

Jakub Dębiński 251501
Łukasz Murza 251592

Wtorek 10:30
14.05.2025

Tytuł: Sztuczna inteligencja i systemy ekspertowe
Podtytuł: Zadanie 2: MLP

Cel

Celem tego raportu jest eksperymentalna weryfikacja uniwersalnej sieci neuronowej MLP, która pozwala na dowolne skalowanie liczby warstw i neuronów oraz opcjonalne uwzględnianie wejścia obciążającego (biasu). Sieć będzie uczyć się metodą propagacji wstecznej błędu w trybie on-line z możliwością użycia członu momentum, a neurony przetwarzające będą wykorzystywać sigmoidalną funkcję aktywacji o współczynniku nachylenia równym 1. Praca będzie również obejmować eksperymenty nad wpływem parametrów uczenia na efektywność działania perceptronu.

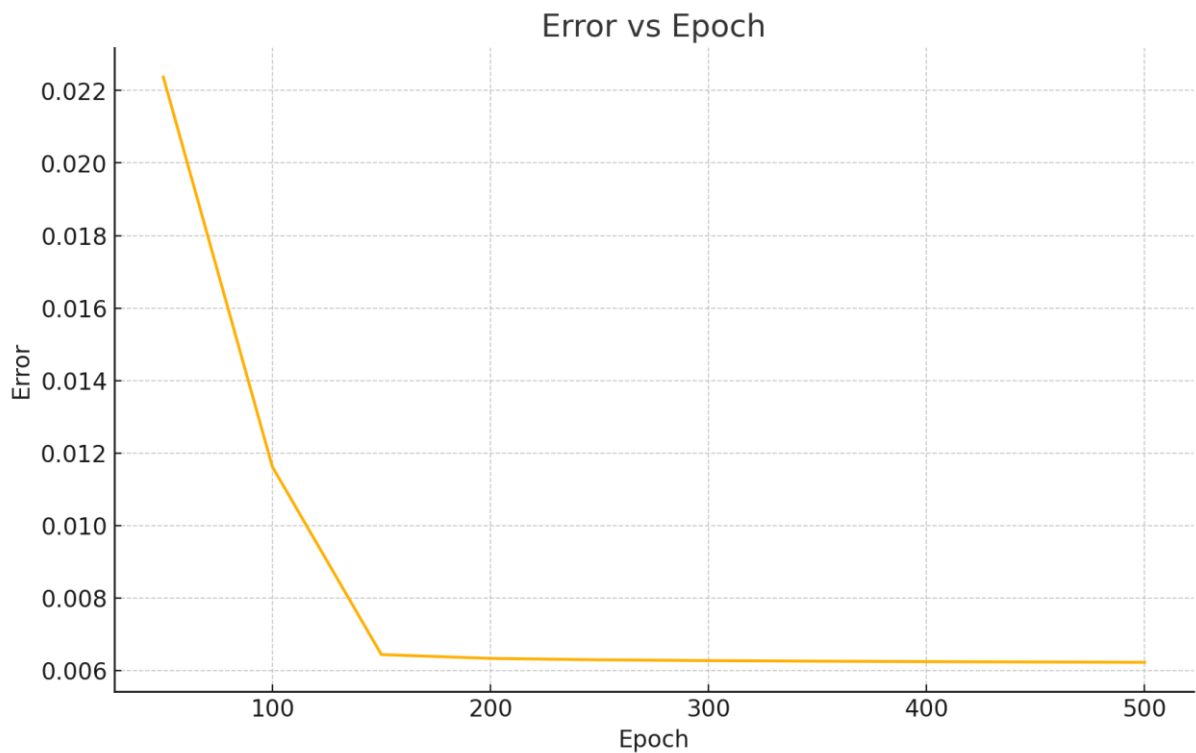
Wyniki

Narzędzie, z jakiego korzystaliśmy do obliczeń oraz do generowania wyników to język interpretowany Python.

Klasyfikacja zbioru Irysów

Przyjęte parametry sieci:

- Liczba wejść: 4
- Liczba wyjść: 3
- Warstwy ukryte: 5 i 4 neuronowe
- Użyto bias: Tak
- Szybkość uczenia: 0.1
- Momentum: 0.9
- Liczba epok: 500
- Błąd docelowy: 0.001
- Częstość logowania: 50



Wynik testu: 1.0000 (100%)

Macierz pomyłek

	Predykcja 0	Predykcja 1	Predykcja 2
Rzeczywista 0	10	0	0
Rzeczywista 1	0	17	0
Rzeczywista 2	0	0	18

Metryki dla poszczególnych klas

Klasa	Precision	Recall	F1
0	1.00	1.00	1.00
1	1.00	1.00	1.00
2	1.00	1.00	1.00

Wnioski:

Dzięki odpowiednio dobranej architekturze i parametrom sieć osiągnęła 100 % skuteczności na zbiorze testowym, przy czym błąd gwałtownie spadał w pierwszych ~200 epokach, a później stabilizował się na ok. 0,006 co sugeruje wprowadzenie early stopping.

Autoasocjacja (sieć typu autoenkoder)

Wyływ biasu na enkoder

Przyjęte parametry sieci:

- Liczba wejść: 4
- Liczba wyjść: 4
- Warstwy ukryte: Jedna 2-neuronowa
- Szybkość uczenia: 0.6
- Momentum: 0.0
- Liczba epok: 1000
- Błąd docelowy: $1e-4$
- Częstość logowania: 100

1. Bias = włączony

$[1,0,0,0] \rightarrow [0.9424, 0.0199]$
 $[0,1,0,0] \rightarrow [0.0132, 0.0377]$
 $[0,0,1,0] \rightarrow [0.9740, 0.9624]$
 $[0,0,0,1] \rightarrow [0.0262, 0.9815]$

Błąd po 1000 epokach wynosi 0.00414. Cztery wyraźnie różne wektory, każda para wzorców dostaje odrębny kod, autoenkoder się nauczył

2. Bias = wyłączony

$[1,0,0,0] \rightarrow [0.2345, 0.2433]$
 $[0,1,0,0] \rightarrow [0.2337, 0.2425]$
 $[0,0,1,0] \rightarrow [0.9855, 0.0608]$
 $[0,0,0,1] \rightarrow [0.0297, 0.9855]$

Błąd po 1000 epokach wynosi 0.08369. Dwa pierwsze wzorce mają prawie identyczny kod ($\sim[0.23, 0.24]$), sieć nie rozróżnia ich i nie potrafi ich poprawnie zrekonstruować.

Szybkość uczenia przy różnych η i μ

Przyjęte parametry sieci:

- Liczba wejść: 4
- Liczba wyjść: 4
- Warstwy ukryte: Jedna 2-neuronowa
- Liczba epok: 2000
- Błąd docelowy: $1e-4$
- Częstość logowania: 500

(η, μ)	błąd ep. 500	błąd ep. 1000	błąd ep. 1500	błąd ep. 2000
(0.9, 0.0)	0.01308	0.00323	0.00171	0.00113
(0.6, 0.0)	0.02475	0.00424	0.00230	0.00158
(0.2, 0.0)	0.08952	0.02176	0.00993	0.00624
(0.9, 0.6)	0.00192	0.00076	0.00047	0.00033
(0.2, 0.9)	0.00182	0.00081	0.00052	0.00038

η – współczynnik nauki

μ – współczynnik momentum

Wyższe η przy zerowym μ przyspiesza naukę ($0.9 > 0.6 > 0.2$).

Momentum ($\mu=0.6$ lub 0.9) znacząco przyspiesza, zwłaszcza przy niższych η :

- Dla ($\eta=0.9, \mu=0.6$) błąd już po 500 epokach jest niższy niż w przypadku bez momentum po 2000 epokach.
- Nawet przy $\eta=0.2$ dodanie dużego $\mu=0.9$ pozwala zbliżyć się do tego, co uzyskiwała sama duża eta.

Wnioski:

1. *Bez biasu autoenkoder nie ma wystarczającej swobody przesunięcia progu aktywacji, przez co dwa wzorce wpadają w tę samą strefę aktywacji i nie da się ich oddzielić. Tylko wersja z bias rozpoznaje wszystkie wzorce.*
2. *Optymalne wartości to $\eta=0.9$ i $\mu\approx 0.6-0.9$. To właśnie one dają najszybsze zbieganie, co oznacza najszybsze uczenie. Zbyt mała wartość η (0.2) znacząco spowalnia tempo uczenia się. Natomiast brak momentum obniża tempo uczenia, nawet przy dużym η . Momentum < 1 pomaga, lecz $\mu \geq 1$ prowadzi do oscylacji lub rozbierzości. Kolejna zmiana wag może być większa niż sam gradient co „przeskakuje” minimum zamiast się do niego zbliżać.*