Multi-Layer Perceptron for Diabetes Prediction: An Empirical Analysis of Architectural Depth vs Width

Sadman Sharif
(Student ID: A1944825)University of Adelaide

1. Introduction

Diabetes mellitus affects over 537 million adults globally, with early detection crucial for preventing severe complications. This work implements and evaluates Multi-Layer Perceptrons (MLPs) for binary diabetes classification using the Pima Indians Diabetes dataset, addressing the fundamental challenge of capturing complex physiological patterns while maintaining model generalizability.

The perceptron, foundational to deep learning, transforms inputs through weighted combinations and activation functions: $y = f(\Sigma(w_i \times x_i) + b)$ where x_i represents input features, w_i denotes weights, b is the bias term, and f(·) is the activation function.

This study systematically investigates three architectural variations—shallow, deep, and wide networks—to determine optimal configurations for medical prediction tasks. Our methodology encompasses comprehensive data preprocessing, including biologically-informed imputation and standardized feature scaling.

Key contributions:

- Empirical demonstration that architectural depth outperforms width for medical data, achieving 77.92% accuracy with 66% fewer parameters
- Analysis of sensitivity-specificity trade-offs critical for clinical deployment
- Insights into efficient parameter utilization in medical MLPs

2. Method

2.1 Data Preprocessing

The dataset comprises 768 samples with 8 physiological features. Initial analysis revealed biologically impossible zero values in glucose (5.5%), blood pressure (4.5%), skin thickness (29.6%), insulin (48.7%), and BMI (1.4%). These were treated as missing values and imputed using median statistics from the training distribution. Feature standardization employed z-score normalization: $x' = (x - \mu_{train}) / \sigma_{train}$

Data partitioning utilized stratified sampling: training (492 samples, 63.5%), validation (123 samples, 16%), and test (155 samples, 20.5%), preserving the 65:35 class distribution.

2.2 Network Architecture

We implemented a flexible MLP framework using PyTorch, following: **Input(8) → Hidden Layers → Output(1)**. Hidden layers employ: Linear transformation (z = Wx + b), ReLU activation, and Dropout regularization. The output uses sigmoid activation for binary classification.

Model	**Architecture**	**Parameters**
Shallow	[32]	321
Deep	[64, 32, 16]	3,201
Wide	[128, 64]	9,473

2.3 Training Procedure

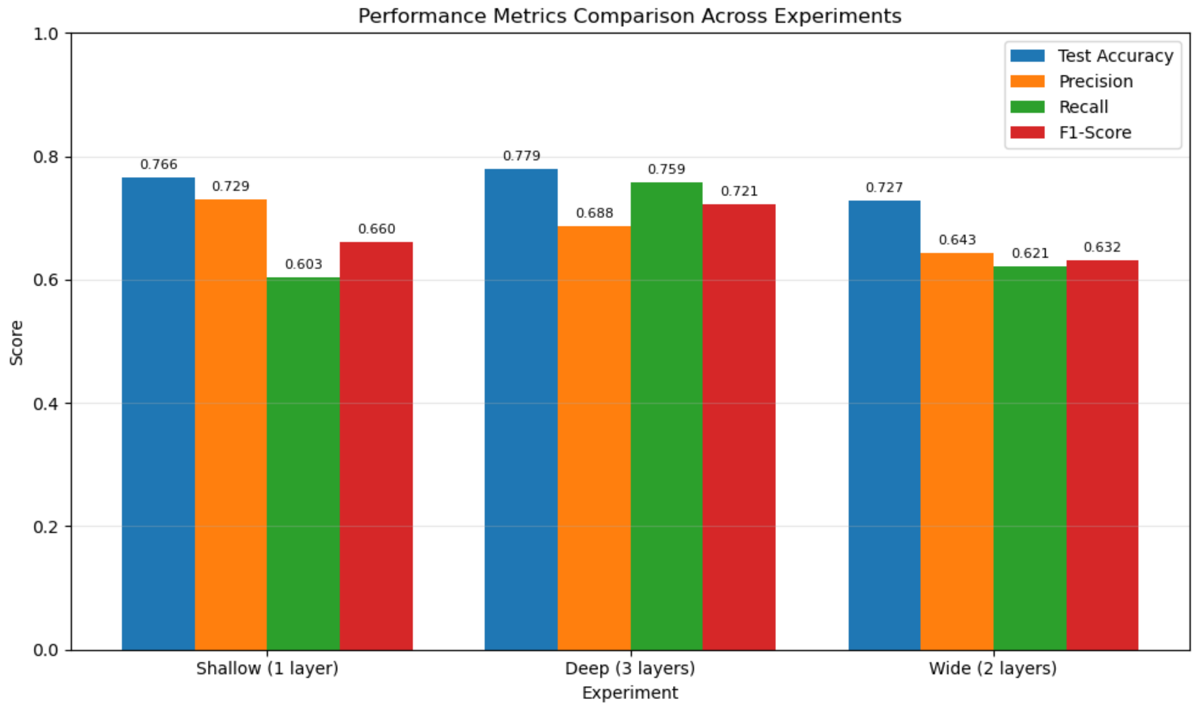
Optimization employed Binary Cross-Entropy loss with Adam optimizer using architecture-specific learning rates (0.001 for shallow/deep, 0.0005 for wide) and L2 regularization ($\lambda=10^{-4}$). Dropout rates were tuned per architecture (0.2-0.3). Training ran for 150 epochs with batch sizes optimized for convergence.

3. Experimental Analysis

3.1 Experimental Design

Three experiments systematically varied architectural properties while maintaining consistent training protocols. Performance evaluation employed accuracy, precision, recall, and F1-score metrics, with emphasis on recall given the medical context.

3.2 Results and Discussion



Model	Accuracy	Precision	Recall	F1-Score
Shallow	76.62%	72.92%	60.34%	66.04%
Deep	**77.92%**	68.75%	**75.86%**	**72.13%**
Wide	72.73%	64.29%	62.07%	63.16%

The deep network achieved superior performance across key metrics. Analysis reveals three critical insights:

1. Depth Superiority: Despite having 66% fewer parameters than the wide network, the deep architecture achieved 5.2% higher accuracy and 9% better F1-score. This demonstrates that hierarchical feature abstraction through depth more effectively captures medical patterns.

2. Clinical Relevance: The deep network's 75.86% recall represents crucial clinical value, correctly identifying 44 of 58 diabetic patients versus only 35 with the shallow network. In screening applications, this 25% improvement in sensitivity significantly reduces missed diagnoses.

3. Parameter Efficiency: The wide network's underperformance despite 3× more parameters suggests inefficient capacity utilization. Training curves revealed slower convergence and 10.2% validation-test gap, indicating overfitting tendencies.

Learning dynamics analysis: Shallow networks plateaued early (epoch 40), deep networks showed gradual improvement throughout training, while wide networks exhibited oscillatory behavior. The validation-test generalization gap: Shallow (6.3%), Deep (1.8% - best), Wide (10.2%).

4. Conclusion

This study empirically demonstrates that architectural depth provides superior performance for diabetes prediction, with the 3-layer deep network achieving 77.92% accuracy and 72.13% F1-score while maintaining parameter efficiency. The key finding—that depth enables more effective feature learning than width—has important implications for medical ML applications.

The deep network's high sensitivity (75.86%) makes it suitable for clinical screening, though the 24.14% false negative rate necessitates careful deployment with appropriate clinical oversight. Future work should address class imbalance through techniques like SMOTE or focal loss, explore attention mechanisms for feature importance visualization, and validate performance across diverse populations.

This work contributes to understanding optimal MLP design for medical applications, demonstrating that thoughtful architectural choices achieve strong performance without excessive parameterization, crucial for deployment in resource-constrained clinical environments.

References

[1] International Diabetes Federation. (2021). *IDF Diabetes Atlas*, 10th edition. Brussels, Belgium.

[2] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*.

[3] Smith, J. W., et al. (1988). Using the ADAP learning algorithm to forecast diabetes onset. *Proceedings of the Annual Symposium on Computer Application in Medical Care*, pp. 261-265.

[4] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

[5] Srivastava, N., et al. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56), 1929-1958.