ann1-3

October 14, 2023

[155]:
```
import numpy as np
 import pandas as pd
 import matplotlib.pyplot as plt
 from sklearn.model_selection import train_test_split
 from sklearn.preprocessing import MinMaxScaler
 #from sklearn.preprocessing import StandardScaler
 from tensorflow.keras.models import Sequential
 from tensorflow.keras.layers import Dense
 from sklearn.linear_model import LinearRegression
 from sklearn.metrics import r2_score
```

[157]:
```
df = pd.read_csv('/content/car_purchasing.csv', encoding='ISO-8859-1')
df.head()
```

[157]:

	customer name	customer e-mail	country	gender	age	annual Salary	credit card debt	net worth	car purchase amount
0	Martina Avila	cubilia.Curae.Phasellus@quisaccumsanconvallis.edu	Bulgaria	0	41.851720	62812.09301	11609.380910	238961.2505	35321.45877
1	Harlan Barnes	eu.dolor@diam.co.uk	Belize	0	40.870623	66646.89292	9572.957136	530973.9078	45115.52566
2	Naomi Rodriquez	vulputate.mauris.sagittis@ametconsectetueradip…	Algeria	1	43.152897	53798.55112	11160.355060	638467.1773	42925.70921
3	Jade Cunningham	malesuada@dignissim.com	Cook Islands	1	58.271369	79370.03798	14426.164850	548599.0524	67422.36313
4	Cedric Leach	felis.ullamcorper.viverra@egetmollislectus.net	Brazil	1	57.313749	59729.15130	5358.712177	560304.0671	55915.46248

[158]:
```
# number of rows and columns
df.shape
```

[158]:
```
(500, 9)
```

[159]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 9 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   customer name        500 non-null    object
 1   customer e-mail      500 non-null    object
 2   country              500 non-null    object
 3   gender               500 non-null    int64
 4   age                  500 non-null    float64
 5   annual Salary        500 non-null    float64
 6   credit card debt     500 non-null    float64
 7   net worth            500 non-null    float64
 8   car purchase amount  500 non-null    float64
dtypes: float64(5), int64(1), object(3)
memory usage: 35.3+ KB
```

[160]:
```
df.isnull().sum()
```

[160]:
```
customer name          0
customer e-mail        0
country                0
gender                 0
age                    0
annual Salary          0
credit card debt       0
net worth              0
car purchase amount    0
dtype: int64
```

[161]:
```
df.isnull().sum().sum()
```

[161]:
```
0
```

[162]:
```
df.head()
```

[162]:

	customer name	customer e-mail
0	Martina Avila	cubilia.Curae.Phasellus@quisaccumsanconvallis.edu
1	Harlan Barnes	eu.dolor@diam.co.uk
2	Naomi Rodriquez	vulputate.mauris.sagittis@ametconsectetueradip…
3	Jade Cunningham	malesuada@dignissim.com
4	Cedric Leach	felis.ullamcorper.viverra@egetmollislectus.net

```
        country  gender        age  annual Salary  credit card debt  \
0      Bulgaria       0  41.851720    62812.09301      11609.380910
1        Belize       0  40.870623    66646.89292       9572.957136
2       Algeria       1  43.152897    53798.55112      11160.355060
3  Cook Islands       1  58.271369    79370.03798      14426.164850
4        Brazil       1  57.313749    59729.15130       5358.712177

     net worth  car purchase amount
0  238961.2505          35321.45877
1  530973.9078          45115.52566
2  638467.1773          42925.70921
3  548599.0524          67422.36313
4  560304.0671          55915.46248
```

[205]:
```
df.drop(columns= ['customer name' , 'customer e-mail' ,'country' , 'gender' ] ,axis = 1 , inplace = True)
```

[206]:
```
target = 'car purchase amount'
X = df.drop(target , axis= 1)
y= df[target]
```

[208]:
```
X
```

[208]:
```
            age  annual Salary  credit card debt    net worth
0     41.851720    62812.09301      11609.380910  238961.2505  0.0
1     40.870623    66646.89292       9572.957136  530973.9078  0.0
2     43.152897    53798.55112      11160.355060  638467.1773  1.0
3     58.271369    79370.03798      14426.164850  548599.0524  1.0
4     57.313749    59729.15130       5358.712177  560304.0671  1.0
..          …            …                …           …   …
495   41.462515    71942.40291       6995.902524  541670.1016  0.0
496   37.642000    56039.49793      12301.456790  360419.0988  1.0
497   53.943497    68888.77805      10611.606860  764531.3203  1.0
498   59.160509    49811.99062      14013.034510  337826.6382  1.0
499   46.731152    61370.67766       9391.341628  462946.4924  1.0

[500 rows x 5 columns]
```

[209]:
```
scaler = MinMaxScaler()

X = scaler.fit_transform(X)
y = scaler.fit_transform(y.values.reshape(-1 , 1))
```

[210]:
```
print(f'The shape of X is {X.shape}')
print(f'The shape of y is {y.shape}')
```

```
The shape of X is (500, 5)
```

```
The shape of y is (500, 1)
```

[211]:
```
X_train , X_test , y_train , y_test = train_test_split(X, y , test_size = 0.2 ,␣
 ↪random_state = 42)

print(f'the shape of X train is {X_train.shape}')
print(f'the shape of y train is {y_train.shape}')
print(f'the shape of X test is {X_test.shape}')
print(f'the shape of y test is {y_test.shape}')
```

```
the shape of X train is (400, 5)
the shape of y train is (400, 1)
the shape of X test is (100, 5)
the shape of y test is (100, 1)
```

[212]:
```
y_mean = y_train.mean()
y_pred_baseline= [y_mean] * len(y_train)
print("Mean apt price:", y_mean)

print("Baseline MAE:", mean_absolute_error(y_train,y_pred_baseline))
```

```
Mean apt price: 0.49219817948908456
Baseline MAE: 0.11921386215413556
```

[215]:
```
model = Sequential()
model.add(Dense(10, activation='relu', input_dim=5))
model.add(Dense(10, activation='relu'))
model.add(Dense(1, activation='linear'))

model.compile(optimizer='adam', loss='mean_squared_error',␣
 ↪metrics=['mean_absolute_error'])
model.summary()
```

```
Model: "sequential_26"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_79 (Dense)            (None, 10)                60

 dense_80 (Dense)            (None, 10)                110

 dense_81 (Dense)            (None, 1)                 11

=================================================================
Total params: 181 (724.00 Byte)
Trainable params: 181 (724.00 Byte)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

[216]:
```
history = model.fit(X_train , y_train , epochs = 50 , validation_split= 0.2)
```

```
Epoch 1/50
10/10 [==============================] - 1s 21ms/step - loss: 0.0926 -
mean_absolute_error: 0.2678 - val_loss: 0.0411 - val_mean_absolute_error: 0.1641
Epoch 2/50
10/10 [==============================] - 0s 7ms/step - loss: 0.0412 -
mean_absolute_error: 0.1697 - val_loss: 0.0238 - val_mean_absolute_error: 0.1265
Epoch 3/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0268 -
mean_absolute_error: 0.1352 - val_loss: 0.0274 - val_mean_absolute_error: 0.1409
Epoch 4/50
10/10 [==============================] - 0s 5ms/step - loss: 0.0245 -
mean_absolute_error: 0.1256 - val_loss: 0.0247 - val_mean_absolute_error: 0.1322
Epoch 5/50
10/10 [==============================] - 0s 7ms/step - loss: 0.0207 -
mean_absolute_error: 0.1155 - val_loss: 0.0185 - val_mean_absolute_error: 0.1119
Epoch 6/50
10/10 [==============================] - 0s 7ms/step - loss: 0.0175 -
mean_absolute_error: 0.1055 - val_loss: 0.0155 - val_mean_absolute_error: 0.1000
Epoch 7/50
10/10 [==============================] - 0s 7ms/step - loss: 0.0155 -
mean_absolute_error: 0.0983 - val_loss: 0.0139 - val_mean_absolute_error: 0.0936
Epoch 8/50
10/10 [==============================] - 0s 6ms/step - loss: 0.0138 -
mean_absolute_error: 0.0923 - val_loss: 0.0130 - val_mean_absolute_error: 0.0902
Epoch 9/50
10/10 [==============================] - 0s 6ms/step - loss: 0.0125 -
mean_absolute_error: 0.0873 - val_loss: 0.0119 - val_mean_absolute_error: 0.0852
Epoch 10/50
10/10 [==============================] - 0s 5ms/step - loss: 0.0116 -
mean_absolute_error: 0.0838 - val_loss: 0.0110 - val_mean_absolute_error: 0.0813
Epoch 11/50
10/10 [==============================] - 0s 5ms/step - loss: 0.0108 -
mean_absolute_error: 0.0809 - val_loss: 0.0101 - val_mean_absolute_error: 0.0770
Epoch 12/50
10/10 [==============================] - 0s 7ms/step - loss: 0.0101 -
mean_absolute_error: 0.0785 - val_loss: 0.0098 - val_mean_absolute_error: 0.0753
Epoch 13/50
10/10 [==============================] - 0s 5ms/step - loss: 0.0096 -
mean_absolute_error: 0.0760 - val_loss: 0.0089 - val_mean_absolute_error: 0.0709
Epoch 14/50
10/10 [==============================] - 0s 7ms/step - loss: 0.0090 -
mean_absolute_error: 0.0739 - val_loss: 0.0083 - val_mean_absolute_error: 0.0676
Epoch 15/50
10/10 [==============================] - 0s 5ms/step - loss: 0.0086 -
mean_absolute_error: 0.0719 - val_loss: 0.0078 - val_mean_absolute_error: 0.0654
```

```
Epoch 16/50
10/10 [==============================] - 0s 7ms/step - loss: 0.0081 -
mean_absolute_error: 0.0699 - val_loss: 0.0074 - val_mean_absolute_error: 0.0631
Epoch 17/50
10/10 [==============================] - 0s 5ms/step - loss: 0.0077 -
mean_absolute_error: 0.0680 - val_loss: 0.0069 - val_mean_absolute_error: 0.0609
Epoch 18/50
10/10 [==============================] - 0s 9ms/step - loss: 0.0073 -
mean_absolute_error: 0.0662 - val_loss: 0.0065 - val_mean_absolute_error: 0.0585
Epoch 19/50
10/10 [==============================] - 0s 5ms/step - loss: 0.0069 -
mean_absolute_error: 0.0645 - val_loss: 0.0060 - val_mean_absolute_error: 0.0560
Epoch 20/50
10/10 [==============================] - 0s 5ms/step - loss: 0.0065 -
mean_absolute_error: 0.0629 - val_loss: 0.0057 - val_mean_absolute_error: 0.0541
Epoch 21/50
10/10 [==============================] - 0s 5ms/step - loss: 0.0061 -
mean_absolute_error: 0.0608 - val_loss: 0.0053 - val_mean_absolute_error: 0.0521
Epoch 22/50
10/10 [==============================] - 0s 9ms/step - loss: 0.0057 -
mean_absolute_error: 0.0586 - val_loss: 0.0049 - val_mean_absolute_error: 0.0499
Epoch 23/50
10/10 [==============================] - 0s 9ms/step - loss: 0.0053 -
mean_absolute_error: 0.0564 - val_loss: 0.0044 - val_mean_absolute_error: 0.0473
Epoch 24/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0049 -
mean_absolute_error: 0.0545 - val_loss: 0.0041 - val_mean_absolute_error: 0.0455
Epoch 25/50
10/10 [==============================] - 0s 9ms/step - loss: 0.0046 -
mean_absolute_error: 0.0530 - val_loss: 0.0037 - val_mean_absolute_error: 0.0432
Epoch 26/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0043 -
mean_absolute_error: 0.0513 - val_loss: 0.0035 - val_mean_absolute_error: 0.0414
Epoch 27/50
10/10 [==============================] - 0s 10ms/step - loss: 0.0041 -
mean_absolute_error: 0.0500 - val_loss: 0.0032 - val_mean_absolute_error: 0.0403
Epoch 28/50
10/10 [==============================] - 0s 9ms/step - loss: 0.0038 -
mean_absolute_error: 0.0486 - val_loss: 0.0030 - val_mean_absolute_error: 0.0384
Epoch 29/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0036 -
mean_absolute_error: 0.0473 - val_loss: 0.0028 - val_mean_absolute_error: 0.0371
Epoch 30/50
10/10 [==============================] - 0s 7ms/step - loss: 0.0034 -
mean_absolute_error: 0.0462 - val_loss: 0.0026 - val_mean_absolute_error: 0.0361
Epoch 31/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0033 -
mean_absolute_error: 0.0450 - val_loss: 0.0024 - val_mean_absolute_error: 0.0347
```

```
Epoch 32/50
10/10 [==============================] - 0s 7ms/step - loss: 0.0031 -
mean_absolute_error: 0.0439 - val_loss: 0.0023 - val_mean_absolute_error: 0.0336
Epoch 33/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0029 -
mean_absolute_error: 0.0428 - val_loss: 0.0022 - val_mean_absolute_error: 0.0331
Epoch 34/50
10/10 [==============================] - 0s 9ms/step - loss: 0.0028 -
mean_absolute_error: 0.0417 - val_loss: 0.0020 - val_mean_absolute_error: 0.0317
Epoch 35/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0027 -
mean_absolute_error: 0.0410 - val_loss: 0.0019 - val_mean_absolute_error: 0.0306
Epoch 36/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0025 -
mean_absolute_error: 0.0395 - val_loss: 0.0018 - val_mean_absolute_error: 0.0306
Epoch 37/50
10/10 [==============================] - 0s 9ms/step - loss: 0.0024 -
mean_absolute_error: 0.0387 - val_loss: 0.0017 - val_mean_absolute_error: 0.0290
Epoch 38/50
10/10 [==============================] - 0s 9ms/step - loss: 0.0022 -
mean_absolute_error: 0.0374 - val_loss: 0.0016 - val_mean_absolute_error: 0.0284
Epoch 39/50
10/10 [==============================] - 0s 7ms/step - loss: 0.0021 -
mean_absolute_error: 0.0365 - val_loss: 0.0015 - val_mean_absolute_error: 0.0276
Epoch 40/50
10/10 [==============================] - 0s 9ms/step - loss: 0.0020 -
mean_absolute_error: 0.0359 - val_loss: 0.0014 - val_mean_absolute_error: 0.0267
Epoch 41/50
10/10 [==============================] - 0s 7ms/step - loss: 0.0020 -
mean_absolute_error: 0.0351 - val_loss: 0.0013 - val_mean_absolute_error: 0.0267
Epoch 42/50
10/10 [==============================] - 0s 7ms/step - loss: 0.0018 -
mean_absolute_error: 0.0333 - val_loss: 0.0012 - val_mean_absolute_error: 0.0253
Epoch 43/50
10/10 [==============================] - 0s 9ms/step - loss: 0.0017 -
mean_absolute_error: 0.0327 - val_loss: 0.0012 - val_mean_absolute_error: 0.0249
Epoch 44/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0016 -
mean_absolute_error: 0.0318 - val_loss: 0.0011 - val_mean_absolute_error: 0.0243
Epoch 45/50
10/10 [==============================] - 0s 7ms/step - loss: 0.0015 -
mean_absolute_error: 0.0310 - val_loss: 0.0010 - val_mean_absolute_error: 0.0234
Epoch 46/50
10/10 [==============================] - 0s 10ms/step - loss: 0.0014 -
mean_absolute_error: 0.0300 - val_loss: 0.0010 - val_mean_absolute_error: 0.0235
Epoch 47/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0013 -
mean_absolute_error: 0.0287 - val_loss: 9.1863e-04 - val_mean_absolute_error:
```

```
0.0224
Epoch 48/50
10/10 [==============================] - 0s 9ms/step - loss: 0.0013 -
mean_absolute_error: 0.0281 - val_loss: 8.8108e-04 - val_mean_absolute_error:
0.0220
Epoch 49/50
10/10 [==============================] - 0s 9ms/step - loss: 0.0012 -
mean_absolute_error: 0.0272 - val_loss: 8.2292e-04 - val_mean_absolute_error:
0.0212
Epoch 50/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0011 -
mean_absolute_error: 0.0263 - val_loss: 7.7785e-04 - val_mean_absolute_error:
0.0207
```

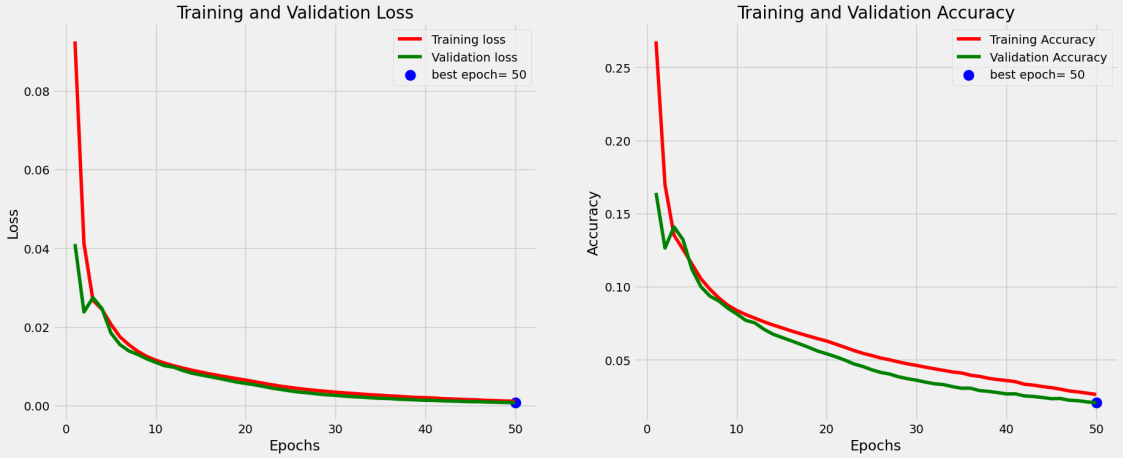
[217]:
```
tr_acc = history.history['mean_absolute_error']
tr_loss = history.history['loss']
val_acc = history.history['val_mean_absolute_error']
val_loss = history.history['val_loss']
index_loss = np.argmin(val_loss)
val_lowest = val_loss[index_loss]
index_acc = np.argmin(val_acc)
acc_highest = val_acc[index_acc]
Epochs = [i+1 for i in range(len(tr_acc))]
loss_label = f'best epoch= {str(index_loss + 1)}'
acc_label = f'best epoch= {str(index_acc + 1)}'

# Plot training history
plt.figure(figsize= (20, 8))
plt.style.use('fivethirtyeight')

plt.subplot(1, 2, 1)
plt.plot(Epochs, tr_loss, 'r', label= 'Training loss')
plt.plot(Epochs, val_loss, 'g', label= 'Validation loss')
plt.scatter(index_loss + 1, val_lowest, s= 150, c= 'blue', label= loss_label)
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(Epochs, tr_acc, 'r', label= 'Training Accuracy')
plt.plot(Epochs, val_acc, 'g', label= 'Validation Accuracy')
plt.scatter(index_acc + 1 , acc_highest, s= 150, c= 'blue', label= acc_label)
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
```

```
plt.tight_layout
plt.show()
```



[218]:
```
y_pred = model.predict(X_test)
```

```
4/4 [==============================] - 0s 2ms/step
```

[219]:
```
from sklearn.metrics import r2_score
```

[220]:
```
test_acc = r2_score(y_test , y_pred)
print(f'R2 Score = {test_acc}')
```

```
R2 Score = 0.962160778590368
```

[293]:
```
import keras
from keras.models import Sequential
from keras.layers import Dense #sparse or dense
from keras.layers import LeakyReLU,PReLU,ELU # activation functions


# intialize the empty artifical neural network without inputs and outputs
classifier=Sequential()


# adding the input layer and the first hidden layer
classifier.add(Dense(units =4, kernel_initializer =␣
 ↪'he_uniform',activation='relu',input_dim = 5)) #units are output␣
 ↪dimension(neurons),keral_intializers are weight intilalization Technqiues
```

```
# adding the second input layer and the first hidden layer
classifier.add(Dense(units = 4, kernel_initializer =␣
 ↪'he_uniform',activation='relu')) # for relu u can use he_uniform, he_normal


# adding the output layer
classifier.
 ↪add(Dense(units=1,kernel_initializer='glorot_uniform',activation="relu")) #␣
 ↪for sigmoid we have to use weight intialzers as a glorot



# sigmoid actiavtion function will be used in output layers if your ouptut is␣
 ↪binary classification


#compling the ANN
classifier.
 ↪compile(optimizer='Adamax',loss="binary_crossentropy",metrics=["mae"])
```

[294]:
```
annhistory=classifier.fit(X_train,y_train,validation_split=.
 ↪33,batch_size=10,epochs=100)
```

```
Epoch 1/100
27/27 [==============================] - 1s 9ms/step - loss: 7.1948 - mae:
0.4916 - val_loss: 6.7039 - val_mae: 0.4755
Epoch 2/100
27/27 [==============================] - 0s 4ms/step - loss: 6.9027 - mae:
0.4881 - val_loss: 6.3652 - val_mae: 0.4705
Epoch 3/100
27/27 [==============================] - 0s 3ms/step - loss: 6.5648 - mae:
0.4836 - val_loss: 5.7855 - val_mae: 0.4638
Epoch 4/100
27/27 [==============================] - 0s 3ms/step - loss: 6.2303 - mae:
0.4793 - val_loss: 5.4448 - val_mae: 0.4588
Epoch 5/100
27/27 [==============================] - 0s 4ms/step - loss: 6.0295 - mae:
0.4753 - val_loss: 5.2280 - val_mae: 0.4529
Epoch 6/100
27/27 [==============================] - 0s 3ms/step - loss: 5.7904 - mae:
0.4712 - val_loss: 5.1482 - val_mae: 0.4493
Epoch 7/100
27/27 [==============================] - 0s 5ms/step - loss: 5.6736 - mae:
0.4695 - val_loss: 5.1359 - val_mae: 0.4481
Epoch 8/100
27/27 [==============================] - 0s 3ms/step - loss: 5.6329 - mae:
```

```
0.4686 - val_loss: 5.0945 - val_mae: 0.4471
Epoch 9/100
27/27 [==============================] - 0s 3ms/step - loss: 5.5689 - mae:
0.4678 - val_loss: 5.0335 - val_mae: 0.4458
Epoch 10/100
27/27 [==============================] - 0s 3ms/step - loss: 5.5400 - mae:
0.4663 - val_loss: 4.9752 - val_mae: 0.4441
Epoch 11/100
27/27 [==============================] - 0s 3ms/step - loss: 5.5277 - mae:
0.4654 - val_loss: 4.9676 - val_mae: 0.4434
Epoch 12/100
27/27 [==============================] - 0s 3ms/step - loss: 5.5081 - mae:
0.4648 - val_loss: 4.9081 - val_mae: 0.4424
Epoch 13/100
27/27 [==============================] - 0s 3ms/step - loss: 5.4982 - mae:
0.4641 - val_loss: 4.8949 - val_mae: 0.4418
Epoch 14/100
27/27 [==============================] - 0s 4ms/step - loss: 5.4590 - mae:
0.4634 - val_loss: 4.8018 - val_mae: 0.4403
Epoch 15/100
27/27 [==============================] - 0s 4ms/step - loss: 5.3704 - mae:
0.4617 - val_loss: 4.7498 - val_mae: 0.4384
Epoch 16/100
27/27 [==============================] - 0s 3ms/step - loss: 5.3252 - mae:
0.4607 - val_loss: 4.7389 - val_mae: 0.4374
Epoch 17/100
27/27 [==============================] - 0s 3ms/step - loss: 5.3061 - mae:
0.4599 - val_loss: 4.7028 - val_mae: 0.4362
Epoch 18/100
27/27 [==============================] - 0s 3ms/step - loss: 5.1992 - mae:
0.4582 - val_loss: 4.6520 - val_mae: 0.4340
Epoch 19/100
27/27 [==============================] - 0s 5ms/step - loss: 5.1379 - mae:
0.4568 - val_loss: 4.6396 - val_mae: 0.4326
Epoch 20/100
27/27 [==============================] - 0s 4ms/step - loss: 5.1056 - mae:
0.4558 - val_loss: 4.5959 - val_mae: 0.4315
Epoch 21/100
27/27 [==============================] - 0s 3ms/step - loss: 5.0653 - mae:
0.4549 - val_loss: 4.5843 - val_mae: 0.4305
Epoch 22/100
27/27 [==============================] - 0s 3ms/step - loss: 5.0376 - mae:
0.4540 - val_loss: 4.5509 - val_mae: 0.4296
Epoch 23/100
27/27 [==============================] - 0s 3ms/step - loss: 4.9809 - mae:
0.4531 - val_loss: 4.4504 - val_mae: 0.4280
Epoch 24/100
27/27 [==============================] - 0s 4ms/step - loss: 4.9473 - mae:
```

```
0.4518 - val_loss: 4.4299 - val_mae: 0.4268
Epoch 25/100
27/27 [==============================] - 0s 3ms/step - loss: 4.9379 - mae:
0.4509 - val_loss: 4.4021 - val_mae: 0.4259
Epoch 26/100
27/27 [==============================] - 0s 3ms/step - loss: 4.9307 - mae:
0.4503 - val_loss: 4.3927 - val_mae: 0.4252
Epoch 27/100
27/27 [==============================] - 0s 3ms/step - loss: 4.9251 - mae:
0.4496 - val_loss: 4.3850 - val_mae: 0.4244
Epoch 28/100
27/27 [==============================] - 0s 3ms/step - loss: 4.8791 - mae:
0.4485 - val_loss: 4.3688 - val_mae: 0.4225
Epoch 29/100
27/27 [==============================] - 0s 4ms/step - loss: 4.8325 - mae:
0.4473 - val_loss: 4.3418 - val_mae: 0.4213
Epoch 30/100
27/27 [==============================] - 0s 4ms/step - loss: 4.7348 - mae:
0.4457 - val_loss: 4.1505 - val_mae: 0.4180
Epoch 31/100
27/27 [==============================] - 0s 3ms/step - loss: 4.5755 - mae:
0.4427 - val_loss: 4.0677 - val_mae: 0.4156
Epoch 32/100
27/27 [==============================] - 0s 3ms/step - loss: 4.4384 - mae:
0.4407 - val_loss: 3.8057 - val_mae: 0.4117
Epoch 33/100
27/27 [==============================] - 0s 3ms/step - loss: 4.2707 - mae:
0.4366 - val_loss: 3.4825 - val_mae: 0.4079
Epoch 34/100
27/27 [==============================] - 0s 4ms/step - loss: 4.0954 - mae:
0.4344 - val_loss: 3.4025 - val_mae: 0.4064
Epoch 35/100
27/27 [==============================] - 0s 3ms/step - loss: 4.0609 - mae:
0.4334 - val_loss: 3.3834 - val_mae: 0.4054
Epoch 36/100
27/27 [==============================] - 0s 3ms/step - loss: 3.9080 - mae:
0.4322 - val_loss: 3.2291 - val_mae: 0.4019
Epoch 37/100
27/27 [==============================] - 0s 3ms/step - loss: 3.6413 - mae:
0.4282 - val_loss: 3.1520 - val_mae: 0.3987
Epoch 38/100
27/27 [==============================] - 0s 4ms/step - loss: 3.5529 - mae:
0.4263 - val_loss: 3.0705 - val_mae: 0.3972
Epoch 39/100
27/27 [==============================] - 0s 4ms/step - loss: 3.4773 - mae:
0.4248 - val_loss: 2.9852 - val_mae: 0.3954
Epoch 40/100
27/27 [==============================] - 0s 4ms/step - loss: 3.3641 - mae:
```

```
0.4231 - val_loss: 2.7962 - val_mae: 0.3935
Epoch 41/100
27/27 [==============================] - 0s 3ms/step - loss: 3.2708 - mae:
0.4210 - val_loss: 2.6138 - val_mae: 0.3895
Epoch 42/100
27/27 [==============================] - 0s 4ms/step - loss: 3.0481 - mae:
0.4174 - val_loss: 2.5298 - val_mae: 0.3871
Epoch 43/100
27/27 [==============================] - 0s 5ms/step - loss: 2.9372 - mae:
0.4155 - val_loss: 2.4040 - val_mae: 0.3849
Epoch 44/100
27/27 [==============================] - 0s 4ms/step - loss: 2.8887 - mae:
0.4136 - val_loss: 2.3761 - val_mae: 0.3832
Epoch 45/100
27/27 [==============================] - 0s 3ms/step - loss: 2.7945 - mae:
0.4120 - val_loss: 2.1862 - val_mae: 0.3809
Epoch 46/100
27/27 [==============================] - 0s 3ms/step - loss: 2.7004 - mae:
0.4093 - val_loss: 2.0507 - val_mae: 0.3779
Epoch 47/100
27/27 [==============================] - 0s 3ms/step - loss: 2.4393 - mae:
0.4058 - val_loss: 1.8047 - val_mae: 0.3719
Epoch 48/100
27/27 [==============================] - 0s 3ms/step - loss: 2.1042 - mae:
0.4006 - val_loss: 1.7538 - val_mae: 0.3684
Epoch 49/100
27/27 [==============================] - 0s 4ms/step - loss: 1.9854 - mae:
0.3978 - val_loss: 1.6881 - val_mae: 0.3658
Epoch 50/100
27/27 [==============================] - 0s 3ms/step - loss: 1.8702 - mae:
0.3956 - val_loss: 1.6084 - val_mae: 0.3629
Epoch 51/100
27/27 [==============================] - 0s 3ms/step - loss: 1.7687 - mae:
0.3923 - val_loss: 1.5394 - val_mae: 0.3600
Epoch 52/100
27/27 [==============================] - 0s 3ms/step - loss: 1.7106 - mae:
0.3897 - val_loss: 1.5157 - val_mae: 0.3576
Epoch 53/100
27/27 [==============================] - 0s 3ms/step - loss: 1.6604 - mae:
0.3878 - val_loss: 1.4995 - val_mae: 0.3557
Epoch 54/100
27/27 [==============================] - 0s 3ms/step - loss: 1.5680 - mae:
0.3855 - val_loss: 1.3635 - val_mae: 0.3530
Epoch 55/100
27/27 [==============================] - 0s 3ms/step - loss: 1.5244 - mae:
0.3822 - val_loss: 1.3158 - val_mae: 0.3497
Epoch 56/100
27/27 [==============================] - 0s 3ms/step - loss: 1.5027 - mae:
```

```
0.3801 - val_loss: 1.2650 - val_mae: 0.3481
Epoch 57/100
27/27 [==============================] - 0s 4ms/step - loss: 1.4636 - mae:
0.3785 - val_loss: 1.2418 - val_mae: 0.3461
Epoch 58/100
27/27 [==============================] - 0s 3ms/step - loss: 1.3730 - mae:
0.3759 - val_loss: 1.2164 - val_mae: 0.3431
Epoch 59/100
27/27 [==============================] - 0s 3ms/step - loss: 1.3128 - mae:
0.3734 - val_loss: 1.2006 - val_mae: 0.3409
Epoch 60/100
27/27 [==============================] - 0s 3ms/step - loss: 1.2927 - mae:
0.3715 - val_loss: 1.1895 - val_mae: 0.3392
Epoch 61/100
27/27 [==============================] - 0s 4ms/step - loss: 1.2782 - mae:
0.3699 - val_loss: 1.1801 - val_mae: 0.3376
Epoch 62/100
27/27 [==============================] - 0s 3ms/step - loss: 1.2661 - mae:
0.3683 - val_loss: 1.1706 - val_mae: 0.3359
Epoch 63/100
27/27 [==============================] - 0s 3ms/step - loss: 1.2550 - mae:
0.3667 - val_loss: 1.1621 - val_mae: 0.3344
Epoch 64/100
27/27 [==============================] - 0s 4ms/step - loss: 1.2443 - mae:
0.3652 - val_loss: 1.1546 - val_mae: 0.3329
Epoch 65/100
27/27 [==============================] - 0s 3ms/step - loss: 1.2351 - mae:
0.3637 - val_loss: 1.1465 - val_mae: 0.3313
Epoch 66/100
27/27 [==============================] - 0s 3ms/step - loss: 1.2256 - mae:
0.3623 - val_loss: 1.1394 - val_mae: 0.3298
Epoch 67/100
27/27 [==============================] - 0s 4ms/step - loss: 1.2168 - mae:
0.3608 - val_loss: 1.1323 - val_mae: 0.3283
Epoch 68/100
27/27 [==============================] - 0s 3ms/step - loss: 1.2084 - mae:
0.3593 - val_loss: 1.1253 - val_mae: 0.3267
Epoch 69/100
27/27 [==============================] - 0s 4ms/step - loss: 1.2000 - mae:
0.3578 - val_loss: 1.1188 - val_mae: 0.3253
Epoch 70/100
27/27 [==============================] - 0s 3ms/step - loss: 1.1922 - mae:
0.3563 - val_loss: 1.1123 - val_mae: 0.3238
Epoch 71/100
27/27 [==============================] - 0s 3ms/step - loss: 1.1844 - mae:
0.3549 - val_loss: 1.1061 - val_mae: 0.3223
Epoch 72/100
27/27 [==============================] - 0s 3ms/step - loss: 1.1769 - mae:
```

```
0.3534 - val_loss: 1.0999 - val_mae: 0.3207
Epoch 73/100
27/27 [==============================] - 0s 5ms/step - loss: 1.1697 - mae:
0.3519 - val_loss: 1.0935 - val_mae: 0.3192
Epoch 74/100
27/27 [==============================] - 0s 5ms/step - loss: 1.1624 - mae:
0.3504 - val_loss: 1.0876 - val_mae: 0.3176
Epoch 75/100
27/27 [==============================] - 0s 6ms/step - loss: 1.1552 - mae:
0.3490 - val_loss: 1.0820 - val_mae: 0.3161
Epoch 76/100
27/27 [==============================] - 0s 5ms/step - loss: 1.1486 - mae:
0.3474 - val_loss: 1.0759 - val_mae: 0.3145
Epoch 77/100
27/27 [==============================] - 0s 5ms/step - loss: 1.1416 - mae:
0.3459 - val_loss: 1.0701 - val_mae: 0.3130
Epoch 78/100
27/27 [==============================] - 0s 5ms/step - loss: 1.1349 - mae:
0.3444 - val_loss: 1.0645 - val_mae: 0.3114
Epoch 79/100
27/27 [==============================] - 0s 5ms/step - loss: 1.1282 - mae:
0.3430 - val_loss: 1.0590 - val_mae: 0.3098
Epoch 80/100
27/27 [==============================] - 0s 7ms/step - loss: 1.1217 - mae:
0.3414 - val_loss: 1.0536 - val_mae: 0.3082
Epoch 81/100
27/27 [==============================] - 0s 6ms/step - loss: 1.1154 - mae:
0.3399 - val_loss: 1.0480 - val_mae: 0.3066
Epoch 82/100
27/27 [==============================] - 0s 5ms/step - loss: 1.1090 - mae:
0.3384 - val_loss: 1.0425 - val_mae: 0.3049
Epoch 83/100
27/27 [==============================] - 0s 6ms/step - loss: 1.1028 - mae:
0.3368 - val_loss: 1.0370 - val_mae: 0.3033
Epoch 84/100
27/27 [==============================] - 0s 5ms/step - loss: 1.0964 - mae:
0.3353 - val_loss: 1.0320 - val_mae: 0.3018
Epoch 85/100
27/27 [==============================] - 0s 5ms/step - loss: 1.0904 - mae:
0.3337 - val_loss: 1.0266 - val_mae: 0.3001
Epoch 86/100
27/27 [==============================] - 0s 5ms/step - loss: 1.0842 - mae:
0.3321 - val_loss: 1.0215 - val_mae: 0.2985
Epoch 87/100
27/27 [==============================] - 0s 7ms/step - loss: 1.0783 - mae:
0.3306 - val_loss: 1.0162 - val_mae: 0.2968
Epoch 88/100
27/27 [==============================] - 0s 5ms/step - loss: 1.0723 - mae:
```

```
0.3289 - val_loss: 1.0110 - val_mae: 0.2951
Epoch 89/100
27/27 [==============================] - 0s 7ms/step - loss: 1.0664 - mae:
0.3272 - val_loss: 1.0058 - val_mae: 0.2934
Epoch 90/100
27/27 [==============================] - 0s 5ms/step - loss: 1.0604 - mae:
0.3257 - val_loss: 1.0008 - val_mae: 0.2917
Epoch 91/100
27/27 [==============================] - 0s 3ms/step - loss: 1.0546 - mae:
0.3240 - val_loss: 0.9957 - val_mae: 0.2901
Epoch 92/100
27/27 [==============================] - 0s 4ms/step - loss: 1.0487 - mae:
0.3223 - val_loss: 0.9908 - val_mae: 0.2884
Epoch 93/100
27/27 [==============================] - 0s 3ms/step - loss: 1.0431 - mae:
0.3206 - val_loss: 0.9856 - val_mae: 0.2866
Epoch 94/100
27/27 [==============================] - 0s 3ms/step - loss: 1.0372 - mae:
0.3189 - val_loss: 0.9808 - val_mae: 0.2849
Epoch 95/100
27/27 [==============================] - 0s 4ms/step - loss: 1.0317 - mae:
0.3171 - val_loss: 0.9758 - val_mae: 0.2831
Epoch 96/100
27/27 [==============================] - 0s 3ms/step - loss: 1.0259 - mae:
0.3154 - val_loss: 0.9710 - val_mae: 0.2813
Epoch 97/100
27/27 [==============================] - 0s 3ms/step - loss: 1.0204 - mae:
0.3136 - val_loss: 0.9660 - val_mae: 0.2794
Epoch 98/100
27/27 [==============================] - 0s 4ms/step - loss: 1.0148 - mae:
0.3118 - val_loss: 0.9613 - val_mae: 0.2775
Epoch 99/100
27/27 [==============================] - 0s 5ms/step - loss: 1.0093 - mae:
0.3101 - val_loss: 0.9565 - val_mae: 0.2757
Epoch 100/100
27/27 [==============================] - 0s 3ms/step - loss: 1.0038 - mae:
0.3082 - val_loss: 0.9517 - val_mae: 0.2738
```

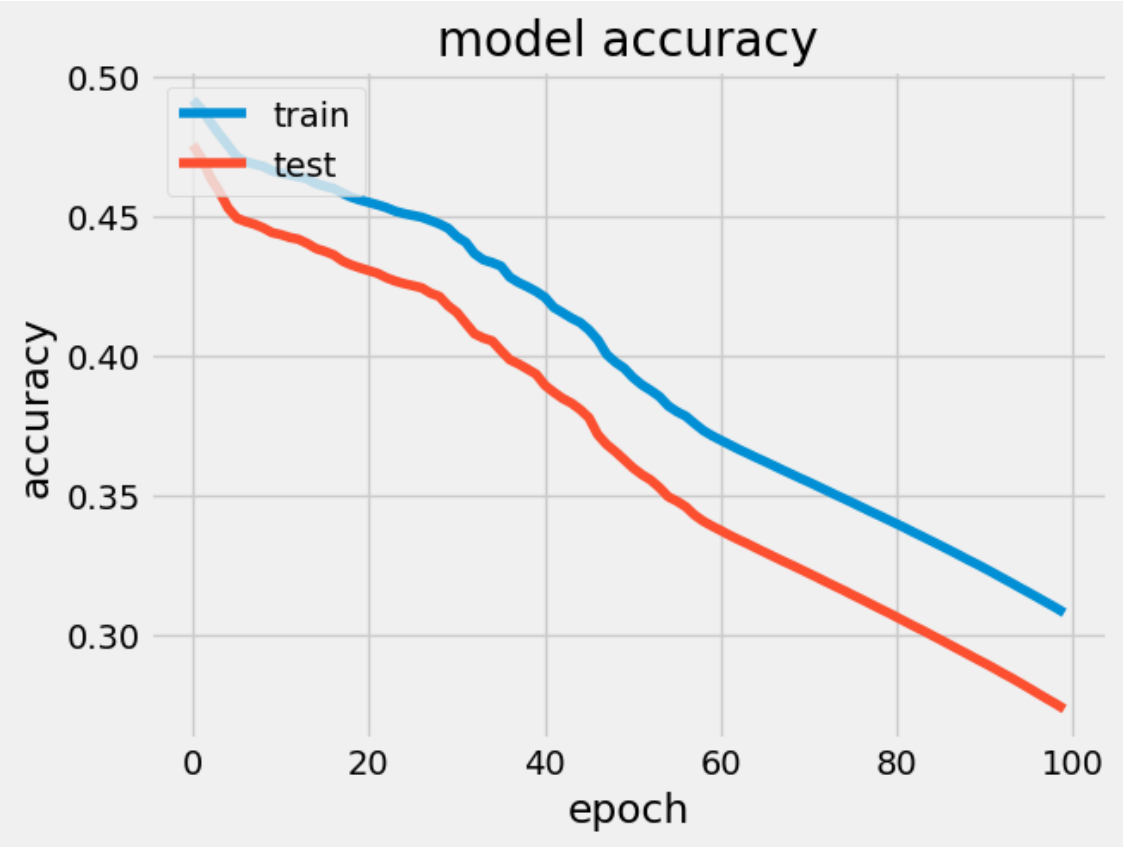
[295]:
```
print(annhistory.history.keys())
```

```
dict_keys(['loss', 'mae', 'val_loss', 'val_mae'])
```

[297]:
```
#summarize histiry for accuracy
plt.plot(annhistory.history['mae'])
plt.plot(annhistory.history['val_mae'])
plt.title('model accuracy')
plt.ylabel('accuracy')
```

```
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

# summarize history for loss
plt.plot(annhistory.history['loss'])
plt.plot(annhistory.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```




model loss
train
test
loss
epoch
7
6
5
4
3
2
1
0
20
40
60
80
100