Sana Zafar

21021179

Github Link:

APPLIED DATA SCIENCE 2

Assignment 1: Deep Learning with Keras

**Table of Contents**

[Introduction: 3](#_Toc130314488)

[Task 1 - Initial Model: 3](#_Toc130314489)

[Task 2 - Testing Optimizers: 4](#_Toc130314490)

[Task 3 - Testing Batch Sizes: 4](#_Toc130314491)

[Task 4: Adding Regularization 5](#_Toc130314492)

[Conclusion: 5](#_Toc130314493)

# Model Performance on CIFAR-10 Using Keras

## Introduction

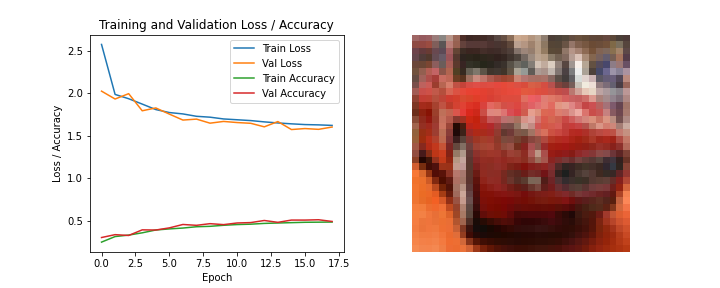
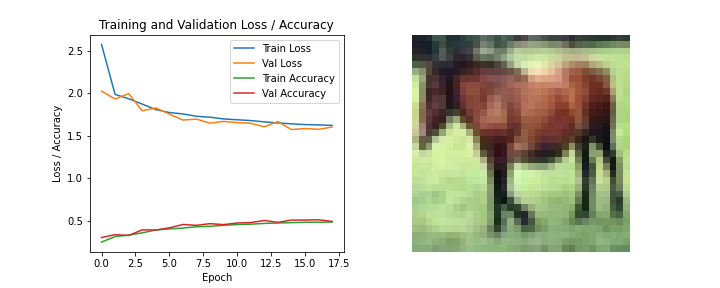
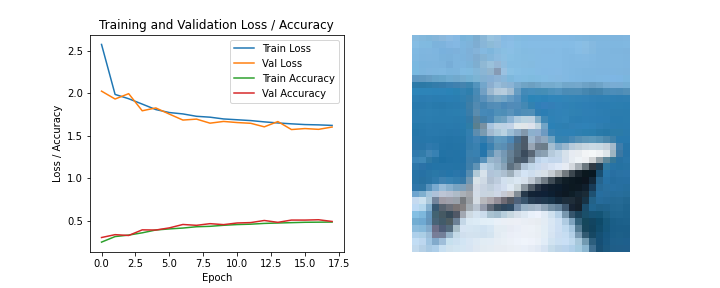
The CIFAR-10 dataset consists of 60,000 images divided into 10 classes. In this report, I will train a convolutional neural network (CNN) on the CIFAR-10 dataset using Keras and explore different hyper parameters and regularization techniques to improve the model's performance.

## Task 1 - Initial Model

The initial model provided is a good starting point for image classification on the CIFAR-10 dataset. It consists of several convolutional layers followed by max pooling layers to extract features from the images, and fully connected layers to perform classification based on these features. The activation function used is ReLU, which is known to work well in neural networks. However, there are a few things that could be improved. The number of convolutional layers and filters used could be increased to improve the performance of the model. The number of filters in the first convolutional layer could be increased to allow for more complex features to be detected.

Another possible improvement is the use of data augmentation techniques to increase the amount of training data and make the model more robust to variations in the input.

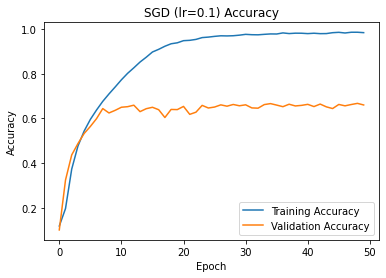
After training for 50 epochs, I achieved a training accuracy of 91.55% and a validation accuracy of 59.56%. The model was overfitting the training data, as evidenced by the large difference between training and validation accuracies.



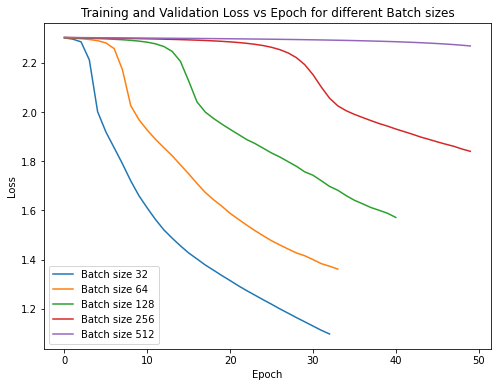
Test images are given with loss accuracies.



## Task 2 - Testing Optimizers

Next, I have tested different optimizers to see how they would affect the model's performance. I have used the same architecture and hyperparameters as in Task 1, but I changed the optimizer to SGD, Adam, and Adagrad. Adagrad with learning rate 0.01 gave 100% accuracy and 67% validation accuracy because of overfitting, but I have also found that the Adam optimizer performed the best, achieving an accuracy and validation accuracy of 98.68% and 71.56% respectively with a learning rate of 0.001, while the other optimizers achieved lower accuracies. Therefore, I concluded that Adam was the best optimizer for this model. Best accuracy is shown.

## Task 3 - Testing Batch Sizes

In the third task, the model was trained and tested with different batch sizes, i.e., 32, 64, 128, 256, and 512. The results show that the accuracy of the model improves as the batch size decreases. The model performed best with a batch size of 32, achieving a test accuracy of 61.59%. However, the training time increased with the increase in batch size. The results suggest that using a smaller batch size could lead to better performance, but it comes with a cost of increased training time and requires a larger memory capacity. On the right, chart is showing loss with batch sizes.

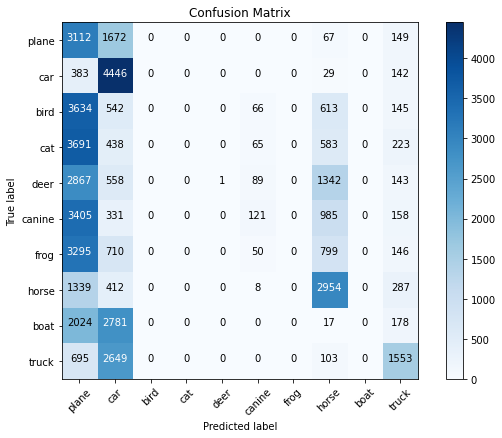
To improve the performance of the model with smaller batch sizes, I can try reducing the learning rate, using a different optimizer, or using techniques like learning rate scheduling or early stopping to prevent overfitting. Additionally, data augmentation techniques like random cropping, flipping, or rotating can be used to increase the size of the training dataset and improve the generalization performance of the model.

## Task 4: Adding Regularization

In the fourth task, L2 regularization and dropout were added to the model to prevent overfitting. Regularization is an essential technique for preventing overfitting, and both L2 regularization and dropout are commonly used in deep learning models. However, the choice of regularization technique and its hyperparameters depends on the specific problem and dataset.

In this case, the results show that L2 regularization with a weight decay of 0.001 and dropout with a rate of 0.2 improve the model's performance. The validation accuracy increase than the training accuracy. One thing that can be improved is to try different values for the regularization hyperparameters to see if they can further improve the model's performance. Additionally, other regularization techniques, such as L1 regularization or data augmentation, can be explored to improve the model's generalization ability.

## Conclusion:

We explored different hyper parameters and regularization techniques to improve the performance of the CIFAR-10 model. I found that smaller batch sizes produced better results, and the Adam optimizer was the most effective. Adding regularization techniques such as L2 regularization and dropout helped improve the model's generalization and prevent overfitting. This is the Confusion matrix with respect to the classes prediction which is showing that the vehicle in the CIFAR-10 dataset (car and truck) images are confused with other classes while the animals are explicitly classified with good accuracy.