Liz Simpson, 951428

951428@swansea.ac.uk

m45J: Object recognition

Comparing the use of AlexNet and ResNet Convolutional Neural Networks to classify objects

Introduction

The aim of this paper is to discuss ways that machines can accurately classify images. Convolutional Neural Networks (CNN) are a machine learning technique used to create a model that can accurately predict the classification of unknown objects (Yang, Choi and Lin, 2016). However, there are many different architectures that can provide powerful frame works for classification. In this paper we will discuss which CNN architecture creates the most effective machine leaning model. We implement two CNN architectures – AlexNet and ResNet and compare the accuracy of the models using a subset of the CIFAR-100 dataset. The CIFAR-100 dataset is a large collection of objects with two sets of labels for the same dataset, “Fine” labels and “Coarse” labels. “Coarse” labels is less detailed and therefore splits the objects into fewer categories. Firstly, we implement an AlexNet CNN model, we train the model using fine and coarse labelling sets and compare its effectiveness to a more recent architecture, the ResNet CNN. Both models were scored on their learning time, accuracy of results using coarse labelling, accuracy of results using fine labelling and, on the accuracy of predicting objects from an unseen dataset. We will then evaluate the results by comparing our findings to the work completed by Sharma et al. to analyse whether our results are accurate and suggesting reasons why our models may differ.

Method

In this section we will discuss the dataset and the implementation, training, and testing of the AlexNet and ResNet Networks. We will discuss how features are extracted, normalised and how the models were trained and tested. All CNN’s will be written in jupyter notebooks using the tensorfloew and keras framework. I chose to use CNNs for this project as it is a supervised technique and can provide high accuracy results for image recognition.

The Dataset

A large dataset is needed to build a CNN. In this instance I used a subset of the CIFAR-100 dataset. The CIFAR-100 subset contains 32x32 images of 50000 objects, ranging from plants to tractors. As CNNs are a supervised learning technique, each image is labelled with a “fine” (more specific) and “coarse” (more general) label (Krizhevsky, Nair and Hinton, 2009). Fine labels identify 100 classes while the coarse dataset identifies 20 super classes ranging from fish to trees. We will analyse the effectiveness of AlexNet and ResNet models using both label variations.

Pre-processing of the data is needed to ensure all the data is within the appropriate format. Before using this data in a model, the data will be shuffled as to avoid unnecessary bias by the ordering of the images. For each model, the dataset is split into 2 sections. 80% of the dataset will be used as training data while the remaining 20% will be withheld and used to test the effectiveness of the models. All the input images must be 32x32 pixels and normalised so that there is no degeneration of gradients. The input shape for the models will be (32,32,3) holding information about height x width x RGB.

CNN Architecture

CNN’s are made up of multiple layers. AlexNet and ResNet both use layers of Conv2D, batch\_normalisation, activation, pooling 2d layers to take input data and their labels and outpt a classification model. However, ResNet employs skip connections with batch normalisation which significantly lowers the complexity of previous networks and can achieve human level accuracy (He *et al.*, 2016).

Brief note about Layers

Conv2d – Filters (kernels) are applied to the input. (Stewart, 2019)

Batch normalisation – A standardisation layer, used to speed up and stabilise the network by using the mean and variance of the batch to reduce dependency between layers. (Ioffe and Szegedy, 2015)

Activation Function Layers – In both models, a Rectified linear Unit (ReLu) function is used in the hidden layers. The last layer of the fully connected network will use a different activation function, the softmax function. The functions return elementwise calculations on the input tensor. (Keras, 2020)

Max Pooling layers – A Conv2D that takes the maximum value of filter only, reducing dimensionality of the network. (Stewart, 2019)

AlexNet

AlexNet was chosen because it is a relatively simple and early implementation of a CNN, made in 2012 (Khan *et al.*, 2020). There are 9 layers, made up of convolutional (partial) Conv2D, Batch Normalisation and Pooling layers. 5 layers are partial, meaning that each node in the layer is connected only to it’s close neighbours in the next layer, before 3 layers of fully connevcted nodes – all nodes in the layer connect to all the nodes in the next layer and finally an output layer that uses softmax to reduce the outputs into our 20 classes (Kumar, 2020). The difference between architecture comes from the differences in layers (Sharma, Jain and Mishra, 2018). The kernel sizes start off at 5x5 and are reduced to 3x3. Dropout was also used in the fully-connected layers to reduce overfitting.

The implementation of the AlexNet CNN was based on the original code by Kumar.

Features are extracted by …

Feature processing is …

how they are trained and tested.

ResNet

Resnet is short for Residual Network (He *et al.*, 2016) and is a much deeper model, meaning this model has more layers. Generally speaking deeper models are more accurate but are more complex and thus take longer to train. Deeper networks can also start to overfit the data so the deeper the network, the more likely your accuracy will plateo and state to degrade (Sharma, Jain and Mishra, 2018). Resnet was chosen for this project as it is claimed to be the next biggest development after AlexNet so would make a good comparison. Resnet poses using blocks and skipping layers to reduce the issue of vanishing gradients. (Feng, 2017)While there are many variants of ResNet we will focus on ResNet50 as it is the simplest version of the improved architecture so should be the easiest to implement and possibly feasible given the time constraints to train using the limited equipment accessible for this project. ResNet50 has 50 layers and to aid me in building a reliable ResNet I will be using the ResNetModel package which claimed 98% peak perfomace in 62 training epochs for the CIFAR-100 dataset created by Blocks Average pooling layers, lamba, concatenate. Also makes use of ReLu activation functions.

My First method will to try and use McDonnell’s ResNetModel package to create and train my own ResNet Model. If my models do not show the desired levels of accuracy I will try and implement an evaluation on the pre-trained models McDonnell has fine tuned (McDonnell, 2018).

However, it is observed that the main thrust in CNN performance improvement came from the restructuring of processing units and the designing of new blocks. (Khan *et al.*, 2020)

The initial hypothesis is that the ResNet will be the most accurate model.

Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **AlexNet** | **ResNet50 – Liz** | **McDonnell** |
| **Fine Labels** | **Accuracy** | **0.90** |  |  |
|  | **Accuracy\_val** | **0.33** |  |  |
|  | **Time to Train** | **4.6 hours** |  |  |
|  | **Score on unseen** |  |  |  |
|  | **epochs** |  |  |  |
|  | **Steps per epoch** | **100** |  |  |
| **Coarse Labels** | **Accuracy** | 0.9410 |  |  |
|  | **Accuracy\_val** | 0.4749 |  |  |
|  | **Time to Train** | **4.5 hours** |  |  |
|  | **Score on unseen** |  |  |  |
|  | **loss** | 0.1842 |  |  |
|  | **Val\_loss** | 2.7415 |  |  |
|  | **Steps per epoch** | **350** |  |  |

* Confusion matrix

A well-presented and thorough evaluation. The results provide a clear insight into the experimentation proposed within the methodology section. A clear understanding of the results is evident.

Present your experimental results in thissection. Explain the evaluation metric(s) you use and present the quantitative results (including the confusion matrix).

Evaluation

For the evaluation I will use this paper to discuss the differences in my report.

Batch size increases the accuracy of each epoch but increases the time taken to train the model.

AlexNet performed as expected.

ResNet models were tested on fine and coarse labels but proved more difficult to implement. Several attempts at resnet models resulted in no learning. in high accuracy and low val\_accuracy this would indicate overfitting.

The model may be too small/big that it underfits/overfits. (Number of Parameters)

As a result of this a different ResNet model

The model may need more time to converge. (Training for more epochs)

Perhaphs the ResNet may not be suited for this classification, however successful documentation of using resNet50 on CIFAR-100 datasets have proven possible.

Pretrained weights that you used may not be suited for this classification.

The learning rate may be too small/big.

Hindsight would indicate that the use of number\_of\_epochs, batch\_size, input\_generators, pre-processing, learning rates should all be included in all models and kept as consistent as possible to allow for a more meaningful evaluation of the models.

Conclusion

From this study we can conclude that ResNet performed better than AlexNet when properly modelled. Models are more accurate on broad classification. Finer analysis of images requires a deeper model and consequently longer training times. During this study, AlexNet CNN was implemented with an accuracy of 30%-50%. ResNet arcitectures were implemented, more care needs to be taken when constructing a more complex networks, consideration to be given to the number of epochs, steps per epoch, batch size and image preprocessing. ResNet arcitetcures are capcable of exceeding human recognition but only when properly tuned, which may not be a menial task for a beginner.

With the growth of popularity in VR and AR technologies, future work should include the evaluation of 3D volumetric images using a 3D AlexNet and 3D ResNet models to progress image classification into a virtual world concept.

A concise summary of the report. The critical analysis shows clear understanding of the materials and findings. Shows well considered suggestions for future work.Provide a summary for your method and the results. Provide your critical analysis; including shortcomings of the methods and how they may be improved.

References**.**

Das, S. (2017) *CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more… | by Siddharth Das | Analytics Vidhya | Medium*. Available at: https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5 (Accessed: 13 April 2021).

Feng, V. (2017) *An Overview of ResNet and its Variants | by Vincent Fung | Towards Data Science*, *Towards data science*. Available at: https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035 (Accessed: 17 April 2021).

He, K. *et al.* (2016) ‘Deep residual learning for image recognition’, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 770–778. doi: 10.1109/CVPR.2016.90.

Ioffe, S. and Szegedy, C. (2015) ‘Batch normalization: Accelerating deep network training by reducing internal covariate shift’, in *32nd International Conference on Machine Learning, ICML 2015*. International Machine Learning Society (IMLS), pp. 448–456.

Keras (2020) *Layer activation functions*. Available at: https://keras.io/api/layers/activations/ (Accessed: 17 April 2021).

Khan, A. *et al.* (2020) ‘A survey of the recent architectures of deep convolutional neural networks’, *Artificial Intelligence Review*, 53(8), pp. 5455–5516. doi: 10.1007/s10462-020-09825-6.

Krizhevsky, A., Nair, V. and Hinton, G. (2009) *CIFAR-10 and CIFAR-100 datasets*, *https://www.cs.toronto.edu/~kriz/cifar.html*. Available at: https://www.cs.toronto.edu/~kriz/cifar.html (Accessed: 13 April 2021).

Kumar, V. (2020) *Hands-on Guide To Implementing AlexNet With Keras For Multi-Class Image Classification*, *Analytics India Magazine*. Available at: https://analyticsindiamag.com/hands-on-guide-to-implementing-alexnet-with-keras-for-multi-class-image-classification/ (Accessed: 13 April 2021).

McDonnell, M. D. (2018) ‘TRAINING WIDE RESIDUAL NETWORKS FOR DEPLOYMENT USING A SINGLE BIT FOR EACH WEIGHT’, *arXiv*. Available at: https://github.com/McDonnell-Lab/1-bit-per-weight (Accessed: 17 April 2021).

Sharma, N., Jain, V. and Mishra, A. (2018) ‘An Analysis of Convolutional Neural Networks for Image Classification’, *Procedia Computer Science*, 132(Iccids), pp. 377–384. doi: 10.1016/j.procs.2018.05.198.

Stewart, M. (2019) *Simple Introduction to Convolutional Neural Networks | by Matthew Stewart, PhD Researcher | Towards Data Science*, *Towards Data Science*. Available at: https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac (Accessed: 17 April 2021).

Yang, F., Choi, W. and Lin, Y. (2016) ‘Exploit All the Layers: Fast and Accurate CNN Object Detector with Scale Dependent Pooling and Cascaded Rejection Classifiers’, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2129–2137. doi: 10.1109/CVPR.2016.234.