Data Streaming Algorithms Final Project

Yarden Shoham
206561367

Daniel Neimark
204128656

February 2023

1 Introduction

Dimensionality reduction is a critical technique in machine learning, particularly when it comes to handling large datasets with numerous features. By reducing the number of dimensions, we can simplify the dataset and make it easier to analyze, visualize, and work with. There are various methods for dimensionality reduction, including principal component analysis (PCA), t-distributed stochastic neighbor embedding (t-SNE), and autoencoders. Each method has its own advantages and limitations, and the choice of which method to use depends on the specific requirements of the problem at hand.

In the context of edge IoT devices, dimensionality reduction is particularly important. These devices often have limited processing power, memory, and battery life, which makes it difficult to perform complex computations on large datasets. By reducing the dimensionality of the data, we can make it easier to process and transmit, which is crucial for achieving real-time performance in resource-constrained environments. Therefore, researchers are actively investigating ways to incorporate dimensionality reduction techniques into edge IoT devices, which would enable them to process and analyze large datasets more efficiently. This would have significant implications for a wide range of applications, from healthcare monitoring to industrial automation to smart homes.

Deep learning models have achieved remarkable success in various computer vision tasks. But using dimensionality reduction on the embedding of such models is not always trivial. One critical aspect of these models to be able to do dimensionality reduction is the choice of an appropriate loss function that enables them to learn useful features. In this work, we investigate the effectiveness of two loss functions for image classification: central loss and supervised contrastive loss.

Central loss [1] is a metric learning approach that encourages the network to learn discriminative features by minimizing the distance between the features of samples from the same class by learning a center for the deep features of each class and updates it during training to minimize the distances between the deep features and their corresponding class centers. On the other hand, supervised contrastive loss (SupContrast) [2] is a recently proposed contrastive learning approach that leverages negative examples in addition to positive examples to learn feature representations. Both loss functions aim to learn an embedding space that has L2 properties between vectors representing the same object in an image.

We also explore the effect of dimensionality reduction on the learned features using sparse random projection, principal component analysis (PCA), and trained auto-encoders. We use the k-Nearest Neighbor Classification (KNN) algorithm to evaluate the quality of the learned features in the reduced dimensionality space. Additionally, we analyze the confidence threshold of the KNN classifier for different dimensionality reductions.

The rest of the paper is organized as follows. In Section 2, we describe the experimental design and methodology used to evaluate the effectiveness of central loss and SupContrast. In Section 3, we present the results of our experiments, including the impact of dimensionality reduction on the learned features and the confidence threshold analysis of the KNN classifier. Finally, we conclude the paper in Section 4 with a summary of our findings and potential future work.

2 Experiments

In this section, we outline the experimental design and methodology used to evaluate the performance of two different loss functions for image classification: central loss [1] and supervised contrastive loss (SupContrast) [2].
Both loss functions encourage the network to learn an embedding layer with L2 properties between vectors representing the same object in the image. To evaluate the effectiveness of these loss functions, we conducted experiments on two well-known datasets for image classification, CIFAR-10, and CIFAR-100. CIFAR-10 contains 60,000 32x32 color images in 10 classes, with 6,000 images per class, while CIFAR-100 contains the same number of images but in 100 classes. We report top-1 accuracy for both datasets.

Our experimental design consisted of training a Resnet18 network with the relevant loss function. In the case of central loss, we modified the first layer by changing the 2D-convolution to be with 64 channels and a kernel size of 3, due to the fact that the Resnet architecture is not optimized for CIFAR datasets with inputs of 32x32. However, in the case of SupContrast loss, this modification was unnecessary.

We extracted features for both the training and validation datasets, and then determined the optimal k for using k-Nearest Neighbor Classification (KNN) [3] for the original feature dimension (512). Next, we used Sparse Random Projection, which relies on the Johnson-Lindenstrauss lemma, Principal Component Analysis (PCA) [4], and trained auto-encoders to reduce the original feature dimension into [256,128,64,32,16,4,2]. We then tested the results of the KNN algorithm on each reduced feature dimension using the same k.

For each KNN classifier, we analyzed the confidence threshold by defining a range of confidence thresholds tr=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,0.95]. If the probability that the KNN algorithm was less than the threshold, we ignored the prediction and reported accuracy only on the predictions that passed the threshold. We also report the percentage of the validation that passed the threshold. Our experimental design consisted of multiple parts, and the results from each part of the experiment are reported in the following sections.

The experiments were conducted using an Nvidia-2070ti GPU. We used a random crop for training, followed by random horizontal flip, random rotation, and normalization. For validation and feature extraction, we used only normalization. All other hyperparameters are documented in the git repository.

3 Results

First, we trained the backbone on CIFAR-10 dataset with Central loss and cross-entropy, we can see in the TSNE visualization in figure 1 that the loss indeed helped cluster the classes, the classifier achieved a top-1 accuracy of 94.69%, and the KNN algorithm on the embedding space with $k = 10$ achieved 94.72%, showing that even the KNN outperforms the classifier using the original embeddings. Next, we used the sparse random projection, PCA and autoencoders for dimensionality reduction, the results showed at figure 2 show that the sparse random projection and the PCA perform almost the same with a slight advantage to the PCA, both have similar results up to a very small dimension of 4 (PCA drops 0.39% at this dimension and the sparse random projection 0.68%), while the autoencoders performed the worse. Finally, we show the confidence threshold and the percentage of validation data that passed the threshold for a certain dimensionality reduction in figure 3, it is clear that as high the threshold the higher the accuracy, but here with selecting the right threshold it is clear that the random projection can yield high accuracy with a high percentage of data remaining.

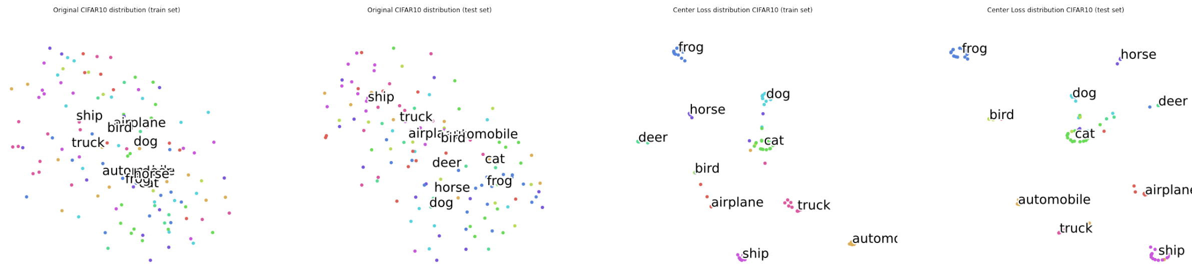

Figure 1: TSNE visualization of subset from CIFAR-10, before and after training the backbone with center-loss

We repeated the same experiments with the SupContrast loss and framework, and the results were similar, as shown in figure 4 (in the appendix), this set of experiments shows that as long as the loss forces the embedding space to be metric, our conclusion holds.

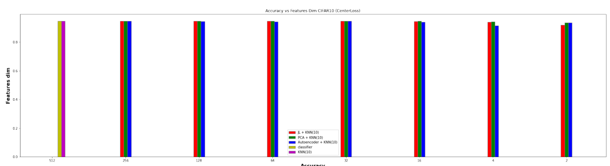

Figure 2: Results for the classifier and the KNN on the original embedding dimension and the results of the KNN with different dimensionality reduction methods

Finally, we run the same set of experiments on the CIFAR-100, the results are consistent with our findings so far, achieving an accuracy of 74.71% with the classifier, 74.59% with $k = 50$ on the original embedding space, here we can see a significant drop when reducing the embedding features to 4, the PCA dropped 6.43%, the random sparse projection 10.72% and the autoencoders 36.23%, while when the reduced dimension is 16 we get an accuracy drop of 0.36% for the PCA, 1.27% for the random sparse projection and 4.64% for the auto-encoder, this can be explained with the fact that this dataset has 100 labels, and there is a limitation for how low dimension the features can get without having an accuracy drop. Nevertheless, the threshold experiment yielded the same type of results, all presented in figure 5 (in the appendix)

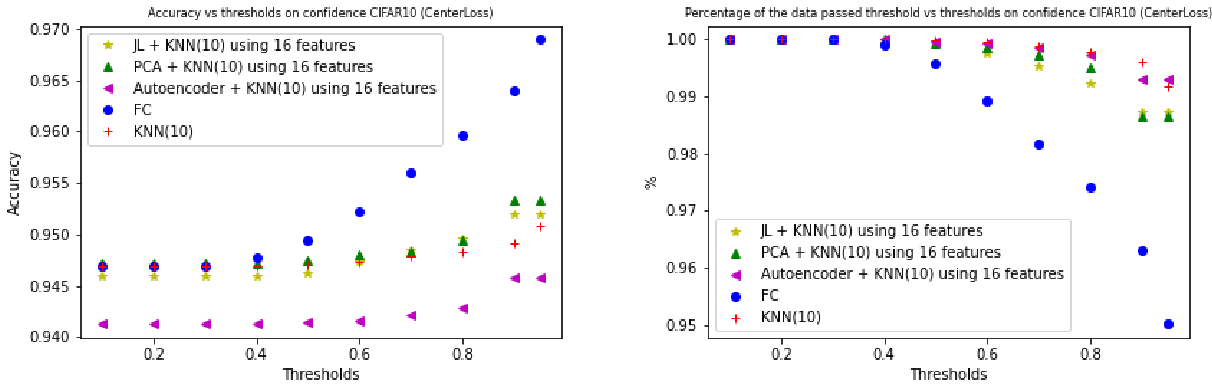

Figure 3: Left plot shows accuracy at a given threshold, the right plot shows the percentage of the data passed the threshold for given threshold

4 Discussion

From our experiment, we can conclude that the Principal Component Analysis and the Sparse Random Projection perform better than deep methods when the backbone is trained with loss that imposes the embedding features to have some metric relationship, this is validated across two datasets and two loss functions. Although the PCA has slightly higher accuracy in some scenarios, like in active learning cases, in which we would like to increase the number of examples that the KNN - classification is made from, we would benefit from using the sparse random projection, first, we can get from the original picture dimension of 3X32X32 into 512, and using the projection store only 16 features for each frame, then on each edge IoT device store the mapping of the train data and update it with samples classified with higher confidence that we choose, resulting with high probability better classifier, and eventually, we can sync with minimal overhead all the edge devices sets, as it compressed.
In addition, future work might extend this to other, bigger, and more complex datasets, or perhaps use different loss functions.

References

[1] Yandong Wen et al. "A Discriminative Feature Learning Approach for Deep Face Recognition". In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe et al. Cham: Springer International Publishing, 2016, pp. 499–515. ISBN: 978-3-319-46478-7.

[2] Prannay Khosla et al. "Supervised Contrastive Learning". In: *arXiv preprint arXiv:2004.11362* (2020).

[3] Antonio Mucherino, Petraq J. Papajorgji, and Panos M. Pardalos. "k-Nearest Neighbor Classification". In: *Data Mining in Agriculture*. New York, NY: Springer New York, 2009, pp. 83–106. ISBN: 978-0-387-88615-2. DOI: 10.1007/978-0-387-88615-2_4. URL: https://doi.org/10.1007/978-0-387-88615-2_4.

[4] Karl Pearson F.R.S. "LIII. On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572. DOI: 10.1080/14786440109462720.

5 Appendix

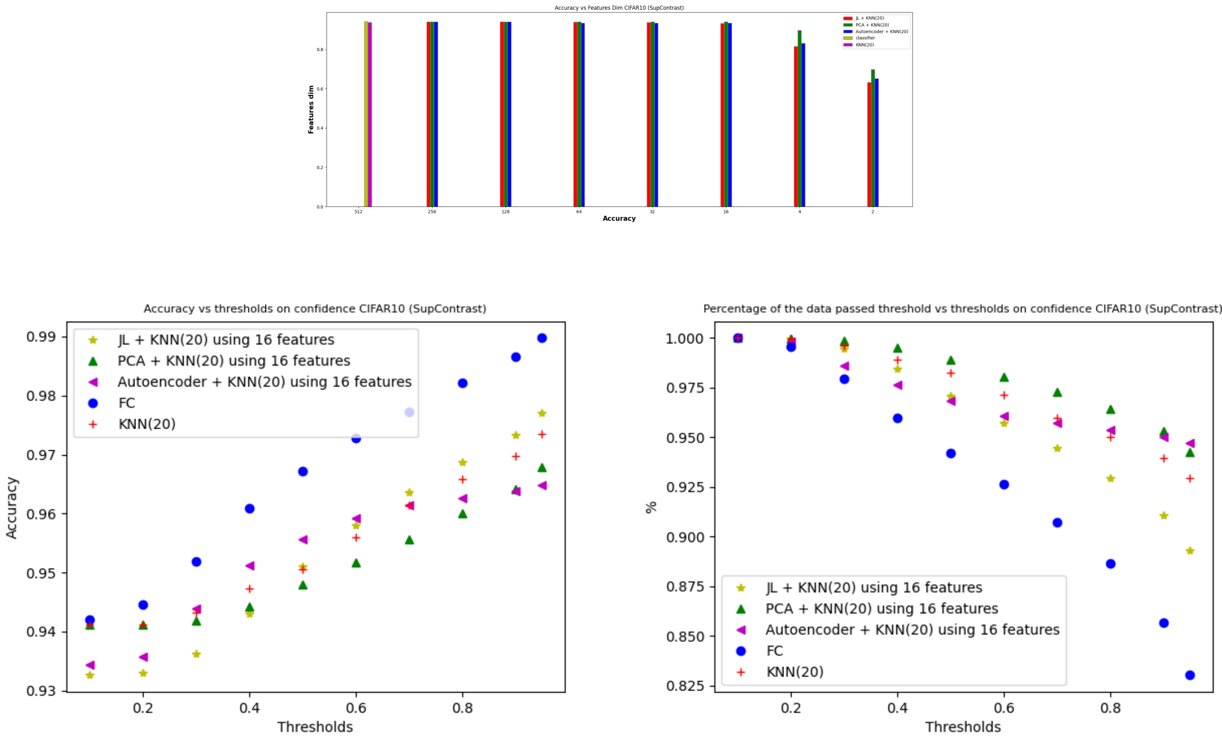

Figure 4: Results while training with supervised contrastive loss

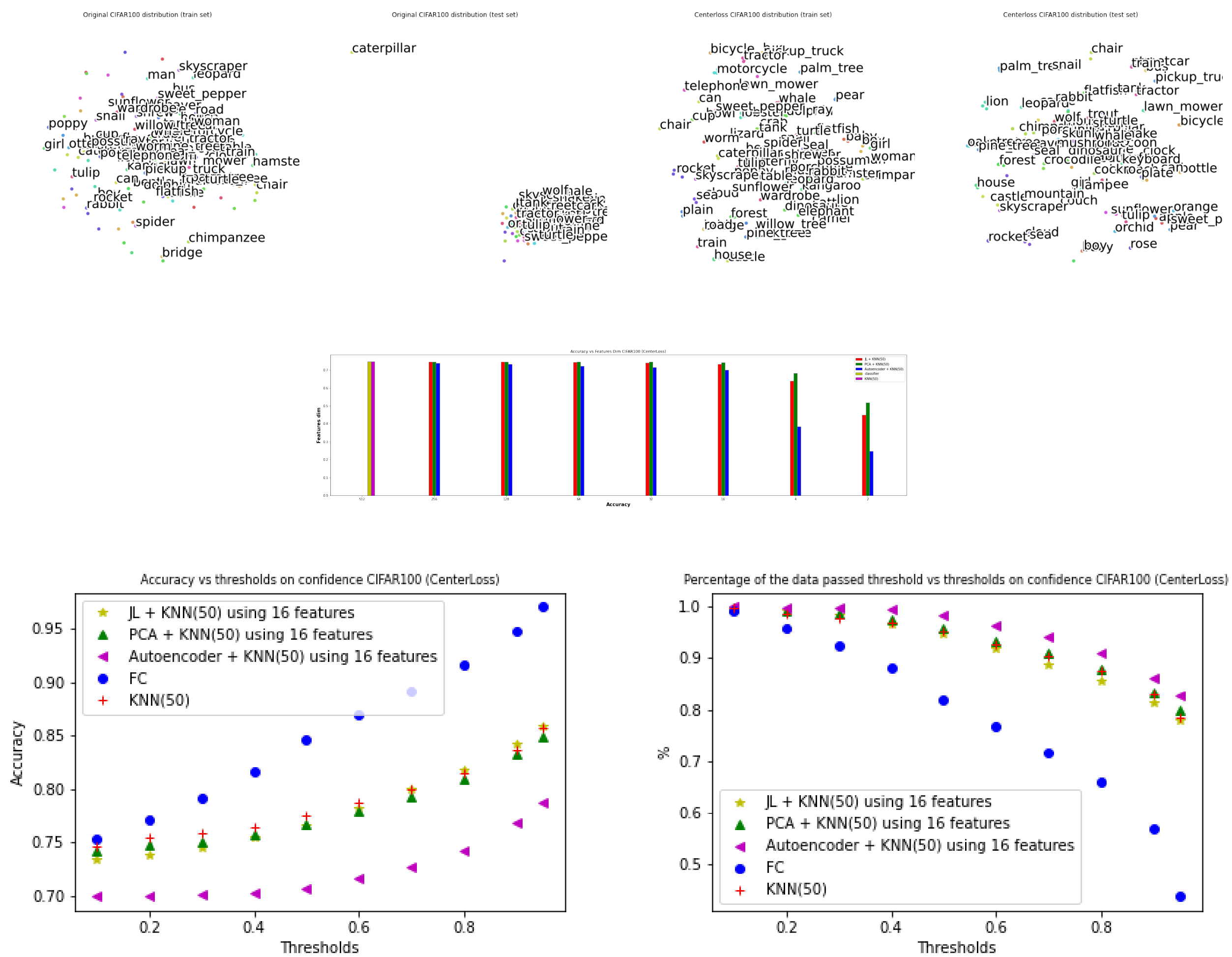

Figure 5: Results training on the CIFAR100 dataset with central loss