EE4016 AI with Deep Learning

Semester B 2025-2026

Assignment 2

Due: Mar 21, 2026, at 11:00 PM

Section A: (2 marks per question for Q1 to Q5)

Q1. Describe the hierarchical feature learning manner in CNNs.

Q2. What is the effect of increasing filter size in a convolutional layer of a CNN?

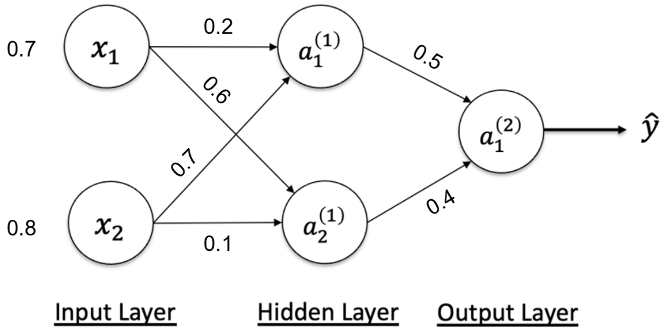
Q3. What is "padding" in convolutional layers? Explain valid padding and same padding.

Q4. What is the main innovation of VGGNets in Convolutional Neural Networks (CNNs)?

Q5. Provide 3 examples of Variants of Convolutional Operation.

Question 6 [20 marks]

A two-layer multilayer perceptron (MLP) is given as below, consisting of a hidden layer with a ReLU activation function and an output layer with a Sigmoid activation function. Notably, all neurons in this network do not have bias terms.



(a) Considering the activation functions used, what is the most appropriate use case for this MLP model, and how can its output be interpreted? [3 marks]

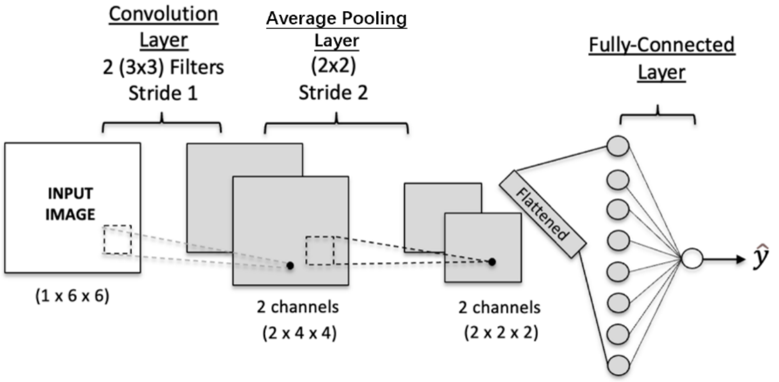
(b) Represent the neural network in a matrix form and calculate the output, providing a step-by-step explanation of the computations involved and supporting mathematical equations. [8 marks]

(c) Given that the ground truth output of the network is $y = 1$, use backpropagation to update the weights of this MLP model for one iteration using **Binary Cross-Entropy (BCE) loss** and learning rate of 0.5. Demonstrate how the gradients are computed for the weights in both the hidden layer and the output layer. (Hint: BCE loss $\mathcal{L} = -y \log \hat{y} - (1-y)\log(1-\hat{y})$ and $\nabla\mathcal{L} = \frac{\partial \mathcal{L}}{\partial \hat{y}} = -\frac{y}{\hat{y}} + \frac{1-y}{1-\hat{y}} = -\frac{1}{\hat{y}}$) [10 marks]

(d) Compute the new output $\hat{y}$ using the updated weights. [4 marks]

<u>Question 7</u> [20 marks]

We have a basic CNN architecture that takes a 6x6 grayscale image as input. The initial convolutional layer applies two 3x3 filters to produce two feature maps, followed by a ReLU activation function. The output is then downsampled using an average pooling operation. The network concludes with one fully connected layer that utilizes a sigmoid activation function to generate a single output value. Notably, all layers in the network - both convolutional and fully connected - **omit bias terms**.



(a) What is the total number of learnable weights in the initial convolutional layer? [3 marks]

(b) How many times is the ReLU activation function applied during the forward propagation of the network? [3 marks]

(c) What is the total number of weights that need to be optimized for the entire neural network? [5 marks]

(d) Using the provided 6x6 input image and the two 3x3 filter kernels, calculate and display the resulting feature maps after the convolution operation and subsequent pooling operation. [10 marks]

20	**20**	**20**	0	0	0
20	0	**20**	0	0	0
20	**20**	**20**	0	0	0
0	0	0	**20**	**20**	**20**
0	0	0	**20**	0	**20**
0	0	0	**20**	**20**	**20**

-2	0	1
-1	0	1
-1	0	2

-1	-2	-1
0	0	0
1	2	1

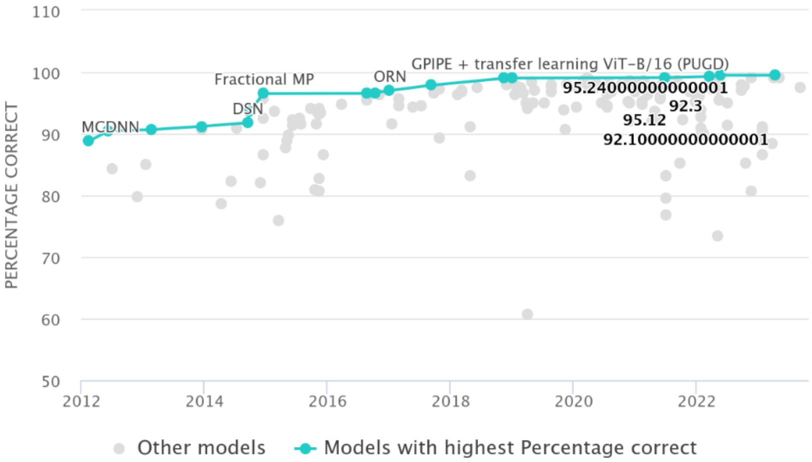
(e) Utilize the feature maps obtained from part (d) to propagate through the remaining fully connected layer with weights [0.5, -0.4, 0.3, -0.8, 2.5, -2.6, 0.9, 0.6], compute the final output $\hat{y}$ and determine the predicted classification outcome for the given 6x6 input image. [4 marks]

Section C: **Programming Assignment** [40 marks]

Objective:

In this programming assignment, students will implement a Convolutional Neural Network (CNN) model for image classification on the CIFAR-10 color image dataset. The goal is to enhance model performance by exploring different advance convolutional network architectures, optimizations, activation function, regularization, batch normalization, image data augmentation, etc. Students can study state-of-the-art deep learning models on the CIFAR-10 dataset in the figure link below, which may help you design models for this assignment.

https://paperswithcode.com/sota/image-classification-on-cifar-10?tag_filter=93%2C6%2C231%2C397%2C0



Guidelines:

A starter Jupyter notebook is provided for loading the dataset and implementing a 4-layer CNN with two convolutional layers following with pooling layers and two fully connected layers. Students should extend this notebook (or rewrite the whole notebook) to improve accuracy, F1 score, precision, and recall on the test set.

https://colab.research.google.com/drive/1W-CpyU3mwWr_ueSm_86C-do6pm361VMe?usp=sharing#scrollTo=4tPgLqmBveuL

- **Students should can train the CNN model from randomly initialized parameters or by transfer learning**
- **Students may still use well-known CNN architectures (e.g., ResNet, VGG blocks, etc.) as long as the training starts from scratch.**

The marking scheme for this programming assignment is:

- ≥83% (5 marks)
- ≥85% (10 marks)
- ≥87% (15 marks)
- ≥89% (20 marks)
- ≥91% (25 marks)
- ≥93% (30 marks)

The **top 10** performing models of the class as measured by test set accuracy will each receive an **additional 10 marks**. In the event of a tie, preference will be given to models with fewer parameters and/or lower complexity in terms of Floating-Point Operations (FLOPs).

Students are encouraged to explore advanced and well-known CNN architectures. Novel methods for model training that improve upon standard approaches are also welcome. The goal is to incentivize maximizing accuracy while maintaining efficient models in terms of size and computational requirements.

Simplified models and better coding that achieve strong results through architectural innovations or training procedures will be recognized. The aim is to push beyond standard layered CNNs through creativity. Consider how techniques like residual connections, attention mechanisms, adversarial training etc. could enable better performance on this task.

Write a summary of your model and training design as well as analysis of experimental results at the end of the notebook. Submit a zip file that consists of pdf file of your answers of the questions and the Jupyter notebook file to the Assignment 2 on CASVAS with following file name format:

- Filename format : Assignment02_StudentName_StudentID.zip
- Filename example: Assignment02_Chen_Hoi_501234567.zip