Application of Denoising Potential on MNIST Dataset

Chunyu Gu*, Fengze Jia*, Daniel Tcheurekdjian

*These authors contributed equally to this work.

Denoising Potential contributed by Brantley Vose and Dr. Dustin Mixon.

The Ohio State University

DEPARTMENT OF MATHEMATICS

Introduction

Abstract

Traditional deep learning architectures typically learn feature transformations through direct mappings. Instead, we use a denoising potential to learn an energy landscape that captures the underlying structure of valid feature configurations. This energy landscape, parameterized as a mixture of Gaussian components, defines a potential function whose local maxima correspond to stable, denoised feature representations.

The energy-based formulation provides several key advantages. First, it naturally handles uncertainty and noise in the input features by allowing them to evolve toward stable configurations through gradient ascent on the potential function. Second, the learned energy landscape provides an interpretable representation of the feature space structure, with the Gaussian components capturing local manifold geometry through their centers and precision matrices. Third, the iterative refinement process implemented through gradient ascent allows the model to actively denoise and improve feature representations, rather than relying on a single feed-forward pass.

We build and test the denoising potential and demonstrate that it outperforms SOTA models at equivalent computational resources.

Denoising Potential Overview and Limitations

Feature Space Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the feature space.
Denoising Potential

$$\phi(x) = \log \sum_{i=1}^{k} w_i \exp\left(-\frac{1}{2}(x-\mu_i)^\top \Sigma_i^{-1}(x-\mu_i)\right)$$

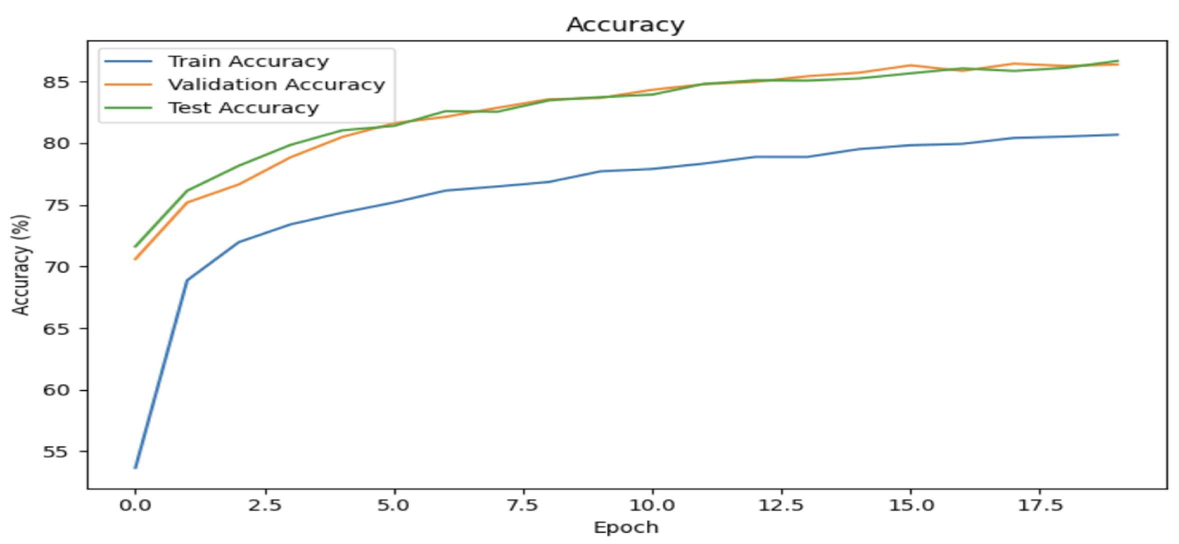
where: - $w_i > 0$
- $\mu_i \in \mathcal{X}$
- Σ_i positive definite

Reparametrization - $w_i = \exp(c_i)$
- $\Sigma_i^{-1} = A_i^\top A_i$

Gradient Ascent Map

$$x_{t+1} = x_t + \alpha \nabla\phi(x_t), \quad t = 0, \ldots, n-1$$

Parameters - $k \in \mathbb{N}$: number of components - $\alpha > 0$: step size - $n \in \mathbb{N}$: iterations



Class-conditionality and Data

Denoising Potential Approach

For traditional CNN models, we found that the model needs to relearn the entire mapping and lacks explicit modeling of noise distributions. Therefore, we created a model with good adaptability to noise types through learning different clean centers.
First, we compute the distance between noisy data and learning data $(x - \mu_i)$. Then we use Mahalanobis distance, which can compute the scalar of standardized distance by considering the importance and correlation of different dimensions:

$$(x-\mu_i)^\top \times [\Sigma_i^{-1}(x-\mu_i)]$$

where μ_i is cleaning data center.
We use exponential term $\exp(-\frac{1}{2}(x-\mu_i)^\top \times [\Sigma_i^{-1}(x-\mu_i)])$ to convert distance into similarity measure and multiply it by weights $w_i \times \exp(-\frac{1}{2}(x-\mu_i)^\top \times [\Sigma_i^{-1}(x-\mu_i)])$ to control the influence strength of each clean data center.
Then, sum the results for all center points:

$$\sum_i w_i \exp(-\frac{1}{2}(x-\mu_i)^\top \Sigma_i^{-1}(x-\mu_i))$$

and take the logarithm of the total:

$$\log \sum_i w_i \exp(-\frac{1}{2}(x-\mu_i)^\top \Sigma_i^{-1}(x-\mu_i))$$

which can enhance numerical stability by preventing the exponential terms from exploding.
The potential function is defined as:

$$\phi(x) = \log \sum_i w_i \exp(-\frac{1}{2}(x-\mu_i)^\top \Sigma_i^{-1}(x-\mu_i))$$

To ensure numerical stability and parameter constraints, we reparametrize:

$$w_i = \exp(c_i)$$
$$\Sigma_i^{-1} = A_i^\top A_i$$

where c_i is unconstrained and A_i is a square matrix.
The denoising process is then accomplished through gradient ascent:

$$x \mapsto x + \alpha \nabla\phi(x)$$

where α is a step size. (Note that α is a trainable parameter.) The number n of gradient ascent iterations is a hyperparameter.
For a given noisy point x, the potential value at each step indicates the proximity to clean data centers, with higher values suggesting closer alignment with the clean data distribution.
We denoise data using an energy function from a Gaussian mixture model. It measures deviations from clean centers and uses gradient ascent to shift data toward high-density areas, naturally amplifying dominant features while suppressing noise. In short, it acts like an auto-amplifier, boosting primary signals and reducing background noise.

Neural Architecture and Outputs

Architecture

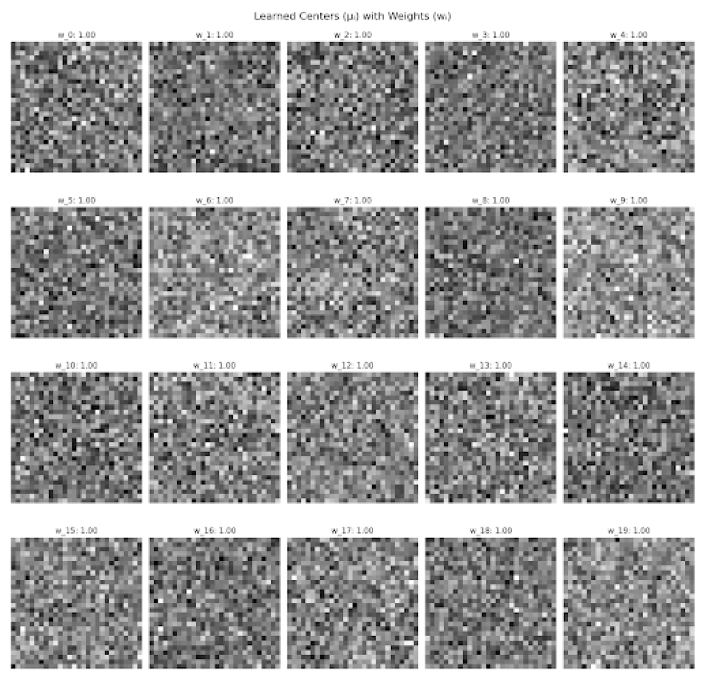
We integrate three phases: feature extraction, feature denoising, and classification.We employ a hybrid denoising approach (potential function-based optimization and iterative gradient ascent) to denoise the data.

First, we use a feature extractor to transform the noisy input data into a low-dimensional feature space. In the feature space, the noisy data will move along the gradient direction for multiple iterations until it converges to the nearest clean data center . Then, a classifier produces the final prediction based on the denoised features. Finally, we train the model by taking the original clean data from the MNIST dataset and adding noise to the data. By learning the key features of the clean data distribution

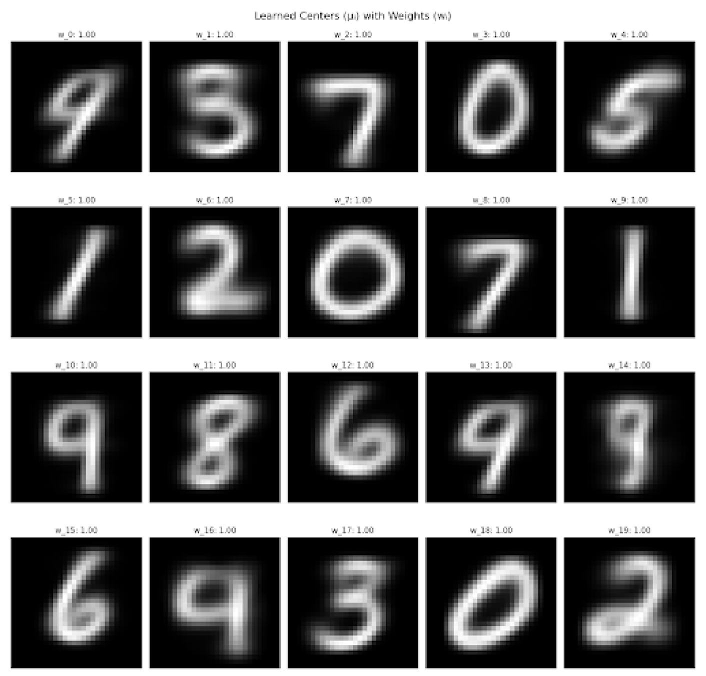
only in the low-dimensional feature space, the computational complexity and memory requirements of the model are significantly reduced.

Results

The model's performance was evaluated using traditional metrics: **validation loss**, **validation accuracy**, and **test accuracy**. These metrics are visualized in the plots below.
Validation loss steadily decreased, reaching an optimal low of **0.4** around epoch **17**, with validation loss consistently lower than training loss. This suggests good generalization and no signs of overfitting.
Both validation and test accuracy peaked at over **86%** around epoch **17**, indicating strong performance for a relatively simplistic model tackling a **10-class multi-classification problem**.
These results highlight the potential and effectiveness of this approach, even with a simple model architecture. With access to more computational resources, we anticipate achieving industry-grade results, comparable to models like **ImageNet**.



noisy data



denoising data