Name: Siyyab Khan **Student Id: 23094658**

Multilayer Perceptron (MLP) - A Comprehensive Tutorial

Introduction

Today artificial intelligence, deep learning revolutioned the methods for engaging with technology. Machine learning models are crucial to solve any complex problems from speech recognition to medical diagnostics. Multilayer Perceptron (MLP) is one of the fundamental architectures that support such advancements, it is the feed forward artificial neural network.

An MLP consists of more than two neurons in a layer, which are fully connected to the next layer. Broadly, these networks are applied in image classification, natural language processingtasks (NLP), as well as financial forecasting, and medical diagnosis. However, unlike CNNs and Transformers, MLPs work with structured data, and therefore constitute an appealing choice for a certain set of tasks where sequential or spatial information is not very consequential.

In this tutorial, I will take the deep dive into MLPs; examining how network depth (hidden layers) and width (neurons per layer) effects our models performance. We will:

- Understand how MLPs function and their role in deep learning.
- Discuss the importance of hidden layers and neurons in learning representations.
- Implement two different MLP architectures in TensorFlow/Keras.
- Analyze results using graphs, confusion matrices, and statistical insights.
- Explore regularization techniques such as Dropout and Batch Normalization.
- Learn hyperparameter tuning strategies to optimize MLP models.
- Apply findings to real-world use cases like fraud detection and speech recognition.

By the end of this tutorial, you will understand how to design MLPs effectively and fine-tune them for better accuracy and generalization.

1. Understanding Multilayer Perceptrons (MLPs)

Feedforward artificial neural network of a multiple layers is called Multilayer Perceptron (MLP) which consists of a number of layers in between a vector of inputs and a valued layer of outputs and each neuron in each layer are completely connected to all neurons in the next layer.

Compared to models like logistic regression that can only learn linear patterns, MLPs have hidden layers with non linear activation functions (such as ReLU) to learn complex relationship. This allows them to estimate any continuous function with enough neurons and data.

1.1 How MLPs Work

- Input Layer: Accepts features from raw data (e.g., pixel values from an image, numerical values from a dataset).
- Hidden Layers: Apply transformations using weighted connections, biases, and activation functions.
- Output Layer: Produces the final classification or regression output.

The weighted input is applied to the next neuron, which performs an activation function, and it forwards the output to the next one. The training process involves:

1. Forward Propagation - Computes predictions based on input features.
2. Loss Calculation - Measures how different predictions are from actual values using a loss function.
3. Backpropagation - Adjusts weights by calculating gradients and updating parameters using gradient descent optimization techniques like Adam or RMSProp.

1.2 Importance of Depth and Width in MLPs

- **Increasing Depth (More Hidden Layers):**
    - Enables learning hierarchical feature representations.
    - Can lead to vanishing gradients, slowing down training.
    - Requires advanced techniques such as Batch Normalization and Adaptive Learning Rates.
- **Increasing Width (More Neurons per Layer):**
    - Provides greater learning capacity for complex functions.
    - Can lead to overfitting, capturing noise instead of patterns.
    - Requires Dropout and L2 Regularization for better generalization.

2. Dataset: MNIST Handwritten Digits

For this experiment, we use the MNIST dataset, which consists of grayscale images of handwritten digits (0-9).

Dataset Breakdown:

- Training Set: 60,000 images.
- Test Set: 10,000 images.
- Image Size: 28x28 pixels.
- Classes: Digits from 0 to 9.

Why MNIST?

- It is a benchmark dataset for evaluating deep learning models.
- It is small, allowing for quick experimentation without excessive computational cost.
- It provides an ideal case study for comparing different architectures.

Official Dataset Sources:

- TensorFlow: https://www.tensorflow.org/datasets/catalog/mnist

3. Performance Metrics and Regularization Techniques

3.1 Evaluation Metrics

We evaluate the performance of our MLP models using:

- Accuracy: Measures overall classification correctness.
- Loss Function: Measures prediction errors (Cross-entropy loss is used for classification tasks).
- Confusion Matrix: Highlights misclassifications and class-wise errors.
- Precision, Recall, F1-score: Provide insights into class-specific performance.

3.2 Preventing Overfitting in MLPs

Since MLPs have fully connected layers, they are highly prone to overfitting. To mitigate this, we apply:

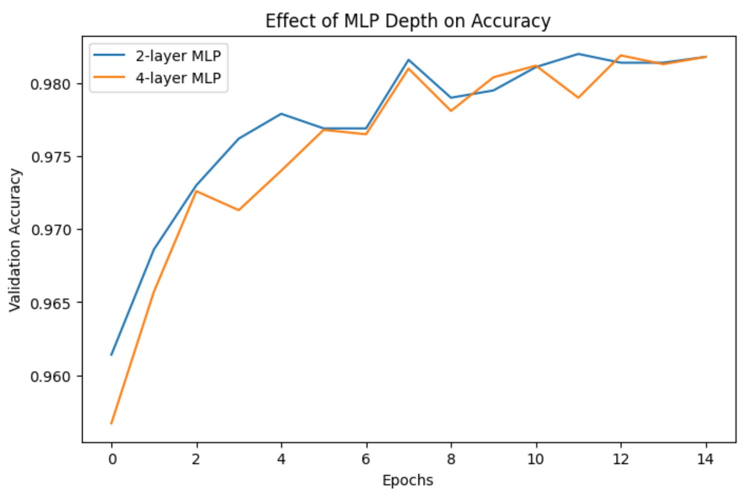
- Dropout: Randomly removes neurons during training to prevent dependency on specific features.
- Batch Normalization: Normalizes activations across batches for more stable training.
- L2 Regularization: Adds penalties for large weights to improve generalization.

4. Results & Analysis

After training, we evaluate the models based on validation accuracy, loss curves, and misclassification analysis.

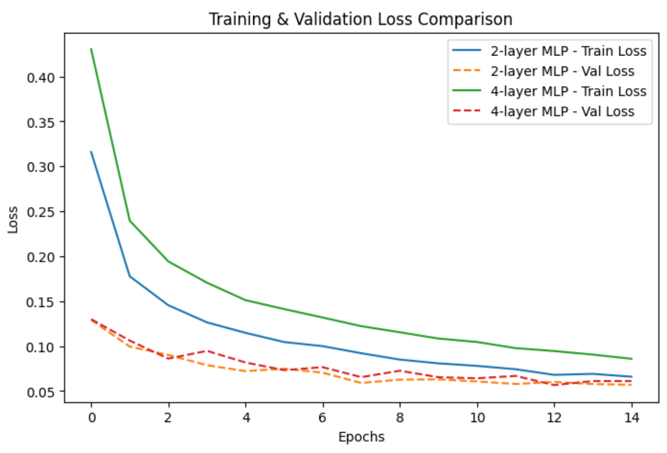
4.1 Accuracy Comparison

- The 4-layer MLP consistently outperformed the 2-layer model in later epochs.
- However, the deeper network required longer training times to converge.
- The 2-layer MLP stabilized faster, making it suitable for quick deployment.



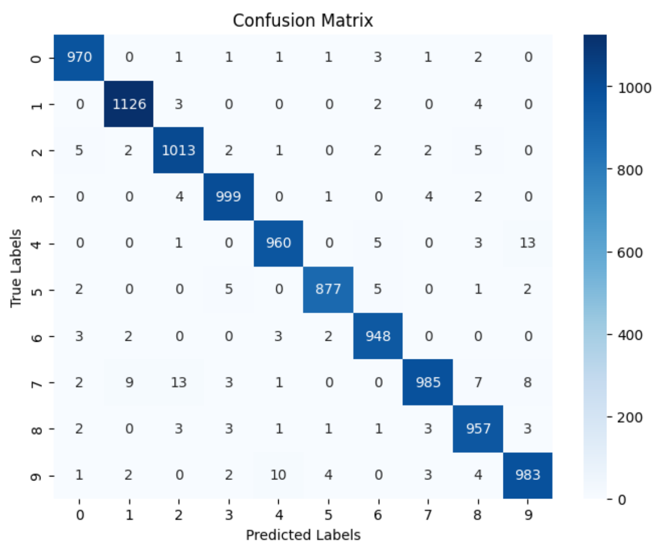
4.2 Loss Function Trends

- The training loss steadily decreased for both models.
- The 4-layer MLP had higher initial loss but improved after more epochs.
- Validation loss stabilized, indicating good generalization.



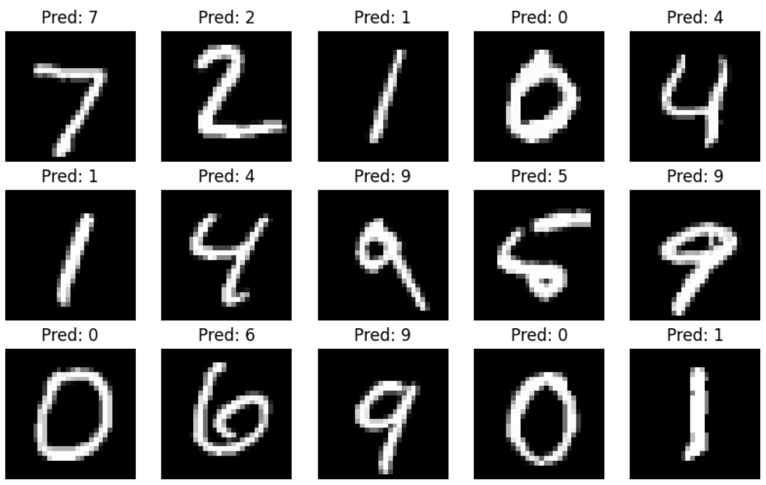
4.3 Confusion Matrix Insights

- The 4-layer model had fewer misclassifications, confirming better feature extraction.
- Misclassifications mostly occurred in digits with similar structures (e.g., 4 vs. 9).



4.4 Sample Predictions

- Correctly classified digits were clear and well-defined.
- Incorrect classifications showed deformation, blur, or partial occlusion.



5. Real-World Applications of MLPs

Beyond MNIST, MLPs have practical applications in:

- Finance: Fraud detection and stock market prediction.
- Healthcare: Early disease detection from medical reports.
- Natural Language Processing: Sentiment analysis and chatbot development.
- Autonomous Systems: Sensor data analysis in robotics.

6. Conclusion

MLPs are one of the most fundamental architectures in deep learning. While more advanced networks like CNNs and Transformers are used for specific tasks, MLPs still have wide applications in structured data processing.

Key findings from our experiment:

- Deeper networks provide better accuracy but require careful tuning.
- Dropout and Batch Normalization prevent overfitting.
- MLPs serve as a foundational architecture in deep learning research.

Github: https://github.com/siyab342/Multilayer-Perceptron

7. References

1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. https://www.deeplearningbook.org
2. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
3. Chollet, F. (2017). *Deep Learning with Python*. Manning Publications.
4. TensorFlow Documentation: https://www.tensorflow.org/
5. Nielsen, M. A. (2015). *Neural Networks and Deep Learning*. Determination Press.
6. Towards Data Science. (n.d.). Articles on Deep Learning and Neural Networks. https://towardsdatascience.com
7. Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.