2021 Deakin AI Challenge Submission: Simpson's Image Classification Task

Peter J Allen*
pjal@deakin.edu.au
Postgraduate Student
Deakin University
Australia

ABSTRACT

The submitted deep learning model was developed via a long iterative process, trialling multiple combinations and permutations of models and associated hyperparameters until arriving at the model which was submitted. The final submitted model involved an ensemble of two Convnet models each of which was designed using a pretrained convolutional base. Following is a description of the process used to arrive at the final model.

KEYWORDS

deep learning

ACM Reference Format:
Peter J Allen. 2021. 2021 Deakin AI Challenge Submission: Simpson's Image Classification Task. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

2021 Deakin AI Challenge Submission: Simpson's Image Classification Task

1 MODEL DEVELOPMENT

The first model created involved constructing a multilayer Convnet with some densely connected layers on top to assist with classification.. However despite multiple adjustments to depth and size of layers, as well as regularisation strategies (including dropout) and other hyperparameter adjustments, this model was unable achieve accuracies much above 0.90. It's performance improved slightly once data augmentation was added but was still not performing significantly higher than 0.91 accuracy.

Subsequent models were designed using a pretrained convolutional base. Several of the pretrained models in Keras were trialled, with the Densnet201 model (pretrained on the Imagenet dataset) performing best. While the EfficientNet models may have outperformed this, insufficient time and computational resources were available to train an EfficientNet.

Early attempts at fine-tuning the final layers of the model with the Densenet convultational base produced significant improvements on earlier models, especially when data augmentation was added. However after multiple fine-tuning attempts it became clear that tuning the entire convolutional base yielded better results than simply fine-tuning the last few layers. This was presumably because features of the cartoon Simpsons images differed significantly in key characteristics to the primary features of the model trained on the ImageNet data. Hence not only the layers representing higher order representations needed tuning, but there was also fine-tuning required of the early layers that captured lower order representations. If this was the case, it was potentially because some of the basic feature detection representations required for the cartoon images were no doubt slightly different from the general Imagenet based features.

In order to terminate training at the optimum epoch for generalisation, a callback was established in Keras to save the model at the highest accuracy on the validation set during training. The downside of this approach is that it risked overfitting on the validation set. Hence it was decided to ensemble two convnets to reduce the extent of overfitting on the training and validation set, and maximise potential generalisation.to the test set.

The two best models were selected based on their accuracy scores on the validation set. The two models selected performed with accuracies of 0.9938 and 0.9928 respectively. The two models had the identical structure, with a Densenet201 convolutional base and a dense(1024) layer followed by a dense(20) layer for classification, and softmax activation layer overlaid on the dense(10) layer . The only differences between the two models were: bach normalisation on the Dense layer in one, and both models were trained with different data augmentation, with one model trained on much more subtle data transformations than the other. It was deemed that these differences were sufficient to provide some significant disparity in model weighting for the two models which would aid generalisation of the ensembled model.

The data augmentation provided best scores on validation set when the augmentation strategies were set to somewhat higher than is often used on non-cartoon images. (eg. the rotation angles were greater and the horizontal shifts were wider). This may have to do with the greater magnitudes of variation implicit to features in cartoon images.

In order to ensemble the models, the final softmax activation layer was removed from the models using the code in the attached github link. An averaging layer was then created, averaging the outputs of their final dense(20) layers and this averaged layer was then passed through a new softmax activation layer to provide the

*Postgraduate Student

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

output of the combined models. The combined ensemble model performed with accuracy of 0.9943 on the validation dataset.

2 ALTERNATIVE APPROACHES

Several other approaches which may have yielded superior accuracy score were considered but not pursued due to time constraints and/or limitations in computational resources. One of these was the possibility or first filtering images through a face detection function such as is available in OpenCV. The rest of the image could then be converted to white or black pixels and a model could have been run on face detection alone. This face recognition model could then have been ensembled with the standard CNN to provide a more robust model. The effectiveness of this strategy would depend on the accuracy of the OpenCV function in detecting the faces on the animated Simpsons characters.

As noted above, the EffcientNet pretrained model was also considered, but it was determined that this was too computationally expensive given available resources.

In addition, a model was developed using the "VGG face" pretrained model. However it was not clear if the underlying software.versions would be compatible with the software in the Codalab competition environment, and hence this approach was abandoned early on.

Extra data collection was also considered but insufficient time was available to pursue this option.

3 GITHUB LINK

The code used to generate the above described model can be found at the below github link:

https://github.com/pjasyd/Deakin_AI_Challenge_Code