PS18

January 25, 2023

1 Car test-time prediction

1.1 Loading MB dataset

[1]:
```
import pandas as pd
data = pd.read_csv('mercedes_test.csv')
```

1.2 Data pre-processing

[2]:
```
# Choose categorical data columns
cf = data.select_dtypes(include=['object']).columns
# To change it into "categorical" data type
data[cf]=data[cf].astype('category')
# One hot encoding
data = pd.get_dummies(data)
# Obtain X from data (excluding 'ID' and 'y')
X_df = data.drop(['ID','y'],axis=1)
# Obtain y from data
y_df = data['y']

# Convert y_df into binary labels
import numpy as np
TF_vector= (y_df<np.median(y_df))
y_df=TF_vector.astype(float)

# Conver data frame into numpy array
X,y = X_df.values, y_df.values

# Split into train and test datasets
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.1,stratify=y)
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(3788, 563)
(421, 563)
```

```
(3788,)
(421,)
```

1.3 Use of TensorBoard for visualization

[3]:
```
# Load the TensorBoard notebook extension
%load_ext tensorboard
```

[4]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.regularizers import l2
from tensorflow.keras.callbacks import EarlyStopping, TensorBoard,␣
 ↪LearningRateScheduler
from tensorflow.keras.optimizers import Adam, SGD
import os
```

[5]:
```
# Construction of an DNN model
def create_model(n_layer=2, lambda_=0):
    model = Sequential()
    for i in range(n_layer-1):
        model.add(Dense(10, activation='relu',
                        kernel_regularizer=l2(lambda_),␣
 ↪bias_regularizer=l2(lambda_)))

    model.add(Dense(1, activation='sigmoid',
              kernel_regularizer=l2(lambda_), bias_regularizer=l2(lambda_)))
    return model
```

1.4 [1] 2-layer DNN, w/o regularization

[6]:
```
model = create_model(n_layer=2,lambda_=0)
opt = Adam(learning_rate=1e-3)
model.compile(optimizer=opt,
              loss='binary_crossentropy',
              metrics=['acc'])
logdir = os.path.join('logs','[1]no_regularization')
#file_name = 'no_regularization'
tb_callback = TensorBoard(logdir)
#tb_callback = TensorBoard(log_dir="logs\\{}".format(file_name))
es_callback = EarlyStopping(monitor='val_acc',patience=20)
hist = model.fit(X_train, y_train,
                 validation_split=1/9, epochs=100,
                 verbose=0,callbacks=[tb_callback, es_callback])
model.evaluate(X_test, y_test)
```

```
14/14 [==============================] - 0s 767us/step - loss: 0.3428 - acc:
0.8456
```

[6]: [0.342769980430603, 0.8456056714057922]

[7]:
```
%tensorboard --logdir=logs/
```

```
Reusing TensorBoard on port 6006 (pid 9224), started 0:54:21 ago. (Use '!kill␣
↪9224' to kill it.)
```

```
<IPython.core.display.HTML object>
```

1.5 [2] Number of layers (w/o regularization)

[8]:
```
n_layer_list = [1,2,3,4,5]

for n_layer in n_layer_list:
    model = create_model(n_layer=n_layer,lambda_=0)
    opt = Adam(learning_rate=1e-3)
    model.compile(optimizer=opt,
                  loss='binary_crossentropy',
                  metrics=['acc'])
    logdir = os.path.join('logs','[2]n_layer_wo_regularization','{}_layer'.
↪format(n_layer))
    tb_callback = TensorBoard(logdir)
    es_callback = EarlyStopping(monitor='val_acc', patience=20)
    hist = model.fit(X_train, y_train,
                  validation_split=1/9, epochs = 100,
                  verbose=0,callbacks=[tb_callback, es_callback])
    model.evaluate(X_test, y_test)
```

```
14/14 [==============================] - 0s 690us/step - loss: 0.3235 - acc:
0.8789
14/14 [==============================] - 0s 921us/step - loss: 0.3347 - acc:
0.8694
14/14 [==============================] - 0s 691us/step - loss: 0.3521 - acc:
0.8599
14/14 [==============================] - 0s 691us/step - loss: 0.3421 - acc:
0.8765
14/14 [==============================] - 0s 690us/step - loss: 0.3806 - acc:
0.8599
```

[9]:
```
%tensorboard --logdir=logs/[2]n_layer_wo_regularization
```

```
<IPython.core.display.HTML object>
```

1.6 [3] With regularization (2-layer DNN)

[10]:
```
lambda_list = [1e-3, 1e-2, 1e-1, 1, 10]
for lambda_ in lambda_list:
    model = create_model(n_layer=2, lambda_=lambda_)
    opt = Adam(learning_rate=1e-3)
    model.compile(optimizer=opt,
                  loss='binary_crossentropy',
                  metrics=['acc'])
    logdir = os.path.join('logs', '[3]2_layer_w_regularization', 'lambda_{}'.
 ↪format(lambda_))
    tb_callback = TensorBoard(logdir)
    es_callback = EarlyStopping(monitor='val_acc', patience=20)
    hist = model.fit(X_train, y_train,
                  validation_split=1/9, epochs = 100,
                  verbose=0, callbacks=[tb_callback, es_callback] )
    model.evaluate(X_test, y_test)
```

```
14/14 [==============================] - 0s 614us/step - loss: 0.3545 - acc:
0.8622
14/14 [==============================] - 0s 618us/step - loss: 0.3986 - acc:
0.8741
14/14 [==============================] - 0s 691us/step - loss: 0.6349 - acc:
0.8599
14/14 [==============================] - 0s 764us/step - loss: 0.6931 - acc:
0.5012
14/14 [==============================] - 0s 691us/step - loss: 0.6932 - acc:
0.5012
```

[11]:
```
%tensorboard --logdir=logs/[3]2_layer_w_regularization
```

```
Reusing TensorBoard on port 6006 (pid 28892), started 0:19:55 ago. (Use '!kill␣
 ↪28892' to kill it.)
```

```
<IPython.core.display.HTML object>
```

1.7 [4] Learning rate (2-layer DNN)

[13]:
```
lr_list = [1e-1, 1e-2, 1e-3, 1e-4]

for lr in lr_list:
    model = create_model(n_layer=2, lambda_=0)
    opt = Adam(learning_rate=lr)
    model.compile(optimizer=opt,
                  loss='binary_crossentropy',
                  metrics=['acc'])
    logdir = os.path.join('logs', '[4]2_layer_lr', 'lr_{}'.format(lr))
    tb_callback = TensorBoard(logdir)
```

```
    es_callback = EarlyStopping(monitor='val_acc', patience=20)
    hist = model.fit(X_train, y_train,
                     validation_split=1/9, epochs = 100,
                     verbose=0, callbacks=[tb_callback, es_callback] )
    model.evaluate(X_test, y_test)
```

```
14/14 [==============================] - 0s 614us/step - loss: 0.3471 - acc:
0.8694
14/14 [==============================] - 0s 690us/step - loss: 0.4463 - acc:
0.8646
14/14 [==============================] - 0s 616us/step - loss: 0.3337 - acc:
0.8575
14/14 [==============================] - 0s 537us/step - loss: 0.3281 - acc:
0.8741
```

[14]:
```
%tensorboard --logdir=logs/[4]2_layer_lr
```

```
Reusing TensorBoard on port 6006 (pid 16824), started 0:12:25 ago. (Use '!kill␣
↪16824' to kill it.)
```

```
<IPython.core.display.HTML object>
```

1.8 [5] Effect of learning rate decay

[15]:
```
# w/o learning rate decay
model = create_model(n_layer=2, lambda_=0)
opt = Adam(learning_rate=1e-3)
model.compile(optimizer=opt,
              loss='binary_crossentropy',
              metrics=['acc'])
logdir = os.path.join('logs', '[5]lr_decay', 'none')
tb_callback = TensorBoard(logdir)
es_callback = EarlyStopping(monitor='val_acc', patience=20)
hist = model.fit(X_train, y_train,
                 validation_split=1/9, epochs = 100,
                 verbose=0, callbacks=[tb_callback, es_callback] )
model.evaluate(X_test, y_test)
```

```
14/14 [==============================] - 0s 844us/step - loss: 0.3238 - acc:
0.8717
```

[15]:
```
[0.32383662462234497, 0.8717339634895325]
```

[16]:
```
# w/ learning rate decay
model = create_model(n_layer=2, lambda_=0)
opt = Adam(learning_rate=1e-3)
model.compile(optimizer=opt,
              loss='binary_crossentropy',
```

```
              metrics=['acc'])
logdir = os.path.join('logs','[5]lr_decay','step_decay')
tb_callback = TensorBoard(logdir)
es_callback = EarlyStopping(monitor='val_acc',patience=20)
def scheduler(epoch, lr):
    if epoch in [30, 60, 90]:
        lr = lr*0.1
    return lr
lrs_callback = LearningRateScheduler(scheduler)
hist = model.fit(X_train, y_train,
                 validation_split=1/9, epochs = 100,
                 verbose=0, callbacks=[tb_callback,es_callback,lrs_callback] )
model.evaluate(X_test, y_test)
```

```
14/14 [==============================] - 0s 614us/step - loss: 0.3404 - acc:
0.8646
```

[16]:
```
[0.34042230248451233, 0.8646080493927002]
```

[17]:
```
%tensorboard --logdir=logs/[5]lr_decay
```

```
<IPython.core.display.HTML object>
```