

**Zadanie nr 2 - uczenie i testowanie sieci
MADALINE do rozpoznawania znaków**
Inteligentna Analiza Danych

Karol Kazusek - 254189, Sebastian Zych - 254264

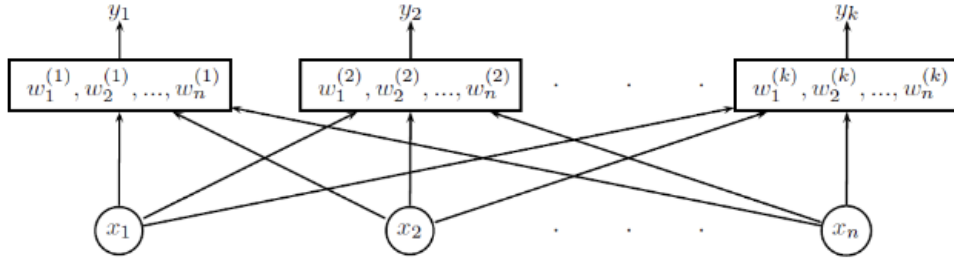
21.10.2024

1 Cel zadania

Celem zadania drugiego było zaimplementowanie sieci MADALINE do rozpoznawania znaków.

2 Wstęp teoretyczny

Sieć MADALINE (ang. Multiple ADaptive LINear Elements) to sieć zbudowana z pojedynczej warstwy neuronów liniowych.



Rysunek 1: Architektura sieci MADALINE

Sieć MADALINE składa się z jednej warstwy neuronów liniowych o liczbie k . Każdy neuron w tej warstwie posiada swój własny wektor wag:

$$\mathbf{w}^{(l)} = [w_1^{(l)}, w_2^{(l)}, \dots, w_n^{(l)}] \quad (1)$$

gdzie indeks l oznacza numer neuronu (od 1 do k). Wszystkie neurony w sieci MADALINE otrzymują ten sam wektor wejściowy:

$$\mathbf{x} = [x_1, x_2, \dots, x_n] \quad (2)$$

i obliczają swoje wyjście $y^{(l)}$ jako iloczyn skalarny wektora wag $\mathbf{w}^{(l)}$ oraz wektora wejściowego \mathbf{x} :

$$y^{(l)} = \mathbf{w}^{(l)} \cdot \mathbf{x} \quad (3)$$

Każdy neuron w sieci oblicza podobieństwo między swoim wektorem wag a wspólnym wektorem wejściowym. Warto zaznaczyć, że aby wyniki były poprawne, wszystkie wektory wag $\mathbf{w}^{(l)}$ neuronów muszą być znormalizowane do długości równej jeden, tak jak wektor wejściowy.

Aby poprawnie obliczyć iloczyn skalarny (miarę podobieństwa cosinusowego) między wektorem wejściowym a wektorem wagowym każdego neuronu, konieczne jest wcześniejsze

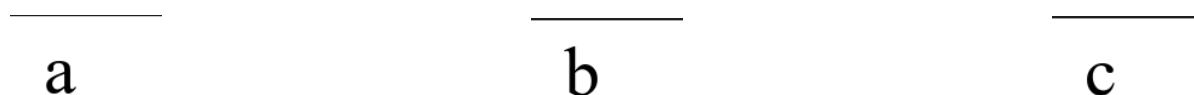
znormalizowanie wektora wejściowego. Normalizacja ta polega na przeskalowaniu wektora przez odwrotność jego długości, tak aby jego długość wynosiła 1:

$$\mathbf{x}' = \frac{\mathbf{x}}{\|\mathbf{x}\|}$$

Proces ten pozwala na zachowanie proporcji między współrzędnymi wektora, jednocześnie zapewniając, że długość każdego wektora wejściowego jest stała. Dzięki temu możliwe jest dokładne porównanie każdego znormalizowanego wektora wejściowego, utworzonego na podstawie danych z pliku graficznego (np. obrazu znaku zapisanego w formacie .png), z wektorami wagowymi neuronów w sieci. Iloczyn skalarny, który mierzy podobieństwo cosinusowe, umożliwia rozpoznanie, który z neuronów najlepiej dopasowuje się do zadanego wzorca i w konsekwencji rozpoznanie, jaki znak (np. litera) został przedstawiony.

3 Eksperymenty i wyniki

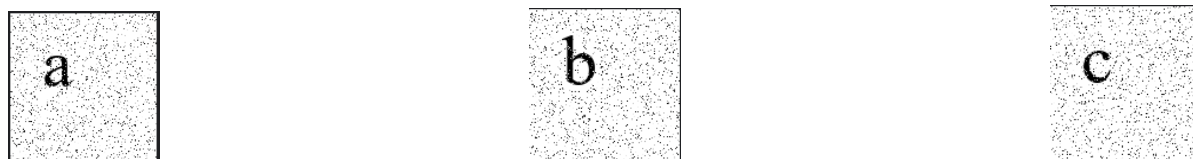
W tym zadaniu wykonano pięć eksperymentów: dla szumu 10%, 30%, 50%, 90% oraz szumu 0%. Każdy z neuronów ma na celu rozpoznawanie daną literę, zatem do trenowania sieci i dostosowania odpowiednich wag dla każdego z neuronów użyto liter bez szumu:



Rysunek 2: Litery 'a', 'b' i 'c' bez szumu

3.1 Eksperyment nr 1

Pierwszy eksperyment polegał na trenowaniu liter 'a', 'b' oraz 'c' o zaszumieniu 10% oraz ich predykcji.



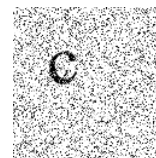
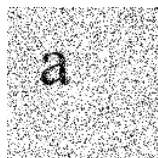
Rysunek 3: Litery 'a', 'b' i 'c' z szumem 10%

Tabela 1: Wyniki rozpoznawania liter z szumem 10%

Wejście	Wyjście	Confidence
Zaszumione a 10%	'a'	0.949
Zaszumione a 10%	'b'	0.935
Zaszumione a 10%	'c'	0.941
Zaszumione b 10%	'a'	0.936
Zaszumione b 10%	'b'	0.949
Zaszumione b 10%	'c'	0.940
Zaszumione c 10%	'a'	0.941
Zaszumione c 10%	'b'	0.940
Zaszumione c 10%	'c'	0.949

3.2 Eksperyment nr 2

Pierwszy eksperyment polegał na trenowaniu liter 'a', 'b' oraz 'c' o zaszumieniu 30% oraz ich predykcji.



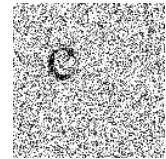
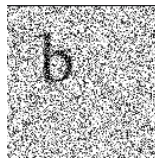
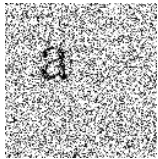
Rysunek 4: Litery 'a', 'b' i 'c' z szumem 30%

Tabela 2: Wyniki rozpoznawania liter z szumem 30%

Wejście	Wyjście	Confidence
Zaszumione a 30%	'a'	0.837
Zaszumione a 30%	'b'	0.825
Zaszumione a 30%	'c'	0.831
Zaszumione b 30%	'a'	0.825
Zaszumione b 30%	'b'	0.837
Zaszumione b 30%	'c'	0.828
Zaszumione c 30%	'a'	0.830
Zaszumione c 30%	'b'	0.828
Zaszumione c 30%	'c'	0.837

3.3 Eksperyment nr 3

Pierwszy eksperyment polegał na trenowaniu liter 'a', 'b' oraz 'c' o zaszumieniu 50% oraz ich predykcji.



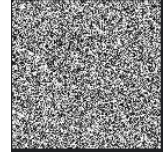
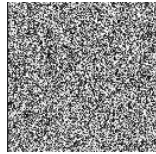
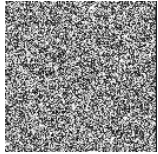
Rysunek 5: Litery 'a', 'b' i 'c' z szumem 50%

Tabela 3: Wyniki rozpoznawania liter z szumem 30%

Wejście	Wyjście	Confidence
Zaszumione a 50%	'a'	0.706
Zaszumione a 50%	'b'	0.697
Zaszumione a 50%	'c'	0.701
Zaszumione b 50%	'a'	0.705
Zaszumione b 50%	'b'	0.714
Zaszumione b 50%	'c'	0.708
Zaszumione c 50%	'a'	0.703
Zaszumione c 50%	'b'	0.702
Zaszumione c 50%	'c'	0.708

3.4 Eksperyment nr 4

Pierwszy eksperyment polegał na trenowaniu liter 'a', 'b' oraz 'c' o zaszumieniu 90% oraz ich predykcji.



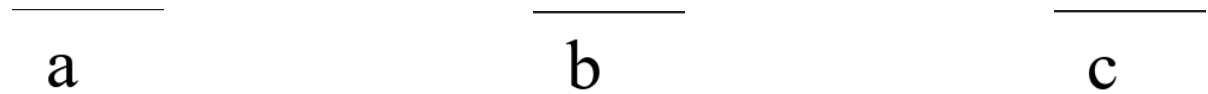
Rysunek 6: Litery 'a', 'b' i 'c' z szumem 90%

Tabela 4: Wyniki rozpoznawania liter z szumem 90%

Wejście	Wyjście	Confidence
Zaszumione a 90%	'a'	0.321
Zaszumione a 90%	'b'	0.315
Zaszumione a 90%	'c'	0.318
Zaszumione b 90%	'a'	0.319
Zaszumione b 90%	'b'	0.323
Zaszumione b 90%	'c'	0.321
Zaszumione c 90%	'a'	0.322
Zaszumione c 90%	'b'	0.319
Zaszumione c 90%	'c'	0.324

3.5 Eksperyment nr 5

Pierwszy eksperyment polegał na trenowaniu liter 'a', 'b' oraz 'c' o zaszumieniu 0% oraz ich predykcji.



Rysunek 7: Litery 'a', 'b' i 'c' z szumem 0%

Tabela 5: Wyniki rozpoznawania liter z szumem 0%

Wejście	Wyjście	Confidence
Zaszumione a 0%	'a'	1.000
Zaszumione a 0%	'b'	0.986
Zaszumione a 0%	'c'	0.992
Zaszumione b 0%	'a'	0.986
Zaszumione b 0%	'b'	1.000
Zaszumione b 0%	'c'	0.990
Zaszumione c 0%	'a'	0.992
Zaszumione c 0%	'b'	0.990
Zaszumione c 0%	'c'	1.000

4 Wnioski

Wyniki przeprowadzonych eksperymentów pozwalają wyciągnąć następujące wnioski dotyczące działania sieci MADALINE w zadaniu rozpoznawania liter przy różnych poziomach zaszumienia:

1. Skuteczność sieci przy braku szumu:

Dla liter bez szumu (0%) sieć MADALINE wykazała bardzo wysoką skuteczność. We wszystkich przypadkach uzyskano pewność (*confidence*) na poziomie bliskim 1, co wskazuje, że sieć dobrze wytrenowała się na niezaszumionych danych i była w stanie bezbłędnie rozpoznać litery 'a', 'b' i 'c'. Wartości *confidence* dla nieprawidłowych przewidywań również były relatywnie wysokie, jednak były wyraźnie niższe od wartości dla prawidłowych wyników.

2. Wpływ szumu na rozpoznawanie liter:

W miarę wzrostu poziomu szumu, sieć zaczęła mieć trudności z rozpoznawaniem liter. Największe trudności zauważono przy szumie na poziomie 90%, gdzie różnice w *confidence* dla różnych liter były stosunkowo małe. Wartości *confidence* dla prawidłowo rozpoznanych liter były bliskie wartościom dla nieprawidłowych przewidywań, co wskazuje na trudności sieci z wyraźnym odróżnieniem zaszumionych liter.

3. Symetria *confidence* dla różnych liter:

Wyniki pokazały, że *confidence* dla każdej litery w danym eksperymencie (np. litera 'a' przy 50% szumie) były stosunkowo symetryczne, co może sugerować, że litery 'a', 'b' i 'c' są w danych wejściowych bardzo podobne. Zmniejszenie różnic w *confidence* może być również efektem działania prostej sieci MADALINE, która ma ograniczone zdolności do modelowania bardziej złożonych zależności.

4. Zachowanie sieci dla wysokiego poziomu szumu:

W przypadku 90% szumu sieć nadal była w stanie zdołać rozpoznać prawidłowe litery, jednak pewność przewidywań była znacznie niższa, a różnice między prawidłowymi i nieprawidłowymi przewidywaniami były minimalne. Może to wskazywać na granice efektywności sieci MADALINE przy bardziej skomplikowanych zadaniach lub wyższych poziomach szumu.

5. Znaczenie normalizacji wektorów:

W trakcie eksperymentów potwierdzono, że normalizacja wektorów wagowych i wejściowych była kluczowa dla poprawnych wyników sieci. Dzięki normalizacji możliwe było uzyskanie wartości *confidence* odzwierciedlających miarę podobieństwa cosinusowego między wektorem wejściowym a wagami neuronów.

6. Ograniczenia sieci MADALINE:

Wyniki eksperymentów sugerują, że sieć MADALINE dobrze sprawdza się w zadaniach, gdzie wzorce są wyraźnie różne (np. dla niezaszumionych danych), jednak przy wyższych poziomach szumu lub bardziej subtelnych różnicach między literami może napotkać trudności z rozpoznawaniem. W takich przypadkach bardziej zaawansowane sieci, np. wielowarstwowe perceptrony, mogą być bardziej skuteczne.

Podsumowując, sieć MADALINE okazała się skuteczna w zadaniu rozpoznawania liter, jednak wraz ze wzrostem poziomu szumu jej efektywność znacząco spadała. Wyniki eksperymentów mogą stanowić podstawę do dalszych badań nad efektywnością prostych sieci neuronowych w trudniejszych warunkach, a także nad możliwością ich rozbudowy i poprawy w złożonych zadaniach.