Empirical Analysis of Hidden-Layer Configurations in Feedforward Neural Networks for Handwritten Digit Classification

Michael Wen
VP, CIO, CISO
Jet Optoelectronics
wentaihao@yahoo.com

This paper presents an extensive empirical examination of hidden-layer width, depth, and optimization dynamics in feedforward neural networks trained on the Kaggle Digit Recognizer dataset. Inspired by the experimental rigor commonly found in landmark works such as *Attention Is All You Need*, this study evaluates over one hundred architectural configurations across single- and dual-hidden-layer networks. Results reveal that network width plays a substantially more significant role than depth for this task, with an optimal single-layer configuration achieving over 94% training accuracy and 92% validation accuracy. These findings emphasize the importance of empirical architecture tuning and illustrate that deeper networks are not universally advantageous, even in domains where neural methods are dominant.

Introduction

Neural network architecture design remains a central challenge in applied machine learning research. Although deep learning has revolutionized computer vision, natural language processing, and speech recognition, the relationship between layer depth, layer width, and empirical performance is highly task-dependent. Motivated by theoretical curiosity and practical deployment considerations, this study conducts a structured investigation into the performance characteristics of shallow and moderately deep feedforward networks trained on a widely studied handwritten digit classification benchmark.

Background

The Kaggle Digit Recognizer dataset, derived from MNIST, consists of grayscale 28×28 pixel images of handwritten digits. Feedforward neural networks have long been considered a baseline method for this dataset. Modern literature highlights several themes relevant to this work: (1) deeper networks can express more complex functions but may suffer from vanishing gradients; (2) wider networks can approximate functions with fewer layers but require careful regularization; and (3) the effectiveness of ReLU and softmax activations has been widely validated across classification tasks.

Experimental Methodology

All networks were implemented in Python using NumPy-based forward and backward propagation routines. The dataset was split into a training set of 41,000 samples and a validation set of 1,000 samples. Input layers contained 784 neurons (28×28 pixels), and output layers had 10 neurons (one per digit). ReLU was used for hidden layers and softmax for the output layer. The learning rate α was fixed at 0.1 for all experiments.

We conducted two main sets of experiments: single-hidden-layer networks and two-hidden-layer networks. Each experiment varied the number of neurons and training iterations, with three trials per configuration to capture variance.

Single-Hidden-Layer Experiments

The first set of experiments explored single-hidden-layer architectures with varying neurons and iterations. Table 1 summarizes key results.

Single-hidden-layer network results. Range indicates three trials.

Neurons	Iterations	Training Accuracy	Validation Accuracy
10	1000	0.884–0.882	0.876–0.89
20	1000	0.896–0.901	0.896–0.898
50	1000	0.913–0.916	0.896–0.915
392	100	0.809–0.832	0.813–0.869
392	200	0.873–0.875	0.844–0.865
392	1000	0.938–0.942	0.907–0.927

Two-Hidden-Layer Experiments

Two-hidden-layer networks were evaluated with symmetric (equal neurons per layer), asymmetric, and reduced neuron configurations. Table 2 shows representative results.

Two-hidden-layer network results including configurations where hidden layers have fewer neurons than output. Reduced neurons lead to poor performance.

Layer 1	Layer 2	Iterations	Validation Accuracy Range
10	10	500	0.828–0.841
20	20	500	0.860–0.868
50	50	500	0.890–0.900
10	20	500	0.820–0.859
10	50	500	0.844–0.865
20	10	500	0.847–0.873
50	10	500	0.845–0.885

Layer 1	Layer 2	Iterations	Validation Accuracy Range
100	10	500	0.875–0.890
100	100	500	0.891–0.906
8	8	500	0.789–0.826
5	5	500	0.477–0.745
8	20	500	0.789–0.824
20	8	500	0.781–0.844

Observations and Analysis

- One hidden layer is sufficient for this classification task, provided neuron count is appropriate.
- Accuracy improves with increased iterations; low-iteration trials show high variance.
- Two hidden layers do not outperform a well-tuned single hidden layer.
- Symmetric two-layer architectures generally outperform asymmetric ones.
- Hidden layers with fewer neurons than the output layer severely degrade performance.

Conclusion

The results of this study show that increased depth does not inherently improve performance for structured, low-dimensional data. Instead, a single well-chosen hidden-layer width—specifically 392 neurons—offers the best balance of accuracy, stability, and computational efficiency. Reduced hidden-layer neurons drastically decrease performance. These findings reinforce the principle that architectural selection must be adapted to the problem domain rather than guided solely by trends in unrelated fields.

References

1. Vaswani, A., et al. “Attention Is All You Need.” Advances in Neural Information Processing Systems, 2017.
2. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. “Gradient-Based Learning Applied to Document Recognition.” Proceedings of the IEEE, 1998.
3. Goodfellow, I., Bengio, Y., & Courville, A. *Deep Learning*. MIT Press, 2016.
4. Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.