

# Tensorflow DNN Model

November 28, 2022

```
[1]: import numpy as np
import pandas as pd
from metrics import MSE
import tensorflow as tf
import matplotlib.pyplot as plt
```

```
[2]: X_train = pd.read_pickle("../Datasets/final/X_train.pkl").
    ↪drop(columns=["gPlusUserId", "gPlusPlaceId"])
c = X_train.columns.tolist()
X_train = X_train.to_numpy()
X_val = pd.read_pickle("../Datasets/final/X_val.pkl").
    ↪drop(columns=["gPlusUserId", "gPlusPlaceId"]).to_numpy()
X_test = pd.read_pickle("../Datasets/final/X_test.pkl").
    ↪drop(columns=["gPlusUserId", "gPlusPlaceId"]).to_numpy()
y_train = pd.read_pickle("../Datasets/final/y_train.pkl").to_numpy()
y_val = pd.read_pickle("../Datasets/final/y_val.pkl").to_numpy()
y_test = pd.read_pickle("../Datasets/final/y_test.pkl").to_numpy()
```

```
[3]: def build_and_compile_model():
    model = tf.keras.Sequential([
        # tf.keras.layers.Dense(128, activation=tf.nn.relu),
        # tf.keras.layers.Dropout(0.4),
        # tf.keras.layers.Dense(64, activation=tf.nn.relu),
        # tf.keras.layers.Dropout(0.2),
        tf.keras.layers.Dense(16, activation=tf.nn.relu),
        # tf.keras.layers.Dropout(0.1),
        tf.keras.layers.Dense(1)
    ])
    model.compile(optimizer=tf.keras.optimizers.Adam(0.001),
                  loss=tf.keras.losses.MeanSquaredError())
    return model
```

```
[4]: dnn = build_and_compile_model()
dnn.build(X_train.shape)
dnn.summary()
```

Model: "sequential"

-----

Layer (type)	Output Shape	Param #
dense (Dense)	(484576, 16)	1360
dense_1 (Dense)	(484576, 1)	17

Total params: 1,377  
 Trainable params: 1,377  
 Non-trainable params: 0

```
[5]: early_stopping_callback = tf.keras.callbacks.EarlyStopping(monitor='val_loss',
    ↪patience=10)
checkpoint_filepath = './dnn_model_checkpoint'
model_checkpoint_callback = tf.keras.callbacks.ModelCheckpoint(
    filepath=checkpoint_filepath,
    save_weights_only=True,
    monitor='val_loss',
    mode='min',
    save_best_only=True)
```

```
[6]: history = dnn.fit(x=X_train.astype('float32'),
    y=y_train.astype('float32'),
    epochs=100,
    validation_data=(X_val.astype('float32'), y_val.
    ↪astype('float32')),
    callbacks=[early_stopping_callback,
    ↪model_checkpoint_callback])
```

```
Epoch 1/100
109/15143 [...] - ETA: 6s - loss: 7.3808

2022-11-28 22:58:18.291854: W
tensorflow/core/platform/profile_utils/cpu_utils.cc:128] Failed to get CPU
frequency: 0 Hz

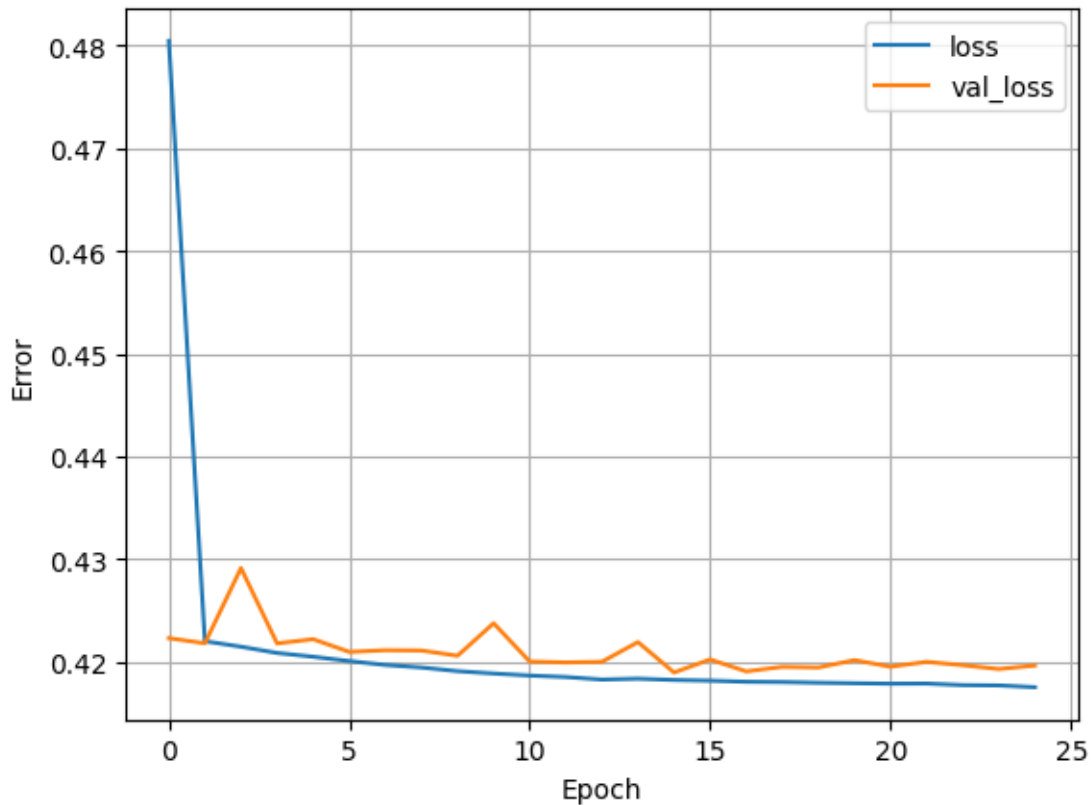
15143/15143 [=====] - 7s 477us/step - loss: 0.4805 -
val_loss: 0.4223
Epoch 2/100
15143/15143 [=====] - 7s 475us/step - loss: 0.4220 -
val_loss: 0.4218
Epoch 3/100
15143/15143 [=====] - 7s 472us/step - loss: 0.4215 -
val_loss: 0.4291
Epoch 4/100
15143/15143 [=====] - 7s 474us/step - loss: 0.4209 -
val_loss: 0.4218
Epoch 5/100
```

```
15143/15143 [=====] - 7s 472us/step - loss: 0.4205 -  
val_loss: 0.4222  
Epoch 6/100  
15143/15143 [=====] - 7s 472us/step - loss: 0.4201 -  
val_loss: 0.4210  
Epoch 7/100  
15143/15143 [=====] - 7s 472us/step - loss: 0.4197 -  
val_loss: 0.4211  
Epoch 8/100  
15143/15143 [=====] - 7s 477us/step - loss: 0.4194 -  
val_loss: 0.4211  
Epoch 9/100  
15143/15143 [=====] - 7s 470us/step - loss: 0.4191 -  
val_loss: 0.4206  
Epoch 10/100  
15143/15143 [=====] - 7s 474us/step - loss: 0.4189 -  
val_loss: 0.4238  
Epoch 11/100  
15143/15143 [=====] - 7s 477us/step - loss: 0.4187 -  
val_loss: 0.4200  
Epoch 12/100  
15143/15143 [=====] - 7s 473us/step - loss: 0.4185 -  
val_loss: 0.4199  
Epoch 13/100  
15143/15143 [=====] - 7s 475us/step - loss: 0.4183 -  
val_loss: 0.4200  
Epoch 14/100  
15143/15143 [=====] - 7s 476us/step - loss: 0.4184 -  
val_loss: 0.4219  
Epoch 15/100  
15143/15143 [=====] - 7s 471us/step - loss: 0.4182 -  
val_loss: 0.4190  
Epoch 16/100  
15143/15143 [=====] - 7s 472us/step - loss: 0.4182 -  
val_loss: 0.4202  
Epoch 17/100  
15143/15143 [=====] - 7s 477us/step - loss: 0.4180 -  
val_loss: 0.4191  
Epoch 18/100  
15143/15143 [=====] - 7s 472us/step - loss: 0.4180 -  
val_loss: 0.4195  
Epoch 19/100  
15143/15143 [=====] - 7s 472us/step - loss: 0.4180 -  
val_loss: 0.4194  
Epoch 20/100  
15143/15143 [=====] - 7s 474us/step - loss: 0.4179 -  
val_loss: 0.4202  
Epoch 21/100
```

```
15143/15143 [=====] - 7s 471us/step - loss: 0.4179 -  
val_loss: 0.4195  
Epoch 22/100  
15143/15143 [=====] - 7s 471us/step - loss: 0.4179 -  
val_loss: 0.4200  
Epoch 23/100  
15143/15143 [=====] - 7s 472us/step - loss: 0.4177 -  
val_loss: 0.4197  
Epoch 24/100  
15143/15143 [=====] - 7s 472us/step - loss: 0.4177 -  
val_loss: 0.4193  
Epoch 25/100  
15143/15143 [=====] - 7s 470us/step - loss: 0.4175 -  
val_loss: 0.4196
```

```
[7]: def plot_loss(history):  
    plt.plot(history.history['loss'], label='loss')  
    plt.plot(history.history['val_loss'], label='val_loss')  
    # plt.ylim([0.2, 1])  
    plt.xlabel('Epoch')  
    plt.ylabel('Error')  
    plt.legend()  
    plt.grid(True)
```

```
[8]: plot_loss(history=history)
```



```
[9]: dnn.load_weights(checkpoint_filepath)
```

```
[9]: <tensorflow.python.checkpoint.checkpoint.CheckpointLoadStatus at 0x289af3a00>
```

```
[10]: dnn.evaluate(X_test.astype('float32'), y_test.astype('float32'))
```

```
5048/5048 [=====] - 2s 307us/step - loss: 0.4167
```

```
[10]: 0.4166768789291382
```

```
[11]: y_pred = dnn.predict(X_test.astype('float32')).flatten()
```

```
5048/5048 [=====] - 1s 254us/step
```

```
[12]: # error = y_pred - y_test
      # plt.hist(error, bins=25)
      # plt.xlabel('Prediction Error')
      # _ = plt.ylabel('Count')
```