```
import pandas as pd
import numpy as np
import os

# Plotting libraries
import matplotlib.pyplot as plt


# SKLearn libraries
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split

# Tensorflow libraries
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models import Sequential


# Data file path
FILE_PATH = '/content/IRIS.csv'

# Dataframe from csv file
iris_data = pd.read_csv(FILE_PATH, header=0)


iris_data.info()
print("=="*40)
iris_data.head(10)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
================================================================================
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
5	5.4	3.9	1.7	0.4	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa
7	5.0	3.4	1.5	0.2	Iris-setosa
8	4.4	2.9	1.4	0.2	Iris-setosa
9	4.9	3.1	1.5	0.1	Iris-setosa

preparing dataset

```
X = iris_data.loc[:, iris_data.columns != 'species']
y = iris_data.loc[:, ['species']]


y_enc = LabelEncoder().fit_transform(y)
# Converting the label into a matrix form
y_label = tf.keras.utils.to_categorical(y_enc)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_label.py:116: DataConversionWarning: A column-vector y was passed when a
  y = column_or_1d(y, warn=True)
```

```
# X_train_full, X_test, y_train_full, y_test = train_test_split(X, y_label, test_size=0.15)

# Validation set
# X_train, X_valid, y_train, y_valid = train_test_split(X_train_full, y_train_full)

X_train, X_test, y_train, y_test = train_test_split(X, y_label, test_size=0.3)


print(f"Train shape : {X_train.shape}, Y Train : {y_train.shape}")
print(X_train.shape[1:])
```

```
 Train shape : (105, 4), Y Train : (105, 3)
 (4,)
```

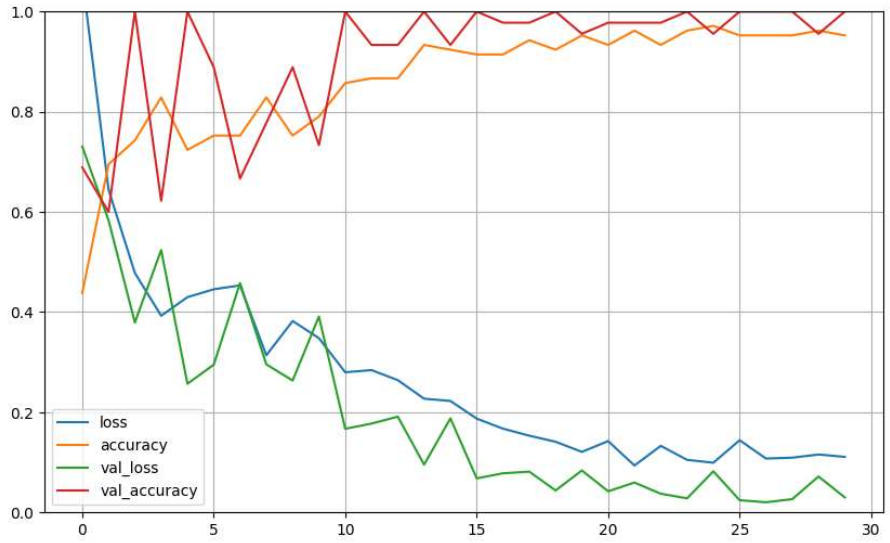
```
def get_model():
    model = Sequential([
        keras.layers.Input(shape=X_train.shape[1:]),
        keras.layers.Dense(1000, activation='relu'),
        keras.layers.Dense(500, activation='relu',),
        keras.layers.Dense(300, activation='relu'),
        keras.layers.Dropout(0.2),
        keras.layers.Dense(3, activation='softmax')
    ])

    return model
    model.summary()


model = get_model()

# Compile the model
model.compile(optimizer='adam',
              loss=keras.losses.CategoricalCrossentropy(),
             metrics=['accuracy'])


history = model.fit(X_train, y_train, epochs=30, validation_data=(X_test, y_test), verbose=1)
```

```
 Epoch 2/30
 4/4 [==============================] - 0s 22ms/step - loss: 0.6442 - accuracy: 0.6952 - val_loss: 0.5832 - val_accuracy: 0.6000
 Epoch 3/30
 4/4 [==============================] - 0s 26ms/step - loss: 0.4782 - accuracy: 0.7429 - val_loss: 0.3791 - val_accuracy: 1.0000
 Epoch 4/30
 4/4 [==============================] - 0s 31ms/step - loss: 0.3926 - accuracy: 0.8286 - val_loss: 0.5239 - val_accuracy: 0.6222
 Epoch 5/30
 4/4 [==============================] - 0s 26ms/step - loss: 0.4299 - accuracy: 0.7238 - val_loss: 0.2569 - val_accuracy: 1.0000
 Epoch 6/30
 4/4 [==============================] - 0s 26ms/step - loss: 0.4455 - accuracy: 0.7524 - val_loss: 0.2950 - val_accuracy: 0.8889
 Epoch 7/30
 4/4 [==============================] - 0s 27ms/step - loss: 0.4533 - accuracy: 0.7524 - val_loss: 0.4579 - val_accuracy: 0.6667
 Epoch 8/30
 4/4 [==============================] - 0s 22ms/step - loss: 0.3138 - accuracy: 0.8286 - val_loss: 0.2956 - val_accuracy: 0.7778
 Epoch 9/30
 4/4 [==============================] - 0s 27ms/step - loss: 0.3822 - accuracy: 0.7524 - val_loss: 0.2633 - val_accuracy: 0.8889
 Epoch 10/30
 4/4 [==============================] - 0s 27ms/step - loss: 0.3479 - accuracy: 0.7905 - val_loss: 0.3911 - val_accuracy: 0.7333
 Epoch 11/30
 4/4 [==============================] - 0s 21ms/step - loss: 0.2799 - accuracy: 0.8571 - val_loss: 0.1670 - val_accuracy: 1.0000
 Epoch 12/30
 4/4 [==============================] - 0s 21ms/step - loss: 0.2841 - accuracy: 0.8667 - val_loss: 0.1774 - val_accuracy: 0.9333
 Epoch 13/30
 4/4 [==============================] - 0s 22ms/step - loss: 0.2641 - accuracy: 0.8667 - val_loss: 0.1912 - val_accuracy: 0.9333
 Epoch 14/30
 4/4 [==============================] - 0s 28ms/step - loss: 0.2271 - accuracy: 0.9333 - val_loss: 0.0955 - val_accuracy: 1.0000
 Epoch 15/30
 4/4 [==============================] - 0s 21ms/step - loss: 0.2226 - accuracy: 0.9238 - val_loss: 0.1878 - val_accuracy: 0.9333
 Epoch 16/30
 4/4 [==============================] - 0s 26ms/step - loss: 0.1873 - accuracy: 0.9143 - val_loss: 0.0680 - val_accuracy: 1.0000
 Epoch 17/30
 4/4 [==============================] - 0s 24ms/step - loss: 0.1673 - accuracy: 0.9143 - val_loss: 0.0782 - val_accuracy: 0.9778
 Epoch 18/30
 4/4 [==============================] - 0s 27ms/step - loss: 0.1533 - accuracy: 0.9429 - val_loss: 0.0815 - val_accuracy: 0.9778
 Epoch 19/30
 4/4 [==============================] - 0s 26ms/step - loss: 0.1411 - accuracy: 0.9238 - val_loss: 0.0439 - val_accuracy: 1.0000
 Epoch 20/30
 4/4 [==============================] - 0s 23ms/step - loss: 0.1209 - accuracy: 0.9524 - val_loss: 0.0838 - val_accuracy: 0.9556
 Epoch 21/30
 4/4 [==============================] - 0s 28ms/step - loss: 0.1424 - accuracy: 0.9333 - val_loss: 0.0422 - val_accuracy: 0.9778
 Epoch 22/30
```

```
4/4 [==============================] - 0s 27ms/step - loss: 0.1329 - accuracy: 0.9333 - val_loss: 0.0373 - val_accuracy: 0.9778
Epoch 24/30
4/4 [==============================] - 0s 21ms/step - loss: 0.1050 - accuracy: 0.9619 - val_loss: 0.0281 - val_accuracy: 1.0000
Epoch 25/30
4/4 [==============================] - 0s 27ms/step - loss: 0.0992 - accuracy: 0.9714 - val_loss: 0.0821 - val_accuracy: 0.9556
Epoch 26/30
4/4 [==============================] - 0s 27ms/step - loss: 0.1442 - accuracy: 0.9524 - val_loss: 0.0244 - val_accuracy: 1.0000
Epoch 27/30
4/4 [==============================] - 0s 24ms/step - loss: 0.1076 - accuracy: 0.9524 - val_loss: 0.0203 - val_accuracy: 1.0000
Epoch 28/30
4/4 [==============================] - 0s 27ms/step - loss: 0.1093 - accuracy: 0.9524 - val_loss: 0.0265 - val_accuracy: 1.0000
Epoch 29/30
4/4 [==============================] - 0s 22ms/step - loss: 0.1156 - accuracy: 0.9619 - val_loss: 0.0718 - val_accuracy: 0.9556
Epoch 30/30
4/4 [==============================] - 0s 26ms/step - loss: 0.1109 - accuracy: 0.9524 - val_loss: 0.0299 - val_accuracy: 1.0000
```

performance monitor

```
pd.DataFrame(history.history).plot(figsize=(10,6))
plt.grid(True)
plt.gca().set_ylim(0, 1)
plt.show()
```




```
new_data, y_actual = X_test[:3], y_test[:3]

y_proba = model.predict(new_data)

print(f"Actual data : {y_actual}")

for pred in y_proba:
    print(np.argmax(pred))
```

```
1/1 [==============================] - 0s 112ms/step
Actual data : [[0. 1. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
1
1
2
```