

Họ và tên: Quách Xuân Phúc

MSV: B20DCCN513

1. Load data và output 5 dòng thể hiện các thuộc tính

```
[1]: import pandas as pd
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

data = pd.read_csv('pima-indians-diabetes.csv', header = None)

print(data.head(5))
```

	0	1	2	3	4	5	6	7	8
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

2. Sử dụng mô hình 1

```
X = data.iloc[:, 0:8]
y = data.iloc[:, 8]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

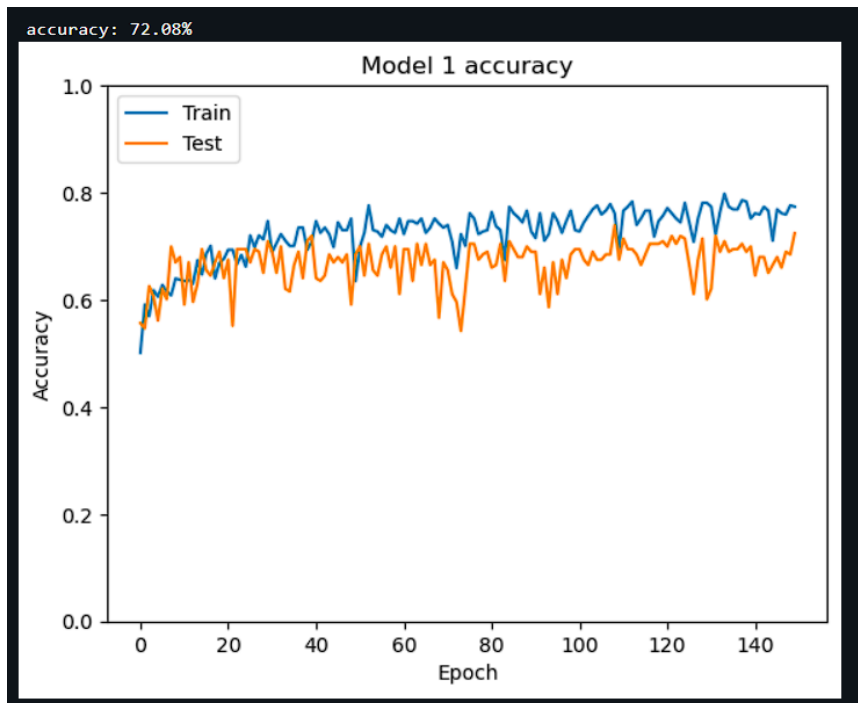
# Model 1
model = Sequential()
model.add(Dense(15, input_dim=8, activation='relu'))
model.add(Dense(10, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

history = model.fit(X_train, y_train, validation_split=0.33, epochs=150, batch_size=10, verbose=0)

# Evaluate the model
scores = model.evaluate(X_test, y_test)
print("\n%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))

# Plot accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model 1 accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.ylim(0, 1)
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```



3. Sử dụng mô hình 2

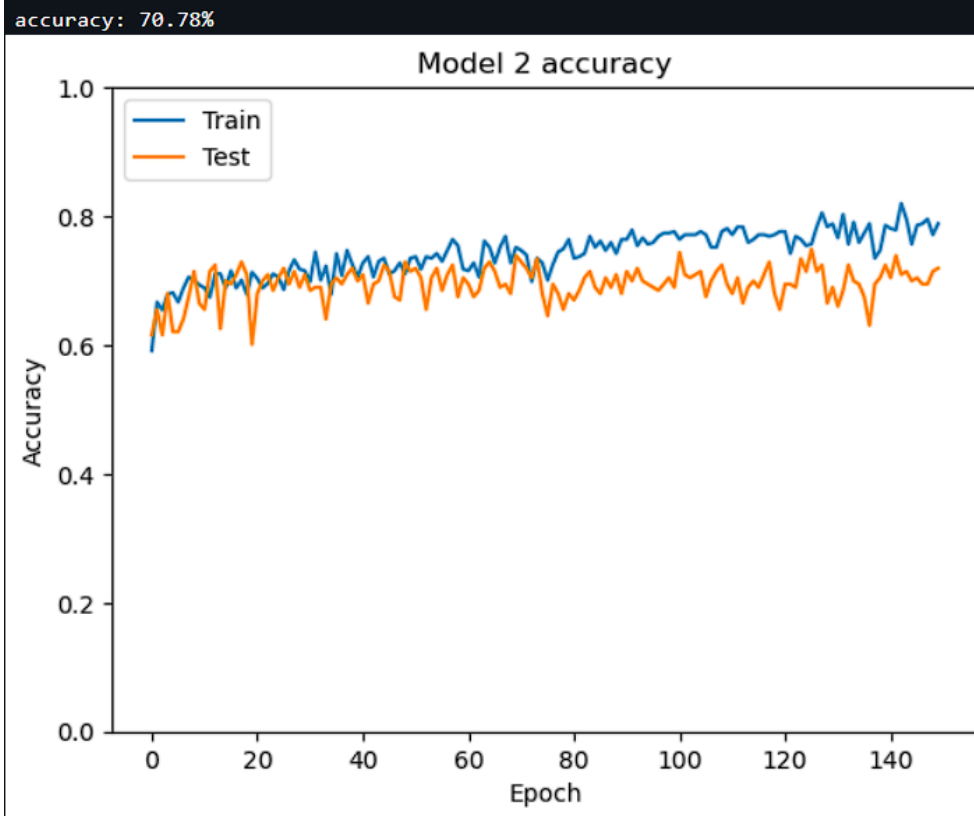
```
# Model 2
model = Sequential()
model.add(Dense(20, input_dim=8, activation='relu'))
model.add(Dense(15, activation='relu'))
model.add(Dense(10, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# Fit the model
history = model.fit(X_train, y_train, validation_split=0.33, epochs=150, batch_size=10, verbose=0)

# Evaluate the model
scores = model.evaluate(X_test, y_test)
print("\n%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))

# Plot accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model 2 accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.ylim(0, 1)
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```



4. Thêm dropout 10%

- Add Dropout 10% Model 1

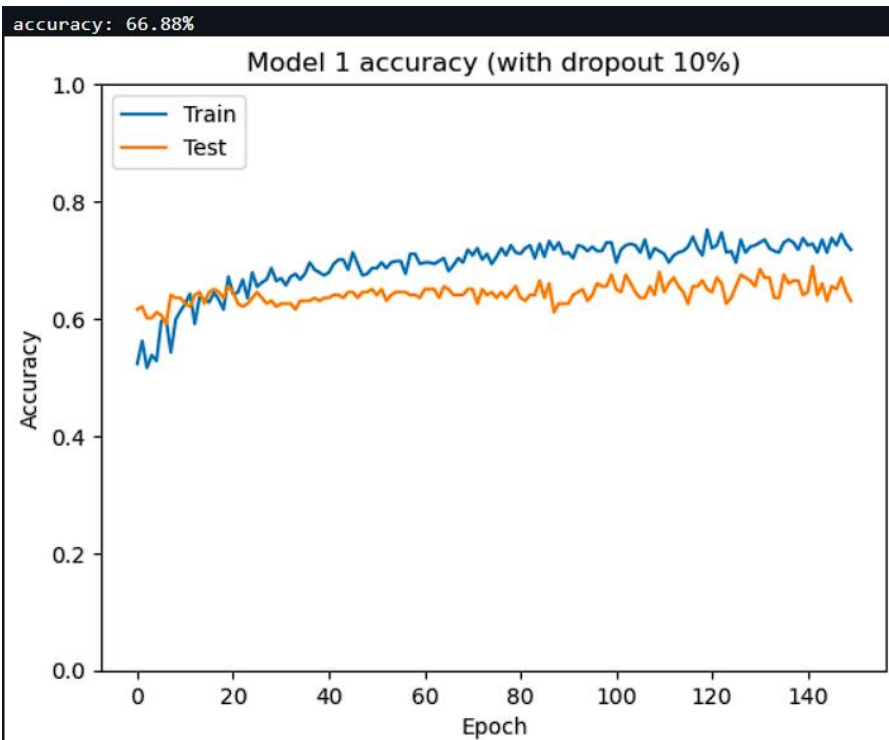
```
# Add Dropout 10% Model 1
model = Sequential()
model.add(Dense(15, input_dim=8, activation='relu'))
model.add(Dropout(0.1))
model.add(Dense(10, activation='relu'))
model.add(Dropout(0.1))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

history = model.fit(X_train, y_train, validation_split=0.33, epochs=150, batch_size=10, verbose=0)

# Evaluate the model
scores = model.evaluate(X_test, y_test)
print("\n%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))

# Plot accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model 1 accuracy (with dropout 10%)')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.ylim(0, 1)
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```



- Add Dropout 10% Model 2

```
# Add Dropout 10% Model 2
model = Sequential()
model.add(Dense(20, input_dim=8, activation='relu'))
model.add(Dropout(0.1))
model.add(Dense(15, activation='relu'))
model.add(Dropout(0.1))
model.add(Dense(10, activation='relu'))
model.add(Dropout(0.1))
model.add(Dense(8, activation='relu'))
model.add(Dropout(0.1))
model.add(Dense(1, activation='sigmoid'))

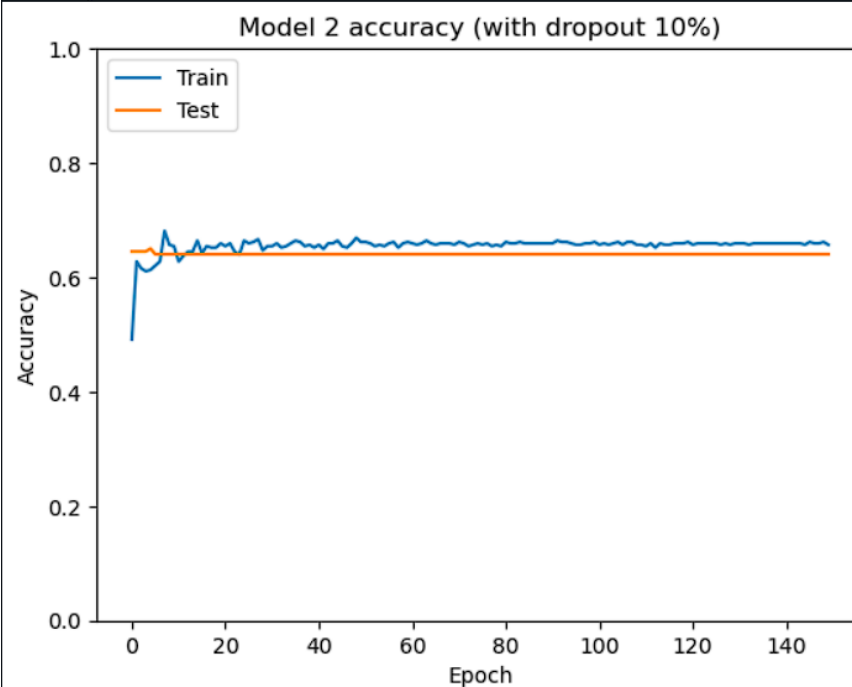
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# Fit the model
history = model.fit(X_train, y_train, validation_split=0.33, epochs=150, batch_size=10, verbose=0)

# Evaluate the model
scores = model.evaluate(X_test, y_test)
print("\n%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))

# Plot accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model 2 accuracy (with dropout 10%)')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.ylim(0, 1)
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```

accuracy: 64.29%



5. Nhận xét về độ chính xác của 4 mô hình:

- Model 1:

- Trên biểu đồ đánh giá độ chính xác, độ chính xác trên tập huấn luyện và tập kiểm tra có xu hướng tăng dần khi số epoch tăng. Tuy nhiên, có dấu hiệu overfitting khi độ chính xác trên tập huấn luyện cao hơn độ chính xác trên tập kiểm tra từ một số epoch nhất định. Điều này cho thấy mô hình có khả năng học quá mức từ dữ liệu huấn luyện và không khái quát hóa tốt trên dữ liệu mới.
- Độ chính xác trên tập kiểm tra đạt một giá trị ổn định sau một số epoch nhất định, và không có sự tăng trưởng đáng kể sau đó. Điều này có thể cho thấy mô hình đã đạt được sự hội tụ và không cần tiếp tục huấn luyện để cải thiện thêm.

- Model 2:

- Tương tự như mô hình 1 không sử dụng Dropout, mô hình 2 cũng có dấu hiệu overfitting khi độ chính xác trên tập huấn luyện cao hơn độ chính xác trên tập kiểm tra sau một số epoch nhất định.
- Độ chính xác trên tập kiểm tra không tăng quá nhanh và không đạt độ chính xác tốt như trên tập huấn luyện. Điều này cho thấy mô hình không khái quát hóa tốt trên dữ liệu mới.

- Model 1 sử dụng Dropout 10%:

- Sự thêm lớp Dropout vào mô hình 1 giúp giảm hiện tượng overfitting. Độ chính xác trên tập kiểm tra tăng lên và gần hơn với độ chính xác trên tập huấn luyện, cho thấy mô hình có khả năng khái quát hóa tốt hơn trên dữ liệu mới.
- Độ chính xác trên tập kiểm tra đạt một giá trị ổn định sau một số epoch nhất định, tương tự như mô hình không sử dụng Dropout. Tuy nhiên, mô hình có xu hướng hội tụ nhanh hơn và đạt độ chính xác tương đối cao.

- Model 2 sử dụng Dropout 10%:

- Sự thêm lớp Dropout vào mô hình 2 cũng giúp giảm overfitting và cải thiện khả năng khái quát hóa của mô hình.
- Độ chính xác trên tập huấn luyện và tập kiểm tra có xu hướng tăng dần khi số epoch tăng, và đạt một độ chính xác tương đối cao. Độ chính xác trên tập kiểm tra gần hơn với độ chính xác trên tập huấn luyện, cho thấy khả năng khái quát hóa tốt hơn của mô hình.