Clustering with Neural Networks: SHA-12 Model on MNIST Dataset

MD. Shadman Murshed

May 19, 2025

Abstract

This study presents the SHA-12 model(My name is Shadman and I born on 12 May.So, I named SHA-12), a Deep Embedding Clustering (DEC) neural network for unsupervised clustering of the MNIST dataset from Hugging Face. SHA-12 uses a convolutional autoencoder to create a 64-dimensional latent space, followed by K-Means clustering. Optimized with batch normalization, dropout, and L2 regularization, it achieves a Silhouette Score of 0.7925, a Davies-Bouldin Index of 0.3518, and a Calinski-Harabasz Index of 152847.9241. Compared to Self-Organizing Maps (SOM) and K-Means, SHA-12 shows strong performance but with areas for improvement. The report covers dataset analysis, hyperparameter tuning, model parameter counting, regularization techniques, clustering comparisons, accuracy labeling methods, and solutions to implementation challenges, highlighting SHA-12's potential in high-dimensional clustering tasks while identifying avenues for further enhancement.

1 Introduction

Clustering is a key unsupervised learning technique for grouping similar data points without labeled outputs [**?**]. The SHA-12 model, based on Deep Embedding Clustering (DEC), is designed to cluster the MNIST dataset, which includes 60,000 training and 10,000 test grayscale images of handwritten digits (0–9) [**?**]. Unlike traditional methods such as K-Means or Self-Organizing Maps (SOM), SHA-12 leverages neural networks to learn compact latent representations, aiming to improve clustering performance on high-dimensional data.

This report details the SHA-12 model's architecture, dataset analysis, hyperparameter optimization, and performance evaluation. It addresses the model's parameter count, regularization strategies, comparisons with existing clustering methods, cluster labeling approaches, and solutions to implementation challenges. The findings demonstrate SHA-12's effectiveness while pointing to areas for further development.

2 Methodology

2.1 Dataset Analysis

The MNIST dataset, sourced from Hugging Face, contains 60,000 training and 10,000 test images, each 28x28 pixels with a single grayscale channel. Analysis revealed:

- **Class Distribution**: 10 classes (digits 0–9) with nearly balanced distribution, verified by counting labels in the training set using `np.bincount`.
- **Data Characteristics**: Images were normalized to [0,1] by dividing pixel values by 255, ensuring

consistent input scales.

- **Preprocessing**: Data was batched (size 128), cached, and shuffled using TensorFlow Datasets to optimize training efficiency.

The balanced class distribution and preprocessing steps ensured unbiased clustering without significant data imbalances.

2.2 SHA-12 Model Architecture

The SHA-12 model employs a DEC framework with a convolutional autoencoder to learn a 64-dimensional latent representation for clustering [**?**]. The architecture consists of:

- **Encoder**: Three convolutional layers with 32, 64, and 128 filters (3x3 kernels), each followed by batch normalization, ReLU activation, and max-pooling (2x2). A dense layer projects the flattened output to a 64-dimensional latent space.
- **Latent Space**: A 64-dimensional bottleneck optimized for clustering using a Student's t-distribution to compute soft cluster assignments.
- **Decoder**: Symmetric to the encoder, with upsampling and deconvolutional layers to reconstruct the input image.
- **Clustering Layer**: Minimizes KL divergence between soft cluster assignments and an auxiliary target distribution.

The loss function combines reconstruction and clustering objectives:

$$L = L_{\text{reconstruction}} + \lambda \cdot L_{\text{clustering}},$$

where $L_{\text{reconstruction}}$ is the mean squared error, and:

$$L_{\text{clustering}} = KL(P\|Q) = \sum_i \sum_j p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right).$$

2.3 Model Parameter Counting

The SHA-12 model's trainable parameters were calculated as follows:

- **First Convolutional Layer**: $(3 \times 3 \times 1 \times 32) + 32 = 320$ (weights + biases).
- **Second Convolutional Layer**: $(3 \times 3 \times 32 \times 64) + 64 = 18,496$.
- **Third Convolutional Layer**: $(3 \times 3 \times 64 \times 128) + 128 = 73,856$.
- **Dense Layer**: $(128 \times 7 \times 7 \times 64) + 64 = 401,472$ (after flattening).
- **Decoder**: Symmetric to the encoder, approximately 493,664 parameters.
- **Total**: Approximately 987,808 trainable parameters.

Batch normalization layers contribute non-trainable parameters (e.g., mean and variance), but these are minimal.

2.4 Hyperparameter Optimization

Hyperparameters were tuned using grid search and cross-validation to maximize clustering performance:

- **Learning Rate**: Tested values $\{0.1, 0.01, 0.001, 0.0001\}$; 0.001 provided stable convergence.
- **Batch Size**: Evaluated $\{32, 64, 128, 256\}$; 128 balanced memory usage and gradient stability.
- **Epochs**: Set to 50 for pre-training and 30 for clustering, based on convergence of the loss function.
- **Loss Weight (λ)**: Tested $\{0.1, 0.5, 1.0\}$; 0.5 optimized the balance between reconstruction and clustering losses.
- **Latent Dimension**: Tested $\{32, 64, 128\}$; 64 yielded the best clustering metrics.

The Adam optimizer was selected for its adaptive learning rate capabilities, ensuring efficient training.

2.5 Regularization Techniques

To enhance generalization and prevent overfitting, the following techniques were applied:

- **Batch Normalization**: Implemented after each convolutional layer to normalize activations, stabilizing training and accelerating convergence.
- **Dropout**: Applied at a 0.3 rate after dense layers in the encoder to randomly drop units, reducing overfitting.
- **L2 Regularization**: Added to convolutional and dense layers with a weight decay of 0.0001 to penalize large weights, improving model robustness.

2.6 Training Strategy

Training occurred in two phases:

1. **Pre-training**: The autoencoder was trained for 50 epochs to minimize reconstruction loss, ensuring robust latent representations.
2. **Clustering**: The clustering layer was initialized, and the model was fine-tuned for 30 epochs with the combined loss. K-Means clustering (10 clusters) was then applied to the latent embeddings.

Training was performed on a CUDA-enabled GPU using TensorFlow, with data caching and prefetching to optimize performance.

3 Results

3.1 Clustering Performance

SHA-12 was compared against SOM and traditional K-Means (applied to flattened MNIST images). The results are summarized in Table 1.

SHA-12 outperformed both baselines, achieving a Silhouette Score of 0.7925 (vs. 0.4777 for SOM, 0.5123 for K-Means), a Davies-Bouldin Index of 0.3518 (vs. 0.7270 for SOM, 0.6894 for K-Means), and a Calinski-Harabasz Index of 152847.9241 (vs. 7089.3218 for SOM, 8214.5673 for K-Means). The KL divergence of 0.1423 (vs. SOM's quantization error of 5.2572) indicates effective but improvable cluster assignments.

Table 1: Comparison of Clustering Methods on MNIST

Metric	SHA-12 (DEC)	SOM	K-Means	Best Improvement (%)
Silhouette Score	0.7925	0.4777	0.5123	65.9 (vs. SOM)
Davies-Bouldin Index	0.3518	0.7270	0.6894	51.6 (vs. SOM)
Calinski-Harabasz Index	152847.9241	7089.3218	8214.5673	2055.6 (vs. SOM)
KL / Quantization Error	0.1423	5.2572	N/A	97.3 (vs. SOM)

3.2 Clustering Accuracy Labeling

To evaluate clustering accuracy in an unsupervised context, clusters were mapped to true digit labels using:

- **Hungarian Algorithm**: Aligned cluster assignments with ground-truth labels to maximize correspondence, ensuring optimal matching.
- **Adjusted Rand Index (ARI)**: Measured agreement between cluster assignments and true labels, yielding an ARI of 0.85, indicating good alignment with digit classes but with potential for refinement.

These methods confirmed that SHA-12's clusters captured the underlying structure of the MNIST dataset, though some misalignments suggest room for improvement.

4 Discussion

4.1 Comparison with Existing Methods

SHA-12's DEC approach surpasses SOM and K-Means by learning compact, discriminative latent representations. SOM's topological mapping struggles with high-dimensional data, resulting in a high quantization error (5.2572). K-Means, applied directly to pixel data, lacks feature extraction, leading to poorer metrics. SHA-12's 65.9% improvement in Silhouette Score and 51.6% in Davies-Bouldin Index over SOM demonstrate its effectiveness, though the Davies-Bouldin Index of 0.3518 suggests that cluster separation could be enhanced.

4.2 Limitations and Obstacles

Several challenges were encountered during development:

- **Computational Cost**: Training the deep neural network required significant GPU resources. *Solution*: Optimized I/O with TensorFlow's data caching and prefetching.
- **Hyperparameter Sensitivity**: Clustering performance varied with λ and latent dimension. *Solution*: Conducted grid search and used early stopping based on validation loss.
- **Cluster Initialization**: Poor initial cluster centers affected K-Means performance. *Solution*: Adopted k-means++ initialization for better starting points.
- **Labeling Challenges**: The unsupervised nature complicated accuracy assessment. *Solution*: Employed the Hungarian algorithm and ARI for robust evaluation.

4.3 Future Work

Future improvements could include:

- Integrating contrastive or triplet losses to enhance latent representations, potentially improving the Davies-Bouldin Index.
- Testing SHA-12 on diverse datasets (e.g., ag_news, glue/sst2) to evaluate generalizability.
- Reducing computational demands through model pruning or quantization techniques.
- Fine-tuning the latent dimension and λ to further reduce the KL divergence.

5 Conclusion

The SHA-12 model demonstrates strong clustering performance on the MNIST dataset, leveraging a DEC-based convolutional autoencoder. With a Silhouette Score of 0.7925, Davies-Bouldin Index of 0.3518, and Calinski-Harabasz Index of 152847.9241, it outperforms SOM and K-Means, though metrics like the Davies-Bouldin Index and KL divergence (0.1423) indicate areas for improvement. Thorough dataset analysis, hyperparameter tuning, regularization, and solutions to computational and labeling challenges ensured robust results. SHA-12 shows promise as a tool for unsupervised clustering, with clear pathways for enhancement in future work.