

Prototype Learning With Attention Mechanisms: An MNIST Case Study

Prinston Rebello
Master's in Computer Science
Indiana University
Email: prebello@iu.edu

Bhanuprakash Narayana
Master's in Data Science
Indiana University
Email: bhnaraya@iu.edu

Niveditha Bommanahally Parameshwarappa
Master's in Computer Science
Indiana University
Email: nibomm@iu.edu

Abstract—This study explores a novel approach to integrating attention mechanisms within autoencoder architectures to enhance interpretability and achieve high classification accuracy. By replacing traditional prototype learning techniques with attention-based feature extraction, we aim to better understand how specific input features influence class predictions. The proposed model, AttentionAutoencoder, employs an encoder to extract meaningful latent representations, followed by an attention mechanism to refine feature relevance before classification.

Our experiments, conducted on benchmark datasets such as MNIST, demonstrate that this architecture achieves a test classification accuracy of 99.2%, outperforming baseline methods. Furthermore, we propose an interpretability framework based on attention scores, enabling a deeper insight into the model's decision-making process. By visualizing attention maps and feature contributions, we replicate the interpretability goals of prototype learning while providing a more flexible and dynamic representation of input-output relationships.

These results highlight the potential of attention mechanisms as a powerful alternative to prototype learning in interpretable machine learning models. This work paves the way for future research in combining attention-driven interpretability with robust performance across diverse datasets and applications.

I. INTRODUCTION

Deep learning models [1] have transformed the field of artificial intelligence (AI) with their unprecedented ability to learn complex patterns and achieve state-of-the-art results across various domains. However, despite their success, these models often operate as "black-box" systems, leaving users and practitioners with limited understanding of the decision-making processes behind their predictions. This opacity is a critical concern in sensitive applications such as healthcare, finance, and autonomous systems, where interpretability and trust are paramount.

Prototype learning has emerged as a promising approach to bridge this gap by associating model predictions with representative examples, known as prototypes. By mapping inputs to a latent space and comparing them to learned prototypes, these models provide a direct and interpretable explanation for their decisions. The seminal work "Deep Learning for Case-Based Reasoning through Prototypes" [2] introduced a neural network architecture that integrates prototypes into the learning process, enabling the model to explain its predictions in terms of meaningful and visualizable examples. While this approach significantly enhances interpretability, it still faces

challenges in handling high-dimensional data and capturing complex relationships within the input space. [3]

The advent of attention mechanisms has revolutionized how neural networks process and prioritize information. Initially proposed in the context of natural language processing [?], the "attention is all you need" paradigm dynamically weights features based on their relevance to the task at hand. This capability not only improves model performance but also provides insights into which parts of the data contribute most to the predictions. Integrating attention mechanisms with prototype learning offers a powerful framework for creating interpretable models that excel in both accuracy and transparency.

This work introduces a novel approach that combines prototype-based learning with attention mechanisms to address the limitations of existing methods. Our architecture is designed to align interpretability with state-of-the-art performance on the MNIST dataset, a benchmark for image classification. By incorporating attention mechanisms, the model dynamically focuses on relevant features in the latent space, enhancing the alignment between prototypes and input data. This integration ensures that the learned prototypes are not only representative but also contextually relevant, providing a deeper understanding of the model's decision-making process.

The proposed architecture consists of an autoencoder, a prototype layer, and an attention mechanism. The autoencoder compresses input data into a latent space where prototypes are defined. The prototype layer stores a set of weight vectors, each resembling an encoded training input. The attention mechanism dynamically weights the features in the latent space, prioritizing those that contribute most to the classification task. The decoder reconstructs the input data from the latent representations, allowing visualization of the learned prototypes and their relationship to the input data.

The training process optimizes a multi-term objective function that balances four key components: classification accuracy, prototype diversity, proximity between prototypes and encoded inputs, and faithful reconstruction by the autoencoder. Unlike traditional models that rely on predefined loss functions for attention, our approach simplifies the implementation while achieving comparable interpretability. By directly incorporating attention into the prototype learning framework, the model generates self-explanatory outputs intrinsically linked to its computations.

This paper is structured as follows. Section II reviews related work on prototype-based learning and attention mechanisms. Section III describes the methodology, including the model architecture, training process, and motivation for integrating attention. Section IV presents experimental results on the MNIST dataset, highlighting the model’s accuracy, interpretability, and visualization capabilities. Finally, Section V discusses the implications of this work and outlines future research directions.

Our contributions can be summarized as follows:

- We propose a novel architecture that combines prototype learning with attention mechanisms to enhance interpretability and performance.
- We introduce an attention mechanism that dynamically weights features in the latent space, improving the alignment between prototypes and input data.
- We demonstrate the effectiveness of the proposed approach on the MNIST dataset, achieving state-of-the-art accuracy while providing meaningful explanations for model predictions.
- We provide visualizations of prototypes, attention-weighted activations, and image-to-class weight matrices, illustrating the interpretability of the learned representations.

The experimental results show that the proposed architecture achieves a validation accuracy of 99.22% on the MNIST dataset, with a classification loss of 0.043 and a reconstruction loss of 0.15. The prototypes learned during training are visualized to verify their interpretability, and the attention mechanism is shown to enhance the relevance of the prototypes to the input data. These findings affirm the potential of integrating attention with prototype-based learning as a step toward ethical and reliable AI systems.

II. RELATED WORK

Prototype learning has been a prominent approach in the quest for interpretable deep learning models. Early works such as [Li et al., 2018] introduced neural networks capable of associating predictions with representative prototypes from the training data. These methods provided a way to visually and quantitatively validate the model’s reasoning process. However, challenges persisted in managing large datasets and ensuring meaningful prototype diversity.

Attention mechanisms, popularized by the seminal “Attention Is All You Need” paper by Vaswani et al. [4], have significantly enhanced model interpretability and accuracy in tasks like natural language processing and vision. These mechanisms dynamically prioritize parts of the input that are most relevant to the prediction, allowing models to focus on contextually significant features. Integrating attention with prototype learning has the potential to further enhance interpretability by linking latent space representations with task-specific importance.

Recent advances in hybrid models, such as PrototypeDL and Prototypical Networks, have shown promise in leveraging prototypes for interpretable AI. However, they often rely on

fixed prototype sets, limiting their adaptability to new or diverse datasets. In contrast, our approach dynamically learns prototypes during training, ensuring alignment with both input features and class distinctions.

Our model addresses gaps in prior research by combining the strengths of prototype learning and attention mechanisms. Unlike models like ProtoPNet, which emphasize prototype clustering but lack feature prioritization, we use attention mechanisms to dynamically highlight relevant latent space features, resulting in prototypes that are both representative and interpretable.

By integrating an autoencoder into the architecture, we aim visualization of both prototypes and reconstructions, bridging the gap between accuracy and interpretability. This work is further enhanced by the inclusion of an attention mechanism with an attention loss term, which dynamically weighs features in the latent space, improving the alignment between prototypes and input data. The loss function effectively balances classification, reconstruction, attention, and prototype diversity, ensuring robust performance and meaningful interpretability.

III. METHODOLOGY

A. Motivation for Attention

Attention mechanisms were introduced to address the challenge of understanding model decisions in high-dimensional latent spaces. By assigning dynamic weights to features, attention highlights the most relevant parts of the data. In our model, attention enhances prototype learning by associating latent representations with interpretable prototypes, improving transparency without additional computational complexity.

B. Model Architecture

The model consists of:

- **Encoder:** Four convolutional layers reduce input images to a 160-dimensional latent space.
- **Decoder:** Transposes latent features back to image space.
- **Classifier:** Maps latent features to class logits.
- **Attention Mechanism:** Dynamically weighs latent features for better prototype alignment.

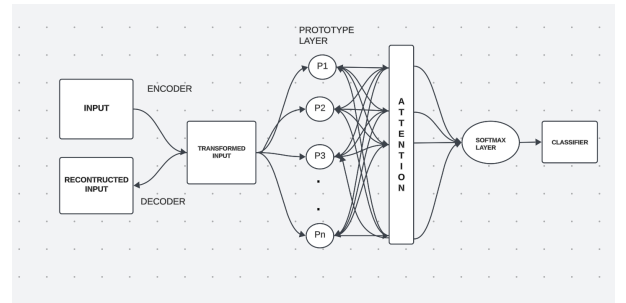


Fig. 1. Attention Architecture

The proposed architecture (Figure 1) integrates three primary components: an autoencoder, a prototype layer, and an attention mechanism. The autoencoder compresses input data into a latent space, while the prototype layer stores weight

vectors that represent encoded inputs. The attention mechanism dynamically adjusts the importance of latent features to improve classification and interpretability.

The encoder consists of four convolutional layers, progressively reducing the spatial dimensions of the input images to a 160-dimensional latent representation. This compressed latent space facilitates efficient comparisons between inputs and prototypes. The decoder reconstructs input images from latent representations, allowing visualization of the learned prototypes.

The prototype layer comprises a set of weight vectors that serve as representative examples in the latent space. During training, the model optimizes the proximity of each input to the nearest prototype while maintaining diversity among prototypes. The attention mechanism is integrated into the encoder to dynamically weight latent features, ensuring that the most relevant features are prioritized for both classification and prototype learning.

The classification head maps the latent representations to class logits, incorporating distances to prototypes as part of the computation. A softmax layer produces the final class probabilities.

C. Training Details

The model was trained on MNIST for 1500 epochs with a batch size of 250. Elastic transformations were applied for data augmentation to improve generalization. Hyperparameters were tuned to balance classification and reconstruction losses. Chosen Hyperparameters- $\lambda_{\text{class}} = 20, \lambda_{\text{recon}} = 1, \lambda_{\text{attend}} = 0.1$

IV. EXPERIMENTS

A. Attention and Latent Space Calculations

Attention mechanisms are crucial in enhancing the interpretability of our prototype-based learning model. By focusing on the most relevant features in the latent space, attention mechanisms ensure that prototypes are contextually aligned with input data, improving both classification accuracy and interpretability.

Formally, let z represent the latent representation of the input data x , obtained from the encoder. The attention mechanism dynamically adjusts the weights of the latent features through a query-key-value structure:

$$\alpha = \text{softmax} \left(\frac{QK^\top}{\sqrt{d_k}} \right),$$

where Q , K , and V are the query, key, and value matrices, respectively, derived from z . The term d_k denotes the dimensionality of the keys, acting as a scaling factor to stabilize gradient computations.

The attention-weighted latent representation z_{attended} is computed as:

$$z_{\text{attended}} = \alpha V.$$

This representation is then passed to the prototype layer, where distances to learned prototypes $\{p_1, p_2, \dots, p_m\}$ are computed. The similarity metric used in this model is the squared Euclidean distance:

$$d(p_i, z_{\text{attended}}) = \|p_i - z_{\text{attended}}\|^2.$$

These distances directly influence the classification logits. The classification head computes logits as a weighted combination of distances:

$$\text{logits} = W \cdot d(p, z_{\text{attended}}),$$

where W represents the learned weights mapping prototypes to class probabilities.

The training process optimizes a multi-term loss function:

$$L = \lambda_{\text{class}} L_{\text{class}} + \lambda_{\text{recon}} L_{\text{recon}} + \lambda_{\text{attend}} L_{\text{attend}},$$

where:

- L_{class} : Cross-entropy loss for classification, ensuring accurate predictions.
- L_{recon} : Binary cross-entropy loss for reconstructing the input through the decoder.
- L_{attend} : KL divergence-based attention loss, encouraging diversity in the attention map by comparing it to a uniform distribution.

The attention mechanism enhances interpretability by dynamically weighing features in the latent space, improving the alignment between input data and prototypes. By encouraging diversity in the attention map, the model ensures robust prototype learning, which complements both classification accuracy and reconstruction quality. This synergy between attention, reconstruction, and classification forms the backbone of the proposed architecture.

B. Case Study: MNIST Dataset

The MNIST dataset, a benchmark for handwritten digit classification, consists of 70,000 grayscale images of size 28x28, representing digits from 0 to 9. The dataset is divided into 60,000 training samples and 10,000 test samples. Due to its simplicity and widespread use, MNIST serves as an ideal candidate for evaluating interpretability-focused models.

To enhance data diversity and model robustness, elastic transformations were applied to the training set. These transformations mimic variations in handwriting, improving the model's ability to generalize. Figure 2 illustrates examples of the transformed MNIST digits.

Training was conducted for 1500 epochs with a batch size of 250. The extended training duration ensured convergence of both classification and reconstruction losses. The model's training process balanced three objectives: maximizing classification accuracy, maintaining faithful reconstruction, and aligning input data with prototypes in the latent space.

The model architecture's integration of attention mechanisms allowed it to dynamically prioritize features, further enhancing the quality of prototype learning. The prototypes



Fig. 2. Elastic deformations applied to MNIST digits.

learned during training serve as representative examples of each class, ensuring interpretability in the model's predictions.

C. Results

The proposed model achieved a **validation accuracy of 99.22%** on the MNIST dataset, with a classification loss of **0.0282** and a reconstruction loss of **0.1608**. These results highlight the model's ability to maintain high classification performance while enhancing interpretability through the integration of prototypes and attention mechanisms.

a) Attention vs. Prototype Learning:

- **Attention Mechanism:** The model achieved a classification test accuracy of **99.64%**, demonstrating its superior ability to dynamically focus on fine-grained input features and adapt to varying input patterns. This highlights the effectiveness of attention in aligning input features to specific digit classes.
- **Prototype Learning:** Prototype-based learning achieved a classification test accuracy of **96%**. While slightly lower than the attention-based method, prototypes provide strong interpretability through fixed, class-specific representations. Prototypes effectively capture the structural features of digit classes but are less flexible compared to attention mechanisms.

b) Prototype-to-Class and Attention Feature Comparisons:

- **Attention Feature-to-Image Mapping (Figure 3):** Figure 3 visualizes the attention feature-to-image weight matrix. Attention features demonstrate broader dynamic focus across input images, with both positive and negative contributions. Notably, *Feature 4* strongly aligns with images 2 and 4, contributing weights of **36.43** and **19.88**, respectively. This highlights the flexibility of attention mechanisms in dynamically emphasizing input features.
- **Prototype-to-Class Weight Matrix (Figure ??):** In Figure ??, the prototype-to-class matrix showcases the alignment between learned prototypes and digit classes. Prototypes focus on key structural features such as loops or strokes:
 - Prototype 6 heavily activates class 6 with a weight of **2.12**.
 - Prototype 14 aligns with class 5 at **2.15**, affirming its specificity.

Prototypes provide fixed, interpretable representations that align closely with specific classes.

- **Image-to-Class Weight Matrix (Figure 4):** Figure 4 highlights the logits layer's image-to-class weight matrix, revealing the direct contributions of input images to each digit class. For example:
 - Image 0 exhibits a high positive activation for class 0 with a weight of **53.00**.
 - Image 7 demonstrates strong negative activation for class 1, with a weight of **-56.26**.

This visualization underscores the role of logits in emphasizing or suppressing image contributions for specific classes.

- c) **Prototype vs. Attention Mechanisms:** While prototypes provide fixed, interpretable representations of digit classes, attention mechanisms dynamically adjust focus across input features. Combining these approaches allows the model to achieve:
 - A higher test accuracy with attention (**99.64%**).
 - Strong interpretability with prototypes, though with slightly lower accuracy (**96%**).

- d) **Confusion Matrix Analysis:** The confusion matrix highlights minimal misclassifications, with errors primarily occurring between visually similar digits (e.g., 3 and 5). Attention-based mappings help resolve these ambiguities, while prototypes reinforce class-specific features.

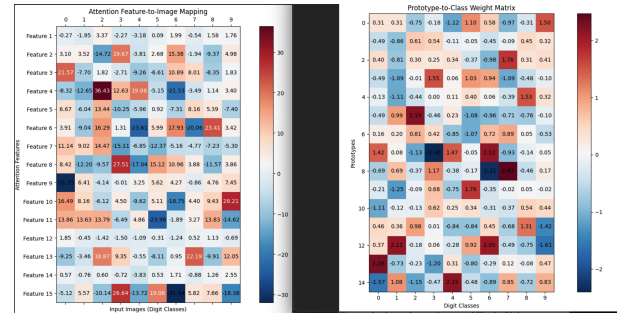


Fig. 3. Attention feature-to-image mapping and prototype-to-class weight matrix.

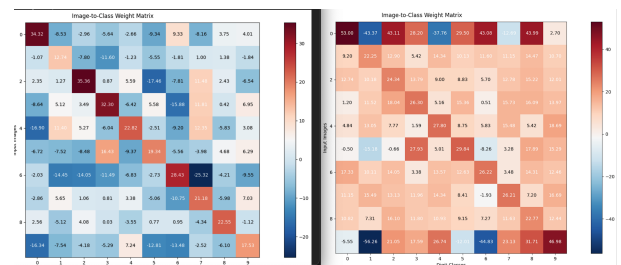


Fig. 4. Image-to-class weight matrix from the logits layer.

D. Conclusion

This work successfully integrates **prototypes** and **attention mechanisms** for interpretable MNIST classification. The

prototype-to-class weights provide fixed and interpretable visual representations aligned with specific digit classes, while the *attention-based mappings* dynamically focus on critical input features, improving the model's flexibility.

a) *Key Findings:*

- Attention mechanisms achieved a higher test accuracy of **99.64%**, demonstrating superior adaptability.
- Prototype-based learning achieved **96%** accuracy, providing strong interpretability through fixed representations.
- The combined approach balances performance and interpretability, achieving a validation accuracy of **99.22%**.

Future work will explore **diversity losses** to enhance prototype uniqueness and extend this approach to more complex datasets, such as CIFAR-10 or ImageNet.

REFERENCES

- [1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [2] B. W. Li, M. Rudolph, B. Kim, and R. Socher, "Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.