Curriculum Learning Using Clustering

Introduction

Deep learning networks are constantly getting larger and deeper in order to solve more complex problems. As a result, the training process becomes gradually more time and energy consuming.
That is why a lot of effort and interest is invested recently in finding more efficient and faster ways to train deep neural networks.
In this project I aim to attack the problem from a different direction than the usual, by proposing a generic method that accelerates the training process, and can be composed with any existing architecture.
The basic idea is to exploit a clustering algorithm in order to create a curriculum to the model training process.
The proposed method modifies only the sampler, thus can be composed with any existing model without changing its unique architecture.

Short Review On Curriculum Learning

Humans and animals learn much better when the examples are not randomly presented but organized in a meaningful order which illustrates gradually more concepts, and gradually more complex ones. [1].
Curriculum learning describes a type of learning in which you first start out with the easy examples of a task and then gradually increase the task difficulty. As humans, I believe we are learning according to this principle, and it was interesting for me to try and apply it into neural networks.
The common way to apply curriculum learning nowadays, is to sort all the samples according to their assumed difficulties, then during the training procedure to provide the network with the samples in a sorted order, from the easiest to the hardest.

Methods

One of the trickiest part in curriculum learning is therefore determining the difficulty level of a sample. In the proposed method I used a simple Kmeans-clustering with the model loss to sort the samples.

More precisely, the difficulty hierarchy between the samples is determined by this pseudo-algorithm:

- Split training set into two groups: one is called the training group, and the other is called the difficulty-level-evaluation group.
- Run few warm-up epochs. Use only the training group for this task.
- Run another epoch, this time on both groups. Save the feature map that the model produces for each sample. Intuitively, this feature map represents the way the model ”sees” the sample.
- Apply the Kmeans-clustering algorithm on the feature maps of all samples.
- For each cluster, compute the loss of the model on each sample in this cluster that belongs to the difficulty-level-evaluation group. Then, compute the mean over those values.
- Finally, sort all clusters according to this mean value, from the hardest to the easiest.

Next, the sampler uses the above described sorted order between clusters for calculating each sample probability to be sampled in each epoch.
More formally, a sample which belongs to the i’th cluster (in the sorted order) has a probability to be sampled according to this equation:

$$p = min(e^{(-0.2 \cdot (M-i))}, 1) \quad (1)$$

Where M is a decreasing parameter, starts with a value of number-of-clusters, decreases each epoch by a fixed hyper-parameter DEC_M, to the minimal value of zero.
Notice that when M is equal to zero, according to this equation all samples will be to be chosen in the current epoch with probability 1.

Here are few examples which demonstrate the hierarchy that our algorithm creates.

Figure 1: easy cluster

Figure 2: easy cluster

Figure 3: hard cluster

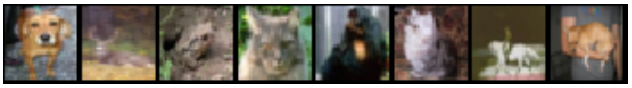

Figure 4: hard cluster

Figures 1, 2 show a visualization of some easy-level clusters. This picture is taken from a training done on CIFAR-10 dataset. This dataset contains only one class with a sky background. In this case, it makes sense that the model defines the cluster as easy because it could determine the class of the samples only by using the background color.
On the other hand, in figure 3 the cluster is assumed to be in a higher difficulty level. Samples in this cluster may be considered more confusing to the model because cars and trucks have a very similar shape and background, what makes it more difficult to distinguish between the two. Another interesting thing to notice here is that airplanes on the ground were also put in this cluster. This comes in contrast with the common approach in curriculum learning - which is simply splitting the samples according to their classes.
Similar conclusions can arise from figure 4.

Ease Of Use

The advantage of this method is that from the user point-of-view, this method replaces nothing but the sampler part. As most of the models usually use the naive sampler, this method can be added easily to any existing architecture and boost the model training time.

Results

First step was a POC experiment, which was done by applying the sampler into UPANets and training on CIFAR-10, with the exact same parameters from the UPANets article. [2]

Figure 5 shows the results from this training.
From these results it is possible to conclude that the proposed method saves approximately 20 percent of the training time.

Next, I used different model architecture and different dataset, to check whether the sampler works well independently of the network or the data, and therefore to prove its generality.

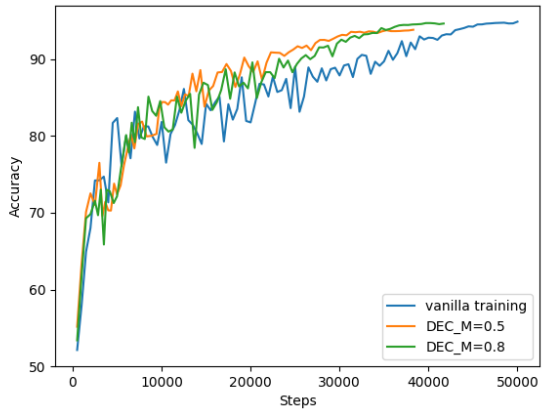

Figure 5: CIFAR-10 using UPANet results

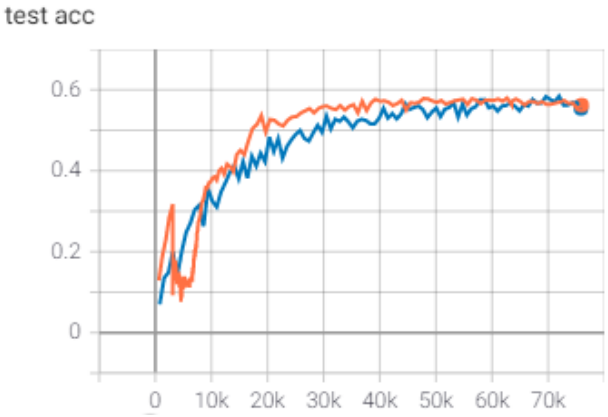

Figure 6: Tiny-ImageNet using Densenet results

I used the Tiny-ImageNet dataset, on the network taken from the following paper: ”DenseNet Models for Tiny ImageNet Classification”, which was the SOTA for tiny-imanget until 01-2021.
As shown in figure 6, the acceleration in the training process exists on Tiny-ImageNet as well. In step 40K the model that uses the proposed sampler reaches the same results as step 70k for the naive sampler.

Lastly, I wanted to examine if it is possible for the sampler to improve the model in terms of accuracy. Sadly, I didn't get any improvements in this part.

Conclusion

In an attempt to revive the field of curriculum learning I created a method that can accelerate the training process and decrease the amount of training steps needed to get to convergence by 20-35 percent.
The power of the method comes from its robustness to the dataset and the model architecture, and the fact that it can be easily composed in every existing model.

Future Studies

This project was mainly focused on small networks and small datasets due to lack in resources. Future work might be to generalize the idea to some other tasks in the field of computer vision, i.e Object Detection, or even NLP tasks.

References

[1] “Curriculum Learning”. In: (2009).

[2] ”Ching-Hsun Tseng Shin-Jye Lee Jia-Nan Feng Shengzhong Mao Yu-Ping Wu Jia-Yu Shang Mou-Chung Tseng Xiao-Jun Zeng”. “UPANET”. In: (2021).