

# WSILista1

PAWEŁ KĘDZIERSKI

March 27, 2025

## Zadanie 1

Celem zadania było stworzenie i wytrenowanie sieci neuronowej, która będzie rozpoznawała cyfry z podanego zbioru danych, a także ocena wydajności modelu na podstawie różnych wskaźników: **dokładności, czułości i precyzji**.

## Opis użytych narzędzi

Do realizacji zadania użyłem **Keras** (część TensorFlow) jako frameworku do budowy i trenowania sieci neuronowej. Model jest prostą siecią neuronową, wykorzystującą warstwy gęste (dense layers) oraz funkcję aktywacji ReLU. Do trenowania modelu użyłem funkcji **Adam** jako optymalizatora oraz **SparseCategoricalCrossentropy** jako funkcji straty, co jest odpowiednie dla problemów klasyfikacji wieloklasowej.

## Parametry sieci neuronowej

- **Liczba epok:** 10
- **Liczba warstw:** 2 warstwy gęste (Dense)
- **Funkcja aktywacji:** ReLU dla warstw ukrytych, Softmax dla warstwy wyjściowej

## Opis procesu trenowania

Model został trenowany przez 10 epok. Oto wyniki dla każdej z epok:

- **Epoka 1:** Dokładność = 15.62%, Strata na testowym = 2.436, Strata na walidacyjnym = 0.1275
- **Epoka 2:** Dokładność = 35.29%, Strata na testowym = 1.9344, Strata na walidacyjnym = 0.0917
- **Epoka 3:** Dokładność = 95.62%, Strata na testowym = 0.1518, Strata na walidacyjnym = 0.0834
- **Epoka 4:** Dokładność = 96.68%, Strata na testowym = 0.1152, Strata na walidacyjnym = 0.0711
- **Epoka 5:** Dokładność = 97.05%, Strata na testowym = 0.0992, Strata na walidacyjnym = 0.0672
- **Epoka 6:** Dokładność = 97.50%, Strata na testowym = 0.0817, Strata na walidacyjnym = 0.0709
- **Epoka 7:** Dokładność = 97.63%, Strata na testowym = 0.0785, Strata na walidacyjnym = 0.0717
- **Epoka 8:** Dokładność = 97.60%, Strata na testowym = 0.0775, Strata na walidacyjnym = 0.0637
- **Epoka 9:** Dokładność = 97.87%, Strata na testowym = 0.0693, Strata na walidacyjnym = 0.0698
- **Epoka 10:** Dokładność = 97.82%, Strata na testowym = 0.0659, Strata na walidacyjnym = 0.0653

W każdej epoce dokładność modelu wzrastała, a strata na zbiorze walidacyjnym malała, co wskazuje na postępującą poprawę wydajności modelu. Model osiągnął najwyższą dokładność na epokę 10, gdzie dokładność wyniosła 97.82%, a strata na zbiorze testowym spadła do 0.0659.

## Wyniki na zbiorze testowym

Po zakończeniu trenowania modelu, testowaliśmy go na zbiorze testowym (10000 obrazów). Oto wyniki:

- **Test Accuracy:** 98.19%

To oznacza, że sieć neuronowa rozpoznała poprawnie 98.19% obrazów z testowego zbioru danych, co wskazuje na bardzo dobrą zdolność rozpoznawania cyfr.

## Czułość i precyzja

Obliczyłem również miary jakości klasyfikacji: **precyzję**, **czułość** oraz **f1-score** dla każdej z cyfr:

	precision	recall	f1-score
0	0.9861	0.9950	0.9905
1	0.9920	0.9910	0.9915
2	0.9693	0.9800	0.9746
3	0.9808	0.9690	0.9748
4	0.9819	0.9740	0.9779
5	0.9733	0.9860	0.9796
6	0.9920	0.9930	0.9925
7	0.9910	0.9870	0.9890
8	0.9837	0.9670	0.9753
9	0.9692	0.9770	0.9731

## Wnioski

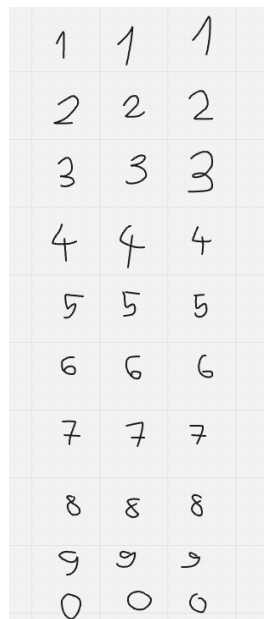
- Model osiągnął bardzo wysoką **dokładność** na zbiorze testowym (98.19%).
- **Precyzja** oraz **czułość** są również na wysokim poziomie, co oznacza, że model radzi sobie dobrze zarówno z dokładnym przewidywaniem cyfr, jak i z wykrywaniem rzeczywistych cyfr w danych testowych.
- **F1-score** jest również wysokie, co świadczy o dobrej równowadze między precyzją a czułością.

## Podsumowanie

Stworzony model sieci neuronowej osiągnął bardzo dobry wynik na zbiorze testowym, z wysoką dokładnością, precyzją, czułością i F1-score. Model jest skuteczny w rozpoznawaniu cyfr w zbiorze MNIST.

## Zadanie 2

Celem zadania było stworzenie własnego zbioru testowego zawierającego co najmniej trzy próbki każdej cyfry, a następnie przetestowanie, jak model rozpoznający cyfry, wytrenowany w poprzednim zadaniu, radzi sobie z nowymi danymi.



## Własny zbiór testowy - Rysunek 1

Do testowania modelu wykorzystano własne próbki pisma, które były zapisane w formacie .png. Każda próbka zawierała cyfry zapisane odręcznie.

## Przetwarzanie danych

Obrazy zostały wczytane przy użyciu biblioteki PIL, przekonwertowane na skalę szarości, obrócone o 90 stopni, a także zmieniono ich rozmiar na 28x28 pikseli, co odpowiada wymaganiom sieci neuronowej wytrenowanej na zbiorze MNIST. Obrazy zostały również odwrócone (czyli ciemniejsze fragmenty stały się jaśniejsze i odwrotnie).

## Model

Do klasyfikacji obrazów wykorzystano wcześniej wytrenowaną sieć neuronową zapisaną w pliku `trained_model.h5`. Model ten został wytrenowany na zbiorze MNIST.

## Wyniki

Oto szczegółowe wyniki:

	precision	recall	f1-score	support
0	1.0000	1.0000	1.0000	3
1	0.0000	0.0000	0.0000	3
2	0.5000	1.0000	0.6667	3
3	1.0000	1.0000	1.0000	3
4	0.7500	1.0000	0.8571	3
5	0.4000	0.6667	0.5000	3
6	0.0000	0.0000	0.0000	3
7	0.7500	1.0000	0.8571	3
8	0.5000	0.3333	0.4000	3
9	0.0000	0.0000	0.0000	3
accuracy			0.6000	30
macro avg	0.4900	0.6000	0.5281	30
weighted avg	0.4900	0.6000	0.5281	30

## Wnioski z analizy wyników

- **Współczynnik rozpoznawalności:** Model uzyskał ogólną dokładność 60% (0.6000), co jest wynikiem relatywnie niskim w porównaniu do testów na zbiorze MNIST, gdzie dokładność była znacznie wyższa.
- **Błędy:**
  - Model bardzo dobrze poradził sobie z niektórymi cyframi, uzyskując dokładność 100% dla cyfr 0, 3, oraz 4. Przyczyny tej wysokiej skuteczności mogą obejmować czytelność tych cyfr w próbkach lub bliskość ich pisma do cyfrowych reprezentacji.
  - Jednakże dla cyfr 1, 6, 9 model osiągnął 0% precyzji i czułości, co wskazuje na bardzo słabą zdolność rozpoznawania tych cyfr w próbce własnej.
- **Możliwe przyczyny błędów:**
  - Własne próbki pisma mogą różnić się od standardowych cyfr w zbiorze MNIST, przez co model ma trudności w ich rozpoznaniu. Modele wytrenowane na takich zbiorach jak MNIST mogą nie radzić sobie dobrze z różnorodnymi stylami pisma, szczególnie gdy są one zniekształcone lub nieuporządkowane.

## Wizualizacja wyników

Obrazy, na których model popełnił błędy, zostały wyświetlone na wykresach, aby zobaczyć, jak wyglądały próbki pisma, które sprawiły modelowi trudność. Wykresy te mogą wskazać, jakie cechy obrazów były problematyczne.

## 1 Cel zadania

Celem zadania było stworzenie własnego zbioru testowego zawierającego co najmniej trzy próbki każdej cyfry, a następnie przetestowanie, jak model rozpoznający cyfry, wytrenowany w poprzednim zadaniu, radzi sobie z nowymi danymi.

## Własny zbiór testowy

Do testowania modelu wykorzystano własne próbki pisma, które były zapisane w formacie `.png`. Każda próbka zawierała cyfry zapisane odręcznie. Aby przygotować dane, każdemu obrazowi przypisano odpowiednią etykietę, zgodnie z cyfrą, którą przedstawiała. W sumie przygotowano 30 obrazów, po trzy próbki dla każdej cyfry (0-9).

## Przetwarzanie danych

Obrazy zostały wczytane przy użyciu biblioteki `PIL`, przekonwertowane na skalę szarości, obrócone o 90 stopni, a także zmieniono ich rozmiar na 28x28 pikseli, co odpowiada wymaganiom sieci neuronowej wytrenowanej na zbiorze MNIST. Obrazy zostały również znormalizowane (wartości pikseli zostały skalowane do zakresu od 0 do 1) oraz odwrócone (czyli ciemniejsze fragmenty stały się jaśniejsze i odwrotnie).

## Model

Do klasyfikacji obrazów wykorzystano wcześniej wytrenowaną sieć neuronową zapisaną w pliku `trained_model.h5`. Model ten został wytrenowany na zbiorze MNIST, a jego struktura i parametry zostały wykorzystane do klasyfikacji nowych danych.

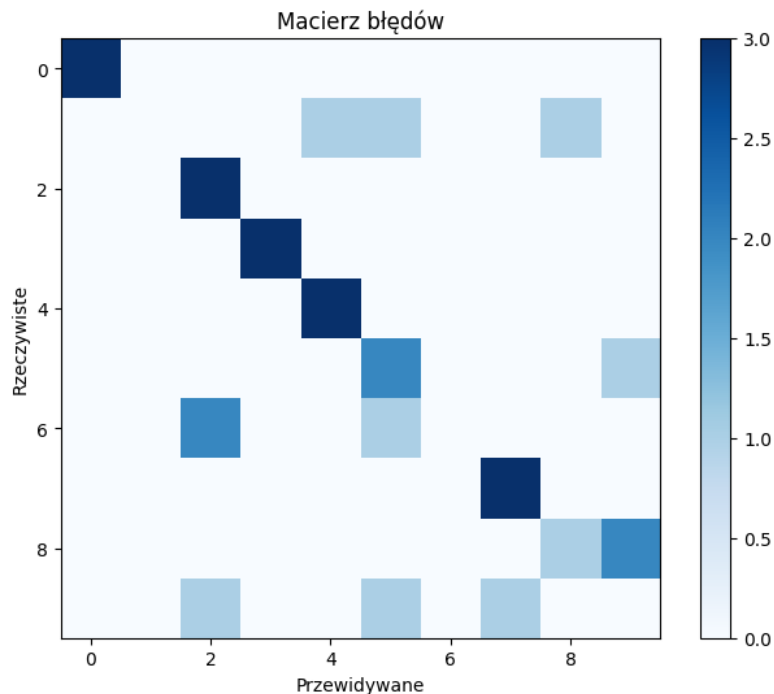
## Wyniki klasyfikacji

Po wczytaniu własnych danych testowych, model dokonał klasyfikacji, a wyniki zostały przedstawione w formie raportu klasyfikacyjnego. Oto szczegóły wyników:

	precision	recall	f1-score	support
0	1.0000	1.0000	1.0000	3
1	0.0000	0.0000	0.0000	3
2	0.5000	1.0000	0.6667	3
3	1.0000	1.0000	1.0000	3
4	0.7500	1.0000	0.8571	3
5	0.4000	0.6667	0.5000	3
6	0.0000	0.0000	0.0000	3
7	0.7500	1.0000	0.8571	3
8	0.5000	0.3333	0.4000	3
9	0.0000	0.0000	0.0000	3
accuracy			0.6000	30
macro avg	0.4900	0.6000	0.5281	30
weighted avg	0.4900	0.6000	0.5281	30

## Wnioski z analizy wyników

- **Współczynnik rozpoznawalności:** Model uzyskał ogólną dokładność 60% (0.6000), co jest wynikiem relatywnie niskim w porównaniu do testów na zbiorze MNIST, gdzie dokładność była znacznie wyższa.
- **Błędy:**
  - Model bardzo dobrze poradził sobie z niektórymi cyframi, uzyskując dokładność 100% dla cyfr 0, 3, oraz 4. Przyczyny tej wysokiej skuteczności mogą obejmować czytelność tych cyfr w próbkach lub bliskość ich pisma do cyfrowych reprezentacji.
  - Jednakże dla cyfr 1, 6, 9 model osiągnął 0% precyzji i czułości, co wskazuje na bardzo słabą zdolność rozpoznawania tych cyfr w próbce własnej.
- **Możliwe przyczyny błędów**  
Własne próbki pisma mogą różnić się od standardowych cyfr w zbiorze



MNIST, przez co model ma trudności w ich rozpoznaniu. Modele wytrenowane na takich zbiorach jak MNIST mogą nie radzić sobie dobrze z różnorodnymi stylami pisma, szczególnie gdy są one zniekształcone.

### Zadanie 3

Celem zadania było stworzenie klasyfikatora opierającego się na algorytmie **Random Forest**, który będzie rozpoznawał cyfry na podstawie zbioru danych **EMNIST**. Model ten został wytrenowany na zbiorze treningowym, a następnie oceniony na zbiorze testowym. Zadanie obejmowało obliczenie współczynnika prawidłowej rozpoznawalności, precyzji oraz czułości klasyfikatora.



## Opis metodologii

Do klasyfikacji zastosowano algorytm Random Forest z 100 drzewami decyzyjnymi. Model został wytrenowany na zbiorze treningowym, a jego wyniki oceniono na zbiorze testowym.

## Wyniki

Po wytrenowaniu modelu uzyskano następujące wyniki:

Dokładność = 0.9738

Cyfra	Precyzja	Czułość	F1-score	Support
0	0.98	0.99	0.99	1000
1	0.99	0.99	0.99	1000
2	0.96	0.97	0.97	1000
3	0.97	0.96	0.96	1000
4	0.96	0.97	0.97	1000
5	0.97	0.97	0.97	1000
6	0.99	0.99	0.99	1000
7	0.99	0.98	0.98	1000
8	0.97	0.95	0.96	1000
9	0.96	0.96	0.96	1000
Średnia	0.97	0.97	0.97	10000

Wartości precyzji, czułości i F1-score dla każdej cyfry są zadowalające, a średnia dokładność dla całego modelu również jest wysoka.

Na podstawie macierzy błędów można zauważyć, że model najczęściej myli cyfry 3 i 5, oraz 8 i 9, co jest typowe w przypadku ręcznie pisanych cyfr, gdzie te znaki są podobne.

## Wnioski

Model Random Forest osiągnął bardzo wysoką dokładność na zbiorze testowym wynoszącą 97.38%. Precyzja, czułość i F1-score dla wszystkich cyfr są również zadowalające, a w szczególności dla cyfr 0, 1, 6 i 7, gdzie model osiągnął wyniki bliskie 1. W przypadku cyfr takich jak 2, 3, 8 i 9, chociaż wyniki są nadal dobre, występują drobne błędy, które mogą wynikać z podobieństwa tych cyfr w ręcznym pisaniu.

