Unsupervised Machine Learning and Data Mining (DS 5230) Spring 2022

Project Report

Project Title : Deep Clustering for Unsupervised Learning of Visual Features A Reproduction

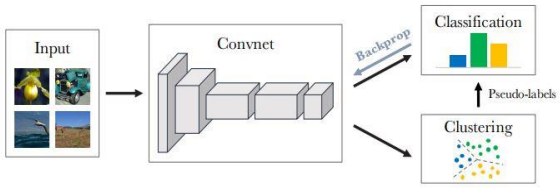
Isha Hemant Arora

1. Abstract

Clustering is a type of unsupervised learning algorithm which makes the use of unlabeled data and forms clusters. Clusters are nothing but a group of similar items that are grouped together based on some clustering algorithms, which are different from the other groups. The aim to segregate the similar datapoints based which poses similar traits. It is quite famous and extensively used in many applications such as recommendation systems, network analysis, Image segmentation etc. It is widely used in computer vision where the role of image segmentation is to partition the image into several segments. In our project, we are using a deep clustering algorithm (also called DeepCluster), where the method jointly learns the parameters of a neural network and the cluster assignments of the resulting features. It iteratively groups together the features with the help of common clustering algorithm like k-means clustering (we have used Power Iteration Clustering) and then uses the subsequent assignments as a supervision to update the weights of the network. It involves applying deep cluster to the unsupervised training of the deep learning model, convolutional neural networks on the ImageNet dataset and the Animals dataset.

2. Introduction

The unsupervised machine learning techniques have been widely studied and used in the machine learning domain. These provide a variety of algorithms for clustering, dimensionality reduction, computer vision applications etc. The main reason for the wide growth and their success is that they can be applied on any different domain or variety of datasets. Despite this, only a very few are used or proposed to adapt the end-to-end training of convnets. They are primarily used for the linear models, on a fixed number of features and do not work well when the features are to be learned simultaneously. In this project, we have implemented a novel clustering algorithm for the large-scale end-to-end training of the convnets, where the general-purpose visual features can be obtained with a clustering framework using the standard clustering algorithm, like k-means. It should be noted that this clustering algorithm can be any other available clustering algorithm (like spectral or power iteration clustering)



3. Problem definition

In our project we have tried to reproduced the paper [1] “Deep Clustering for Unsupervised Learning of Visual Features” by Caron et al. of Facebook AI Research. The model proposed outlines applying Deep Clustering (popularly known as DeepCluster), a clustering method which jointly learns the neural network parameters and clusters of the assignments of the resulting features. This approach is proposed to account for end-to-end training of visual features on large scale datasets. The idea is to accurately be able to label huge datasets without having to go through the tiresome process of manual annotations.

4. Dataset description

In this project, we have used 2 datasets to help us with the project. The first is the ImageNet; it is a very large dataset containing annotated photographs used for computer vision research. The dataset nearly consists of 21K classes; 1M images that have bounding box annotations. To counter space and hardware constraints, we have used the Mini-ImageNet dataset which consists of randomly selected 64 classes each containing 600 images. The second dataset used is the Animals dataset. We downloaded this dataset from a Kaggle competition. It consists of about 28K medium quality images. The images belong to 10 categories present: dog, cat, horse, spyder, butterfly, chicken, sheep, cow, squirrel, and elephant.

ImageNet Dataset

Animals Dataset :

5. Existing Methodologies

The existing method is that of the Convolution Neural Network, a class of deep learning neural network. These CNNs are most commonly used to analyze the visual imagery and represent a huge breakthrough in the image recognition/classification. They take an input as a picture and output a class (example: “Dog”). Inherently supervised, it consists of convolution layers, ReLU layers, pooling layers and a fully connected layer. Its major limitations are that it takes a lot of time and is slow for a large dataset. Therefore, by implementing deep clustering algorithms we can process large high dimensional datasets with a reasonable time complexity.

The another widely used algorithm is the K-means clustering, where the algorithm aims to partition ‘n’ observations into ‘k’ clusters. Each observation ‘Ni’ belongs to the cluster with the nearest centroid. The squared Euclidean distance between the observation and the centroid of the cluster is minimized in the iterations. There are multiple other clustering algorithms like the Spectral Clustering and Power Iteration Clustering that have their roots in graph theory.

6. Proposed Method

The model we have used in our project is the Deep Clustering model [2], as discussed earlier. For the training data and convnets architecture we have trained

the deep cluster on the dataset, discarded the labels, and for comparison used AlexNet architecture.

Starting with the dataset, for the training we took unlabeled images from the dataset and took the mean and standard deviation to perform the transformations. In the preprocessing, transformations are applied to the images so that the features that are learned are invariant to augmentations.

We performed two augmentations: transformation when doing clustering and when training. We resized the image by performing center crop on the image and then normalizing it using the mean and standard deviation.

Then while training the image was horizontally flipped with a 50% chance and then again normalizing it. After getting the normalized images we converted them into grayscale. For the edge detection we performed Sobel filtering, i.e., increasing the local contrast of the image. Edge detection is performed to reduce the amount of data and filter out the unneeded information.

Before the basic clustering was applied, we performed dimensionality reduction to the image-feature matrix. For the dimensionality reduction, we used the principal component analysis (PCA) algorithm to reduce the number of features. We made the use of faiss library to perform this at a scale. The values were also whitened.

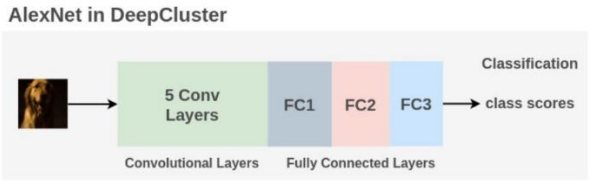
After the PCA, L2 normalization was applied to the values. Following the preprocessing, we performed random clustering on the pre-processed features to get the images and their corresponding clusters. These clusters acted as pseudo labels on which the model was trained.

After having all the images and clusters, we applied our deep learning model, i.e. trained our ConvNet model. The model was trained for 100 epochs. The clustering used in our project was PIC. Power Iteration Clustering (PIC) is a scalable graph clustering technique. PIC finds a low-dimensional embedding of the dataset by using truncated power iteration on a normalized pair-wise similarity matrix of the data. This embedding turns out to be an effective cluster indicator, consistently outperforming widely used spectral methods such as NCut on real datasets. PIC is very fast on large datasets and runs over 1,000 times faster than an NCut implementation.

6.1 AlexNet

AlexNet is a leading convolutional neural network (CNN) architecture primarily used in object detection tasks. It is the first CNN which used GPU to boost the performance by employing multi-GPU training thus also helping in cutting down the training time.

The AlexNet architecture [3] consists of eight layers: five convolutional layers and three fully-connected layers.



There are few more features that are used, which are some of the new approaches to the convolution neural networks:

6.1.1 ReLU Nonlinearity: AlexNet uses Rectified Linear Units, the ReLU units in the place of the tanh activation function, which are standard functions. The main advantage of ReLU while was the training time required. A CNN model when using ReLU was able to reach a 25 percent of error on a dataset named CIFAR-10. It was six times faster than a CNN model using tanh.

6.1.2 Multiple GPUs: The AlexNet allows for the multi-GPU training by putting allowing half of the models' neurons on 1 GPU and the other half of the neurons on another GPU. This makes the training period shorter and can be used to train bigger models.

6.1.3 Overlapping Pooling: The CNN model usually pools the output of the neighboring groups of neurons with zero overlapping. In the AlexNet model the authors introduced the overlapping and were able to reduce the error by about 0.5%. Also, the overlapping models do not overfit easily.

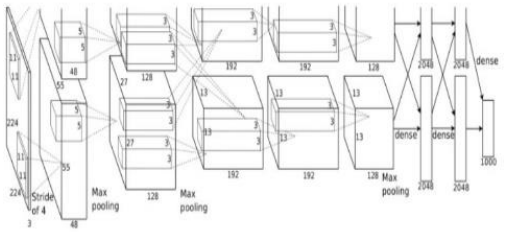


Illustration of AlexNet's architecture. Image credits to Krizhevsky et al., the original authors of the AlexNet paper.

6.1.4 The overfitting problem: There was a problem of overfitting where 2 methods were employed to reduce the overfitting. First was the data augmentation, where the label preserving transformation was used to make the data more varied. Also, Principal component analysis (PCA) was performed on the RGB pixel values to change the

intensities of the RGB channels. This helped in reducing the error rate by more than 1%. The other method was Dropout; In this method the neurons are turned off with a predetermined probability of fifty percent. In each iteration there is a different sample of model's parameters, which forced each of the neuron to have more robust features. However, the dropout mechanism increases the training time needed for the models convergence.

6.2 PCA (Principal Component Analysis)

Principal Component Analysis [4] is a dimensionality reduction technique which is used in unsupervised machine learning algorithms for reducing the huge number of dimensions; it is basically reducing the dimension of the variables of the large dataset and compressing it to a smaller one, without losing any information.

Principal Component Analysis for Image Compression:

The images are a combination of several pixels which are placed in a row one after the another to form a whole, single image. Each pixel value represents the intensity value of the image. With multiple images being present a matrix can be formed considering a row of pixels as a vector. Since it requires a huge storage amount, we perform PCA to compress it and try preserving the data as much as possible. Thus, the main objective of principle component analysis is to reduce the storage without any loss of the useful information.

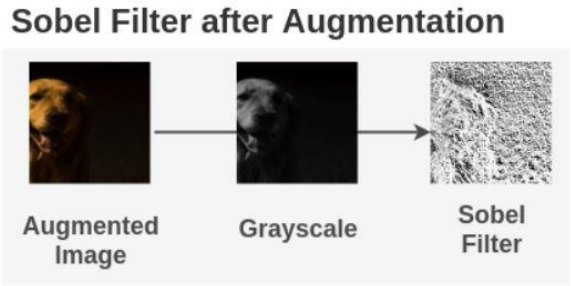
6.3 L2 Regularization:

Regularization is basically performed to avoid the chance of overfitting of our model. In the neural networks the regularization is a technique used to reduce the likelihood of the model overfitting.

There are two types of regularization techniques: L1 and L2 regularization. We have employed the L2 regularization as a part of our project. It works by adding a term to the error function used by the training algorithm.

6.4 Sobel Filtering

Sobel filtering is used for the edge detection. Edge detection is needed to be performed for the images because the images contain noise, which generate sudden transitions of pixel values. There are three steps in the edge detection: Noise Reduction, Edge Enhancement, Edge Localization. Sobel filtering works by calculating the gradient of the image intensity at each pixel within the image. It then finds the direction of the largest increase from the light to dark and the rate of the change is measured in that direction.



7. Results and Discussion

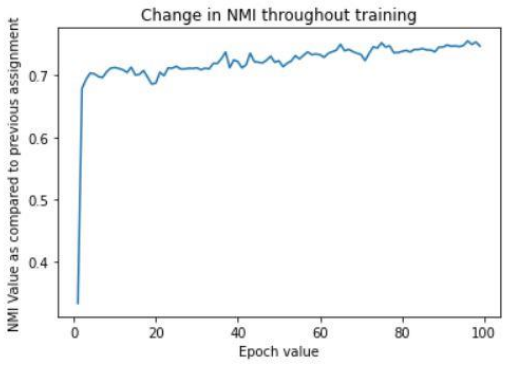
For the evaluation, we used the NMI score.

NMI stands for Normalized Mutual Information, and it is related to the information theory. Normalized Mutual Information (NMI) gives the total reduction in the entropy of the class labels when the cluster labels are given. It is very similar to the information gain in the decision trees. NMI measures the reduction in the uncertainty. Thus, we can say that it is a measure of the quality of our clustering. It is an external measure because the class labels of the instances are necessarily required for the determination of the NMI. Higher the NMI value, the better the clusters are. Since the NMI is normalized it can be used to compare between different clusters having different number of clusters. In our project we have used it to compare the new cluster and the previous cluster. The model was trained for 100 epochs.

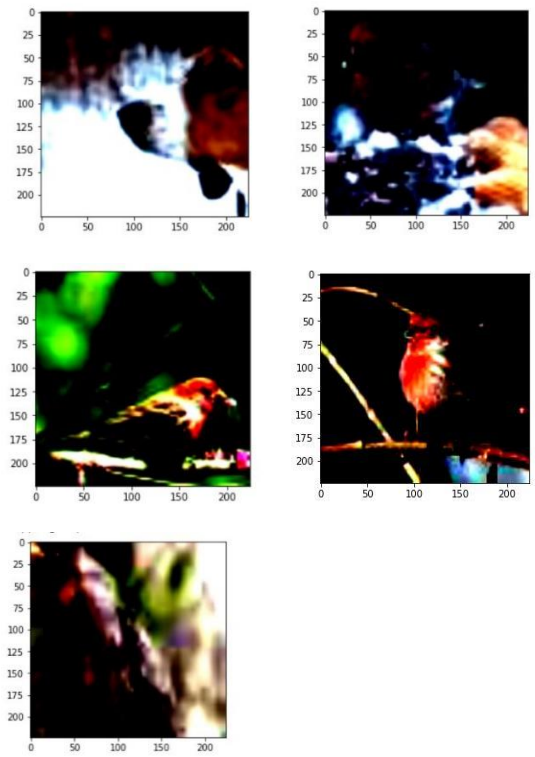
NMI graphs were generated for our each of our 2 datasets, which are shown below:

7.1 NMI graph for the ImageNet Dataset:

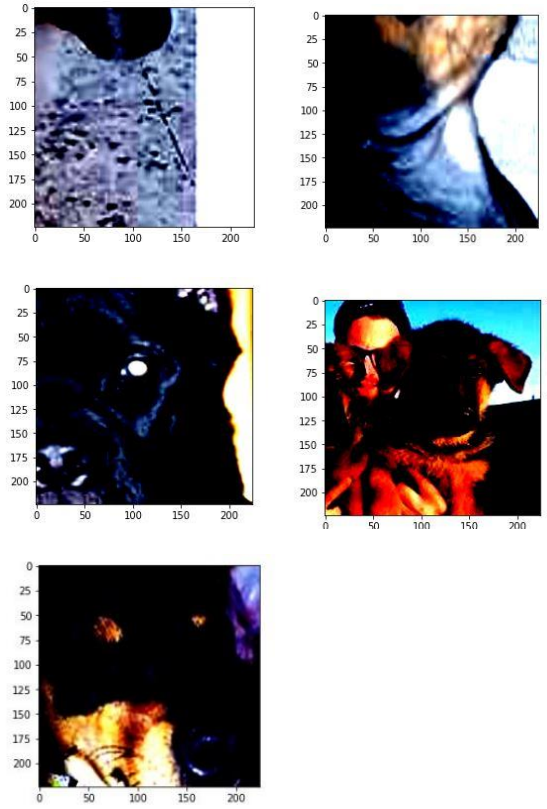
As inferred from the graph, we can tell that the initially the value was zero, but it suddenly gave a rise and jumps to the height of 0.7. The NMI value did not increase beyond 0.75. The number of oscillations were less as compared to the Animals dataset.



Sample Cluster Output (for one cluster) of the ImageNet Dataset:

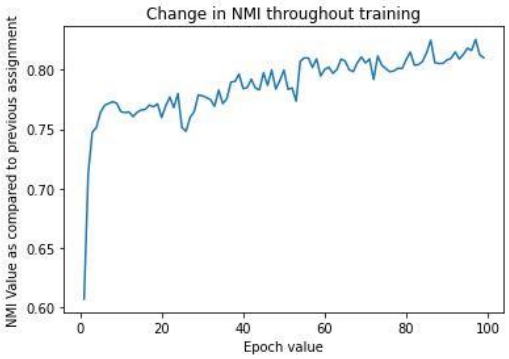


Sample Cluster Output (for one cluster) of the Animals Dataset:



7.2 NMI graph for the Animals Dataset:

The original data for the animals dataset consists of 10 classes (each containing around 4000 images). Thus, any change in the clusters was more obvious (the data is separated on 10 clusters, all with huge amounts of images) in this kind if dataset than one with more classes. Similar to the ImageNet dataset, the NMI between consecutive clusters first jumps to a value of 0.7 and then continues to increase with a smaller gradient (and lots of oscillations) to just above a value of 0.8.



8. Takeaway points are Future work

As a part of the future work, we will be implementing mini-batch Spectral Clustering. It is a graph-based clustering techniques where the eigenvalues of the matrix (constructed by the distances between the points) are used to cluster the similar objects. The major issue we faced during the implementation of spectral clustering was that the system and the cloud hardware that we were using, was unable to use it well. While PIC is far better algorithm and does work faster, we would like to see how well spectral clustering with NCut could perform and if there are any other takeaways from this kind of clustering. We would want to check if the clusters are better with this type of clustering.

Another implementation would be to implement deep neural network using various layers. Although, the original paper consists of VGG-16 architecture we did not implement it in out project since it is a very heavy model as compared to AlexNet. The VGG-16 model being more complex, it is huge possibility that the clusters created would be a lot better and more accurate. We would also like to compare it to a simpler CNN model to see if there if the difference in accuracy is extremely vast or not, consequently considering if the

computation cost of running such complex models is as significant.

9. Conclusion

Therefore, in out project we have implemented a scalable clustering approach for the unsupervised learning of the Convnets. It is a process of iteration between the clustering by k-means and the features produced by the convnets and the updating of weights by predicting the cluster assignments as pseudo labels.

10. References

[1] M. Caron, P. Bojanowski, A. Joulin and M. Douze, "Deep Clustering for Unsupervised Learning," 2018.

[2] Deepnotes.io/deep-clustering

[3] Introduction to the Architecture of AlexNet – Analytics Vidya

[4] PCA - Sartorius