Psychoinformatics - Week 11 (Exercises)

by boyonglin (r10945002@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) v
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))
```

```
0.8
```

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, Dropout
from keras.utils import to_categorical
from sklearn.preprocessing import StandardScaler
from keras.optimizers import Adam

# Normalize input features
```

```
scaler = StandardScaler()
X = scaler.fit_transform(X)

model = Sequential()
model.add(Dense(units=128, activation='relu', input_dim=2))
model.add(Dropout(0.5))
model.add(Dense(units=64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(units=32, 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(learning_rate=
    early_stopping = keras.callbacks.EarlyStopping(monitor='val_loss', patience=50,
    new_model.fit(X_train[:,0:2], to_categorical(Y_train), epochs=300, batch_size=1
    acc.append(np.mean(argmax(new_model.predict(X_test[:,0:2]),1)==Y_test)) # testi
print(acc, np.mean(acc))
```

```
1/1 [==============================] - 0s 118ms/step
1/1 [==============================] - 0s 118ms/step
1/1 [==============================] - 0s 86ms/step
1/1 [==============================] - 0s 106ms/step
[0.9333333333333333, 0.6666666666666666, 0.8] 0.8000000000000002
```

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

經過測試，深度神經架構只使用兩個特徵來訓練深度學習網路時，預測準確率難以高於SVM，可能是因為深度學習網路通常在高維度的數據上表現較好，能夠學習到數據的複雜特徵。如果只有兩個特徵，**數據的維度較低**，過擬合的風險可能增加，因為模型更容易記憶訓練資料。

相反地，SVM 是一種較為簡單的模型，對於數據維度和超參數的要求較低，因此在只有兩個特徵的數據上，SVM 的預測準確率可能會高於深度學習網路。