# Comparative Analysis of Shallow and Deep Neural Networks Fully Connected vs. Convolutional Architectures

Varsha Rajashekar
University of South Dakota
Varsha.rajashekar@coyotes.usd.edu

## Objective

The primary objective of this assignment is to compare the performance of shallow and deep neural networks, focusing on Fully Connected Neural Networks (FCNNs) and Convolutional Neural Networks (CNNs). By maintaining comparable model complexity across architectures, you will gain insights into the impact of depth and architecture type on model performance and computational efficiency.

## Methodology

### Dataset Selection:

The MNIST dataset consists of 60,000 training images and 10,000 test images.
The datasets are used for handwritten digit recognition, which consists of 28x28 grayscale images.
The images are accessible through TensorFlow, PyTorch, or Keras datasets.

### Model Architecture:

### 1. Shallow Fully Connected Neural Network (FCNN):

- It's the neural network with only one hidden layer between the input and output layers.
- Every neuron in one layer is connected to every neuron in the next layer, making it a fully connected structure.
- It has 784 neurons for the input layer (For the MNIST dataset) and one hidden layer with a specific number of neurons to match complexity.
- It uses ReLU activation in the hidden layer and SoftMax activation in the output layer.

### 2. Deep Fully Connected Neural Network (FCNN):

- It's a neural network architecture where every neuron in one layer is connected to every neuron in the next layer.
- It has 784 neurons for the input and 4-5 hidden layers that are chosen neurons to match the parameter count of other models.
- Uses ReLU activations in each hidden layer to introduce non-linearity.
- Designed to test whether increasing network depth improves classification accuracy.

### 3. Shallow Convolutional Neural Network (CNN):

- A simple CNN architecture with a limited number of layers followed by a fully connected output layer.
- Uses ReLU activation to introduce non-linearity before flattening the feature maps. Intended to compare CNN performance against FCNN models.

### 4. Deep Convolutional Neural Network (CNN):

- A more complex CNN with three convolutional layers.
- Uses multiple feature extraction layers to better capture spatial hierarchies in the image.
- Expected to outperform shallow models due to hierarchical feature learning.

## Parameter Balancing:

To make a fair comparison between the four models, I have adjusted the key parameters like the number of neurons for FCNNs, kernel sizes of filters in CNNs, and overall network depth to share a similar overall number of trainable parameters. For FCNNs, complexity was maintained by adjusting neuron numbers. Such an arrangement allowed the deep FCNN to take the benefit of hierarchical feature extraction and minimize the overfitting risk to make sure both the networks had almost the same number of parameters.

For CNNs, controlled the number of filters and kernel sizes. The deep CNN consisted of three convolutional layers with 8, 16, and 32 filters, each with a 5×5 kernel, whereas the shallow CNN had one convolutional layer with 18 filters and a 5×5 kernel. This systematic increase in the number of filters enabled efficient hierarchical feature extraction while maintaining parameter consistency.

To reduce the amount of parameters, CNN architectures kept the number of fully connected layers low. The final output layer in all models contained 10 neurons (one per MNIST digit) to keep things uniform. These modifications ensured that differences in performance arose from architecture differences rather than differences in parameters. This allowed objective analysis of how the depth of the network and architecture design (FCNN vs. CNN) affected MNIST classification.

## Training Setup:

**Activation Function**: ReLU (Rectified Linear Unit) was applied in all hidden layers to efficiently handle vanishing gradients and enhance model convergence.

**Optimizer**: Adam (Adaptive Moment Estimation) was selected for its adaptive learning rates and momentum-based updates, making it highly effective for deep networks.

**Loss Function**: Sparse Categorical Cross-Entropy Loss was utilized, as it is ideal for classification tasks with integer-labeled categories. It measures the difference between the predicted probability distribution and the actual class labels.

**Batch Size & Epochs**: A batch size of 32 was chosen to balance computational efficiency and generalization. Models were trained for up to 20 epochs, with an early stopping mechanism (patience of 5 epochs) to prevent overfitting.

**Validation Split**: 20% of the training data was reserved for validation to track model performance.

**Evaluation Metrics**: Accuracy was the primary metric for assessing performance on both training and test datasets. Additionally, loss curves were analyzed to identify signs of overfitting or underfitting.

Evaluation Results for Shallow FCNN:

| Metric | Value |
|---|---|
| Precision | 0.98 |
| Recall | 0.98 |
| F1-Score | 0.98 |

Evaluation Results for Deep FCNN:

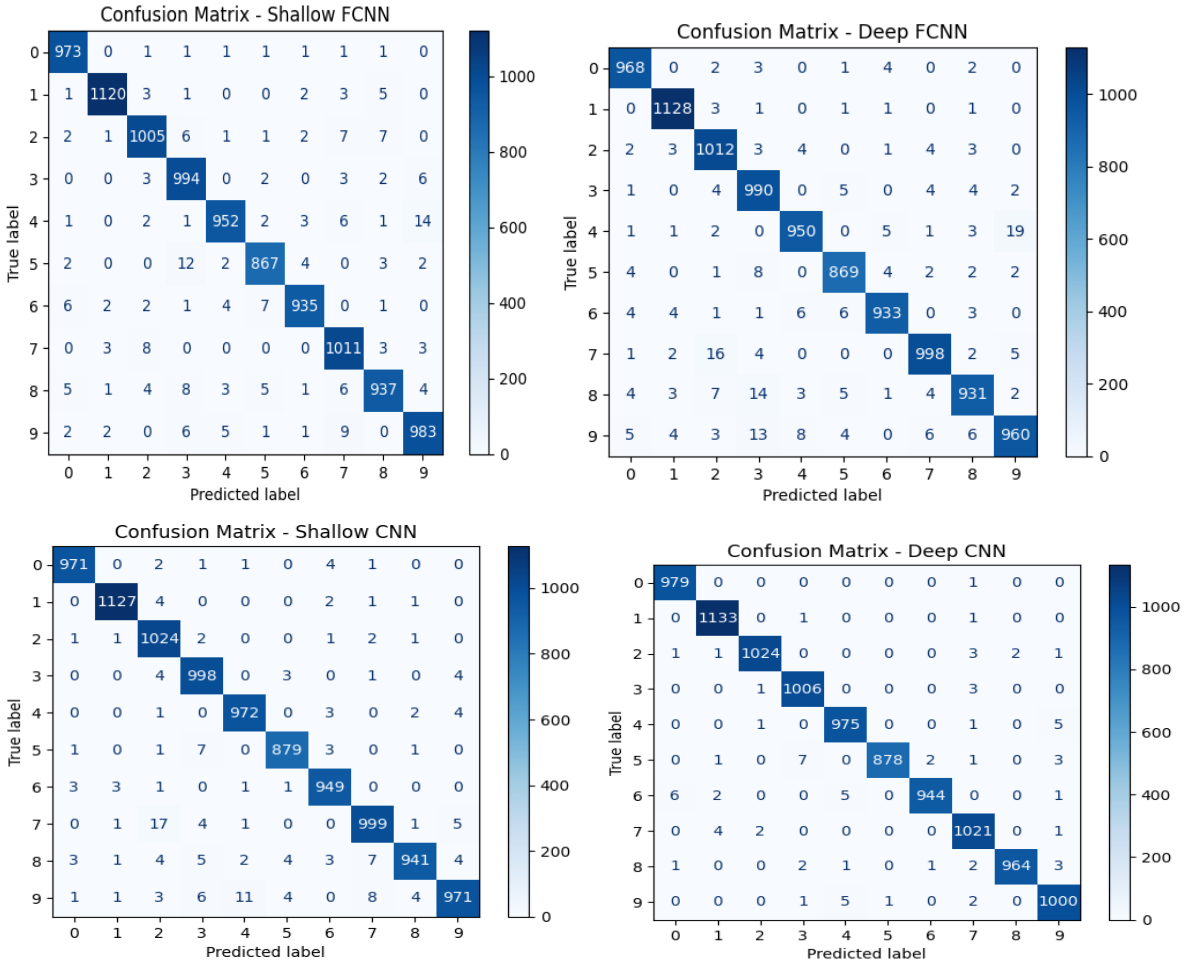
| Metric | Value |
|---|---|
| Precision | 0.98 |
| Recall | 0.98 |
| F1-Score | 0.98 |

```
Evaluation Results for Shallow CNN:
------------------------------------
Metric          Value
------------------------------------
Precision         0.98
Recall            0.98
F1-Score          0.98
------------------------------------
```

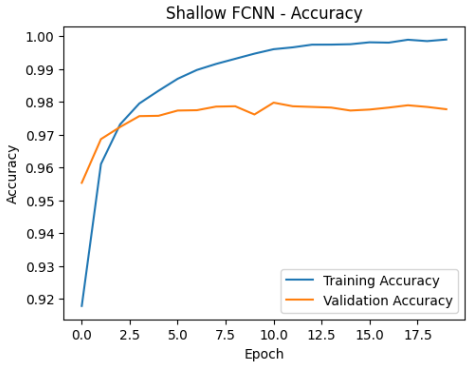
```
Evaluation Results for Deep CNN:
------------------------------------
Metric          Value
------------------------------------
Precision         0.99
Recall            0.99
F1-Score          0.99
------------------------------------
```

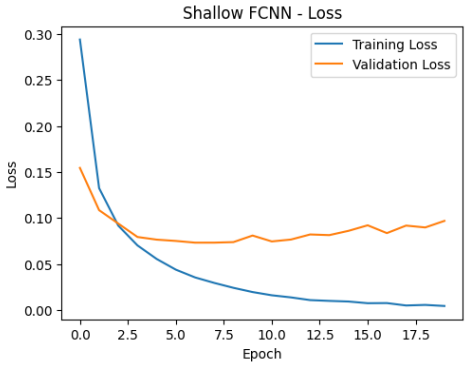
**Confusion Matrix:**

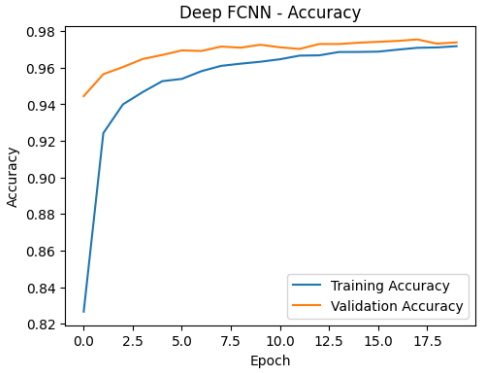
A confusion matrix visualizes and summarizes the performance of a classification algorithm. It compares the actual values of a target variable to the predicted values.

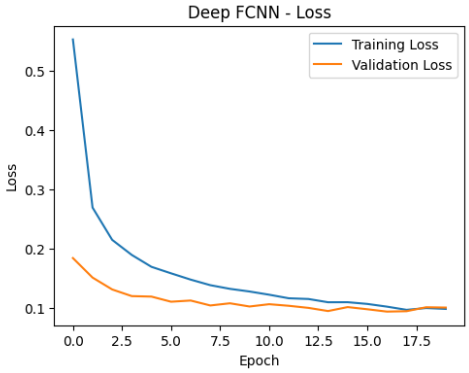


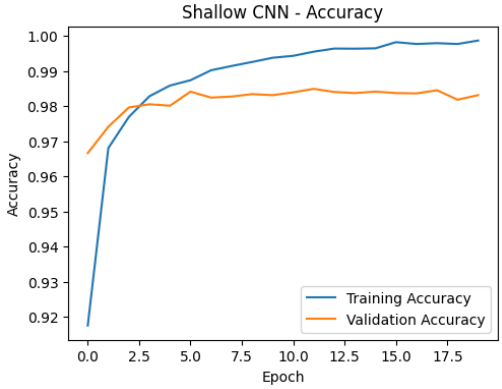
## Plots:

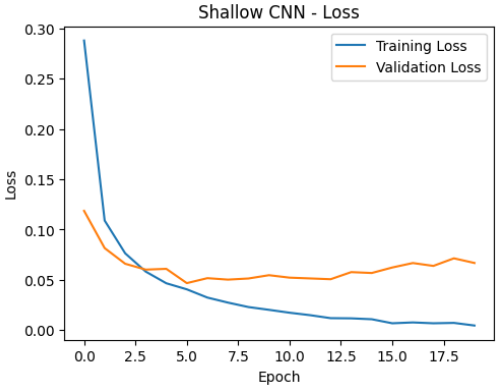

Shallow FCNN - Accuracy
Accuracy
Epoch
Training Accuracy
Validation Accuracy

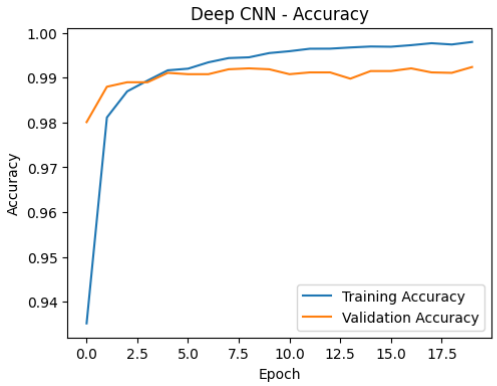

Shallow FCNN - Loss
Loss
Epoch
Training Loss
Validation Loss

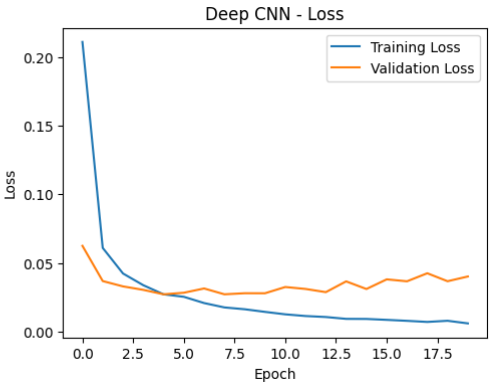

Deep FCNN - Accuracy
Accuracy
Epoch
Training Accuracy
Validation Accuracy


Deep FCNN - Loss
Loss
Epoch
Training Loss
Validation Loss


Shallow CNN - Accuracy
Accuracy
Epoch
Training Accuracy
Validation Accuracy


Shallow CNN - Loss
Loss
Epoch
Training Loss
Validation Loss


Deep CNN - Accuracy
Accuracy
Epoch
Training Accuracy
Validation Accuracy


Deep CNN - Loss
Loss
Epoch
Training Loss
Validation Loss

## Analysis

**Performance Comparison**: Deep FCNN and CNN models were likely to be more accurate than their shallow counterparts, likely since they possess greater ability to learn complex data representations.

**Architecture Impact**: CNNs performed significantly better than FCNNs on the MNIST data set, as they can effectively learn spatial information and local features, which is very important in image recognition.

**Overfitting and Underfitting**: Deep FCNN suffered from overfitting, as indicated by the significant gap between validation and training accuracy. This may be due to its large number of parameters, which makes it prone to memorization of noise during training.

**Training Efficiency**: Despite their complexity in structure, CNNs may be computationally more efficient than FCNNs with the same parameters due to parameter sharing and sparse connectivity among convolutional layers.

**Visualization Observations**: Loss and accuracy plots exhibited the usual pattern of loss diminishing and accuracy enhancing as training proceeded. The Deep FCNN, however, showed a clear divergence in training and validation accuracy, an overfitting indicator.

**Model Depth vs. Performance**: Increased model depth did not necessarily result in a corresponding increase in performance. The Deep FCNN, despite all its depth, did not always outperform the other models, indicating that excessive depth can cause diminishing returns and overfitting.

## Conclusion:

This analysis highlights the effectiveness of CNNs for image classification on the MNIST dataset. While deeper models tended to achieve higher accuracy, overfitting became a challenge as depth increased. CNNs outperformed FCNNs due to their ability to efficiently capture spatial features. These results emphasize the importance of architectural choices and the need to carefully balance model complexity for optimal performance.