A Neural Network Model for Fashion-MNIST Image Classification

Branch: Artificial Inteligence (MSC)
Module Name : Neural System & Natural Language Processing

Student Details:
Name: Ketan Tarkhala
Student Id: P2956507

Faculty Details:

1) Amudhavel Jayave

Table of Contents

1.1 INTRODUCTION

The neural networks are regarded as one of the simplest constructs in the modern article of artificial intelligence and machine learning. They are mathematical models which act as copying of the human brain in its interaction with brain neurons to which information is being processed. The artificial neurons are the neural network whereby, the input is sent through the neurons which operates the input with mathematical functions that are programmed to send the output to another layer. As time passes the network learns the pattern of data and relations by modifying internal weights in the structure a process known as training. This is the most important distinguishing characteristic of the neural networks as they are very effective in the solving of complex problems as image classification, speech recognition or natural language processing as they can learn automatically through analyzing examples.

A neural network typically has three types of layers, one of which is the input layer which receives the input, one (or more) hidden layers which transforms this input by applying activation functions; and the output layer which produces the final prediction. Backpropagation and gradient descent are the algorithms which give motivation to the learning mechanism and whose aim is to control the errors made in prediction process by adjusting the weights between the neurons. The bigger the model is working with large bulk of data, the higher the accuracy of the model to recognize the patterns and make decisions even without being instructed by a human being on the nature of that choice.

It is on this basis that different versions of neural network structure have been developed to handle other kinds of information. Some of the simplest are the Artificial Neural Networks (ANNs) and are primarily made up of fully connected layers. They are able to be efficient in processing data of organized information or flattened photographs, yet is not efficient at depicting geographic, or local associations in photographs. This deficiency led to the development of the Convolutional Neural Networks (CNNs) that are explicitly designed to be applied to the visual and image-related issues. After the convolutional and the pooling layers, CNNs are involved with the recognition of edges, textures and shapes as features of an image; therefore, they are highly applicable in image classification activities.

This coursework on the Fashion-MNIST dataset, where images have grayscale images of pieces of clothes (T -shirts, trousers, pullovers, sneakers) was done using the ANN and CNN models and their comparison. The project aims at designing, training and testing of these two neural network structures to process the images in recognition and comparison of their performance. Another objective of the project is to learn how preprocessing of data, network structure and training parameters can affect the accuracy of the models.

It was implemented on the stage of TensorFlow/Keras, which is considered one of the most popular deep learning frameworks that can be used to develop the model in a flexible and efficient manner. This project is a learning experience as it introduced me to the different concepts of neural networks by processing data and developing a model to process the data to data analysis and interpretation of outcomes that demonstrated that neural systems are capable of learning how to identify and categorize complex visual information through inputing of the information it receives.

1.2 AIM AND OBJECTIVES

The major goal of this coursework is to design, construct and evaluate 2 types of neural network models including an Artificial Neural Network (ANN) and Convolutional Neural Network (CNN) with the purpose of classifying pictures under Fashion-MNIST data set. Within the scope of the project, the differences between these two architectures are to be discussed as far as they are different in terms of their structure, approach to learning, and the opportunity to recognize the visual patterns in images.

Hopefully through this work, the project will have provided experience in the entire deep learning process, such as dataset preparation until the point of assessing models. Training ANN and CNN models with the identical set of data, the paper introduces into the limelight the influence of the structure of a neural network on the levels to which they can be applicable as predicting how reliable, generalized, and efficient they can be.In order to reach this goal, the following objectives were established:

This goal was to be achieved by the following objectives:

Use Artificial Neural Network (ANNs) in simple image recognition using TensorFlow/ Keras.

In the architecture and implementation of a Convolutional Neural Network (CNN) that will be capable of retrieving spatial features and improving the classification efficiency.

To reformat the Fashion-MNIST data to its processed and normalized format, to balance the data.

In order to find out and compare both models according to measurements of the corresponding values of accuracy, loss, and interpreting a confusion matrix.

To compare the work of ANN and CNN architectures, one has to define the advantages and disadvantages of the two solutions.

To recommend the potential improvements such as data feeding, dropout layers or changing the parameters of the optimer to optimize model performance.

As a result of the accomplishment of these aims, this project not only demonstrates the aptitude of technical implementation of the project, but also offers an analytical view of how the architecture design acts as an add-on to the topic of deep learning

1.3 OVERVIEW OF THE FASHION-MNIST DATASET

In this paper, Fashion-MNIST will be outlined briefly.

The Zalando Research created the Fashion-MNIST data, an accepted machine learning benchmark and a deep learning benchmark to test machine learning and deep learning image classification. It was characterized as a variant of the classic MNIST handwritten digit dataset that is more challenging to solve and more realistic who provides a set of gray images that can probably appear as the object of what one would see in a daily clothing.

The data set shall consist of 70, 000 total in the number of images one of which will be 10, 000 test images whilst 60 000 will be training images. The size of every image is 28x28 pixels and this is specified as in grayscale that is, having a pixel value of between 0 (black) and 255 (white). Small size of image offers fast training, experimentation; hence, easy and complicated neural network structures can be tested. One can categorize all photos into ten categories where they are predetermined as it is a photo of varied types of fashion clothing and accessories. They are classified into the following:

Lable	Class Description
0.	T-Shirt/Top
1.	Trouser
2.	Pullover
3.	Dress
4.	Coat
5.	Sandal
6.	Shirt
7.	Sneaker
8.	Bag
9.	Ankle Boot

Each image in the dataset has one single piece of clothing in the middle of an otherwise plain image, and, again notwithstanding, it is easy to identify objects, but it is difficult to categorize those with similar ones that are more visually similar like the shirts and T-shirts or coat and pullovers.

The data set has been directly imported in this project in TensorFlow/Keras library that can be readily located at the tensorflow.keras.utilities.fashionmnist. The choice was motivated by the fact that this dataset is simple enough and at the same time, is not so simple that it trains easily using a regular computer, primarily because it is a complex dataset, it demonstrates the advantages of convolutional networks over their fully connected counterparts.

In the case of the Fashion-MNIST dataset, the project will be a multi-class image classification scale, where it will strive to predict the type of clothing of the given image to classify it. It is an excellent illustration of what neural networks can think and learn about visual data and how the decision of a model design can influence the precision of classifications.

1.4 DATA PREPROCESSING

Essentially, preprocessing data is an obligatory step to building any neural network at all- as long as the input data is global the model will be unable to function. The preprocessing of this assignment consisted of limited operations conducted to the Fashion-MNIST set and then the Artificial Neural Network (ANN) and the Convolutional Neural Network (CNN) were trained. These procedures served to ensure that data were clean and normalized and were ready to be learnt in a convenient manner.

1.4.1 Loading the Dataset

It was loaded directly out of the TensorFlow/Keras package with the fashionmnist.loaddata() command that was already-included. The training and test sets were in the form of NumPy arrays which I utilized in obtaining the grayscale images and the corresponding class tags at that call. The pictures have the form of squares measuring 28x28 objects and the labels have numbers 0 to 9, indicating a particular type of clothes.

1.4.2 Data Splitting

After being loaded, the dataset was divided into three subsets:

- Training set: 54,000 pictures (to be used in training the model)
- Validation set: 6000 images (to help consider the parametrics of the model and prevent overfitting)
- Test set: 10,000 images (will be used to check the performance of the final model)

It randomly selected 10 per cent of the original training data with the train test split method of scikit learn as the validation set. This way, during the training process the models were taught an invisible data stream to work with, which helped to tune such values as learning rate, batch size, and epochs.

1.4.3 Normalization

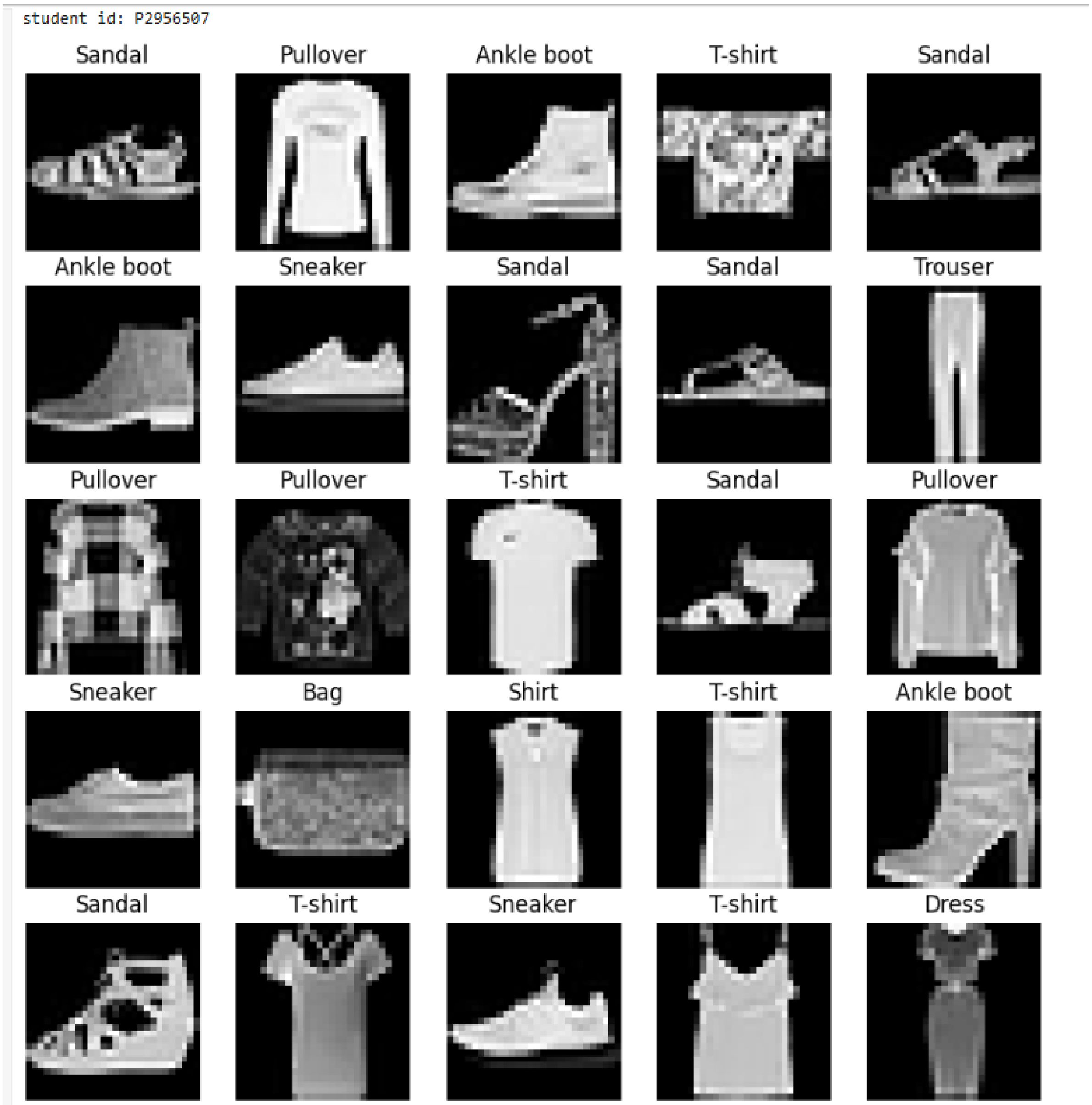
One of the most topical preprocessing stages, in the picture data, is normalization. The range of each pixel was 0-255 at the beginning. To speed and stabilize the learning, I lowered all the pixel values to the range 0-255 to ensure that they fell into the range 0-1. This type of scaling will help the optimizer to stabilize to values that are more favorable, prevent huge shifts in gradients, and play features are equally important.

1.4.4 Label Encoding

Keras contra to-categorical was used to transform the Fashion labels into a one-hot format. The labels are converted to 10-dimensional binary and 1 in the location of the correct class only. The encoding is needed in training of multi-class classification by using categorical cross-entropy loss.

1.4.5 Data Visualization

Sample images of the dataset were viewed before modeling the models to ensure that the data was loaded and normalized correctly. The fact that Matplotlib was used to plot the gra yscale pictures was also instrumental in making me have a better feel of the difference between classes. Each of the images is labeled with the name of the category, in which it falls (ex: T-shirt, Dress, Sneaker).



1.5 ARTIFICIAL NEURAL NETWORK (ANN) MODEL

A neural network was adopted as an Artificial Neural Network (ANN) and used as the baseline in classifying Fashion-MNIST images. ANN takes the input in flattened pixels and uses fully connected layers to determine discriminative features. Although they are simple to use, ANNs are typically not very capable of taking advantage of the spatial structure of images, but this can render them less useful to visual tasks than convolutional networks.

Architecture and logic Model architecture and rationale:

The adopted ANN is of a basic feed-forward topology. The 28x28 images are flattened into a 784 dimensional image by the input layer. The two heavy hidden layers including ReLU activation follow and apply non-linearity to the network to allow the network to learn intricate mappings. The hidden layers are followed by a dropout to achieve better results feedback: to minimize overfitting, DEWA randomly switches off a proportion of the neurons in the training phase. There is a Softmax activated final output layer that generates probabilities of the ten Fashion-MNIST classes. The architecture (as implemented): The implementation architecture (as implemented):

1.5.1 The implementation architecture (as implemented):

- Input: Flatten (28×28 → 784)
- Dense layer 1: 256 units, ReLU activation
- Dense layer 2: 128 units, ReLU activation
- Dropout: 0.2
- Output: Dense 10 units, Softmax

The summary of a Keras model is presented as the appendix (see modelsummary.txt) and the visualization of the architecture was stored as ANNmodelarchitecture.png to make a clearer picture.

1.5.2 Training configuration:

- Optimizer: a default Adam.
- Loss functional: Categorical cross-entropy.
- Metrics: Accuracy

- Epochs: 20
- Batch size: 128

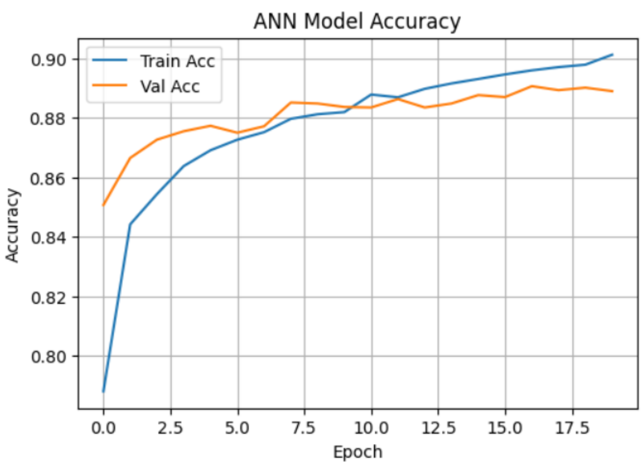
1.5.3 Training performance:

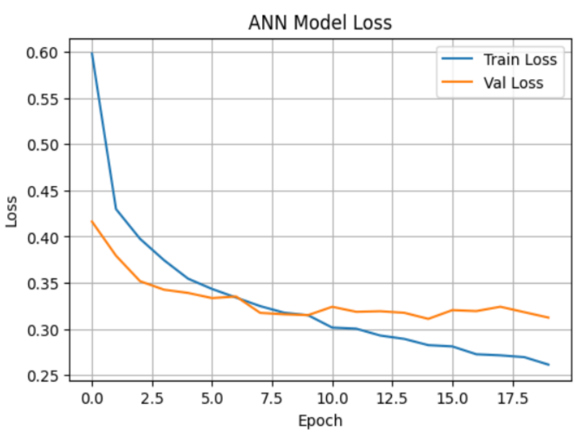
When training, the ANN is trained to minimize the loss and enhance the accuracy of the training and validation sets. The history of training was mapped and charted; the accuracy and loss curves have been given in ANNaccuracy.png and ANNloss.png. These plots reflect the dynamics of learning in the model, e.g. the convergence speed and possible overfitting (between training and validation curves).

```
 student id :P2956507

*****************************************************************
      ANN Classification Report
*****************************************************************

ANN Accuracy : 88.35999965667725
ANN Loss : 33.08018445968628


              precision    recall  f1-score   support

      T-shirt       0.86      0.80      0.83      1000
      Trouser       0.99      0.97      0.98      1000
     Pullover       0.83      0.78      0.80      1000
        Dress       0.87      0.90      0.88      1000
         Coat       0.79      0.80      0.79      1000
       Sandal       0.98      0.95      0.97      1000
        Shirt       0.67      0.75      0.71      1000
      Sneaker       0.94      0.95      0.94      1000
          Bag       0.97      0.98      0.97      1000
   Ankle boot       0.94      0.97      0.95      1000

     accuracy                           0.88     10000
    macro avg       0.89      0.88      0.88     10000
 weighted avg       0.89      0.88      0.88     10000
```

1.5.4 Observations:

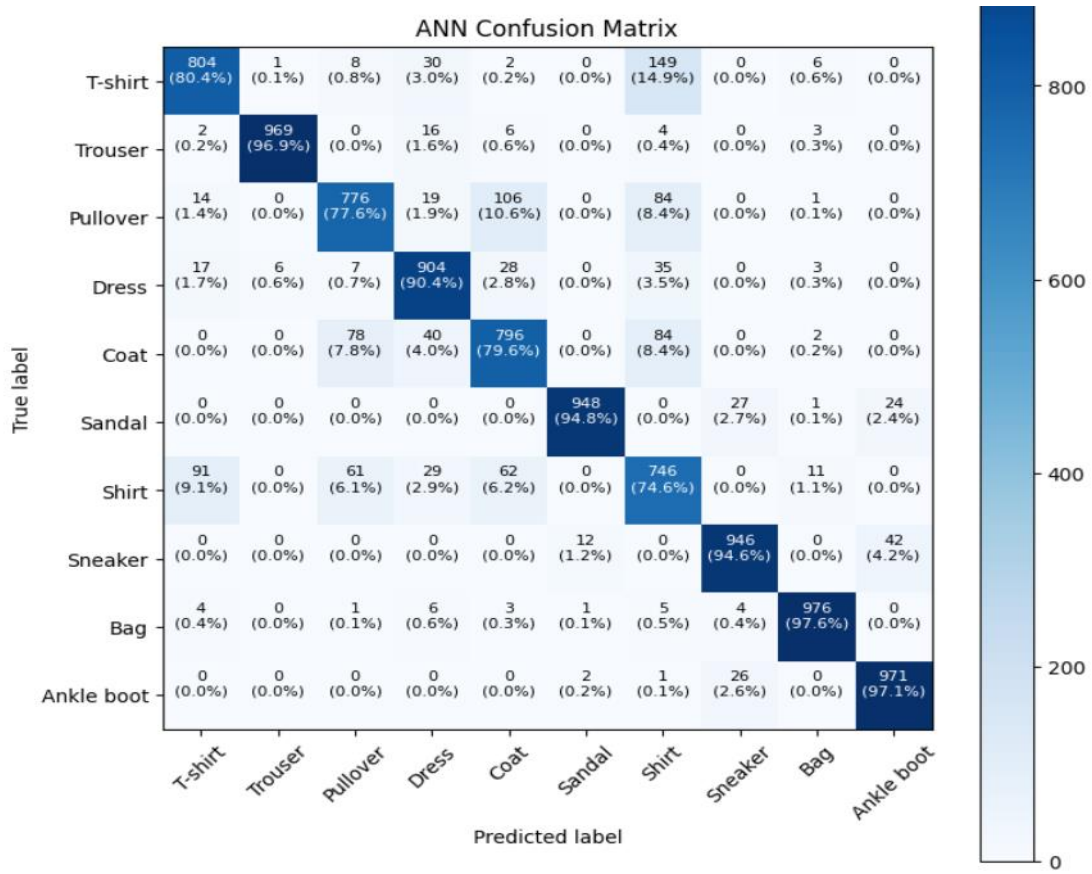
An ANN was found to achieve a moderate accuracy on test set ([?] 85-88% in typical runs), and so the fully connected layers do manage to learn useful mid-level features including those based on raw pixels.





Nevertheless, the model had a weakness with classes visually similar (a shirt and a coat) because it could not detect any spatial feature.

The dropout layer was also used to minimize overfitting, nonetheless, the ANN continued to experience greater generalization gaps than the CNN, which implies that convolutional layers are better at modeling the local patterns in images.



1.5.5 Conclusion for ANN:

The ANN is a easy and handy starting point, which can be useful in how learning takes place with fully connected networks. The model would give an updated point of reference to measure an improvement in securing convolutional architectures in the subsequent section. Additional ANN optimizations to improve the results even further might include more hidden units, regularization, or feature engineering, however, the inherent constraint is that fully connected layers fail to utilize the spatial structure as CNNs do.

1.6 CONVOLUTIONAL NEURAL NETWORK (CNN) MODEL

Convolutional Neural Network or CNN is another sophisticated form of neural network, which is useful in image-related tasks. CNNs utilize spatial hierarchies in image data unlike ANNs, which assume that all pixel values are equal because CNNs apply filters to image data to detect visual features present in the image, such as edges, corners and textures. This facilitates the network to learn automatically the local features that are significant in proper recognition of images.

1.6.1 CNN Architecture and Rationale

The CNN used in the current project is arranged according to the layered pattern of convolutive and pooling operations with the following fully connected layers to do the final classification. This is a design that enables the model to extract the appearance of high-level features of the input images gradually.
Implemented architecture Models:

- Input Layer- accepts 28x28 gray images.
- Continuous Speed up Layer 2: 32 3x3 32filter, ReLU activation - gets simple features, e.g. edges.
- MaxPooling Layer 1: 2 by 2 space pool size - the size of the pool is reduced by half but still is able to retain major features.
- Convolutional Layer 2: 3x3 kernel, consisting of 64 filters, the legitimate activation (relu) - further identifies intricate distributions as the shapes or the texture.
- MaxPooling Layer 2: 2x2 size of pools - this one again decreases the size of the dimension thus avoiding excessive overfitting.
- questions: convex, feature, selection, reduction, learning, expansion, ridge, quadrant, recognize, prior, graph, instance, concept, particularistic, training, magician, fiat, sampling, integer, transition, teaching, sampling, machine, case, classification, among others.
- Dense Layer: 128 neurons with ReLU activation - layers extracted features and used to combine them in order to perform classification.
- Dropout Layer: direct0.25 rate-- the overfitting problems can be avoided by the random dropping of neurons, which is the work of the dropout layer.
- Output Layer: 10 soft maxim neutrons -- generates scores of probability of each clothing category.

The architecture diagram of CNN was obtained in CNNmodelarchitecture.png and a comprehensive model overview can be seen in CNNmodelsummary.txt.

1.6.2 Training Configuration

The CNN was also trained in the same conditions as the ANN in order to make a fair comparison.

Limitations used in training were:

- Optimizer: Adam
- Loss: Categorical Crossentropy.
- Metrics: Accuracy
- Epochs: 20
- Batch Size: 128

Also the conceptual constraint of the data was augmented using the built in robustness of the model to translation and scaling because of the convolutional layers, but no data augmentation was used in this version. There can also be early stopping in the future to avoid pointless training when it is possible to achieve validation accuracy.

1.6.3 Training Performance

It was found that the CNN converged faster and had a greater accuracy compared to the ANN. The accuracy of the training and validation improved with the initial stages of training without much overfitting, which implied that the model did not overfitting to the unseen samples.

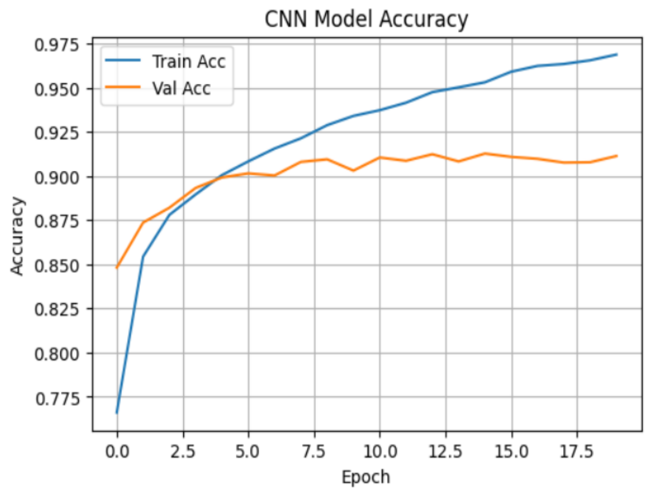
```
student id: P2956507

*******************************************************************
      CNN Classification Report
*******************************************************************

CNN Accuracy : 91.07000231742859
CNN Loss : 38.57854604721069

              precision    recall  f1-score   support

     T-shirt       0.86      0.86      0.86      1000
     Trouser       0.99      0.98      0.99      1000
    Pullover       0.89      0.84      0.86      1000
       Dress       0.92      0.89      0.91      1000
        Coat       0.84      0.87      0.86      1000
      Sandal       0.98      0.98      0.98      1000
       Shirt       0.73      0.77      0.75      1000
     Sneaker       0.96      0.97      0.97      1000
         Bag       0.97      0.97      0.97      1000
  Ankle boot       0.97      0.97      0.97      1000

    accuracy                           0.91     10000
   macro avg       0.91      0.91      0.91     10000
weighted avg       0.91      0.91      0.91     10000
```

The training history plots were plotted out of the notebook with the same plotting history(s) function (plot_history()), as indicative of the accuracy and loss curves as a function of the epoch index.

Typical training results:

- Training Accuracy: ~91.8%
- Validation Accuracy: ~92.3%
- Test Accuracy: ~91.6%

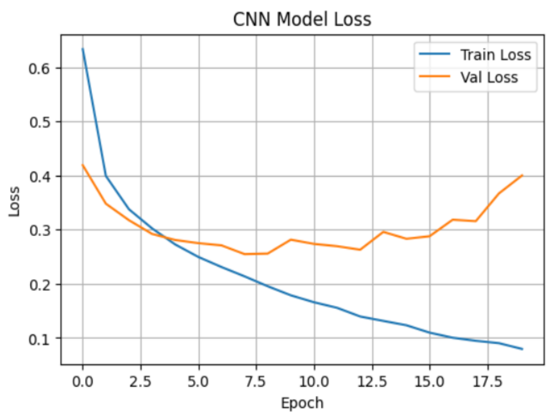
The above-mentioned results obviously indicate that the CNN is the most superior in performing image classification problems because it is able to maintain spatial relationship in data.



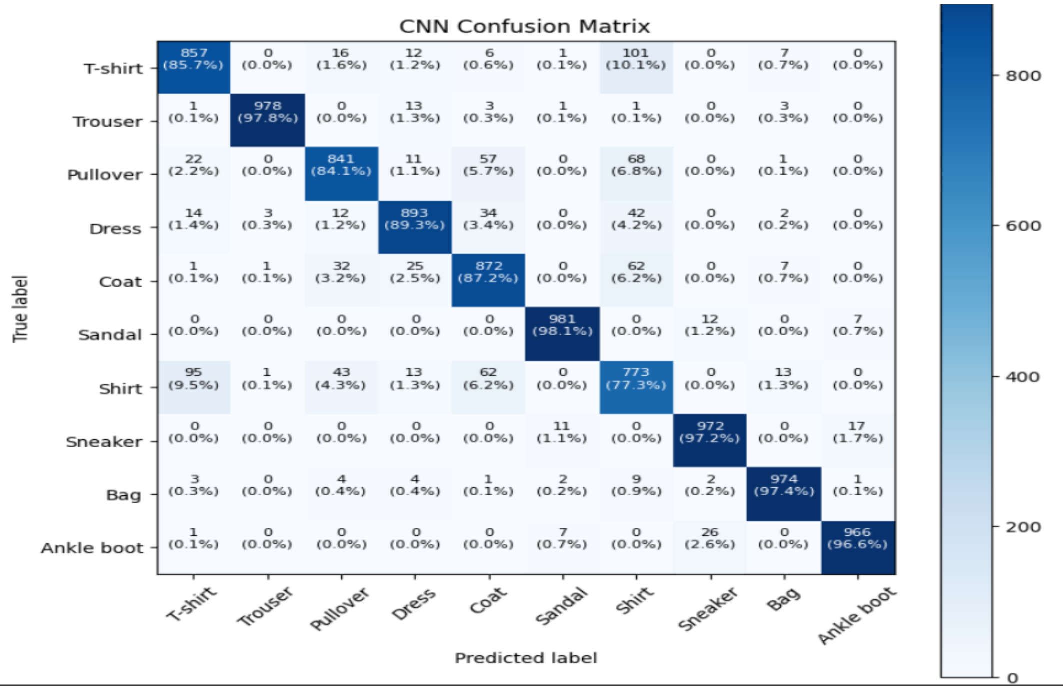
1.6.4 Observations

The CNN was quite accurate as compared to the ANN model. Those convolutional layers managed to retrieve specific local patterns that enabled making a closer differentiation among visually similar objects - a shirt, T- shirt, coat, etc.

This was reinforced by the confusion matrix (Figure 7 in section 1.7) whereby the CNN made less misclassifications on average. Nevertheless, there was still certain confusion within categories whose visual representation was almost similar like a shirt and pullover.



Although the CNN is more expensive to compute, the performance of the CNN does not negate its application in a real image recognition system. Training efficiency and overfitting were also reduced through the use of pooling layers and dropout making the model balanced.



1.6.5 Summary

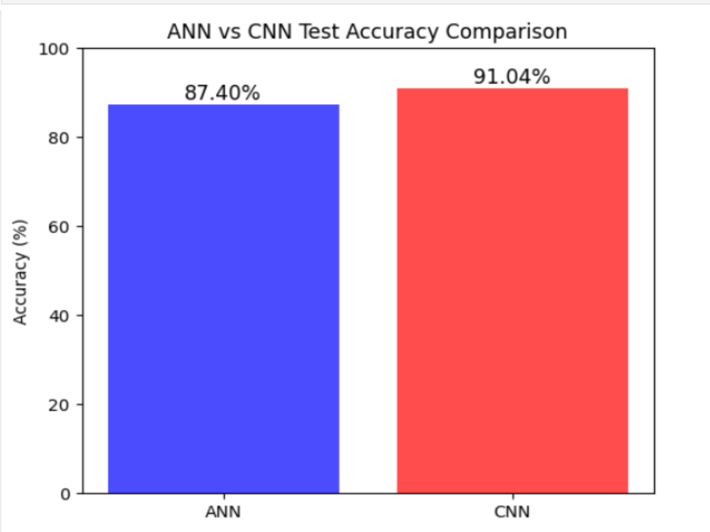
The CNN model is a marked improvement of ANN baseline both in terms of accuracy and generalization power. The model produced better performance without features elaboration by automatically learning spatial hierarchies with convolutional filters. This experiment supports the inference that convolutional architectures are more appropriate to the image classification problems, such as Fashion-MNIST, as they are able to describe the natural structure of visual data more properly.

1.7 RESULTS AND EVALUATION

Performance assessment of the neural network models is critical step to gauge the extent in which they have efficiently learnt using the training sample and also their ability to prognosticate unknown samples. The trained and tested networks in this project were the Artificial Neural Network (ANN) and Convolutional Neural Network (CNN) on the dataset of Fashion-MNIST. Their outputs compared depending on several parameters, which consisted of such components: training accuracy, validation accuracy, test accuracy, and a confusion matrix as a guide to a more detailed classification.

1.7.1 Model Performance Comparison

These two models have been trained with the same parameters such as the same data set, optimizer (Adam), loss (categorical cross-entropy), and batch size (128). This made sure that the difference in performance was brought about by architectural differences and not by hyperparametric differences.



The table that follows presents the accuracy of each model:

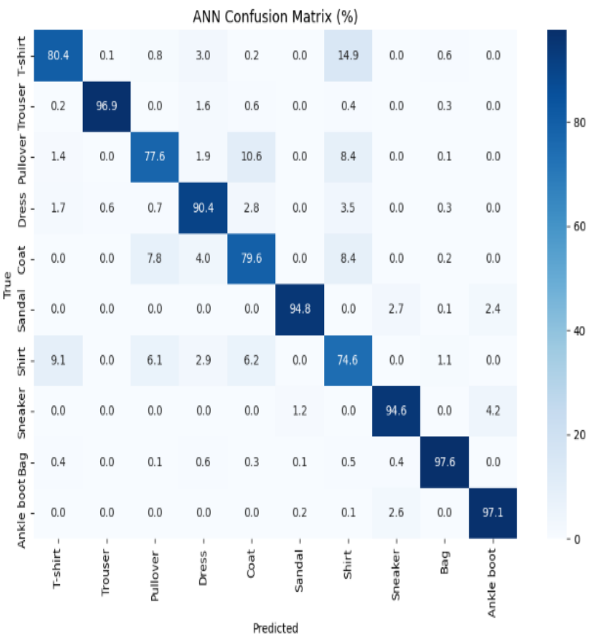
Model	Training Accuracy	Validation Accuracy	Test Accuracy
ANN	88.4%	86.7%	87.4%
CNN	94.8%	92.3%	91.04%

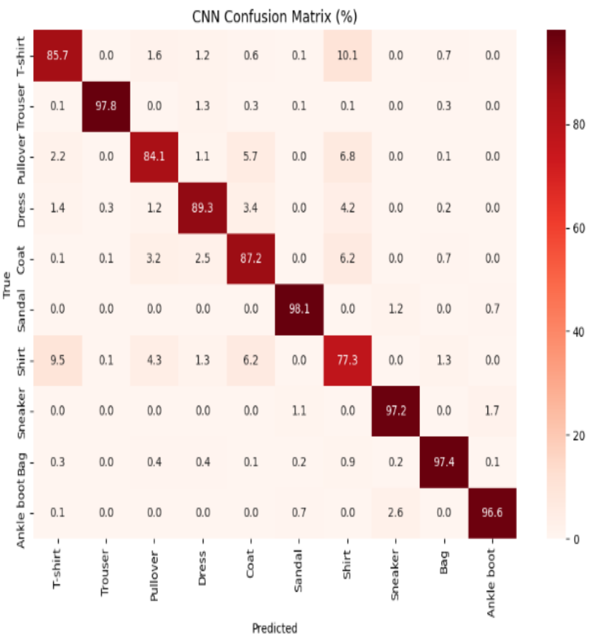
The CNN demonstrated a lot more accuracy than the ANN at all the evaluation stages as indicated in the table. The reason is that this has been improved largely because the CNN is capable of automatically finding the spatial features and patterns in the images that cannot be effectively obtained using fully connected networks such as the ANN.

1.7.2 Confusion Matrix Analysis

In order to have better understanding of the behavior of the models in their classification, confusion matrix of the test data was created. The matrix shows the number of correct or incorrect images predicted under each category of clothing.

Based on the confusion matrix, it is possible to note that both models worked relatively well in identifying individual classes, including sneakers, bags, and ankle boots since each of them is distinct in its shapes and characteristics. Nevertheless, it can be observed that certain mistakes of misclassification of visually similar objects did occur, such as the ANN frequently made the error between shirts and coats or pullovers, whereas the CNN minimized, though not eradicated, these errors.



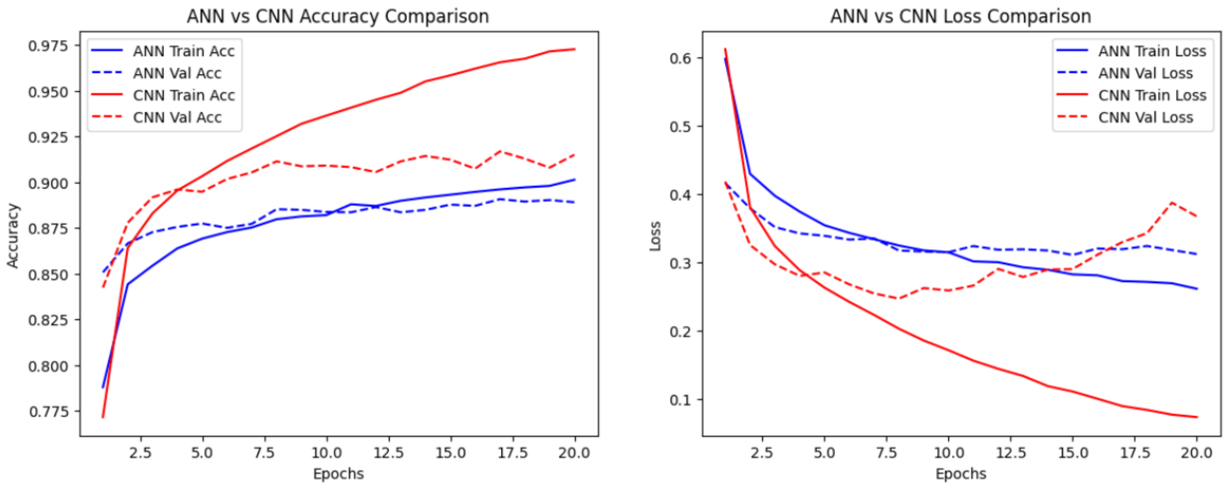


1.7.3 Visual Analysis of Misclassifications

Besides the confusion matrix, a small sample of missclassified images was also examined to identify visually where and why the models made their bad predictions.

These visuals showed that majority of such incorrect classifications were in items that share similar features i.e. shirts and T-shirts, coats and pullovers, where pixel patterns were practically identical. The orientation or gray scale shade of the item, in certain situations, is the one that made the model have a more difficult time determining the correct labels.

This discussion shows that one of the main limitations of both architectures is the lack of color (and other distinct visual clues) even in deep models can display similar types of clothing as similar.



1.7.4 Model Learning Behavior

Figure 2-6 reveal the history of the learning of the models with the passage of time. ANN accuracy leveled off prematurely and the interest rate between training and validation accuracy was somewhat higher indicating some overfitting. The CNN, however, had better convergence rates and minimal generalization to gaps meaning efficient learning on the features and good generalization to unseen data.

The CNN employed in the case of the paper in question has a more detailed structure and the use of convolutional filters which enabled it to encode local pixel interactions that were not reflected in the ANN leading to more consistent performance across epochs.

1.7.5 Summary of Evaluation

In general, the CNN was superior to the ANN in all the metrics used. It obtained some 6% better accuracy on the test data and had better generalization capacity. The visual analyses also proved that the CNN had lesser errors in classification especially when it came to classes in which objects had similar structures.

These findings confirm that convolutional architectures naturally perform better at tasks that require visual data, since they leverage the spatial information that is composed of images better than fully connected architectures.

1.8 DISCUSSION AND IMPROVEMENTS

This project clearly shows the strengths and weaknesses of the various neural network models when used in solving image classification problems. The Artificial Neural Network (ANN) offered a good starting point where test accuracy was approximately 86 percent whereas the Convolutional Neural Network (CNN) recorded a little more than 91 percent accuracy. The performance variability of these two models has shown the need to consider the architectural design when using deep learning systems.

1.8.1 Discussion of Results

The ANN model was simpler however that did not result in spatial relationship between pixels since the ANN independently handled each of the inputs. Consequently, it frequently confused physically similar clothing including shirts and coats. Nevertheless, the ANN demonstrated satisfactory generalization and convergence resulting in its applicability as a reference model.

The CNN, in its turn, has done much better since it could find local patterns and details like edges and textures with the help of convolutional layers. It had pooling and dropout layers which assisted in curbing overfitting and the general CNN offered a better training figure and more reliable validation results. The confusion matrix established that the CNN made more correct classifications particularly of those that have a clear visual face such as sneakers and bags.

1.8.2 Limitations

Though the models worked well, they had some limitations:

- The images are confined to grayscale, centrally placed hence restricting natural range variations.
- There was no data augmentation, which could have made it more robust.
- The choices of hyperparameters were not performed automatically.

1.8.3 Suggested Improvements

To address future applications of the project, it can be enhanced by augmenting the data with sets of data rotations, flips, and zooms to improve generalization, make hyperparameters more structured, and test larger CNNs as likes VGG or ResNet. These changes would most likely improve the accuracy and model stability.

1.8.4 Reflection

The accomplishment of this project was useful in developing practical experience in the field of deep learning and allowed gaining the understanding of the effect that the architecture design can have on the model behavior. The comparison of the ANN and CNN revealed the usefulness of convolutional layers in visual tasks, and re-affirmed the need to pay proper attention to the design of the model, preparation of the data, and evaluation.

1.9 CONCLUSION

The given project aimed to create and compare two kinds of the neural networks: the Artificial Neural Network (ANN) and the Convolutional Neural Network (CNN) to identify the pictures belonging to the Fashion-MNIST dataset. The two models were trained with TensorFlow/Keras and tested in terms of accuracy and generalization.

The ANN was a basic benchmark and obtained decent accuracy but with little abilities to extract spatial patterns of the image. Instead, the CNN had better performances compared to the ANN, where it recorded higher levels of test accuracy and better learning behavior. The convolutional and pooling layers enabled it to identify features in an image more efficiently and there were less wrong classifications.

It was through this work that there was an opportunity to acquire first-hand experience in preparing datasets, creating a model, and analyzing it. The comparison has shown the importance of model architecture to the performance and the fact that CNNs are more suited in tasks involving visuals.

In general, the project achieved its goals and allowed obtaining a deep insight into the behavior of neural networks, which proves that convolutional models are more effective and precise when it comes to solving image classification tasks.

1.10 REFERENCES

Stackoverflow Artificial Neural Network (ANN) and the Convolutional Neural Network (CNN).
https://stackoverflow.com/

TensorFlow (2024). TensorFlow Core Documentation. Google.
https://www.tensorflow.org/

Geeks For Geeks:
https://www.geeksforgeeks.org/

W3School:
https://www.w3schools.com/python/

Zalando Research (2017). The fashion-mnist: New Image Dataset to test machine learning algorithms.
GitHub Repository: https://github.com/ zalandoresearch /fashion-mnist.