PS5

January 15, 2024

[1]:
```
from shap.datasets import adult

X, y = adult()

print(X)
print(y)
```

```
        Age  Workclass  Education-Num  Marital Status  Occupation  \
0      39.0          7           13.0               4           1
1      50.0          6           13.0               2           4
2      38.0          4            9.0               0           6
3      53.0          4            7.0               2           6
4      28.0          4           13.0               2          10
…       …          …              …               …           …
32556  27.0          4           12.0               2          13
32557  40.0          4            9.0               2           7
32558  58.0          4            9.0               6           1
32559  22.0          4            9.0               4           1
32560  52.0          5            9.0               2           4

       Relationship  Race  Sex  Capital Gain  Capital Loss  Hours per week  \
0                 0     4    1        2174.0           0.0            40.0
1                 4     4    1           0.0           0.0            13.0
2                 0     4    1           0.0           0.0            40.0
3                 4     2    1           0.0           0.0            40.0
4                 5     2    0           0.0           0.0            40.0
…               …     …    …           …             …              …
32556             5     4    0           0.0           0.0            38.0
32557             4     4    1           0.0           0.0            40.0
32558             1     4    0           0.0           0.0            40.0
32559             3     4    1           0.0           0.0            20.0
32560             5     4    0       15024.0           0.0            40.0

       Country
0           39
1           39
2           39
3           39
```

```
4            5
…         …
32556       39
32557       39
32558       39
32559       39
32560       39

[32561 rows x 12 columns]
[False False False … False False  True]
```

[2]:
```
numerical_columns = ['Age','Education-Num','Capital Gain','Capital Loss','Hours␣
↪per week']
categorical_columns = ['Workclass','Marital␣
↪Status','Occupation','Relationship','Race','Sex','Country']
```

0.1 Conversion of categorical data

[3]:
```
import pandas as pd # for one-hot encoding

from sklearn.preprocessing import StandardScaler # for normalization
```

[4]:
```
# Normalization of numerical data
for column in numerical_columns:
    scaler = StandardScaler()
    X[column] = scaler.fit_transform(X[column].values.reshape(-1,1))

print(X)
```

```
            Age  Workclass  Education-Num  Marital Status  Occupation  \
0      0.030671          7       1.134739               4           1
1      0.837109          6       1.134739               2           4
2     -0.042642          4      -0.420060               0           6
3      1.057047          4      -1.197459               2           6
4     -0.775768          4       1.134739               2          10
…           …        …            …           …           …
32556 -0.849080          4       0.746039               2          13
32557  0.103983          4      -0.420060               2           7
32558  1.423610          4      -0.420060               6           1
32559 -1.215643          4      -0.420060               4           1
32560  0.983734          5      -0.420060               2           4

       Relationship  Race  Sex  Capital Gain  Capital Loss  Hours per week  \
0                 0     4    1      0.148453      -0.21666       -0.035429
1                 4     4    1     -0.145920      -0.21666       -2.222153
2                 0     4    1     -0.145920      -0.21666       -0.035429
3                 4     2    1     -0.145920      -0.21666       -0.035429
4                 5     2    0     -0.145920      -0.21666       -0.035429
```

```
…                …   …  …          …          …           …
32556              5     4    0     -0.145920     -0.21666       -0.197409
32557              4     4    1     -0.145920     -0.21666       -0.035429
32558              1     4    0     -0.145920     -0.21666       -0.035429
32559              3     4    1     -0.145920     -0.21666       -1.655225
32560              5     4    0      1.888424     -0.21666       -0.035429

       Country
0           39
1           39
2           39
3           39
4            5
…         …
32556       39
32557       39
32558       39
32559       39
32560       39

[32561 rows x 12 columns]
```

[5]:
```python
# Data type change of categorical data
for column in categorical_columns:
    X[column] = X[column].astype('category')

print(X)
```

```
            Age Workclass  Education-Num Marital Status Occupation  \
0      0.030671         7       1.134739              4          1
1      0.837109         6       1.134739              2          4
2     -0.042642         4      -0.420060              0          6
3      1.057047         4      -1.197459              2          6
4     -0.775768         4       1.134739              2         10
…           …        …              …           …          …
32556 -0.849080         4       0.746039              2         13
32557  0.103983         4      -0.420060              2          7
32558  1.423610         4      -0.420060              6          1
32559 -1.215643         4      -0.420060              4          1
32560  0.983734         5      -0.420060              2          4

      Relationship Race Sex  Capital Gain  Capital Loss  Hours per week  \
0                0    4   1      0.148453      -0.21666       -0.035429
1                4    4   1     -0.145920      -0.21666       -2.222153
2                0    4   1     -0.145920      -0.21666       -0.035429
3                4    2   1     -0.145920      -0.21666       -0.035429
4                5    2   0     -0.145920      -0.21666       -0.035429
…              …   …  ..           …            …              …
```

```
32556         5   4   0   -0.145920   -0.21666   -0.197409
32557         4   4   1   -0.145920   -0.21666   -0.035429
32558         1   4   0   -0.145920   -0.21666   -0.035429
32559         3   4   1   -0.145920   -0.21666   -1.655225
32560         5   4   0    1.888424   -0.21666   -0.035429

       Country
0           39
1           39
2           39
3           39
4            5
…       …
32556       39
32557       39
32558       39
32559       39
32560       39

[32561 rows x 12 columns]
```

[6]:

```
# One-hot encoding of categorical data
X = pd.get_dummies(X)

# Conversion of data frame to numpy
X = X.values

# Converision: {False, True} --> {0., 1.}
y = y.astype(float)
```

[7]:

```
print(X.shape)
print(y.shape)
print(y)
```

```
(32561, 91)
(32561,)
[0. 0. 0. … 0. 0. 1.]
```

0.2 train-val-test split

[8]:

```
from sklearn.model_selection import train_test_split

X_,X_test,y_,y_test = train_test_split(X,y,test_size=1/10,stratify=y)
X_train,X_val,y_train,y_val = train_test_split(X_,y_,test_size=1/9,stratify=y_)

print(X_train.shape)
print(X_val.shape)
print(X_test.shape)
```

```
(26048, 91)
(3256, 91)
(3257, 91)
```

0.3 Logistic regression

[9]:
```python
from sklearn.linear_model import LogisticRegression
```

[10]:
```python
model_LR = LogisticRegression()

# training
model_LR.fit(X_train, y_train)

# evaulation
val_acc = model_LR.score(X_val, y_val)

print(val_acc)
```

```
0.8541154791154791
```

```
lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
```

[11]:
```python
from joblib import dump

dump(model_LR, 'LR_sample.joblib')
```

[11]: ['LR_sample.joblib']

[12]:
```python
from joblib import load

load('LR_sample.joblib')
```

[12]: LogisticRegression()

0.4 A 2-layer DNN

[13]:
```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import Adam

model = Sequential()
model.add(Dense(128, activation='relu'))
```

```python
model.add(Dense(1, activation='sigmoid'))

opt = Adam(learning_rate=0.01,
          beta_1 = 0.9,
          beta_2 = 0.999)


#model.compile(optimizer=opt,
#              loss='binary_crossentropy')


model.compile(optimizer=opt,
             loss='binary_crossentropy',
             metrics=['acc'])

model.fit(X_train,y_train, epochs=10)
```

```
Epoch 1/10
814/814 [==============================] - 3s 2ms/step - loss: 0.3232 - acc:
0.8516
Epoch 2/10
814/814 [==============================] - 2s 2ms/step - loss: 0.3138 - acc:
0.8529
Epoch 3/10
814/814 [==============================] - 2s 2ms/step - loss: 0.3065 - acc:
0.8579
Epoch 4/10
814/814 [==============================] - 2s 3ms/step - loss: 0.3027 - acc:
0.8590
Epoch 5/10
814/814 [==============================] - 2s 3ms/step - loss: 0.2979 - acc:
0.8612
Epoch 6/10
814/814 [==============================] - 2s 3ms/step - loss: 0.2940 - acc:
0.8613
Epoch 7/10
814/814 [==============================] - 2s 3ms/step - loss: 0.2905 - acc:
0.8660
Epoch 8/10
814/814 [==============================] - 2s 2ms/step - loss: 0.2864 - acc:
0.8664
Epoch 9/10
814/814 [==============================] - 2s 3ms/step - loss: 0.2831 - acc:
0.8674
Epoch 10/10
814/814 [==============================] - 2s 3ms/step - loss: 0.2817 - acc:
0.8689
```

[13]: <keras.callbacks.History at 0x1a10329ba30>

[14]:
```
val_hist = model.evaluate(X_val,y_val)

print(val_hist)
```

```
102/102 [==============================] - 1s 4ms/step - loss: 0.3328 - acc:
0.8600
[0.3327931761741638, 0.859950840473175]
```

0.5 Regularization

[15]:
```
from tensorflow.keras.regularizers import l2

model = Sequential()

model.add(Dense(128,
          kernel_regularizer=l2(0.01),
          bias_regularizer=l2(0.01)))
```

0.6 Early stopping

[16]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping

model = Sequential()
model.add(Dense(128,kernel_regularizer=l2(0.01),
          bias_regularizer=l2(0.01),
          activation='relu'))
model.add(Dense(1,kernel_regularizer=l2(0.01),
          bias_regularizer=l2(0.01),
          activation='sigmoid'))
opt = Adam(learning_rate=0.01,beta_1 = 0.9,beta_2 = 0.999)
model.compile(optimizer=opt,
             loss='binary_crossentropy')

#model.compile(optimizer=opt,
#              loss='binary_crossentropy',
#              metrics=['acc'])

#es_callback = EarlyStopping(monitor='val_acc', patience=15)
#es_callback = EarlyStopping(monitor='val_loss', patience=15)
es_callback = EarlyStopping(monitor='val_loss', patience=15)


hist = model.fit(X_train, y_train,
```

```
                 validation_data=(X_val, y_val),
                 epochs=100,callbacks=[es_callback])
```

```
Epoch 1/100
814/814 [==============================] - 4s 4ms/step - loss: 0.4216 -
val_loss: 0.3996
Epoch 2/100
814/814 [==============================] - 3s 3ms/step - loss: 0.4030 -
val_loss: 0.4026
Epoch 3/100
814/814 [==============================] - 2s 2ms/step - loss: 0.4030 -
val_loss: 0.3996
Epoch 4/100
814/814 [==============================] - 2s 2ms/step - loss: 0.3999 -
val_loss: 0.4011
Epoch 5/100
814/814 [==============================] - 2s 2ms/step - loss: 0.4002 -
val_loss: 0.4001
Epoch 6/100
814/814 [==============================] - 2s 3ms/step - loss: 0.3978 -
val_loss: 0.4024
Epoch 7/100
814/814 [==============================] - 2s 2ms/step - loss: 0.3986 -
val_loss: 0.3934
Epoch 8/100
814/814 [==============================] - 2s 3ms/step - loss: 0.3986 -
val_loss: 0.3961
Epoch 9/100
814/814 [==============================] - 2s 2ms/step - loss: 0.3975 -
val_loss: 0.4015
Epoch 10/100
814/814 [==============================] - 2s 3ms/step - loss: 0.3978 -
val_loss: 0.3999
Epoch 11/100
814/814 [==============================] - 2s 3ms/step - loss: 0.3978 -
val_loss: 0.3975
Epoch 12/100
814/814 [==============================] - 2s 3ms/step - loss: 0.3985 -
val_loss: 0.3974
Epoch 13/100
814/814 [==============================] - 2s 3ms/step - loss: 0.3966 -
val_loss: 0.3942
Epoch 14/100
814/814 [==============================] - 2s 2ms/step - loss: 0.3981 -
val_loss: 0.3993
Epoch 15/100
814/814 [==============================] - 2s 2ms/step - loss: 0.3978 -
val_loss: 0.3936
```

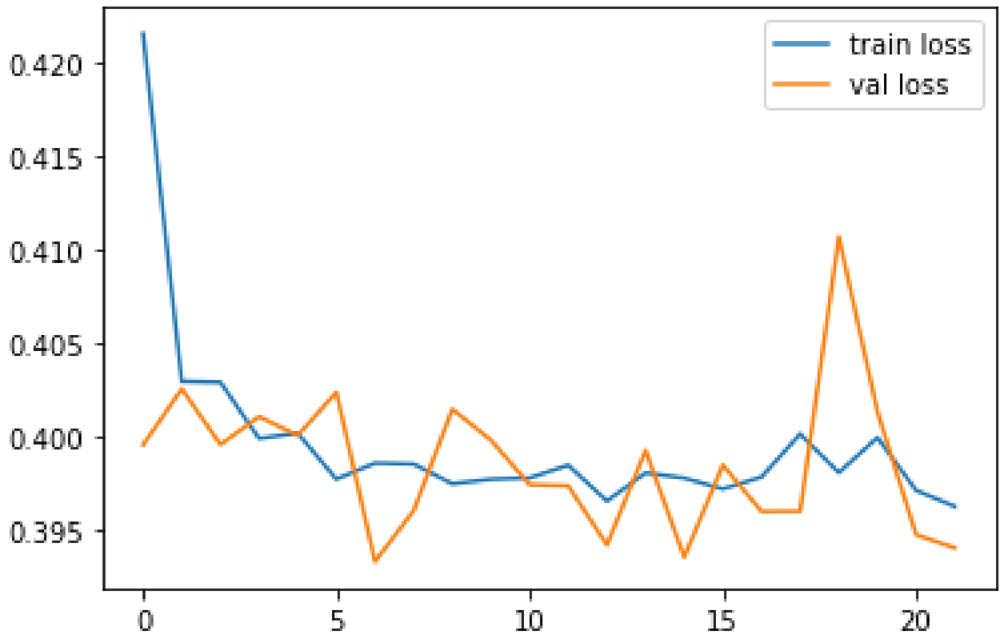
```
Epoch 16/100
814/814 [==============================] - 2s 2ms/step - loss: 0.3972 -
val_loss: 0.3985
Epoch 17/100
814/814 [==============================] - 2s 3ms/step - loss: 0.3979 -
val_loss: 0.3960
Epoch 18/100
814/814 [==============================] - 3s 3ms/step - loss: 0.4002 -
val_loss: 0.3960
Epoch 19/100
814/814 [==============================] - 2s 2ms/step - loss: 0.3981 -
val_loss: 0.4107
Epoch 20/100
814/814 [==============================] - 2s 3ms/step - loss: 0.4000 -
val_loss: 0.4014
Epoch 21/100
814/814 [==============================] - 2s 3ms/step - loss: 0.3972 -
val_loss: 0.3948
Epoch 22/100
814/814 [==============================] - 2s 2ms/step - loss: 0.3963 -
val_loss: 0.3941
```

[17]:

```python
train_loss = hist.history['loss']
val_loss = hist.history['val_loss']
#train_acc = hist.history['acc']
#val_acc = hist.history['val_acc']

import matplotlib.pyplot as plt

plt.plot(train_loss,label='train loss')
plt.plot(val_loss,label='val loss')
plt.legend()
plt.show()
```



0.7 Dropout

[18]:
```
from tensorflow.keras.layers import Dropout

model = Sequential()
model.add(Dense(128,kernel_regularizer=l2(0.01),
          bias_regularizer=l2(0.01),
          activation='relu'))

model.add(Dropout(0.9))
```

0.8 Weight initialization

[19]:
```
from tensorflow.keras.initializers import HeNormal

init = HeNormal()

model.add(Dense(128,kernel_regularizer=l2(0.01),
          bias_regularizer=l2(0.01),
          kernel_initializer=init,
          activation='relu'))
```

0.9 Batch normalization

[20]:
```
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.layers import ReLU

init = HeNormal()

model.add(Dense(128,kernel_regularizer=l2(0.01),
          bias_regularizer=l2(0.01),
          kernel_initializer=init))

model.add(BatchNormalization())

model.add(ReLU())
```

0.10 Learning rate decaying

[21]:
```
from tensorflow.keras.initializers import HeNormal
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.layers import ReLU
from tensorflow.keras.layers import Dropout
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.callbacks import LearningRateScheduler

init = HeNormal()

model = Sequential()
model.add(Dense(128,kernel_regularizer=l2(0.01),
          bias_regularizer=l2(0.01),
          kernel_initializer=init))
model.add(BatchNormalization())
model.add(ReLU())
model.add(Dropout(0.5))
model.add(Dense(1,activation='sigmoid'))

opt = Adam(learning_rate=0.01,beta_1 = 0.9,beta_2 = 0.999)
#model.compile(optimizer=opt,
#              loss='binary_crossentropy',
#              metrics=['acc'])
model.compile(optimizer=opt,
              loss='binary_crossentropy')

es_callback = EarlyStopping(monitor='val_loss', patience=15)
#es_callback = EarlyStopping(monitor='val_acc', patience=15)

def scheduler(epoch, lr):
    if epoch in [20,40,60]:
```

```python
        lr = 0.1*lr
    else:
        lr = lr
    return lr

ls_callback = LearningRateScheduler(scheduler)


hist = model.fit(X_train, y_train,
                 validation_data=(X_val, y_val),
                 epochs=100, callbacks=[es_callback,ls_callback])
```

```
Epoch 1/100
814/814 [==============================] - 3s 3ms/step - loss: 0.4951 -
val_loss: 0.4659 - lr: 0.0100
Epoch 2/100
814/814 [==============================] - 2s 3ms/step - loss: 0.4420 -
val_loss: 0.4038 - lr: 0.0100
Epoch 3/100
814/814 [==============================] - 2s 3ms/step - loss: 0.4307 -
val_loss: 0.4165 - lr: 0.0100
Epoch 4/100
814/814 [==============================] - 3s 3ms/step - loss: 0.4267 -
val_loss: 0.4006 - lr: 0.0100
Epoch 5/100
814/814 [==============================] - 2s 3ms/step - loss: 0.4259 -
val_loss: 0.4203 - lr: 0.0100
Epoch 6/100
814/814 [==============================] - 2s 3ms/step - loss: 0.4253 -
val_loss: 0.4111 - lr: 0.0100
Epoch 7/100
814/814 [==============================] - 2s 3ms/step - loss: 0.4245 -
val_loss: 0.3966 - lr: 0.0100
Epoch 8/100
814/814 [==============================] - 3s 3ms/step - loss: 0.4228 -
val_loss: 0.3943 - lr: 0.0100
Epoch 9/100
814/814 [==============================] - 3s 3ms/step - loss: 0.4227 -
val_loss: 0.3920 - lr: 0.0100
Epoch 10/100
814/814 [==============================] - 3s 3ms/step - loss: 0.4204 -
val_loss: 0.4045 - lr: 0.0100
Epoch 11/100
814/814 [==============================] - 3s 4ms/step - loss: 0.4207 -
val_loss: 0.4148 - lr: 0.0100
Epoch 12/100
814/814 [==============================] - 3s 3ms/step - loss: 0.4207 -
val_loss: 0.3959 - lr: 0.0100
```

```
Epoch 13/100
814/814 [==============================] - 3s 3ms/step - loss: 0.4214 -
val_loss: 0.4031 - lr: 0.0100
Epoch 14/100
814/814 [==============================] - 3s 3ms/step - loss: 0.4186 -
val_loss: 0.4039 - lr: 0.0100
Epoch 15/100
814/814 [==============================] - 2s 3ms/step - loss: 0.4201 -
val_loss: 0.3973 - lr: 0.0100
Epoch 16/100
814/814 [==============================] - 3s 4ms/step - loss: 0.4164 -
val_loss: 0.3890 - lr: 0.0100
Epoch 17/100
814/814 [==============================] - 3s 3ms/step - loss: 0.4185 -
val_loss: 0.4017 - lr: 0.0100
Epoch 18/100
814/814 [==============================] - 2s 3ms/step - loss: 0.4198 -
val_loss: 0.3992 - lr: 0.0100
Epoch 19/100
814/814 [==============================] - 2s 3ms/step - loss: 0.4193 -
val_loss: 0.3931 - lr: 0.0100
Epoch 20/100
814/814 [==============================] - 3s 3ms/step - loss: 0.4203 -
val_loss: 0.3863 - lr: 0.0100
Epoch 21/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3694 -
val_loss: 0.3417 - lr: 1.0000e-03
Epoch 22/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3557 -
val_loss: 0.3389 - lr: 1.0000e-03
Epoch 23/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3487 -
val_loss: 0.3373 - lr: 1.0000e-03
Epoch 24/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3488 -
val_loss: 0.3351 - lr: 1.0000e-03
Epoch 25/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3455 -
val_loss: 0.3318 - lr: 1.0000e-03
Epoch 26/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3476 -
val_loss: 0.3328 - lr: 1.0000e-03
Epoch 27/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3471 -
val_loss: 0.3327 - lr: 1.0000e-03
Epoch 28/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3477 -
val_loss: 0.3393 - lr: 1.0000e-03
```

```
Epoch 29/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3475 -
val_loss: 0.3408 - lr: 1.0000e-03
Epoch 30/100
814/814 [==============================] - 4s 4ms/step - loss: 0.3467 -
val_loss: 0.3379 - lr: 1.0000e-03
Epoch 31/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3486 -
val_loss: 0.3344 - lr: 1.0000e-03
Epoch 32/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3470 -
val_loss: 0.3319 - lr: 1.0000e-03
Epoch 33/100
814/814 [==============================] - 2s 3ms/step - loss: 0.3447 -
val_loss: 0.3334 - lr: 1.0000e-03
Epoch 34/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3474 -
val_loss: 0.3366 - lr: 1.0000e-03
Epoch 35/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3471 -
val_loss: 0.3358 - lr: 1.0000e-03
Epoch 36/100
814/814 [==============================] - 2s 3ms/step - loss: 0.3481 -
val_loss: 0.3347 - lr: 1.0000e-03
Epoch 37/100
814/814 [==============================] - 2s 3ms/step - loss: 0.3476 -
val_loss: 0.3300 - lr: 1.0000e-03
Epoch 38/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3475 -
val_loss: 0.3356 - lr: 1.0000e-03
Epoch 39/100
814/814 [==============================] - 2s 2ms/step - loss: 0.3478 -
val_loss: 0.3344 - lr: 1.0000e-03
Epoch 40/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3463 -
val_loss: 0.3370 - lr: 1.0000e-03
Epoch 41/100
814/814 [==============================] - 2s 3ms/step - loss: 0.3403 -
val_loss: 0.3296 - lr: 1.0000e-04
Epoch 42/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3388 -
val_loss: 0.3290 - lr: 1.0000e-04
Epoch 43/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3372 -
val_loss: 0.3275 - lr: 1.0000e-04
Epoch 44/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3345 -
val_loss: 0.3266 - lr: 1.0000e-04
```

```
Epoch 45/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3292 -
val_loss: 0.3270 - lr: 1.0000e-04
Epoch 46/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3307 -
val_loss: 0.3248 - lr: 1.0000e-04
Epoch 47/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3317 -
val_loss: 0.3250 - lr: 1.0000e-04
Epoch 48/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3315 -
val_loss: 0.3237 - lr: 1.0000e-04
Epoch 49/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3316 -
val_loss: 0.3243 - lr: 1.0000e-04
Epoch 50/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3311 -
val_loss: 0.3255 - lr: 1.0000e-04
Epoch 51/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3298 -
val_loss: 0.3232 - lr: 1.0000e-04
Epoch 52/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3276 -
val_loss: 0.3222 - lr: 1.0000e-04
Epoch 53/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3289 -
val_loss: 0.3236 - lr: 1.0000e-04
Epoch 54/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3294 -
val_loss: 0.3214 - lr: 1.0000e-04
Epoch 55/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3288 -
val_loss: 0.3223 - lr: 1.0000e-04
Epoch 56/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3297 -
val_loss: 0.3215 - lr: 1.0000e-04
Epoch 57/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3294 -
val_loss: 0.3231 - lr: 1.0000e-04
Epoch 58/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3271 -
val_loss: 0.3204 - lr: 1.0000e-04
Epoch 59/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3256 -
val_loss: 0.3211 - lr: 1.0000e-04
Epoch 60/100
814/814 [==============================] - 2s 3ms/step - loss: 0.3267 -
val_loss: 0.3205 - lr: 1.0000e-04
```

```
Epoch 61/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3253 -
val_loss: 0.3200 - lr: 1.0000e-05
Epoch 62/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3269 -
val_loss: 0.3222 - lr: 1.0000e-05
Epoch 63/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3228 -
val_loss: 0.3213 - lr: 1.0000e-05
Epoch 64/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3237 -
val_loss: 0.3213 - lr: 1.0000e-05
Epoch 65/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3260 -
val_loss: 0.3216 - lr: 1.0000e-05
Epoch 66/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3249 -
val_loss: 0.3204 - lr: 1.0000e-05
Epoch 67/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3249 -
val_loss: 0.3209 - lr: 1.0000e-05
Epoch 68/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3240 -
val_loss: 0.3198 - lr: 1.0000e-05
Epoch 69/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3260 -
val_loss: 0.3218 - lr: 1.0000e-05
Epoch 70/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3249 -
val_loss: 0.3210 - lr: 1.0000e-05
Epoch 71/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3267 -
val_loss: 0.3205 - lr: 1.0000e-05
Epoch 72/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3216 -
val_loss: 0.3220 - lr: 1.0000e-05
Epoch 73/100
814/814 [==============================] - 2s 3ms/step - loss: 0.3207 -
val_loss: 0.3200 - lr: 1.0000e-05
Epoch 74/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3244 -
val_loss: 0.3211 - lr: 1.0000e-05
Epoch 75/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3245 -
val_loss: 0.3216 - lr: 1.0000e-05
Epoch 76/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3236 -
val_loss: 0.3193 - lr: 1.0000e-05
```

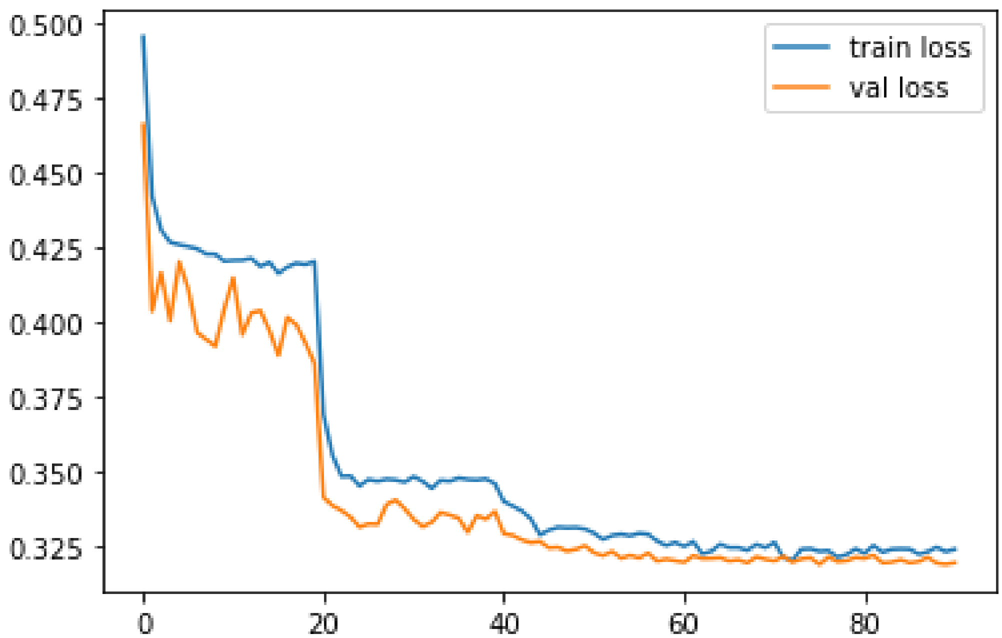
```
Epoch 77/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3239 -
val_loss: 0.3218 - lr: 1.0000e-05
Epoch 78/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3217 -
val_loss: 0.3202 - lr: 1.0000e-05
Epoch 79/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3226 -
val_loss: 0.3205 - lr: 1.0000e-05
Epoch 80/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3245 -
val_loss: 0.3216 - lr: 1.0000e-05
Epoch 81/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3230 -
val_loss: 0.3213 - lr: 1.0000e-05
Epoch 82/100
814/814 [==============================] - 2s 3ms/step - loss: 0.3256 -
val_loss: 0.3223 - lr: 1.0000e-05
Epoch 83/100
814/814 [==============================] - 2s 3ms/step - loss: 0.3233 -
val_loss: 0.3198 - lr: 1.0000e-05
Epoch 84/100
814/814 [==============================] - 2s 3ms/step - loss: 0.3244 -
val_loss: 0.3201 - lr: 1.0000e-05
Epoch 85/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3245 -
val_loss: 0.3209 - lr: 1.0000e-05
Epoch 86/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3244 -
val_loss: 0.3198 - lr: 1.0000e-05
Epoch 87/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3227 -
val_loss: 0.3203 - lr: 1.0000e-05
Epoch 88/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3235 -
val_loss: 0.3217 - lr: 1.0000e-05
Epoch 89/100
814/814 [==============================] - 3s 4ms/step - loss: 0.3250 -
val_loss: 0.3197 - lr: 1.0000e-05
Epoch 90/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3236 -
val_loss: 0.3193 - lr: 1.0000e-05
Epoch 91/100
814/814 [==============================] - 3s 3ms/step - loss: 0.3244 -
val_loss: 0.3199 - lr: 1.0000e-05
```

[22]:

```
train_loss = hist.history['loss']
val_loss = hist.history['val_loss']
#train_acc = hist.history['acc']
#val_acc = hist.history['val_acc']

import matplotlib.pyplot as plt

plt.plot(train_loss,label='train loss')
plt.plot(val_loss,label='val loss')
plt.legend()
plt.show()
```



[ ]:

```
```