November 1, 2025

Joseph DeLeonardis
UIN: 0820000866

CSCE-633

Machine Learning Homework #4: CNN

Question 1: Convolution Operation

Given the following 7×7 input matrix and 3×3 filter:

Input Matrix:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 4 & 1 & 0 & 0 \\ 0 & 3 & 1 & 1 & 0 & 1 & 0 \\ 0 & 2 & 4 & 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 5 & 2 & 2 & 0 \\ 0 & 0 & 1 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Filter:

$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

Part 1: Without max pooling and with a stride of 1

Calculation approach:

For convolution, we slide the 3×3 filter across the input matrix by the stride value.
The output dimensions are calculated as:

$$\text{Output size} = \frac{\text{Input size} - \text{Filter size}}{\text{Stride}} + 1 = \frac{7-3}{1} + 1 = 5$$

So the output will be a 5×5 matrix.

Example calculation for position (0,0):

The filter overlaps with the top-left 3×3 region of the input:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 3 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

Element-wise multiplication and sum:

$$\begin{aligned} \text{Output}[0,0] &= (0 \times 1) + (0 \times 0) + (0 \times -1) \\ &\quad + (0 \times 1) + (0 \times 0) + (2 \times -1) \\ &\quad + (0 \times 1) + (3 \times 0) + (1 \times -1) \\ &= 0 + 0 + 0 + 0 + 0 + (-2) + 0 + 0 + (-1) \\ &= -3 \end{aligned}$$

Example calculation for position (0,1):

Shifting the filter right by 1, it now overlaps with columns 1-3:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 4 \\ 3 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

Element-wise multiplication and sum:

$$\begin{aligned}\text{Output}[0,1] &= (0\times 1)+(0\times 0)+(0\times -1)\\ &\quad +(0\times 1)+(2\times 0)+(4\times -1)\\ &\quad +(3\times 1)+(1\times 0)+(1\times -1)\\ &= 0+0+0+0+0+(-4)+3+0+(-1)\\ &= -2\end{aligned}$$

Following this process for all positions yields the complete output matrix.
Final Output (5×5):

$$\begin{bmatrix}-3 & -2 & 2 & 4 & 1\\ -7 & -1 & 6 & 4 & 1\\ -5 & 0 & 3 & 3 & 2\\ -5 & -5 & 1 & 5 & 4\\ -1 & -6 & -3 & 5 & 4\end{bmatrix}$$

Part 2: Without max pooling and with a stride of 2

Calculation approach:

With stride 2, the filter moves 2 positions at a time (both horizontally and vertically).
The output dimensions are:

$$\text{Output size} = \frac{7-3}{2}+1 = 3$$

So the output will be a 3×3 matrix.
Example calculation for position (0,0):
Same as Part 1, this gives us Output[0,0] = −3.
Example calculation for position (0,1):
Now with stride 2, the filter shifts to columns 2-4 (skipping column 1):

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

Element-wise multiplication and sum:

$$\begin{aligned}\text{Output}[0,1] &= (0\times 1)+(0\times 0)+(0\times -1)\\ &\quad +(2\times 1)+(4\times 0)+(1\times -1)\\ &\quad +(1\times 1)+(1\times 0)+(0\times -1)\\ &= 0+0+0+2+0+(-1)+1+0+0\\ &= 2\end{aligned}$$

Final Output (3×3):

$$\begin{bmatrix}-3 & 2 & 1\\ -5 & 3 & 2\\ -1 & -3 & 4\end{bmatrix}$$

Part 3: With max pooling and with a stride of 1

Note on ambiguity: The problem statement says "stride of 1" but the image shows "stride = 2" for max pooling. To be thorough, I provide both interpretations below.

Step 1: Convolution with stride 1

First, perform convolution with stride 1 (same as Part 1):

$$\begin{bmatrix} -3 & -2 & 2 & 4 & 1 \\ -7 & -1 & 6 & 4 & 1 \\ -5 & 0 & 3 & 3 & 2 \\ -5 & -5 & 1 & 5 & 4 \\ -1 & -6 & -3 & 5 & 4 \end{bmatrix}$$

Interpretation A: Max pooling with stride = 1

If we interpret "stride of 1" as applying to the max pooling operation:

Max pooling takes the maximum value from each 3×3 region with stride 1.

The output dimensions after max pooling are:

$$\text{Output size} = \frac{5-3}{1} + 1 = 3$$

So the final output will be a 3×3 matrix.

Example: Position (0,0)

The top-left 3×3 region is:

$$\begin{bmatrix} -3 & -2 & 2 \\ -7 & -1 & 6 \\ -5 & 0 & 3 \end{bmatrix}$$

The maximum value in this region is $\max(-3, -2, 2, -7, -1, 6, -5, 0, 3) = 6$.

Example: Position (0,1) - Moved right by 1

Shifting right by stride 1, we get columns 1-3:

$$\begin{bmatrix} -2 & 2 & 4 \\ -1 & 6 & 4 \\ 0 & 3 & 3 \end{bmatrix}$$

The maximum value is $\max(-2, 2, 4, -1, 6, 4, 0, 3, 3) = 6$.

Note: This pool overlaps 2 columns with the previous pool.

Final Output (Interpretation A) - 3×3:

$$\begin{bmatrix} 6 & 6 & 6 \\ 6 & 6 & 6 \\ 3 & 5 & 5 \end{bmatrix}$$

Interpretation B: Max pooling with stride = 2 (as shown in image)

If we follow the image which shows "stride = 2" for max pooling:

Max pooling takes the maximum value from each 3×3 region with stride 2.

The output dimensions after max pooling are:

$$\text{Output size} = \frac{5-3}{2} + 1 = 2$$

So the final output will be a 2×2 matrix.

Example: Position (0,0)

Same top-left 3×3 region as Interpretation A, maximum value = 6.

Example: Position (0,1) - Moved right by 2

Shifting right by stride 2, we get columns 2-4:

$$\begin{bmatrix} 2 & 4 & 1 \\ 6 & 4 & 1 \\ 3 & 3 & 2 \end{bmatrix}$$

The maximum value is $\max(2, 4, 1, 6, 4, 1, 3, 3, 2) = 6$.

Note: This pool does NOT overlap with the previous pool.

Final Output (Interpretation B) - 2×2:

$$\begin{bmatrix} 6 & 6 \\ 3 & 5 \end{bmatrix}$$

Note: Python code was written to compute and verify these results, as manual computation is prone to errors. The examples shown demonstrate understanding of the hand calculation process.

Key concepts demonstrated: - Output size determines the dimensions of our output matrix - Stride determines how much we move the filter across the input matrix - Max pooling takes the maximum value from each pooling window - Pools can overlap depending on the stride value (stride=1 overlaps, stride=2 does not)

Due to ambiguity in the problem statement (text says ”stride of 1” while the image shows ”stride = 2” for max pooling), both interpretations were computed to ensure completeness.

Question 2: Image Classification using CNN

The required functions for the SUN397Dataset and CNN model were implemented in the submitted Python code.

Strategy and Implementation

The final model employed a **5-layer CNN architecture** chosen after iterative testing found 3-layer designs to underfit the data. The model was trained using the **Adam** optimizer and a **ReduceLROnPlateau** learning rate scheduler. The entire training process required approximately 8 hours to complete 30 epochs on a CPU.

Dataset Analysis

The SUN397 dataset subset consisted of **6,042 images** distributed across **4 classes**:

- airport_terminal (label: 0)
- bedroom (label: 1)
- dining_room (label: 2)
- living_room (label: 3)

Labels were assigned alphabetically as specified in the assignment requirements.
The dataset normalization statistics were computed across all training images:

- **Mean (R, G, B):** 0.5286, 0.4668, 0.4109
- **Std (R, G, B):** 0.2157, 0.2236, 0.2264

However, the normalization check failed on Gradescope, and it was difficult to determine why the computed values did not match the expected results.

Strategy and Implementation

The final model employed a **5-layer CNN architecture** chosen after iterative testing found 3-layer designs to underfit the data. The model was trained using the **Adam** optimizer and a **ReduceLROnPlateau** learning rate scheduler. The entire training process required approximately 8 hours to complete 30 epochs on a CPU, processing all 6,042 training images.

Results

- **Final Accuracy:** After 30 epochs, the training set achieved an accuracy of approximately **78.00**%, successfully surpassing the **75**% target.
- **Submission:** The final model weights were saved to the required **model.pt** file.

Use of AI and External Resources

Claude.ai was used to assist with the following aspects of this assignment:
The following prompts were used to get assistance with the assignment:

- "Explain a standard starting point for CNN values? "
- "Why is my 3 layer network under performing?"
- "What is an Adam Optimizer?"
- "Provide a structure for print statements to aid in locally testing code."