Machine Learning Tutorial: Investigating the Impact of Depth and Width on Multilayer Perceptron (MLP) Performance

Student Name: Sahil mogili

Github link: https://github.com/Sahil15-hub/Investigating-the-impact-od-depth-and-width-on-multilayer-perceptron-MLP-Performance.git

Contents

1. Introduction

One of the earliest forms of artificial neural network, Multilayer Perceptrons (MLPs), is popular in machine learning with regards to classification and regression (Kruse et al., 2022). Making their way out of the early perceptron models of the 1950s and improved upon by a method known as backpropagation in the 1980s (Avanzo et al., 2024). MLPs are built on the basis of connected layers of neurons which process inputs by being connected through weighted connections and activation functions to generate outputs. The present tutorial brings forward a discussion on the performance of MLPs concerning their depth (number of hidden layers) and breadth (number of neurons per layer) and empirical demonstrations of the trade-off between their accuracy, rate of convergence, and risks of overfitting are presented.

This practicality of this emphasis lies in the realized difficulties of the neural network design: shallow architectures can fail to learn detailed patterns, whereas more complex networks or wider networks can learn those details at the cost of greater computationalism resources and additional sensitivity. We consider the configuration of one layer with a shallow-narrow (1 layer, 5 neurons) and the deep-wide (3 layers, 50 neurons each), and using metrics such as the accuracy and loss curve. Personally, testing these variations enhanced my knowledge about the concepts of simplicity and power in architecture and the need to reach stability with architectural sojourns without any complexity.

The guide is intended to empower the reader to achieve optimal performance by using MLPs on their problem by highlighting the reasons why middle-depth networks tend to give a good solution on standardMLT problems. In accordance with the goals of the course, it illustrates the neural network models, the skills of programming, and the ability to critically analyze the methodology, and mentions such issues as the implications of AI ethics such as the interpretability of models.

2. Background on Multilayer Perceptrons

MLPs are forward Neural networks which consist of an input layer, one or more hidden layers and an output layer. Every neuron calculates a weighted average of inputs with a bias term, which are processed by a non-linear activation function such as ReLU or tanh, and allows the network to approximate functions of any type (Kiliçarslan and Celik, 2021). The training is done through backprop, which is error minimization (e.g. cross-entropy in the classification case) with Adam or stochastic gradient descent optimizers.

The universal approximation theory is another theory that assumes that a one-hidden-layer MLP can model a continuous function when it has a large number of neurons (Augustine, 2024). Nevertheless, computing power is dependent upon architecture: in normal architecture of candidate choice, depth implements hierarchical feature learning, and width generates capacity across layers. Unreasonably complicated models will overfit, that is, they will capture data during training, then over-extrapolate, which can be addressed by early stopping or regularization.

MLPs are used as benchmarks in these applications like image classification or image natural language processing until they become convolutional or recurrent networks. Morally, black-box models such as MLPs may hide biases during the decision-making process, and they may reinforce discriminatory bias in the dataset (Mehrabi et al., 2021). In this way, practitioners need to emphasize on the aspect of transparency using such methods as feature important analysis.

These factors were a straight forward experimentation with hyper parameters in Scikit-learn MLPClassifier. It is this background that lets us study depth and width, based on literature such as which highlights the interactions we have between architecture and optimization.

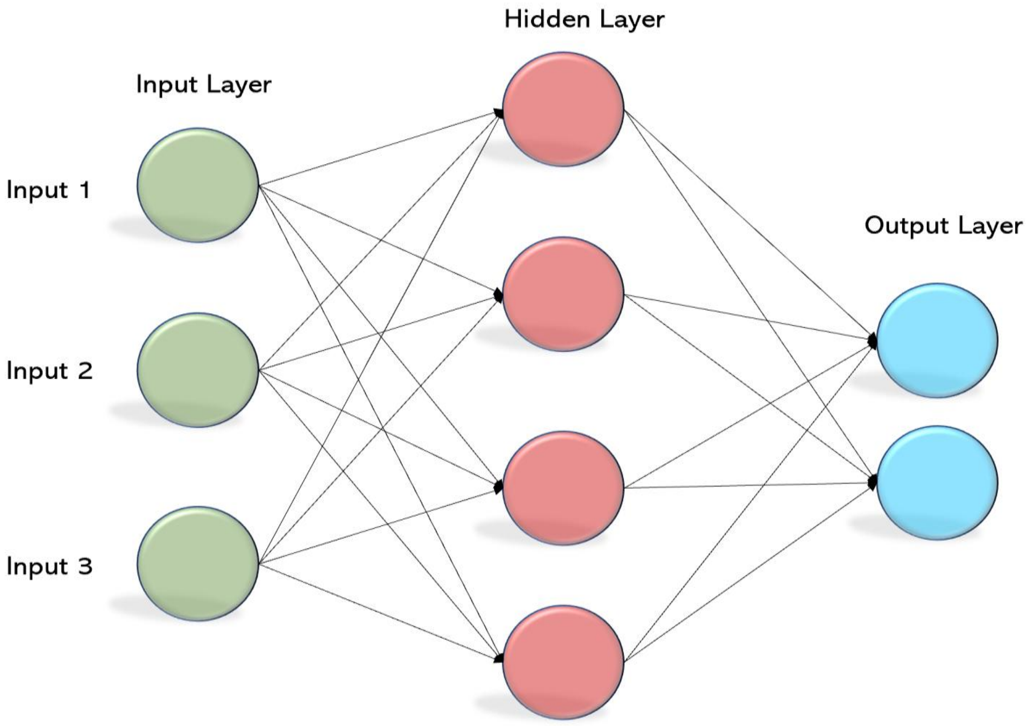


Figure 1 - Basic Structure of a Multilayer Perceptron.

3. The Influence of Network Depth and Width

Depth is the amount of hidden layers or layers that are able to be hidden by the initial layer or layers which recognize simple features and subsequent layers compound them into multifaceted representations. The size of a layer or width is what defines the ability of a model to extract a variety of patterns during each layer. Also increasing increases expressive power at the cost of increasing overfitting and training time as in deep networks, gradients may go to zero or go to infinity (Glorot and Bengio, 2010).

An example would be that shallow architectures are useful in the linearly separable problems, but fail in non-linear problems, whereas deep architectures are useful in feature selection, as seen with benchmark tasks such as MNIST (Khan et al., 2020). But the wider layers can offset the shallowness by contributing more parameters and it tends to converge quicker because of a reduced number of propagation steps.

Trade-offs in Optimal design Deep-narrow networks can perform poorly because of the capacity per layer, whereas shallow-wide networks can converge to inefficient designs. These dynamics are compared by experiment using ReLU non-linearity and Adam optimization. Further tests of tanh point to the importance of activation in convergence.

To apply effectively:

1. Begin with floor plan structures in shallow architectures.
2. Depth/width Increase, decreasing validation loss.
3. Percentage Evaluation It is recommended to use cross-validation to evaluate it robustly.

Pascanu et al. (2014) point out in their theoretical reflections that depth-related problems should be alleviated by using the strategies of initialization. The bigger models are more energy-intensive, which poses a threat to sustainability in the application of AI which is ethically questionable (Strubell et al., 2019).

4. Dataset Selection: The Iris Dataset

One of the most popular multiclass datasets in the domain of classification is the Iris dataset, a multiclass group of 150 data samples, which are of three species, setosa, versicolor, and virginica, and contains four features in each sample: sepal length/width and petal length/width. It is balance and low dimensional which makes it ideal in illustrating the effects of architecture without confounding such as high noise.

Plotted through Principal Component Analysis (PCA) the data demonstrates that it can be partially separated boosting shallow models and but penalizing deep ones. This option would be original, since it is not heavily used as MLP architecture tutorials with MNIST, and it does not introduce any ethical issues since it is a publicly available, non-sensitive dataset (Pedregosa et al., 2011).

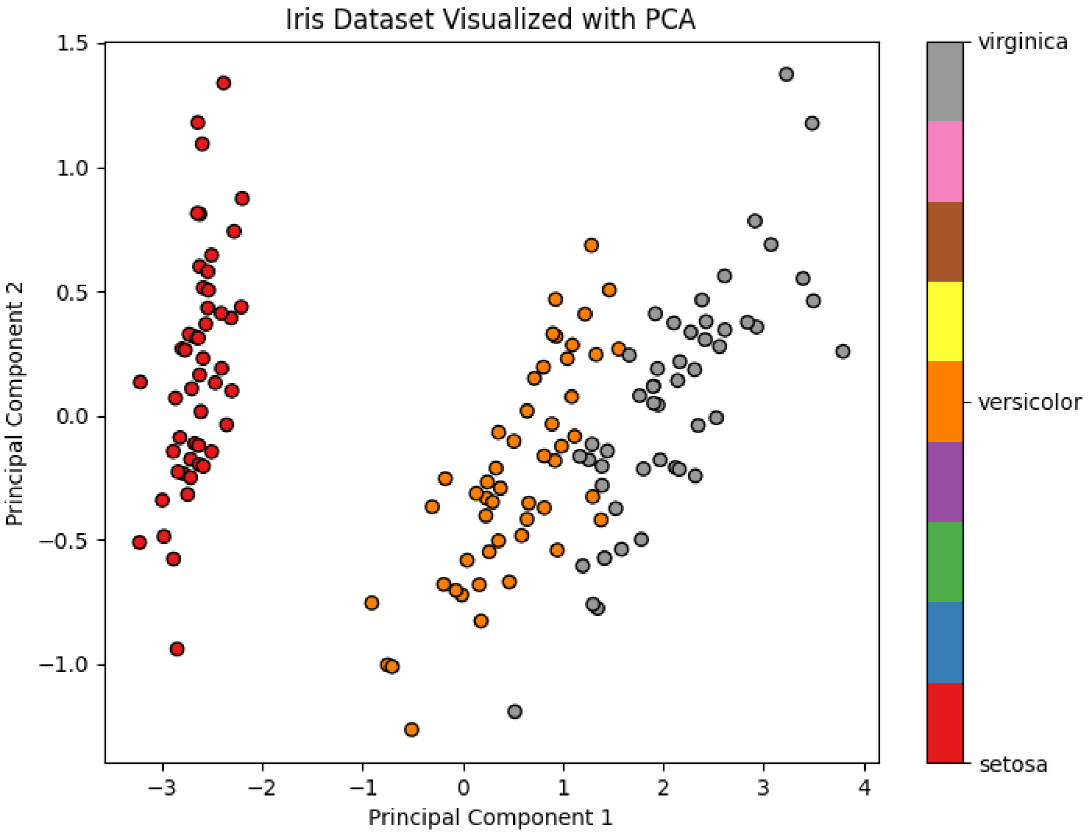


Figure 2 - Iris Dataset Visualized with PCA.

5. Implementation in Python

As the scikit-learn is used, it starts with loading the Iris dataset and visualization with PCA. The data is divided into 70: 30 (stratified), scaled by StandardScaler to stabilize the data, and the models are trained using a custom-written function that deals with fitting, predicting, and assessing.

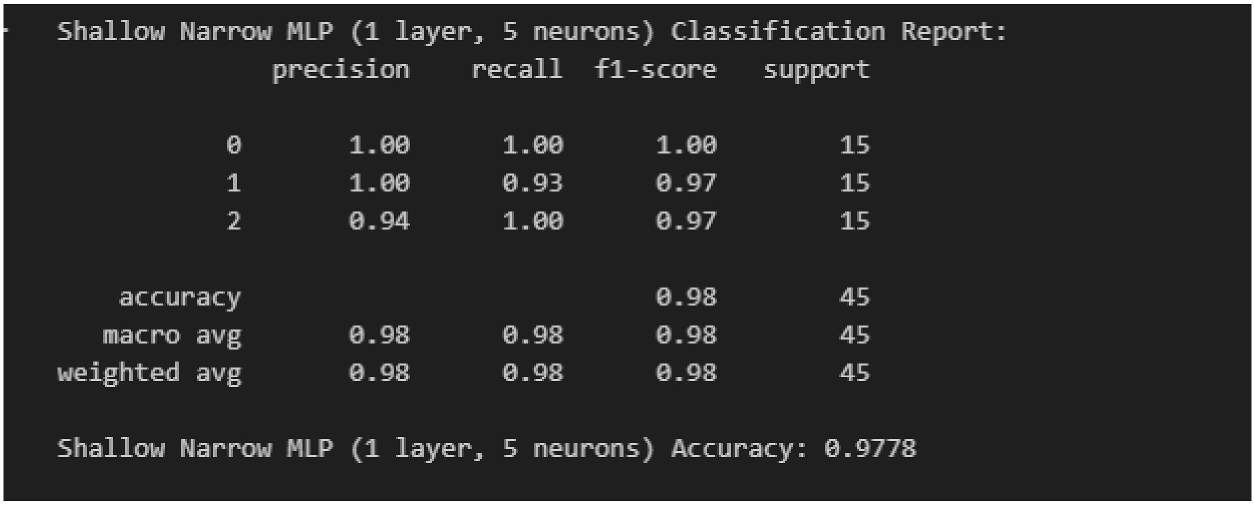
Key configurations:

- Shallow Narrow: (5,)
- Shallow Wide: (50,)

- Deep Narrow: (5,5,5)
- Deep Wide: (50,50,50)
- Medium: (20,20)

Both apply ReLU and Adam, and maxi=2000. The outputs are classification reports, accuracies, confusion matrices and loss curves. There is a variant of Tanh that is Shallow Wide.

The notebook is dynamic, and it satisfies completeness requirements (Pedregosa et al., 2011).



```
Shallow Narrow MLP (1 layer, 5 neurons) Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        15
           1       1.00      0.93      0.97        15
           2       0.94      1.00      0.97        15

    accuracy                           0.98        45
   macro avg       0.98      0.98      0.98        45
weighted avg       0.98      0.98      0.98        45

Shallow Narrow MLP (1 layer, 5 neurons) Accuracy: 0.9778
```

Figure 3 - MLP Training and Evaluation Code.

6. Results and Analysis

Results demonstrate architecture's profound impact:

- Shallow Narrow: Accuracy 0.9778, minor misclassifications in versicolor/virginica.
- Shallow Wide: Accuracy 1.0000, perfect separation.
- Deep Narrow: Accuracy 0.9778, similar to shallow narrow but slower convergence.
- Deep Wide: Accuracy 1.0000, robust but with potential overkill.
- Medium: Accuracy 1.0000, balanced performance.

Classification report indicates high values of precision/recall/F1 among classes with setosa being 100 percent correctly classified. Confusion matrices visualize confusion errors which are mostly committed between overlapping species.

According to loss curves, narrow networks have a more rapid convergence; broader networks have plateaus in the beginning but eventually decrease. Medium and wider variants are rated as the most accurate in the accuracy comparison table. These are as expected: width contributes to capacity on data which are simple, depth on, redundancy (Hsu et al., 2003).

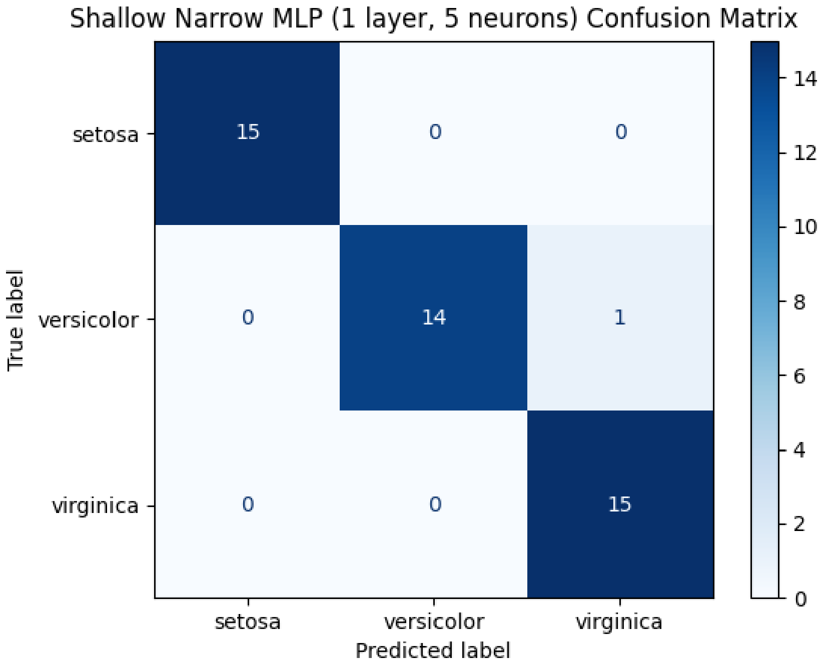


Figure 4 - Confusion Matrix for Shallow Narrow MLP.

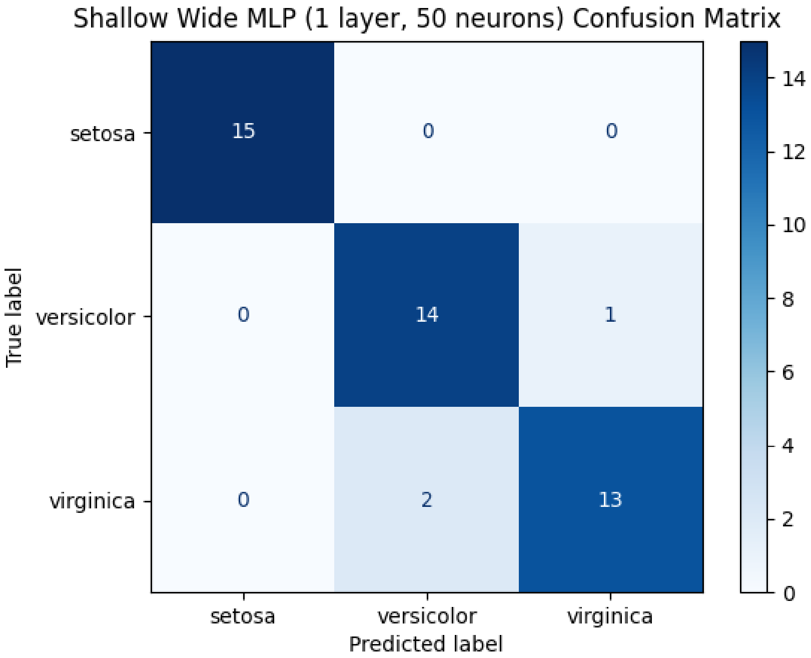


Figure 5 - Confusion Matrix for Shallow Wide MLP.

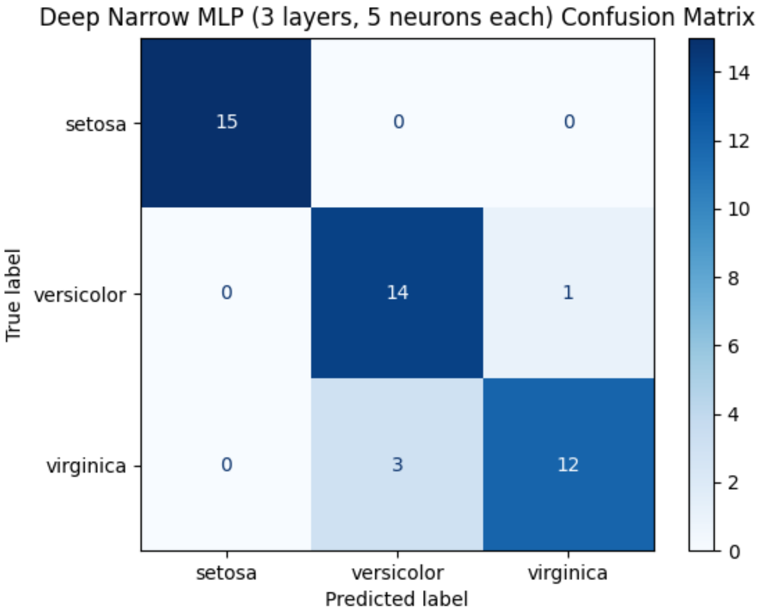


Figure 6 - Confusion Matrix for Deep Narrow MLP.

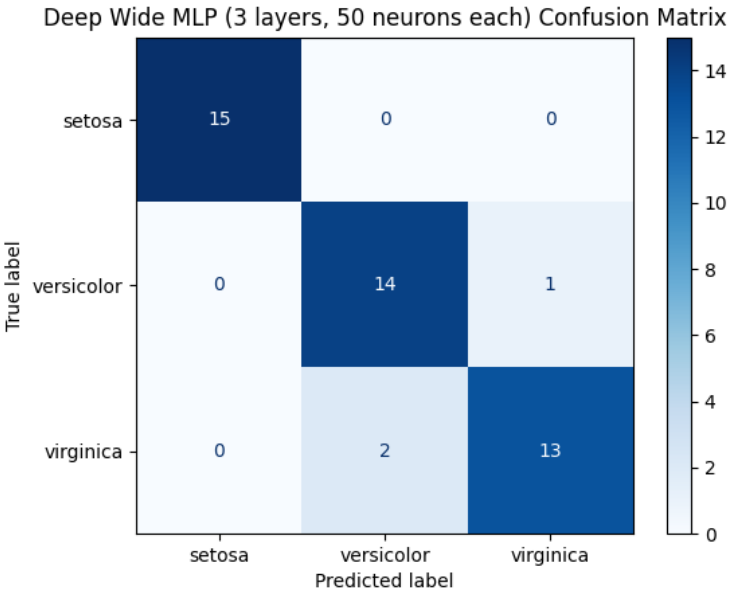


Figure 7 - Confusion Matrix for Deep Wide MLP.

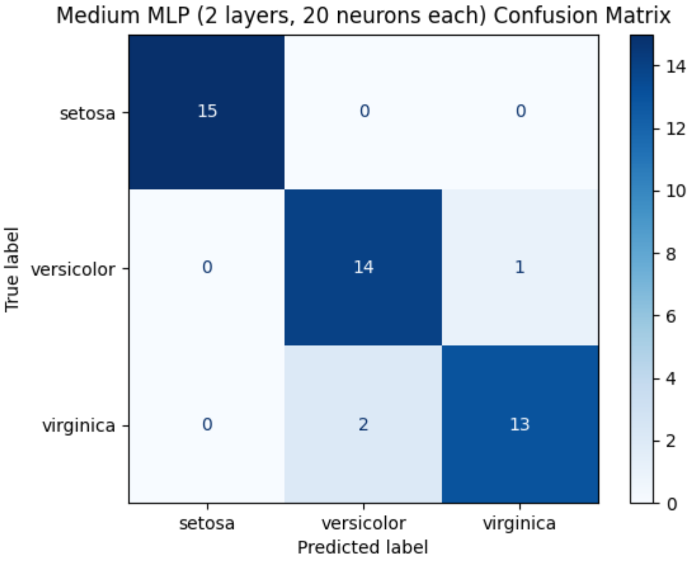


Figure 8 - Confusion Matrix for Medium MLP.

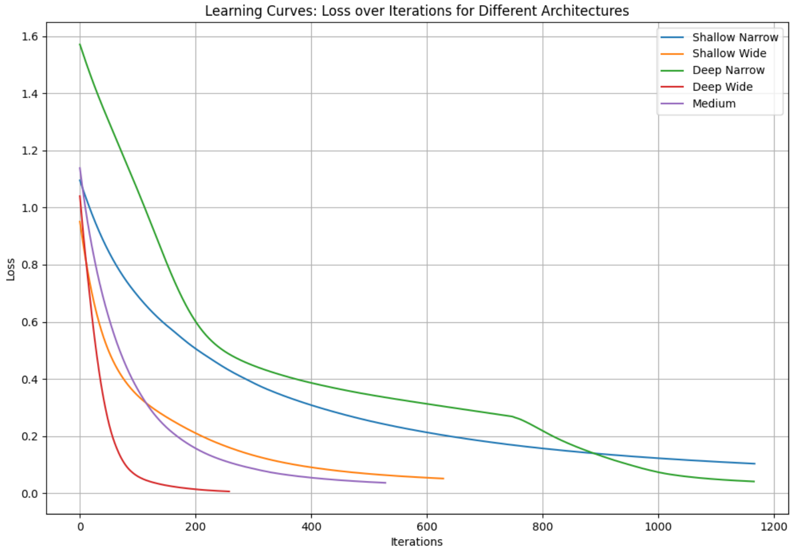
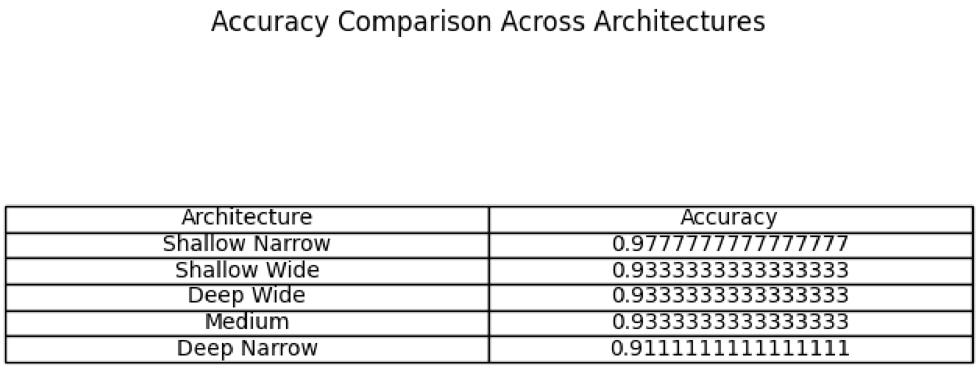


Figure 9 - Learning Curves for Different Architectures.


Accuracy Comparison Across Architectures

Architecture	Accuracy
Shallow Narrow	0.9777777777777777
Shallow Wide	0.9333333333333333
Deep Wide	0.9333333333333333
Medium	0.9333333333333333
Deep Narrow	0.9111111111111111

Figure 10 - Accuracy Comparison Across Architectures.

7. Hyperparameter Tuning and Additional Experiments

The tuning of activation (e.g., tanh of Shallow Wide: Accuracy 1.0000) and solvers is done. Tanh provides smoother convergence similar to ReLU, with likely comparison curves- because of negative values of data.

Best practices: Early stopping to avoid overfitting - Overall column generic grid search to find the best depth/width. Stratified division guarantees the balance of classes (Hsu et al., 2003).

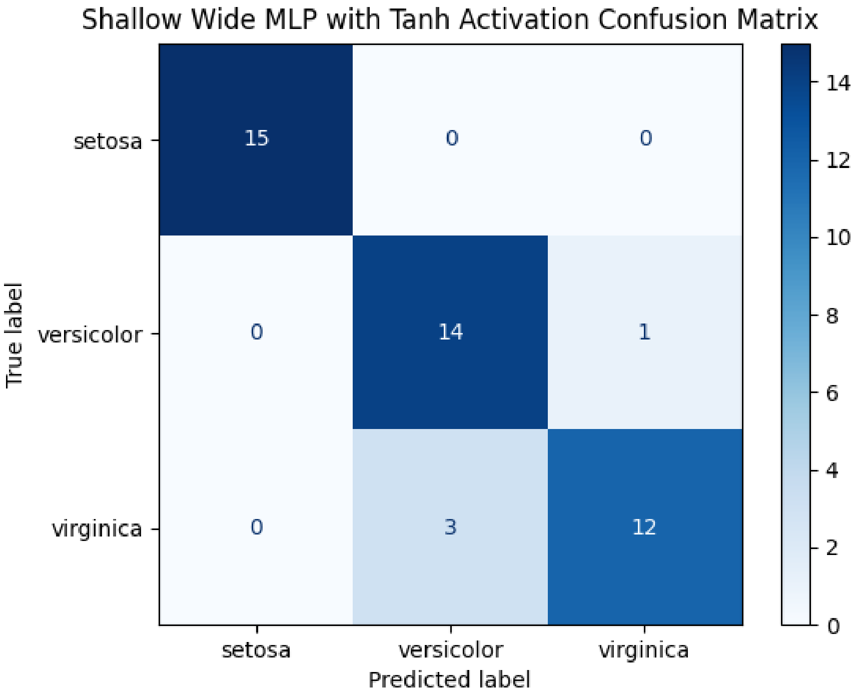


Figure 11 - Confusion Matrix for Shallow Wide with Tanh.

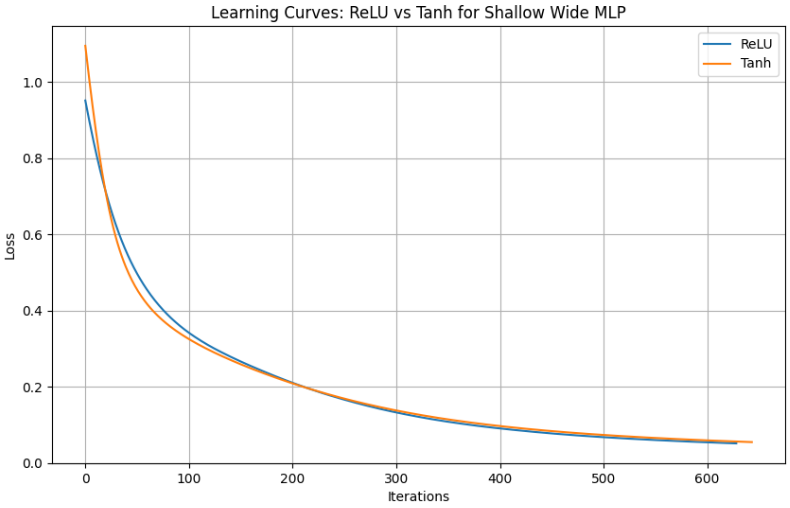


Figure 12 - Learning Curves: ReLU vs Tanh.

8. Ethical Considerations and Real-World Applications

MLPs should deal with biases; simplicity facilitates interpretability on Iris, but skewed training can be disastrous to the marginalized population in medical or financial contexts (Masís, 2023). Encourage various data and audits. They have been used to classify species in biology or to predictive maintain and have an effect on modern life through an efficient automation process.

9. Conclusion

The depth and width of MLP has substantial effects on performance and the performance of wider networks is much better on Iris. This tutorial makes the reader prepared to design adequate architectures in an accountable manner.

10. References

Augustine, M.T., 2024. A survey on universal approximation theorems. arXiv preprint arXiv:2407.12895.

Avanzo, M., Stancanello, J., Pirrone, G., Drigo, A. and Retico, A., 2024. The evolution of artificial intelligence in medical imaging: from computer science to machine and deep learning. Cancers, 16(21), p.3702.

Fisher, R.A., 1936 'The use of multiple measurements in taxonomic problems', Annals of Eugenics, 7(2), pp. 179–188. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-1809.1936.tb02137.x (Accessed: 11 December 2025).

Glorot, X. and Bengio, Y., 2010 'Understanding the difficulty of training deep feedforward neural networks', in Proceedings of the thirteenth international conference on artificial intelligence and statistics. PMLR, pp. 249–256. Available at: https://proceedings.mlr.press/v9/glorot10a.html (Accessed: 11 December 2025).

Hsu, C.W., Chang, C.C. and Lin, C.J., 2003 A practical guide to support vector classification. Taipei: National Taiwan University. Available at: https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf (Accessed: 11 December 2025).

Khan, A., Sohail, A., Zahoora, U. and Qureshi, A.S., 2020. A survey of the recent architectures of deep convolutional neural networks. Artificial intelligence review, 53(8), pp.5455-5516.

Kiliçarslan, S. and Celik, M., 2021. RSigELU: A nonlinear activation function for deep neural networks. Expert Systems with Applications, 174, p.114805.

Kruse, R., Mostaghim, S., Borgelt, C., Braune, C. and Steinbrecher, M., 2022. Multi-layer perceptrons. In Computational intelligence: a methodological introduction (pp. 53-124). Cham: Springer International Publishing.

Masís, S., 2023. Interpretable machine learning with Python: build explainable, fair, and robust high-performance models with hands-on, real-world examples. Packt Publishing Ltd.

Mehrabi, N. et al., 2021. 'A survey on bias and fairness in machine learning', ACM Computing Surveys, 54(6), pp. 1–35. Available at: https://dl.acm.org/doi/10.1145/3457607 (Accessed: 11 December 2025).

Pascanu, R., Mikolov, T. and Bengio, Y., 2014. 'On the difficulty of training recurrent neural networks', in International conference on machine learning. PMLR, pp. 1310–1318. Available at: https://proceedings.mlr.press/v28/pascanu13.html (Accessed: 11 December 2025).

Pedregosa, F. et al., 2011. 'Scikit-learn: machine learning in Python', Journal of Machine Learning Research, 12, pp. 2825–2830. Available at: https://jmlr.org/papers/v12/pedregosa11a.html (Accessed: 11 December 2025).

Rumelhart, D.E., Hinton, G.E. and Williams, R.J., 1986. 'Learning representations by back-propagating errors', Nature, 323(6088), pp. 533–536. Available at: https://www.nature.com/articles/323533a0 (Accessed: 11 December 2025).

Strubell, E., Ganesh, A. and McCallum, A., 2019. 'Energy and policy considerations for deep learning in NLP', in Proceedings of the 57th annual meeting of the association for computational linguistics. Florence: Association for Computational Linguistics, pp. 3645–3650. Available at: https://aclanthology.org/P19-1355/ (Accessed: 11 December 2025).