Names: Daniel Rodriguez, Sriharsha Aitharaju

Course: ISC 4242 Data Science II

Date: April 27, 2023

Comparative Analysis of Machine Learning Clustering Methods & Neural Networks in PyTorch for Image Generation on the Fashion MINST Dataset

Introduction

This project aims to evaluate the performance and effectiveness of traditional machine learning clustering methods, such as k-Nearest Neighbors (KNN), from the scikit-learn library against neural networks implemented in PyTorch. The dataset itself is enticing due to the composition of the matrix structure, as the set contains 60,000 images with a 28x28 pixel grayscale composition. To address this, we will begin by applying scikit-learn's k-Nearest Neighbors first and visualize. Following this preliminary test, we will begin with a rudimentary neural network that we will optimize further with various neural network architecture practices. Alternatives include convolutional neural networks, other methods of clustering contained within scikit-learn, such as DBSCAN and agglomerative clustering. The implementation leveraged is of primary interest due to its foundational structure. As k-Nearest Neighbors and "simplistic" neural networks are the core structure of the further developed alternatives.

Discussion

The Fashion MNIST dataset, consisting of 60,000 grayscale images of clothing items, will be used to assess the performance of each clustering method and the neural network. Due to the nature of the dataset, clothing images, we are more able to recognize and classify the images generated off of the clusters. By leveraging pixel clustering, this project will highlight the strengths and weaknesses of each approach with a visual representation highlighting the key differences, if any. These images are not necessarily easily separable in their feature space, making it a challenging problem for clustering algorithms. To further test efficacy we will also integrate random selection from the dataset, reducing to 5,000 points of data which also reduce computational cost. Additionally, since the dataset has been widely used for image classification tasks, it provides a useful benchmark for comparing the performance of clustering algorithms to that of other machine learning methods, such as KNN and neural networks in our case.

We are leveraging k-Nearest Neighbors and a "simplistic" neural network over the alternatives, which are better optimized for our task, in order to shift our primary focus onto algorithmic optimization. Since these algorithms are foundational to the alternatives listed prior this gives the opportunity to improve on a baseline implementation. With that said we chose the linear architecture to commence our neural network implementation, the advantage of which is to learn what exactly the network requires to interpret the dataset we have selected. We kept our encoding direct to only two linear layers, ingesting the matrix structure, and two rectified linear unit to introduce non-linearity. To expand upon this, we plan on leveraging various optimizers, criterions, increasing the complexity of the encoder, batch size optimization, dropout optimizations, learning rate optimization, and increasing the number of epochs once generated.

Implementation

In the initial stages of optimizing the neural network for clustering, the primary focus was on minimizing loss, as it serves as an indicator of the model's performance. However, it was observed that lower loss did not always result in better accuracy or clustering ability. This discovery highlights the importance of not solely relying on loss minimization as the sole optimization goal. Instead, the optimization process should also consider improving the model's accuracy and clustering ability, ensuring a well-rounded and robust model that performs well on various evaluation metrics. By striking a balance between loss minimization and other performance aspects, the neural network can be better optimized for clustering tasks.

Loss Minimization Overview

A total of 16 iterations were utilized to demonstrate various optimization strategies for a neural network working with the Fashion MNIST dataset. By adjusting factors such as the encoder size, loss criterion, optimizer, learning rate, batch size, and dropout rate, we can observe how different configurations affect the model's performance. This iterative process helps to fine-tune the model for better results. Prior to strategically iterating through potential improvements, a linear neural network was utilized with a start of 20 and 50 epochs to test effectiveness.

The strategies commenced with changing encoder size and complexity which leveraged different techniques from Leaky ReLU instead of ReLU to improve the model’s non-linear pattern recognition, integrating dropout for regularization, and sigmoid at the end to aid with pixel value reconstruction. Following such, the networks criterion and optimizer were tested with differing methods where the optimizer utilized Stochastic Gradient Descent to test a simpler configuration, in comparison to Adam which adapts the learning for each parameter. With said change, the criterion was also swapped with L1 loss, temporarily, instead of MSE to encourage sparsity in the model’s weights. With both configuration changes learning rates were also tested to allow for the model to learn more gradually and potentially find a better minimum.

All tested, the preferred model was to leverage MSE and Adam as the criterion and optimizer respectively. With the selected configuration it was time to test batch sizes and learning rates in order to test for improvements in gradient estimates and permitting the model to learn more gradually. After such the model itself was tested with different parameters for Dropout to introduce more regularization. To complete the loss minimization testing, an increase in epochs from 50 to 500 and finally to 2500 was to be used in attempt to further test the model.

Accuracy Testing Overview

After having compared the multitude of loss minimization techniques to generate the model with the lowest loss, while still not overfitting, it was found that the output images from the network did not resemble anything from the Fashion MNIST dataset. When comparing k-Nearest Neighbors and k-Means of the neural network, to test the resemblance of clustered images with the ones in the dataset, it was found that k-Means output was pixels based on the proximity to the image rather than the image itself (see Figure 1 below). Whereas the k-Nearest Neighbors generated pixel clusters more akin to that of clothing that would be contained in the Fashion MNIST dataset (see Figure 2 below). To understand whether the model was really accurate, WCSS and Silhouette Coefficient Score were tested to see how close the clusters were to the original images.

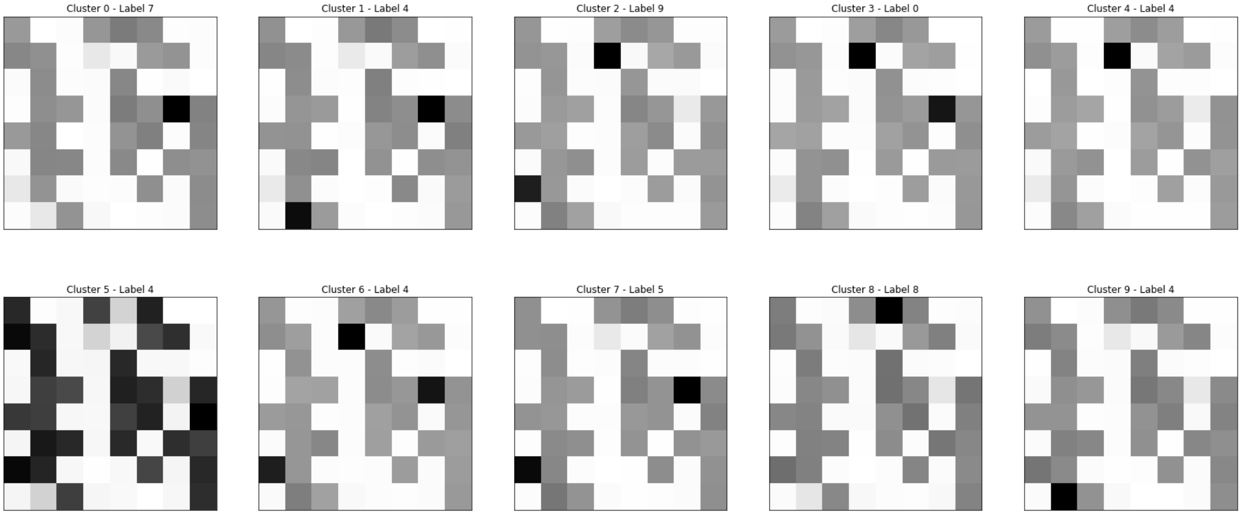

Figure 1 – k-Means Neural Network (500 Epochs, 0.001 LR, Large Encoder, Adam Optimizer, MSE Criterion, 64 Batch Size, 0.5 Dropout)

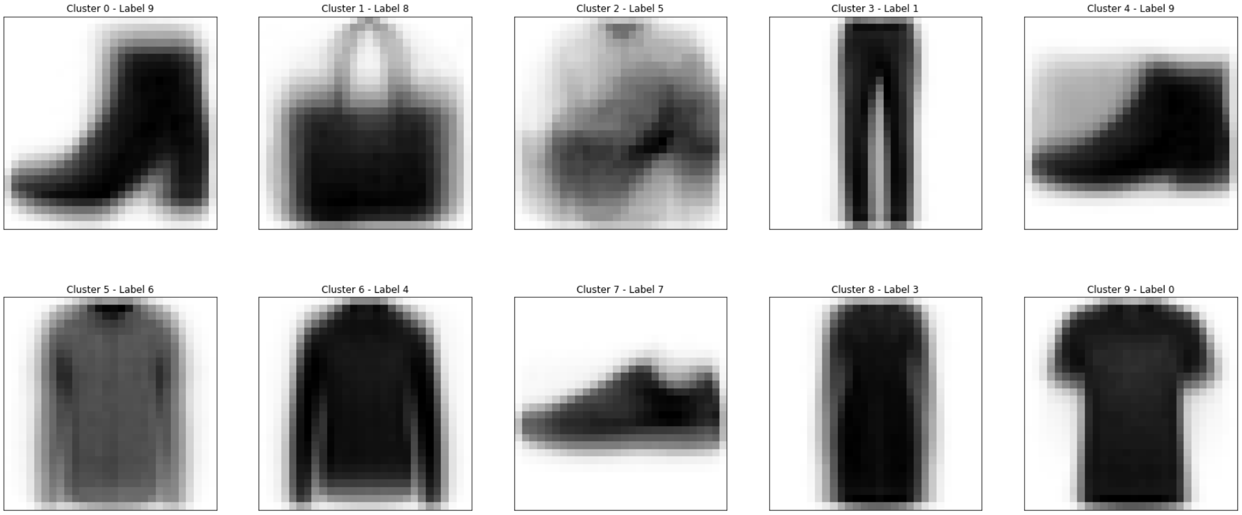

Figure 2 – k-Nearest Neighbors

Results

The Loss Minimization table below shows the collection of results from the iterative runs of each neural network method and configuration. The runs correlate directly with the overview previously provided and accompany said overview to give insight for the rationale of each weighted decision. Ultimately, we ended with a larger encoder model, using mean squared error as the criterion and Adam as the optimizer. That being said, the results were not necessarily apropos for the Fashion MNIST dataset and inquired further testing on the accuracy of the final neural network in comparison to the implementation of k-Nearest Neighbors, see Figure 1 and Figure 2.

Loss Minimization						
Iteration	# of Epochs	Encoder	Criterion	Optimizer	Learning Rate	Avg Loss
1	20	Small	MSE	Adam	0.001	2.37871E-06
2	50	Small	MSE	Adam	0.001	2.34342E-08
3	50	Big	MSE	Adam	0.001	6.17225E-06
4	50	Big	MSE	SGD	0.001	0.250532582
5	50	Big	L1	SGD	0.001	0.500615772
6	50	Big	L1	SGD	0.01	0.497479243
7	50	Big	MSE	Adam	0.001	9.60991E-06
8	50	Big	MSE	Adam	0.001	0.001051621
9	50	Big	MSE	Adam	0.001	0.004554313
10	50	Big	MSE	Adam	0.01	6.58817E-39
11	50	Big	MSE	Adam	0.1	0
12	50	Big	MSE	Adam	0.0001	0.011077632
13	50	Big	MSE	Adam	0.001	5.20533E-06
14	50	Big	MSE	Adam	0.001	1.15236E-05
15	500	Big	MSE	Adam	0.001	4.668E-07
16	2500	Big	MSE	Adam	0.001	2.10916E-07

Accuracy Testing Results

When calculating WCSS and Silhouette Score for the neural network algorithm, the calculations generated suggested that the sum of squared distances between the data points and their respective clusters is smaller, while the Silhouette coefficient values suggested that the separation between the clusters may not be as distinct. The k-Nearest Neighbors values suggested better separation between clusters possibly being a factor in determining the quality of the images. Although the neural network model was expected to perform better than the k-Nearest Neighbors model, since the Fashion MNIST dataset was cut down from 60,000 to 5,000 for testing purposes, this possibly led to a decrease in clustering quality.

	NN	KNN
WCSS	63621.27	1.04E+10
Silhouette Score	0.031618	0.137913