

## NAI – mały projekt programistyczny 3

### Uczenie jednowarstwowej sieci neuronowej

Napisz program, który przeprowadza uczenie jednowarstwowej sieci neuronowej na podstawie danych w formacie CSV. Sieć powinna:

- mieć architekturę **lokalną**
- używać bipolarnej obciętej liniowej funkcji aktywacji  
(tzn.  $f(x) = x$  dla  $x \in [-1, 1]$  oraz  $\text{sgn}(x)$  wpp.)
- wybierać kategorię na podstawie maksymalnej aktywacji neuronu.

Program powinien pobierać następujące argumenty:

- `train-data`: plik z danymi treningowymi w formacie CSV, w którym ostatnia kolumna jest typu tekstowego i opisuje kategorię przypisaną do danego przykładu
- `test-data`: plik z danymi testowymi w formacie takim samym, jak `train-data`
- `alpha`: wartość stałej uczenia
- `k`: ilość “iteracji” uczenia (1 “iteracja” = po 1 kroku dla każdego przykładu treningowego)

Uwaga: Pliki `train-data` i `test-data` mogą mieć dowolną liczbę kolumn – program powinien radzić sobie z różnymi sytuacjami.

Proszę przetestować swój program na plikach `train.csv` i `test.csv`, zamieszczonych w MS Teams (Files > MPP 3 > jezyki\_wikipedia). Pliki te zawierają rozkłady częstości występowania liter w próbkach różnych języków naturalnych (pl / en / fr / it / es / pt).

Próbki zostały pobrane z Wikipedii i oczyszczone (usunięcie znaków diakrytycznych, przestankowych, odstępów i innych nietypowych znaków, konwersja do małych liter). Każdy rekord w pliku CSV opisuje próbkę 500 znaków; wartości `freq_a`, ..., `freq_z` oznaczają częstotliwości występowania odpowiednich znaków w danej próbce; ich suma wynosi zawsze 1.

Program powinien:

1. Wykonać uczenie sieci ( $k$  iteracji) na przykładach z części treningowej
2. Przy pomocy wyuczonej sieci przypisać kategorie do przykładów z części testowej
3. W odniesieniu do testu z punktu 2, wypisać na ekran skuteczność oraz *macierz omyłek* (proszę przeczytać jej definicję w wykładzie 4 na Gakko, slajd nr 16)
4. Wyświetlić ostatecznie wyuczone wagi (oraz progi) perceptronów w sieci

5. [opcjonalnie – za dodatkowy punkt z aktywności] po procesie uczenia pobrać z ekranu fragment tekstu w języku naturalnym (można założyć brak znaków diakrytycznych, interpunkcji itp.) i sklasyfikować otrzymany fragment pod względem języka.

Rady techniczne:

1. Do uczenia perceptronów wykorzystujemy regułę delta.
2. Proszę zapewnić stałą normę wektora wag.
3. Proszę poeksperymentować z normą z punktu 1. (Ponieważ nasze przykładowe dane zawierają bardzo małe liczby, to używając znormalizowanych wag otrzymamy iloczyny skalarne bardzo bliskie 0. To z kolei prowadzi do “rozbujaania” uczenia w sytuacji, gdy za wartości pożądane bierzemy 1 bądź -1. To zjawisko pogarsza wynik uczenia, ale można je zniwelować pracując z wektorami wag o większej normie).
4. Aby uzyskać dobre wyniki, warto cierpliwie poszukać dobrej kombinacji  $\alpha$  oraz  $k$ .

Uwaga: Program należy wykonać samodzielnie. **Plagiat** lub **niezrozumienie** rozwiązania skutkuje **brakiem zaliczenia projektu**.

Nie można korzystać z gotowych bibliotek do uczenia maszynowego ani operacji na wektorach. Wszystkie szczegóły algorytmu należy samodzielnie przećwiczyć kodując.

Termin: **17 kwietnia**