Introduction

This document utilises Keras, a high-level deep-learning framework developed on TensorFlow, to define and train neural networks. Keras streamlines the development of intricate models, thereby making it particularly suitable for constructing and training neural networks. A Multilayer Perceptron (MLP) constitutes a fully connected network comprising a minimum of three layers —input, hidden, and output—for classification and regression purposes. In contrast, a Convolutional Neural Network (CNN) is specifically designed to analyse images and videos, where convolutional layers effectively capture spatial patterns. This assignment aims to compare MLP and CNN architectures, adjust hyperparameters, and assess their performance.

Task 1

Task 1 consists of two parts:

1. **Part 1**: Using the MNIST dataset of handwritten digits, explore two Keras examples with an MLP and a CNN to build basic neural networks.
2. **Part 2**:
    - **a)** Develop an MLP for the Fashion MNIST dataset (28x28 grayscale images, 10 classes) and apply the same model to the CIFAR-10 dataset (32x32 RGB images, 10 classes).
    - **b)** Build a CNN for Fashion MNIST and replicate it for CIFAR-10.

These exercises will deepen our understanding of initialisation, activation functions, optimisers, regularisation, and architectures.

Task 1.1

We undertook experiments utilising various metrics to comprehend the influence of architecture and hyperparameter tuning on the performance of Multi-Layer Perceptrons (MLP) and Convolutional Neural Networks (CNN). The metrics employed encompassed accuracy, precision, recall, top-k categorical accuracy, and mean squared error, thus providing valuable insights into the efficacy of each model.

Our initial experiments involved adjusting layers. For the MLP, increasing depth and width led to slight performance drops, likely due to the model's increased complexity on simpler datasets. For the CNN, adding `Conv2D` layers before dense layers significantly improved performance by capturing spatial patterns more effectively. We tested multiple activation functions (`tanh`, `elu`, `leaky-relu`), observing minimal variation among them. Initialisations such as Glorot, He, and Orthogonal yielded close results within a 2% range. Among optimisers, SGD slightly outperformed others, with similar results from Adam and RMSProp. Dropout at 0.4 proved most effective for regularisation.

This experiment demonstrated the influence of architecture and hyperparameters on MLP and CNN models, enhancing our ability to tune networks for optimal performance across different tasks.

Task 1.2.a

This series of experiments investigates the performance impact of various neural network architectures and hyperparameter configurations on the Fashion MNIST and CIFAR-10 datasets. Employing a Multilayer Perceptron (MLP) approach, we assess three distinct configurations featuring varying activation functions, dropout rates, and optimisers. By applying these models to both datasets, our objective is to elucidate how architectural choices influence performance, especially when progressing from a simpler dataset (Fashion MNIST) to a more complex, higher-dimensional one (CIFAR-10).

Experiment 1: The inaugural model incorporates ReLU activations, a three-layer structure, HeNormal initialisation, and a 30% dropout rate. It is optimised using the Adam optimiser and aims to strike a balance between depth and regularisation.
Experiment 2: The subsequent model utilises ELU activations with a slightly reduced depth (three dense layers comprising 512, 256, and 128 nodes) alongside a 25% dropout rate with HeNormal initialisation. This model is optimised with RMSprop to examine the implications of a smoother gradient approach.
Experiment 3: We experiment with a larger model employing LeakyReLU activations in the final configuration. This model consists of 1024, 512, and 256 nodes per layer, accompanied by a higher dropout rate (40%) to mitigate overfitting. Optimised with Stochastic Gradient Descent (SGD) and momentum, it assesses the impact of a gradual, stabilised gradient descent on both datasets.

Each model was initially trained on the Fashion MNIST dataset, following which the same configurations were applied to CIFAR-10. To facilitate comparing performance across datasets, accuracy-related metrics (precision, recall, and top-k categorical accuracy) and error-related metrics (mean squared error) were thoroughly evaluated.

The experiments produced significant insights into the ways architectural choices affect performance on diverse datasets. The first model, characterised by ReLU activations and the Adam optimiser, performed effectively on Fashion MNIST but exhibited limited adaptability to the more complex CIFAR-10 data, likely due to the simpler activation and optimiser configurations. Conversely, utilising ELU and RMSprop, the second model displayed enhanced adaptability to CIFAR-10, manifesting a smoother convergence across epochs. Nevertheless, the third model, integrating LeakyReLU and SGD, yielded the most balanced performance, particularly on CIFAR-10, where the heightened complexity necessitated robust regularisation and meticulous gradient descent. In summary, this series of experiments accentuates the significance of regularisation, activation function selection, and optimiser tuning when transitioning from a simpler dataset to a more intricate one, such as CIFAR-10.

Task 1.2.b

In this series of experiments, we investigate the performance of three distinct Convolutional Neural Network (CNN) architectures applied to the Fashion MNIST and CIFAR-10 datasets. Each model configuration incorporates unique combinations of activation functions, optimisers, and regularisation techniques. By implementing these models on both datasets, we aim to elucidate how architectural designs and hyperparameter selections influence performance across varying levels of data complexity.

Experiment 1: This model utilises ReLU activations and consists of two convolutional layers incorporating MaxPooling for down-sampling, followed by a fully connected layer with a dropout mechanism. The Adam optimiser is employed, with the objective of achieving rapid convergence accompanied by minimal regularisation.

Experiment 2: This model builds upon the first by integrating BatchNormalisation after each convolutional layer to stabilise and expedite the training process. With three convolutional layers and a more extensive dense layer, the RMSprop optimiser, which maintains a decaying moving average of gradients for enhanced learning control, is utilised.

Experiment 3: The final model employs LeakyReLU activations and incorporates L2 regularisation for each convolutional layer to mitigate overfitting. It also features an elevated dropout rate and utilises SGD with momentum, aiming to balance regularisation and stable, gradual optimisation.

Each model is initially trained and evaluated on the relatively simpler Fashion MNIST dataset and subsequently tested on the more complex CIFAR-10 dataset. Performance is assessed using metrics such as accuracy, precision, recall, top-k categorical accuracy, and loss.

The experiments underscored the influence of CNN depth, normalisation, and activation functions on performance across datasets. The first model employing ReLU and Adam exhibited reasonable performance on Fashion MNIST but encountered difficulties with CIFAR-10, revealing constraints in generalising to more complex data. The second model, characterised by BatchNormalisation and RMSprop, demonstrated enhanced stability when applied to CIFAR-10, implying that normalisation facilitates handling higher-dimensional inputs. The third model, featuring LeakyReLU and regularisation, exhibited the most effective adaptability to CIFAR-10, particularly due to its robust regularisation and gradual optimisation through SGD. This study accentuates the significance of architectural choices and regularisation strategies when transferring models from simpler datasets to more intricate ones.

Task 1 Conclusion

This study conducted a comparative analysis of Multi-Layer Perceptron (MLP) and Convolutional Neural Network (CNN) architectures utilising the Fashion MNIST and CIFAR-10 datasets. The findings indicate that Convolutional Neural Networks significantly surpass Multi-Layer Perceptrons in performance when engaging with complex datasets such as CIFAR-10, attributable to their enhanced capacity for capturing spatial patterns. While Multi-Layer Perceptrons exhibited satisfactory performance on the less intricate Fashion MNIST dataset, they encountered considerable difficulties with CIFAR-10, even after undergoing hyperparameter tuning, thereby underscoring their limitations in handling high-dimensional image data. In contrast, Convolutional Neural Networks, particularly those incorporating dropout, L2 regularisation, and BatchNormalisation, demonstrated a commendable adaptability to CIFAR-10, effectively learning spatial dependencies. This observation reinforces the assertion that Convolutional Neural Networks are essential for tasks involving complex and structured data, highlighting the necessity for architectural choices that align with the dataset’s complexity.