

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”  
КАФЕДРА ІНФОРМАТИКИ ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Проектування та дослідження систем із штучним інтелектом

**ЗВІТ**

до лабораторної роботи №3

Виконав

ІП-41мн Пашковський Євгеній Сергійович

Київ 2025

## Завдання

Написати програму що реалізує нейронну мережу прямого розповсюдження для розпізнавання рукописних цифр.

## Хід роботи

### Код програми

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

```
with np.load('mnist.npz') as npz:
    x_train, y_train, x_test, y_test = [npz[k].astype('float32') for k
in ['x_train', 'y_train', 'x_test', 'y_test']]

fig, ax = plt.subplots(figsize=(20, 4), ncols=5)
for a in ax:
    i = np.random.randint(x_train.shape[0])
    a.matshow(x_train[i], cmap='gray')
    a.set_title(f'Label: {y_train[i]}')
    a.axis('off')

x_train = np.array(x_train).reshape((60000, 784))
x_test = np.array(x_test).reshape((10000, 784))

y_train = y_train[:, np.newaxis]
y_test = y_test[:, np.newaxis]

print(f'x_train shape: {x_train.shape}')
print(f'x_test shape: {x_test.shape}')

print(f'y_train shape: {y_train.shape}')
print(f'y_test shape: {y_test.shape}')
```

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Input, Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.losses import SparseCategoricalCrossentropy
```

```

import keras_tuner as kt

def build_model(hp):
    hp_layers = hp.Int('layers', min_value=1, max_value=10, step=1)
    hp_learning_rate = hp.Float(
        'learning_rate',
        min_value=1e-6,
        max_value=1e-2,
        sampling='LOG',
        default=1e-3
    )

    model = Sequential()

    model.add(Input(shape=(784,)))

    for i in range(hp_layers):
        model.add(Dense(hp.Int(f'units_{i}', min_value=32, max_value=2048,
step=32), activation='relu'))
        model.add(Dropout(rate=hp.Float(
            f'dropout_rate_{i}',
            min_value=0.0,
            max_value=0.5,
            default=0.25,
            step=0.05,
        )))

    model.add(Dense(10, activation='softmax'))

    model.compile(optimizer=Adam(learning_rate=hp_learning_rate),
                  loss=SparseCategoricalCrossentropy(),
                  metrics=['accuracy'])

    return model

tuner = kt.Hyperband(build_model,
                     objective='val_accuracy',
                     max_epochs=10,
                     factor=3,
                     directory='tmp',
                     project_name='mnist_MLP')

```

```
stop_early = tf.keras.callbacks.EarlyStopping(monitor='val_loss',
patience=5)

tuner.search(x_train, y_train, epochs=500, validation_split=0.2,
callbacks=[stop_early])

best_hps=tuner.get_best_hyperparameters(num_trials=1)[0]

print(f"Найкращі параметри:\n")
print(best_hps.values)
```

```
hypermodel = tuner.hypermodel.build(best_hps)

hypermodel.fit(x_train, y_train, epochs=500, validation_split=0.2)
```

```
eval_result = hypermodel.evaluate(x_test, y_test)
print("[Loss, Accuracy]:", eval_result)
```

```
predictions = hypermodel.predict(x_test).argmax(axis=1)

predictions
```

```
from sklearn.metrics import confusion_matrix

conf_matrix = confusion_matrix(y_test, predictions)
fig, ax = plt.subplots(figsize=(30, 8))
cax = ax.matshow(conf_matrix, cmap="Blues")

plt.colorbar(cax)

class_labels=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

ax.set_xticks(range(len(class_labels)))
ax.set_yticks(range(len(class_labels)))
ax.set_xticklabels(class_labels, fontsize=7)
ax.set_yticklabels(class_labels, fontsize=7)

plt.title("Confusion Matrix", pad=20)
plt.xlabel("Predicted", labelpad=10)
plt.ylabel("Actual", labelpad=10)
```

```

for (i, j), value in np.ndenumerate(conf_matrix):
    ax.text(j, i, f'{value}', ha='center', va='center', color='black',
            fontsize=7)

plt.tight_layout()
plt.show()

```

Отриманий результат

```

[Loss, Accuracy]: [0.2070523500442505, 0.9839000105857849]

```

Рисунок 1 – Кінцева похибка та точність отриманої моделі

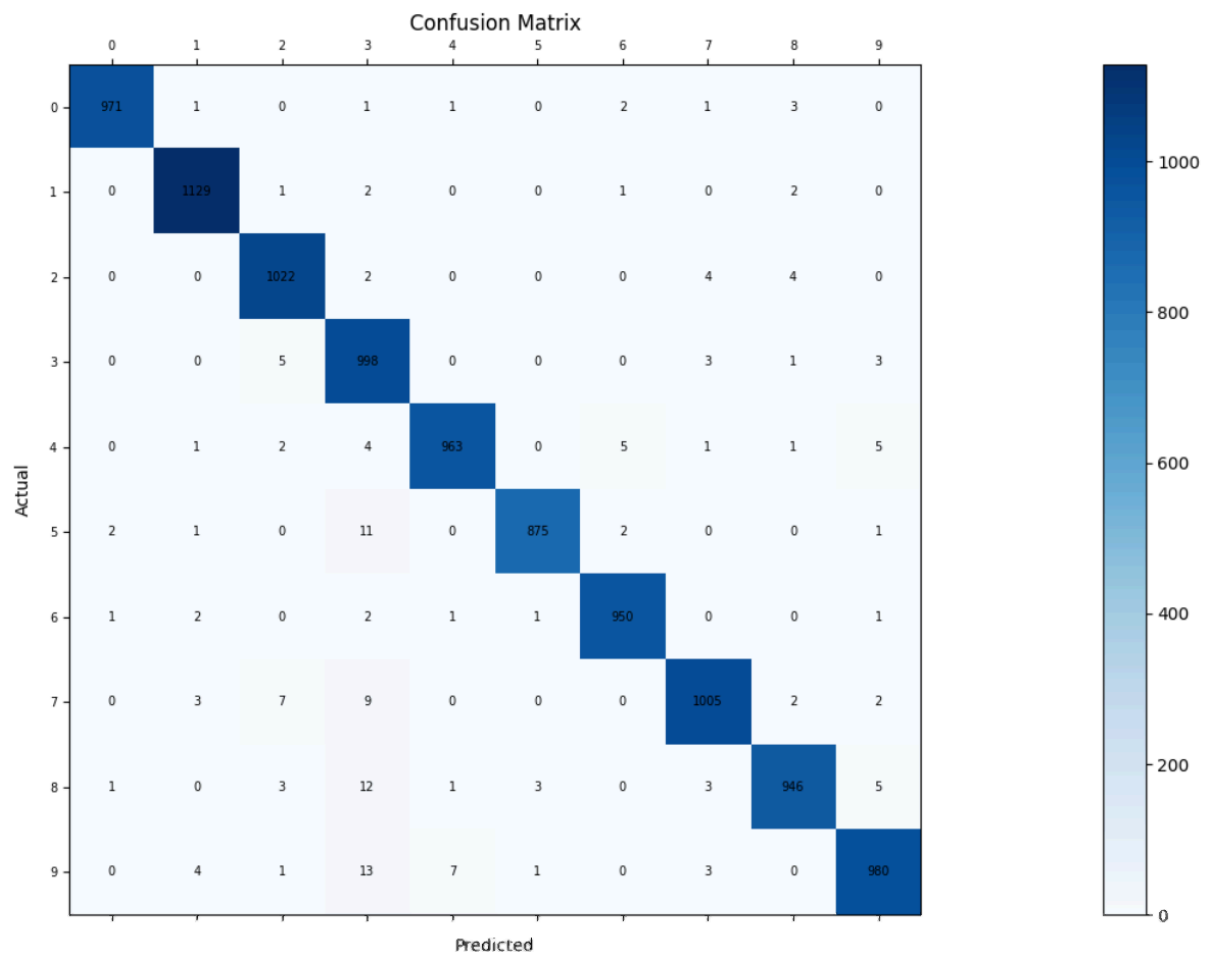


Рисунок 2 – Confusion matrix результатів від отриманої моделі

Висновки

В рамках цієї лабораторної роботи було побудовано, навчено та проаналізовано модель прямого розповсюдження для розпізнавання рукописних цифр. Модель показує досить високу точність та тестових даних (більше 98%).