Rightarrow 128 \times 128 \times 16$$
$$\text{Number of Parameters}_1 = (5 \times 5 \times 3) \times 16 + 16 = 1216$$

Layer 1 (Max Pooling Layer):

$$\text{Output Size}_2 = \frac{128 - 2}{2} + 1 = 64 \times 64 \times 16$$
$$\text{Number of Parameters}_2 = 0$$

Layer 1 (ReLU Activation):

$$\text{Output Size}_3 = 64 \times 64 \times 16$$
$$\text{Number of Parameters}_3 = 0$$

Layer 2 (Convolutional Layer):

$$\text{Output Size}_4 = \frac{64 - 5 + 2 \times 2}{1} + 1 = 64 \times 64 \times 16$$
$$\text{Number of Parameters}_4 = (5 \times 5 \times 16) \times 16 + 16 = 6416$$

Layer 2 (Max Pooling Layer):

$$\text{Output Size}_5 = \frac{64-2}{2} + 1 = 32 \times 32 \times 16$$
$$\text{Number of Parameters}_5 = 0$$

Layer 2 (ReLU Activation):

$$\text{Output Size}_6 = 32 \times 32 \times 16$$
$$\text{Number of Parameters}_6 = 0$$

Layer 3 (Convolutional Layer):

$$\text{Output Size}_7 = \frac{32-5+2\times 2}{1} + 1 = 32 \times 32 \times 16$$
$$\text{Number of Parameters}_7 = (5 \times 5 \times 16) \times 16 + 16 = 6416$$

Layer 3 (Max Pooling Layer):

$$\text{Output Size}_8 = \frac{32-2}{2} + 1 = 16 \times 16 \times 16$$
$$\text{Number of Parameters}_8 = 0$$

Layer 3 (ReLU Activation):

$$\text{Output Size}_9 = 16 \times 16 \times 16$$
$$\text{Number of Parameters}_9 = 0$$

Total Layer:

$$\text{Output Size} = 16 \times 16 \times 16$$
$$\text{Number of Parameters} = 1216 + 6416 + 6416 = 14048$$

c

Suppose that we have 1 fully connected.

$$\begin{aligned}\text{Number of Parameters} &= \text{Number of Input Neurons} \times \text{Number of Output Neurons} \\ &+ \text{Number of Parameters of CNN}\end{aligned}$$
$$\Rightarrow \text{Number of Parameters} = (16 \times 16 \times 16) + 14048 = 40960 + 14048 = 55008$$

d

Receptive Field Size $=$ (Filter Size -1) $\times$ Stride $+1+$ Previous Receptive Field Size

Convolutional Layer 1:

$$\text{Receptive Field Size}_1 = (5-1)\times 1 + 1 = 5$$

Max Pooling Layer 1:

$$\text{Receptive Field Size}_2 = (2-1)\times 2 + 1 + 5 = 6$$

Convolutional Layer 2:

$$\text{Receptive Field Size}_4 = (5-1)\times 1 + 1 + 6 = 11$$

Max Pooling Layer 2:

$$\text{Receptive Field Size}_5 = (2-1)\times 2 + 1 + 11 = 12$$

Convolutional Layer 3:

$$\text{Receptive Field Size}_7 = (5-1)\times 1 + 1 + 12 = 17$$

Max Pooling Layer 3:

$$\text{Receptive Field Size}_8 = (2-1)\times 2 + 1 + 17 = 18$$

Therefore, the receptive field for a neuron in the third layer (before the fully connected layer) is 18×18 in terms of the input tensor size.

Question 3

1-a

The U-Net architecture differs from a traditional convolutional neural network in its unique structure and is distinctively characterized by the use of skip connections. The reason we observe a U-shaped structure is that it employs a contracting path, followed by an expansive path, resembling the letter "U." The key feature distinguishing U-Net from regular convolutional networks is the incorporation of skip connections. These connections enable the direct flow of information from the contracting path to the corresponding layers in the expansive path. This facilitates the preservation of spatial information and details that may be lost during the downsampling stages in the contracting path.

1-b

1. **Preservation of Spatial Information:**
   - During the contracting path, the resolution of the input data decreases due to pooling operations, leading to a loss of fine-grained spatial information.
   - Skip connections create shortcuts that enable the direct transfer of information from early layers in the contracting path to the corresponding layers in the expansive path.
   - This helps in preserving detailed spatial information that might be lost in the downsampling stages.
2. **Combining Low-Level and High-Level Features:**
   - Skip connections facilitate the fusion of low-level features (captured by early layers) and high-level features (captured by later layers).
   - By combining information from different scales of the network, U-Net can better understand both local and global context in the input data.
3. **Mitigating the Vanishing Gradient Problem:**
   - Skip connections provide a direct gradient flow during backpropagation, which helps mitigate the vanishing gradient problem.
   - This is especially important when training deep networks, as it allows for more effective learning of meaningful representations.

4. **Improving Segmentation Accuracy:**
    - In tasks like image segmentation, where U-Net is commonly applied, the combination of skip connections enhances the network's ability to precisely delineate object boundaries and generate accurate segmentation masks.

1-c

1. **Preservation of Fine Details:**
    - Medical images often contain fine structures and subtle details that are crucial for accurate diagnosis. Skip connections help preserve these details during the downsampling process, preventing the loss of critical information.
2. **Enhanced Feature Fusion:**
    - In medical images, relevant information may be distributed across different scales and levels of abstraction. Skip connections facilitate the fusion of low-level and high-level features, allowing the network to capture both local details and global context. This is particularly important in medical image analysis, where abnormalities may manifest at various scales.
3. **Improved Localization of Anomalies:**
    - Medical diagnoses frequently rely on the precise localization of anomalies or pathologies within an image. Skip connections enable the network to combine information from multiple resolutions, enhancing its ability to accurately locate and delineate abnormalities.
4. **Addressing Class Imbalance and Irregular Shapes:**
    - Medical datasets often exhibit class imbalance, where normal cases significantly outnumber abnormal ones. Skip connections assist in handling class imbalance by providing a mechanism for the network to focus on relevant details related to abnormalities. Moreover, medical structures can have irregular shapes, and skip connections aid in capturing these complex spatial configurations.
5. **Optimized Training and Gradient Flow:**

- The vanishing gradient problem can be particularly challenging in deep networks applied to medical images. Skip connections alleviate this issue by providing shortcut paths for gradient flow during backpropagation. This facilitates more effective training of deep models on medical datasets.

6. **Increased Sensitivity and Specificity:**
   - Sensitivity (the ability to detect true positives) and specificity (the ability to avoid false positives) are critical metrics in medical image analysis. Skip connections contribute to a network's ability to achieve a balance between sensitivity and specificity by preserving essential information and reducing the likelihood of false positives or negatives.

2-a

$$H = W = 256$$

- **After the first downsampling:**

$$H/2 \times W/2 = 128 \times 128$$

- **After the second downsampling:**

$$H/4 \times W/4 = 64 \times 64$$

- **After the third downsampling:**

$$H/8 \times W/8 = 32 \times 32$$

- **After the fourth downsampling:**

$$H/16 \times W/16 = 16 \times 16$$

2-b

To calculate the number of parameters in a convolutional layer, we use the formula:

$$\text{Number of Parameters} = (\text{Number of input channels} \times \text{Size of the kernel})$$

$$\times \text{Number of filters} + \text{Number of filters}$$

$$\Rightarrow \text{Number of Parameters} = (64 \times 3 \times 3) \times 128 + 128 = 73856$$

3-a

Dense Connections in DenseNet:

- **Information Flow:** In DenseNet, each layer receives feature maps from all preceding layers and passes its own feature maps to all subsequent layers, creating a dense connectivity pattern.
- **Concatenation:** Feature maps from different layers are concatenated along the channel dimension before being passed to subsequent layers.
- **Advantage:** Dense connections promote feature reuse, allowing each layer direct access to the features learned by all preceding layers.

Residual Connections in ResNet:

- **Information Flow:** In ResNet, each residual block has a shortcut connection that bypasses one or more layers. ResNet focuses on learning the residual mapping, which is added to the original input.
- **Addition:** The output of a residual block is obtained by adding the transformed features (learned by the block) to the original input.
- **Advantage:** Residual connections facilitate the training of very deep networks by addressing the vanishing gradient problem. The shortcut connections provide a direct path for the gradient to flow through the network.

While both approaches aim to address challenges in training deep networks, they have distinct mechanisms for handling information flow and gradients, leading to different connectivity patterns within the network. Dense connections create a dense inter-layer connectivity, while residual connections introduce shortcut paths for gradient flow.

3-b

Mitigation of Vanishing Gradient Problem:

- **Dense Connectivity:** In DenseNet, each layer receives feature maps from all preceding layers and passes its own feature maps to all subsequent layers, facilitating information flow.

- **Short Paths for Gradients:** Dense connections create shorter paths for gradients during backpropagation, addressing the vanishing gradient problem associated with deep networks.
- **Promotion of Feature Reuse:** Direct access to features learned by preceding layers encourages feature reuse, mitigating the vanishing gradient problem.

Computational Advantages:

- **Parameter Efficiency:** Dense connectivity reduces redundant parameters, making the model more parameter-efficient compared to traditional architectures.
- **Enhanced Feature Propagation:** Dense connections facilitate feature reuse and propagation, leading to improved feature representations.
- **Improved Gradients for Training:** Short paths for gradients ease training of very deep networks by improving gradient flow during backpropagation.
- **Regularization Effect:** Dense connectivity acts as implicit feature-wise regularization, making the network more robust and less prone to overfitting.

4-a

In a DenseNet, each layer in a dense block receives as input the feature maps produced by all preceding layers in that dense block.
So, the third layer will receive $64 + 128 = 192$ feature maps as input.

4-b

Number of channels in the nth layer = Initial number of channels + $(n-1) \times k$
In this case, for the third layer ($n = 3$), the calculation would be:

$$\text{Number of channels in the third layer} = 32 + (3-1) \times 24$$

$$= 32 + 2 \times 24 = 32 + 48 = 80$$