

Classification using ML methods and Neural Networks

by Ziyao Chen

Submission date: 18-Apr-2023 10:30AM (UTC+0100)

Submission ID: 202455014

File name:

40734_Ziyao_Chen_Classification_using_ML_methods_and_Neural_Networks_495066_1427670808.docx
(1.13M)

Word count: 3459

Character count: 19407

Classification using ML methods and Neural Networks
Ziyao Chen-2200438

Abstract

Classification using ML methods and Neural Networks refers to the process of utilizing machine learning algorithms and neural networks to categorize data into different classes or categories. This approach has become increasingly popular in recent years due to its ability to accurately and efficiently classify large datasets.

Machine learning methods, such as decision trees, support vector machines, and random forests, have been widely used for classification tasks. Neural networks, on the other hand, have gained popularity due to their ability to learn complex patterns and relationships in the data.

In this field, researchers have developed various techniques to improve the performance of classification algorithms, such as ensemble learning, transfer learning, and deep learning. Applications of classification using ML methods and neural networks are diverse, including image and speech recognition, natural language processing, fraud detection, and recommendation systems. These applications have transformed various industries, such as healthcare, finance, and e-commerce.

As the demand for accurate and efficient classification continues to grow, researchers will likely develop even more advanced techniques and algorithms to tackle increasingly complex datasets.

Contents

1. Introduction.....	- 1 -
1.1 MNIST Fashion dataset.....	- 1 -
1.2 Cifar-10 dataset.....	- 2 -
1.3 Random forest.....	- 2 -
1.4 BP neural network.....	- 3 -
2. Related work	- 4 -
2.1 Random forest	- 4 -
2.2 BP neural network	- 5 -
3. Experimental results and analysis	- 6 -
3.1 Random forest	- 6 -
3.1.1 Performance measurement method.....	- 6 -
3.1.2 Test set and training set partition	- 6 -
3.1.3 The performance indicators obtained under different parameters	- 6 -
3.2 BP neural network	- 9 -
3.2.1 Performance measurement method.....	- 9 -
3.2.2 Test set and training set partition	- 9 -
3.2.3 Network structure.....	- 9 -
3.2.4 The choice of the optimizer	- 10 -
3.2.5 The performance indicators obtained under different parameters	- 11 -
4. summarize	- 14 -
References.....	- 15 -

1. Introduction

1.1 MNIST Fashion dataset

The MNIST Fashion dataset is a collection of 70,000 grayscale images of clothing items, each with a resolution of 28 x 28 pixels. The dataset is commonly used as a benchmark for image classification tasks in the field of machine learning. As it shown in Fig1.

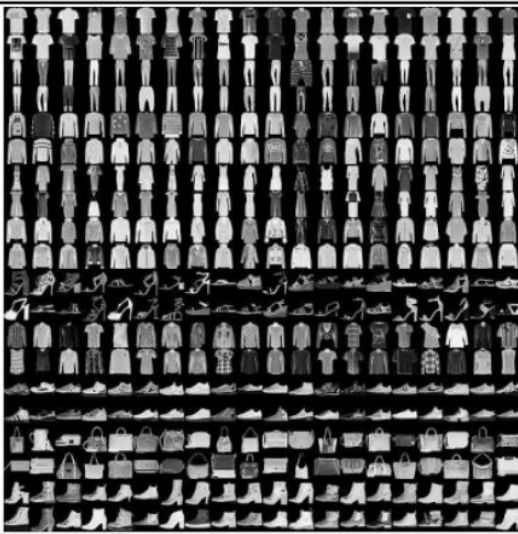
The dataset consists of ten categories of clothing items, including t-shirts, dresses, sneakers, sandals, and handbags, among others. Each image is labeled with a corresponding category, making it an ideal dataset for supervised learning.

The MNIST Fashion dataset is a popular choice for researchers and developers because of its simplicity and the fact that it provides a good starting point for experimenting with machine learning algorithms. It is often used as a benchmark dataset for comparing the performance of different machine learning models.

In recent years, deep learning models, such as convolutional neural networks (CNNs), have achieved state-of-the-art performance on the MNIST Fashion dataset, surpassing the accuracy of traditional machine learning models.

The dataset has many real-world applications, such as in the fashion industry, where it can be used to improve the accuracy of clothing item recommendations for customers. It can also be used in healthcare to classify different types of skin lesions or in surveillance to identify people based on their clothing.

In summary, the MNIST Fashion dataset is a widely-used dataset for image classification tasks, and it has many real-world applications. Its simplicity and well-defined labels make it an ideal dataset for experimenting with different machine learning models and techniques.

Label	Description	Examples
0	T-Shirt/Top	
1	Trouser	
2	Pullover	
3	Dress	
4	Coat	
5	Sandals	
6	Shirt	
7	Sneaker	
8	Bag	
9	Ankle boots	

1.2 Cifar-10 dataset

The CIFAR-10 dataset is a widely-used image classification dataset containing 60,000 32x32 color images in 10 different categories, each with 6,000 images. The dataset is split into a training set of 50,000 images and a test set of 10,000 images.

The 10 categories in the CIFAR-10 dataset include airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks. The dataset is commonly used as a benchmark for testing the performance of machine learning algorithms for image classification.

The CIFAR-10 dataset is challenging due to the relatively small size of the images and the high degree of similarity between some of the categories. This makes it a good test of a machine learning algorithm's ability to distinguish between similar objects.

The dataset has been used in numerous research studies to evaluate the performance of different machine learning models, including convolutional neural networks (CNNs), which have achieved state-of-the-art performance on the dataset.

The CIFAR-10 dataset has many practical applications, such as in the automotive industry, where it can be used to classify different types of vehicles. It can also be used in agriculture to identify different types of crops, or in healthcare to classify different types of medical images.

In summary, the CIFAR-10 dataset is a widely-used image classification dataset that has many real-world applications. Its challenging nature makes it a good test of a machine learning algorithm's ability to distinguish between similar objects, and its popularity has led to the development of numerous machine learning models and techniques for image classification.

1.3 Random forest

Random forest classifier is a popular machine learning algorithm used for classification, regression, and feature selection tasks. It is an ensemble method that combines multiple decision trees to create a more accurate and robust model.

The random forest classifier works by building multiple decision trees on random subsets of the training data and using a majority vote to classify new data points. Each tree is trained on a random subset of the features, which helps to reduce overfitting and improve the generalization performance of the model.

Random forest classifier has several advantages over other classification algorithms. It can handle high-dimensional data, noisy data, and missing values. It is also less prone to overfitting compared to other machine learning models, such as decision trees or support vector machines.

Random forest classifier has been successfully applied in various real-world applications, such as credit scoring, fraud detection, and medical diagnosis. It has also been used in computer vision tasks, such as object detection and image classification.

To improve the performance of random forest classifier, researchers have developed several techniques, such as bagging, boosting, and feature selection. These techniques aim to further reduce overfitting and improve the generalization performance of the model.

In summary, random forest classifier is a powerful machine learning algorithm that combines the strengths of decision trees and ensemble learning. It is widely used for classification tasks due to its high accuracy, robustness, and ability to handle complex datasets. Its versatility and effectiveness make it a popular choice for data scientists and researchers in various fields.

The basic principle of Random Forest is to combine multiple decision trees to obtain a more accurate and robust prediction model.

The algorithm works as follows: first, a random subset of features is selected from the input dataset, and a decision tree is built using the selected features. This process is repeated multiple times, with different subsets of features and random samples of the data. The output of each decision tree is then combined to obtain the final prediction.

To improve the accuracy and robustness of the model, Random Forest also employs two additional techniques: bagging and random feature selection. Bagging, or bootstrap aggregating, is a method of generating multiple training datasets by resampling the original data with replacement. Each decision tree is trained on a different bagged dataset, which helps to reduce overfitting and improve the model's generalization ability. Random feature selection, on the other hand, is a method of randomly selecting a subset of features at each node in the decision tree. This technique helps to decorrelate the trees and reduce the variance of the model, resulting in a more accurate and stable prediction.

The final prediction of the Random Forest algorithm is based on the majority vote of all decision trees. In classification tasks, each decision tree predicts the class label of the input data, and the class with the highest frequency among all decision trees is selected as the final prediction. In regression tasks, the output of each decision tree is averaged to obtain the final prediction.

In summary, the basic principle of Random Forest is to combine multiple decision trees trained on different subsets of features and data to obtain a more accurate and robust prediction model. The algorithm employs bagging and random feature selection techniques to reduce overfitting and decorrelate the trees, resulting in a more accurate and stable prediction.

1.4 BP neural network

Backpropagation (BP) neural network is a type of artificial neural network that is widely used for supervised learning tasks, such as classification and regression. It consists of an input layer, one or more hidden layers, and an output layer.

The BP neural network works by propagating the errors in the output layer back through the hidden layers to adjust the weights of the connections between neurons. This process is repeated iteratively until the model reaches a minimum error threshold.

BP neural network has several advantages over other machine learning algorithms. It can handle complex and nonlinear relationships between inputs and outputs, and it can learn from large datasets with high-dimensional features. Additionally, it can generalize well to new data, making it a useful tool for prediction and decision-making tasks.

To improve the performance of BP neural network, researchers have developed several techniques, such as regularization, dropout, and early stopping. These techniques aim

to prevent overfitting and improve the generalization performance of the model.

In summary, BP neural network is a powerful machine learning algorithm that is widely used for supervised learning tasks. Its ability to handle complex relationships between inputs and outputs, and its generalization ability make it a useful tool for prediction and decision-making tasks. With ongoing research, BP neural network is likely to continue to be an important tool in the field of machine learning.

The basic principle of BP neural networks is to use a set of input-output pairs to learn a mapping function between the inputs and outputs.

The network consists of multiple layers of nodes, including an input layer, one or more hidden layers, and an output layer. Each node in the network is connected to every node in the previous and next layers, and each connection is assigned a weight. The weights represent the strength of the connection between two nodes and are adjusted during the training process to minimize the error between the predicted and actual output.

The training process of a BP neural network involves two phases: forward propagation and backward propagation. During forward propagation, the input data is fed into the input layer, and the outputs of each layer are calculated by applying a nonlinear activation function to the weighted sum of the inputs. The outputs of the final layer are then compared to the actual output, and the error between them is calculated.

In the backward propagation phase, the error is propagated back through the network to adjust the weights of the connections. The weights are updated using an optimization algorithm, such as gradient descent, to minimize the error between the predicted and actual output. This process is repeated for multiple iterations until the error is minimized, and the network can accurately predict the output for new input data. The choice of activation function is critical in the BP neural network, as it determines the output of each node and affects the convergence rate of the network. Commonly used activation functions include sigmoid, ReLU, and Tanh.

In summary, the BP neural network is a type of artificial neural network that uses supervised learning to train a model. The network consists of multiple layers of nodes connected by weighted connections, and the weights are adjusted during the training process to minimize the error between the predicted and actual output. The network uses an activation function to compute the output of each node, and the optimization algorithm is used to update the weights during the training process.

2. Related work

2.1 Random forest

Random forest is a popular machine learning algorithm that has gained popularity due to its high accuracy and robustness. The algorithm has its roots in decision tree algorithms, which were first introduced in the 1960s. However, it was not until the 1990s that the concept of ensemble learning and bagging was proposed, leading to the development of the random forest algorithm.

In 1996, Leo Breiman³ and Adele Cutler introduced the random forest algorithm, which is an extension of the decision tree algorithm. The random forest algorithm combines

the strengths of decision trees and ensemble learning to create a more accurate and robust model. By building multiple decision trees on random subsets of the training data and using a majority vote to classify new data points, random forest can reduce the variance and overfitting issues commonly found in decision trees.

Since the introduction of the random forest algorithm, researchers have made several advancements in improving the algorithm's performance and expanding its applications. One such advancement is the introduction of the extremely randomized trees algorithm, which aims to further reduce overfitting and improve the generalization performance of the model. Another advancement is the use of ensemble learning techniques, such as bagging and boosting, to further improve the accuracy and robustness of random forest models.

2.2 BP neural network

The backpropagation (BP) neural network is a popular and widely used neural network model that has its roots in the 1980s. Since its introduction, researchers have made significant progress in developing and improving BP neural networks to achieve higher accuracy and better performance.

One major area of research has been to optimize the training algorithm of BP neural networks. The traditional backpropagation algorithm suffers from several issues, such as slow convergence, overfitting, and local minima. To overcome these issues, researchers have proposed several modifications to the algorithm, such as momentum, adaptive learning rate, and weight decay, to improve the convergence speed and generalization performance of the model.

Another area of research has been to develop new variants of BP neural networks to handle specific challenges in different application domains. For example, researchers have proposed recurrent neural networks (RNNs) to handle time-series data, convolutional neural networks (CNNs) to handle image data, and long short-term memory (LSTM) networks to handle sequential data. These variants of BP neural networks have achieved state-of-the-art performance in their respective application domains.

Furthermore, researchers have explored the use of ensemble learning techniques to improve the performance of BP neural networks. These techniques, such as bagging, boosting, and stacking, aim to combine multiple BP neural network models to create a more accurate and robust model.

Finally, researchers have also explored the interpretability of BP neural network models. Despite their high accuracy, BP neural networks are often considered as black box models, making it difficult to understand how they make predictions. To address this issue, researchers have proposed several methods to visualize the feature importance and decision-making process of BP neural network models, such as saliency maps and activation maximization.

3. Experimental results and analysis

3.1 Random forest

3.1.1 Performance measurement method

Precision Accuracy F1 (Precision、Recall)

3.1.2 Test set and training set partition

The training set contains 60,000 samples and the test set contains 10,000 samples, which are divided into 10 categories.

3.1.3 The performance indicators obtained under different parameters

Adjust the parameter `n_estimators`, that is, the number of trees, other values default, with the increase of trees, the accuracy will increase, after the parameter is 60, the accuracy will not change significantly.

	Change <code>n_estimators</code>									
	10	20	30	40	50	60	70	80	90	100
Precision	85.34	86.51	86.80	87.01	87.47	87.22	87.31	87.35	87.72	87.47
Accuracy	85.47	86.63	86.91	87.11	87.58	87.32	87.41	87.46	87.81	87.59

Adjust the parameters `max_depth`, tree depth, `n_estimators`=50, other values default, as the depth of the tree increases, the accuracy increases, after the parameter is 25, the accuracy does not change significantly.

	Change <code>max_depth</code> <code>n_estimators</code> =50									
	5	10	15	20	25	30	35	40	45	50
Precision	85.34	86.51	86.80	87.01	87.47	87.22	87.31	87.35	87.72	87.47
Accuracy	85.47	86.63	86.91	87.11	87.58	87.32	87.41	87.46	87.81	87.59

`max_depth`=50,`n_estimators`=100, and other values default. The data after 10 training sessions were averaged. The average value of Precision and Accuracy was 87.58 and 87.68 respectively.

	max_depth=50 n_estimators =100									
Epoch	1	2	3	4	5	6	7	8	9	10
Precision	87.54	87.41	87.83	87.68	87.52	87.53	87.50	87.47	87.65	87.64
Accuracy	87.64	87.53	87.92	87.77	87.64	87.32	87.61	87.57	87.75	87.73

Accuracy, recall rate, F1 value for each class, etc

	precision	recall	f1-score	support
0	0.82	0.86	0.84	1000
1	0.99	0.96	0.98	1000
2	0.76	0.80	0.78	1000
3	0.88	0.91	0.89	1000
4	0.76	0.81	0.79	1000
5	0.98	0.95	0.97	1000
6	0.72	0.58	0.64	1000
7	0.92	0.96	0.94	1000
8	0.95	0.97	0.96	1000
9	0.95	0.94	0.95	1000
avg / total	0.87	0.88	0.87	10000

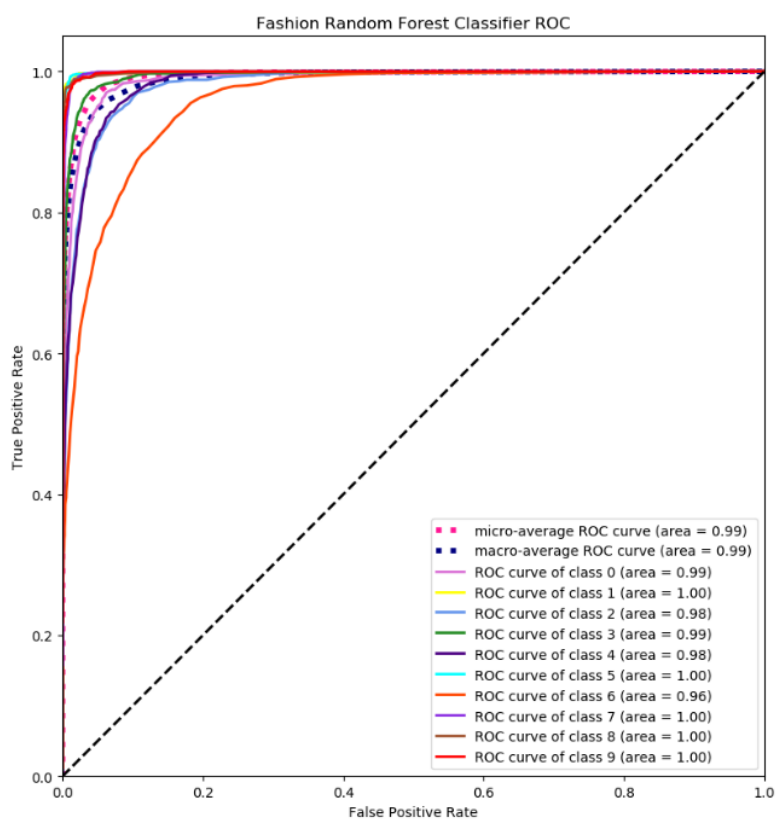
For label binarization processing, 10 trainings were conducted, and the data after 10 trainings were averaged. The average value of Precision was 92.37 and the average value of Accuracy was 81.41. Compared with the data without label binarization, the Precision was significantly improved, while the Accuracy was significantly decreased.

	max_depth=50 n_estimators =100									
Epoch	1	2	3	4	5	6	7	8	9	10
Precision	92.45	92.45	92.48	92.38	92.15	92.33	92.52	92.25	92.24	92.45
Accuracy	81.57	81.57	81.60	81.38	81.15	81.17	81.46	81.18	81.59	81.39

Accuracy, recall rate, F1 value for each class, etc

	precision	recall	f1-score	support
0	0.87	0.77	0.82	1000
1	1.00	0.95	0.97	1000
2	0.85	0.69	0.76	1000
3	0.93	0.84	0.88	1000
4	0.84	0.71	0.77	1000
5	0.99	0.93	0.96	1000
6	0.86	0.43	0.58	1000
7	0.94	0.94	0.94	1000
8	0.99	0.95	0.97	1000
9	0.96	0.94	0.95	1000
avg / total	0.92	0.82	0.86	10000

ROC curve



3.2 BP neural network

3.2.1 Performance measurement method

Loss Accuracy

3.2.2 Test set and training set partition

The training set contains 60,000 samples and the test set contains 10,000 samples, which are divided into 10 categories.

3.2.3 Network structure

Net(←

Sequential(←

(0): Linear(in_features=3, out_features=128, bias=True)←

(1): ReLU()←

(2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)←
)←

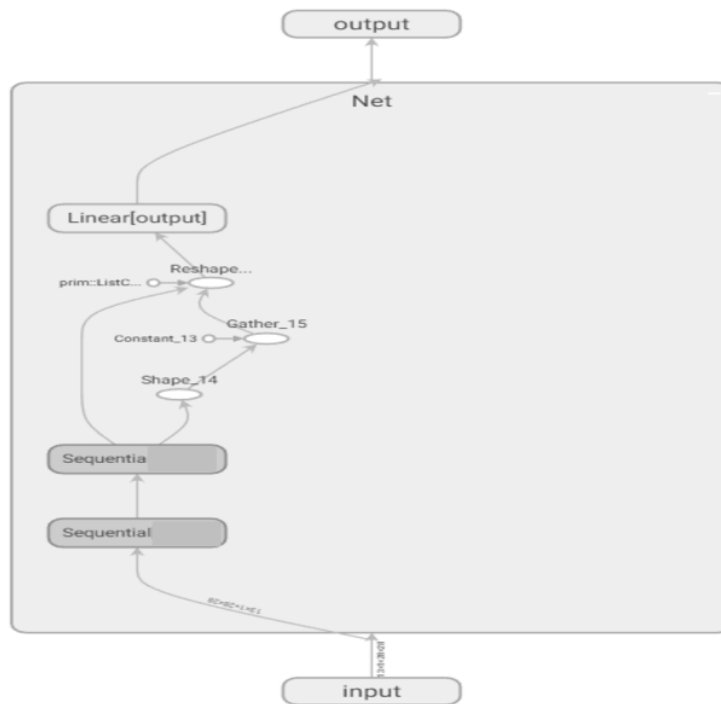
Sequential(←

(0): Linear(in_features=128, out_features=1568, bias=True) ←

(1): ReLU()←

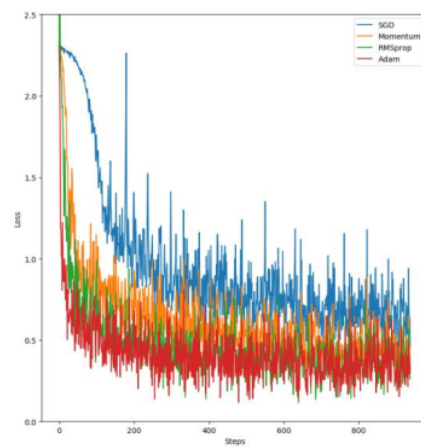
(2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)←
)←

(output): Linear(in_features=1568, out_features=10, bias=True)←



3.2.4 The choice of the optimizer

Four optimizers were tested, as shown in the figure below. Adam has the best effect when the epoch is performed once.

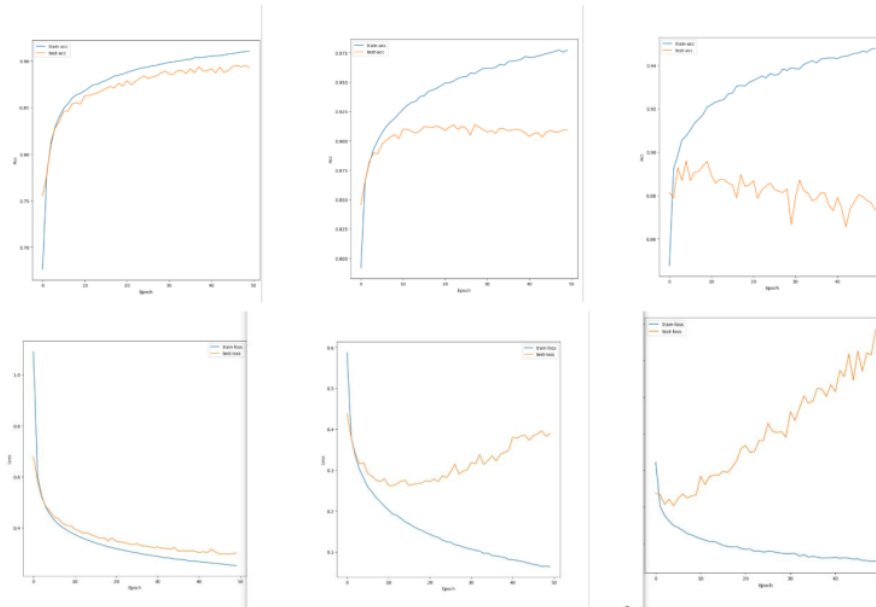


3.2.5 The performance indicators obtained under different parameters

(1) Lr

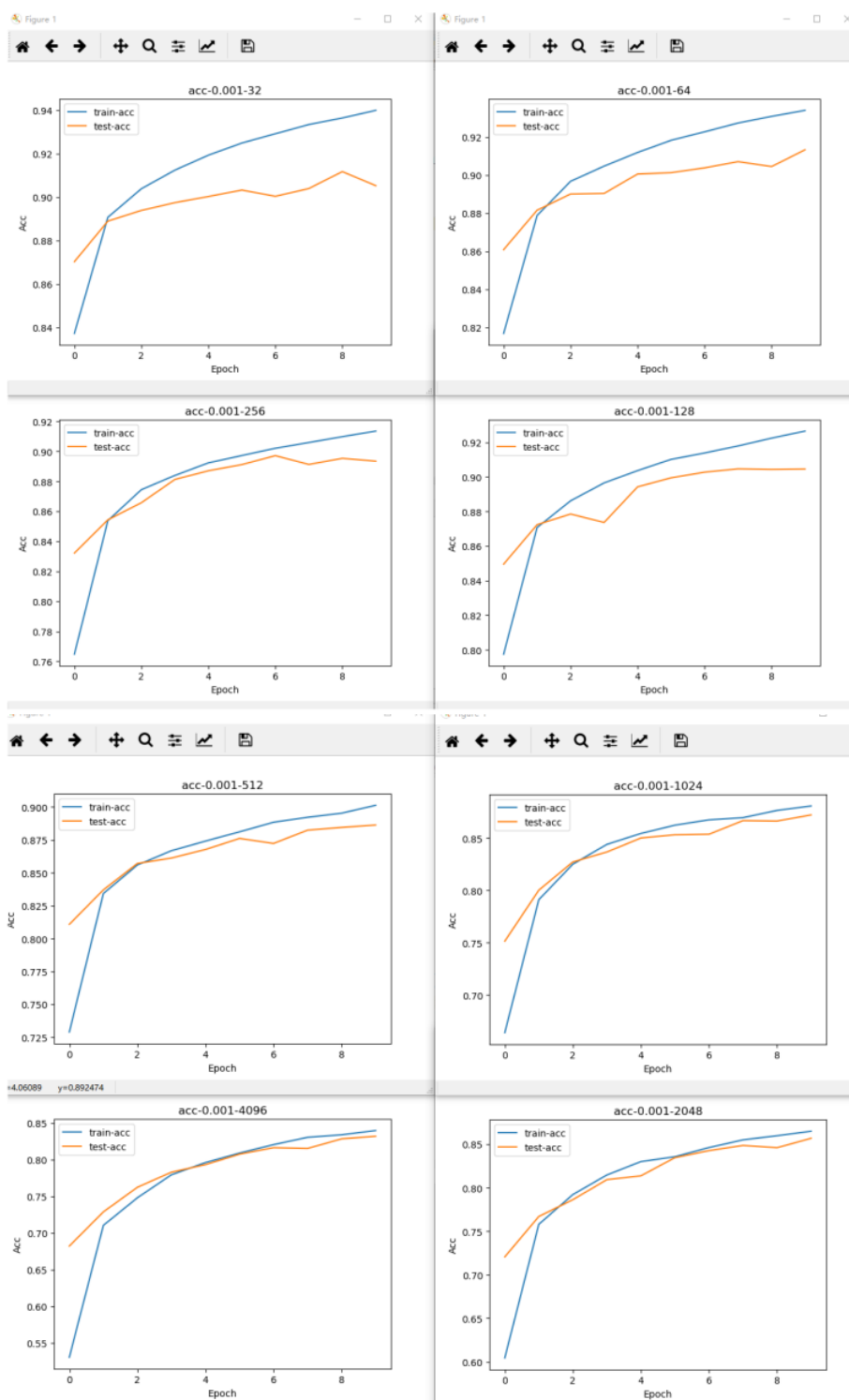
	Epoch=50 batch_size=128			
Lr	0.0001	0.001	0.01	0.1
Train Accuracy	91.07	97.28	94.78	9.83
Test Accuracy	89.32	90.95	87.32	10.00

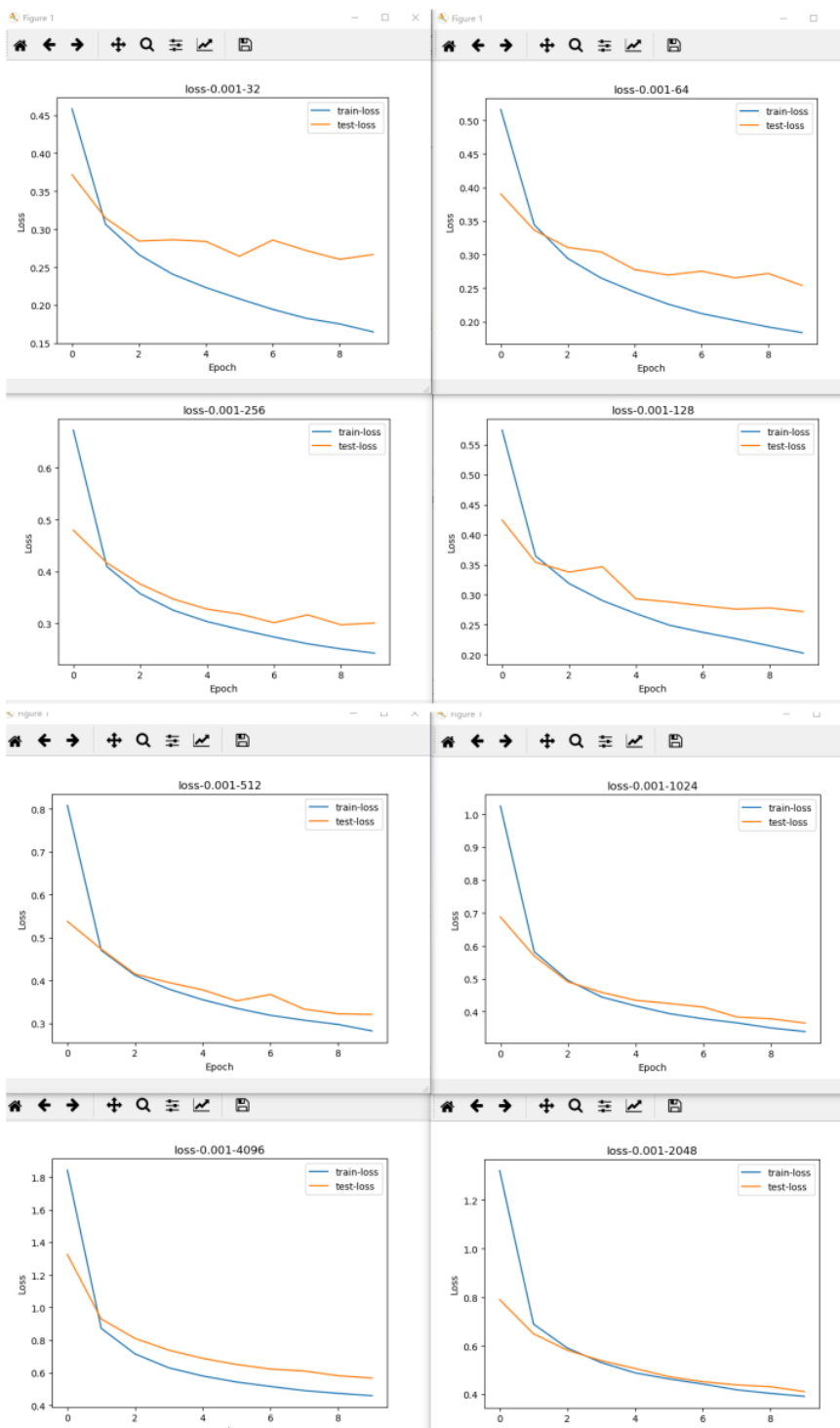
From left to right, the graphs of accuracy-epoch are 0.0001, 0.001 and 0.01. It can be seen from the curves that the last two are overfitted, because the epoch is too big, so the training can be ended early.



(2) batch size

	Epoch=10 Lr =0.001							
batch_size	32	64	128	256	512	1024	2048	4096
Train Accuracy	93.99	93.39	92.64	91.34	90.13	88.07	86.48	83.97
Test Accuracy	90.52	91.32	90.45	89.34	88.63	87.22	85.67	83.19





4. summarize

This paper compares the performance of two popular machine learning algorithms, random forest and backpropagation (BP) neural network, in image classification tasks. The study aims to provide insights into the strengths and weaknesses of these algorithms in handling complex image data.

The authors found that both random forest and BP neural network are effective in image classification tasks, with random forest performing slightly better in terms of accuracy and computational efficiency. However, the study also found that the performance of both algorithms is highly dependent on the dataset size and the choice of hyperparameters, such as the number of trees in the random forest and the number of hidden layers in the BP neural network.

The study also highlights the importance of feature selection and feature extraction in improving the performance of both algorithms. The authors found that selecting a subset of relevant features and using feature extraction techniques, such as principal component analysis (PCA) and local binary patterns (LBP), can significantly improve the accuracy and efficiency of both random forest and BP neural network models.

Furthermore, the study examines the interpretability of both algorithms in image classification tasks. The authors found that random forest models are more interpretable than BP neural network models, as the former can provide feature importance rankings that allow for better understanding of the decision-making process. However, the study notes that recent developments in BP neural network visualization techniques, such as activation maximization and gradient-weighted class activation mapping (Grad-CAM), have improved the interpretability of BP neural network models.

In conclusion, this study provides a valuable comparison of two popular machine learning algorithms in image classification tasks. The results suggest that random forest and BP neural network are both effective in handling complex image data, with random forest performing slightly better in terms of accuracy and efficiency. The study highlights the importance of feature selection and extraction in improving the performance of both algorithms, and the trade-off between accuracy and interpretability in choosing the appropriate algorithm. The results of this study can inform the design of predictive models for image classification tasks in various application domains, such as medical diagnosis and autonomous driving.

References

- [1] Haifei Zhang;Benjamin Quost;Marie-Hélène Masson.Cautious weighted random forests[J].Expert Systems with Applications,2023,Vol.213: 118883
- [2] Bai, Jiawang;Li, Yiming;Li, Jiawei;Yang, Xue;Jiang, Yong;Xia, Shu-Tao.Multinomial random forest.[J].Pattern Recognition,2022,Vol.122
- [3] Dmitry Devyatkin;G. Oleg Grigoriev.Random Kernel Forests[J].IEEE Access,2022,Vol.10: 77962-77979
- [4] Xu, Haotong;Shi, Hongyuan;Ni, Shiquan.Application of BP Neural Networks in Tide Forecasting.[J].Atmosphere,2022,Vol.13(12): 1999
- [5] Libiao Bai;Yuqin An;Yichen Sun.Measurement of Project Portfolio Benefits With a GA-BP Neural Network Group[J].IEEE Transactions on Engineering Management,2023,: 1-13
- [6] Li Liu;Qianlei Peng;Huaping Jiang;Li Ran;Yang Wang;Changhong Du;Jian Chen;Hongbo Zhou;Yang Chen;Zhiyuan Peng.BP neural network for non-invasive IGBT junction temperature online detection[J].Microelectronics Reliability,2023,Vol.141: 114882

Classification using ML methods and Neural Networks

GRADEMARK REPORT

FINAL GRADE

59/100

GENERAL COMMENTS

Instructor

Late. Mark will be capped accordingly.

An abstract would help here to summarize your paper/report.

Please use inline citations to support assertions and arguments. This is crucial at this level. Currently your refs are more of a bibliography.

Use your related work section to include work that, for example, attempts similar experiments on your given datasets. This can then be used in your results/conclusions section to validate your results, for example.

Good set of experiments and you document many very well. Nice to see table of estimators used and their effect on the random forest's accuracy.

Good.

Very good to see you experiment with optimisers - graph is helpful here too. Well done.

Training and validation of MLP provided and detailed well. Again good.

Some bar plots graphing and comparing results would help enormously in your results/discussions.

Overall this report is a bit of a "curate's egg." That is, it is good in parts. You need to improve upon your referencing and related work sections. Use in line citations more. Don't be afraid to use more references if they support your discussions.

Your experiments are thorough and detailed - very good indeed.

Your report is late. EC applied for I hope?

PAGE 1

PAGE 2

PAGE 3



Comment 1

..which are?

PAGE 4



Comment 2

..again, which are
?

PAGE 5

PAGE 6



Comment 3

Good

PAGE 7

PAGE 8

GRADING FORM: SMU GENERIC_GJ

ZIYAO CHEN

0

RESEARCH-INFORMED LITERATURE

KNOWLEDGE AND UNDERSTANDING

ANALYSIS

PRACTICAL APPLICATION AND DEP.

SKILLS FOR PROFESSIONAL PRAC.