Comparing Basic DNN Architectures on MNIST dataset

Christos Panourgias

Contents

1 Introduction

Deep Neural Networks (DNN) and Convolutional Neural Networks (CNN) are two powerful machine learning algorithms that have been widely used in the field of computer vision. Top performances on various datasets, including the MNIST dataset, have been achieved with these algorithms.

The MNIST dataset, consisting of 60,000 training images and 10,000 test images, represents grayscale images of handwritten digits ranging from 0 to 9. The goal is to develop models capable of accurately classifying these images in their respective categories. Extensive research has been conducted on this dataset, making it a fundamental resource for evaluating and comparing different algorithms.

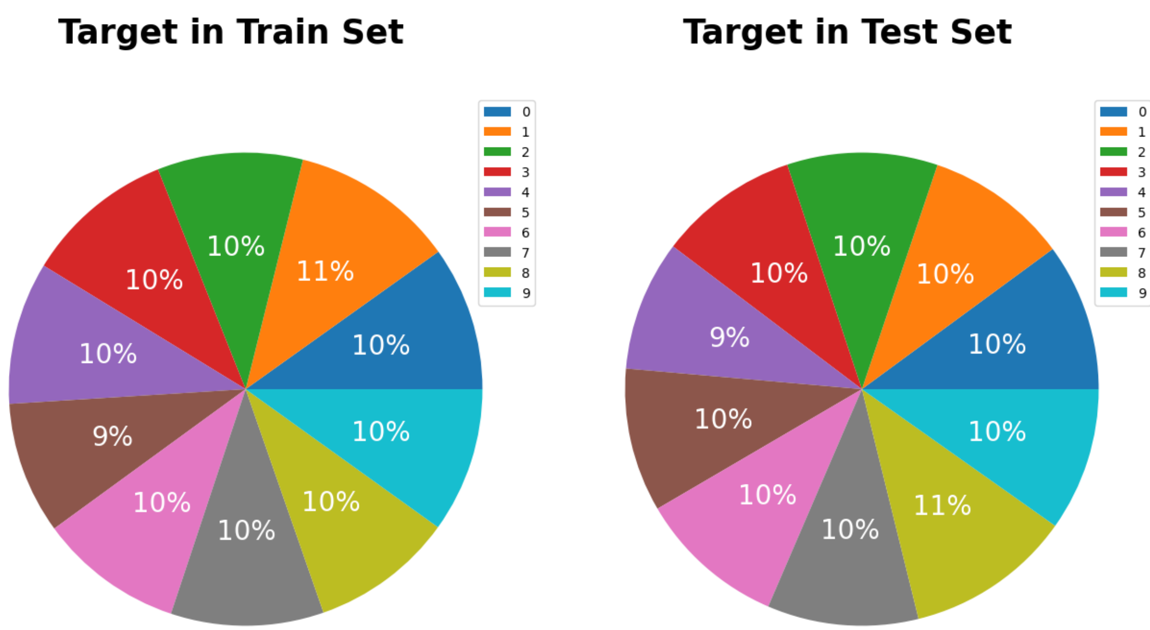
In this work, the performances of DNN and CNN on the MNIST dataset were compared. DNNs are artificial neural networks that can learn hierarchical representations of input data through multiple layers of neurons. On the other hand, CNNs are specifically designed for image processing tasks, incorporating features such as convolution and pooling layers, which allow them to efficiently capture spatial dependencies in images.

The objective of this work is to analyze and compare the capabilities of these two neural network architectures in terms of accuracy, and computational efficiency. By examining their performance on the MNIST dataset, insights into the strengths and weaknesses of DNN and CNN can be gained, making it easier to understand which architecture is more suitable for image classification tasks .

2 Applied Methods

2.1 Class Balance Inspection

First, the percentage of each class in the data set was inspected to ensure that models will give equal importance to each class. As shown below, the classes are satisfactorily balanced in both sets.



2.2 Data Normalization and Partitioning

The data are grayscale images which are 28×28 pixel and each pixel receives a value from 0 to 255. Therefore, to normalize the data it is sufficient to divide the images with 255.

In loading the data there is already a split into a training set and a test set, however in choosing the hyper-parameters of the model it will be necessary to split the training set so that a validation set is created. This helps to unbiasedly assess the model's performance on new data, which is critical in determining its ability to generalize and make accurate predictions. For this separation, the stratified 6-fold cross-validation method was used, where the training set was divided 6 times into a training set and a validation set, maintaining the balance of the classes that the original set had.

In addition, the data were formatted differently to be inputted into each model. Where for DNN they were reshaped in the form $(1, 28 \cdot 28)$ so as to be inserted into the neuron layers. While for CNN batches of the images themselves were created.

2.3 Building and Training Different Model Architectures

5 different DNN model architectures were constructed by randomly choosing parameter settings such as the size of batches, the number of epochs, the number of neuron layers, the number of neurons for each layer and the assignment or not of Dropout layers. In the same way, 5 different CNN model architectures were constructed by randomly choosing the additional parameters for pooling size, number of filters and their size.

Thus the following model architectures were constructed.

- DNN

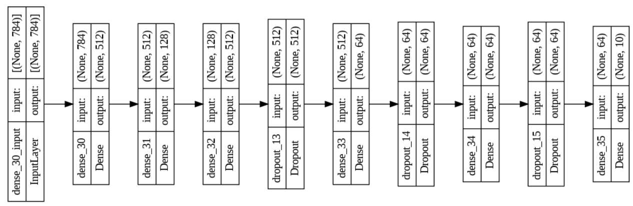
- CNN

2.4 Model Training and Selection

The above models were trained with the Adam optimizer and with respect to the categorical cross entropy error, the corresponding classification metrics were calculated for each fold and then the metrics were averaged over the test set by all folds. Therefore the model that achieved the highest mean accuracy in the test set was selected.

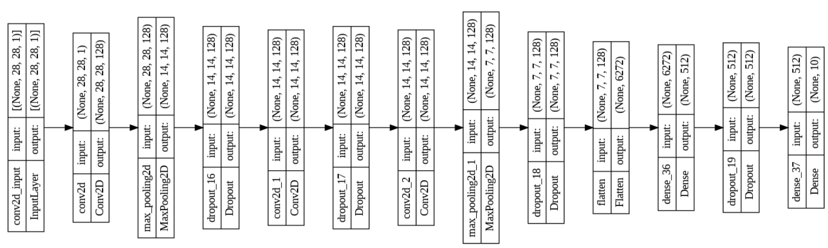
3 Experimental Results

The DNN architecture that proved to be the best is the following



Where, averaged over all folds, it extracted the following metric evaluation on the test set : accuracy = 97.86%, precision = 97.88%, recall = 97.84%, f1 = 97.85%

The CNN architecture that proved to be the best is the following

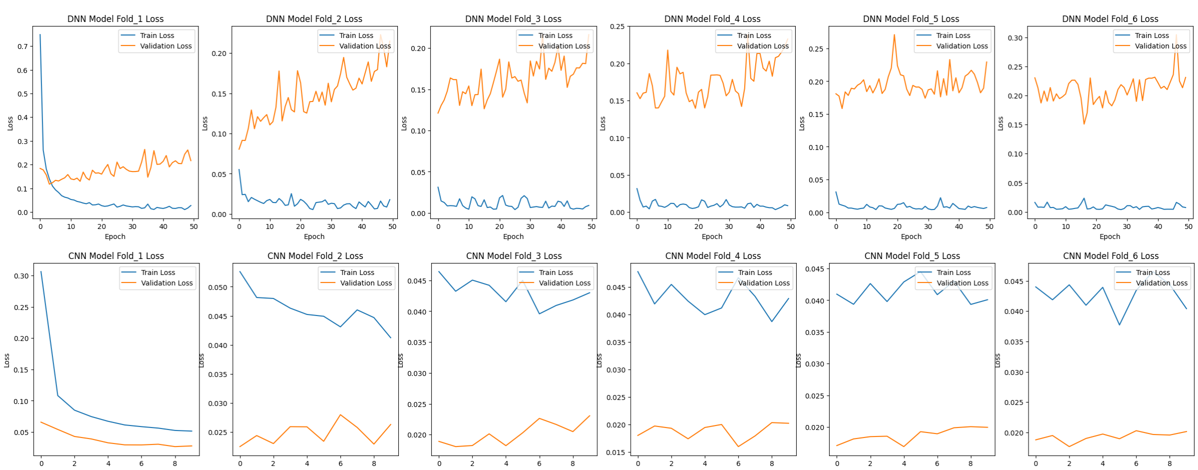


Where, averaged over all folds, it extracted the following metric evaluation on the test set : accuracy = 99.41%, precision = 99.41%, recall = 99.40%, f1 = 99.40%

The following table shows the evaluations of these models for each fold from which it can be seen that, while the DNN model had quite decent results, the CNN model had higher evaluations in the metrics.

Technique Name	Set	Fold Number	Accuracy	Precision	Recall	F1-Score
DNN	Train	1	0.997475	0.997408	0.9974925	0.997445
DNN	Test	1	0.9799	0.979717	0.9799198	0.979776
DNN	Train	2	0.997	0.997046	0.9969337	0.996977
DNN	Test	2	0.9764	0.976857	0.9760431	0.976296
DNN	Train	3	0.996425	0.996501	0.9963682	0.996417
DNN	Test	3	0.974	0.974884	0.9735829	0.973925
DNN	Train	4	0.99975	0.999737	0.9997507	0.999744
DNN	Test	4	0.9812	0.9812	0.980963	0.981036
DNN	Train	5	0.999675	0.999672	0.9996698	0.999671
DNN	Test	5	0.9798	0.979874	0.9796111	0.979704
DNN	Train	6	0.9998	0.999792	0.9997984	0.999795
DNN	Test	6	0.9805	0.980499	0.9802804	0.980352
CNN	Train	1	0.996075	0.996061	0.9960587	0.996057
CNN	Test	1	0.9928	0.992774	0.992758	0.992757
CNN	Train	2	0.997575	0.997595	0.9975846	0.997588
CNN	Test	2	0.9937	0.993706	0.9935753	0.993632
CNN	Train	3	0.997975	0.997985	0.9979718	0.997976
CNN	Test	3	0.9945	0.99448	0.9943998	0.994431
CNN	Train	4	0.9986	0.998597	0.9986053	0.998601
CNN	Test	4	0.9948	0.994716	0.9947117	0.994712
CNN	Train	5	0.99845	0.998473	0.998439	0.998455
CNN	Test	5	0.9942	0.99422	0.9940785	0.994145
CNN	Train	6	0.999025	0.999037	0.9990135	0.999025
CNN	Test	6	0.9948	0.994763	0.994741	0.994749

In addition, it is worth noting that the CNN model was trained for 10 epochs while the DNN model was trained for 50 epochs. As can be seen in the graphs below, the training of the CNN is clearly faster and more efficient than DNN



It is also observed that DNN exhibits a greater deviation in the validation set error compared with the training set error than CNN, this implies that CNN generalizes better the results of and thus is more accurate in predicting new data.

4 Conclusions

After a thorough analysis and comparison of DNN and CNN on the MNIST dataset, it is evident that the CNN model outperforms the DNN model in terms of classification accuracy . CNN's ability to capture spatial dependencies in images through its convolutional layers proved advantageous for the task of handwritten digit recognition. The CNN model achieved higher accuracy rates and showed superior generalization capabilities when evaluated on both the validation and test sets. These results highlight the importance of exploiting the architectural design and specialized features of CNNs in image classification tasks, especially on datasets such as MNIST.

In addition to achieving higher classification accuracy, the CNN model also showed superior performance in terms of computational efficiency and training time. The inherent architecture of CNNs, with shared weights and local receptive fields, allows them to effectively reduce the number of parameters and exploit spatial location, resulting in more efficient computations compared to DNN.

Additionally, the CNN model's ability to automatically learn hierarchical representations of the input data through its convolutional layers proved extremely beneficial for the MNIST dataset. By exploiting local patterns and spatial relationships, the CNN effectively extracted important features from the handwritten digit images, leading to enhanced discrimination and improved classification performance.

While DNN are powerful models capable of learning complex representations, their performance on the MNIST dataset can be limited due to the lack of native mechanisms for handling spatial information. DNNs are based on fully connected layers, which do not take into account the spatial arrangement of pixels in an image. As a result, they may struggle to effectively capture the local patterns and dependencies needed to accurately classify digits.

It is important to note that the superiority of CNNs over DNNs on the MNIST dataset does not mean that CNNs are always better in all image classification tasks. The performance comparison is specific to the MNIST data set and the nature of handwritten digit recognition. Depending on the characteristics of the data set and the specific problem, the choice between DNN and CNN may differ. In conclusion, for applications involving the classification of handwritten digits, it is recommended to use a CNN model to achieve optimal results.