ST 311 Assignment 3 (theory part)

Due by **5pm, 17 March, 2023**

Candidate number:

Instruction: Attempt all questions. The total marks is 44.

1. LeNet is among the first published CNNs to capture wide attention for its performance on computer vision tasks. See below for the architecture of LeNet-5:

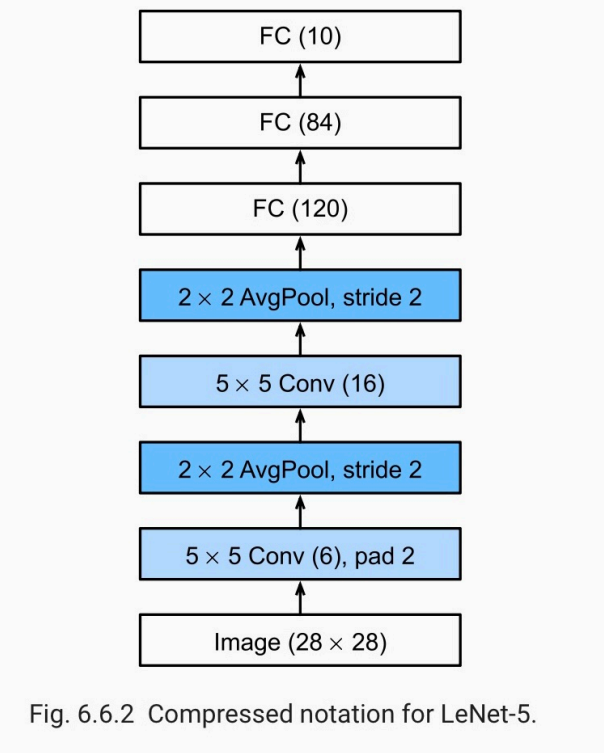


Figure 1: Architecture of LeNet-5

   (a) By passing an image of size 28×28 through the network, what are the output sizes for each layer (including the number of channels)? Please show your intermediate steps. Note that 'pad 2' means 2 rows and columns are zero padded on top/bottom and left/right sides respectively, so a total of 4 rows and 4 columns

of zeros are added. [10 marks]

(b) How many parameters (including weights and bias) are you supposed to estimate for each layer? Please show your intermediate steps and complete the following table. [10 marks]

Table 1: Number of parameters for each layer.

	Weights	Bias
First Conv		
First Pooling		
Second Conv		
Second Pooling		
First Linear		
Second Linear		
Last Linear		

2. Suppose you have a convolutional network with the following architecture:

- The input is an image of size 256×256.
- The first layer is a convolutional layer with 32-channel maps and filters of size 3×3.
- The second layer is a pooling layer with a stride of 2 and pooling groups of size 3×3.
- The third layer is a convolutional layer with 64-channel maps and filters of size 3×3.

(a) Suppose the first layer uses a stride of 1. Determine the size of the receptive field for a single unit in the second layer. [8 marks]

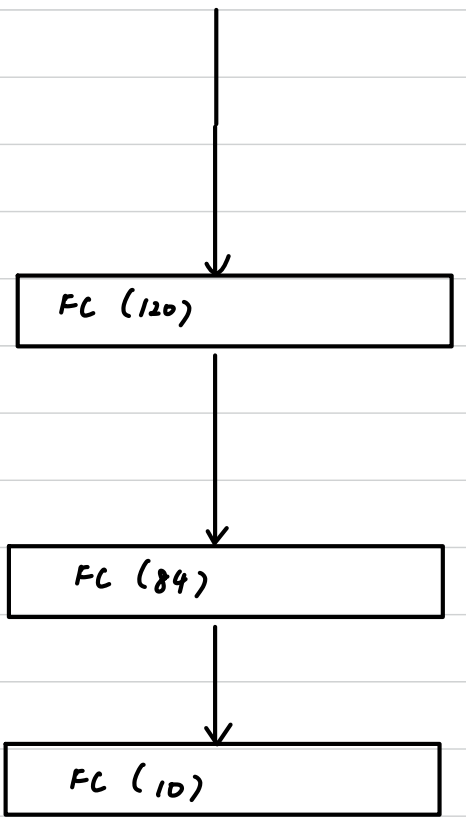
(b) Suppose the first layer uses a stride of 2. Determine the size of the receptive field for a single unit in the second layer. [8 marks]

(c) Suppose the first layer uses a stride of 1 and the third layer uses a stride of 1. Determine the size of the receptive field for a single unit in the third layer. [8 marks]

Hint: you may want to draw a one-dimensional convolutional network to reason about each part of the problem.

1a)

Compressed notation for LeNet-5	Output shape s.t. torch.Size([batch size, number of channel, height, weight])	
Image (28x28)	torch.Size([1, 1, 28, 28])	
5x5 Conv(6), pad 2	torch.Size([1, 6, 28, 28])	This convutional layer has 6 output channels size of feature map = input size - kernel size + 1 + 2(padding size) = 28 - 5 + 1 + 4 = 28
2x2 Avg Pool, stride 2	torch.Size([1, 6, 14, 14])	average pooling reduces size of feature map by half 28/2 = 14 (both height and width) pooling layer does not affect the number of channel
5x5 Conv(16)	torch.Size([1, 16, 10, 10])	This convutional layer has 16 output channels size of feature map = input size - kernel size + 1 = 14 - 5 + 1 = 10
2x2 Avg Pool, stride 2	torch.Size([1, 16, 5, 5])	average pooling reduces size of feature map reduced by half 10/2 = 5 (both height and width) pooling layer does not affect the number of channel



Output shape

torch.Size([1, 400])

entire pooled feature map matrix is transformed into a single column
1x16x5x5
= 400

torch.Size([1, 120])

Each fully connected layer reduces dimensionality

torch.Size([1, 84])

torch.Size([1, 10])

1b)

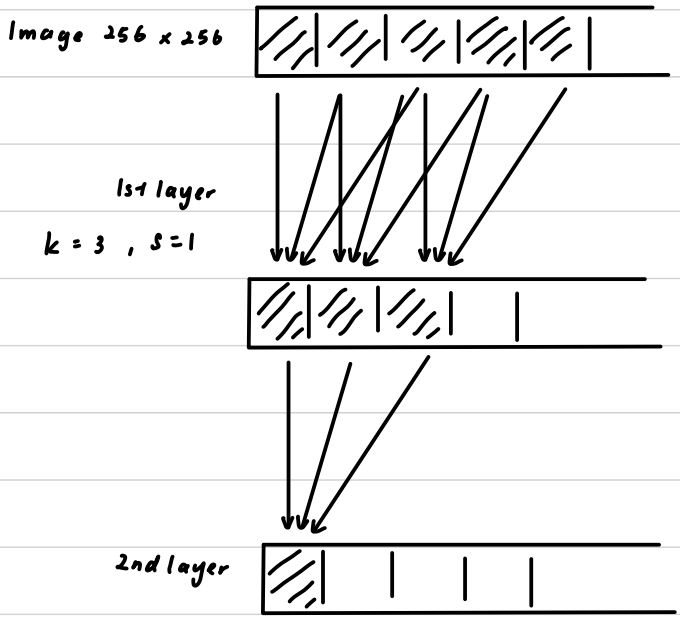
	Weights	Bias
First Conv	# filter x kernel size x input channel $= 6(5 \times 5)(1)$ $= 150$	$6(1)$ $= 6$
First Pooling	0	0
Second Conv	$(16)(5 \times 5)(6)$ $= 2400$	$16(1)$ $= 16$
Second Pooling	0	0
First Linear	$1 \times 16 \times 5 \times 5 = 400$ $120(400)$ $= 48,000$	$120(1)$ $= 120$
Second Linear	$84(120)$ $= 10,080$	$84(1)$ $= 84$
Last Linear	$10(84)$ $= 840$	$10(1)$ $= 10$

k : filter size

S : stride size

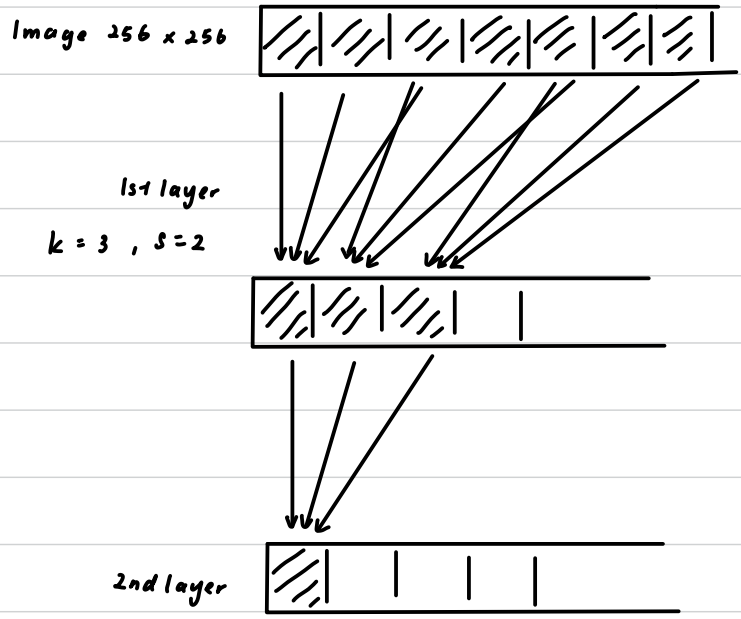
p = pooling size

2a)



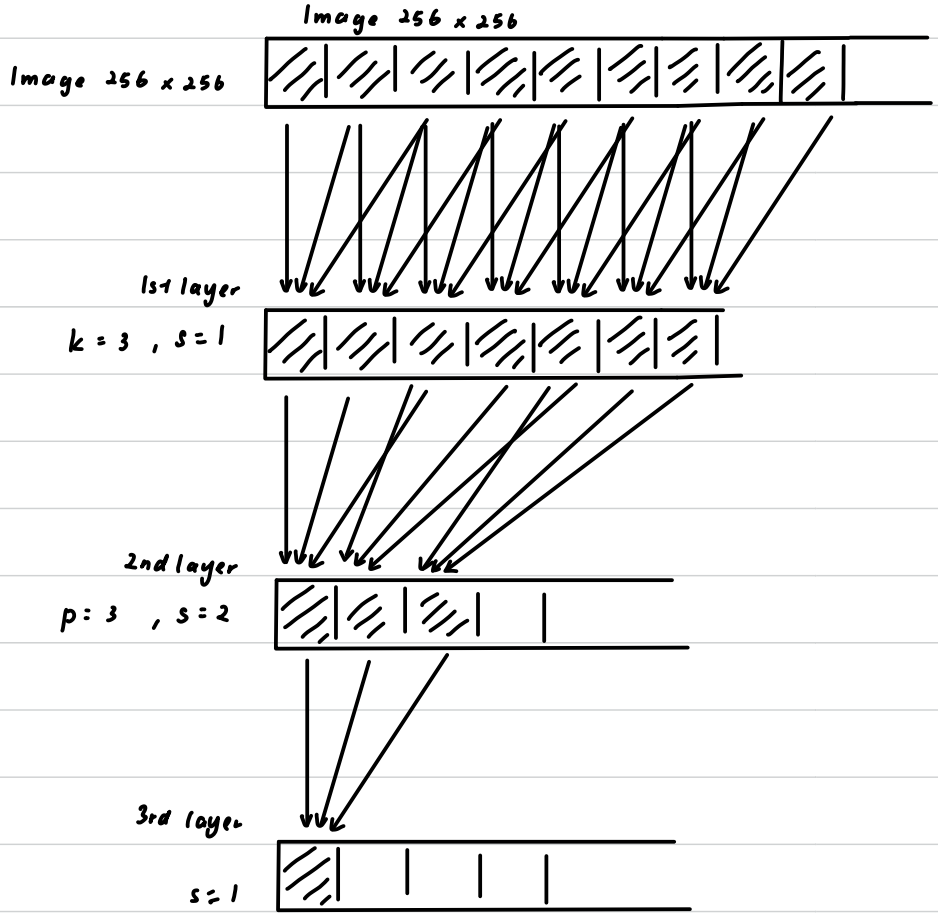
∴ Size of receptive field for single unit in 2nd layer is 5

2b)



∴ Size of receptive field for single unit in 2nd layer is 7

20)



∴ Size of receptive field for single unit in 2nd layer is 9