

- NAME :- JUPUDI VISHNU PAVAN KUMAR
- SID NUMBER :- 2449129
- ASSIGNMENT :- WEEK 4

Jupyter Week-4-MLP_S&P_ML_in-Finance_Final Last Checkpoint: 5 days ago

File Edit View Run Kernel Settings Help

Not Trusted

JupyterLab Python 3 (ipykernel)

Lab Logbook Requirement:

1. Create your own Multi-layer Perceptron (MLP) with two hidden layers, where the first hidden layer cells' number equals the last three digits of your SID. The number of cells in the next hidden layer is approximately two times smaller.

For example, if your SID is 2287167, the number of cells on the first hidden layer is 167, and on the second - 84. Take epochs=10. Leave other parameters the same as in the practical session.

2. Compile the model.
3. Train your MLP with the same datasets and demonstrate the received MAE on the test dataset.
4. Compare your MAE with the MAE of the MLP in the practical session.
5. Please only add to your Lab Logbook a print-screen(s) of your MLP architecture using `model.summary()`, MLP training code and process, and the resulting MAE on the test dataset.

Jupyter Week-4-MLP_S&P_ML_in-Finance_Final Last Checkpoint: 5 days ago

File Edit View Run Kernel Settings Help

Not Trusted

JupyterLab Python 3 (ipykernel)

```
[20]: import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.metrics import MeanAbsoluteError as MAE
from tensorflow.keras.datasets import boston_housing
from sklearn.preprocessing import StandardScaler
(x_train, y_train), (x_test, y_test) = boston_housing.load_data()
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
n1 = 129
n2 = 99
model = Sequential([
    Dense(n1, activation='relu', input_shape=(x_train.shape[1],)),
    Dense(n2, activation='relu'),
    Dense(1)
])
model.compile(
    optimizer=Adam(),
    loss='mae',
    metrics=[MAE()]
)
model.summary()
history = model.fit(
    x_train, y_train,
```



```
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
n1 = 129
n2 = 99
model = Sequential([
    Dense(n1, activation='relu', input_shape=(x_train.shape[1],)),
    Dense(n2, activation='relu'),
    Dense(1)
])
model.compile(
    optimizer=Adam(),
    loss='mae',
    metrics=[MAE()]
)
model.summary()
history = model.fit(
    x_train, y_train,
    epochs=10,
    batch_size=32,
    validation_split=0.2,
    verbose=1
)
test_loss, test_mae = model.evaluate(x_test, y_test, verbose=0)
print(f"\nTest MAE: {test_mae:.4f}")
```

Jupyter Week-4-MLP_S&P_ML_in-Finance_Final Last Checkpoint: 5 days ago

File Edit View Run Kernel Settings Help

Not Trusted

JupyterLab Python 3 (ipykernel)

Model: "sequential_4"

Layer (type)	Output Shape	Param #
dense_12 (Dense)	(None, 129)	1,806
dense_13 (Dense)	(None, 99)	12,870
dense_14 (Dense)	(None, 1)	100

Total params: 14,776 (57.72 KB)

Trainable params: 14,776 (57.72 KB)

Non-trainable params: 0 (0.00 B)

Epoch 1/10
11/11 5s 47ms/step - loss: 21.8810 - mean_absolute_error: 21.8810 - val_loss: 22.4340 - val_mean_absolute_error: 22.4340
Epoch 2/10
11/11 0s 14ms/step - loss: 20.1146 - mean_absolute_error: 20.1146 - val_loss: 20.6063 - val_mean_absolute_error: 20.6063

Jupyter Week-4-MLP_S&P_ML_in-Finance_Final Last Checkpoint: 5 days ago

File Edit View Run Kernel Settings Help

Not Trusted

JupyterLab Python 3 (ipykernel)

Epoch 1/10
11/11 5s 47ms/step - loss: 21.8810 - mean_absolute_error: 21.8810 - val_loss: 22.4340 - val_mean_absolute_error: 22.4340
Epoch 2/10
11/11 0s 14ms/step - loss: 20.1146 - mean_absolute_error: 20.1146 - val_loss: 20.6063 - val_mean_absolute_error: 20.6063
Epoch 3/10
11/11 0s 12ms/step - loss: 17.9545 - mean_absolute_error: 17.9545 - val_loss: 17.9962 - val_mean_absolute_error: 17.9962
Epoch 4/10
11/11 0s 14ms/step - loss: 14.8249 - mean_absolute_error: 14.8249 - val_loss: 14.1738 - val_mean_absolute_error: 14.1738
Epoch 5/10
11/11 0s 17ms/step - loss: 10.7897 - mean_absolute_error: 10.7897 - val_loss: 9.4211 - val_mean_absolute_error: 9.4211
Epoch 6/10
11/11 0s 17ms/step - loss: 7.4454 - mean_absolute_error: 7.4454 - val_loss: 6.8670 - val_mean_absolute_error: 6.8670
Epoch 7/10
11/11 0s 18ms/step - loss: 6.0474 - mean_absolute_error: 6.0474 - val_loss: 5.8240 - val_mean_absolute_error: 5.8240
Epoch 8/10
11/11 0s 16ms/step - loss: 5.1088 - mean_absolute_error: 5.1088 - val_loss: 5.0319 - val_mean_absolute_error: 5.0319
Epoch 9/10
11/11 0s 16ms/step - loss: 4.3735 - mean_absolute_error: 4.3735 - val_loss: 4.4452 - val_mean_absolute_error: 4.4452
Epoch 10/10
11/11 0s 16ms/step - loss: 3.8682 - mean_absolute_error: 3.8682 - val_loss: 4.0433 - val_mean_absolute_error: 4.0433
Test MAE: 4.5674

```
[21]: print(f"Mean Absloute Error in test data: {test_mae:.4f}")
```

Mean Absloute Error in test data: 4.5674