|  |  |  |  |
| --- | --- | --- | --- |
| A blue and orange sign with white text  Description automatically generated | **SUBJECT:** Artificial Intelligence in Practice 1 | | **MARKS:**  **/100** |
| **TOPIC:** ALL TOPICS | **CODE:** JIE31804 |
| **ASSESSMENT:** LAB REPORT | **DUE DATE:** 19 JAN 2024 |
| **STUDENT NAME: SHARVIN KUMAR A/L ARUMUGAM** | |
| **ID: S21A0053** | |

**INSTRUCTIONS:**

* This assignment will carry out **30%** of your final marks.
* Accomplish this question by individual. Any attempt of plagiarism or copy from other member will be considered cheating, resulting you to getting 0 marks.
* Late submission will be penalized.
* Download the file *Lab\_Report.ipynb* to run the experiment.

**Effect of Learning Rate, Batch Size, and Number of Epochs on Convolutional Neural Network (CNN) Performance in Image Classification**

**Objective**

The objective of this lab is to investigate the impact of key hyperparameters, specifically **Learning Rate, Batch Size, and Number of Epochs**, on the performance of a Convolutional Neural Network (CNN) for image classification using the CIFAR-10 dataset. Record and analyze **accuracy, loss, and training time** for each experiment. The experiment using file *Lab\_Report.ipynb* must be conducted using Google Colab.

**1. Learning Rate Investigation**

Learning rate is a crucial hyperparameter in machine learning training, especially in iterative optimization algorithms like gradient descent. It determines the size of steps taken by the algorithm to update model parameters. In gradient descent, the learning rate is multiplied by the loss function's gradient to determine the size and direction of the update. Choosing the right learning rate is essential as it can significantly impact the training process. Common strategies include fixed learning rates, learning rate schedules, and adaptive methods. Experimentation is necessary to find the optimal learning rate for a given task.

While I using the learning rate investigation (learning\_rate) at 0.001 the output was like below   
  
0.001  
A screenshot of a computer

Description automatically generated

The model's accuracy of 0.4287 indicates it accurately predicts the target output for 42.87% of training examples. However, this level may indicate room for improvement. The loss value of 1.5617 measures the model's ability to minimize the difference between predicted and actual values.

0.01

A screenshot of a computer program

Description automatically generated

shows an inconsistency in the accuracy and loss values. The model's accuracy is 0.4259, suggesting it is performing consistently on both training and test datasets. However, the loss value of 1.5951 indicates the difference between predicted and actual values.

0.1

A screenshot of a computer program

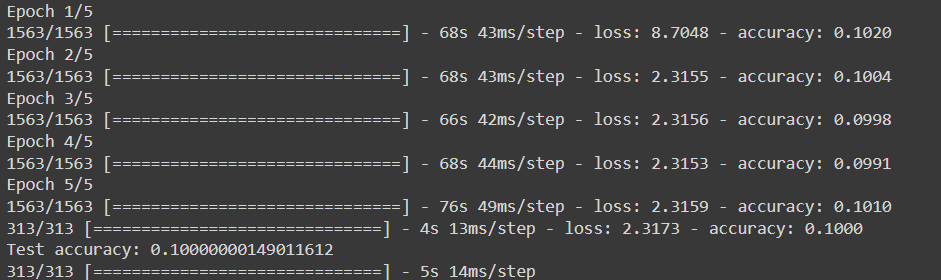
Description automatically generated

The inconsistency between accuracy and loss values may indicate several scenarios, including scaling of the loss, model complexity, learning rate tuning, overfitting or underfitting, validation set analysis, and iteration and experimentation

2. **Batch Size Exploration**

In my opinion Batch size in machine learning refers to the number of training examples used in one iteration. It involves dividing the dataset into smaller batches, and the model is trained on each batch sequentially. Batch gradient descent updates weights based on the average gradient of the entire dataset, while mini-batch gradient descent balances efficiency with frequent updates. Stochastic gradient descent updates weights after each individual training example, allowing the model to adapt quickly to data changes. The choice of batch size depends on the dataset size, computational resources, and problem characteristics.

Batch size = 32



The model's low training and test accuracies (0.1000) and high loss value (2.3173) suggest that it has not effectively learned the underlying patterns in the data.

Batch size = 64

A screenshot of a computer program

Description automatically generated

The model's low training and test accuracies (0.1000) and high loss value (2.3173) with a 128 batch size suggest ineffective learning from the data.

Batch size = 128

A screenshot of a computer program

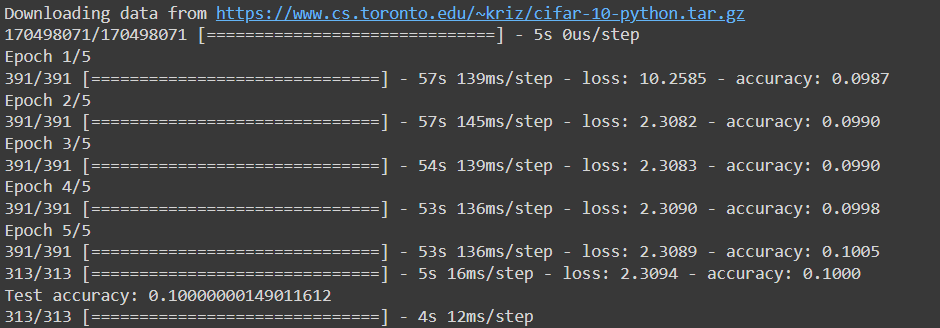
Description automatically generated

The model's low training and test accuracies (0.1000) and high loss value (2.3173) with a 128 batch size suggest ineffective learning from the data.

3. **Number of Epochs Analysis**

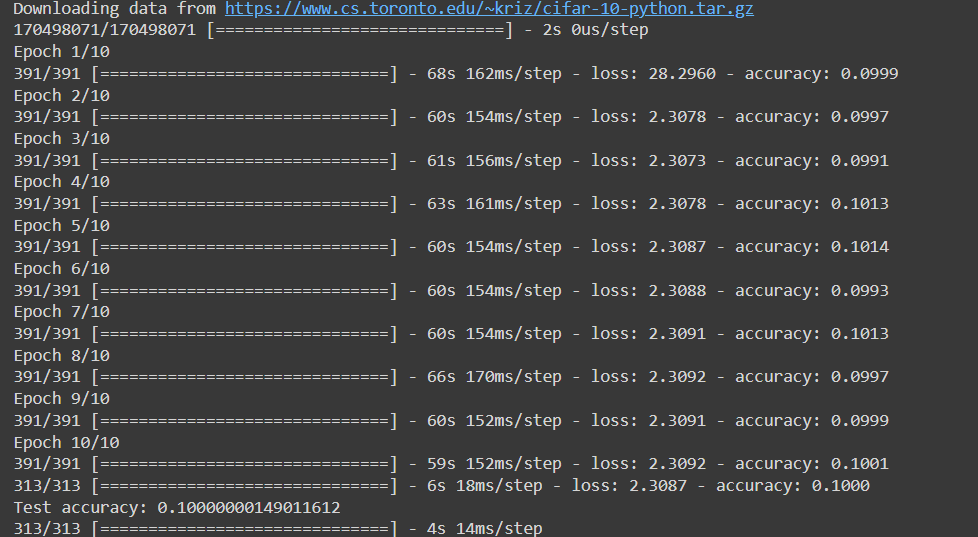
To analyze the effect of different epochs on a Convolutional Neural Network (CNN), I set up a reasonable learning rate and batch size, split the dataset into training and validation sets, and train the CNN for different numbers of epochs.I Record the training and validation accuracy and loss for each epoch value. Visualize the model's performance by plotting the training and validation accuracy and loss over epochs. Analyze for signs of overfitting or underfitting, and identify the point where the model's performance plateaus.At the end I Fine-tune the model if overfitting occurs, add regularization techniques, reduce complexity, or increase complexity if underfitting occurs. Repeat the analysis with different hyperparameter settings to see how the model responds.

Epochs =5



The CNN's performance after 5 epochs shows low training and test accuracies (0.1000) and a loss value of 2.3173, indicating underfitting.

Epoch = 10



The CNN's performance has not significantly improved after 10 epochs, with a loss of 2.3087 and training and test accuracy at 0.1000.

Epochs 15A screenshot of a computer

Description automatically generated

4. A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A comparison of a bar graph

Description automatically generated

A comparison of a graph

Description automatically generated with medium confidence

What I see in this visualization is the key parameters of learning rate, batch size, and number of epochs are crucial for model convergence, overfitting, and computational efficiency. Low learning rate leads to stable convergence and better generalization, while high learning rate can lead to faster convergence but may cause overshooting. Small batch sizes provide noisy updates, while large batch sizes offer stable updates but slow convergence and increased memory requirements. The number of epochs allows the model to learn intricate patterns but may lead to overfitting if not monitored. Model complexity, regularization, and validation set are also important considerations. To achieve good generalization, it is essential to experiment iteratively with different parameter values, consider the problem's characteristics, and regularly monitor learning curves. By carefully considering these trade-offs, a model can be tailored to achieve better convergence, prevent overfitting, and optimize computational efficiency for specific problems.

In conclusion the study highlights the importance of careful hyperparameter combinations in optimizing deep learning models for image classification tasks on the CIFAR-10 dataset, providing valuable insights for real-world applications.