

Projekt z NAI: Rozpoznawanie cyfr

S. Hoa Nguyen

1 Rozpoznawanie cyfr - 40 punktów (max 2 osoby)

Cel badania: Zaprojektować i przeprowadzić uczenie **dwuwarstwowej sieci neuronowej** do rozpoznawania 10 cyfr od 0 do 9. Cyfry są reprezentowane przez mapy bitowe o wymiarze 6×5 . Przykładowe wzorce dla cyfry 0 są niżej podane.

1	1	1	1	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	1	1	1	1

0	1	1	1	0
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
0	1	1	1	0

0	1	1	1	0
1	1	0	1	1
1	0	0	0	1
1	0	0	0	1
1	1	0	1	1
0	1	1	1	0

Zadania 1. Tworzenie zbioru treningowego i testowego

- *Zbiór treningowy:* Dla każdej z cyfr od 0 do 9, utwórz 3 do 5 różnych przykładów (wzorców). Razem jest **od 30 do 50 przykładów treningowych**.
- *Zbiór testowy:* Dla każdej z cyfr od 0 do 9, utwórz 2 inne niż w zbiorze treningowym przykłady (wzorce). Razem jest **20 przykładów testowych**.
- *Wektoryzacja:* Konwertować mapy 6 na 5 na wektory binarne o długości 30. Przypisać klasę (od 0 do 9) wektorom treningowym.

Zadania 2. Definiowanie sieci neuronowej

- Ustalić architekturę sieci: *liczba wejść, liczba wyjść, liczba neuronów w każdej warstwie*.
- Zdefiniować funkcję aktywacji: *unipolarna sigmoidalna* lub *bipolarna sigmoidalna*.

Zadania 3. Trenowanie sieci

- Wykorzystać *algorytm wstecznej propagacji błędów* do uczenia sieci.
Można skorzystać z gotowej biblioteki ML, np. **Python Scikit-learn, Keras, R, Weka**.
- Ustawić parametry sieci, między innymi: *liczba ukrytych neuronów, liczba epok uczenia i współczynnik uczenia*. Przeprowadzić uczenia.
- Po każdej epoce uczenia (iteracji) obliczyć *błąd sieci*.
- Sporządzić wykres błęd vs. liczby epok uczenia. Obserwować, czy błąd maleje do zera?

Zadania 4. Testowanie sieci

- Sprawdzić czy sieć rozpoznaje wszystkie cyfry treningowe?
- Obliczyć, ile przykładów w zbiorze testowym zostało dobrze sklasyfikowanych?

Zadania 5. Raportowanie wyników

- Opisać parametry sieci (z odpowiednim rysunkiem ilustrującym): architektura i funkcja aktywacji.
- Opisać schemat algorytmu uczenia (alg. wsteczna propagacja błędów) i narzędzie programistyczne.
- Umieścić wykres zależności między liczbą epok uczenia a błędem sieci.
- Napisać wyniki eksperymentalne: liczba przykładów **dobrze sklasyfikowanych** w zbiorze uczącym i testowym.

Wskazówka:

- Przekształcić mapy bitowe na wektory binarne o długości $6 * 5 = 30$. One są wektorami treningowymi (wejściowymi) dla sieci.
- Liczba wejść do sieci jest tyle samo co długość wektora treningowego.
- Oprócz standardowych wejść, neurony mają jedno wejście stałe (na wartość progową).
- Liczba wyjść (neuronów wyjściowych) jest tyle samo co liczba klas (10).
- Wagi początkowe i wartości progowe są wylosowane z przedziału $(-1,1)$.