COMP5318

Machine Learning and Data Mining

Assignment 2: Image classification

Team members: Group A2 109

Palak Kalra (SID : 520386039)
Reenal Raina Pereira (SID : 520112117)

Contents

Table of Contents

Introduction

Image processing has been an important area of investigation for many years, with its vast applications in various industries such as in medicine, biology, engineering etc. There has been an eminent integration of it with Machine learning to solve complex problems. (Dolechek, Cho,2022). It comprises the categorization of data and assigning labels to a group of pixels to identify to which the label belongs.

The primary objective of our model comparison and selection is better performance of the machine learning solution and also the time taken to train that model. This report explains and elaborates about 3 major image classification models used on a prominent dataset of Zalando Research's Fashion-MNIST. Support Vector Machine, Multi-layer Perceptron and a Convolutional Neural Network were designed and built, and the results were compared to get the best accuracy on the dataset. Among the models built, CNN gave the most accuracy on the dataset with 92.76% on the test set when compared to other classifiers; 89% with a Support Vector Machine and 89.07% with a fully connected neural network. Furthermore, Team compared the accuracy and training time with other classification models. The goals were successfully achieved, and, in some classifiers, it outperformed CNN.

Data

1. Data description and data exploration:

Fashion-MNIST is a dataset of Zalando's article images which consists of 60,000 samples of training and 10,000 samples of testing data of fashion and clothing items. These are examples associated with 10 different categories, such as shirts and dresses from the Zalando e-commerce website (Xiao, Rasul, Vollgraf (2017)). Each image is 28 pixels in height and 28 pixels in width in both the training and testing dataset, for a total of 784 pixels in total. Zalando Research data is commonly used as a tool to learn various machine learning algorithms. The pixel values lie between 0-255. The lightness and darkness of a pixel are connected with the number with which it is associated, which means the higher the number, the more darkness. It is used as a benchmark for a lot of algorithms in machine learning and is worldwide accepted by the community for testing and developing algorithms.

Labels**:** Each training and test example is assigned to one of the following labels:(Xiao, Rasul, Vollgraf (2017))

Label	Description
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

In data exploration, we checked out the data structure, and how the images appear in datasets. We did check the shape of the testing and training dataset and plotted images corresponding to their labels for a couple of images to get a better understanding of how the image looks like for all the labels. The figure below shows the various samples of data.

Figure 1: Sample data

2. Pre-processing:

Normalising data: Normalization of images is done to convert images into familiar pixels or in a familiar range so that computation cost is decreased and easy to process the data. Thus, to normalize the data, the pixel values are converted in the range of 0-1. The pixel value is stored as an 8-bit integer with 0 representing black and 1 as white. Therefore, in our scenario, we have divided each pixel cell by 255 so that they come in the range of 0-1 and are normalised to use forward in the algorithms we wish to train data on.("Data Exploration: Fashion-MNIST Data", (n.d)).

Converting dimensionality from 3D to 2D: Classifiers like SVM and Logistic Regression accept only 2D data. So as a part of the requirement for us to implement SVM on the training data, we have converted the dataset from 3D to 2D. Hence, the shape and size of the datasets - X_train_full, X_test were reshaped and reduced into 2D arrays. (Pramoditha,2022)

Splitting Training dataset into Training and Validation dataset: As the models are trained and tuned continuously several times with a combination of different

hyperparameters, the training dataset is split into training and validation datasets comprising of 10% training data. This helps in minimising overfitting and helps in selecting the best model from various models. (Pramoditha,2022)

Adding dimension to training data to use in CNN: Most of the standard CNNs expect a 4-dimensional array as input: (batch_size, width, height, n_channels) and according to this we have added an extra dimension to the end of the input array. The final dimension, n_channels, correlates to the number of values per image pixel. (Geron (2019))

Methods

1. Summary

We used 3 classifiers to classify data in the Fashion-MNIST dataset.

Support Vector Machine: SVM is a supervised machine learning algorithm that follows a hyperplane-based learning scheme and can be used both for regression and classification problems. (Ray,2015),(Chandra,Bedi,2018). During Classification, the training data is transformed into multidimensional space and a hyper-plane is constructed in a higher dimension. The goal of the SVM is to create an optimal hyperplane with minimum error % and find the maximum marginal hyperplane that divides the input dataset into various labels or classes. In our project, we have used Radial Basis Function (RBF) kernel to find perfect linear separation between the various classes in the high dimensional features space. (Sreenivasa,2020)

Fully connected neural network: A neural network is a system of interconnection between artificial "neurons" to exchange messages with each other. The connections are associated with numeric weights. Every network consists of multiple layers and these layers contain numerous "neurons". These respond to different combinations of inputs from the previous layers. (Hijazi, Kumar, Rowen, (n.d.)) FCNN consists of a series of fully connected layers which connect each neuron in one layer to each neuron in the other layer. (Mahajan,2020). It is a feed-forward neural network that has interconnected neurons in three layers i.e input layer, an output layer and one or more hidden layers. The information from the outside is fed into the algorithm through the input layer and from there, it's further passed to the hidden layer and the output layer gives the results from the processing of the algorithm. ("What is a Neural Network?",(n.d)). Each of these dense layers maintains the weight (and bias) matrix for the connections between the neurons and their inputs. (Geron (2019))

Convolutional neural network (CNN): Convolutional neural network is a type of deep learning model for processing image data efficiently to give accurate results in the form of predictions. These are designed to extract and learn from the special hierarchies of features of the images provided in the input layer adaptively and automatically. Three main types of layers are used to build CNN architecture: Convolutional Layer, Pooling Layer, and Fully Connected Layer. (Mahajan,2020). The Convolution layer and Pooling layers are the basically ones who are responsible for feature extractions while the third layer maps those

extracted features with the final output i.e the classification. (Yamashita, Nishio, Do, Togashi,(2018))

2. Strengths and weaknesses

SVM: It works relatively well when there is a clear margin of separation between classes. It is more effective in high-dimensional spaces and in cases where the number of dimensions is greater than the number of samples. SVMs are resistant to overfitting because of the regularization of the C parameter. Careful tuning of the C parameter helps in prevent overfitting in SVM even in cases where the number of observations is less than the number of attributes. ("SVM, Overfitting, the curse of dimensionality",2012).

SVM does not perform very well when the data set has more noise i.e. target classes are overlapping. It also does not perform well when we have large datasets because the training time is higher. As the support vector classifier works by putting data points, above and below the classifying hyperplane there is no probabilistic explanation for the classification. (K,2019)

Fully Connected Neural Network: Neural networks are more capable of performing tasks that are complex when compared to other algorithms as they are built in the same structure as that of a human brain. Thus, they lead to effective visual analysis. The time that goes into training these networks is comparatively less as they are more adaptable. One of the strengths and weaknesses of fully connected neural networks is that they are "structure agnostic" which means that we do not need to make any prior assumptions about the input while training it. However, such networks tend to have weaker performance than networks that are structured for a special purpose. transform to new environments quickly. (Mahajan,2020)(Rawat,2022)

CNN: This model is the best suitable network used for image classification. Both these terms are used interchangeably in deep learning. One of the strengths of the CNN is its weight-sharing feature which lowers the number of parameters for trainable network and in turn avoid overfitting.
The weakness of CNN architectures is that they already presume that input is images which makes us encode certain properties in the model architecture. Moreover, CNN's require a large amount of training data to train the network effectively and they have a hard time encoding and classifying images that have different positions. (Rawat,2022)(Alzubaidi,et al.(2021))

3. Architecture and hyperparameters

SVM: For SVM, we have chosen the basic algorithm with the popular RBF kernel for the classifier and Research to find the optimal hyperparameter for the SVM classifier. Since the value of C and gamma, affects the performance of the SVM, we performed hyper-parameter tuning on these two parameters.

C parameter adds a penalty for each misclassified data point. If C is small, the penalty for misclassified points is low so a decision boundary with a large margin is chosen at the expense of a greater number of misclassifications. If C is large, SVM tries to minimize the number of misclassified examples due to a high penalty which results in a decision boundary with a smaller margin.

C parameters are regularised to mitigate overfitting. It also controls the balance in trading off between attaining a low training error and a low testing error. The main goal of the C parameter is to set a larger margin and minimize the misclassification rate. ("RBF SVM parameters" (n.d))

The **Gamma** parameter controls the distance of influence of a single training point. Low values of gamma indicate a large similarity radius which results in more points being grouped. For high values of gamma, the points need to be very close to each other to be considered in the same group (or class). Therefore, models with very large gamma values tend to overfit. (Yıldırım,2020)

In this scenario, we have passed the grid of parameters to the evaluation function in our case Grid Search CV as a dictionary, where the keys are the parameters and different values are the settings to be tested. We have created a separate function for hyperparameter tuning for SVM in which with the help of the grid search cv method and cross-validation value 2, the function returns the best parameters for C and gamma ie; 10 and 0.01 respectively.

The hyperparameter tuning of SVM took the longest with 4.37 hrs to be precise for 3 values of C and 3 values of gamma. With these best parameters, we trained the model and achieved an accuracy of 89%.

FCNN: For a fully connected neural network, our team chose an architecture with 2 hidden layers. The first layer simply flattens our input for the next layer. Here, we also define the size of the input, which will determine the number of weights between the input layer and the first hidden layer. Each of the dense layers maintains the weight (and bias) matrix for the connections between the neurons and their inputs. The final layer is determined by the nature of our classification problem. We have ten classes, so we require 10 output neurons, and we utilise the SoftMax function to convert the raw outputs of this layer into a probability distribution over the classes. (Geron,2019)

.

ReLu activation function: One of the commonly used non-linear activation functions in the deep learning domain is the ReLu function. ReLu stands for Rectified Linear unit function. The primary advantage of this function when compared to other functions is: it doesn't activate all the neurons at the same time. If the output of linear transformation is less than 0 only then neurons will be deactivated.

Softmax activation function: The mathematical function that converts a vector of numbers into a vector of probabilities, where the probabilities of each value are equivalent to the relative scale of each value in the vector. Specifically, the network is configured to output N values, to every class in the classification task, the SoftMax function normalizes the outputs,

converting them from weighted sum values into probabilities of that sum to one to obtain the weights. Each value in the output of the softmax function is interpreted as the probability of membership for each class. (Brownlee(2022))

While training the model, we need to pass the loss function, for that, we have used sparse categorical cross entropy for a multi-class classification model where the output label is assigned an integer value (0, 1, 2, 3…). (Kumar,2020)

An optimizer is a function or an algorithm that modifies the attributes of the neural network, such as weights and learning rates. Thus, it helps in reducing the overall loss and improving accuracy. We have chosen Stochastic Gradient Descent (SGD) as an optimiser and it tries to update the model's parameters more frequently. In this, the model parameters are altered after the computation of loss on each training example. (Doshi,2019)

As a part of hyperparameter tuning for FCNN, we have applied grid search CV on several values of learning rate and activation function. In the activation function parameter grid, we have passed three activation functions with one being when no activation function is passed, then by default the value of the activation function which is "Linear" is used, and the other two being Relu and Sigmoid.

After tuning hyperparameters the best parameters for the FCNN model are:
Activation function: ReLu and optimized learning rate would be: 0.1
The hyperparameter tuning took approximately 11 mins to tune these parameters for 3 different values of learning rate and activation function. With the best parameters, we trained the model and obtained an accuracy of 88%.

CNN: We have used two convolutional and two pooling layers in our model. Multiple sets of weights are reused for every neighbourhood of pixels to detect different features across the image. These sets of weights are called kernels or filters, and after training, they will learn to pick up different features, such as lines and curves in the first convolutional layer. The next convolutional layer performs the same process but will assemble the information about these low-level features to detect higher-level features such as shapes. (Geron (2019))

The convolution layer involves Mathematical calculation between input data (ex: image) and a specified size filter. The dot product of the filter and input image is obtained by moving the filter over the input image considering the size of the filter. (Gurucharan,2022)

Pooling layers reduce the number of parameters in the network, these layers down sample dimensions by taking the max value over input windows defined by pool size. (Geron (2019))

To avoid overfitting, we have added 1 more layer of dropout to be used to defeat this issue where a couple of neurons are eliminated from the network while preparing the measure bringing about decreased model size. We have passed a dropout value of 0.5, 50% of nodes are exited haphazardly from network organization. (Meghakumar (2021))

Our model is inspired by LeNet architecture, which is quite ancient in terms of neural networks and is inspired by the way the brain processes optical information. The series of convolutions and pooling process an image to extract more and more adequate features. (Rupp,2020). As a part of this assignment, since we had to run and train the models on our

hardware, we chose Lenet because it consumes less time to train and has less computational expense and also gives effective results.

The learning rate is one of the major hyperparameters when it comes to the configuration of neural networks. It manages the change in the model w.r.t the approximate error every time the model weighs. Picking up the right learning rate is an important task because if the value is too small then it will have major effects on the training process and can run for hours. If the learning rate is excessively large then the training process can be unstable. (Brownlee,2019)

The stride defines the step size of the kernel when traversing the image. While its default is usually 1.

In hyper-parameter tuning, we have taken into account stride, kernel and activation function.

As mentioned earlier for a fully connected neural network, we have used the GridSearch CV, we have used the same method to tune our parameters for CNN and find the best one to train at the end. We have passed an epoch value of 20. In the epoch, we train the neural network data with all the training data for a single cycle. (Baeldung (2022)) With the help of the grid search cv method and cross-validation value 2, the function returns the best parameters for kernel size : (3,3), learning rate as 0.1, and strides as (1,1).

After tuning hyperparameters the best parameters for the CNN model are: kernel size : (3,3) and optimized learning rate would be: 0.1and strides : (1,1). With the best parameters, we trained the model and obtained an accuracy of 92.76% and it took around 15 mins to train the model with 20 epochs.

Results and Discussion

Confusion matrix: As the name says, it is the matrix that displays how our classification model is confused while making predictions. The matrix will give the summary of all the information on how well the model has succeeded in predicting correctly and what errors or wrong predictions it has made.

SVM :
After tuning hyperparameters the best parameters for the SVM model are C: 10 and gamma: 0.01.

From Figure 4: Label Bag and Sandal have outperformed with great predictions. While Coat and Shirt have underperformed. The coat has confused with the Shirt and pullover and the Shirt has a poorly performed T-shirt/top and coat.

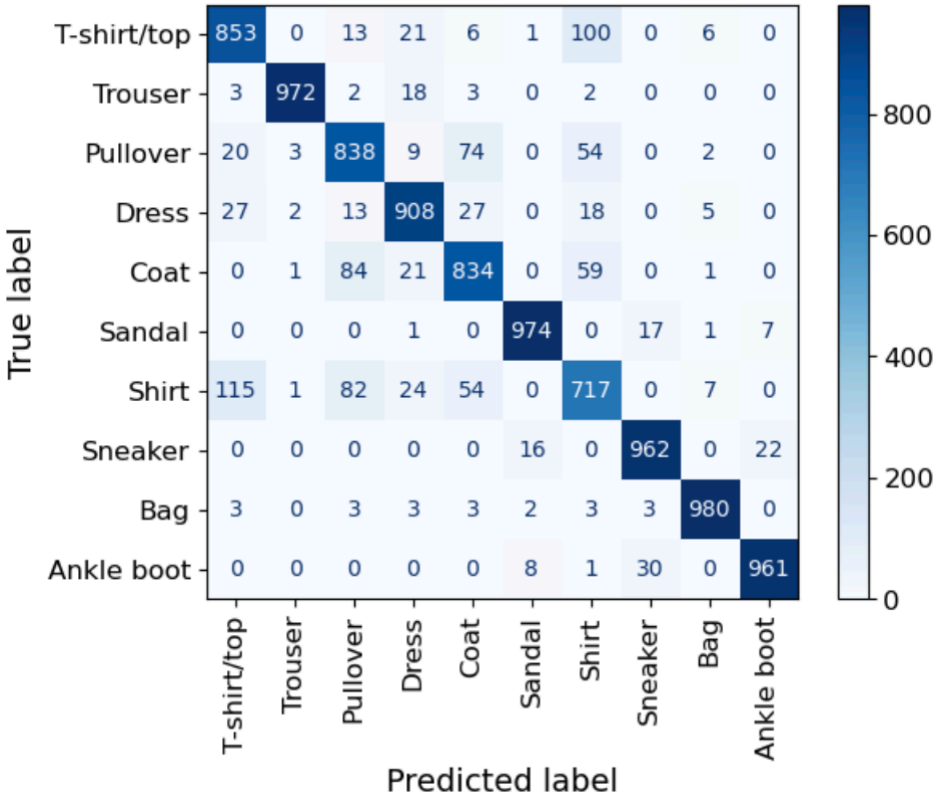

Figure 4: Confusion matrix for SVM.

The accuracy for the SVM classifier is 90%. (Figure 5) It can be seen that f1-score for 6 labels i.e trousers, Dress, sandals, sneakers, bag, the ankle boot is above 90% and mostly between 95-100%, that our model has predicted those labels precisely. And the label having the least precision is a shirt, 100 samples of that have been predicted as Tshirt/Top.

	precision	recall	f1-score	support
0	0.84	0.85	0.84	1000
1	0.99	0.97	0.98	1000
2	0.81	0.84	0.82	1000
3	0.90	0.91	0.91	1000
4	0.83	0.83	0.83	1000
5	0.97	0.97	0.97	1000
6	0.75	0.72	0.73	1000
7	0.95	0.96	0.96	1000
8	0.98	0.98	0.98	1000
9	0.97	0.96	0.97	1000
accuracy			0.90	10000
macro avg	0.90	0.90	0.90	10000
weighted avg	0.90	0.90	0.90	10000

Figure 5: Classification report for SVM

FCNN:
Label Bag and Trouser has outperformed with great predictions. While Pullover and Shirt has underperformed.
Pullover has confused with Shirt and coat and Shirt has poorly performed Tshirt/top and pullover.

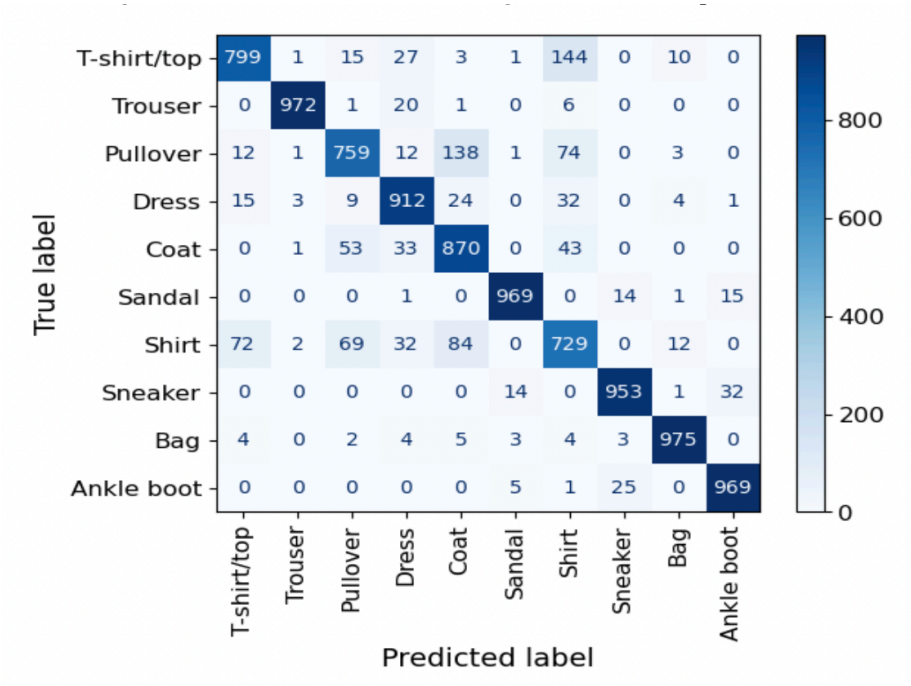

Figure 6: Confusion matrix for fully connected neural network

The accuracy for the FCNN classifier is 89%. (Figure 7) It can be seen that, out of 10 labels, the f1-score of the 6 labels is around .97 and .98 which means it was successfully able to predict those label images easily with fewer errors. Out of these 6, Label Bag and Trouser have outperformed with great predictions. While Shirt and Pullover have underperformed with f1- scores of 0.72 and 0.80. Pullover has mostly been confused with a Shirt and the Shirt has poorly performed by confusing it with a T-shirt/top and pullover.

	precision	recall	f1-score	support
0	0.89	0.80	0.84	1000
1	0.99	0.97	0.98	1000
2	0.84	0.76	0.80	1000
3	0.88	0.91	0.89	1000
4	0.77	0.87	0.82	1000
5	0.98	0.97	0.97	1000
6	0.71	0.73	0.72	1000
7	0.96	0.95	0.96	1000
8	0.97	0.97	0.97	1000
9	0.95	0.97	0.96	1000
accuracy			0.89	10000
macro avg	0.89	0.89	0.89	10000
weighted avg	0.89	0.89	0.89	10000

Figure 7: Classification report for fully connected neural network

In the graph shown below, it could be seen that while tuning the hyperparameters for a fully connected neural network, as the value of the learning rate increases, so the mean score also increases. Although, the activation function does play a role as well in the mean accuracy, we could see an effect of the learning rate is directly proportional to the mean score.

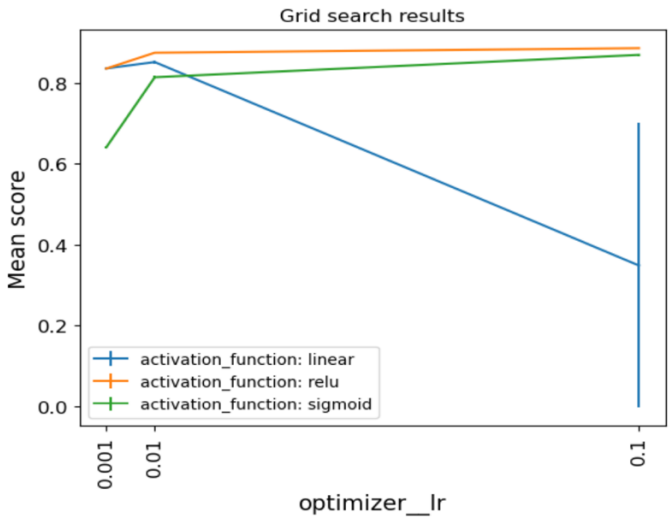

Figure 8: Grid search results of hyperparameter tuning of FCNN

The model accuracy is depicted in the graph below and we can notice that after 10 epochs, validation set accuracy decreases might be due to overfitting of the training data. But the training accuracy keeps on increasing as we increase the number of epochs.

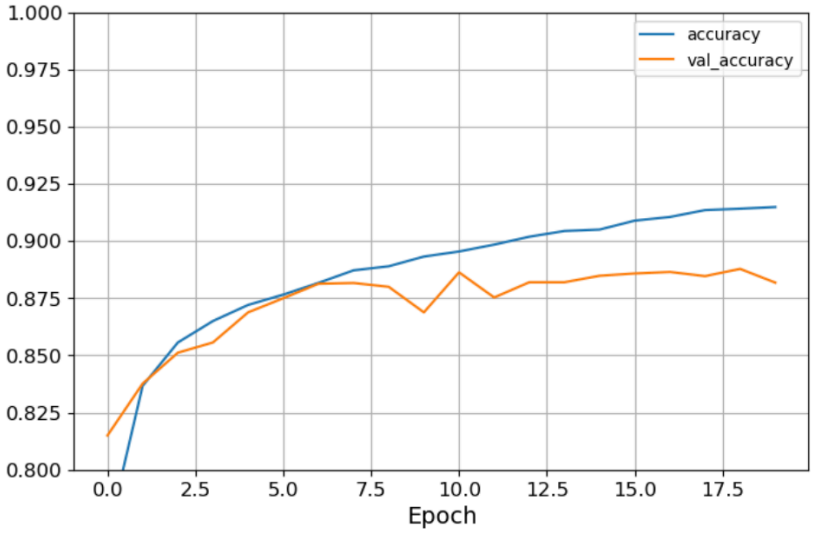

Figure 9: Training model accuracy at each epoch

CNN: From Figure 10: Label Sneaker and Trouser have outperformed with great predictions. While Shirt and Pullover have underperformed.
The shirt has poorly performed by confusing it with a T-shirt/top and pullover.

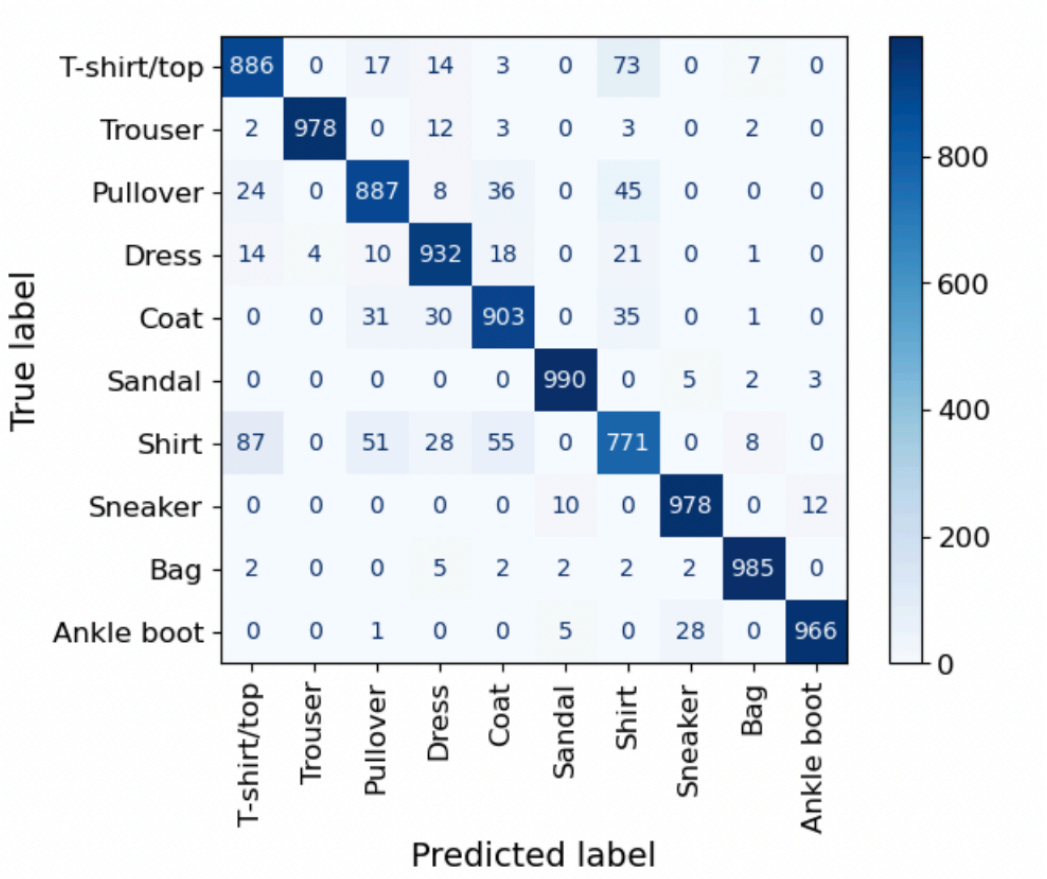

Figure 10: Confusion matrix for Convolutional neural network :

The accuracy for the CNN classifier is 91%. (Figure 11) By looking at the f1-score of the various labels, on 5 labels we score an f1-score of more than 97%,. While checking the other 5 labels, we could see the shirt label had a precision score of just 79% when compared to others.

	precision	recall	f1-score	support
0	0.87	0.89	0.88	1000
1	1.00	0.98	0.99	1000
2	0.89	0.89	0.89	1000
3	0.91	0.93	0.92	1000
4	0.89	0.90	0.89	1000
5	0.98	0.99	0.99	1000
6	0.81	0.77	0.79	1000
7	0.97	0.98	0.97	1000
8	0.98	0.98	0.98	1000
9	0.98	0.97	0.98	1000
accuracy			0.93	10000
macro avg	0.93	0.93	0.93	10000
weighted avg	0.93	0.93	0.93	10000

Figure 11: Classification report for Convolutional neural network

It can be observed from the graph below that how various combinations of kernel size, strides and learning rate affect the mean score of the grid cv results. Again, as we increase the value of the learning rate, the mean score increases.

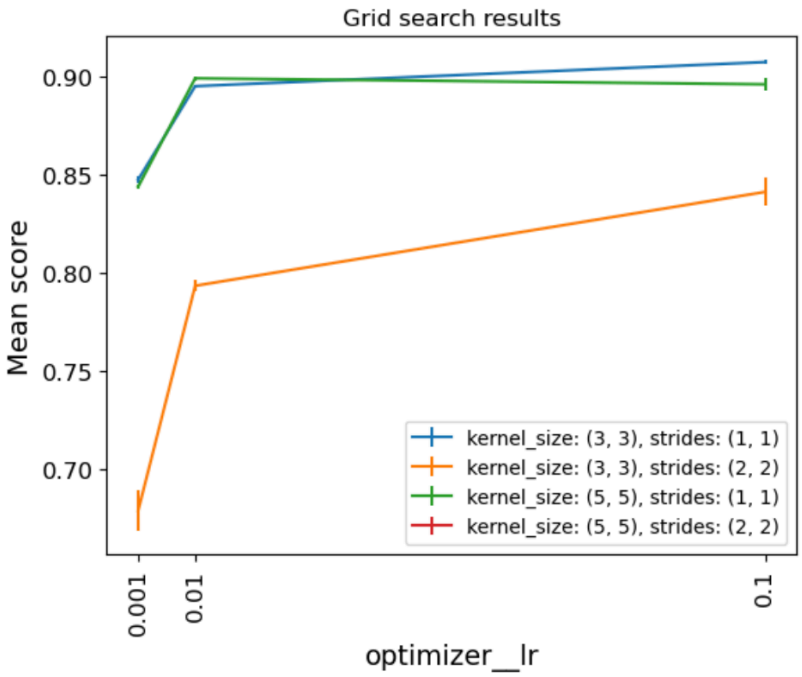

Figure 12: Hyperparameter tuning score for CNN

As shown in figure 13, The model accuracy of the training model with the best hyperparameters is depicted in the graph below and it can be observed that as the training accuracy increases, validation accuracy also acts similarly. At the 12th epoch, it could be seen that validation accuracy saw a dip. But as the number of epochs increases, we observed a slight difference in validation set accuracy. Thus, the number of epochs and batch size changes affect the training and thus accuracy of the model.

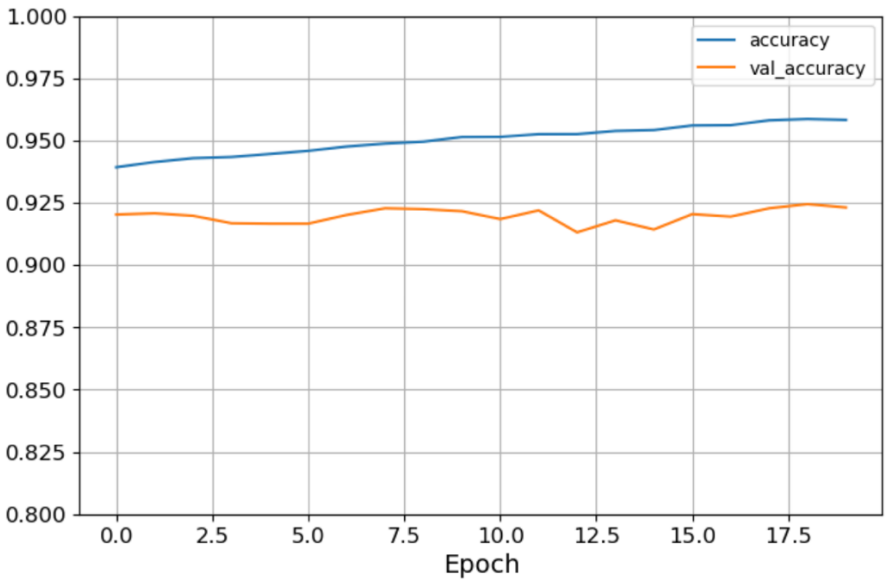

Figure 13: Grid search results of hyperparameter tuning of FCNN

Conclusions

Summary

Algorithm	Hyperparameters	Train time	Test time	Accuracy
SVM	C : 10, gamma : 0.01 kernel= 'rbf'	4min 3s	1min 8s	90%
FCNN	Activation function : ReLu, optimized learning rate : 0.1 Number of neurons = 200 epochs=20	20.7 s	659 ms	89%
CNN	kernel size : (3,3) , optimized learning rate : 0.1, strides : (1,1) Activation function ='relu' , 2 convd layers, 2 pooling layer, 1 dropout layer. 1 fully connected layer	15min 47s	4.43s	92.76%

As we can see in the summary above, we can say that CNN has performed better among the 3 models and there could be several reasons; it takes more hyperparameters when compared to the other 2 models. This can aid in better tuning and accuracy can be increased. It normally takes a longer time to train a model like CNN but with the help of fine hyperparameters computation, can be completed in 20 epochs.

From confusion matrices and evaluation models, we have received quite convincing accuracies for the models implemented. However, if we observe in the previous section, the confusion matrix has displayed some incorrect predictions for the Class 6 - Shirt, Class 2 - pullover and Class 4 - coat categories. One of the primary reasons could be, the pictures under these categories have a lot of similarities.

Although SVM consumes minimal energy to implement and compute among the 3 models with reasonably good accuracy, CNN gives the maximum accuracy. This would of course consume more computing requirements and higher training time. Apart from this FCNN consumes lesser time to train and test with lower accuracy when compared to SVM.

As per this report, the CNN model would always be a good option to pick in terms of precision because it will give us the freedom to use this model to implement on a complex dataset. Although it will use more parameters but with the considerate amount of training and testing time, it can provide high accuracy. Choosing the right kernel and hyperparameters for the model can be tricky. But considering the timing, a model takes to train the algorithm, SVM standouts with 90% accuracy by taking training time 1/3rd that of CNN.

For CNN, by increasing the training iteration; epoch =50 or 100, better accuracy can be achieved. Furthermore, a feature of the ensemble is recommended to observe for improvements in various models. A specialised processor for graphics and GPU could be

used. So far, we have observed and voted CNN as the best model among the 3 models. To improvise this in the future high-level model architectures like VGG16, higher versions of LeNet can be used. This would definitely require large hardware setup but in order to be cost-efficient some virtual components can be used.

Reflections

Through this assignment, we as a team learnt about 3 different algorithms i.e SVM, fully connected neural network and Convolutional neural network. We used various metrics to measure the accuracy, training time and testing time. We get to research different areas like hyperparameter tuning for each of these, do observations on the patterns of how accuracy changes when the parameters change and overall how exactly we implement them on a real dataset. During our course of work, we went through a lot of resources to understand and do better implementation in terms of the various aspects of results visualisation and explore about the dataset. Moreover, we feel when we started working on it, we were less aware of the fact what goes into each of these algorithms and what is the purpose and how internally it works, but after the exploration, and implementation of each and every step, we obtained a clear understanding of what goes into machine learning algorithms.

References

1. Bansal S.,(2021) *What is Classification Algorithm in Machine Learning? With Examples* Retrieved from: https://www.analytixlabs.co.in/blog/classification-in-machine-learning/#:~:text=The%20aim%20of%20classification%20is,can%20be%20numerical%20or%20categorical.
2. Ghosh S., (2022) *How to Compare Machine Learning Models and Algorithms* Retrieved from: https://neptune.ai/blog/how-to-compare-machine-learning-models-and-algorithms#:~:text=The%20primary%20objective%20of%20model,data%20and%20the%20business%20requirements.
3. Ray S., (2015) *Understanding Support Vector Machine(SVM) algorithm from examples (along with code)* Retrieved from: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/#:~:text=%E2%80%9CSupport%20Vector%20Machine%E2%80%9D%20(SVM,mostly%20used%20in%20classification%20problems.
4. Chandra M. A., Bedi S. S., (2018) *Survey on SVM and their application in image classification* Retrieved from https://link.springer.com/article/10.1007/s41870-017-0080-1
5. Ramsundar B.,& Zadeh R., (2018). *Tensorflow for deep learning: from linear regression to reinforcement learning*
6. Al-Masri A., (2019) *How Does Back-Propagation in Artificial Neural Networks Work?* Retrieved from: https://towardsdatascience.com/how-does-back-propagation-in-artificial-neural-networks-work-c7cad873ea7#:~:text=Back%2Dpropagation%20is%20the%20essence,reliable%20by%20increasing%20its%20generalization.
7. Samer Hijazi, Rishi Kumar, and Chris Rowen, IP Group, Cadence. (n.d.) *Using Convolutional Neural Networks for Image Recognition.* Retrieved from http://site.eet-china.com/webinar/pdf/Cadence_0425_webinar_WP.pdf
8. Mahajan P., (2020) *Fully Connected vs Convolutional Neural Networks* Retrieved from: https://medium.com/swlh/fully-connected-vs-convolutional-neural-networks-813ca7bc6ee5
9. Dolechek G. J., Cho N., (2022) *Advances in image processing using machine learning techniques* Retrieved from : https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/sil2.12146
10. Xiao H., Rasul K., Vollgraf R. (2017). Retrieved from https://github.com/zalandoresearch/fashion-mnist
11. *Data Exploration: Fashion-MNIST Data* (n.d) Retrieved from https://dax-cdn.cdn.appdomain.cloud/dax-fashion-mnist/1.0.2/data-preview/Part%201%20-%20Data%20Exploration.html
12. Pramoditha R., (2022) *Why Do We Need a Validation Set in Addition to Training and Test Sets?* Retrieved from : https://towardsdatascience.com/why-do-we-need-a-validation-set-in-addition-to-training-and-test-sets-5cf4a65550e0#:~:text=This%20is%20because%20you%20need,generalize%20on%20new%20unseen%20data.
13. Aurelien Geron (2019). *Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow, O'Reilly.*

14. Sreenivasa S., (2020) Radial Basis Function (RBF) Kernel: The Go-To Kernel Retrieved from: https://towardsdatascience.com/radial-basis-function-rbf-kernel-the-go-to-kernel-acf0d22c798a#:~:text=RBF%20Kernel%20is%20popular%20because,and%20not%20the%20entire%20dataset.
15. *RBF SVM parameters* (n.d) Retrieved from: https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html#:~:text=The%20C%20parameter%20trades%20off,classifying%20all%20training%20points%20correctly.
16. *What is a Neural Network?* (n.d) Retrieved from: https://aws.amazon.com/what-is/neural-network/#:~:text=A%20neural%20network%20is%20a,that%20resembles%20the%20human%20brain
17. Brownlee J., (2020) *Softmax Activation Function with Python* Retrieved from: https://machinelearningmastery.com/softmax-activation-function-with-python/
18. Yamashita R., Nishio M., Do R. K. G., Togashi K.,(2018) *Convolutional neural networks: an overview and application in radiology* Retrieved from : https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9
19. *SVM, Overfitting, curse of dimensionality* (2012) Retrieved from : https://stats.stackexchange.com/questions/35276/svm-overfitting-curse-of-dimensionality
20. K D., (2019) *Top 4 advantages and disadvantages of Support Vector Machine or SVM* Retrieved from: https://dhirajkumarblog.medium.com/top-4-advantages-and-disadvantages-of-support-vector-machine-or-svm-a3c06a2b107
21. Rawat S.,(2022) *Advantages and Disadvantages of Neural Networks* Retrieved from : https://www.analyticssteps.com/blogs/advantages-and-disadvantages-neural-networks
22. Alzubaidi L. S., Zhang J.,Humaidi A. J.,Dujaili A. A., (2021) *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions* Retrieved from : https://www.researchgate.net/publication/350527503_Review_of_deep_learning_concepts_CNN_architectures_challenges_applications_future_directions
23. Yıldırım S., (2020) *Hyperparameter Tuning for Support Vector Machines — C and Gamma Parameters* Retrieved from : https://towardsdatascience.com/hyperparameter-tuning-for-support-vector-machines-c-and-gamma-parameters-6a5097416167
24. Kumar A., (2020) *Keras – Categorical Cross Entropy Loss Function* Retrieved from : https://vitalflux.com/keras-categorical-cross-entropy-loss-function/
25. Meghakumar (2021) *CNN MODEL ARCHITECTURES WITH THEIR STRENGTHS AND WEAKNESSES* Retrieved from: https://meghakumar245.medium.com/cnn-model-architectures-with-their-strengths-and-weaknesses-4cf81fc49cc2
26. Rupp M., (2020) *Artificial Intelligence - Anatomy of the LeNet-1 Convolutional Network and How It Can Be Used in Order to Classify RADAR Data* Retrieved from : https://www.skyradar.com/blog/anatomy-of-the-lenet-1-convolutional-network-and-how-it-can-be-used-in-order-to-classify-radar-data#:~:text=The%20LeNet%2D1%20architecture%2C%20which,more%20and%20more%20adequate%20features.
27. Baeldung (2022) *Epoch in Neural Networks* Retrieved from: https://www.baeldung.com/cs/epoch-neural-networks#:~:text=An%20epoch%20means%20training%20the,to%20train%20the%20neural%20network.
28. Brownlee J.,(2019) *Understand the Impact of Learning Rate on Neural Network Performance* Retrieved from: https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/

29. Gurucharan MK,(2022) *Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network* Retrieved from : https://www.upgrad.com/blog/basic-cnn-architecture/#:~:text=other%20advanced%20tasks.-,What%20is%20the%20architecture%20of%20CNN%3F,the%20main%20responsibility%20for%20computation