Image Classification With CIFAR10-Dataset

Erik Eliasson
DV2578
Blekinge tekniska högskola
Karlskrona
Erik.eli@live.se

I. Introduction

This report will cover the project of image classification with the help of the Cifar10 dataset [1]. The main part of the machine learning uses the library TensorFlow and their sub library Keras to perform different actions and there will be discussions of how an increase of parameters in a neural network increases the performance of the network and as well compare the two optimizers Adam and RMSprop. As well comparing both the sizes of the neural network and the optimizers to see if a combination of them could lower one of the issues within image classification, intra-classing. [2]

II. Related work

Image classification and recognition is a well explored subject within machine learning that got several different types of datasets and different types of libraries. The main source of inspiration for this project is the website "Data flair" that uses the Cifar10 dataset and Keras to perform the machine learning [3].

But there is many more types of image classification and recognition libraries and datasets. One of the more advance types of image classifications is the open-source Mask_RCNN library that can identify several different types of objects within a picture [4].

The main challenges of image classification are divided into six categories [5]. The first one is intra-class variation, as an object can have several types of under categorization that looks vastly different. Such as an office chair and a camping char looks very difference, but both are chairs.

The second one is scale and this is self-explained, the scale of the objects impacts the machine learning a lot. The third issue is point of view of the object as several objects looks different depending on what point of view you have of the object.

Forth issue is occlusion, when the object is not shown in its entirety and only shows half the object it can be very hard to detect accurately what the object is. The fifth and sixth is illumination and background clutter as this impacts the colors of the image and makes it hard to connect the image to other similar objects if there is a big difference in illumination or a lot of objects in the background.

III. Method

The first step of the project was to decide what type of dataset was the best fit for the task. With some researching the Cifar10 dataset was the best fit as it was easy to download with the help of the library Keras and as it had a lot of documentation and usage in previous work of others it made a lot of sense to use for my image classifier.

As Keras made it simple to load this dataset the decision was made to use Keras as well for this reason as it gives access to a sequential model that it is easy to add layers onto and gives access to a great number of optimizers.

With the decision of using Keras the next step was to decide what optimizers and loss function was the optimal for the image classification task. The optimizer of choice was the Adam optimizer, this is a stochastic gradient descent method that bases on adaptive estimation of first- and second-order moments. The reason why this was chosen was that it had a small memory requirement and as the task is performed on a normal pc this could benefit the performance of training time.

But I also decided to compare the Adam optimizer to another popular optimizer which is the RMSprop optimizer. the gist of RMSprop optimizer is to maintain a moving average of the square of the gradients and divide the gradient by the root of this average. It also uses plain momentum instead of Nesterov momentum.

The loss function that was decided to be use was Categorical-Crossentropy as this was a good loss function to use when there are two or more label classes and as Cifar10 uses 10 different labels this seemed to be the optimal choice.

When choosing how to train the model the decision was made to use a neural network as most documented usage of the cifar10 dataset was using the Keras neural network and this choice gave a lot of material to use and research to find different type of layers to use to try to optimize the result.

The next step was defining the layers of the smaller neural network, the layers are used as weights and is a core part of later predicting an image. The first layer uses the whole 30x30 pixels of the image and later sends it to another layer with half the size, 15x15. This layer takes the maximum value of a 2x2 area and saves it to the next step.

This step is repeated three times as the image is scale down to a 4x4 image. The image is then flattened and sent to three different dense layers with the sizes 256, 128 and 10. The last size is a refence to the number of classes and is the layer as well that decides what the neural network think the image is. This results in a total of 258 186 trainable parameters.

The bigger model uses the similar layers that shrinks the image but uses one large dense layer instead of three smaller one. This results in roughly one million parameters more at 1 250 282 trainable parameters.

But before doing the training an early stopping function was implemented to stop the training if overfitting was detected.

IV. RESULT & ANALYSIS

A. *Accuacy*

1) *Smaller neural network*

When comparing the optimizer with the smaller network, Adam is the better optimizer as it both reached higher accuracy while also being able to be trained for longer without reaching a state of overfitting, as seen in figure 1.

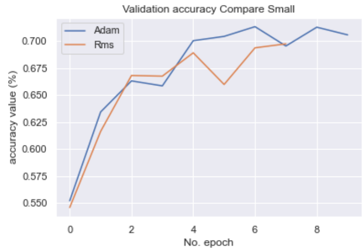

Figure 2. Validation accuracy for the smaller neutral network.

2) *Bigger neural network*

When comparing the optimizers with the bigger network though it was a much closer accuracy but still was Adam able to be trained for longer and at the end reached a higher accuracy, as seen in figure 2.

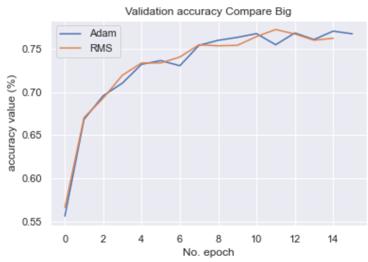

Figure 1. Validation accuracy for the bigger neutral network.

3) *Compering bigger and smaller neural network*

Comparing the bigger and smaller neural networks gave huge differences though as seen in figure 3 and 4. It is clear though that Adam did not benefit as much as RMSprop from the increase of parameters as RMSProp made the biggest increase. But the increase of roughly 70% accuracy of both optimizers with the small optimizers to an accuracy of a tiny bit higher than 75% is a great result.

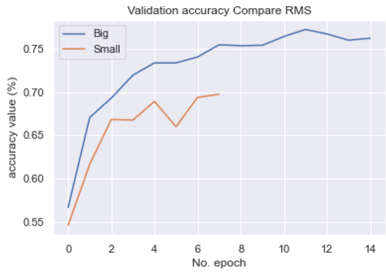

Figure 3. Validation accuracy big vs small neural network with RMSProp as optimizer

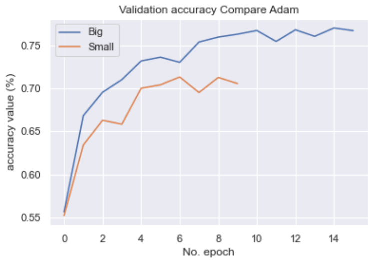

Figure 4. Validation accuracy big vs small neural network with Adam as optimizer

B. *Loss*

Just as the accuracy, the loss became lower when using the bigger neural network with both optimizers. Down from 90% loss with the smaller network to 70% with the bigger network when using RMSProp and with Adam it goes from 95% loss with the smaller network to 70% with the bigger network.

Here we also see that with the smaller networks the RMSprop got a better loss % compared to the Adam optimizer.

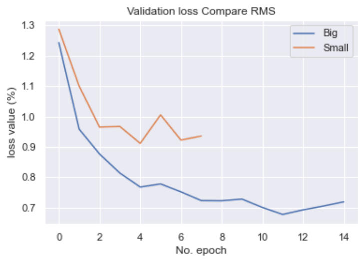

Figure 5. Validation loss big vs small with RMSProp as optimizer

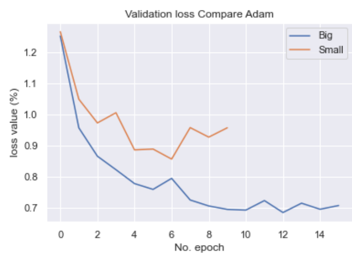

Figure 6. Validation loss big vs small with Adam as optimizer

C. *Time*

One big difference when training the two neural networks where time as the bigger network took on average 1min10s per epoch and the smaller network only took 20s per epoch. This is an increase of 350% when training the models.

D. Confusion matrix result

1) Smaller neural network

When comparing the smaller neural networks we can see that both are having a hard time trying to identify cats but with RMSProp there are 50 more cats identified correctly. But Adam has three classes with over 800 identified correct while RMSProp only has one.

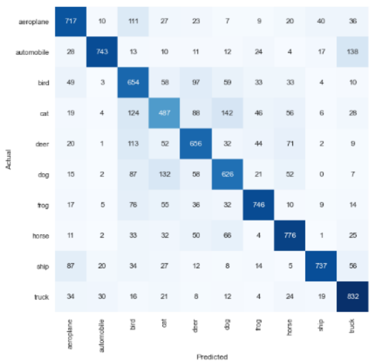

Figure 7. Small neural network with RMSProp as optimizer

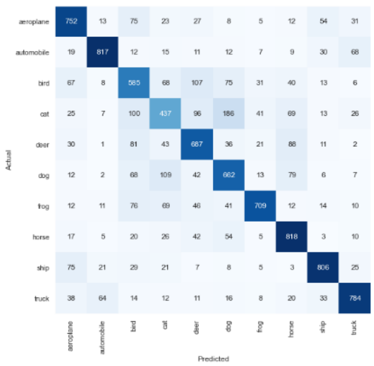

Figure 8. Small neural network with Adam as optimizer

2) Bigger neural network

When using the bigger networks the RMSProp is the one having issues with the cats while the Adam optimizer is still having six classes over 800 correct guesses while RMSProp only has four.

Figure 9. Big neural network with RMSProp as optimizer

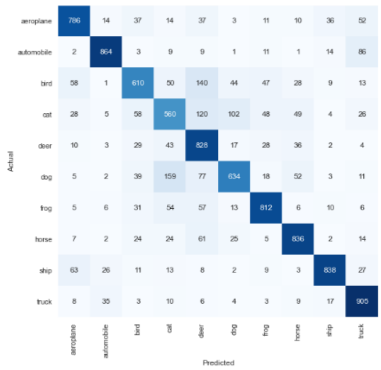

Figure 10. Big neural network with Adam as optimizer

V. Conclusions

The first conclusion that is clear is that increasing the number of parameters in a neural network greatly increased the accuracy of the trained model while also a hefty increase of training time. The choice of optimizer was still interesting as when reading the confusion matrixes it was clear that Adam was a lot better at identifying specific classes while RMS is getting its accuracy by being decent with several classes.

When reflecting about the issues with image recognition its clear that cats and dogs fall can fall under the first issue, intra-class. As both cats and dog are very similar animals and on every confusion matrix cats and dogs are the most mistaken classes mistaken for each other.

But by using the RMSProp optimizer and a bigger neural network we can see that there are not as many dogs identified as cats and this is a huge step forward and as RMSProp is not confusing the other animal classes for each other as much it is clear that using RMSProp for minimizing the issue of intra-classes.

This shows that just because a model has higher accuracy it is important to analyze the accuracy to minimize the type of errors you do not want. As RMSProp got less issue of intra-classes but has a smaller amount of accuracy.

REFERENCES

[1] A. Krizhevsky, “CIFAR10 dataset,” 8 april 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[2] Tensorflow, “GitHub,” Tensorflow, 06 January 2021. [Online]. Available: https://github.com/tensorflow/tensorflow/#readme. [Accessed 09 January 2021].

[3] Data Flair, “Data Flair,” [Online]. Available: https://data-flair.training/blogs/image-classification-deep-learning-project-python-keras/. [Accessed 09 january 2021].

[4] W. Abdulla, “Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow,” GitHub, 2017. [Online]. Available: https://github.com/matterport/Mask_RCNN. [Accessed 09 January 2021].

[5] R. Gilani, ”Towards data science,” 13 Juny 2020. [Online]. Available: https://towardsdatascience.com/main-challenges-in-image-classification-ba24dc78b558. [Använd 10 Febuary 2021].

[6] R. Benenson, ”What is the class of this image ?,” 22 Febuary 2016. [Online]. Available: https://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html. [Använd 09 January 2021].