

Sprawozdanie 1 - Wprowadzenie do Sztucznej Inteligencji

Michał Kallas

26 marca 2025

1 Zadanie 1

1.1 Polecenie

Korzystając z jednej z dostępnych bibliotek (np. Keras/TensorFlow, PyTorch, czy scikit-learn) stwórz i wytrenuj sieć neuronową rozpoznającą cyfry z podanego zbioru danych. Jaki współczynnik prawidłowej rozpoznawalności ma wyuczona sieć na zbiorze testowym? Jaka jest czułość i precyzja?

1.2 Architektura sieci

Sieć neuronowa została zaprojektowana jako konwolucyjna sieć neuronowa (CNN), ponieważ tego typu architektury są szczególnie skuteczne w zadaniach przetwarzania obrazów. CNN potrafią automatycznie ekstraktować istotne cechy obrazu, takie jak krawędzie czy wzory, co czyni je znacznie lepszymi od klasycznych modeli klasyfikacyjnych.

Sieć zawiera następujące warstwy:

- Warstwa konwolucyjna (32 filtry, rozmiar 3x3, ReLU)
- Warstwa max pooling (2x2)
- Warstwa konwolucyjna (64 filtry, rozmiar 3x3, ReLU)
- Warstwa max pooling (2x2)
- Warstwa w pełni połączona (128 neuronów, ReLU)
- Warstwa Dropout (50%)
- Warstwa wyjściowa (10 neuronów, Softmax)

Model został skompilowany z funkcją straty categorical crossentropy i optymalizatorem Adam. W implementacji została wykorzystana biblioteka **TensorFlow**.

Struktura sieci została zaprojektowana w taki sposób, aby stopniowo zmniejszać wymiar obrazu i jednocześnie zwiększać poziom abstrakcji przetwarzanych cech. Warstwy konwolucyjne pozwalają na automatyczne wykrywanie istotnych wzorców, a warstwy pooling redukują ilość przetwarzanych danych, co przyspiesza działanie modelu. Warstwa w pełni połączona łączy wyodrębnione cechy w jedną reprezentację, a funkcja aktywacji softmax na końcu umożliwia klasyfikację do jednej z 10 kategorii. Warstwa Dropout w trakcie treningu losowo wyłącza 50% neuronów, aby zapobiec przeuczeniu modelu (żeby nie nauczył się przypadkowych szczegółów zamiast ogólnych reguł).

1.3 Wyniki testowe

Po przeprowadzeniu treningu na zbiorze MNIST (5 epok, batch size 128), osiągnięto następujące wyniki:

- Dokładność: **98.93%**
- Precyzja: **98.94%**
- Czulość: **98.91%**

Cyfra	Precyzja	Czulość	F1-score	Liczba egzemplarzy
0	0.9819	0.9980	0.9899	980
1	0.9930	0.9965	0.9947	1135
2	0.9865	0.9884	0.9874	1032
3	0.9872	0.9901	0.9886	1010
4	0.9969	0.9888	0.9928	982
5	0.9955	0.9843	0.9899	892
6	0.9989	0.9802	0.9895	958
7	0.9798	0.9893	0.9845	1028
8	0.9877	0.9877	0.9877	974
9	0.9871	0.9881	0.9876	1009
Średnia	0.9894	0.9891	0.9893	10000

Tabela 1: Szczegółowe wyniki klasyfikacji na zbiorze MNIST.

Model bardzo dobrze radzi sobie z klasyfikowaniem cyfr w zbiorze MNIST. Wysoka dokładność oraz zbliżone wartości precyzji i czulości wskazują na dobrą równowagę między wykrywaniem prawidłowych cyfr a minimalizowaniem fałszywych alarmów.

Analizując szczegółowe wyniki, można zauważyć, że:

- Najlepiej rozpoznawane cyfry to 4, 5 i 6, co sugeruje, że ich kształty są wyraźnie odróżnialne przez model.

- Cyfra 7 ma najniższą precyzję (97.98%), co może sugerować, że jest mylona z innymi cyframi, np. 1 lub 2.

2 Zadanie 2

2.1 Polecenie

Na podstawie własnych próbek pisma, stwórz swój własny zbiór testowy (co najmniej po trzy egzemplarze każdej cyfry; zachowaj także do obejrzenia obrazy swoich próbek). Sprawdź jak sieć stworzona w poprzednim punkcie reaguje na ten zbiór. Opisz krótko wnioski (współczynnik rozpoznawalności, błędy i ich prawdopodobna przyczyna, itp.)

2.2 Opis zbioru własnych próbek

Zbiór testowy zawierał trzy egzemplarze każdej cyfry napisanej odręcznie. Obrazy były wstępnie przetwarzane, aby dopasować je do formatu MNIST (rozmiar 28x28, normalizacja do wartości $[0,1]$).



Rysunek 1: Pierwszy zestaw odręcznie zapisanych cyfr.

2.3 Wyniki testowe na własnym zbiorze

Sieć osiągnęła dokładność na poziomie **83.33%**. Szczegółowe wyniki dla poszczególnych cyfr przedstawiono poniżej:

- **Cyfry rozpoznane poprawnie w 100%:** 0, 1, 2, 3, 4, 5, 8.
- **Cyfry z błędami:**
 - 6: *dwa razy pomyłona z 8*
 - 7: *raz pomyłona z 1*
 - 9: *dwa razy pomyłona z 4*

2.4 Analiza błędów

Błędy w rozpoznawaniu mogą wynikać z:

- Słabego odróżniania podobnych cyfr (np. 4 i 9, 6 i 8).
- Indywidualnych cech pisma ręcznego użytkownika.

- Braku zróżnicowania treningowego zbioru danych.
- Faktu, że EMNIST, to amerykański zbiór, a Amerykanie zapisują cyfry w trochę inny sposób niż Polacy.

3 Zadanie 3

3.1 Polecenie

Korzystając z jednej z dostępnych bibliotek stwórz i wytrenuj klasyfikator oparty o *Random Forest* rozpoznającą cyfry z podanego zbioru danych. Jaki współczynnik prawidłowej rozpoznawalności ma ten klasyfikator na zbiorze testowym? Jaka jest czułość i precyzja?

3.2 Opis modelu

Do klasyfikacji cyfr wykorzystano model Random Forest dostępny w bibliotece `scikit-learn`. Algorytm Random Forest, to jeden z najpopularniejszych klasyfikatorów opartych na drzewach decyzyjnych. Random Forest buduje wiele drzew decyzyjnych na różnych próbkach treningowych i używa większościowego głosowania do klasyfikacji. Dane wejściowe zostały spłaszczone do postaci wektora o wymiarze 784 (28x28 pikseli), co jest standardową metodą przygotowania danych do klasyfikacji z użyciem algorytmu Random Forest.

3.3 Wyniki testowe

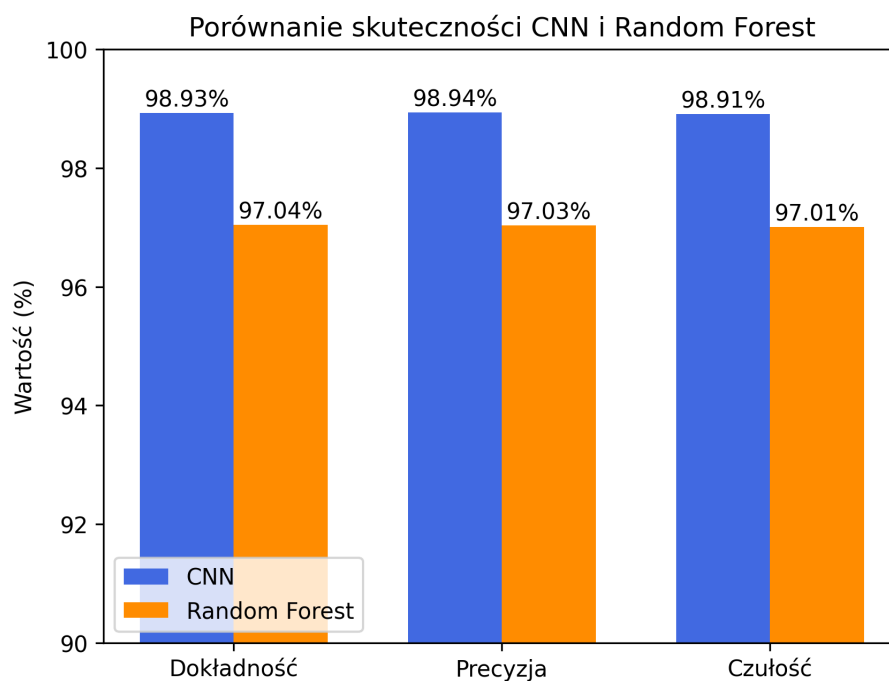
Klasyfikator został przetestowany na zbiorze MNIST, osiągając następujące wyniki:

- Dokładność: **97.04%**
- Precyzja: **97.03%**
- Czułość: **97.01%**

Cyfra	Precyzja	Czułość	F1-score	Liczba egzemplarzy
0	0.9681	0.9908	0.9793	980
1	0.9877	0.9930	0.9903	1135
2	0.9616	0.9709	0.9662	1032
3	0.9633	0.9624	0.9629	1010
4	0.9745	0.9725	0.9735	982
5	0.9762	0.9641	0.9701	892
6	0.9760	0.9781	0.9771	958
7	0.9715	0.9621	0.9668	1028
8	0.9617	0.9548	0.9583	974
9	0.9620	0.9524	0.9572	1009
Średnia	0.9703	0.9701	0.9702	10000

Tabela 2: Szczegółowe wyniki klasyfikacji dla Random Forest na zbiorze MNIST.

3.4 Porównanie wyników



Rysunek 2: Porównanie skuteczności CNN i Random Forest.

W porównaniu do sieci neuronowej, klasyfikator Random Forest osiągnął nieco gorsze wyniki. Może to wynikać z faktu, że modele oparte na drzewach decy-

zyjnych nie są tak skuteczne w analizie obrazów jak głębokie sieci neuronowe, które lepiej radzą sobie z wykrywaniem złożonych wzorców wizualnych. Niemniej jednak, Random Forest pozostaje efektywną alternatywą w przypadku ograniczonych zasobów obliczeniowych, ponieważ nie wymaga tak dużej ilości danych treningowych ani intensywnego przetwarzania jak głębokie sieci neuronowe.

4 Podsumowanie

Sieć neuronowa wykazała bardzo wysoką skuteczność na zbiorze MNIST, jednak jej skuteczność na zbiorze własnych próbek była zauważalnie niższa. Klasyfikator Random Forest osiągnął nieco gorsze wyniki na MNIST, ale nadal prezentował wysoką skuteczność. Aby poprawić dokładność na niestandardowych próbkach, można by:

- **Powiększenie zbioru danych treningowych:** Zwiększenie liczby próbek w zbiorze treningowym, szczególnie poprzez dodanie bardziej zróżnicowanych przykładów pisma odręcznego (np. od różnych osób, w różnych stylach), mogłoby pomóc modelowi lepiej generalizować.
- **Zastosowanie metod augmentacji danych:** Techniki takie jak obrót, skalowanie, przesunięcie czy dodanie szumu do obrazów mogą sztucznie zwiększyć różnorodność zbioru treningowego. Na przykład, obrót o kilka stopni lub zmiana grubości linii w cyfrach mogłaby lepiej przygotować model na różnice w indywidualnym piśmie.
- **Dostosowanie architektury modelu:** W przypadku sieci neuronowej można rozważyć dodanie większej liczby warstw konwolucyjnych lub zwiększenie liczby filtrów, aby model mógł wychwycić bardziej subtelne cechy pisma. Dla Random Forest można by zwiększyć liczbę drzew decyzyjnych lub dostosować parametry, takie jak maksymalna głębokość drzew, aby lepiej radzić sobie z różnorodnością danych.