MULTILAYER PERCEPTRONS (MLPS): EXPLORING THE EFFECT OF NETWORK DEPTH ON PERFORMANCE

Name: Shaharbanu Parlimoolayil Kadirukunju

GITHUB: https://github.com/shaharbanu-123/ML-NN-INDVTL-ASSGNMNT

1. Introduction

Multilayer Perceptrons (MLPs) are one of the foundational models in deep learning, particularly in tasks such as classification, regression, and prediction. As a class of feedforward neural networks, MLPs use fully connected layers where each neuron in one layer is connected to every neuron in the next layer. The network learns by adjusting its weights during training via a process called backpropagation, utilizing gradient descent or its variants.

An essential characteristic of MLPs is their depth, which refers to the number of hidden layers between the input and output layers. This depth has profound effects on how the model learns, generalizes, and computes the output. A network with greater depth has more capacity to learn complex, hierarchical features from the input data, but at the same time, it faces the risk of overfitting—a phenomenon where the model becomes too specialized to the training data and performs poorly on unseen data.

In this tutorial, we explore the effect of network depth on **MLP performance**, focusing on how the number of hidden layers influences the **accuracy of the model's training, validation, and generalization capabilities**. We utilize the **CIFAR-10 dataset**, which contains 60,000 32x32 color images classified into 10 distinct categories. This dataset offers a challenging yet manageable problem for experimenting with the effect of network depth.

Objective of This Study

Understanding how the depth of the MLP affects its performance is the aim of this study. We specifically look at how the model's capacity to recognize intricate patterns, prevent overfitting, and successfully generalize to new data is impacted by the number of hidden layers. We will determine whether deeper networks always yield greater performance or if they come with extra difficulties, such overfitting, by experimenting with various network depths.

2. Theoretical Background

Comprehending the MLP Structure

An MLP is made up of three primary types of layers:

Input Layer: This layer signifies the input data (such as pixel values of an image or feature vectors). Each neuron in this layer corresponds to a specific feature of the input.

Hidden Layers: These layers analyze the input data by applying learned weights and biases, which aid the network in identifying patterns and relationships within the data. The hidden layers utilize activation functions to add non-linearity to the network, allowing it to learn complex patterns.

Output Layer: The output layer generates the final prediction or classification. In scenarios involving classification, this layer employs a **Softmax** activation to produce a probability distribution across the possible categories.

Important Elements of MLPs

Weights and Biases: Weights are the parameters that the model learns during training, where as biases aid in introducing flexibility and shifting the activation function.

Activation Functions

Activation Functions: Given a weighted sum of inputs, activation functions determine a neu ron's output. They are crucial for giving the network non-
linearity so that it may approximate more complicated functions.
Typical activation functions consist of:

Rectified linear units, or ReLUs, are frequently found in MLPs' hidden layers. They output the input value for positive inputs and zero for negative inputs.
It reduces the **vanishing gradient issue** and is computationally effective. An S-
shaped function that produces values between 0 and 1 is called a sigmoid.
Although it was employed in the past, deep networks suffer from disappearing gradients.

Tanh: More appropriate for some applications, this model is comparable to Sigmoid but produces values between -1 and 1.
Softmax: It transforms input logits into probability values and is used in the output layer for multi-class classification problems.
Gradient Descent and Backpropagation: The backpropagation algorithm propagates the error backward through the network to calculate the gradient of the loss function with respect to each weight. To minimize the loss, the weights are iteratively updated using gradient descent or its variations (e.g., Adam optimizer).

The Impact of Network Depth on Performance

An MLP's representational capacity is directly impacted by its depth. Intricate and hierarchical features can be learned by deeper networks, but they also come with drawbacks like overfitting, gradient problems, and processing expenses. Determining the ideal network size for a particular activity requires an understanding of the link between depth and performance.

Shallow Networks: Limited Capacity and Underfitting

The intricacy of high-dimensional data can be difficult for shallow networks (with fewer hidden layers) to capture. A network with just one hidden layer, for instance, could not be able to understand intricate data distributions, which could result in underfitting. Poor training and validation performance results from a model that is too simplistic to learn the underlying patterns.

Deep Networks: High Capacity and Overfitting

However, deep networks with a large number of hidden layers are far more capable of learning and recognizing intricate patterns.

Overfitting: Overfitting is a problem associated with this enhanced capacity, though. When a model gets overly specialized in the training data, it overfits and captures not only the pertinent patterns but also the noise or erratic variations in the data. As a result, the model may perform well on training data but poorly on data that hasn't been seen yet.

Exploding and Vanishing Gradients: In deep networks, the vanishing gradients phenomena also gains importance. It can be challenging for the model to modify the weights in the earlier layers because the gradients of the loss function can get extremely small as they are propagated backward through numerous levels. This poses a special challenge when training extremely deep networks.

Strategies like dropout, early halting, and L2 regularization are used to lessen these problems. By decreasing the model's dependence on particular neurons or by stopping training early to stop it from fitting too much to the noise in the data, these strategies seek to enhance generalization and avoid overfitting.

Computational Cost: Training deeper networks, particularly with huge datasets, takes more time and computational resources.

Regularization Techniques to Combat Overfitting

Overfitting occurs when a model learns the noise or irrelevant patterns in the training data rather than generalizing to unseen data. For deep MLPs, overfitting can be mitigated through various regularization techniques:

Dropout: In dropout, randomly selected neurons are ignored during each training iteration. This prevents the network from relying too heavily on any single neuron and forces it to learn more robust features.

Early Stopping: Early stopping involves monitoring the validation error during training and halting training when the validation error starts to increase, even though the training error might still be decreasing.

L2 Regularization (Weight Decay): L2 regularization adds a penalty term to the loss function that discourages large weight values. This prevents the model from overfitting by reducing the complexity of the learned functions.

Batch Normalization: This technique normalizes the activations of each layer to have zero mean and unit variance. It accelerates training and helps in stabilizing the learning process in deep networks.

3. Preprocessing and Dataset Selection

The Dataset for CIFAR-10
We use the **CIFAR-10 dataset**, which is popular for image classification tasks, for this investigation. The **60,000 32x32 color photos** in the collection are categorized into 10 classes, including dog, cat, airplane, and automobile. The dataset, which is split into **10,000 test images** and **50,000 training images**, presents a difficult task for neural networks because of the high class variation and small image size.

Steps in Preprocessing

We use the following preprocessing methods prior to training:

1. **Normalization:** To guarantee numerical stability, pixel values are divided by 255 and scaled to the interval [0,1]. By guaranteeing that all of the features are on the same scale, this normalization step speeds up the model's convergence.
2. **One-Hot Encoding:** To work with categorical loss functions, class labels are transformed into binary vectors. This procedure converts the categorical labels into a neural network training format.
3. **Data Augmentation:** To boost generalization and artificially expand the dataset size, optional procedures like flipping, rotating, or moving images are employed. The model becomes more resilient to changes in the data with the aid of augmentation.

4. Design of the Model

We construct MLP models with different numbers of hidden layers as shown in the figure (figure 1) in order to examine the effect of depth:

- **Shallow MLP:** One hidden layer with 128 neurons..
- **Medium-Depth MLP**: Two hidden layers with 128 and 64 neurons..
- **Deep MLP:**. Four hidden layers with 256, 128, 64, and 32 neurons.
- **Very Deep MLP:** Six hidden layers with 512, 256, 128, 64, 32, and 16 neurons.

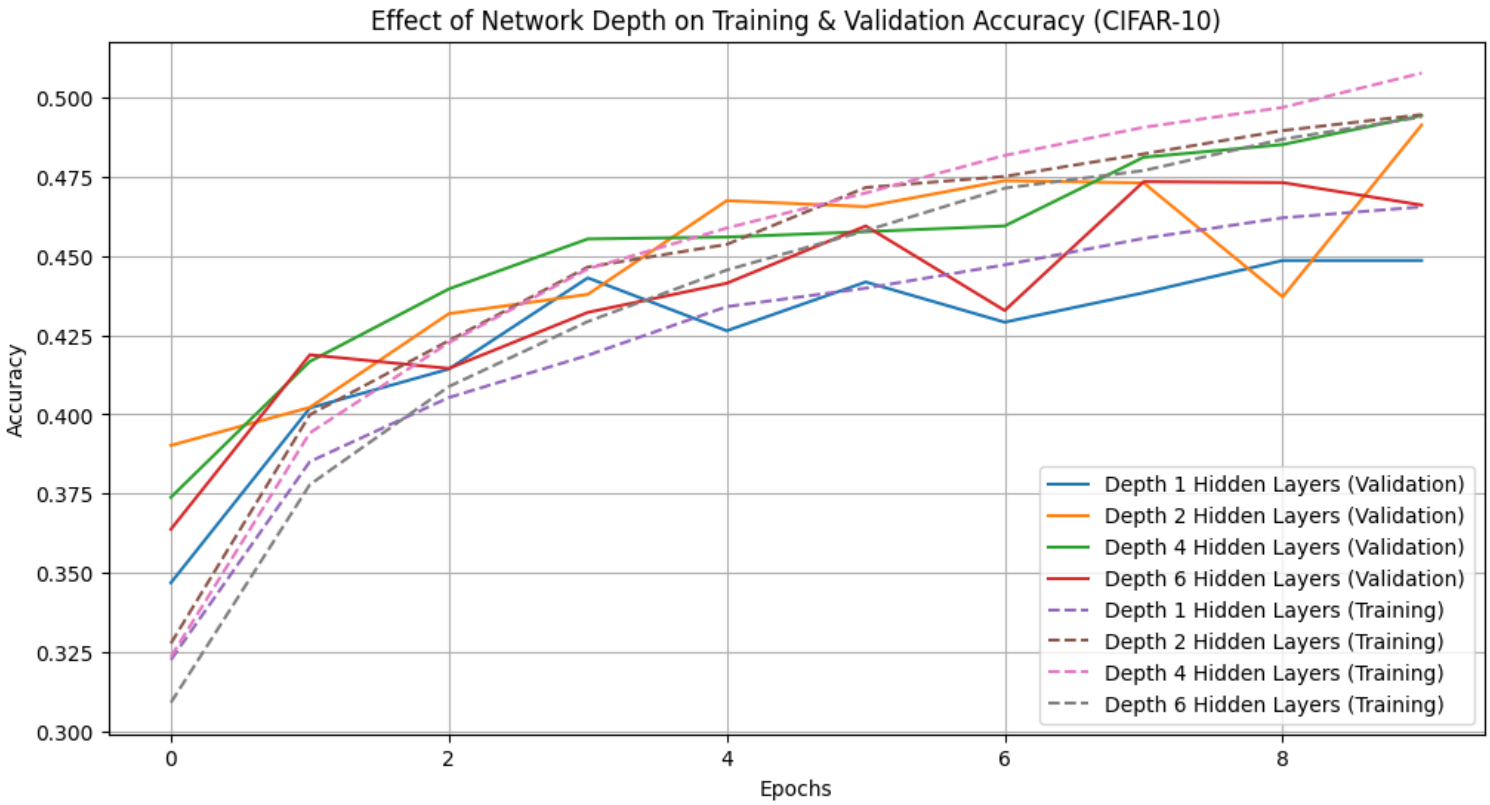

Figure 1 -Effect of Network Depth on Training and Validation Accuracy (CIFAR-10)

(Alt :A graph of different colored lines)

Every model adheres to the same architectural guidelines:

- **ReLU activation is used in hidden layers**. As a result, the network can learn more efficiently and the vanishing gradient issue is lessened.
- To assign input data to one of the ten categories in the CIFAR-10 dataset, **the output layer uses softmax activation.**
- For multi-class classification jobs, **categorical crossentropy loss** is the best option.
- For effective learning, the **Adam optimizer** is selected. Adam is a popular adaptive learning rate optimizer for deep learning model training that exhibits good performance in real-world scenarios.

5. Results Analysis

Accuracy of Training

Deeper networks typically produce higher training accuracy, as might be predicted. The network's ability to recognize intricate patterns in the data improves with each extra layer, which boosts training efficiency. Validation accuracy, however, can not necessarily follow the same pattern as training accuracy.

Accuracy of Validation and Overfitting

We find that when the model begins to overfit the training data, the **validation accuracy** for deeper networks first increases before plateauing or even declining. As seen by the discrepancy between training and validation accuracy, the six-layer model, for instance, demonstrated outstanding training accuracy but substantial overfitting. However, the shallow network's poor performance on both training and validation sets was caused by its incapacity to recognize the intricate patterns in the data.

Overfitting Findings

The model's capacity to generalize to new data decreases with increasing network depth. This was especially evident in the deeper networks, where after a while the validation accuracy began to deviate from the training accuracy. This highlights how crucial regularization strategies are to avoiding overfitting, particularly when working with deep neural networks.

Performance in Generalization

One important indicator of a model's performance on unknown data is generalization. The **medium depth** network displayed a better balance between training and validation accuracy, but the **shallow network's** underfitting resulted in poor generalization. Despite having a high capacity, the **highly deep network** has trouble generalizing because to overfitting.

6. Final thoughts

How Depth Affects MLP Performance

In conclusion, an MLP's capacity to identify intricate patterns in the data is greatly influenced by its depth. Although deeper networks can represent more data, they are also more likely to overfit. While larger networks may not generalize well if they are not properly regularized, shallow networks have a tendency to underfit the data. This emphasizes how crucial it is to determine the ideal network depth for a particular dataset and task.

Prospects for the Future

Because **convolutional neural networks (CNNs)** are better at processing picture data and are more effective at learning spatial hierarchies, researchers may investigate their application for image classification tasks in future study. To get better results, more **sophisticated architectures** like **Residual Networks (ResNets),** which use skip connections to make training deep networks easier, should also be investigated.

7. References

- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
- LeCun, Y., et al. (1998). Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE.
- Srivastava, N., Hinton, G., Krizhevsky, A., et al. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research