

Introduction

The dataset used for this project is the CIFAR-10 dataset. It consists of 60,000 images, each being 32x32 pixels and has 3 RGB values that range from 0 to 255. This is further divided into 10 classes which are labels for the images such as dog, truck, ship, and cat.

This data was imported using the keras package and was split into a training dataset of 40,000 images and a testing dataset of 10,000 images. This split was chosen to avoid overfitting and only determine accuracy through test data which is not used in the training dataset. The data was then normalised by dividing the train and test datasets by 255 as this is the RGB values. There is an additional split of the training data into a validation dataset, also including 10,000 images.

Although often conflated with the test dataset, the purpose of the validation dataset is to be able to provide an unbiased evaluation of the fit of the model on the training dataset (Shah, 2020). This is used to help tune the hyperparameters (Pramoditha, 2022). In other words, the performance of the model is assessed using data that is not used in the training of the network. This will also help reduce overfitting the training data.

Methodology

For the methodology, an Artificial Neural Network will be created as a point of comparison to the later Convolutional Neural Network iterations. Then, several CNN models will be built with increasing convolutional layers as well as optimisation methods such as dropout and batch normalisation.

Initial models will only use 10 epochs. This was chosen empirically as, generally, these models begin to plateau after the tenth epoch and 10 epochs will give a good gauge on how the model will perform. Latter models will be using 100+ epochs where performance in terms of accuracy is likely to improve with more epochs (though these may be limited by early stopping).

This is a balanced multi-class dataset, meaning that each class has the same number of images. Thus, the main metric to evaluate the performance of the networks is accuracy.

ANN

The initial ANN model has 3 layers, the first with 30,000 neurons, the second with 1000, and finally the third with 10 neurons as there are 10 classes (Codebasics, 2020). This was over 10 epochs, and, as can be observed in the accuracy learning curve, whilst accuracy does begin to improve after each epoch, the accuracy for both training and validation datasets is low with the validation dataset only reaching 50% accuracy and the training dataset 55%. Performance would likely improve given more epochs, but because the accuracy is only 50% it is best to move on and try a different network. This remains a good start for a baseline benchmark; however, CNN will likely prove more accurate.

VGG

The basis for all CNN models in this project is the VGG architecture. This utilises 3x3 filters and stacks convolutional layers before a max pooling layer which formulates a block (Boesch, 2023 & Brownlee, 2020). By including pooling layers, computation will be faster as the size of feature maps is decreased thus reducing the number of training parameters (Hasan, 2023). Additionally, the number of filters is increased after each block. This allows the network to recognise more complex shapes and objects as the layers get deeper. Moreover, the size of the

filter reduces network tendencies to overfit. After these blocks are linear fully connected layers which are closest to the output. The final layers convey the information from the images and converts it to categories. In this dataset, it will be 'ship', 'truck' or any of the other classes. The softmax function in the final layer converts numerical values into probability values for each class (Johnson, 2022).

CNN

For the first CNN model, only 1 convolutional layer is used at first to gauge initial accuracy of this network and, as for all other models, will use the Rectified Linear Unit (ReLU) activation function; this effectively helps to achieve linearity by replacing any negative values with 0s. It is often touted as the best activation function for deep learning and is used in the VGG architecture to reduce training time (Sharma, 2022).

The optimizer used for this was Adam – Adaptive Moment Estimation. It has an adaptive learning rate and moment estimation capabilities (Gupta, 2023). Essentially, it helps adjust parameters within the neural network in real time to boost speed and accuracy. It is considered to have faster computation time, requires fewer tuning parameters, and is a powerful tool for improving accuracy and ensuring neural networks converge faster (Mahendra, 2023).

Finally, because this is a classification model, the loss function (which is used to measure the difference between the predicted output and the actual output) is sparse categorical cross entropy as the labels are integers. In this case, it will measure the difference between the predicted probability distribution and the actual probability distribution and how well they match (Rahman, 2023).

Other choices for initial parameters are the use of padding. This ensures the width and height of the output match the input (Brahmane, 2020).

We can see that after 10 epochs the validation accuracy is much higher than ANN with 64% - 14% higher. The training accuracy is also higher, and, if given more epochs, this accuracy would likely improve. Conversely, after 7 epochs, the validation accuracy does not improve enough to warrant further epochs. Thus, before considering adding more epochs, more convolutional layers are added to see if this improves accuracy.

Using the same hyperparameters as the first model, with three convolutional layers with 32, 64, 128 filters respectively, we can see an improvement in validation accuracy over 10 epochs in comparison to the first CNN model, this time rising by 7% to 71%. This suggests that with more convolutional layers the accuracy will be higher. However, the learning curves, particularly for the loss curve, show that there is overfitting in the model, and this should be addressed before adding more blocks or layers.

Improving upon CNN model - Dropout

In order to improve upon the model, there are several regularisation methods to improve overfitting. Firstly, dropout is added.

By adding dropout layers after each block, we can see if the model accuracy improves. This has been done by an increase in dropout rate (from 0.2 to 0.5) as the layers get further from the input. This means that 20%-50% of nodes are dropped after each layer with 50% being closest to the output.

The result is an 11% increase in accuracy to 82% up from 71%. We can also observe from the learning curves that the model is no longer overfitting to the extent that it was. This a good baseline to improve upon further. The model may benefit from further epochs, but this iteration of the model can be improved further by using batch normalisation.

Improving upon CNN model – Batch normalisation

Batch normalisation is an approach in deep learning to improve the efficiency and reliability of the neural network. It does this by reducing the internal covariate shift that occurs during training. In other words, it normalises the interlayer output of the CNN. This means that subsequent layers get a reset of the output from the preceding layer (*What is batch normalization*, 2022). This allows more effective analysis of the data by the network. Batch normalisation layers help to stabilise learning whilst also accelerating it. This ultimately reduces the number of epochs required for optimal training.

For this particular version of the model, as always there is a risk of overfitting. To reduce this, an early stop has been introduced so that when validation loss stops improving, after 5 epochs the training will stop (Mustafeez, 2023 & Vijay, 2020). This means that after 30 epochs, the training stopped. As a result of this, by introducing batch normalisation after each layer as well as continuing to increase the dropout, the accuracy improves again to 85%.

We can see in the learning curves that it may have been likely that training and validation may have increased further given more epochs, but overfitting may have been a bigger issue when it comes to loss.

Evaluating the Model

These network iterations and optimisations have greatly increased the initial validation accuracy of the ANN from 50% to a CNN with 85% validation accuracy. We can visualise the accuracy using the test images. From the plot, it can be observed that this latest version of the neural network correctly predicts all the images presented. An interesting note is that for the frog in the 2nd row, 3rd column, it correctly predicts at 50%, but followed closely is a prediction for dog. Looking at the image, we can see that this could be construed as a dog.

We can also look at other performance metrics and accuracy for each class. It can be observed that the recall and subsequently f1-score for class 3 (cats) is lower than any other class. This demonstrates that the model found it hard to detect positive samples for this class as recall is calculated as the ratio between the number of positive samples that are correctly classified as positive to the total number of positive samples (Gad, 2021). The same is also true for class 5, dog.

Conclusion

It is clear that CNNs are best suited for this type of dataset which is a challenging image classification task. This is largely due to CNNs being more efficient and scalable – but are also more computationally costly (Sharma, 2023).

By following VGG architecture which has proven to be a winning architecture in several competitions, the final model was quick to build and fairly simple to optimise. Each layer was carefully utilised to increase efficiency and effectiveness, such as adding dropout and including pooling layers.

The most interesting aspects of the creating of the neural networks is seeing how much tweaking can be done to improve the model and how there are many various techniques to iterate upon models.

There is no doubt that the accuracy of this model could be further improved as there are numerous examples of models that reach up to 99% accuracy (Franky, 2022). However, generally these models require more computational power, and the model for this project was limited to use only CPU power. Fortunately, with only a small number of epochs, the models performance is good for the resources that were available at the time it was created.

References

- Boesch, G. (2023) *VGG very deep convolutional networks (vggnet) - what you need to know*, *viso.ai*. Available at: <https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/> (Accessed: 12 October 2023).
- Brahmane, A. (2020) *Deep learning with CIFAR-10 image classification*, *Medium*. Available at: <https://towardsdatascience.com/deep-learning-with-cifar-10-image-classification-64ab92110d79> (Accessed: 12 October 2023).
- Brownlee, J. (2020) *How to develop a CNN from scratch for CIFAR-10 photo classification*, *MachineLearningMastery.com*. Available at: <https://machinelearningmastery.com/how-to-develop-a-cnn-from-scratch-for-cifar-10-photo-classification/> (Accessed: 12 October 2023).
- Codebasics (2020) *Image classification using CNN (CIFAR10 dataset) | deep learning tutorial 24 (Tensorflow & Python)*, *YouTube*. Available at: <https://www.youtube.com/watch?v=7HPwo4wnJeA> (Accessed: 12 October 2023).
- Franky (2022) *Once upon a time in CIFAR-10*, *Medium*. Available at: <https://franky07724-57962.medium.com/once-upon-a-time-in-cifar-10-c26bb056b4ce> (Accessed: 12 October 2023).
- Gad, A.F. (2021) *Accuracy, precision, and recall in deep learning*, *Paperspace Blog*. Available at: <https://blog.paperspace.com/deep-learning-metrics-precision-recall-accuracy/> (Accessed: 12 October 2023).
- Gupta, A. (2023) *A comprehensive guide on Optimizers in deep learning*, *Analytics Vidhya*. Available at: <https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/> (Accessed: 12 October 2023).
- Hasan, F. (2023) *Educative answers - trusted answers to developer questions*, *Educative*. Available at: <https://www.educative.io/answers/what-are-some-deep-details-about-pooling-layers-in-cnn> (Accessed: 12 October 2023).
- Johnson, M. (2022) *Why is the VGG network commonly used?*, *nnart*. Available at: <https://nnart.org/why-is-vgg-commonly-used/> (Accessed: 12 October 2023).

- Mahendra, S. (2023) *What is the adam optimizer and how is it used in machine learning, Artificial Intelligence +*. Available at: <https://www.aiplusinfo.com/blog/what-is-the-adam-optimizer-and-how-is-it-used-in-machine-learning/> (Accessed: 12 October 2023).
- Mustafeez, A.Z. (2023) *Educative answers - trusted answers to developer questions, Educative*. Available at: <https://www.educative.io/answers/what-is-early-stopping> (Accessed: 12 October 2023).
- Pramoditha, R. (2022) *Why do we need a validation set in addition to training and test sets?*, *Medium*. Available at: <https://towardsdatascience.com/why-do-we-need-a-validation-set-in-addition-to-training-and-test-sets-5cf4a65550e0> (Accessed: 12 October 2023).
- Rahman, M. (2023) *What you need to know about sparse categorical cross entropy*, *Medium*. Available at: <https://rmoklesur.medium.com/what-you-need-to-know-about-sparse-categorical-cross-entropy-9f07497e3a6f> (Accessed: 12 October 2023).
- Shah, T. (2020) *About train, validation and test sets in machine learning*, *Medium*. Available at: <https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7> (Accessed: 12 October 2023).
- Sharma, P. (2023) *CNN vs ann for image classification, Online Tutorials, Courses, and eBooks Library*. Available at: <https://www.tutorialspoint.com/cnn-vs-ann-for-image-classification> (Accessed: 12 October 2023).
- Sharma, S. (2022) *Activation functions in neural networks*, *Medium*. Available at: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6> (Accessed: 12 October 2023).
- Understanding VGG16: Concepts, architecture, and performance* (2023) *Datagen*. Available at: <https://datagen.tech/guides/computer-vision/vgg16/> (Accessed: 12 October 2023).
- Vijay, U. (2020) *Early stopping to avoid overfitting in neural network- Keras*, *Medium*. Available at: <https://medium.com/zero-equals-false/early-stopping-to-avoid-overfitting-in-neural-network-keras-b68c96ed05d9> (Accessed: 12 October 2023).
- What is batch normalization* (2022) *Deepchecks*. Available at: <https://deepchecks.com/glossary/batch-normalization/> (Accessed: 12 October 2023).

Feedback: **Knowledge and understanding of the topic/ issues under consideration (25%)**

There is a good demonstration of knowledge and understanding, displaying originality and a good understanding of all essential knowledge relevant to the work. The discussion of data is good, but more details could be included. You could include some examples and more details of the dataset planned to be used.

Very good analysis of the model evaluation and the use of CNN. The script includes references and that demonstrates a very good effort.

Presentation and communication skills (30%)

A very good presentation and demonstration of the communication skills. Good discussion of the application of knowledge and understanding to address the learning outcomes assessed by the assignment.

Criticality (25%)

In general, the team included a good set of discussions, yet I would prefer to see further analysis and elaboration using the literature. This could be in the form of critical analysis e.g. using literature or academic articles to support your arguments and recommendations.

The use of the methodology is good addition, as well as using ANN. The analysis of the ANN could be further extended. The discussion about improving the CNN model could include some practical examples and further analysis.

Structure and Presentation (as detailed in the assessment guidance) (10%)

A very good structure and presentation, and a very good use of literature to present and elaborate. Further conclusions could be more detailed.

Use of relevant sources (10%)

Good use of references and literature to support discussions.v