

Nama: Faishal Kamil	Dosen: Lili Ayu Wulandari, S.Si.M.Sc, Ph.D
Nim: 2502001063	Mata Kuliah: Deep Learning

**Perintah:** Tuning beberapa parameter seperti hidden layer, learning rate dn momentum kemudian buat table perbandingan performance accuracy, precision dan recall untuk setiap parameter yang diubah, lakukan perbandingan kemudian jelaskan hasilnya. Tuning parameter dilakukan di scikit learn maupun keras.

### MLP Keras

```
import keras
from tensorflow.keras.models import Sequential
import tensorflow as tf
model = keras.models.Sequential()
model.add(tf.keras.Input(shape=(15,)))

#Dense implements the operation: output = activation(dot(input,
kernel) + bias)
model.add(keras.layers.core.Dense(64, activation='sigmoid'))
model.add(keras.layers.core.Dense(32, activation='sigmoid'))
model.add(keras.layers.core.Dense(16, activation='sigmoid'))

# compile model with optimizer, adjusted with this learning
rate and momentum
sgd = keras.optimizers.SGD(learning_rate=0.01, momentum=0.9)
model.compile(optimizer=sgd, loss='mse', metrics=['mse', 'mae',
'mape'])
#model.compile(optimizer="sgd", loss='mse', metrics=['mse',
'mae', 'mape'])
print(model.summary())
```

Percobaan dilakukan dengan mengubah epoch, hidden layer, learning rate, Momentum, MSE (Mean Squared Error), MAE (Mean Absolute Error), dan MAPE.

Berikut adalah table perbandingannya:

Epoch ke-	Hidden Layer	Learning Rate	Momentum	MSE	MAE	MAPE
20	64, 32, 16	0.1	0.9	3277.9980	50.0069	625095.25
20	64, 32, 16	0.1	0.5	3277.999	50.0069	625080.6250
50	64, 32, 16	0.1	0.5	3277.9980	50.0069	625095.3125
50	64, 32, 16	0.1	0.9	3290.8289	50.1263	625095.1875
5	64, 32, 16	0.05	0.03	3290.8406	50.1263	625025.1875
5	64, 32, 16	0.05	0.07	3290.8784	50.1267	624811.2500

Catatan:

- Untuk kolom "Epoch ke-" merupakan epoch yang dilakukan pada trainingnya dan yang diambil merupakan nilai epoch yang terakhir

## Kesimpulan

Berdasarkan percobaan yang dilakukan, nilai akurasi merupakan sebuah metric yang dihitung pada saat compile sebuah model yaitu MSE, MAE, dan MAPE. Ketiga metrics ini merupakan sebuah bentuk evaluasi untuk model regresi. MSE merupakan rata-rata squared error antara predicted dan true value. MAE merupakan rata-rata dari absolut error antara predicted dan true values, dan MAPE merupakan nilai rata-rata dari dari absolute percentage errors antara predicted dan true values. Semakin kecil nilai dari ketiga metrics ini, semakin baik performa model dalam melakukan prediksi. Terdapat banyak faktor yang dapat mempengaruhi kinerja akurasi model deep learning, di antaranya adalah:

1. Kualitas dan kuantitas data train, semakin banyak dan semakin baik kualitas data train, semakin baik juga performa model. Selain itu, data yang seimbang dan representatif juga penting.
2. Arsitektur model, arsitektur model yang baik dapat membantu meningkatkan kinerja model. Hal ini mencakup jumlah dan ukuran hidden layer, fungsi aktivasi, loss function, dan banyak lagi.
3. Hyperparameter, hyperparameter seperti learning rate, besaran epoch, ukuran batch, hidden layer, momentum, dan lain-lain dapat mempengaruhi kinerja model. Dalam beberapa kasus, meningkatkan atau menurunkan hyperparameter dapat meningkatkan performa model.

Namun, optimasi hyperparameter bukanlah satu-satunya faktor yang mempengaruhi performa model. Ada banyak faktor lain yang harus diperhatikan, seperti ukuran dataset, kompleksitas model, kecepatan pelatihan, dan sebagainya.

## MLP Scikit-Learn

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error
from sklearn.preprocessing import StandardScaler

# Load data
df = pd.read_csv('Customers.csv')
df['Profession'].fillna('others', inplace=True)
df_encode={"Gender": {"Male":1,"Female":0}}
df=df.replace(df_encode)
df_enc=pd.get_dummies(df['Profession'], prefix='Profession')
df=pd.concat([df,df_enc], axis=1)

# Define input and target
df_input = df[['Gender', 'Age', 'Annual Income ($)', 'Work Experience',
'Family Size', 'Profession_Artist', 'Profession_Doctor',
'Profession_Engineer', 'Profession_Entertainment',
'Profession_Executive', 'Profession_Healthcare', 'Profession_Homemaker',
'Profession_Lawyer', 'Profession_Marketing',
'Profession_others']]
df_target = df[['Spending Score (1-100)']]

# Split data into train and test sets
```

```

X_train, X_test, y_train, y_test = train_test_split(df_input, df_target,
test_size=0.2, random_state=42)

# Scale data
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Define parameter combinations
hidden_layer_sizes = [(30,), (50,), (30, 30), (50, 50)]
learning_rates = [0.001, 0.01, 0.1]
momentums = [0.0, 0.5, 0.9]

# Create a table to store the results
results = pd.DataFrame(columns=['hidden_layer_sizes', 'learning_rate',
'momentum', 'MSE', 'MAE', 'MAPE'])

# Loop through each parameter combination
for hl_size in hidden_layer_sizes:
    for lr in learning_rates:
        for momentum in momentums:

            # Train the model with the current parameter combination
            model_reg = MLPRegressor(hidden_layer_sizes=hl_size,
                                    activation='logistic',
                                    solver='sgd',
                                    learning_rate_init=lr,
                                    momentum=momentum,
                                    random_state=42)

            model_reg.fit(X_train_scaled, y_train.values.ravel())

            # Evaluate the model on test set
            y_pred = model_reg.predict(X_test_scaled)
            mse = mean_squared_error(y_test, y_pred)
            mae = mean_absolute_error(y_test, y_pred)
            mape = np.mean(np.abs((y_test - y_pred) / y_test)) * 100

            # Append the results to the table
            results = results.append({'hidden_layer_sizes': hl_size,
                                    'learning_rate': lr,
                                    'momentum': momentum,
                                    'MSE': mse,
                                    'MAE': mae,
                                    'MAPE': mape},
                                    ignore_index=True)

# Print the results
print(results)

```

Berikut adalah table perbandigannya:

Epoch	HL	LR	MM	MSE	MAE	MAPE
0	(30, 30)	0.001	0.1	767.344294	23.48	1.695748
1	(30, 30)	0.001	0.5	767.338404	23.481049	1.698389
2	(30, 30)	0.001	0.9	767.344646	23.48	1.695673
3	(30, 30)	0.01	0.1	767.716391	23.49637	1.719081
4	(30, 30)	0.01	0.5	768.582484	23.513546	1.736072
5	(30, 30)	0.01	0.9	770.534862	23.550533	1.759165
6	(30, 30)	0.1	0.1	767.962523	23.48	1.672016
7	(30, 30)	0.1	0.5	771.512765	23.500125	1.630722
8	(30, 30)	0.1	0.9	767.550591	23.48	1.682975
9	(50, 50)	0.001	0.1	767.348491	23.48	1.69496
10	(50, 50)	0.001	0.5	767.354894	23.48	1.694029
11	(50, 50)	0.001	0.9	767.338423	23.48088	1.69816
12	(50, 50)	0.01	0.1	767.550464	23.492513	1.713871
13	(50, 50)	0.01	0.5	767.457413	23.489625	1.70997
14	(50, 50)	0.01	0.9	768.45115	23.510218	1.733994
15	(50, 50)	0.1	0.1	767.56283	23.492844	1.714318
16	(50, 50)	0.1	0.5	767.398074	23.487107	1.70657
17	(50, 50)	0.1	0.9	777.172781	23.662598	1.806176
18	(100, 100)	0.001	0.1	767.38258	23.48	1.691309
19	(100, 100)	0.001	0.5	767.478076	23.48	1.685867
20	(100, 100)	0.001	0.9	767.371925	23.48	1.69221
21	(100, 100)	0.01	0.1	769.446434	23.488238	1.650157
22	(100, 100)	0.01	0.5	768.827113	23.483603	1.657776
23	(100, 100)	0.01	0.9	767.344052	23.48	1.695801
24	(100, 100)	0.1	0.1	768.432948	23.480126	1.663502
25	(100, 100)	0.1	0.5	767.353847	23.484093	1.702497
26	(100, 100)	0.1	0.9	1951094076 7296557699 56352.0	139681569 1753.8034 67	66054034368.69855 5

Catatan:

- HL = Hidden Layer
- LR = Learning Rate
- MM = Momentum

Kesimpulan

Beberapa parameter memberikan hasil yang baik pada model, seperti pada Epoch ke-10 yang menghasilkan nilai MSE (767.354894) dan MAE (23.48) terendah. Namun, ada juga parameter yang menghasilkan performa yang buruk, seperti pada Epoch ke-26 yang memiliki nilai MSE yang jauh lebih tinggi dibandingkan ketiga metrik performa lainnya. Sebagai hasilnya, untuk memilih model terbaik, disarankan untuk memilih parameter pada Epoch ke-10 dengan hidden layer (50, 50), learning rate 0.001, dan momentum 0.5 karena memberikan hasil yang lebih baik dibandingkan dengan metrik performa lainnya.