Aston University

Student Name: Muhammad Aizaz Afzal

Student Number: 200261713

Module: EE4108 – Machine Learning

Program: MSc SmartNet (EMJMD)

Machine Learning Project

Contents

1. Abstract

The main task achieved in this project was to implement a convolutional neural network inspired by AlexNet architecture [1] for classifying images from CIFAR 10 dataset using 8 layered network which employed different techniques to increase the training, validation, and testing accuracy to over 80%.

2. Introduction

The initial model which is very close to AlexNet was initially simulated in google colab [2] using Keras library which acts as an interface for backend TensorFlow. The CIFAR 10 data set contains 60,000 pictures which are 32 x 32 pixels wide and are divided into training, validation, and testing subsets. The initial model gives very high training accuracy (99%) but very low validation and testing accuracy (around 71%) which showed there was plenty of room for improvement hence various tasks were done sequentially such as adding dropout, batch normalizing, and image augmentation etc.

Since the operations were extremely complex therefore the later part of the project was done in Kaggle notebook [3] since its faster than google colab for higher number of epochs which enables the model to reach testing accuracy up to 82%.

3. Methods

The CIFAR 10 dataset [4] consists of 10 classes such as cat, bird, airplane etc. and so every class has 6000 images. Training set has 50,000 images and 10,000 images are in testing set, a subset of 10,000 images from training set is extracted as the validation set during pre-processing stage.

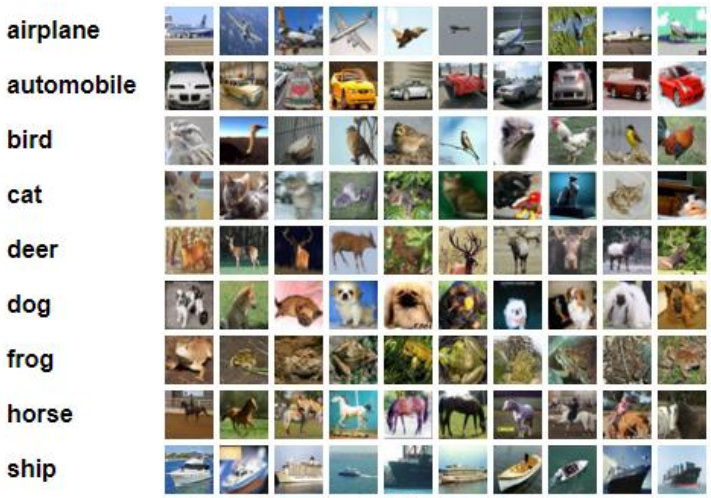

Figure 1 CIFAR 10 Dataset

In the preprocessing block, the dataset was first normalized so that the RGB channel would range values from 0 to 1 instead of 0 to 255, secondly, one hot encoding was performed so that processing and training dataset becomes easier.

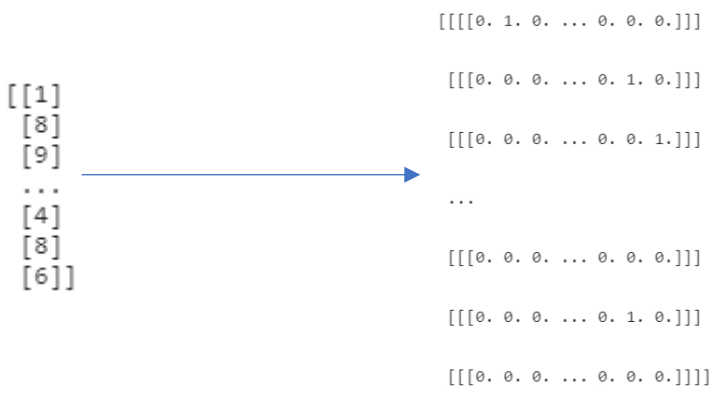

Figure 2 One hot encoding

Firstly, the base model inspired by AlexNet with 50 epochs is implemented. The convolutional neural network consists of 5 convolutional layers having kernel size 5 by 5 with ReLU activation function which activates nodes having a certain minimum probability only and maxpooling is done to down sample/decrease dimensionality and to select the largest values on feature maps and input these for next layer [5], next 3 layers are fully connected with the last layer having softmax activation function. Batch size was fixed at 128 while adam optimizer was used since it has better performance for our dataset as compared to RMS. Through hit and trial method it was found that kernel size 3 by 3 for convolutional layers gives the best performance.

A huge gap between training and validation accuracies indicated overfitting in task 1 hence for the 2nd task, dropout layers were introduced which are to reduce overfitting of the data by intentionally dropping out nodes with certain dropout probability, using hit and trail method it was found that, for spatial dropout the best dropout probability is 0.3 and for regular dropout its 0.5 in most cases, more proof on these is given in result section.

For the 3rd task, batch normalization was introduced in 5 convolutional and 2 fully connected layers, it basically standardizes the inputs i.e., shifts the mean to zero and makes standard deviation equal to 1, it has a regularization effect [6] which helped in increasing the testing accuracy by 3-5%. In the next task, all max pooling layers are removed, and all the layers are made as convolutional layers, this resulted in a drop in overall testing accuracy. In the final task, Image augmentation is used to further enhance the model by training the model with rotated, shifted, and scaled versions of the same images so that the model is better equipped to connect patterns and identify the correct class for every image in testing set.

4. Results

For the performance analysis, we have 2 parameters to judge the performance of the model which are accuracy and loss.

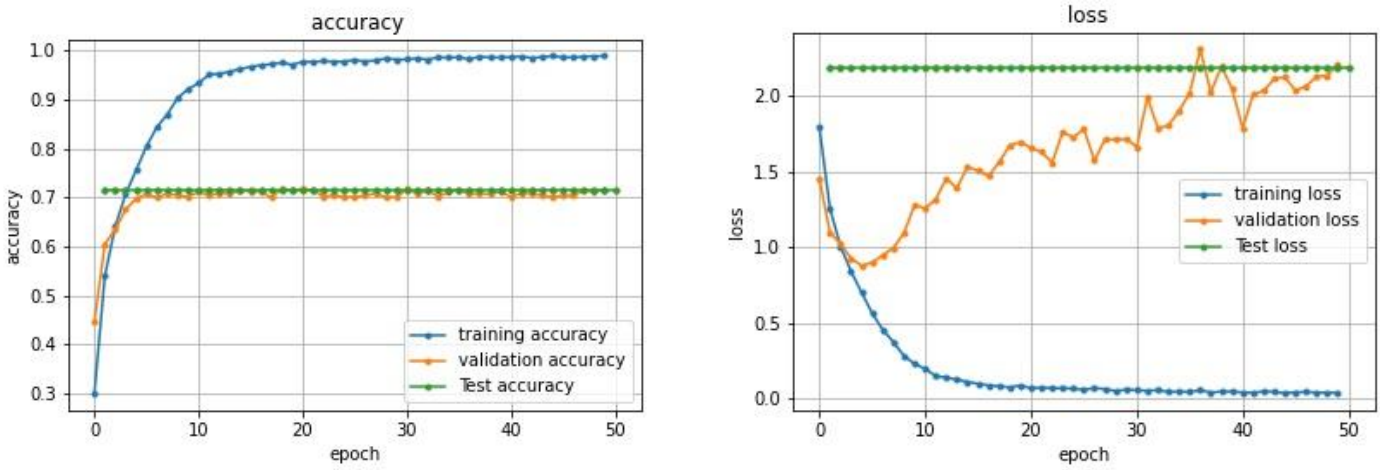

Figure 3 Task 1 accuracy and loss

Base model was implemented for task 1 therefore the testing accuracy is just 71% and overfitting is clearly visible because of multiple layers there are too many nodes and features, more than necessary.

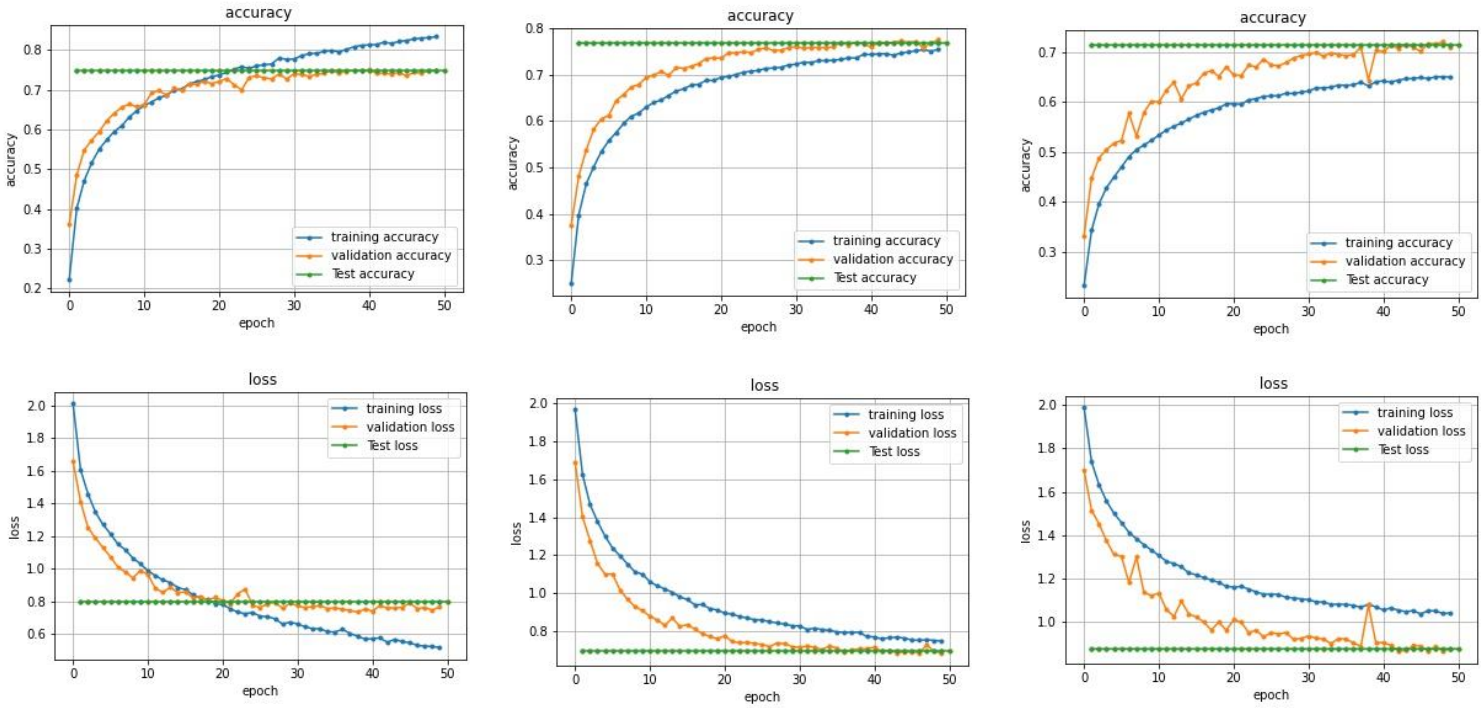

Figure 4 Task 2 accuracy and loss with dropout probabilities 0.2, 0.3 and 0.4 respectively

Introducing dropout helps mitigate the issue of overfitting, hit and trial method shows that dropout probability of 0.3 is the best for our model which gives testing accuracy of 77%, a 6% increase from base model.

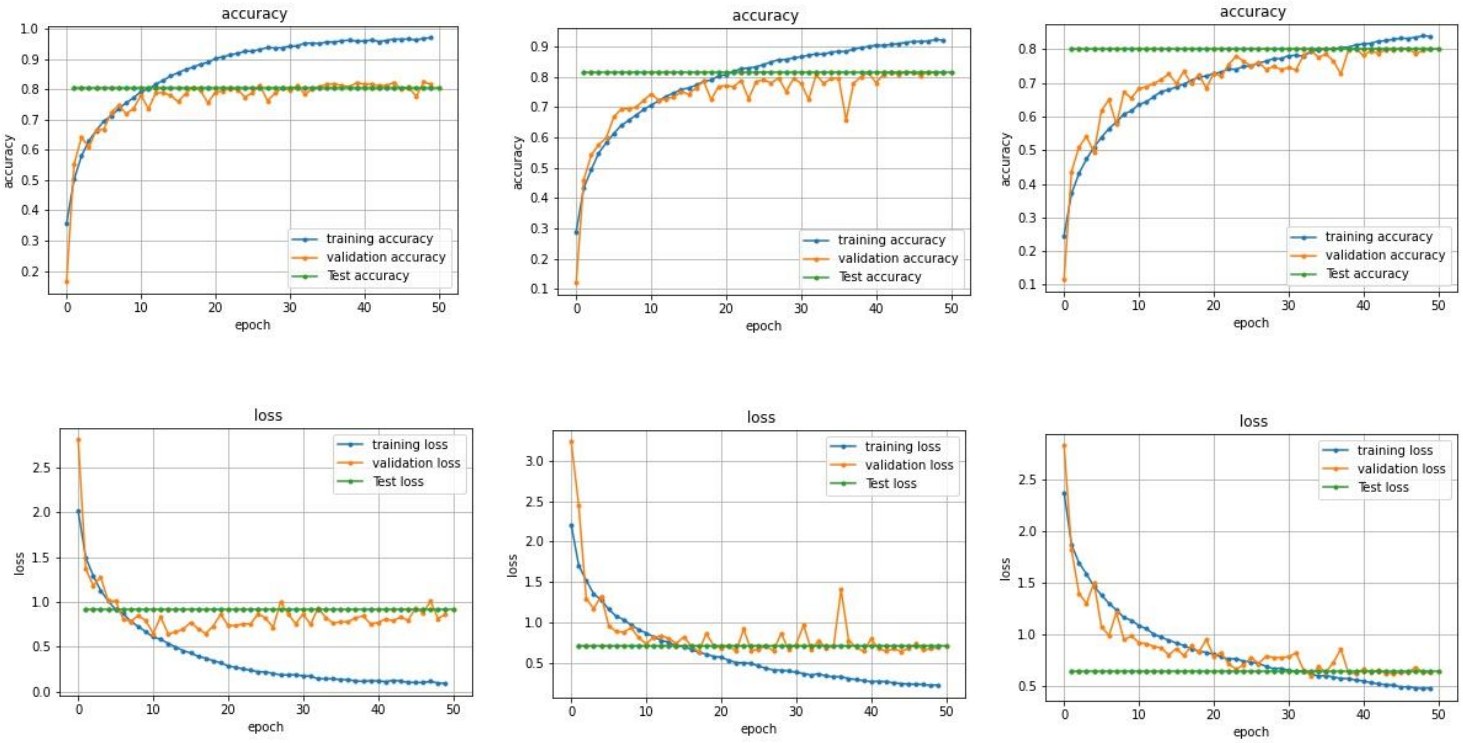

Figure 5 Task 3 accuracy and loss with dropout probabilities 0.2, 0.3 and 0.4 respectively

With batch normalization, improved results were obtained, here again a dropout probability of 0.3 gave the best result, a testing accuracy of 81% although a hint of overfitting is visible from the graph.

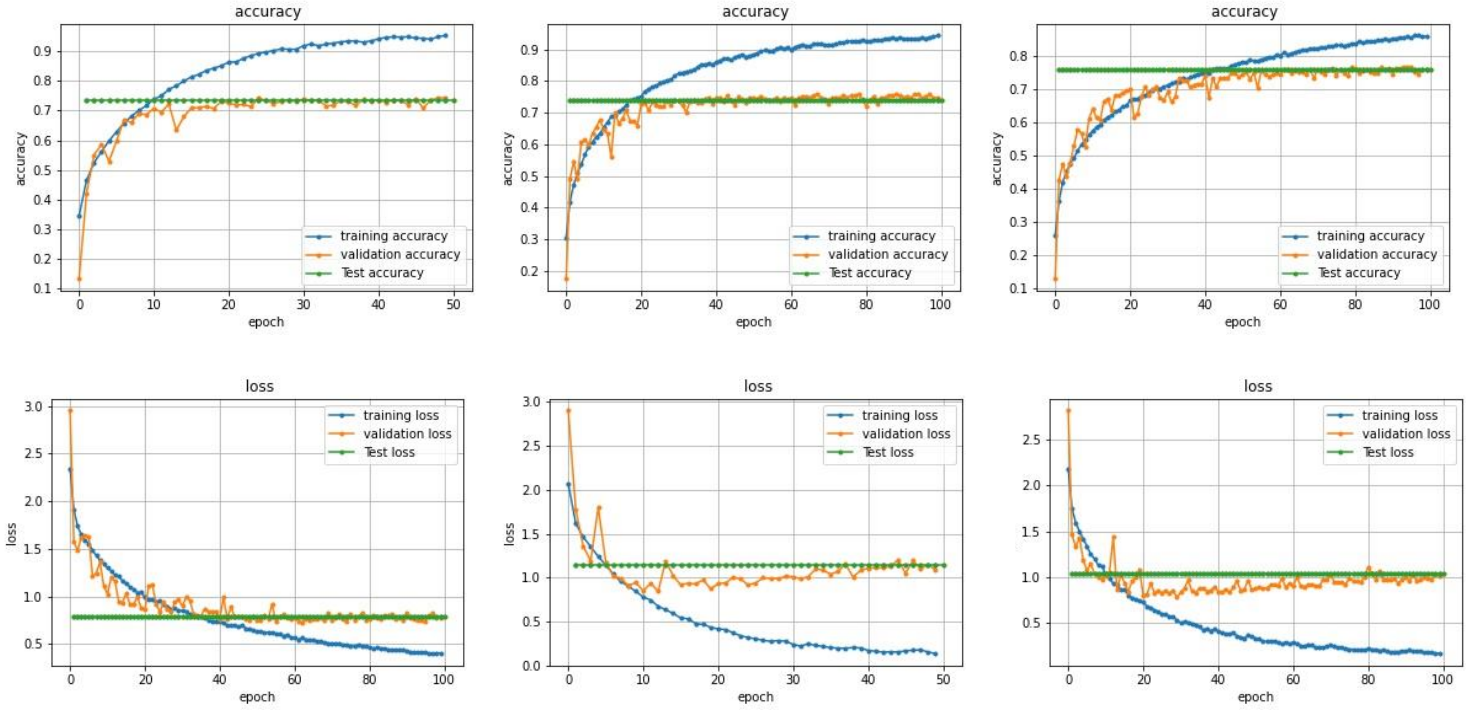

Figure 6 Task 4 accuracy and loss with dropout probabilities 0.2, 0.3 and 0.4 respectively

When the max pooling layers were removed, the model's performance depreciated, and clear signs of overfitting were visible, this time the dropout probability of 0.4 gave the best result i.e., least overfitting with a testing accuracy of 76%, a 5% drop from task 3.

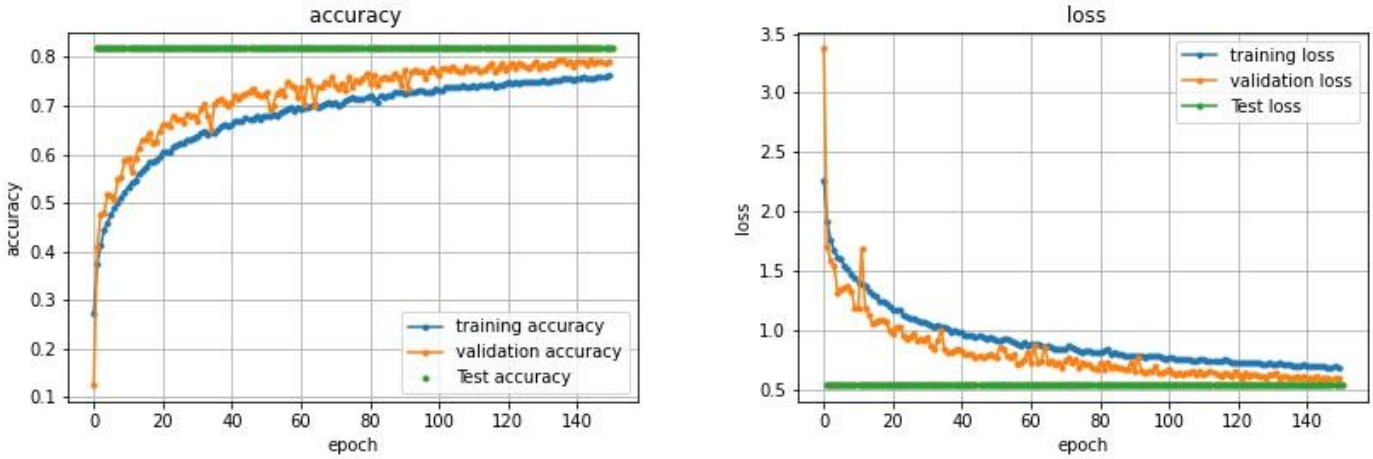

Figure 7 Task 5 accuracy and loss with dropout probability 0.4

In the final task, image augmentation was performed, the images were randomly rotated, scaled, shifted vertically and horizontally, it was done on training, validation and testing sets respectively which improved the overall performance of model since it had more instances of correlating different features of each picture. It increased the testing accuracy to a highest of 82%.

5. *Conclusions*

It can be concluded that the model inspired by AlexNet is quite robust and accurate even though the dataset was quite small. All the layers/functions applied did produce the desired result although there is a lot of room for improvement such as re-applying max pooling layers with image augmentation, it is expected to increase the testing accuracy by 6-7% which will make the model almost 90% accurate, any further tweaking with parameters might cross the 90% accuracy mark. Increased dataset, number of epochs and batch size can also help increase the testing accuracy of the model.

6. *Bibliography*

[1] Krizhevsky, A., Sutskever, I. and Hinton, G., 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), pp.84-90.

[2] Colab.research.google.com. 2021. *Google Colaboratory*. [online] Available at: https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/quickstart/beginner.ipynb

[3] Kaggle.com. 2021. *Run Data Science & Machine Learning Code Online | Kaggle*. [online] Available at: https://www.kaggle.com/code

[4] Cs.toronto.edu. 2021. *CIFAR-10 and CIFAR-100 datasets*. [online] Available at: https://www.cs.toronto.edu/~kriz/cifar.html

[5] Computersciencewiki.org. 2021. *Max-pooling / Pooling - Computer Science Wiki*. [online] Available at: https://computersciencewiki.org/index.php/Max-pooling_/_Pooling

[6] Thakkar, V., Tewary, S. and Chakraborty, C., 2018. Batch Normalization in Convolutional Neural Networks — A comparative study with CIFAR-10 data. *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*,.

7. Appendix

Task	epochs	Kernel size (Layer)					Batch size	Dropout		Batchnorm	Max pooling	Image Augmentation	Accuracy			
		1	2	3	4	5		2D	regular	(Y/N)	(Y/N)	(Y/N)	Train	Valid	Test	
task 1	50	5	5	3	5	5	128	no	no	no	yes	no	0.99	0.71	0.71	
task 2	50	5	5	3	5	5	128	0.2	0.5	no	yes	no	0.86	0.75	0.75	
	50	**3**	**3**	**3**	**3**	**3**	**128**	**0.3**	**0.5**	**no**	**yes**	**no**	**0.75**	**0.77**	**0.77**	**best case**
	50	3	3	3	3	3	128	0.4	0.5	no	yes	no	0.65	0.72	0.72	
task 3	50	3	3	3	3	3	128	0.2	0.5	yes	yes	no	0.91	0.81	0.8	
	50	**3**	**3**	**3**	**3**	**3**	**128**	**0.3**	**0.5**	**yes**	**yes**	**no**	**0.91**	**0.81**	**0.81**	**best case**
	50	3	3	3	3	3	128	0.4	0.5	yes	yes	no	0.84	0.8	0.8	
task 4	50	3	3	3	3	3	128	0.2	0.5	yes	no	no	0.95	0.74	0.73	
	100	3	3	3	3	3	128	0.3	0.5	yes	no	no	0.94	0.75	0.76	
	100	**3**	**3**	**3**	**3**	**3**	**128**	**0.4**	**0.5**	**yes**	**no**	**no**	**0.86**	**0.76**	**0.76**	**best case**
	100	3	3	3	3	3	128	0.5	0.5	yes	no	no	0.74	0.74	0.74	
task 5	**150**	**3**	**3**	**3**	**3**	**3**	**128**	**0.3**	**0.5**	**yes**	**no**	**yes**	**0.79**	**0.82**	**0.82**	**best case**