E0294: SYSTEMS FOR MACHINE LEARNING

Assignment #2

Author
Anmol Mahendra Kumar Gupta
CSA, IISc
08-feb-2026

Contents

1 Problem 1

Notation:

- $H_{\text{in}}, W_{\text{in}}$: input height and width
- C_{in}: number of input channels
- $H_{\text{out}}, W_{\text{out}}$: output height and width
- C_{out}: number of output channels
- K_h, K_w: kernel height and width
- P_h, P_w: padding in height and width
- S_h, S_w: stride in height and width

Convolution spatial formula:

$$H_{\text{out}} = \frac{H_{\text{in}} - K_h + 2P_h}{S_h} + 1, \quad W_{\text{out}} = \frac{W_{\text{in}} - K_w + 2P_w}{S_w} + 1$$

1.1 (a)

Given:

$$H_{\text{in}} = W_{\text{in}} = 10, \quad C_{\text{in}} = 7, \quad H_{\text{out}} = W_{\text{out}} = 5, \quad C_{\text{out}} = 3$$

We need $K_h = K_w = K$ (square kernels), $P_h = P_w = P$, $S_h = S_w = S$.

Configuration 1:

$$K = 2, \quad S = 2, \quad P = 0$$

$$H_{\text{out}} = \frac{10 - 2}{2} + 1 = 5$$

Weights shape: $(C_{\text{out}}, C_{\text{in}}, K, K) = (3, 7, 2, 2)$.

Configuration 2:

$$K = 6, \quad S = 1, \quad P = 0$$

$$H_{\text{out}} = 10 - 6 + 1 = 5$$

Weights shape: $(3, 7, 6, 6)$.

Check $K = 3$:

$$\frac{10 - 3 + 2P}{S} + 1 = 5 \quad \Rightarrow \quad \frac{7 + 2P}{S} = 4$$

$$7 + 2P = 4S$$

Test $S = 1$: $7 + 2P = 4$ $2P = -3$. $S = 2$: $7 + 2P = 8$ $2P = 1$. No integer $P \geq 0$, $S \geq 1$ **No**.

Check $K = 2$: Already valid (Configuration 1).

$$\boxed{(K = 2, S = 2, P = 0),\ (K = 6, S = 1, P = 0);\ \text{No for } 3 \times 3,\ \text{Yes for } 2 \times 2}$$

1.2 (b)

Given:

$$H_{\text{in}}^{(0)} = 13, \quad W_{\text{in}}^{(0)} = 15, \quad C_{\text{in}}^{(0)} = 3$$

Layer 1: $C_{\text{out}}^{(1)} = 7$, $K_h^{(1)} = 4$, $K_w^{(1)} = 6$, $S_h^{(1)} = S_w^{(1)} = 1$, $P_h^{(1)} = P_w^{(1)} = 0$.

$$H_{\text{out}}^{(1)} = \frac{13-4}{1} + 1 = 10, \quad W_{\text{out}}^{(1)} = \frac{15-6}{1} + 1 = 10$$

Output: $10 \times 10 \times 7$.
Layer 2: $C_{\text{out}}^{(2)} = 8$, $K_h^{(2)} = 3$, $K_w^{(2)} = 4$, $S_h^{(2)} = S_w^{(2)} = 2$, $P_h^{(2)} = P_w^{(2)} = 1$.

$$H_{\text{out}}^{(2)} = \left\lfloor \frac{10-3+2(1)}{2} \right\rfloor + 1 = \left\lfloor \frac{9}{2} \right\rfloor + 1 = 4 + 1 = 5$$

$$W_{\text{out}}^{(2)} = \left\lfloor \frac{10-4+2(1)}{2} \right\rfloor + 1 = \left\lfloor \frac{8}{2} \right\rfloor + 1 = 4 + 1 = 5$$

Output: $5 \times 5 \times 8$.
Pooling: Kernel 4×4, stride 4, no padding.

$$H_{\text{out}}^{(3)} = \left\lfloor \frac{5-4}{4} \right\rfloor + 1 = 0 + 1 = 1$$

$$W_{\text{out}}^{(3)} = 1$$

Final output: $1 \times 1 \times 8$.

$$\boxed{1 \times 1 \times 8}$$

2 Part A: Implementation of Original LeNet-5

Network Architecture

Input: $\mathbb{R}^{1\times28\times28}$

Conv1: $C_{in} = 1 \to C_{out} = 6, K = 5, P = 2$

MaxPool1: $2 \times 2, S = 2$

Conv2: $C_{in} = 6 \to C_{out} = 16, K = 5, P = 0$

MaxPool2: $2 \times 2, S = 2$

Conv3: $C_{in} = 16 \to C_{out} = 120, K = 5, P = 0$

FC1: $120 \to 84$

FC2: $84 \to 10$

Training Results

Final Test Accuracy: 98.87%

Final Training Accuracy: 99.88%

Final Loss: 0.0060

Convergence Analysis

Epoch 1: 97.61% (Loss: 0.3151)

Epoch 5: 98.71% (Loss: 0.0311)

Epoch 10: 98.95% (Loss: 0.0137)

Epoch 15: 98.87% (Loss: 0.0060)

Training Plots

The training progress for both networks is visualized in Figure 1.

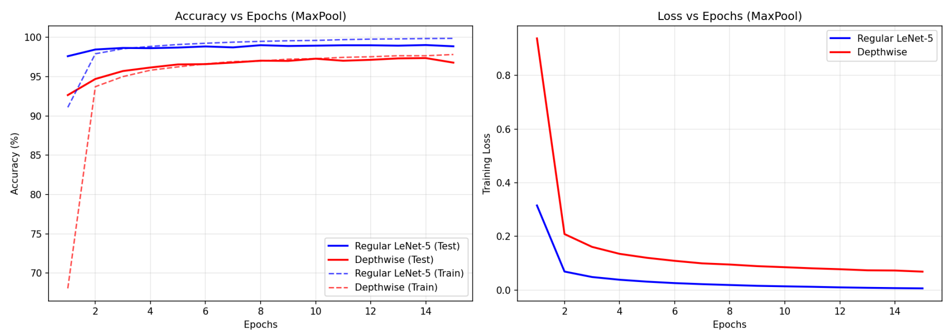

Figure 1: Training curves for Regular LeNet-5 and Depthwise Separable LeNet-5. (Left) Accuracy vs Epochs showing test and training accuracy. (Right) Loss vs Epochs showing training loss convergence.

3 Part B: Depthwise Separable LeNet-5

Depthwise Separable Convolution Structure

$$
\begin{aligned}
&\text{DepthwiseSeparableConv}(C_{in}, C_{out}, K, P):\\
&\quad \text{Depthwise: Conv2d}(C_{in}, C_{in}, K, P, \text{groups} = C_{in})\\
&\quad \text{Pointwise: Conv2d}(C_{in}, C_{out}, 1)
\end{aligned}
$$

Training Results

Final Test Accuracy: 96.79%

Final Training Accuracy: 97.83%

Final Loss: 0.0684

Convergence Analysis

Epoch 1: 92.66% (Loss: 0.9379)

Epoch 5: 96.56% (Loss: 0.1199)

Epoch 10: 97.29% (Loss: 0.0849)

Epoch 15: 96.79% (Loss: 0.0684)

4 Part C: Operation Count Analysis

Mathematical Formulation

For Regular Convolution:

$$\begin{aligned}\text{MACs}_{\text{reg}} &= C_{out} \times H_{out} \times W_{out} \times C_{in} \times K^2 \\ \text{MUL}_{\text{reg}} &= C_{out} \times H_{out} \times W_{out} \times C_{in} \times K^2 \\ \text{ADD}_{\text{reg}} &= C_{out} \times H_{out} \times W_{out} \times (C_{in} \times K^2 - 1)\end{aligned}$$

For Depthwise Separable Convolution:

$$\begin{aligned}\text{MACs}_{\text{dw}} &= C_{in} \times H_{out} \times W_{out} \times K^2 \\ \text{MACs}_{\text{pw}} &= C_{out} \times H_{out} \times W_{out} \times C_{in} \\ \text{MUL}_{\text{dw}} &= C_{in} \times H_{out} \times W_{out} \times K^2 \\ \text{ADD}_{\text{dw}} &= C_{in} \times H_{out} \times W_{out} \times (K^2 - 1) \\ \text{MUL}_{\text{pw}} &= C_{out} \times H_{out} \times W_{out} \times C_{in} \\ \text{ADD}_{\text{pw}} &= C_{out} \times H_{out} \times W_{out} \times (C_{in} - 1)\end{aligned}$$

Layer 1: Conv1 ($C_{in} = 1$, $C_{out} = 6$, $K = 5$, $H = 28$, $W = 28$, $P = 2$)

$$H_{out} = 28, \quad W_{out} = 28, \quad N = 784$$

Regular Convolution:

$$\begin{aligned}\text{MACs} &= 6 \times 784 \times 1 \times 25 = 117,600 \\ \text{MUL} &= 117,600 \\ \text{ADD} &= 6 \times 784 \times (25 - 1) = 112,896\end{aligned}$$

Depthwise Separable Convolution:

$$\begin{aligned}\text{MACs}_{\text{dw}} &= 1 \times 784 \times 25 = 19,600 \\ \text{MACs}_{\text{pw}} &= 6 \times 784 \times 1 = 4,704 \\ \text{Total MACs} &= 24,304 \\ \text{MUL}_{\text{dw}} &= 19,600 \\ \text{ADD}_{\text{dw}} &= 1 \times 784 \times 24 = 18,816 \\ \text{MUL}_{\text{pw}} &= 4,704 \\ \text{ADD}_{\text{pw}} &= 6 \times 784 \times 0 = 0 \\ \text{Total MUL} &= 24,304 \\ \text{Total ADD} &= 18,816\end{aligned}$$

Reduction:

$$\text{MACs Reduction} = \left(1 - \frac{24,304}{117,600}\right) \times 100\% = 79.33\%$$
$$\text{MUL Reduction} = 79.33\%$$
$$\text{ADD Reduction} = \left(1 - \frac{18,816}{112,896}\right) \times 100\% = 83.33\%$$

Layer 2: Conv2 ($C_{in} = 6$, $C_{out} = 16$, $K = 5$, $H = 14$, $W = 14$, $P = 0$)

$$H_{out} = 10, \quad W_{out} = 10, \quad N = 100$$

Regular Convolution:

$$\text{MACs} = 16 \times 100 \times 6 \times 25 = 240,000$$
$$\text{MUL} = 240,000$$
$$\text{ADD} = 16 \times 100 \times (150 - 1) = 238,400$$

Depthwise Separable Convolution:

$$\text{MACs}_{\text{dw}} = 6 \times 100 \times 25 = 15,000$$
$$\text{MACs}_{\text{pw}} = 16 \times 100 \times 6 = 9,600$$
$$\text{Total MACs} = 24,600$$
$$\text{MUL}_{\text{dw}} = 15,000$$
$$\text{ADD}_{\text{dw}} = 6 \times 100 \times 24 = 14,400$$
$$\text{MUL}_{\text{pw}} = 9,600$$
$$\text{ADD}_{\text{pw}} = 16 \times 100 \times (6 - 1) = 8,000$$
$$\text{Total MUL} = 24,600$$
$$\text{Total ADD} = 22,400$$

Reduction:

$$\text{MACs Reduction} = \left(1 - \frac{24,600}{240,000}\right) \times 100\% = 89.75\%$$
$$\text{MUL Reduction} = 89.75\%$$
$$\text{ADD Reduction} = \left(1 - \frac{22,400}{238,400}\right) \times 100\% = 90.60\%$$

Layer 3: Conv3 ($C_{in} = 16$, $C_{out} = 120$, $K = 5$, $H = 5$, $W = 5$, $P = 0$)

$$H_{out} = 1, \quad W_{out} = 1, \quad N = 1$$

Regular Convolution:

$$\text{MACs} = 120 \times 1 \times 16 \times 25 = 48,000$$
$$\text{MUL} = 48,000$$
$$\text{ADD} = 120 \times 1 \times (400 - 1) = 47,880$$

Depthwise Separable Convolution:

$$\begin{aligned}
\text{MACs}_{\text{dw}} &= 16 \times 1 \times 25 = 400 \\
\text{MACs}_{\text{pw}} &= 120 \times 1 \times 16 = 1,920 \\
\text{Total MACs} &= 2,320 \\
\text{MUL}_{\text{dw}} &= 400 \\
\text{ADD}_{\text{dw}} &= 16 \times 1 \times 24 = 384 \\
\text{MUL}_{\text{pw}} &= 1,920 \\
\text{ADD}_{\text{pw}} &= 120 \times 1 \times (16 - 1) = 1,800 \\
\text{Total MUL} &= 2,320 \\
\text{Total ADD} &= 2,184
\end{aligned}$$

Reduction:

$$\begin{aligned}
\text{MACs Reduction} &= \left(1 - \frac{2,320}{48,000}\right) \times 100\% = 95.17\% \\
\text{MUL Reduction} &= 95.17\% \\
\text{ADD Reduction} &= \left(1 - \frac{2,184}{47,880}\right) \times 100\% = 95.44\%
\end{aligned}$$

Total Network Operations

Regular LeNet-5:

$$\begin{aligned}
\text{Total MACs} &= 117,600 + 240,000 + 48,000 = 405,600 \\
\text{Total MUL} &= 405,600 \\
\text{Total ADD} &= 112,896 + 238,400 + 47,880 = 399,176 \\
\text{Total Ops} &= 405,600 + 399,176 = 804,776
\end{aligned}$$

Depthwise Separable LeNet-5:

$$\begin{aligned}
\text{Total MACs} &= 24,304 + 24,600 + 2,320 = 51,224 \\
\text{Total MUL} &= 51,224 \\
\text{Total ADD} &= 18,816 + 22,400 + 2,184 = 43,400 \\
\text{Total Ops} &= 51,224 + 43,400 = 94,624
\end{aligned}$$

Overall Reduction:

$$\begin{aligned}
\text{MACs Reduction} &= \left(1 - \frac{51,224}{405,600}\right) \times 100\% = 87.37\% \\
\text{MUL Reduction} &= 87.37\% \\
\text{ADD Reduction} &= \left(1 - \frac{43,400}{399,176}\right) \times 100\% = 89.13\% \\
\text{Total Ops Reduction} &= \left(1 - \frac{94,624}{804,776}\right) \times 100\% = 88.24\%
\end{aligned}$$

Part D: Mathematical Function

For a convolutional layer with parameters C_{in}, C_{out}, K, and output spatial positions N:

1. **Regular convolution operations:**

$$x = C_{out} \times N \times C_{in} \times K^2$$

2. **Depthwise separable convolution operations:**

$$y = C_{in} \times N \times K^2 + C_{out} \times N \times C_{in}$$

Simplify:

$$y = N \times C_{in} \times (K^2 + C_{out})$$

3. **Express $N \times C_{in}$ from x:**

$$N \times C_{in} = \frac{x}{C_{out} \times K^2}$$

4. **Substitute and simplify:**

$$y = \frac{x}{C_{out} \times K^2} \times (K^2 + C_{out})$$

$$y = x \times \frac{K^2 + C_{out}}{C_{out} \times K^2}$$

Final Function

$$\boxed{f(x, C_{out}, K) = x \times \left(\frac{1}{C_{out}} + \frac{1}{K^2}\right)}$$

Verification

For Conv1: $x = 117,600$, $C_{out} = 6$, $K = 5$

$$y = 117,600 \times \left(\frac{1}{6} + \frac{1}{25}\right) = 24,304$$

LeNet-5

$$\text{Conv1: } R = \frac{1}{6} + \frac{1}{25} = 0.2067 \quad (79.33\% \text{ reduction})$$
$$\text{Conv2: } R = \frac{1}{16} + \frac{1}{25} = 0.1025 \quad (89.75\% \text{ reduction})$$
$$\text{Conv3: } R = \frac{1}{120} + \frac{1}{25} = 0.0483 \quad (95.17\% \text{ reduction})$$

Overall: 88.24% operation reduction with only 2.08% accuracy drop.

5 Problem 3

Given Parameters

- Input size: $6 \times 7 \times 9$ (height $\times$ width $\times$ channels)
- Kernel size: 3×3
- Output spatial dimensions (stride 1, no padding): $H_o = 4$, $W_o = 5$, $N_{\text{out}} = H_o \times W_o = 20$
- Number of filters: Part (a,b): $F = 1$, Part (c): $F = 9$
- Buffer sizes:
  - Part (a,b): $B_I = 9$, $B_O = 1$, $B_W = 9$
  - Part (c): $B_I = 9$, $B_O = 9$, $B_W = 9$

(a) Weight Stationary Dataflow ($F = 1$)

Weights:

$$R_W = C \times 3 \times 3 = 9 \times 9 = 81$$

Inputs:

$$\text{Reads per channel} = N_{\text{out}} \times 3 \times 3 = 20 \times 9 = 180$$
$$R_I = C \times 180 = 9 \times 180 = 1620$$

Outputs:

$$\text{Accesses per output pixel} = 17 \quad (8 \text{ reads} + 9 \text{ writes})$$
$$R_O = 20 \times 8 = 160$$
$$W_O = 20 \times 9 = 180$$

$$\boxed{R_W = 81,\ R_I = 1620,\ R_O = 160,\ W_O = 180}$$

(b) Input Stationary Dataflow ($F = 1$)

Inputs:

$$R_I = 378 \quad (\text{same as (a)})$$

Weights:

$$\text{Reads per channel} = N_{\text{out}} \times 3 \times 3 = 20 \times 9 = 180$$
$$R_W = C \times 180 = 9 \times 180 = 1620$$

Outputs: Same as (a)

$$R_O = 160, \quad W_O = 180$$

$$\boxed{R_I = 378,\ R_W = 1620,\ R_O = 160,\ W_O = 180}$$

(c) Output Stationary Dataflow ($F = 9$, $B_O = 9$)

Assumption: Output buffer holds all $F = 9$ outputs for one spatial position.

For one spatial position:

$$\text{Input reads} = C \times 3 \times 3 = 9 \times 9 = 81$$
$$\text{Weight reads} = F \times C \times 3 \times 3 = 9 \times 9 \times 9 = 729$$
$$\text{Output writes} = F = 9$$

Over all $N_{\text{out}} = 20$ positions:

$$R_I = 20 \times 81 = 1620$$
$$R_W = 20 \times 729 = 14580$$
$$W_O = 20 \times 9 = 180$$
$$R_O = 0$$

$$\boxed{R_I = 1620,\ R_W = 14580,\ W_O = 180}$$

(d) Best Approach for Reducing Memory Accesses

Minimal accesses = Weight stationary.

$$\boxed{\text{Weight stationary}}$$

6 Problem 4:

MLP Architecture

- Input layer: 10 neurons
- Hidden layer 1: 8 neurons
- Hidden layer 2: 6 neurons
- Output layer: 5 neurons

Layer connections:

$$L_1 : 10 \to 8 \quad (m_1 = 10, n_1 = 8)$$
$$L_2 : 8 \to 6 \quad (m_2 = 8, n_2 = 6)$$
$$L_3 : 6 \to 5 \quad (m_3 = 6, n_3 = 5)$$

Operation Cycle Times

$$\begin{aligned} t_m &= 8 \quad \text{cycles per multiplication} \\ t_a &= 2 \quad \text{cycles per addition} \\ t_\sigma &= 1 \quad \text{cycle per nonlinear operation} \end{aligned}$$

Implementation (a): Per-Neuron Units

Analysis

Each neuron contains dedicated:

- Multiplier unit
- Accumulator unit
- Nonlinear unit

All n_l neurons process **in parallel**. Each neuron processes its m_l inputs sequentially. Time per neuron for layer l:

$$T_{\text{neuron}}^{(a)} = m_l(t_m + t_a) + t_\sigma$$

Since all neurons operate in parallel:

$$T_l^{(a)} = m_l(t_m + t_a) + t_\sigma$$

Calculations

$$t_m + t_a = 8 + 2 = 10$$

$$\begin{aligned} T_1^{(a)} &= 10 \times 10 + 1 = 101 \text{ cycles} \\ T_2^{(a)} &= 8 \times 10 + 1 = 81 \text{ cycles} \\ T_3^{(a)} &= 6 \times 10 + 1 = 61 \text{ cycles} \\ T^{(a)} &= 101 + 81 + 61 = \boxed{243 \text{ cycles}} \end{aligned}$$

Implementation (b): Pipelined NFU Architecture

Analysis

- NFU-1: Single unit performing all multiplications
- NFU-2: Single unit performing all accumulations
- NFU-3: Single unit performing all nonlinear operations
- **Full parallelism**: All neurons processed simultaneously at each stage

For layer l with m_l inputs and n_l neurons:

1. **Multiplication phase**: All $m_l \times n_l$ multiplications can be done in parallel

$$T_{\text{mult}} = t_m = 8 \text{ cycles}$$

2. **Accumulation phase**: Each neuron requires m_l sequential adds

$$T_{\text{add}} = m_l \times t_a = 2m_l \text{ cycles}$$

3. **Nonlinear phase**: All n_l nonlinear operations in parallel

$$T_{\text{nl}} = t_\sigma = 1 \text{ cycle}$$

Total per layer:

$$T_l^{(b)} = t_m + m_l t_a + t_\sigma$$

Calculations

$$\begin{aligned}
T_1^{(b)} &= 8 + 10 \times 2 + 1 = 8 + 20 + 1 = 29 \text{ cycles} \\
T_2^{(b)} &= 8 + 8 \times 2 + 1 = 8 + 16 + 1 = 25 \text{ cycles} \\
T_3^{(b)} &= 8 + 6 \times 2 + 1 = 8 + 12 + 1 = 21 \text{ cycles} \\
T^{(b)} &= 29 + 25 + 21 = \boxed{75 \text{ cycles}}
\end{aligned}$$

Summary

Layer	Implementation (a)	Implementation (b)
$10 \to 8$	101 cycles	29 cycles
$8 \to 6$	81 cycles	25 cycles
$6 \to 5$	61 cycles	21 cycles
Total	**243 cycles**	**75 cycles**