

Project_CS230_Fire_Propagation

February 21, 2020

```
[2]: import numpy as np
import pandas as pd

from keras.models import Model
from keras.layers import Dense, Dropout, Activation
from keras.models import Sequential
from keras.layers import Input, Dense
from keras.utils import plot_model
from keras.optimizers import Adam
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt

def CreateDataset(file):
    df = pd.read_csv(file)
    df.columns = ["x_coord", "y_coord", "month", "day",
                  "ffmc", "dmc", "dc", "isi", "temp",
                  "rh", "wind", "rain", "area"]
    np.random.seed(19)

    mms = MinMaxScaler()
    df["area"] = mms.fit_transform(df["area"].values.reshape(-1, 1))

    y = df.pop("area")
    X = df

    return X, y

def create_model():
    model = Sequential()

    model.add(Dense(50, activation='relu'))
    model.add(Dense(50, activation='relu'))
    model.add(Dense(25, activation='relu'))
    model.add(Dense(10, activation='relu'))
    model.add(Dense(1, activation='linear'))
```

```

opt = Adam(lr=0.002, beta_1=0.9, beta_2=0.999, epsilon=1e-08)

model.compile(loss='mse', optimizer=opt, metrics=['accuracy'])

return model

def train_model(X, y):
    X_train, X_test, y_train, y_test = train_test_split(X.values, y.values,
↳test_size=0.15, random_state=19)

    model = create_model()

    history = model.fit(X_train, y_train, epochs=30, verbose=1,
↳validation_data=(X_test, y_test))

    plt.plot(history.history['loss'])
    plt.plot(history.history['val_loss'])
    plt.title("Model's loss")
    plt.ylabel('Loss')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Test'], loc='upper left')
    plt.savefig("Training and test losses")
    plt.show()

    return model

def main():

    X, y = CreateDataset("data.csv")

    model = train_model(X, y)
    data1 = "6,5,9,6,92.5,121.1,674.4,8.6,25.1,27,4,0"
    data2 = "2,2,8,6,93.7,231.1,715.1,8.4,21.9,42,2.2,0"

    data1 = np.array(list(map(float, data1.split(",")))).reshape(-1,1).T
    data2 = np.array(list(map(float, data2.split(",")))).reshape(-1,1).T
    predicted1 = model.predict(data1)
    predicted2 = model.predict(data2)

    print(predicted1, predicted2)

    model.save_weights("model_weights.h5")
    print("\nModel weights saved to: 'model_weights.h5'")

```

```
if __name__ == "__main__":  
    main()
```

Train on 439 samples, validate on 78 samples

Epoch 1/30

439/439 [=====] - 0s 1ms/step - loss: 105.2159 -
accuracy: 0.0228 - val_loss: 30.5532 - val_accuracy: 0.0000e+00

Epoch 2/30

439/439 [=====] - 0s 126us/step - loss: 14.6257 -
accuracy: 0.0638 - val_loss: 6.2474 - val_accuracy: 0.0385

Epoch 3/30

439/439 [=====] - 0s 99us/step - loss: 5.1964 -
accuracy: 0.0888 - val_loss: 1.8013 - val_accuracy: 0.1026

Epoch 4/30

439/439 [=====] - 0s 93us/step - loss: 2.2655 -
accuracy: 0.1777 - val_loss: 1.9851 - val_accuracy: 0.0641

Epoch 5/30

439/439 [=====] - 0s 81us/step - loss: 1.3091 -
accuracy: 0.2141 - val_loss: 0.9192 - val_accuracy: 0.2308

Epoch 6/30

439/439 [=====] - 0s 93us/step - loss: 0.8943 -
accuracy: 0.2506 - val_loss: 0.4966 - val_accuracy: 0.3462

Epoch 7/30

439/439 [=====] - 0s 79us/step - loss: 0.6189 -
accuracy: 0.2597 - val_loss: 0.5971 - val_accuracy: 0.2564

Epoch 8/30

439/439 [=====] - 0s 94us/step - loss: 0.5074 -
accuracy: 0.2961 - val_loss: 0.2700 - val_accuracy: 0.3974

Epoch 9/30

439/439 [=====] - 0s 106us/step - loss: 0.3990 -
accuracy: 0.2984 - val_loss: 0.1802 - val_accuracy: 0.4359

Epoch 10/30

439/439 [=====] - 0s 139us/step - loss: 0.2813 -
accuracy: 0.3394 - val_loss: 0.2557 - val_accuracy: 0.3846

Epoch 11/30

439/439 [=====] - 0s 131us/step - loss: 0.2289 -
accuracy: 0.3462 - val_loss: 0.1129 - val_accuracy: 0.4231

Epoch 12/30

439/439 [=====] - 0s 103us/step - loss: 0.1531 -
accuracy: 0.4009 - val_loss: 0.0963 - val_accuracy: 0.4231

Epoch 13/30

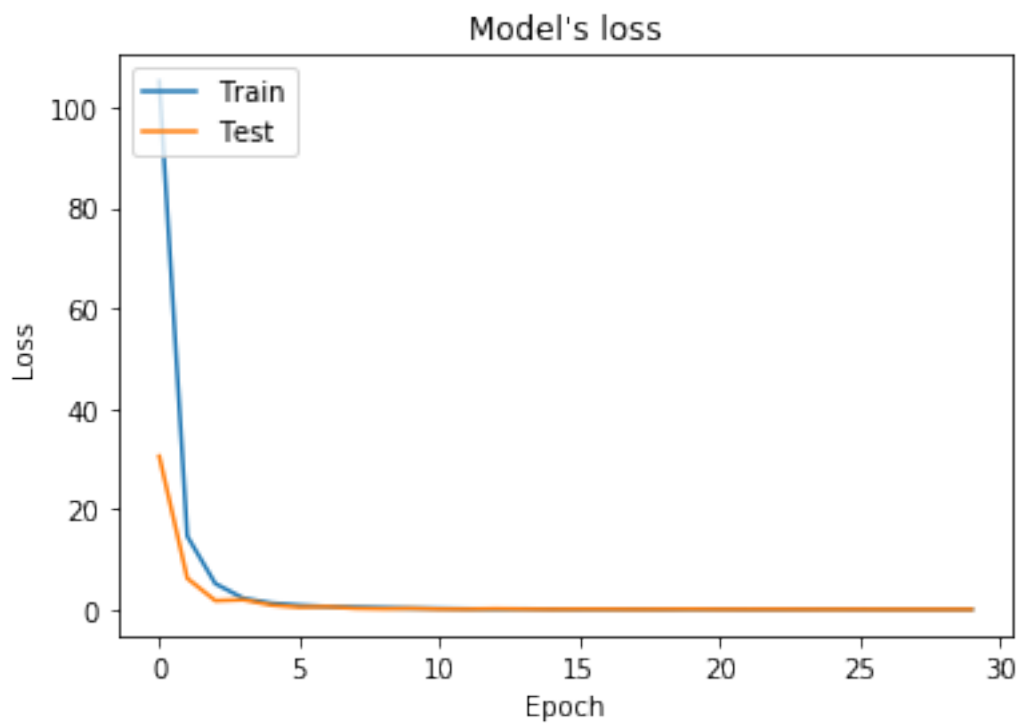
439/439 [=====] - 0s 103us/step - loss: 0.1187 -
accuracy: 0.4100 - val_loss: 0.1851 - val_accuracy: 0.4103

Epoch 14/30

439/439 [=====] - 0s 97us/step - loss: 0.1465 -
accuracy: 0.3964 - val_loss: 0.1176 - val_accuracy: 0.4103

Epoch 15/30

439/439 [=====] - 0s 103us/step - loss: 0.0985 -
accuracy: 0.4282 - val_loss: 0.0719 - val_accuracy: 0.4359
Epoch 16/30
439/439 [=====] - 0s 93us/step - loss: 0.0822 -
accuracy: 0.4328 - val_loss: 0.1010 - val_accuracy: 0.4615
Epoch 17/30
439/439 [=====] - 0s 98us/step - loss: 0.0839 -
accuracy: 0.4374 - val_loss: 0.0936 - val_accuracy: 0.4103
Epoch 18/30
439/439 [=====] - 0s 86us/step - loss: 0.0875 -
accuracy: 0.4351 - val_loss: 0.0622 - val_accuracy: 0.4615
Epoch 19/30
439/439 [=====] - 0s 81us/step - loss: 0.0627 -
accuracy: 0.4510 - val_loss: 0.0842 - val_accuracy: 0.4231
Epoch 20/30
439/439 [=====] - 0s 96us/step - loss: 0.0856 -
accuracy: 0.4305 - val_loss: 0.1113 - val_accuracy: 0.4615
Epoch 21/30
439/439 [=====] - 0s 115us/step - loss: 0.0869 -
accuracy: 0.4351 - val_loss: 0.0690 - val_accuracy: 0.4615
Epoch 22/30
439/439 [=====] - 0s 154us/step - loss: 0.0565 -
accuracy: 0.4624 - val_loss: 0.0733 - val_accuracy: 0.4744
Epoch 23/30
439/439 [=====] - 0s 120us/step - loss: 0.0576 -
accuracy: 0.4533 - val_loss: 0.0830 - val_accuracy: 0.4231
Epoch 24/30
439/439 [=====] - 0s 124us/step - loss: 0.0675 -
accuracy: 0.4533 - val_loss: 0.0624 - val_accuracy: 0.4744
Epoch 25/30
439/439 [=====] - 0s 113us/step - loss: 0.0830 -
accuracy: 0.4374 - val_loss: 0.0588 - val_accuracy: 0.4744
Epoch 26/30
439/439 [=====] - 0s 107us/step - loss: 0.0652 -
accuracy: 0.4442 - val_loss: 0.0695 - val_accuracy: 0.4615
Epoch 27/30
439/439 [=====] - 0s 116us/step - loss: 0.0458 -
accuracy: 0.4624 - val_loss: 0.0533 - val_accuracy: 0.4744
Epoch 28/30
439/439 [=====] - 0s 105us/step - loss: 0.0416 -
accuracy: 0.4647 - val_loss: 0.0557 - val_accuracy: 0.4744
Epoch 29/30
439/439 [=====] - 0s 103us/step - loss: 0.0500 -
accuracy: 0.4601 - val_loss: 0.0541 - val_accuracy: 0.4744
Epoch 30/30
439/439 [=====] - 0s 118us/step - loss: 0.0485 -
accuracy: 0.4670 - val_loss: 0.0513 - val_accuracy: 0.4744



```
[[[-0.17574708]] [[0.07705964]]
```

```
Model weights saved to: 'model_weights.h5'
```

```
[ ]:
```