Big Data and Machine Learning

Coursework: Object Recognition

2122_CSCM45J

Pallav Shukla

2154638

21-04-2022

Abstract

The following report is about object recognition on a big data CIFAR-100 dataset, it consists of large amount of small images which has number of classes in it and that is further subdivided into super classes [1]. Different machine learning algorithms how they have been applied has been discussed in the report [2].

- Keywords: object recognition; CIFAR-100; CIFAR-10; Extremist sentiments; machine learning; big data.

Introduction

Canadian institute for Advanced Research CIFAR-100 is a data set consisting of small images in numerous amount has been provided to apply machine learning algorithms on the basis of object recognition [3] , [1]. CIFAR -100 contains 100 different categories of objects inside which there are 500 images for training 100 images for testing. The objects are further grouped into 20 super-classes. The height with end colour of each image has a fixed size off 32 x 32 x 3 [2]. the CIFAR-100 dataset is it stored in 4D arrays. In order to apply object recognition we researched many algorithms. In order to apply object recognition we researched multiple algorithms. Some of them are Convolution neural network CNN, support vector machine SVM, linear regression and neural network NN. CNN prove to be a very effective approach for object recognition on image but due to its complexity as well as the multiple options of choosing the different parameters to make it efficient it is not being used quite often. We used to methods for implementing CNN. In the first method are accuracy was 50% on the super class where as on the subclass we reached 36% and 47% accuracy.

Method 1

Going through the given problem we started applying convolution neural network CNN. It is a special type of artificial neural network which involves mathematical operations like convolution instead of normal matrix multiplication in more than one layers. The CNN model was designed for going through pixel data to be used in image-recognition and image-processing [4]. We begin our code by importing necessary Python libraries and importing dataset files that were provided in the problem. We used a specific Google drive method to import the files where-in we uploaded the files on drive. When the files were needed we downloaded them using the `gdrive` command and the object links.

We loaded the data and then transposed the training set using the transpose function of numpy library. Then we selected an index to see a sample of the image using imshow. Then we created a dictionary of the super category or the super-class. To fit data into the kera's model our data using the to_categorical function. Then be used to train test split to split the data into the training set and validation set. Next we went ahead to prepare the data so as to normalize it as the pixels are Max of 255 units.Next we build our latest model by importing libraries In which we are doing convolutional 2D layers and Max pooling layer as well this was repeated 2 times changing the required parameters. We used the activation function as Relu and initialized the kernel with he_uniform. We then added for flattening layer and 2 dense layers whose output was set to the number of super classes that is 20.

We use the Stochastic gradient descent optimizer and then compiled the model. After that we fit the training data set with the labels as well as the validation data set.
We developed a reusable plot history function which can plot the training accuracy versus the validation accuracy as well as it also plots training loss and validation loss. The accuracy achieved in this case was 50.24 percent. We use the sklearn confusion Matrix library to plot observation versus prediction. The diagonal line represented that most of our classes got predicted accurately.
We have also elaborated in our code about the classes which were chosen frequently while prediction as well as the classes which were not chosen during prediction. In this model we are super classes classification was being done there were no such classes which were not predicted. We can went ahead and predicted on a sample set of images. And our model predicted the superclasses quite precisely as can be seen in the Prediction image. Although in the samples chosen there are some discrepancies such as aquatic mammals(whale in this case) is predicted as a vehicle 2 , which is supposed to be because of the grey shade of the whale. The model predicts very well other categories such as trees, large carnivores, large man-made outdoor things, fruit and vegetables (the crab/spider in the first image is quite disturbing).

Figure 1: Method 1 on coarse labels(super class) - Accuracy 50.24%

Training the model on finer sub classes

Each of the hundred sub classes contain 500 training images and hundred testing images. We follow the similar approach for the final sub classes and trained our model but this time we change the output of the dense layer to 100 to compile and fitting the data for classifying in the sub classes. We again plotted the history and the curves as above. Also similarly made the prediction on a sample set which was used previously.

As expected, the accuracy of this model dropped significantly in comparison to the super-class trained model. We managed to achieve an accuracy of 35.74 %. The model performed quite badly in predicting the images and on the previous same samples, we didn't get correct predictions as to the ideal human eye. This model predicted a bicycle instead of lion and beaver instead of pears. Definitely we needed to improve this model as there were a lot of subclasses which were not chosen in prediction. Therefore we used regularization in Method 2 to improve the model performance over sub-category labels.

Figure 2: Method 1 on fine labels(sub class) - Accuracy 35.74%

Method 2

For method 2, we followed almost similar approach like above but in this case we utilise the regularisation technique of using the dropout layer. Dropout helps us to ensure that the model does not get biased towards a

particular feature, that is, it ensures that the model performs well even in the absence of that particular feature. We added the dropout by adding new Dropout layers, where the number of nodes removed is specified by the parameter provided while calling the dropout constructor. in our case we use the fixed dropout rate of 20%. Also in this method we added an additional dense layer with dimensionality of the output space starting from 256 instead of 128. Voila ! When the model was trained we achieved an accuracy of 47.36%. The model categorized each of the samples to at least one category. The model predicted very well in comparison to the previous model on finer sub classes. On the same sample set as we have used for prediction of our models since method 1, this model predicted more accurately for example rocket, skyscrapers, castle, pine and unbelievably it predicted the crab as well (calculated lucky guess !). Also as per observation the model significantly performed better as we could see that it predicted a sweet pepper (orangish-yellow) as an orange.

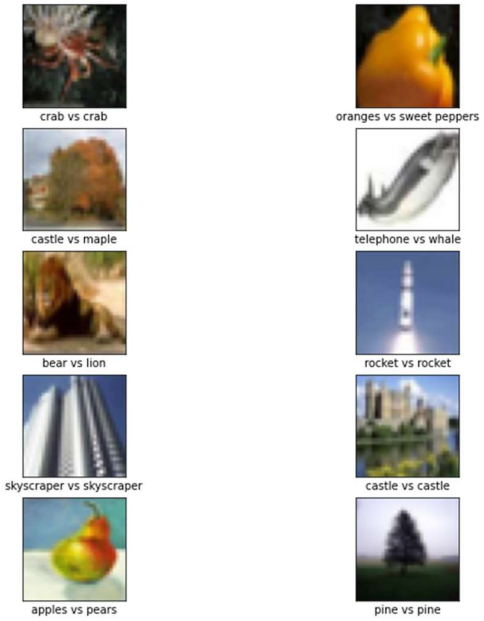

Figure 3: Method 2 on fine labels(sub class) - Accuracy 47.36%

Conclusion

The cnn model developed with the dropout came out to be the best approach for the given problem.

References

1. *Papers with Code - CIFAR-100 Dataset*. Paperswithcode.com. (2022). Retrieved 20 April 2022, from https://paperswithcode.com/dataset/cifar-100.

2. *Sign in to your account*. Canvas.swansea.ac.uk. (2022). Retrieved 20 April 2022, from https://canvas.swansea.ac.uk/courses/34594/assignments/196952.

3. *CIFAR-100: Pre-processing for image recognition task*. Medium. (2022). Retrieved 20 April 2022, from https://towardsdatascience.com/cifar-100-pre-processing-for-image-recognition-task-68015b43d658.

4. *Convolutional neural network - Wikipedia*. En.wikipedia.org. (2022). Retrieved 20 April 2022, from https://en.wikipedia.org/wiki/Convolutional_neural_network#Definition.