WEP Warwick Evans Publishing

International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 3, Number 3, Year 2024
DOI: https://doi.org/10.62051/ijcsit.v3n3.31
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Research on Wine Classification Based on BP Neural Network

Le Jiang

Southwest Minzu University, College of Electronic Information, Chengdu, Sichuan province, China

ABSTRACT

The Wine Wine Dataset is a publicly available dataset from UCI above, which is the result of a chemical analysis of wines grown in the same region of Italy from three different varieties, red, white and rosé. BP neural network is a common artificial neural network model that is used to supervise learning tasks. It is a multi-layer feedforward neural network that updates network parameters through a backpropagation algorithm to minimize the loss function. Principal component analysis (PCA) is a commonly used data dimensionality reduction technique to discover the main features in a dataset. It maps the original data to a new coordinate system through a linear transformation, maximizing the variance of the data in the new coordinate system. This reduces the dimensions of the data while preserving as much information as possible. In this experiment, the wine dataset was used, combined with principal component analysis and BP neural network, to classify and reason three types of wines.

KEYWORDS

Wine; Principal Component Analysis; BP Neural Networks

1. INTRODUCTION

The purpose of this task is to use the combination of PCA (principal component analysis) and BP neural network to classify the wine dataset, which involves data analysis, data normalization, principal component screening, training, testing and data visualization, and then obtain the classification results of each image. The main function of PCA is to reduce the original 13 features of the wine dataset to two features, and the dimensionality reduction process is carried out by selecting the best combination of the two characteristics of the eigenvalues, and finally leaving two groups of the most significant features of the eigenvalues as the classification basis, and then putting these two groups of features into the BP neural network for classification, and finally divided into three categories, represented by 0, 1 and 2. Among them, the principles of data dimensionality reduction are as follows:

(1) Reduce computational complexity: By reducing the dimension of the dataset, dimensionality reduction can reduce the computational cost of machine learning algorithms and improve the training and prediction efficiency of models.

(2) Removal of redundant information: Dimensionality reduction helps to eliminate redundant information and noise in high-dimensional data, thereby improving the generalization performance of the model and reducing the risk of overfitting.

(3) Visualization: Most of the datasets of machine learning are high-dimensional, but our human world is three-dimensional, so images that exceed three dimensions cannot be drawn by man;

Figure 1 shows the data set.

	(Alcohol)	(Malic Acid)	(Ash)	(Alcalinity of Ash)	(Magnesium)	(Total Phenols)	(Flavanoids)	...	
1	14.23	1.71	2.43	15.6	127	2.80	3.06	...	Class 0
2	13.20	1.78	2.14	11.2	100	2.65	2.76	...	Class 1
3	12.37	1.17	1.92	19.6	97	2.10	2.06	...	Class 2

Figure 1. Some datasets

2. ALGORITHM DESIGN IDEAS

In this task, the steps to classify a wine dataset using principal component analysis (PCA) and BP neural network are as follows:

1) Data pre-processing: Load the wine dataset and normalize the data to ensure that the mean value of the data is 0 and the variance is 1.

2) Principal component analysis (PCA): PCA is used to reduce the dimensionality of the data, select an appropriate number of principal components, and project the data into a new low-dimensional space.

3) Data division: The dataset after dimensionality reduction is divided into a training set and a test set.

4) BP neural network model construction: Construct a BP neural network model that includes an input layer, a hidden layer and an output layer.

5) Model training: The BP neural network model is trained using the training set, and the network weights and biases are adjusted so that the model can accurately classify the wine dataset.

6) Model evaluation: The test set is used to evaluate the trained BP neural network model, and the accuracy, precision, recall and other indicators of the model are calculated.

7) Result analysis: Analyze the classification results of the model and visualize the results, and optimize and improve the model.

3. ALGORITHM IMPLEMENTATION PROCESS

3.1. Import Library and Dataset Download

Import the SKLEARN library, matplotlib library, numpy library, font management library, etc., to prepare for prediction. The process is shown in Figure 3.1.

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.datasets import load_wine
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score, classification_report
from matplotlib.font_manager import FontProperties
```

Figure 3.1. Importing a library

Then, the data is downloaded through datasets, and the data is the public dataset wine, and the characteristics and objectives of the data are stored and classified with X and y, respectively, and printed, as shown in Figure 3.2.

```
from sklearn.datasets import load_wine
wine_data = load_wine()
X = wine_data.data
y = wine_data.target
print(X)
print(y)
```

```
[[1.423e+01 1.710e+00 2.430e+00 ... 1.040e+00 3.920e+00 1.065e+03]
 [1.320e+01 1.780e+00 2.140e+00 ... 1.050e+00 3.400e+00 1.050e+03]
 [1.316e+01 2.360e+00 2.670e+00 ... 1.030e+00 3.170e+00 1.185e+03]
 ...
 [1.327e+01 4.280e+00 2.260e+00 ... 5.900e-01 1.560e+00 8.350e+02]
 [1.317e+01 2.590e+00 2.370e+00 ... 6.000e-01 1.620e+00 8.400e+02]
 [1.413e+01 4.100e+00 2.740e+00 ... 6.100e-01 1.600e+00 5.600e+02]]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2]
```

Figure 3.2. Data preparation

The data is then normalized and printed out to see how it works, and in machine learning, different features often have different units of measurement and ranges. This difference can result in some features having a greater impact on the model and others having little to no effect. By normalizing or normalizing, the values of features can be scaled to similar scales, eliminating dimensional differences and making the model more equitable for individual features. The process is shown in Figure 3.3.

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
print(X_scaled)
```

```
[[ 1.51861254 -0.5622498   0.23205254 ...  0.36217728  1.84791957
   1.01300893]
 [ 0.24628963 -0.49941338 -0.82799632 ...  0.40605066  1.1134493
   0.96524152]
 [ 0.19687903  0.02123125  1.10933436 ...  0.31830389  0.78858745
   1.39514818]
 ...
 [ 0.33275817  1.74474449 -0.38935541 ... -1.61212515 -1.48544548
   0.28057537]
 [ 0.20923168  0.22769377  0.01273209 ... -1.56825176 -1.40069891
   0.29649784]
 [ 1.39508604  1.58316512  1.36520822 ... -1.52437837 -1.42894777
  -0.59516041]]
```

Figure 3.3. Data normalization

3.2. Principal Component Analysis (PCA)

The original wine dataset has 13 irrelevant features, such as alcohol content, malic acid, alkalinity, etc., principal component analysis is a commonly used data dimensionality reduction technique, which converts the original data into a set of orthogonal principal components through linear transformation to reduce the dimensionality of the data and retain the important information of the data. The principle of PCA is to find the direction with the largest variance in the data as the first principal component, and then find the direction orthogonal to the first principal component and with the largest variance as the second principal component, and so on until the specified number of principal components are found or sufficient variance is retained. The dimensionality reduction process is shown in Figure 3.4.

```
from sklearn.decomposition import PCA

n_components = 2
pca = PCA(n_components=n_components)
X_pca = pca.fit_transform(X_scaled)
print(X_pca)
```

```
[[ 3.31675081 -1.44346263]
 [ 2.20946492  0.33339289]
 [ 2.51674015 -1.0311513 ]
 [ 3.75706561 -2.75637191]
 [ 1.00890849 -0.86983082]
 [ 3.05025392 -2.12240111]
 [ 2.44908967 -1.17485013]
 [ 2.05943687 -1.60896307]
 [ 2.5108743  -0.91807096]
 [ 2.75362819 -0.78943767]
 [ 3.47973668 -1.30233324]
 [ 1.7547529  -0.61197723]
 [ 2.11346234 -0.67570634]
 [ 3.45815682 -1.13062988]
 [ 4.31278391 -2.09597558]
 [ 2.3051882  -1.66255173]
 [ 2.17195527 -2.32730534]
 [ 1.89897118 -1.63136888]
 [ 3.54198508 -2.51834367]
```

Figure 3.4. PCA dimensionality reduction

3.3. Create a Model and Train It

In this task, the two features of the dimensionality reduction are put into the BP neural network for training and testing after the PCA dimensionality reduction, and the dimensionality reduction dataset is divided into two parts: the training set and the test set, with a ratio of 8 to 2. The basic principle of the BP neural network algorithm is to use gradient descent to adjust the weights in the network so that the output of the network is as close to the actual value as possible. The algorithm consists of two processes, forward propagation, which is used to calculate the output of the network, and backpropagation, which is used to update the weights in the network according to the output error. The specific implementation is shown in Figure 3.5.

```
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
X_train, X_test, y_train, y_test = train_test_split(X_pca, y, test_size=0.2, random_state=42)

model = MLPClassifier(hidden_layer_sizes=(10, 5), max_iter=500, random_state=42)

model.fit(X_train, y_train)
```

```
C:\Users\13161\AppData\Roaming\Python\Python39\site-packages\sklearn\neural_network\_multilayer_perceptron.py:691: ConvergenceWarning: Stoch
astic Optimizer: Maximum iterations (500) reached and the optimization hasn't converged yet.
  warnings.warn(
```

```
MLPClassifier(hidden_layer_sizes=(10, 5), max_iter=500, random_state=42)
```

Figure 3.5. Model training

3.4. Model Evaluation and Data Visualization

After the dataset is trained in step 3.4, the trained model is tested and the test results are visualized, and the evaluation model is to test the trained model on the test set, and the test indicators are accuracy, recall, F1, precision, macro average and weighted average, these values are printed out in tabular form, and then a scatter plot of the classification results is drawn, the classification results are displayed in the figure, the wine is divided into three categories, and the decision boundary after dimensionality reduction is drawn, to facilitate analysis and presentation of results. The specific steps and data presentation are shown in Figures 3.6 and 3.7.

```
y_pred = model.predict(X_test)


accuracy = accuracy_score(y_test, y_pred)
print(f"         {accuracy:.2f}")


report = classification_report(y_test, y_pred)
print("          ")
print(report)


font = FontProperties(family="SimHei", size=12)


plt.figure(figsize=(10, 6))


scatter = plt.scatter(X_pca[:, 0], X_pca[:, 1], c=y, cmap=plt.cm.Set1, edgecolor='k')
legend_labels = [wine_data.target_names[i] for i in np.unique(y)]
plt.legend(handles=scatter.legend_elements()[0], labels=legend_labels, title="    ", prop=font)
plt.xlabel('        ', fontproperties=font)
plt.ylabel('        ', fontproperties=font)

#
h = .02
x_min, x_max = X_pca[:, 0].min() - 1, X_pca[:, 0].max() + 1
y_min, y_max = X_pca[:, 1].min() - 1, X_pca[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))
Z = model.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
plt.contourf(xx, yy, Z, cmap=plt.cm.Set1, alpha=0.8)

plt.show()
```

Figure 3.6. Model evaluation and data visualization code

0.97

	precision	recall	f1-score	support
0	1.00	0.93	0.96	14
1	0.93	1.00	0.97	14
2	1.00	1.00	1.00	8
accuracy			0.97	36
macro avg	0.98	0.98	0.98	36
weighted avg	0.97	0.97	0.97	36

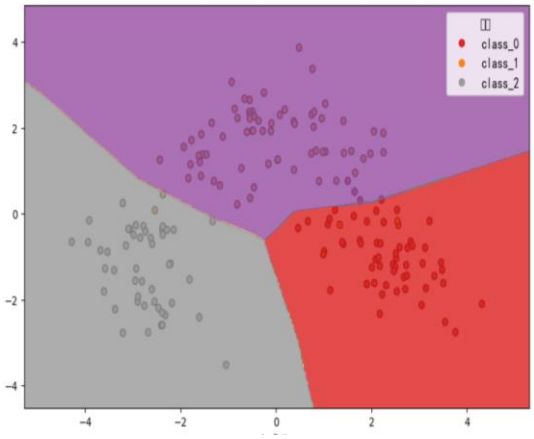

Figure 3.7. Evaluating data and visualizing results

4. EXOERIMENTAL RESULTS AND ANALYSIS

The training results shown in the third part of this task show that the model training effect has reached the best performance, the accuracy of the training set and the test set are more than 96%, and the accuracy and recall rate shown in the training set and the test set are very stable, indicating that the generalization ability of the model is good, and the experimental results basically meet the classification requirements, from the image point of view, the classification results are roughly the same as the expected classification expectations, and from the average of the accuracy and recall rate of its F1 value response, the method used in this training has a good effect on wine classification. However, the dataset is not large enough, so the model may be less accurate for more complex features and training.

5. DESIGN SUMMARY

In this task, on the basis of BP neural network and principal component analysis, the wine classification task is carried out, and after the dimensionality reduction through PCA, the dimensionality reduction features are put into the BP neural network for training, classification and testing, and the model can well identify the wine species on the given dataset to achieve the purpose of classification.

There are also shortcomings in this mission, which are roughly as follows:

1) The number of datasets is small;

2) The method is more targeted, and may have better results for other wine characteristics that are not considered;

3) It needs to be tuned to make the model have better performance, etc.

BP neural network is a class of multi-layer feedforward neural networks based on the BackPropagation (BP) algorithm, which is the most successful neural network learning algorithm to date. When neural networks are used in real-world tasks, they are mostly trained using BP algorithms. The BP algorithm can be used not only for multi-layer feedforward neural networks, but also for other types of neural networks, such as training recurrent neural networks.

ACKNOWLEDGEMENTS

In the course of this study, we are grateful for the help we received in completing this study. First of all, we would like to thank our advisors for their guidance and support for our research direction. The

teachers not only gave us valuable advice and guidance academically, but also provided important help in the process of experimental design and data analysis.

Secondly, we would like to thank the relevant units and organizations that have provided data and resources to support them. Without their data sharing, we would not be able to conduct experiments and analyze the results.

In addition, we would like to thank our colleagues in the lab for their help and inspiration in the experimental operation and discussion of the results. Their cooperation and exchange facilitated the smooth progress of the research work.

Finally, we would like to thank our family and friends for their understanding and support during our research. Their support is our motivation and spiritual pillar.

In short, it is with the support and help of the above parties that we were able to successfully complete this research paper on wine classification based on BP neural network.

REFERENCES

[1] Wu Sihan, Zhao Nannan. Application of BP neural network in image processing [J]. Electronic Technology and Software Engineering, 2021, (23):112-113.)

[2] Li Bingbing. Electricity load prediction method based on k-means clustering and BP neural network [J]. Light Source & Illumination, 2022, (04):165-167.)

[3] Yan Huabing. Foreign object detection of train platform door based on BP neural network [J]. China New Technology and New Products, 2023, (16):86-88. DOI:10.13612/j.cnki.cntp.2023.16.016

[4] Li Yuefang, Wang Xibo, Gao Yanfei et al. Tire pattern groove recognition based on BP neural network [J]. Journal of Shandong Jiaotong University, 2023, 31(01):1-6.)

[5] Zeng Qi. Automatic image recognition technology based on BP neural network [J]. Applications of Automation, 2022, (11):73-75+80. DOI:10.19769/j.zdhy.2022.11.018

[6] Li Jie, Li Denggang, Zeng Wenliang. Research on license plate recognition method based on BP neural network [J]. Electronic Testing, 2022, (16):38-40+46. DOI:10.16520/j.cnki.1000-8519.2022.16.040