Deep Learning
Practical 1

Loran Oosterhaven(S2707888) [a]   Battu Surender Harsha(S4120310) [b]

[a] *University of Groningen, Groningen, The Netherlands*

1 Introduction

The goal of this practical is to better understand and experiment with different Deep architectures such as AlexNet and VGG and their hyperparameters. In this practical 8 different configurations were tested on the same dataset using the same sampling method and randomization seed. Our focus was primarily on the two architectures and on Adam and SGD Optimizers. In several configurations we work with augmented data. We used Pytorch to run the experiments on a jupyter notebook.

2 The Data

2.1 Dataset

In this practical the CIFAR-10 dataset was used. This dataset is commonly used to train machine learning algorithms for image recognition. It contains 60,000 colored images of 32x32 pixels consisting of 10 different classes. Among these classes classes are cars, birds, cats and 7 seven other generic objects. There are 6,000 images per class. The data is normalized and converted to torch tensors when it is loaded

2.1.1 Data Augmentation

In some of the experiments we have augmented the data(each image) using torch transforms in this order:

- Random Horizontal Flip
- Random Rotation by 10 Degrees
- Padding the image by size 3 (image size is 32x32)

Then the data is normalized and converted to tensors for the experiment.

3 Models and Optimizers

3.1 AlexNet

AlexNet is a convolution neural network (CNN) architecture . The network consists of eight layers. The first five are convolution layers of which some of them are followed by a max-pooling layer. The last three layers consist of fully connected dense layers [2].

3.2 VGG

VGG is also a CNN. It has been developed by Visual Geometry Group from Oxford, hence the abbreviation VGG. The variant of VGG we use will be VGG11, which contains 11 layers and 133 millions parameters [4].

3.3 Adam optimizer

Adam is an optimizer which has an adaptive learning rate, it is designed specifically for deep neural networks [1]. Adam is the default and the most common optimizer used in deep learning tasks.

3.4 SGD

Also called Stochastic Gradient Descent, is an old algorithm for training neural networks with a fixed learning rate [3]. It is an older algorithm, but still effective, we have used a learning rate of 0.1 to speed up learning in our experiments

4 Experiment setup

The data has been split into train,test and validation.

- Number of training examples: 45000
- Number of validation examples: 5000
- Number of testing examples: 10000

We use Cross Entropy Loss(not categorical cross entropy as we do not softmax the last layer) as a loss function to optimize. We use accuracy

$$\texttt{accuracy} = \frac{\texttt{correct predictions}}{\texttt{total predictions}}$$

as an evaluation metric. We run each experiment for 10 epochs, and stored the metrics. Here are the settings that we have used in our 8 experiments:

1. AlexNet using Adam Optimizer and CIFAR-10 Data
2. AlexNet using SGD Optimizer and CIFAR-10 Data
3. AlexNet using Adam Optimizer and Augmented CIFAR-10 Data
4. AlexNet using SGD Optimizer and Augmented CIFAR-10 Data
5. VGG11 using Adam Optimizer and CIFAR-10 Data
6. VGG11 using SGD Optimizer and CIFAR-10 Data
7. VGG11 using Adam Optimizer and Augmented CIFAR-10 Data
8. VGG11 using SGD Optimizer and Augmented CIFAR-10 Data

5 Results

We evaluated each setting with a test set, as shown in the Table 1

Settings	Test Accuracy
AlexNet+Adam+Data	63.25
AlexNet+SGD+Data	53.2
AlexNet+Adam+AugmentedData	69.8
AlexNet+SGD+AugmentedData	57.11
VGG11+Adam+Data	68.3
VGG11+SGD+Data	62.46
VGG11+Adam+AugmentedData	9.95
VGG11+SGD+AugmentedData	60.74

Table 1: Showing Test results for each experiment

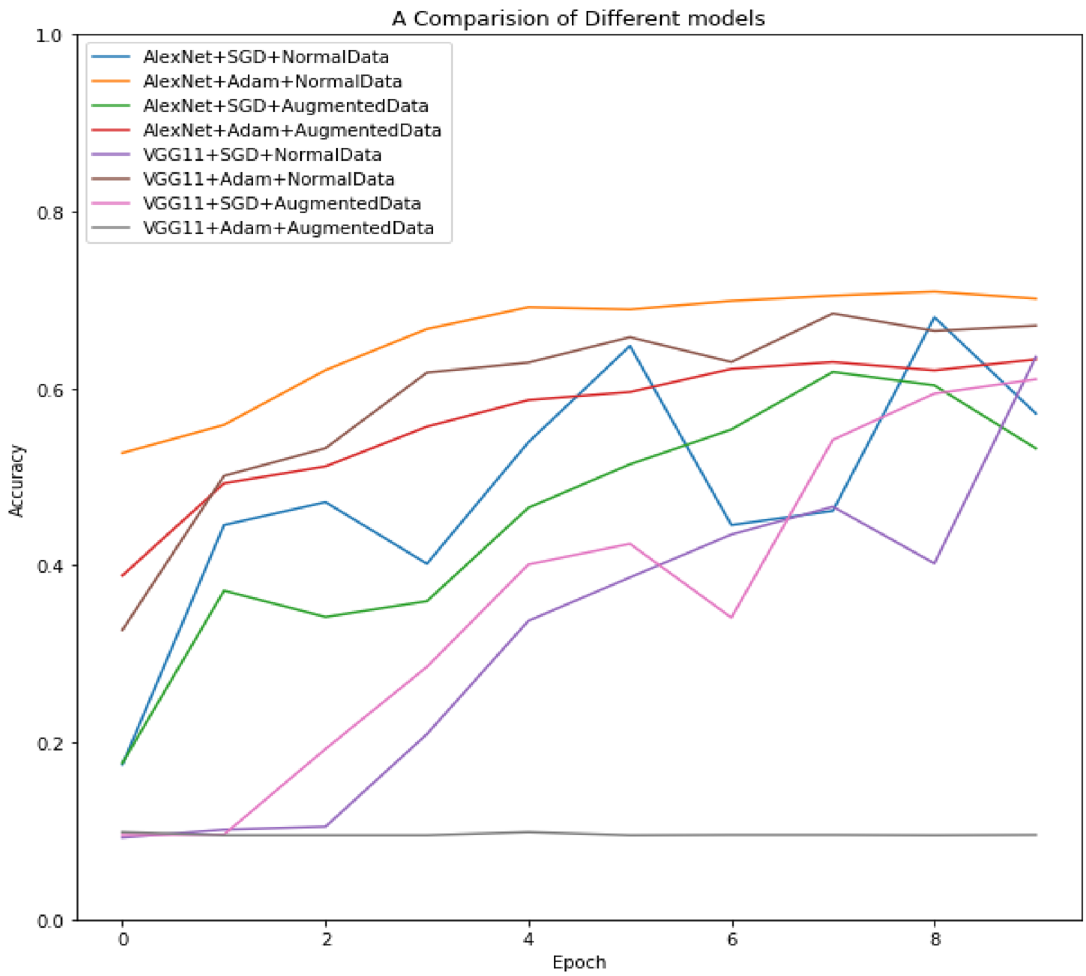

Figure 1: A figure that compares the validation accuracy at every epoch of the experimental settings

The results in terms of accuracy is presented in Figure 1. As can be seen all configurations and architectures end up with an accuracy between approximately 50% and 70%. Nevertheless, AlexNet with Adam optimizer and non augmented data immediately outperforms all other configurations and architectures. The worst performer overall is VGG11 using Adam optimizer and augmented data during several epochs. By inspecting the graph, we observe that overall, the perfomance is worse for Augmented Data, nevertheless augmented data had equal perfomances to normal data in cases like VGG11 and SGD Optimizer.

5.1 Discussion

From the results shown, we can establish Adam is a better optimizer compared to SGD. We provided random transformations on the data to observe its effect on learning, surprisingly the models were able keep with up with their counterparts that learned on normal data, especially VGG11 with SGD Optimizer actually seemed to perform better than its counterpart. This can be purely coincidental. VGG11 overall had worse perfomance on Augmented data,especially with the Adam optimizer, meanwhile VGG11 with Adam optimizer and normal data performed the second highest, which shows that Adam and VGG11 and Adam and Alexnet work well together, but the data they get trained on might be a root cause of failure in their perfomance.

References

[1] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25, 01 2012.

[3] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.