Deep Learning

Assignment2

(CNN)

Submission date: May 9, 2025

#### **Instructions:**

1. This assignment covers key concepts in deep learning: CNN, gradient descent, batch normalization, and optimizers.
2. Answer each question by completing the required tasks.
3. Submit a detailed report explaining your approaches, findings, and the results.
4. ***Plagiarism will not be tolerated. Any submission found to be copied will not be graded. Ensure that all work is your own.***

#### **Question 1: CNN (20 marks)**

Build a small CNN model consisting of 5 convolution layers. Each convolution layer would be followed by a ReLU activation and a max pooling layer. Here is sample code for building one such conv-relu-maxpool block in keras.

model = Sequential()

model.add(Conv2D(16, (3, 3), input\_shape=input\_shape))

model.add(Activation('relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

After 5 such conv-relu-maxpool blocks of layers you should have one dense layer followed by the output layer containing 10 neurons (1 for each of the 10 classes). The input layer should be compatible with the images in the **CIFAR-10**. We are using the **CIFAR-10** dataset, which contains **60,000 images** divided into **10 classes** (airplanes, cars, animals, etc.).

The code should be flexible such that the number of filters, size of filters and activation function in each layer can be changed. You should also be able to change the number of neurons in the dense layer.

(a) What is the total number of computations done by your network? (assume m filters in each layer of size k×k and n neurons in the dense layer)

(b) What is the total number of parameters in your network? (assume m filters in each layer of size k×k and n neurons in the dense layer)

#### **Question 2: CNN (20 marks)**

Implement a CNN model and perform hyperparameter tuning to optimize its performance. Experiment with different optimizers and document the impact of each hyperparameter on the model's performance.

1. Implement a CNN model for a simple classification problem (e.g., CIFAR-10 or MNIST). Train the model on a relevant dataset and visualize how the model separates the classes.
2. Experiment with different optimizers (e.g., SGD, RMSprop, Adam). Report the loss and accuracy for different optimizers. Discuss the conditions under which each optimizer performs best.
3. Document your findings clearly, including the tuning process, optimizer comparisons, and performance metrics, with relevant visualizations and tables.

#### **Question 3: Batch Normalization (20 marks)**

Extend your experimentation from question 1 and perform the following tasks

1. Train the model once with batch normalization and once without batch normalization.
2. Compare the performance of both models by plotting the training and validation loss/accuracy over time.
3. Evaluate and discuss the impact of batch normalization on model convergence, training speed, and generalization in your report.

#### **Question 4: Gradient Descent and Its Variants (20 marks)**

1. Implement the basic gradient descent algorithm. Extend your implementation to include:
   1. Stochastic Gradient Descent (SGD)
   2. Mini-batch Gradient Descent
2. Compare the performance of these variants on a chosen dataset by plotting the convergence of the loss function over time for each method.
3. Discuss the pros and cons of each gradient descent variant in your report.

#### **Question 5: Hyperparameter Tuning (20 marks)**

1. Use the KerasTuner library to find the optimal hyperparameters for a neural network model on a classification task. Perform tuning for:
   1. Learning rate
   2. Number of layers and units in each layer
   3. Batch size
   4. Dropout rate (if applicable)
2. Document the tuning process, results, and the best hyperparameters found. Include a detailed report on how the tuning process was carried out and the impact of each hyperparameter on the model's performance.

### **Submission Requirements:**

* Submit your code in a Jupyter Notebook or Python script format. Ensure your code is well-commented to explain key steps, functions, and any assumptions made.
* Include a report (in PDF or Word format) summarizing your findings, visualizations, and conclusions for each task.
* Ensure your report explains your approach, results, and the reasoning behind your choices.