

# Sprawozdanie z piątego laboratorium WSI

Michał Goławski, 325158

# 1. Opis algorytmu

Algorytm korzysta z sieci neuronowej trzywarstwowej. Na wejściu otrzymuje zdjęcie odręcznie pisanej cyfry w postaci 784 wektora pikseli przeskalowanych do wartości w zakresie  $[0, 1]$ . Algorytm do optymalizacji swoich wag i biasów wykorzystuje metodę gradientu prostego. Każda z warstw neuronów jest warstwą w pełni połączoną, każdy neuron jako funkcję aktywacji wykorzystuje funkcję tangensa. Funkcja straty która jest wykorzystywana do obliczenia  $\frac{\delta E}{\delta Y}$  w ostatniej warstwie jest funkcją błędu średniokwadratowego. Algorytm na wyjściu zwraca 10 elementowy wektor, którego składowe reprezentują prawdopodobieństwo wystąpienia danej pod indeksem wektora cyfry.

## 2. Opis eksperymentów

Algorytm użył do treningu bazy danych MNIST losując z niego 10000 obrazów. Dane z pikselami zostały zamienione z macierzy o rozmiarze  $28 \times 28$  na wektor o 784 składowych. Ponadto wartości pikseli zostały przeskalowane by były w zakresie liczbowym 0-1. W ten sposób algorytm mógł uzyskać lepsze efekty. Etykiety do zbioru treningowego musiały zostać podane algorytmowi w postaci 10 elementowych wektorów zawierających pojedynczą 1 i resztę wypełnioną zerami. Na przykład etykieta reprezentująca cyfrę 2 została zamieniona w wektor  $[0, 0, 1, 0, \dots, 0]$ . Do testowania użyto losowych 1000 obrazów z bazy danych MNIST. Parametry użyte do trenowania sieci neuronowej wynoszą 0.1 dla wielkości kroku  $\beta$  i 50 dla liczby epok (liczby wykonanych iteracji przez metodę gradientu prostego).

## 3. Wyniki eksperymentów

### 3.1. Skuteczność

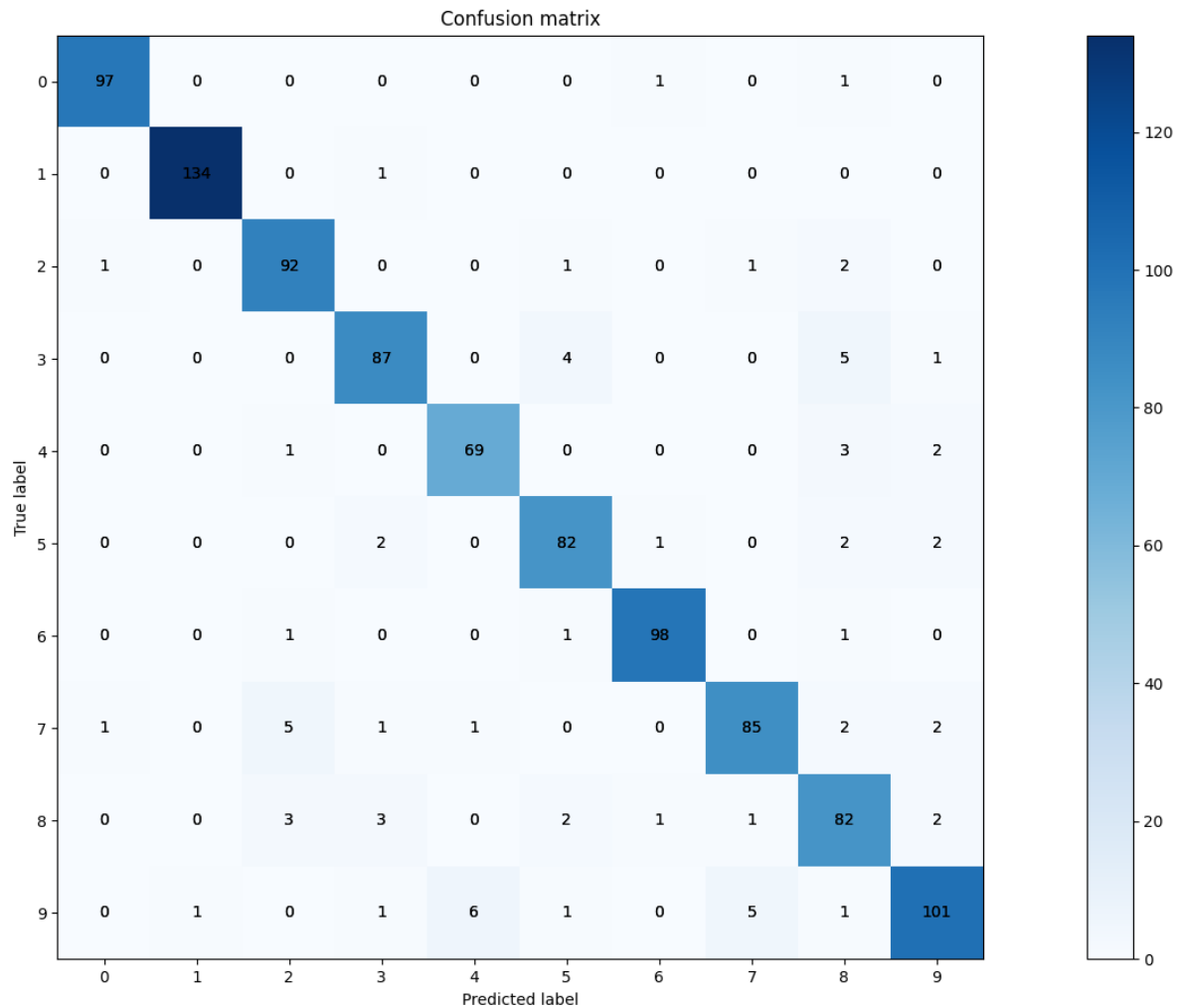
Skuteczność wynosiła 0.927 co w połączeniu z faktem, że klasy były rozłożone mniej więcej równomiernie pokazuje, że algorytm działa dobrze.

### 3.2. Raport klasyfikacji

Class	Precision	Recall	F1-score
0	0.98	0.98	0.98
1	0.99	0.99	0.99
2	0.90	0.95	0.92
3	0.92	0.90	0.91
4	0.91	0.92	0.91
5	0.90	0.92	0.91
6	0.97	0.97	0.97
7	0.92	0.88	0.90
8	0.83	0.87	0.85
9	0.92	0.87	0.89

Jak widzimy na tabeli powyżej wszystkie parametry dla każdej klasy są większe od 0.83 co oznacza, że algorytm klasyfikując elementy danej klasy rzadko się myli przydzielając do niej elementy z innej klasy (parametr precision). Ponadto algorytm klasyfikując elementy danej klasy rzadko przydziela je do innej klasy (parametr recall). Średnia harmoniczna z tych dwóch parametrów czyli f1-score pokazuje wyraźnie, że algorytm działa zadowalająco.

### 3.3. Confusion Matrix



Jak widać na confusion matrix algorytm poprawnie klasyfikuje obrazy liczb, zaciemniony obszar układu się w przekątną co oznacza, że obrazki zostały sklasyfikowane poprawnie w znacznej większości dla danej klasy.