

Student Assignment

**6057CEM Artificial Neural Networks:**

**Developing and Evaluating Neural Network Models for Iris Data**

**Name: GeRui Ye**

**Student ID: 15490111**

**Module Name: Artificial Neural Networks**

**Module Code: 6057CEM**

**Task: Developing and Evaluating Neural Network Models for Iris Data**

**Coventry github: <https://github.coventry.ac.uk/6057CEM-2425/yeg3-sem2>**

**Demonstration video links:**

**<https://livecoventryac-my.sharepoint.com/:f:/g/personal/yeg3_uni_coventry_ac_uk/El3IG1bFIl9Pn7Oilf6BISwBMep7zHj8Qpf_5S7ieXJsLg>**

**Submission Date: 4/8/2025**

Table of Contents

**[1.](#_Toc191155903)****[Introduction](#_Toc191155903)** [3](#_Toc191155903)

**[1.1.](#_Toc191155904)****[Context](#_Toc191155904)** [3](#_Toc191155904)

**[2.](#_Toc191155905)****[Model Development](#_Toc191155905)** [3](#_Toc191155905)

**[2.1.](#_Toc191155906)****[Exploratory Data Analysis](#_Toc191155906)** [3](#_Toc191155906)

**[2.2.](#_Toc191155907)****[Neural Network Design](#_Toc191155907)** [8](#_Toc191155907)

**[2.3.](#_Toc191155908)****[Model Training](#_Toc191155908)** [10](#_Toc191155908)

**[2.4.](#_Toc191155909)****[Evaluation](#_Toc191155909)** [11](#_Toc191155909)

**[3.](#_Toc191155910)****[Conclusion](#_Toc191155910)** [15](#_Toc191155910)

**[3.1.](#_Toc191155911)****[Comparison of models](#_Toc191155911)** [15](#_Toc191155911)

**[3.2.](#_Toc191155912)****[Potential direction](#_Toc191155912)** [15](#_Toc191155912)

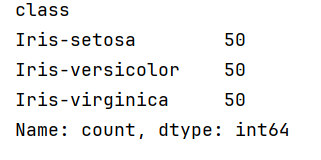
1. **Introduction**
   1. **Context**

Deep learning models have grown extensively in the field of artificial intelligence, and deep learning is gradually becoming known as the gold standard of machine learning (Khemani et al., 2024) therefore deep learning models Feed-Forward(FFNN) and Convolutional Neural Network(CNN) can help AI systems to learn to process real-world data and make decisions; the iris dataset from UCI is used in this study to design two different neural network models using the iris dataset. It can help AI systems to learn to process data and make decisions; to design two different neural network models and analyze their performance in classification problems.

The main topics of this research are (1) Deep analysis of the advantages and limitations of neural network models in processing iris datasets and (2) Detailed exploration of the arithmetic mechanisms of two neural network models. (3) To study and utilize the extensive literature review to explore the potential future direction of deep learning models for Real-World Data.

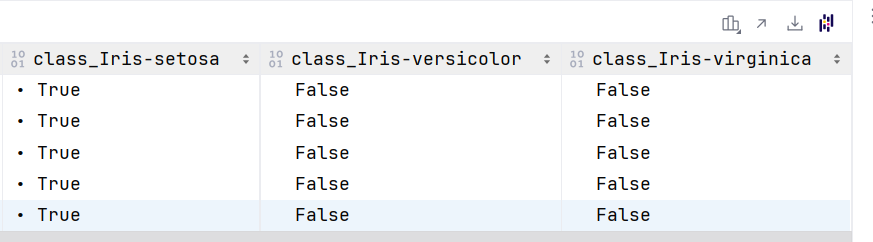
1. **Model Development**
   1. **Exploratory Data Analysis**

The iris dataset from UCI is a multi-class classification problem (Fisher, 1936) divided into three iris species: Setosa, Versicolor, and Virginica, and for this study, the value\_counts() function from the panda library was used to calculate the number of iris species in the class column (pandas development team, 2024):



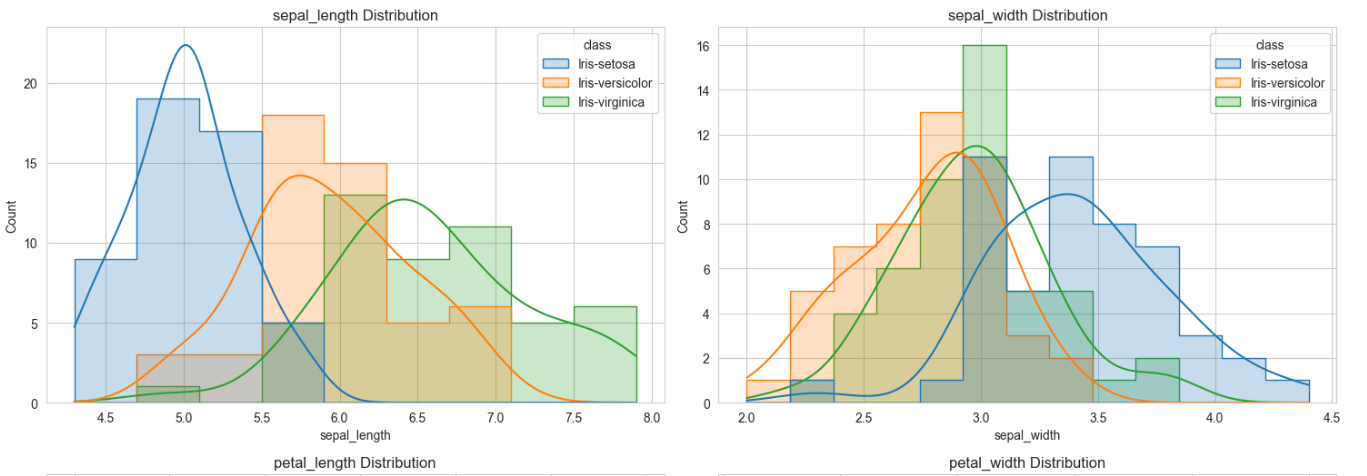
It was found that each sample has 50 each, indicating that this is a balanced classification problem and that each sample has four features: sepal length, sepal width, petal length, and petal width, based on which the species of iris is predicted.

With only three categories in the Iris dataset and no relationships, each Iris species was first categorized into a column using One-Hot Encoding.



Also could be used Embedding for processing; embedding is more suitable for deep learning models and helps to recognize the underlying structure of the data. (Anil & Kaur, 2024)

Next, data visualization of the four features of Iris was observed using Matplotlib and Seaborn, to draw histograms and render images:



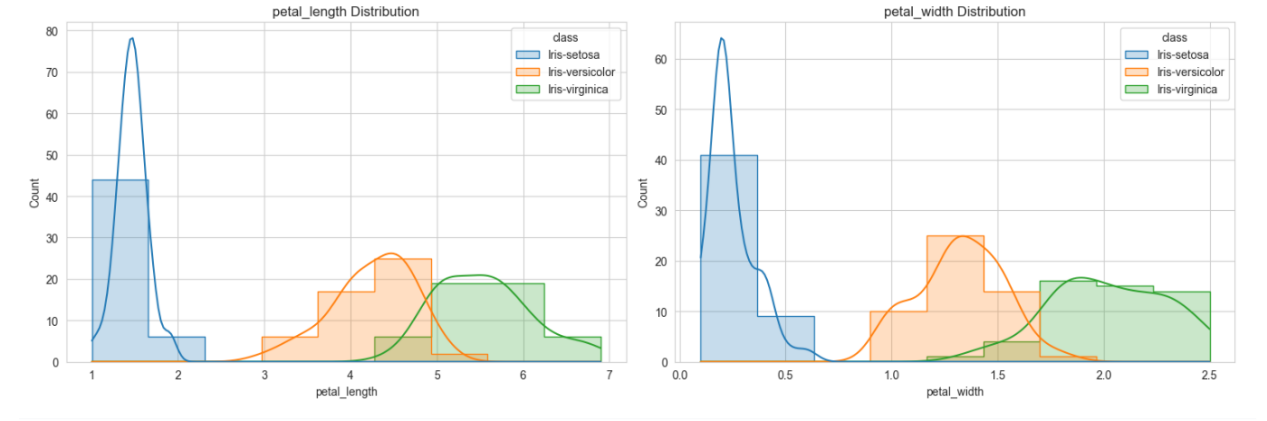
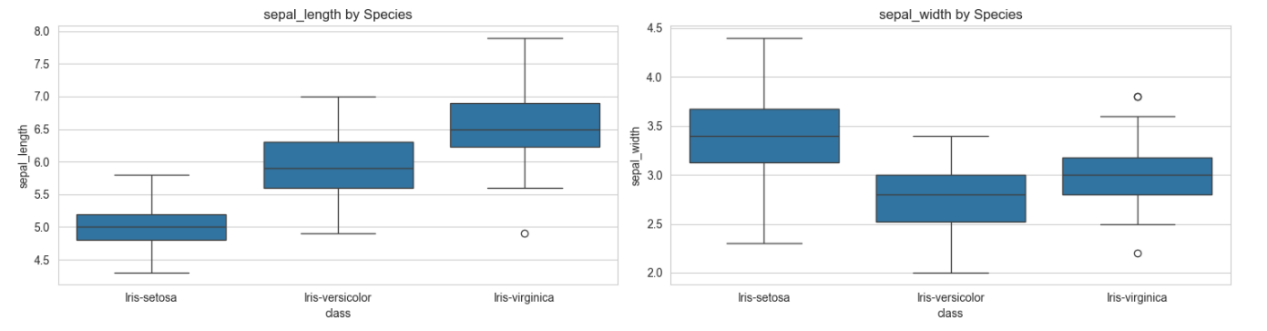


Fig.

The graph clearly shows the peaks of sepal length and width, and petal length and width of different species of Iris, and the general intervals in which the features are located for the different species.

**Make Box Diagram:**

Box plots can be generated using Seaborn's *.boxplot* function, which shows the characteristic intervals of each eigenvalue and identifies possible outliers and median positions.



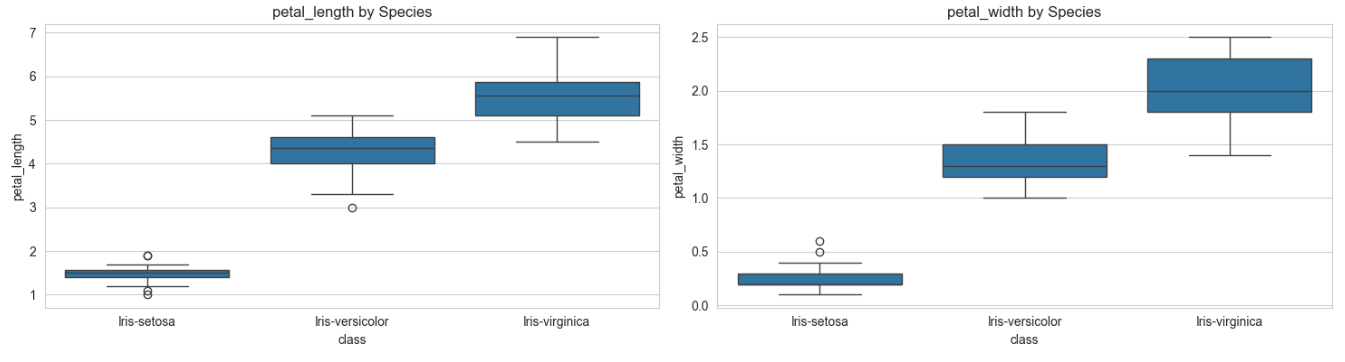


Fig.

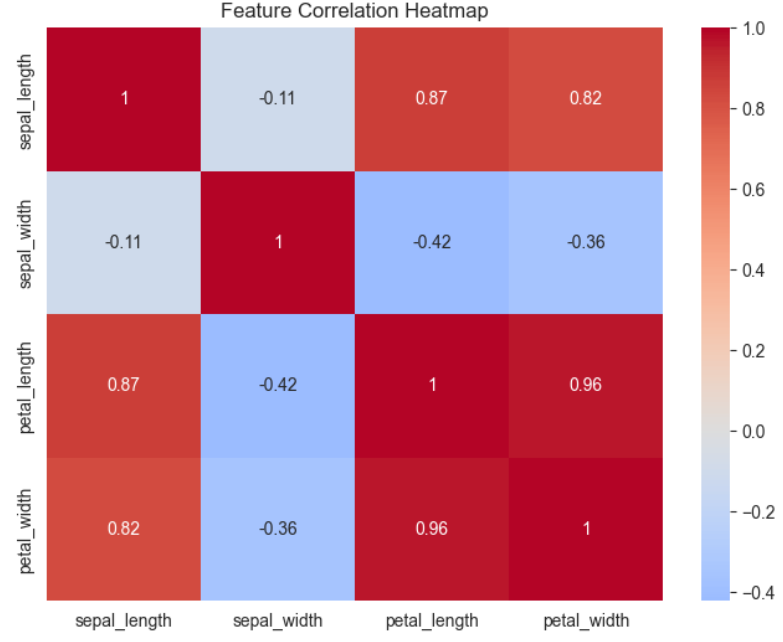
**Scatterplot matrix:**

Plotting the Scatterplot matrix can be used to determine the two-by-two relationship between sepal length and width and petal length and width for different iris species, and by looking at the distribution of points, the approximate location of the iris species can be circled.  


Fig.

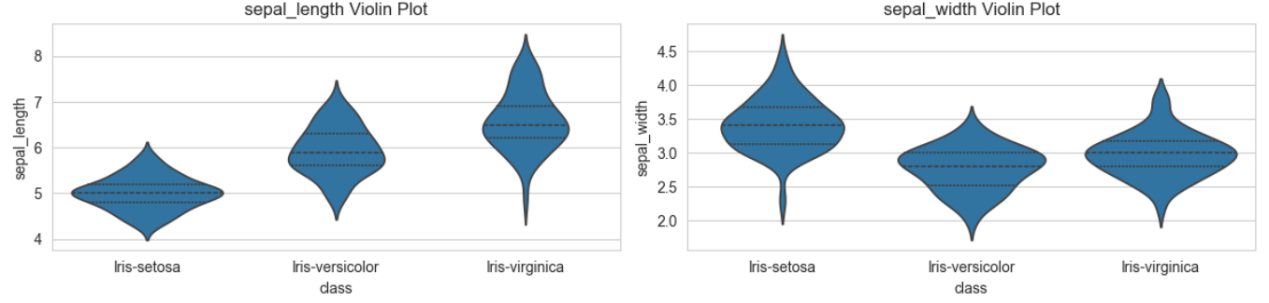
**Correlation matrix heat map:**

Plotting the heat map, it can be observed that the iris eigenvalues relationship, as shown in the figure, is observed that the positive correlation between petal length and petal width reaches 96%, and the negative correlation between sepal width and petal length reaches 42%.



**Violin Diagram:**

Violin Diagram drawn by sns *.violinplot* function, this figure can observe the distribution of features, as shown in the figure, sepal and petal eigenvalues, the two segments are relatively long and thin, indicating that the distribution of data is more centralized, and in the petal data, each violin image, the difference is more obvious and easy to determine the different iris species of the petal difference.



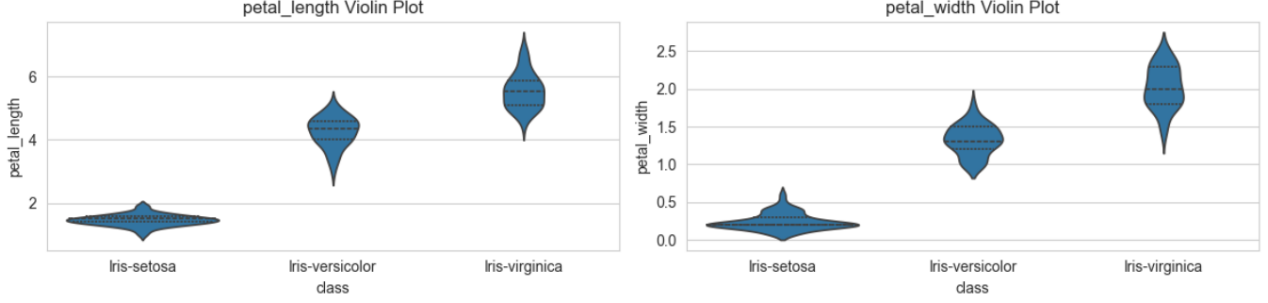


Fig.

Next, this project will perform a train-test split on the iris dataset. The size and complexity of the test and training sets can determine the performance of the deep learning model. Can also specify the MSE or SSIM and the mean-variance on the training set to train the model. A good training set can have a sufficiently representative image complexity to make the model results have better performance results.(Ogun Kirmemis et al. 2018)

This train-test split work will divide 30% as the test set and 70% as the training set and split the data by category using *stratify=y* to ensure that the training and test sets have the same proportion of each category. Otherwise, the split results will be unevenly distributed.

In addition, according to the box diagram, we can find that there are still a small number of outliers in this data, so we adopt standardization to deal with the data, transform each eigenvalue into data with mean 0 and standard deviation 1, and control the outliers generated by iris species to prevent them from affecting the model too much, which can make the trained machine learning model more accurate.

Use the standardization function from the Scikit-learn library to fix the data:

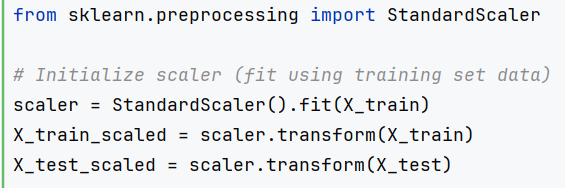


Fig.

Calculate the mean and standard deviation of the training data. Standardize the data, The equation is :



Standardize processing data, the data was a NumPy array, only the value of no column name, to facilitate the subsequent operation, the feature name given to the array was transformed into a DataFrame, so that the data is more intuitive, would not be lost quotation marks. The figure shows one of the five columns after processing:

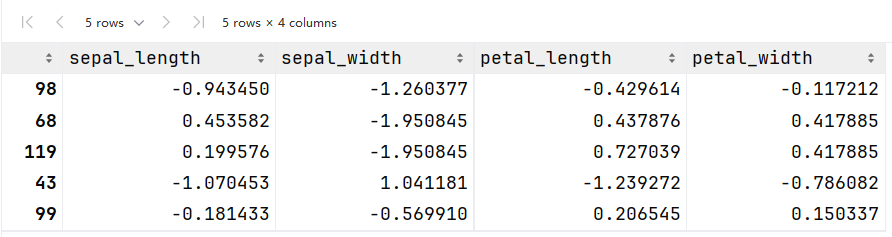


Fig.

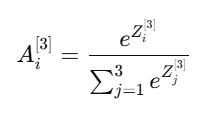
* 1. **Neural Network Design**

**FFNN (Feed-Forward Neural Network):**

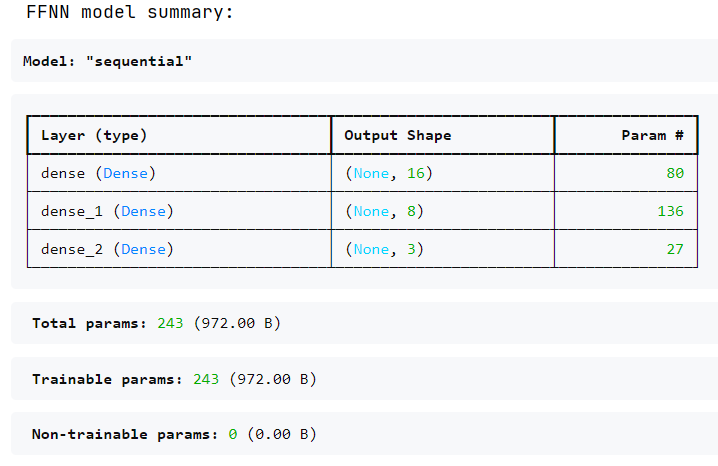
First, import TensorFlow and Keras components. TensorFlow is a deep learning framework developed by Google that can efficiently handle large-scale neural network training. (TensorFlow, 2024) Keras, on the other hand, is a high-level API interface to TensorFlow (Keras, 2024), which is used in this project to build FFNN models.

The Iris dataset has four features, so we define “Input” as the four input layers and “Dense” as the Fully Connected Layer, first, we try to give the first Fully Connected Layer 16 neurons and the activation function Rectified Linear Unit. The connected layer is made up of 16 neurons, and the activation function is a Rectified Linear Unit. The second Hidden layers has 8 neurons and the output layer has 3 neurons corresponding to the three classes of the iris dataset (Setosa, Versicolor, Virginia); later studies can try to modify the number of neurons in the Hidden layers to observe the change in model accuracy.

The first hidden layer is passed through (16, 4) substitute into the formula：, The ReLU activation function is *,* After two layers of calculation, the third output layer uses Softmax activation function:

**

In addition, using the TensorFlow/Keras library .summary() to print the structure of the FFNN model, the printout can include the number of parameters and the model name. So use this function to more intuitively observe the image of the model. The printout is as follows:



The first layer of the computational process is ‘Param=weights+biases(4\*16+16=80)’; The second layer (16\*8+8=136); The third layer (8\*3+3=27) All parameters are trainable, which means that these data will be optimized during gradient descent.

The iris dataset has only 150 pieces of data and a total of 243 parameter calculations, which is a small number of parameters and suitable for small classification tasks; if the parameter amount is too large, the results of overfitting may occur for small data, and it will take up too much GPU memory and RAM in the process of calculation, slowing down the calculation speed and leading to video memory overflow.

**Convolutional Neural Network (CNN):**

CNN is a powerful machine learning model, its main function is to automatically learn the features and spatial structure of the database, and the advantage is that there is no need to manually extract features, in the recognition of images and classification tasks are very effective, Convolutional Neural Network pipeline process for input data into the Convolutional Neural Network pipeline process is the input data into the convolution layer processing, then through the pooling layer processing, and finally through the fully connected layer for computation processing.(Shiri, F. M, 2023)

Iris is a classification problem although it is not an image and the dataset is small, so this project still tries to use the CNN model to simulate iris as a 2\*2 image matrix for processing, firstly, import the Conv1D, MaxPooling1D, Flatten; One- dimensional convolution layer to create 32 convolution kernels, the size of the convolution kernel is set up as 2, the computer will look at the feature relationship between the two iris, and then reduce the dimensionality by MaxPooling, while retaining the important features, and then expand the output of the convolution layer into a one-dimensional by Flatten, and finally by Fully Connected Layer to create 64 neurons to perform the classification task, and finally the output of 3 neurons corresponding to the three species of iris.

As with FFNN, use '.summary' to output the model structure. Observe the trained CNN model, as shown in the figure:

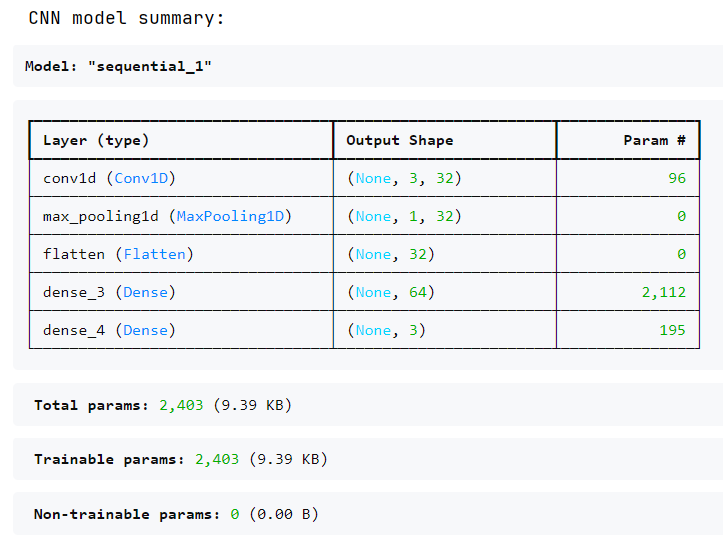


Fig.

It can be seen that CNN computed the iris dataset in 5 layers, and 2403 params were trained; the training process of the Conv1D layer is param=(1×2×32)+32=64+32=96; the Dense param=(32×64)+64=2048+64=2112; and the output layer Dense param = (64×3)+3=192+3=195.

To summarize, in the iris train set, the number of has more than that of CNN training parameters than FFNN, and the computation is a simpler computation process.

* 1. **Model Training**

FFNN Model Compilation:

First use .compile to specify the optimizer, loss function and evaluation metrics; first use Adam (Adaptive Moment Estimation) as the Optimizer; Adam is faster than the traditional SGD training, and the effect of noise on the model will be less, Adam is a very small memory, to carry out Adam is an optimizer that requires very little memory to perform efficient stochastic optimization.(DP Kingma, 2015)Therefore, without specifying an optimizer, the weights between the data cannot be calculated and model training cannot be performed.

After that, specify the Loss Function as 'categorical\_crossentropy', it is a Multi-class cross-entropy loss function, which is very suitable for dealing with the one-hot coded problem, The formula is: Predict the true class probability of the model (1 or 0).

The core step of training the model using .fit() was then used, consisting of several steps of forward propagation, calculating loss, backpropagation, and updating weights (TensorFlow, 2024), designing the number of training rounds to be 150 (the value can be adjusted to choose the optimal value if there is an over- or under-fit) and using 16 sample updates each time to maintain stability, The FFNN model training process As shown in the figure:

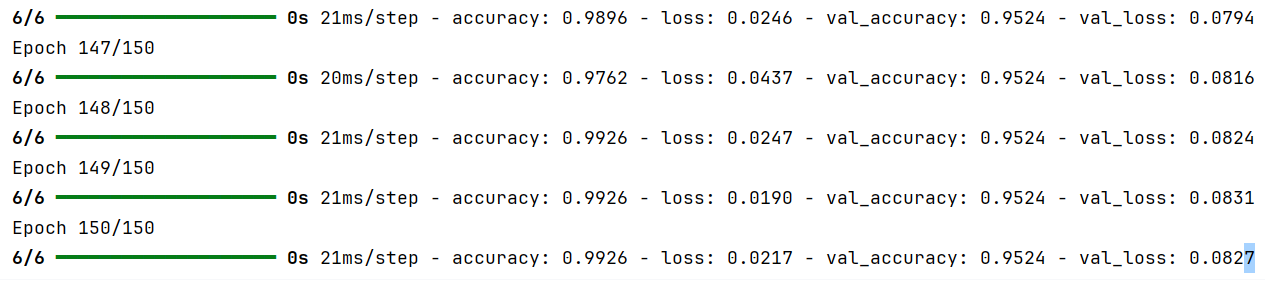


Fig.

CNN Model Compilation:

The CNN training process will have one more step than the FFNN training, "adjusting the data format", because the CNN deals with 3D data, which corresponds to (samples, timesteps, and features) in the iris dataset, this project chooses .reshape() to let Numpy deal with the 2D arrays; and then and the FFNN model the same choice of 150 rounds of training; The CNN model training process.

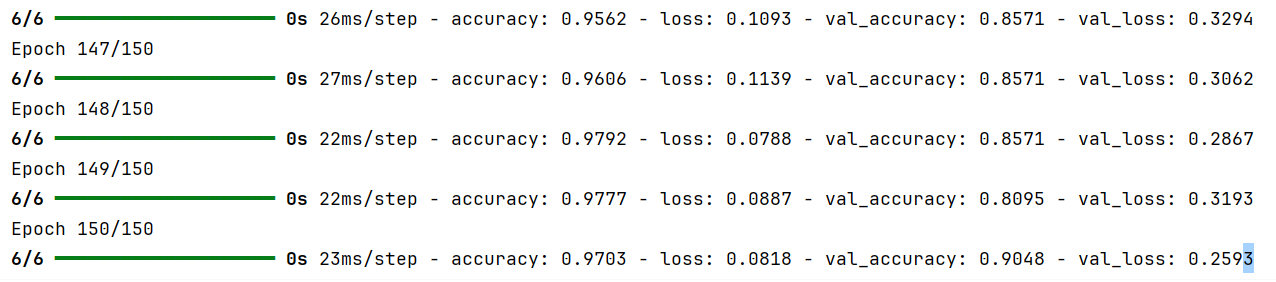


Fig.

* 1. **Evaluation**

The evaluation process of the model I first, using matplotlib to plot the Accuary curve and Loss curve of the CNN model and FFNN model, you can observe the process of the two models 150 training process Accuary and Loss value changes, observe this figure can understand the learning status of the two models, Accuary changes as follows:

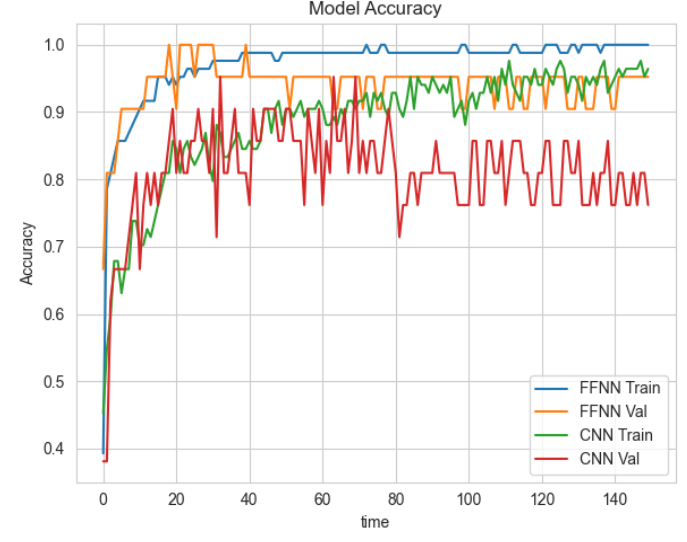


Fig.

It can be seen that (1) the accuracy of the CNN model is steadily increasing in both the test set and the training set, then it shows that there is no overfitting and underfitting, and the model is in a normal learning state, and (2) the stability of the FFNN model is higher than that of the CNN.

Model Loss change:



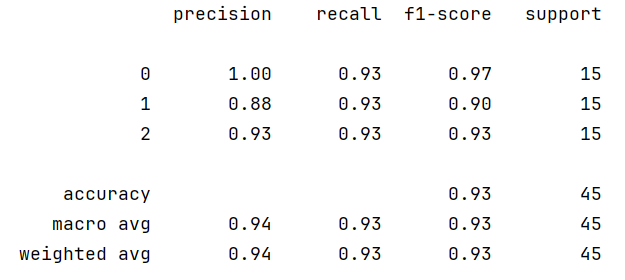
Fig.

Based on the images, the following results can be observed: (1) the Loss of the training set as well as the test set of both FFNN and CNN are steadily decreasing indicating that the models are converging normally; (2) the yellow line of FFNN shows signs of rising, indicating that if the number of training times is increased, there may be signs of overfitting; and (3) the oscillation degree of the CNN's loss curves is higher than that of FFNN indicating that the data distribution stabilization, is worse than FFNN.

Model evaluation:

In the process of model evaluation, the use of Accuracy as the only indicator to evaluate the model is not enough to fully reflect the model problem but also needs to introduce precision, recall and F-score to determine the model's previous competitive situation. (Goutte & Gaussier, 2005)

The steps to visualize the model report are: at the end of the model, the classification\_report is imported from the sklearn library; including .predict, which predicts the probability distribution; and .argmax is used to transform the 1, 0, and 2 in the test set to one-hot coding, and from this the model score belonging to the IRIS multi-class classification problem can be computed report. *FFNN model as show in the figure:*



Use a.heatmap to generate a FFNN heatmap:

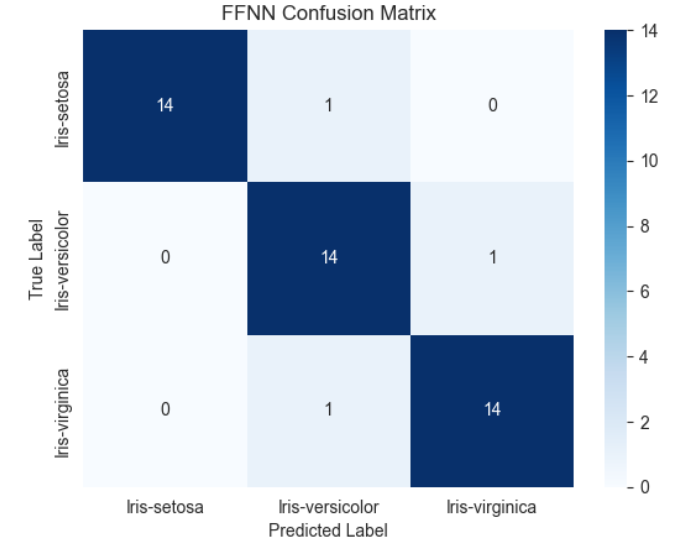


Fig.

According to the report, the following information can be analyzed, the FFNN model has an accuracy of 1 in calculating the Iris Setosa category, indicating that no misclassification has occurred, and the recall, and f1-score are above 90%, indicating that there is no overfitting, and the Accuracy, Macro avg. and Weighted avg. Reflect on the overall average of the data. The overall average, which is above 93%, indicates that the model is performing well.

It can also be seen from the heat map that FFNN has judged that there are only 3 misclassifications of iris species, one for each species.

CNN assessment results :

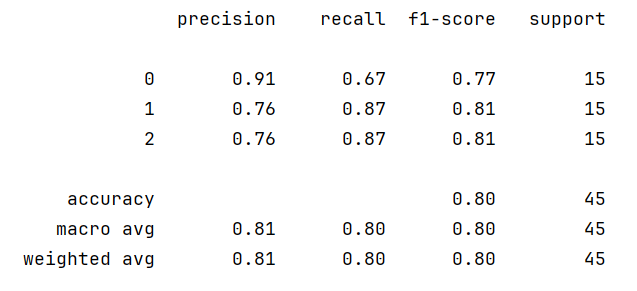


Fig.

CNN Confusion Matrix:

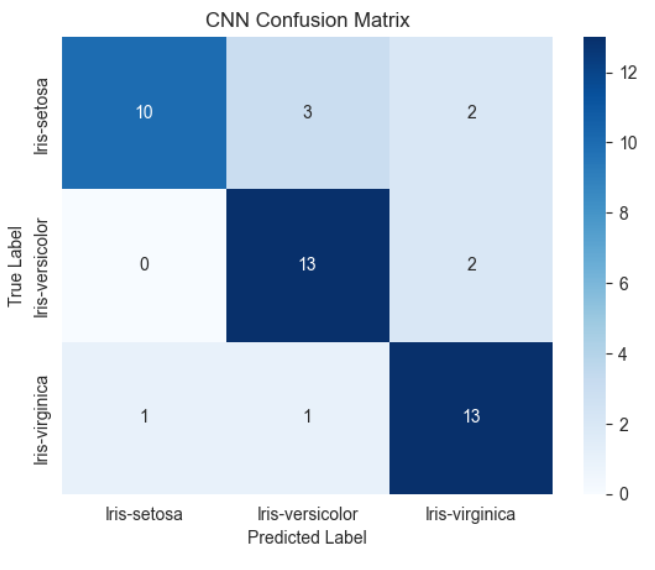


Fig.

From the CNN assessment results and Confusion Matrix, it can be observed that (1) the model trained by CNN is around 80% for each performance metric when judging the iris dataset, and (2) when judging the 45 model samples, the CNN model mistakenly judges Setosa as Versicolor three times, and Virginia two times; and judged Versicolor as Virginia two times.

1. **Conclusion**
   1. **Comparison of models**

In the real-world dataset, for the multi-class classification problem similar to iris, the model of FFNN is better than that of CNN, after the model construction, in the iris dataset, FFNN outperforms the model of CNN by about 10%, and the judgment accuracy is also higher.FFNN also performs a simpler computational process than CNN in the processing of data. The FFNN also performs a simpler computational process than CNN, and it can be judged that CNN may be slightly weak in dealing with simple classification problems.

In the classification problem according to Susithra et al. who used FFNN and CNN to recognize the sentiment problem, it can be seen that (Susithra et al., 2022) CNN is better suited to deal with spatial issues while FFNN is more versatile but not as good as CNN in dealing with spatial issues; FFNN is global and does not have as much spatial structure in iris so it is more comfortable to handle; on the contrary, CNN is less effective in dealing with the IRIS dataset because of the small number of data which is only 150;

Therefore, for the characteristics of CNN, when dealing with complex spatial problems, such as (recognizing soccer balls, recognizing basketball) these problems can be used CNN, which can automatically recognize the local features of the image; while for simple classification problems without spatial characteristics and with few parameters, FFNN can be used.

* 1. **Potential direction**

Among the two Deep learning models, they all have different applicable scenarios, so it is reasonable to use different models to calculate different scenarios, FFNN as a general model, when dealing with regression and classification problems, the neurons of the data, connect the data for this kind of data with no obvious structure, for example, Azkarategi et al. used the FFNN to predict Photovoltaic power in the next 24 hours, the conclusion of the coefficient R reached 0.941 and 0.94. FFNN to predict Photovoltaic power in the next 24 hours, the conclusion coefficient R reached 0.941 and 0.94, they made this tool, which can help people to predict the PV, dealing with energy-renewable.（Azkarategi et al. 2022)

The CNN model can help the data, by layering to reduce the number of parameters, and gradually extract the association in the data, in modern times, face to deal with CV problems, such as recognizing cats and dogs, and recognizing soccer, can be handled by the characteristics of CNN to deal with the spatial problem, and to better cope with the complexity of real-world data.

For example, Cherpanath et al. Their team used CNN for food recognition as well as calorie prediction, classified images in multiple layers, extracted features, and finally achieved more than 90% accuracy results (Cherpanath et al., 2023), which helped people control their diet and maintain their calorie intake.

**References**

1. **Khemani, B., Patil, S., Kotecha, K., et al. (2024). A review of graph neural networks: Concepts, architectures, techniques, challenges, datasets, applications, and future directions.** Journal of Big Data, 11(1), 18. [https://doi.org/10.1186/s40537-023-00876-4](https://doi.org/10.1186/s40537-023-00876-4" \t "https://chat.deepseek.com/a/chat/s/_blank)
2. Fisher, R. (1936). Iris [Dataset]. UCI Machine Learning Repository. <https://doi.org/10.24432/C56C76.>
3. pandas development team. (2024). pandas documentation. <https://pandas.pydata.org/>
4. Anil, N. K., & Kaur, B. P. (2024). Analyzing sentiments with machine learning and deep learning through word embedding methods. In 2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC) (pp. 705–710). Gwalior, India. <https://doi.org/10.1109/AIC61668.2024.10730950>
5. Matplotlib Development Team. (2023). Matplotlib: Visualization with Python. Retrieved June 5, 2024, from <https://matplotlib.org>
6. Waskom, M. L. (2023). Seaborn: Statistical data visualization. Retrieved June 5, 2024, from [https://seaborn.pydata.org](https://seaborn.pydata.org/" \t "https://chat.deepseek.com/a/chat/s/_blank)
7. **Kirmemis, O., & Tekalp, A. M. (2018). Effect of training and test datasets on image restoration and super-resolution by deep learning.** In 2018 26th European Signal Processing Conference (EUSIPCO) (pp. 514–518). IEEE. [https://doi.org/10.23919/EUSIPCO.2018.8552961](https://doi.org/10.23919/EUSIPCO.2018.8552961" \t "https://chat.deepseek.com/a/chat/s/_blank)
8. Scikit-learn developers. (n.d.). Scikit-learn: Machine learning in Python. <https://scikit-learn.org>
9. TensorFlow. (2024). TensorFlow: An end-to-end open source machine learning platform. Retrieved February 19, 2025, from <https://www.tensorflow.org>
10. Keras. (2024). Keras documentation. Retrieved February 19, 2025, from <https://keras.io>
11. Shiri, F. M., Perumal, T., Mustapha, N., & Mohamed, R. (2023). A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU. arXiv preprint arXiv:2305.17473.
12. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
13. Goutte, C., Gaussier, E. (2005). A Probabilistic Interpretation of Precision, Recall and *F*-Score, with Implication for Evaluation. In: Losada, D.E., Fernández-Luna, J.M. (eds) Advances in Information Retrieval. ECIR 2005. Lecture Notes in Computer Science, vol 3408. Springer, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-540-31865-1_25>
14. Susithra, N., Rajalakshmi, K., Ashwath, P., Ajay, B., Rohit, D., & Stewaugh, S. (2022). Speech based emotion recognition and gender identification using FNN and CNN models. 2022 3rd International Conference for Emerging Technology (INCET) (pp. 1–6). <https://doi.org/10.1109/INCET54531.2022.9824908>
15. Azkarategi, G. E., Zapirain Zuazo, I., Ruiz, H. C., Hernández, J., Ugartemendia, J. J., & Curea, O. (2022). Photovoltaic power forecast for the next 24 hours with an analytical model and a FFNN model. 2022 IEEE International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM), 1-6. <https://doi.org/10.1109/CISTEM55808.2022.10044029>
16. Cherpanath, E. D., Nasreen, P. R. F., Pradeep, K., Menon, M., & Jayanthi, V. S. (2023). Food image recognition and calorie prediction using Faster R-CNN and Mask R-CNN. 2023 9th International Conference on Smart Computing and Communications (ICSCC), 83–89. <https://doi.org/10.1109/ICSCC59169.2023.10335053>