CISC-865-Deep Learning

Image classification using CNN with Keras and VGG16

Rochelle Allan

Rochester Institute of Technology

December 11, 2021

1 Introduction

Convolutional networks is one of the classes available in deep learning and plays a huge role in image recognition and classification. This is used at the core of different tagging algorithms like the one present in applications like Instagram,Facebook and the like. This is also found in tagging or identification efforts in self-driving automobiles. Image recognition and classification can also be found in the security and the health industry. They are the popular choice because of how efficient they are along with the speed at which they work. To describe the image classification process, it starts with an input which is most often a picture (a dog) and receives an output that represents the class to which it belongs ("dog" in this case) which is nothing but the highest class probability to which that input belongs judged by the model. The project begins with an image classification problem implementing CNN with a Keras Sequential network to try to address the challenge. The performance of this model is evaluated but the test accuracy achieved was around 76 percent. To improve this accuracy, the focus shifted to using pre-trained VGG16 to extract features and address this challenge. VGG-16 is a type of CNN which is fine tuned and achieves a much better accuracy. The images used for training and testing images used are dogs, pandas and cats. The training size is 9000 images in total and the testing size is 2696 in total. The final test accuracy achieved is 91.4 percent. The proportion of the training data used is shown in fig.1

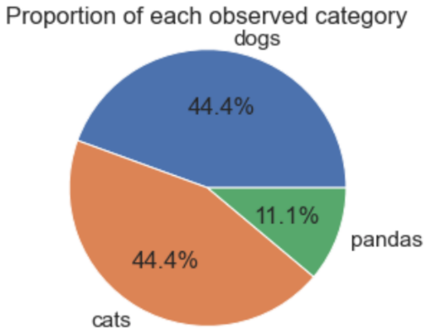

Figure 1: Proportions of categories in the training data

2 Related Work

Image classification finds its applications in a vast number of fields including the medical industry. [1] worked on Image classification with brain images using convolutional neural networks. If the training data is small, it is known that this neural network, CNN, might overfit the data. To combat this, deep convolutional networks (DCNN) with transfer learning came to be. And this paper investigates the potential of these networks with the VGG-16 model with transfer learning. The authors talk about replacing the last couple of layers of the VGG-16 model to account for new image categorization for current applications. This was evaluated with the Harvard Medical School data consisting of images ranging from normal to a vast set of images with neurological problems.

This paper [2] improves on the VGG-16 model with smaller size of the model (88.4 percent smaller in size), works much faster in terms of time taken to train (23.86 percent), uses residual learning giving a more acceptable generalization, matches recognition learning well with the MIT 265-standard scene dataset. They compare this to SqueezeNet. The potential of the work can be found in self-driving cars, video-surveillance and GPU tasks, etc.

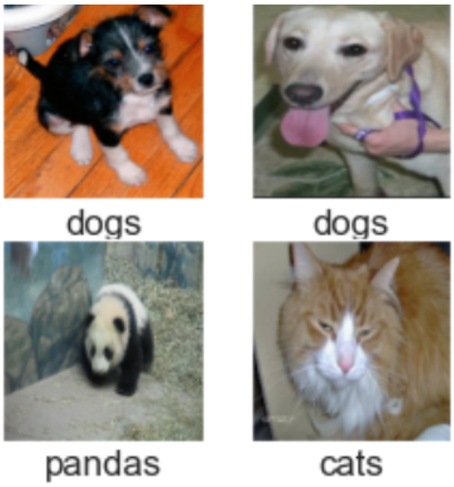

Figure 2: Images labeled with the right class.

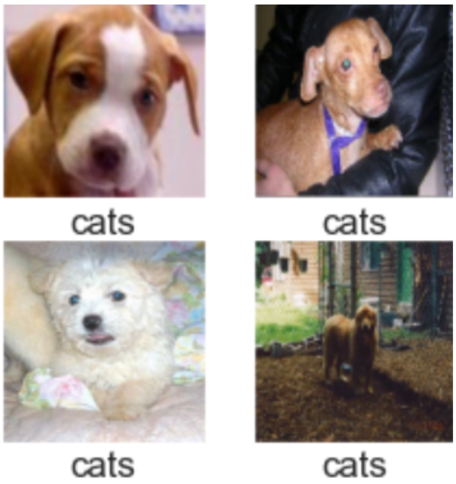

Figure 3: Mislabeled images.

Here, [3] they use a convolutional neural network through a series of layers that are fully connected - convolutional, pooling, flattening for image classification. The model is built fro, the ground up and fine tuned using image augmentation methods. Pre-trained gg16 model is used to classify these images and evaluate the accuracy with the test data. They achieved a 72 percent accuracy with CNN and a 95 percent accuracy with VGG16 along with image augmentation methods.

The ImageNet dataset was used for classification in [4] by generalizing a traditional approach to rectified linear units (ReLU) called Parametric ReLU which improves fitting of the model with barely any overfitting and cost in terms of computation. As a result, they obtained a 4.94 percent test error (top-5) on the ImageNet dataset.

3 Methodology

A birds-eye-view to the approach can be described from loading the data and preprocessing it. Next, a CNN model with Keras Sequential is trained with the data and the performance is evaluated. Next, the use of a pre-trained model called VGG-16 which is another type of CNN model is used to address the problem and evaluated. So, in general the model creation steps are to build and compile the model, train and fit the data to the model, evaluate the model performance on the test set and carry out error analysis. Coming to the first model to be talked about, CNN with Keras - the model is built with different layers, the first being Conv2D where the features will be derived from the image. The next layer is the MaxPooling2D where the images are resized to half the size. The next layer is the flatten layer which transforms the image from a 2-d array to a 1-d array. The Rectified Linear Unit Layer is the next layer which returns the max of two values and finally the softmax layer having a total of 6 neurons carrying with it the probability that the image indeed belongs to one of the classes. Coming to the compilation of this model, the parameters used starts off with an optimizer called ADAM which includes the exponent of weighted averages of squares of previous gradiets called RMSProp and keeps track of previous gradients for better learning, which is Momentum in other words. The loss function used is sparse categorical crossentropy for classification.The training accuracy went up to 96 percent at the tenth epoch. However the test accuracy achieved with this model was 76 percent. An example of correctly classified images can be seen in fig 2 and misclassified images in fig 3. To improve on the accuracy, pre-trained model, VGG16 was used for training the data and was further fine-tuned. The features can be extracted with VGG16 and was visualized using principal component analysis shown in fig 4 which helps reduce the dimensions of the data in question. We are able to identify clusters which correspond to the class a given image may belong to. The results obtained with this model is elaborated in the evaluation section.

4 Research Questions

i) What is the effect of Binary Cross Entropy versus Sparse Categorical Cross Entropy Versus Categorical Cross Entropy for two classes- Dogs and Cats?

When taking only two classes (out of the three) into consideration, that is the dog class and the cat class, we evaluate the accuracies achieved in each case, i.e; from the test data, how many labels were predicted accurately. The CNN with Keras is compiled first with binary cross entropy as its loss function and then with sparse categorical cross entropy as its loss function. In this case, we see that the test accuracy is around 19 percent when binary cross entropy is used as its loss function but when sparse categorical cross entropy is used as the loss function, a test accuracy of 69 percent which is a 50 percent jump from what was achieved using binary cross entropy. When these loss functions are experimented with in the pretrained, fine tuned VGG16 model, we see a better performance using sparse categorical cross entropy instead of binary cross entropy here as well. While considering categorical cross entropy for CNN with Keras, a test accuracy of 50 percent was achieved. This is a lot higher than the binary cross entropy loss function (31 percent higher) but still much lower than the sparse categorical crossentropy (19 percent lower). The VGG16 model with loss function of categorical crossentropy also produced a test accuracy of 50 percent which was significantly lesser than the VGG16 model with sparse categorical cross entropy which achieved 91 percent test-accuracy. Typically one would think that binary crossentropy would work better than sparse categorical cross entropy with just two classes in question or that categorical cross entropy would work better than sparse categorical crossentropy because of its soft probability consideration but this was not the case given the nature of this dataset.

ii) How do the number of epochs affect the training accuracy of the models?

The models were run with 3,5 and compared with the standard 10 (1 epochs which is basically one traversal through the entire dataset) used in the main notebook. Considering the CNN with Keras model first, with three epochs on the training data, the first epoch is at 58 percent val-accuracy going up to 72 percent for the third epoch. The overall accuracy achieved is 73 percent. The pre-trained VGG16 fine-tuned model, the first epoch starts at 87 percent going up to 89 percent. The overall accuracy achieved is 87 percent. Investigating five epoch performance starting with CNN with keras model, the first epoch starts off at 60 percent val-accuracy and gets stable at the third epoch all the way up to the fifth epoch at 70 percent. The overall accuracy achieved was 73 percent. With the VGG16 model, the first epoch starts at 79 percent val-accuracy and gets stable at the second epoch all the way up to the fifth epoch at 92 percent. The overall accuracy achieved is 92 percent. Comparing the above two to the tenepoch learning rate, the CNN with Keras epoch run starts off at 59 percent val-accuracy, gets stable around the fifth epoch and reaches a val-accuracy of 73 percent by the tenth epoch. The overall accuracy achieved is 77 percent. For the VGG16 model, the epoch starts at 92 percent and goes upto 93 percent val-accuracy at the tenth epoch. This confirms the belief that when number of epochs (run on the training data) are increased, the learning rate on the training data can also increase thereby improving the accuracy to label a given image present in the test data.

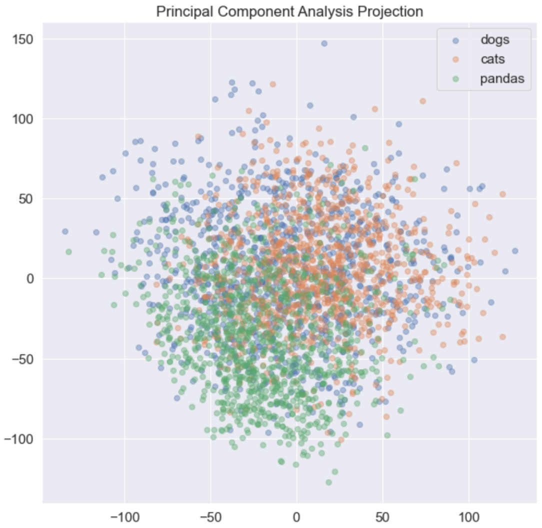

Figure 4: PCA analysis Projection

5 Evaluation

In the case of image classification, the best way to validate the models are to check the test-accuracy with which a model correctly assigns the right label for a given input image from the test data (2696 images)after the model has been trained on the train data- 9000 images. Starting with the evaluation of the CNN model with Keras with 10 epochs run on the training data, one can see that the accuracy (training) starts off with 44 percent (first epoch), gets stable at around the seventh epoch with 90 percent and it steadily increases to 96 percent at the final epoch. However the test accuracy achieved was 77 percent. This needed to be improved and hence the focus was shifted to the pre-trained, fine tuned VGG16 model using 10 epochs on the training data staring off with accuracy (training) of 46 percent (first epoch) to 99 percent (the tenth epoch). The test accuracy achieved with this model however was much higher than the previous- of 91 percent.

6 Conclusion

In conclusion, this deep learning project aimed at image classification using images of dogs,cats and pandas taken from kaggle (multiple locations). We covered the introduction to the problem, related work survey, the methodology, two research questions and the evaluation. I was able to achieve a test accuracy of 91 percent with the VGG16 model which was fine-tuned. This test accuracy achieved was a huge step up from the CNN with Keras Sequential model that was tried initially to tackle this challenge, which achieved a test accuracy of 76 percent. Therefore finally, we were able to boost the test accuracy from 76 percent to 91 percent (15 percent jump). Both models had a very high training accuracy where 10 epochs were used for each.

References

[1] Taranjit Kaur and Tapan Kumar Gandhi. Automated brain image classification based on vgg-16 and transfer learning. In *2019 International Conference on Information Technology (ICIT)*, pages 94–98, 2019.

[2] Hussam Qassim, Abhishek Verma, and David Feinzimer. Compressed residual-vgg16 cnn model for big data places image recognition. In *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 169–175, 2018.

[3] Srikanth Tammina. Transfer learning using vgg-16 with deep convolutional neural network for classifying images. *International Journal of Scientific and Research Publications (IJSRP)*, 9:p9420, 10 2019.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.