Investigating the Effects of Depth and Width on MLP Performance

GitHub Repository Link

- https://github.com/hafizabdulhaseeb/Exploring-the-Role-of-Depth-and-Width-on-the-Performance-of-MLP/tree/main

1. Introduction

This project aims to investigate the impact of network architecture on the performance of Multi-Layer Perceptron's (MLPs), focusing on two key aspects depth, the number of the hidden layers and width, the number of neurons in each layer. It is these architectural parameters which go a long way to define the capacity of the model to learn structure within the data set under analysis. For two reasons this interdependence is vital for the design and optimization of the parameters of neural networks:

This is the reason why the study aims at identifying and analyzing ways in which depth and widths affect the actual generality of data that is not modelled. It is generally understood that more layers make a network capable of learning more complex representations in theory but more layers also increase the risk of over-fitting especially when the training data is scarce. Likewise, increasing values for the width of each layer can improve the representational power of the model but may reach the point of almost no return and worst the model performance beyond that point. This project is an attempt to systematically advance through configurations which are most suitable in terms of struck the right balance between model complexity and performance.

The experiments are performed on a simulated classification data set created using the make classification function of scikit-learn toolbox. This dataset contains 1000 instances with 20 numerical attributes and each instance associated with a single label from two classes. This sort of setup generates a controlled situation since considering depth and width is easier since the structure and complexity of datasets are workable. The used binary classification task provides a reference for evaluating the model in terms of its ability to classify the two classes.

Moreover, this project also discusses some possible issues: too large depth may lead to overfitting and too small depth as well as too small width may cause under fitting. These findings are useful not only for the dataset employed in the present research but also for understanding how to develop MLPs in other contexts.

2. Methodology

2.1. Dataset Preparation

Random synthetic datasets for the current project were generated from the 'make_classification' function – a function meant for classification problems specifically. This makes it easy to specify all the characteristics of the sampling space which facilitates the construction of a good dataset suitable for testing and comparing the performance of MLPs. The key steps in dataset preparation were:

2.2 Dataset Generation:

- The training set was created from scratch and was equal to 1000 samples and contained 20 numerical variables.
- Having two classes of data set made this a binary classification problem where each sample was assigned to one of the two classes.
- Since it is more realistic, the dataset was then designed in such a way to make it contain informative features, redundant features, and noise.

2.3 Data Splitting:

It was separated into training set where 80 percent of the data was used and testing set with 20 percent.

- The first stage involved imported the train data set into MLP models to train them.
- The testing data was used solely for the model generalization evaluation.

2.4 Features and Target:

- The features (X) are the variables on which the model seeks to learn in order to give its prediction. From these 20 numerical variables, the function that identifies between the two classes contains this important information.
- The target variable (y) has the meaning of the class to which the sample belongs. This binary output is used to control the model during training process.

3. Networking Configurations of the Network Structures

The paper analyses the MLP for signs of improvement and decay while modifying different layers to determine the impact on network performance. Two principal parameters of architecture were varied systematically: depth, or the number of hidden layers, and width, or the number of neurons in each layer. These parameters affects the learning phase, generalize ability of the network and it is more prone to over fit or under fit condition.

3.1 Depth:

The complexity of an MLP is defined by the number of layers present in the network excluding the input and output layers. It also enables the model to contain hidden layers where it can input hierarchical and difficult representations of the information proposed.

Range of Configurations:

- The number of specialized layers varied from 1 to 5.
- These configurations included flat networks (1 layer), and relatively complex networks with up to 5 layers.

Effect of Depth:

Shallow Networks (1–2 layers):

- Good for learning conceptually less complex patterns or linearity of what can be observed in the data.
- May sometimes fail to handle highly non-linear or complex data sets because of the insufficient amount of resources available for use.

Moderately Deep Networks (3–5 layers):

- better equipped to model non-linear and hierarchical relationships.
- Enable the model to learn more nuanced patterns in the data.
- However, deeper networks increase computational cost and the risk of overfitting, especially with small datasets.

3.2 Width:

The width of an MLP can be determined as the quantity of neurons, which is present in each hidden layer. The neuron therefore acts as a computing cell which receives the input data and participates in the learning phase.

Range of Configurations:

- The number of neurons per layer was varied to 5, 10, 20, 50, 100 neurons.
- This range has includes the narrow configurations with limited capacity and the wide configurations which are able to capture more comprehensive relationships.

Effect of Width:

Narrow Networks (5–20 neurons):

- May cause under fitting owing to the fact that it has a low learning capability.
- First, he defined linear models which are suitable for simple data sets, and fewer parameters are needed.

Wide Networks (50–100 neurons):

- Help give more capacity to learn more quadratic and non-linear relationships in the data given.
- But they are sensitive to overfitting in the case where the data set is small or not correctly normalized.

4. Model Variants:

This approach allowed the study to systematically build a total of twenty-five different depth and width combinations of network architectures.

4.1 Configurations:

- Five of the herein described pre-defined width values were used to sweep over the layer count for each setting of depth which ranged from 1 to 5 hidden layers.
- This approach was used to guarantee that all combinations of depth and width were investigated providing information on both aspects.

4.2 Purpose of Variants:

- Such variety of configurations enabled the thorough investigation of the key aspects of compromise strategies, optimization of depth and width.
- For example:
    - A network with few layers, with large numbers of nodes in each of those layers, might prove to be efficient or functional in a different way than a network with many layers that connect only a few nodes each.
    - Thus, the depth and width factors have interaction scenarios that offer understanding of how network capacity can be provisioned to work optimally.

5. Conclusion

In this project, the following work was done on the classification tasks analyzing how the architectural aspects of Multi-Layer Perceptrons (MLPs) affect their performance. As the depth and the width of the networks were altered systematically the authors intended to investigate the consequences of the identified design choices on the learning and generalization capabilities of the benchmarked classifier. With clear non-linear patterns both in the raw data samples and between the classes it was an ideal setting to examine these variations for the binary classification dataset. The study made the investigation of 25 architectural combinations through five levels of depth and five width to give a broader view of how various architectural configurations influenced the performance.

The results showed that the depth of a network is a key factor in predicting its ability to identify elaborate and tiered structures inherent in the data. Shallow networks, despite their low computational complexity, were unable to capture complex interactions between variables; they underfitted a number of data sets. However,, deeper networks with more than one hidden layers could be able to learn complex patterns in the data since it was much deeper. However, the rise in depth also brought certain problems like taking comparatively more time in training the network and higher probability of overfitting, it rises along with insufficiency in the amount of regularization techniques or significantly smaller dataset.

The width of the network also became highlighted as an important variable. Models with small numbers of neurons in their hidden layers did not effectively accommodate complex data in the database and were less effective. On the other hand, models with more neurons per layer, that is larger networks, had better representational capabilities to learn more complex patterns of nonlinearity in the data. However, this increased capabilities turned out to have its vehement drawbacks, and incorporated such forms of problems such as overfitting of the program and a higher computational load that made training cumbersome and time-consuming.

Overall, this work offers insights into behavior of MLPs and its architectural design. It will unmask a useful analysis of the depth and width trade-off, making it a more realistic decomposition for designing neural networks. The results presented here help advance the existing knowledge on the task of optimization of the neural networks and offer practical advice to those inclined to enhance the efficiency of their creations in machine learning applications.

6. References

- **Good fellow, I., Bengio, Y., & Courville, A. (2016)**
  Deep Learning. MIT Press.
  - This foundational book covers the theoretical and practical aspects of deep learning, including MLPs, network architectures, and optimization techniques.
- **Hastie, T., Tibshirani, R., & Friedman, J. (2009)**
  The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.
  - Explains fundamental machine learning concepts, including neural networks and their applications.
- **Lacuna, Y., Bengio, Y., & Hinton, G. (2015)**
  "Deep Learning," *Nature*, 521(7553), 436–444.
  - A review article discussing the evolution of deep learning and the role of neural network architectures like MLPs.

- **Heaton, J. (2017)**
  "Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep Learning," *Genetic Programming and Evolvable Machines*, 19(1), 305–307.

  - Provides insights into key optimization challenges and architecture design in deep learning.

- **Pedregosa, F., et al. (2011)**
  "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, 12, 2825–2830.

  - The library used in this project, offering tools for MLP implementation and evaluation.

- **Kingman, D. P., & Ba, J. (2014)**
  "Adam: A Method for Stochastic Optimization," arXiv preprint arXiv:1412.6980.

  - Introduces the Adam optimizer, a widely used algorithm for training neural networks, including MLPs.

- **Srivastava, N., et al. (2014)**
  "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, 15(1), 1929–1958.

  - Discusses the use of dropout for regularization in neural networks, relevant for deeper architectures.

- **Zhang, C., et al. (2017)**
  "Understanding Deep Learning Requires Rethinking Generalization," arXiv preprint arXiv: 1611.03530.

  - Explores the generalization behavior of deep networks, providing context for evaluating model performance.

- **Ng, A. (2011)**
  "Neural Networks and Deep Learning," Coursera Machine Learning Course.

  - Provides a beginner-friendly yet detailed overview of neural networks, including MLP configurations.

- **Bishop, C. M. (1995)**
  Neural Networks for Pattern Recognition. Oxford University Press.

- A classic text focusing on the principles of neural network design and their applications in pattern recognition.