Psychoinformatics - Week 11 (Exercises)

by 彭瑋琳 (r11227120@ntu.edu.tw)

In [ ]:
```
%config IPCompleter.greedy=True
%matplotlib inline

from numpy import *
from matplotlib.pyplot import *
from IPython.display import *
import warnings
warnings.simplefilter('ignore', DeprecationWarning)
from sklearn import *

iris = datasets.load_iris()
X=iris.data; Y=iris.target
```

1 Performance Tuning of a Neural Net (8 points)

1.0 Baseline Performance

SVM can reach an classifcation accuracy ~ 8x% correct for the HARD Iris problem.

In [ ]:
```
sss=model_selection.StratifiedShuffleSplit(n_splits=3,test_size=0.1) # (45+45+45) vs. (5+5+5)
model=svm.SVC(C=10)
acc=[]
for train_index, test_index in sss.split(X, Y): # 3-fold cross-validation
    X_train, X_test = X[train_index], X[test_index]
    Y_train, Y_test = Y[train_index], Y[test_index]
    model.fit(X_train[:,0:2],Y_train) #training
    acc.append(model.predict(X_test[:,0:2])==Y_test) # testing
print(np.mean(acc))
```

```
0.8444444444444444
```

1.1 Tuning your ANN (4 points)

Tune your model hyperparameters (# of layers, # of units in each layer, activation function, optimizer, epochs, batch_size, etc.) to see if you can push your ANN performance up to ~9x% correct for the HARD iris problem.

In [ ]:
```
from keras.models import Sequential,clone_model
from keras.layers import Dense
from keras.utils import to_categorical
from tqdm import tqdm,trange
```

In [ ]:
```
model = Sequential()
model.add(Dense(units=3, activation='relu'))
model.add(Dense(units=3, activation='softmax'))

acc=[]
for train_index, test_index in sss.split(X, Y): # 3-fold cross-validation
    X_train, X_test = X[train_index], X[test_index]
    Y_train, Y_test = Y[train_index], Y[test_index]
    new_model=clone_model(model) # Otherwise the old model will keep learning
    new_model.compile(loss='categorical_crossentropy',optimizer='sgd',metrics=['accuracy'])
    new_model.fit(X_train[:,0:2], to_categorical(Y_train), epochs=50, batch_size=15, verbose = 0)
    acc.append(np.mean(argmax(new_model.predict(X_test[:,0:2]),1)==Y_test)) # testing
print(acc, np.mean(acc))
```

```
1/1 [==============================] - 0s 90ms/step
1/1 [==============================] - 0s 62ms/step
1/1 [==============================] - 0s 67ms/step
[0.3333333333333333, 0.7333333333333333, 0.6666666666666666] 0.5777777777777778
```

In [ ]:
```
def evaluate(nlayer=1, unit1=3, unit2=3, optimizer='sgd', act_function='relu',epo=50, batch=15):

    model = Sequential()
    model.add(Dense(unit1, activation=act_function))
    for i in range(nlayer):
        model.add(Dense(unit2, activation='softmax'))

    acc=[]
    for train_index, test_index in sss.split(X, Y): # 3-fold cross-validation
        X_train, X_test = X[train_index], X[test_index]
        Y_train, Y_test = Y[train_index], Y[test_index]
        new_model=clone_model(model) # Otherwise the old model will keep learning
        new_model.compile(loss='categorical_crossentropy',optimizer= optimizer,metrics=['accuracy'])
        new_model.fit(X_train[:,0:2], to_categorical(Y_train), epochs= epo, batch_size= batch, verbose = 0)
        acc.append(np.mean(argmax(new_model.predict(X_test[:,0:2]),1)==Y_test)) # testing
    return np.mean(acc)
```

Examine- Number of Layers

In [ ]:
```
# optimizer = sgd

num = 0
best_acc = 0
numLayer = []
acc = []
for i in trange(10):
    a = evaluate(nlayer=i)
    numLayer.append(i)
    acc.append(a)
    if a > best_acc:
        num = i
        best_acc = a
print('\n number of layers:',num, '\n Best accuracy:',best_acc)
```

```
  0%|          | 0/10 [00:00<?, ?it/s]
1/1 [==============================] - 0s 90ms/step
WARNING:tensorflow:5 out of the last 5 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7f4579f6a7a0> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has reduce_retracing=True option that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for  more details.
1/1 [==============================] - 0s 52ms/step
WARNING:tensorflow:6 out of the last 6 calls to <function Model.make_predict_function.<locals>.predict_function at 0x7f4579f6b370> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has reduce_retracing=True option that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for  more details.
1/1 [==============================] - 0s 70ms/step
 10%|▉         | 1/10 [00:06<00:54,  6.05s/it]
1/1 [==============================] - 0s 88ms/step
1/1 [==============================] - 0s 62ms/step
1/1 [==============================] - 0s 57ms/step
 20%|██        | 2/10 [00:11<00:45,  5.68s/it]
1/1 [==============================] - 0s 77ms/step
1/1 [==============================] - 0s 78ms/step
1/1 [==============================] - 0s 82ms/step
 30%|███       | 3/10 [00:17<00:40,  5.74s/it]
1/1 [==============================] - 0s 139ms/step
1/1 [==============================] - 0s 129ms/step
1/1 [==============================] - 0s 81ms/step
 40%|████      | 4/10 [00:23<00:34,  5.76s/it]
1/1 [==============================] - 0s 99ms/step
1/1 [==============================] - 0s 128ms/step
1/1 [==============================] - 0s 120ms/step
 50%|█████     | 5/10 [00:28<00:28,  5.80s/it]
1/1 [==============================] - 0s 129ms/step
1/1 [==============================] - 0s 162ms/step
1/1 [==============================] - 0s 167ms/step
 60%|██████    | 6/10 [00:36<00:24,  6.24s/it]
1/1 [==============================] - 0s 120ms/step
1/1 [==============================] - 0s 120ms/step
1/1 [==============================] - 0s 117ms/step
 70%|███████   | 7/10 [00:42<00:19,  6.45s/it]
1/1 [==============================] - 0s 138ms/step
1/1 [==============================] - 0s 223ms/step
1/1 [==============================] - 0s 152ms/step
 80%|████████  | 8/10 [00:51<00:14,  7.27s/it]
1/1 [==============================] - 0s 155ms/step
1/1 [==============================] - 0s 149ms/step
1/1 [==============================] - 0s 151ms/step
 90%|█████████ | 9/10 [00:58<00:07,  7.09s/it]
1/1 [==============================] - 0s 247ms/step
1/1 [==============================] - 0s 157ms/step
1/1 [==============================] - 0s 171ms/step
100%|██████████| 10/10 [01:08<00:00,  6.86s/it]
 number of layers: 1
 Best accuracy: 0.6222222222222222
```

In [ ]:
```
# optimizer = adam

num = 0
best_acc = 0
numLayer = []
acc = []
for i in trange(10):
    a = evaluate(nlayer=i, optimizer = 'Adam')
    numLayer.append(i)
    acc.append(a)
    if a > best_acc:
        num = i
        best_acc = a
print('\n number of layers:',num, '\n Best accuracy:',best_acc)
```

```
  0%|          | 0/10 [00:00<?, ?it/s]
1/1 [==============================] - 0s 91ms/step
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 48ms/step
 10%|▉         | 1/10 [00:05<00:49,  5.46s/it]
1/1 [==============================] - 0s 96ms/step
1/1 [==============================] - 0s 60ms/step
1/1 [==============================] - 0s 59ms/step
 20%|██        | 2/10 [00:11<00:47,  5.99s/it]
1/1 [==============================] - 0s 76ms/step
1/1 [==============================] - 0s 73ms/step
1/1 [==============================] - 0s 73ms/step
 30%|███       | 3/10 [00:17<00:41,  5.98s/it]
1/1 [==============================] - 0s 132ms/step
1/1 [==============================] - 0s 90ms/step
1/1 [==============================] - 0s 105ms/step
 40%|████      | 4/10 [00:26<00:41,  6.91s/it]
1/1 [==============================] - 0s 112ms/step
1/1 [==============================] - 0s 96ms/step
1/1 [==============================] - 0s 149ms/step
 50%|█████     | 5/10 [00:33<00:35,  7.16s/it]
1/1 [==============================] - 0s 121ms/step
1/1 [==============================] - 0s 119ms/step
1/1 [==============================] - 0s 127ms/step
 60%|██████    | 6/10 [00:43<00:32,  8.21s/it]
1/1 [==============================] - 0s 132ms/step
1/1 [==============================] - 0s 137ms/step
1/1 [==============================] - 0s 126ms/step
 70%|███████   | 7/10 [00:54<00:26,  8.85s/it]
1/1 [==============================] - 0s 137ms/step
1/1 [==============================] - 0s 142ms/step
1/1 [==============================] - 0s 132ms/step
 80%|████████  | 8/10 [01:04<00:18,  9.43s/it]
1/1 [==============================] - 0s 148ms/step
1/1 [==============================] - 0s 147ms/step
1/1 [==============================] - 0s 149ms/step
 90%|█████████ | 9/10 [01:14<00:09,  9.38s/it]
1/1 [==============================] - 0s 186ms/step
1/1 [==============================] - 0s 165ms/step
1/1 [==============================] - 0s 165ms/step
100%|██████████| 10/10 [01:26<00:00,  8.68s/it]
 number of layers: 2
 Best accuracy: 0.6
```

Examine- units in each layer

In [ ]:

```
num = 0
best_acc = 0
unitNum = []
acc = []
for i in trange(10):
    a = evaluate(unit1=i, optimizer = 'Adam')
    unitNum.append(i)
    acc.append(a)
    if a > best_acc:
        num = i
        best_acc = a
print('\n Unit number in each layer:',num, '\n Best accuracy:',best_acc)
```

```
  0%|          | 0/10 [00:00<?, ?it/s]
1/1 [==============================] - 0s 90ms/step
1/1 [==============================] - 0s 87ms/step
1/1 [==============================] - 0s 61ms/step
 10%|▉         | 1/10 [00:06<00:55,  6.14s/it]
1/1 [==============================] - 0s 61ms/step
1/1 [==============================] - 0s 63ms/step
1/1 [==============================] - 0s 73ms/step
 20%|██        | 2/10 [00:10<00:41,  5.17s/it]
1/1 [==============================] - 0s 60ms/step
1/1 [==============================] - 0s 59ms/step
1/1 [==============================] - 0s 94ms/step
 30%|███       | 3/10 [00:15<00:35,  5.09s/it]
1/1 [==============================] - 0s 87ms/step
1/1 [==============================] - 0s 61ms/step
1/1 [==============================] - 0s 62ms/step
 40%|████      | 4/10 [00:21<00:32,  5.44s/it]
1/1 [==============================] - 0s 61ms/step
1/1 [==============================] - 0s 63ms/step
1/1 [==============================] - 0s 61ms/step
 50%|█████     | 5/10 [00:26<00:25,  5.08s/it]
1/1 [==============================] - 0s 58ms/step
1/1 [==============================] - 0s 83ms/step
1/1 [==============================] - 0s 86ms/step
 60%|██████    | 6/10 [00:31<00:21,  5.28s/it]
```

```
1/1 [==============================] - 0s 64ms/step
1/1 [==============================] - 0s 60ms/step
1/1 [==============================] - 0s 62ms/step
 70%|███████   | 7/10 [00:36<00:15,  5.14s/it]
1/1 [==============================] - 0s 59ms/step
1/1 [==============================] - 0s 59ms/step
1/1 [==============================] - 0s 67ms/step
 80%|████████  | 8/10 [00:41<00:10,  5.08s/it]
1/1 [==============================] - 0s 65ms/step
1/1 [==============================] - 0s 89ms/step
1/1 [==============================] - 0s 61ms/step
 90%|█████████ | 9/10 [00:47<00:05,  5.35s/it]
1/1 [==============================] - 0s 61ms/step
1/1 [==============================] - 0s 62ms/step
1/1 [==============================] - 0s 61ms/step
100%|██████████| 10/10 [00:52<00:00,  5.29s/it]
 Unit number in each layer: 7
 Best accuracy: 0.6
```

In [ ]:
```
num = 0
best_acc = 0
unitNum = []
acc = []
for i in trange(10):
    a = evaluate(unit1=i, nlayer = 2, optimizer = 'Adam')
    unitNum.append(i)
    acc.append(a)
    if a > best_acc:
        num = i
        best_acc = a
print('\n Unit number in each layer:',num, '\n Best accuracy:',best_acc)
```

```
  0%|          | 0/10 [00:00<?, ?it/s]
1/1 [==============================] - 0s 72ms/step
1/1 [==============================] - 0s 112ms/step
1/1 [==============================] - 0s 112ms/step
 10%|▉         | 1/10 [00:07<01:10,  7.84s/it]
1/1 [==============================] - 0s 73ms/step
1/1 [==============================] - 0s 83ms/step
1/1 [==============================] - 0s 77ms/step
 20%|██        | 2/10 [00:14<00:56,  7.08s/it]
1/1 [==============================] - 0s 75ms/step
1/1 [==============================] - 0s 104ms/step
1/1 [==============================] - 0s 104ms/step
 30%|███       | 3/10 [00:20<00:46,  6.70s/it]
1/1 [==============================] - 0s 75ms/step
1/1 [==============================] - 0s 74ms/step
1/1 [==============================] - 0s 78ms/step
 40%|████      | 4/10 [00:26<00:39,  6.52s/it]
1/1 [==============================] - 0s 72ms/step
1/1 [==============================] - 0s 81ms/step
1/1 [==============================] - 0s 110ms/step
 50%|█████     | 5/10 [00:32<00:31,  6.31s/it]
1/1 [==============================] - 0s 109ms/step
1/1 [==============================] - 0s 78ms/step
1/1 [==============================] - 0s 79ms/step
 60%|██████    | 6/10 [00:39<00:26,  6.55s/it]
1/1 [==============================] - 0s 87ms/step
1/1 [==============================] - 0s 74ms/step
1/1 [==============================] - 0s 90ms/step
 70%|███████   | 7/10 [00:45<00:19,  6.39s/it]
1/1 [==============================] - 0s 106ms/step
1/1 [==============================] - 0s 80ms/step
1/1 [==============================] - 0s 77ms/step
 80%|████████  | 8/10 [00:53<00:13,  6.62s/it]
1/1 [==============================] - 0s 74ms/step
1/1 [==============================] - 0s 77ms/step
1/1 [==============================] - 0s 74ms/step
 90%|█████████ | 9/10 [00:58<00:06,  6.28s/it]
1/1 [==============================] - 0s 106ms/step
1/1 [==============================] - 0s 103ms/step
1/1 [==============================] - 0s 73ms/step
100%|██████████| 10/10 [01:05<00:00,  6.55s/it]
 Unit number in each layer: 6
 Best accuracy: 0.6444444444444444
```

Examine - batch size

In [ ]:
```
num = 0
best_acc = 0
batchsize = []
acc = []
for i in trange(100):
    a = evaluate(batch=i, unit1 = 8, nlayer = 1, optimizer = 'Adam')
    batchsize.append(i)
```

```
    acc.append(a)
    if a > best_acc:
        num = i
        best_acc = a
print('\n Batch size:',num, '\n Best accuracy:',best_acc)
```

```
  0%|          | 0/100 [00:00<?, ?it/s]
1/1 [==============================] - 0s 60ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 48ms/step
  1%|          | 1/100 [00:03<06:17,  3.82s/it]
1/1 [==============================] - 0s 70ms/step
1/1 [==============================] - 0s 91ms/step
1/1 [==============================] - 0s 66ms/step
  2%||          | 2/100 [00:35<33:14, 20.35s/it]
1/1 [==============================] - 0s 46ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 58ms/step
  3%|▏         | 3/100 [00:58<34:45, 21.50s/it]
1/1 [==============================] - 0s 73ms/step
1/1 [==============================] - 0s 62ms/step
1/1 [==============================] - 0s 67ms/step
  4%|▏         | 4/100 [01:15<31:26, 19.65s/it]
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 48ms/step
  5%|▏         | 5/100 [01:26<26:06, 16.49s/it]
1/1 [==============================] - 0s 64ms/step
1/1 [==============================] - 0s 74ms/step
1/1 [==============================] - 0s 64ms/step
  6%|▏         | 6/100 [01:38<23:45, 15.17s/it]
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 78ms/step
  7%|▏         | 7/100 [01:48<20:50, 13.45s/it]
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 54ms/step
  8%|▏         | 8/100 [01:58<18:58, 12.38s/it]
1/1 [==============================] - 0s 72ms/step
1/1 [==============================] - 0s 61ms/step
1/1 [==============================] - 0s 53ms/step
  9%|▏         | 9/100 [02:07<17:08, 11.30s/it]
1/1 [==============================] - 0s 69ms/step
1/1 [==============================] - 0s 61ms/step
1/1 [==============================] - 0s 60ms/step
 10%|▏         | 10/100 [02:13<14:25,  9.61s/it]
1/1 [==============================] - 0s 75ms/step
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 48ms/step
 11%|▏         | 11/100 [02:21<13:32,  9.13s/it]
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 47ms/step
 12%|▏         | 12/100 [02:27<11:50,  8.07s/it]
1/1 [==============================] - 0s 74ms/step
1/1 [==============================] - 0s 46ms/step
1/1 [==============================] - 0s 49ms/step
 13%|▏         | 13/100 [02:34<11:10,  7.70s/it]
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 51ms/step
 14%|▏         | 14/100 [02:41<10:44,  7.49s/it]
1/1 [==============================] - 0s 77ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 53ms/step
 15%|▏         | 15/100 [02:47<10:00,  7.06s/it]
1/1 [==============================] - 0s 60ms/step
1/1 [==============================] - 0s 62ms/step
1/1 [==============================] - 0s 61ms/step
 16%|▏         | 16/100 [02:52<09:00,  6.43s/it]
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 71ms/step
1/1 [==============================] - 0s 68ms/step
 17%|▏         | 17/100 [02:58<08:48,  6.37s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 60ms/step
 18%|▏         | 18/100 [03:04<08:22,  6.13s/it]
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 69ms/step
 19%|▏         | 19/100 [03:10<08:16,  6.12s/it]
1/1 [==============================] - 0s 73ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 49ms/step
```

```
 20%|██        | 20/100 [03:15<08:00,  6.01s/it]
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 49ms/step
 21%|██        | 21/100 [03:19<07:06,  5.40s/it]
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 70ms/step
 22%|██        | 22/100 [03:24<06:49,  5.25s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 47ms/step
 23%|██        | 23/100 [03:30<07:04,  5.52s/it]
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 47ms/step
 24%|██        | 24/100 [03:34<06:23,  5.04s/it]
1/1 [==============================] - 0s 46ms/step
1/1 [==============================] - 0s 68ms/step
1/1 [==============================] - 0s 49ms/step
 25%|██        | 25/100 [03:40<06:30,  5.20s/it]
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 48ms/step
 26%|██        | 26/100 [03:44<05:55,  4.80s/it]
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 47ms/step
 27%|██        | 27/100 [03:48<05:30,  4.53s/it]
1/1 [==============================] - 0s 61ms/step
1/1 [==============================] - 0s 89ms/step
1/1 [==============================] - 0s 66ms/step
 28%|██        | 28/100 [03:53<05:46,  4.81s/it]
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 52ms/step
 29%|██        | 29/100 [03:57<05:21,  4.53s/it]
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 46ms/step
 30%|███       | 30/100 [04:01<05:01,  4.30s/it]
1/1 [==============================] - 0s 75ms/step
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 55ms/step
 31%|███       | 31/100 [04:08<05:54,  5.14s/it]
1/1 [==============================] - 0s 61ms/step
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 49ms/step
 32%|███       | 32/100 [04:12<05:24,  4.78s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 56ms/step
 33%|███       | 33/100 [04:16<05:03,  4.53s/it]
1/1 [==============================] - 0s 76ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 47ms/step
 34%|███       | 34/100 [04:21<05:20,  4.86s/it]
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 57ms/step
 35%|███       | 35/100 [04:25<04:48,  4.44s/it]
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 47ms/step
 36%|███       | 36/100 [04:29<04:31,  4.24s/it]
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 74ms/step
1/1 [==============================] - 0s 86ms/step
 37%|███       | 37/100 [04:34<04:40,  4.45s/it]
1/1 [==============================] - 0s 46ms/step
1/1 [==============================] - 0s 46ms/step
1/1 [==============================] - 0s 46ms/step
 38%|███       | 38/100 [04:37<04:23,  4.25s/it]
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 50ms/step
 39%|███       | 39/100 [04:41<04:06,  4.04s/it]
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 58ms/step
1/1 [==============================] - 0s 71ms/step
 40%|████      | 40/100 [04:46<04:20,  4.34s/it]
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 53ms/step
 41%|████      | 41/100 [04:50<04:08,  4.21s/it]
```

```
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 50ms/step
 42%|████      | 42/100 [04:53<03:54,  4.04s/it]
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 47ms/step
 43%|████      | 43/100 [04:57<03:43,  3.93s/it]
1/1 [==============================] - 0s 77ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 50ms/step
 44%|████      | 44/100 [05:02<04:01,  4.31s/it]
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 55ms/step
 45%|████      | 45/100 [05:06<03:48,  4.16s/it]
1/1 [==============================] - 0s 60ms/step
1/1 [==============================] - 0s 64ms/step
1/1 [==============================] - 0s 61ms/step
 46%|████      | 46/100 [05:10<03:36,  4.01s/it]
1/1 [==============================] - 0s 59ms/step
1/1 [==============================] - 0s 72ms/step
1/1 [==============================] - 0s 75ms/step
 47%|████      | 47/100 [05:14<03:39,  4.14s/it]
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 49ms/step
 48%|████      | 48/100 [05:18<03:28,  4.02s/it]
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 51ms/step
 49%|████      | 49/100 [05:22<03:20,  3.93s/it]
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 82ms/step
1/1 [==============================] - 0s 86ms/step
 50%|████      | 50/100 [05:28<03:55,  4.70s/it]
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 51ms/step
 51%|████      | 51/100 [05:32<03:41,  4.53s/it]
1/1 [==============================] - 0s 57ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 52ms/step
 52%|████      | 52/100 [05:36<03:24,  4.25s/it]
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 73ms/step
 53%|████      | 53/100 [05:40<03:16,  4.18s/it]
1/1 [==============================] - 0s 73ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 51ms/step
 54%|████      | 54/100 [05:44<03:17,  4.28s/it]
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 48ms/step
 55%|████      | 55/100 [05:48<03:02,  4.05s/it]
1/1 [==============================] - 0s 66ms/step
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 53ms/step
 56%|████      | 56/100 [05:52<02:51,  3.90s/it]
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 70ms/step
1/1 [==============================] - 0s 76ms/step
 57%|████      | 57/100 [05:56<02:53,  4.04s/it]
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 51ms/step
 58%|████      | 58/100 [06:00<02:44,  3.92s/it]
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 49ms/step
 59%|████      | 59/100 [06:03<02:38,  3.86s/it]
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 71ms/step
 60%|████      | 60/100 [06:07<02:33,  3.83s/it]
1/1 [==============================] - 0s 71ms/step
1/1 [==============================] - 0s 71ms/step
1/1 [==============================] - 0s 47ms/step
 61%|████      | 61/100 [06:11<02:35,  4.00s/it]
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 48ms/step
 62%|████      | 62/100 [06:15<02:26,  3.86s/it]
```

```
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 52ms/step
 63%|██████    | 63/100 [06:18<02:18,  3.73s/it]
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 71ms/step
1/1 [==============================] - 0s 74ms/step
 64%|██████    | 64/100 [06:23<02:18,  3.85s/it]
1/1 [==============================] - 0s 70ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 48ms/step
 65%|██████    | 65/100 [06:27<02:19,  3.98s/it]
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 59ms/step
 66%|██████    | 66/100 [06:31<02:12,  3.91s/it]
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 48ms/step
 67%|██████    | 67/100 [06:34<02:01,  3.70s/it]
1/1 [==============================] - 0s 75ms/step
1/1 [==============================] - 0s 69ms/step
1/1 [==============================] - 0s 48ms/step
 68%|██████    | 68/100 [06:38<02:06,  3.95s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 47ms/step
 69%|██████    | 69/100 [06:41<01:54,  3.68s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 48ms/step
 70%|██████    | 70/100 [06:44<01:43,  3.47s/it]
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 47ms/step
 71%|██████    | 71/100 [06:47<01:36,  3.33s/it]
1/1 [==============================] - 0s 69ms/step
1/1 [==============================] - 0s 78ms/step
1/1 [==============================] - 0s 60ms/step
 72%|██████    | 72/100 [06:54<01:58,  4.22s/it]
1/1 [==============================] - 0s 59ms/step
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 51ms/step
 73%|██████    | 73/100 [06:57<01:45,  3.92s/it]
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 50ms/step
 74%|██████    | 74/100 [07:00<01:36,  3.70s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 74ms/step
1/1 [==============================] - 0s 70ms/step
 75%|██████    | 75/100 [07:04<01:31,  3.66s/it]
1/1 [==============================] - 0s 79ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 56ms/step
 76%|██████    | 76/100 [07:08<01:34,  3.93s/it]
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 49ms/step
 77%|██████    | 77/100 [07:11<01:24,  3.69s/it]
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 52ms/step
 78%|██████    | 78/100 [07:14<01:17,  3.50s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 71ms/step
1/1 [==============================] - 0s 73ms/step
 79%|██████    | 79/100 [07:18<01:16,  3.64s/it]
1/1 [==============================] - 0s 78ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 48ms/step
 80%|██████    | 80/100 [07:22<01:11,  3.56s/it]
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 52ms/step
 81%|██████    | 81/100 [07:25<01:04,  3.40s/it]
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 51ms/step
 82%|██████    | 82/100 [07:28<00:59,  3.28s/it]
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 46ms/step
1/1 [==============================] - 0s 75ms/step
 83%|██████    | 83/100 [07:31<00:56,  3.29s/it]
```

```
1/1 [==============================] - 0s 73ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 48ms/step
 84%|████████  | 84/100 [07:35<00:56,  3.51s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 53ms/step
 85%|████████  | 85/100 [07:38<00:50,  3.34s/it]
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 48ms/step
 86%|████████  | 86/100 [07:41<00:45,  3.26s/it]
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 71ms/step
 87%|████████  | 87/100 [07:44<00:43,  3.31s/it]
1/1 [==============================] - 0s 78ms/step
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 50ms/step
 88%|████████  | 88/100 [07:49<00:43,  3.60s/it]
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 49ms/step
 89%|████████  | 89/100 [07:52<00:37,  3.45s/it]
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 50ms/step
 90%|████████  | 90/100 [07:55<00:33,  3.32s/it]
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 71ms/step
 91%|████████  | 91/100 [07:58<00:30,  3.36s/it]
1/1 [==============================] - 0s 74ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 49ms/step
 92%|████████  | 92/100 [08:02<00:28,  3.57s/it]
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 48ms/step
 93%|████████  | 93/100 [08:05<00:23,  3.39s/it]
1/1 [==============================] - 0s 58ms/step
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 53ms/step
 94%|████████  | 94/100 [08:08<00:19,  3.27s/it]
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 81ms/step
 95%|████████  | 95/100 [08:11<00:16,  3.21s/it]
1/1 [==============================] - 0s 68ms/step
1/1 [==============================] - 0s 85ms/step
1/1 [==============================] - 0s 47ms/step
 96%|████████  | 96/100 [08:16<00:14,  3.56s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 46ms/step
 97%|████████  | 97/100 [08:19<00:10,  3.40s/it]
1/1 [==============================] - 3s 3s/step
1/1 [==============================] - 0s 62ms/step
1/1 [==============================] - 0s 57ms/step
 98%|████████  | 98/100 [08:25<00:08,  4.09s/it]
1/1 [==============================] - 0s 76ms/step
1/1 [==============================] - 0s 77ms/step
1/1 [==============================] - 0s 52ms/step
 99%|████████  | 99/100 [08:29<00:04,  4.25s/it]
1/1 [==============================] - 0s 57ms/step
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 54ms/step
100%|████████| 100/100 [08:33<00:00,  5.13s/it]
 Batch size: 3
 Best accuracy: 0.7555555555555555
```

Examine - Epoch

In [ ]:

```
num = 0
best_acc = 0
epoch = []
acc = []
for i in trange(100):
    a = evaluate(epo=i, batch=4, unit1 = 8, nlayer = 1, optimizer = 'Adam')
    epoch.append(i)
    acc.append(a)
    if a > best_acc:
        num = i
        best_acc = a
print('\n Epoch:',num, '\n Best accuracy:',best_acc)
```

```
  0%|          | 0/100 [00:00<?, ?it/s]
1/1 [==============================] - 0s 94ms/step
1/1 [==============================] - 0s 100ms/step
1/1 [==============================] - 0s 91ms/step
  1%|          | 1/100 [00:00<01:06,  1.49it/s]
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 48ms/step
  2%||          | 2/100 [00:03<02:45,  1.69s/it]
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 50ms/step
  3%|▏         | 3/100 [00:05<03:22,  2.09s/it]
1/1 [==============================] - 0s 71ms/step
1/1 [==============================] - 0s 70ms/step
1/1 [==============================] - 0s 72ms/step
  4%|▏         | 4/100 [00:09<04:24,  2.76s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 49ms/step
  5%|▏         | 5/100 [00:12<04:31,  2.86s/it]
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 49ms/step
  6%|▏         | 6/100 [00:15<04:33,  2.91s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 47ms/step
  7%|▏         | 7/100 [00:18<04:35,  2.96s/it]
1/1 [==============================] - 0s 57ms/step
1/1 [==============================] - 0s 74ms/step
1/1 [==============================] - 0s 69ms/step
  8%|▏         | 8/100 [00:22<05:15,  3.43s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 57ms/step
  9%|▏         | 9/100 [00:26<05:23,  3.56s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 53ms/step
 10%|▏         | 10/100 [00:30<05:28,  3.65s/it]
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 70ms/step
 11%|▏         | 11/100 [00:34<05:31,  3.73s/it]
1/1 [==============================] - 0s 73ms/step
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 48ms/step
 12%|▏         | 12/100 [00:39<06:13,  4.24s/it]
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 46ms/step
 13%|▏         | 13/100 [00:43<06:00,  4.14s/it]
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 78ms/step
 14%|▏         | 14/100 [00:48<06:12,  4.33s/it]
1/1 [==============================] - 0s 67ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 47ms/step
 15%|▏         | 15/100 [00:54<06:48,  4.81s/it]
1/1 [==============================] - 0s 59ms/step
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 49ms/step
 16%|▏         | 16/100 [01:00<07:03,  5.04s/it]
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 70ms/step
1/1 [==============================] - 0s 53ms/step
 17%|▏         | 17/100 [01:06<07:18,  5.28s/it]
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 48ms/step
 18%|▏         | 18/100 [01:11<07:07,  5.21s/it]
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 79ms/step
 19%|▏         | 19/100 [01:16<07:02,  5.22s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 47ms/step
 20%|▏         | 20/100 [01:23<07:44,  5.81s/it]
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 48ms/step
 21%|▏         | 21/100 [01:29<07:34,  5.76s/it]
```

```
1/1 [==============================] - 0s 71ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 47ms/step
 22%|██        | 22/100 [01:36<08:04,  6.21s/it]
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 55ms/step
 23%|██        | 23/100 [01:42<07:46,  6.06s/it]
1/1 [==============================] - 0s 69ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 48ms/step
 24%|██        | 24/100 [01:49<08:12,  6.48s/it]
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 47ms/step
 25%|██        | 25/100 [01:55<07:51,  6.29s/it]
1/1 [==============================] - 0s 69ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 48ms/step
 26%|██        | 26/100 [02:02<08:12,  6.66s/it]
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 59ms/step
1/1 [==============================] - 0s 73ms/step
 27%|██        | 27/100 [02:10<08:22,  6.88s/it]
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 80ms/step
1/1 [==============================] - 0s 58ms/step
 28%|██        | 28/100 [02:22<10:09,  8.46s/it]
1/1 [==============================] - 0s 74ms/step
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 64ms/step
 29%|██        | 29/100 [02:31<10:13,  8.63s/it]
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 77ms/step
1/1 [==============================] - 0s 91ms/step
 30%|██        | 30/100 [02:41<10:31,  9.02s/it]
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 60ms/step
 31%|██        | 31/100 [02:49<10:02,  8.74s/it]
1/1 [==============================] - 0s 77ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 58ms/step
 32%|██        | 32/100 [02:59<10:13,  9.02s/it]
1/1 [==============================] - 0s 58ms/step
1/1 [==============================] - 0s 75ms/step
1/1 [==============================] - 0s 51ms/step
 33%|██        | 33/100 [03:09<10:28,  9.38s/it]
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 61ms/step
1/1 [==============================] - 0s 50ms/step
 34%|██        | 34/100 [03:18<10:11,  9.26s/it]
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 49ms/step
 35%|██        | 35/100 [03:30<10:48,  9.98s/it]
1/1 [==============================] - 0s 76ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 49ms/step
 36%|██        | 36/100 [03:39<10:27,  9.80s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 83ms/step
 37%|██        | 37/100 [03:48<10:03,  9.58s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 48ms/step
 38%|██        | 38/100 [03:57<09:35,  9.28s/it]
1/1 [==============================] - 0s 84ms/step
1/1 [==============================] - 0s 59ms/step
1/1 [==============================] - 0s 56ms/step
 39%|██        | 39/100 [04:09<10:17, 10.12s/it]
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 74ms/step
1/1 [==============================] - 0s 50ms/step
 40%|██        | 40/100 [04:19<10:06, 10.12s/it]
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 78ms/step
 41%|██        | 41/100 [04:28<09:43,  9.88s/it]
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 53ms/step
 42%|██        | 42/100 [04:38<09:34,  9.91s/it]
```

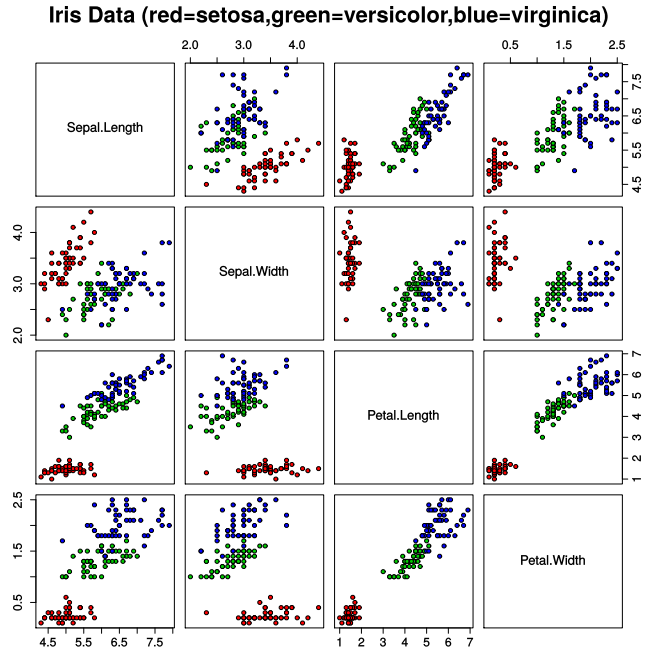
```
1/1 [==============================] - 0s 57ms/step
1/1 [==============================] - 0s 70ms/step
1/1 [==============================] - 0s 53ms/step
 43%|████      | 43/100 [04:49<09:38, 10.14s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 54ms/step
 44%|████      | 44/100 [05:00<09:50, 10.55s/it]
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 62ms/step
1/1 [==============================] - 0s 70ms/step
 45%|████      | 45/100 [05:09<09:14, 10.08s/it]
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 48ms/step
 46%|████      | 46/100 [05:19<09:00, 10.00s/it]
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 48ms/step
 47%|████      | 47/100 [05:31<09:24, 10.64s/it]
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 73ms/step
1/1 [==============================] - 0s 49ms/step
 48%|████      | 48/100 [05:42<09:10, 10.59s/it]
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 74ms/step
 49%|████      | 49/100 [05:51<08:43, 10.27s/it]
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 47ms/step
 50%|████      | 50/100 [06:04<09:07, 10.96s/it]
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 48ms/step
 51%|████      | 51/100 [06:16<09:15, 11.35s/it]
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 52ms/step
 52%|████      | 52/100 [06:28<09:18, 11.64s/it]
1/1 [==============================] - 0s 71ms/step
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 49ms/step
 53%|████      | 53/100 [06:41<09:22, 11.97s/it]
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 57ms/step
 54%|████      | 54/100 [06:56<09:51, 12.86s/it]
1/1 [==============================] - 0s 81ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 49ms/step
 55%|████      | 55/100 [07:10<09:51, 13.15s/it]
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 48ms/step
 56%|████      | 56/100 [07:21<09:14, 12.60s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 48ms/step
 57%|████      | 57/100 [07:34<09:01, 12.59s/it]
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 61ms/step
1/1 [==============================] - 0s 52ms/step
 58%|████      | 58/100 [07:49<09:21, 13.38s/it]
1/1 [==============================] - 0s 75ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 53ms/step
 59%|████      | 59/100 [08:05<09:35, 14.04s/it]
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 78ms/step
 60%|████      | 60/100 [08:20<09:37, 14.44s/it]
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 49ms/step
 61%|████      | 61/100 [08:38<10:02, 15.46s/it]
1/1 [==============================] - 0s 74ms/step
1/1 [==============================] - 0s 58ms/step
1/1 [==============================] - 0s 65ms/step
 62%|████      | 62/100 [08:57<10:31, 16.62s/it]
1/1 [==============================] - 0s 81ms/step
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 61ms/step
 63%|████      | 63/100 [09:13<10:07, 16.42s/it]
```

```
1/1 [==============================] - 0s 57ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 74ms/step
 64%|██████    | 64/100 [09:30<09:58, 16.63s/it]
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 49ms/step
 65%|██████    | 65/100 [09:45<09:27, 16.23s/it]
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 73ms/step
1/1 [==============================] - 0s 53ms/step
 66%|██████    | 66/100 [10:01<09:02, 15.94s/it]
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 58ms/step
1/1 [==============================] - 0s 52ms/step
 67%|██████    | 67/100 [10:17<08:46, 15.95s/it]
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 53ms/step
 68%|██████    | 68/100 [10:32<08:29, 15.92s/it]
1/1 [==============================] - 0s 69ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 53ms/step
 69%|██████    | 69/100 [10:49<08:15, 16.00s/it]
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 76ms/step
 70%|██████    | 70/100 [11:06<08:11, 16.39s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 50ms/step
 71%|██████    | 71/100 [11:20<07:39, 15.83s/it]
1/1 [==============================] - 0s 69ms/step
1/1 [==============================] - 0s 69ms/step
1/1 [==============================] - 0s 51ms/step
 72%|██████    | 72/100 [11:37<07:26, 15.96s/it]
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 78ms/step
1/1 [==============================] - 0s 47ms/step
 73%|██████    | 73/100 [11:52<07:06, 15.80s/it]
1/1 [==============================] - 0s 91ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 48ms/step
 74%|██████    | 74/100 [12:08<06:48, 15.71s/it]
1/1 [==============================] - 0s 79ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 53ms/step
 75%|██████    | 75/100 [12:23<06:32, 15.71s/it]
1/1 [==============================] - 0s 84ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 48ms/step
 76%|██████    | 76/100 [12:38<06:09, 15.40s/it]
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 49ms/step
 77%|██████    | 77/100 [12:53<05:48, 15.17s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 71ms/step
 78%|██████    | 78/100 [13:09<05:40, 15.47s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 59ms/step
1/1 [==============================] - 0s 77ms/step
 79%|██████    | 79/100 [13:24<05:25, 15.48s/it]
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 74ms/step
1/1 [==============================] - 0s 53ms/step
 80%|██████    | 80/100 [13:41<05:17, 15.85s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 67ms/step
1/1 [==============================] - 0s 51ms/step
 81%|██████    | 81/100 [13:58<05:05, 16.08s/it]
1/1 [==============================] - 0s 71ms/step
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 51ms/step
 82%|██████    | 82/100 [14:15<04:55, 16.40s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 71ms/step
 83%|██████    | 83/100 [14:31<04:39, 16.45s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 47ms/step
 84%|██████    | 84/100 [14:49<04:27, 16.69s/it]
```

```
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 70ms/step
1/1 [==============================] - 0s 50ms/step
 85%|████████▌ | 85/100 [15:04<04:05, 16.39s/it]
1/1 [==============================] - 0s 53ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 47ms/step
 86%|████████▌ | 86/100 [15:21<03:50, 16.50s/it]
1/1 [==============================] - 0s 67ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 55ms/step
 87%|████████▋ | 87/100 [15:37<03:33, 16.45s/it]
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 49ms/step
 88%|████████▊ | 88/100 [16:04<03:55, 19.60s/it]
1/1 [==============================] - 0s 76ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 50ms/step
 89%|████████▉ | 89/100 [16:21<03:24, 18.60s/it]
1/1 [==============================] - 0s 57ms/step
1/1 [==============================] - 0s 71ms/step
1/1 [==============================] - 0s 55ms/step
 90%|████████▉ | 90/100 [16:43<03:18, 19.81s/it]
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 54ms/step
1/1 [==============================] - 0s 74ms/step
 91%|█████████ | 91/100 [17:05<03:03, 20.40s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 51ms/step
 92%|█████████▏| 92/100 [17:28<02:48, 21.06s/it]
1/1 [==============================] - 0s 71ms/step
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 51ms/step
 93%|█████████▎| 93/100 [17:45<02:18, 19.79s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 52ms/step
 94%|█████████▍| 94/100 [18:08<02:04, 20.82s/it]
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 47ms/step
 95%|█████████▌| 95/100 [18:25<01:39, 19.89s/it]
1/1 [==============================] - 0s 77ms/step
1/1 [==============================] - 0s 47ms/step
1/1 [==============================] - 0s 73ms/step
 96%|█████████▌| 96/100 [18:43<01:17, 19.25s/it]
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 55ms/step
 97%|█████████▋| 97/100 [19:06<01:01, 20.41s/it]
1/1 [==============================] - 0s 69ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 68ms/step
 98%|█████████▊| 98/100 [19:24<00:39, 19.69s/it]
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 73ms/step
1/1 [==============================] - 0s 50ms/step
 99%|█████████▉| 99/100 [19:48<00:20, 20.80s/it]
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 47ms/step
100%|██████████| 100/100 [20:11<00:00, 12.12s/it]
Epoch: 62
Best accuracy: 0.8444444444444444
```

1.2 Is your (deep) network better than SVM? Why or why not? (4 points)

ANN的結果沒有比SVM更好，結果和SVM差不多。我嘗試tuning各種hyperparameter之後，最佳結果是正確率 84%，和SVM正確率 8x% 差不多。結果不好的原因可能是feature只有兩種，很難從這些特徵分辨出三種類別。從Iris Data來看，Sepal這兩個特徵的藍色和綠色data points都混在一起，所以可能是因為這兩個特徵沒辦法區辨y的第二和第三個類別導致模型表現不佳。



保持與上述ANN最佳一次tuning相同的hyperparameter，只改動feature的選擇和數量。
第一個分析從sepal改為petal的兩個特徵，模型訓練的結果是正確率提升到91%左右，第二個分析則使用X的四個特徵，模型訓練的結果正確率提升到95%左右，顯示特徵的選擇或數量的提升都可能影響模型的區辨力而有較好的表現。

In [ ]:
```
model = Sequential()
model.add(Dense(units=8, activation='relu'))
model.add(Dense(units=3, activation='softmax'))


acc=[]
for train_index, test_index in sss.split(X, Y): # 3-fold cross-validation
    X_train, X_test = X[train_index], X[test_index]
    Y_train, Y_test = Y[train_index], Y[test_index]
    new_model=clone_model(model) # Otherwise the old model will keep learning
    new_model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['accuracy'])
    new_model.fit(X_train[:,2:4], to_categorical(Y_train), epochs=60, batch_size=4, verbose = 0)
    acc.append(np.mean(argmax(new_model.predict(X_test[:,2:4]),1)==Y_test)) # testing
print('\n Accuracy:',np.mean(acc))
```

```
1/1 [==============================] - 0s 51ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 73ms/step

 Accuracy: 0.9111111111111111
```

In [28]:
```
model = Sequential()
model.add(Dense(units=8, activation='relu'))
model.add(Dense(units=3, activation='softmax'))


acc=[]
for train_index, test_index in sss.split(X, Y): # 3-fold cross-validation
    X_train, X_test = X[train_index], X[test_index]
    Y_train, Y_test = Y[train_index], Y[test_index]
    new_model=clone_model(model) # Otherwise the old model will keep learning
    new_model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['accuracy'])
    new_model.fit(X_train, to_categorical(Y_train), epochs=60, batch_size=4, verbose = 0)
    acc.append(np.mean(argmax(new_model.predict(X_test),1)==Y_test)) # testing
print('\n Accuracy:',np.mean(acc))
```

```
1/1 [==============================] - 0s 98ms/step
1/1 [==============================] - 0s 76ms/step
1/1 [==============================] - 0s 57ms/step

 Accuracy: 0.9555555555555556
```