

# Sieci Neuronowe projekt 2

Igor Kołakowski  
Monika Zielińska

17 listopada 2020

## 1 Tematyka projektu

Naszym zadaniem było zaimplementowanie sieci Art2 do zadania klastrowania danych. Skuteczność działania sieci przetestowaliśmy na trzech zbiorach danych.

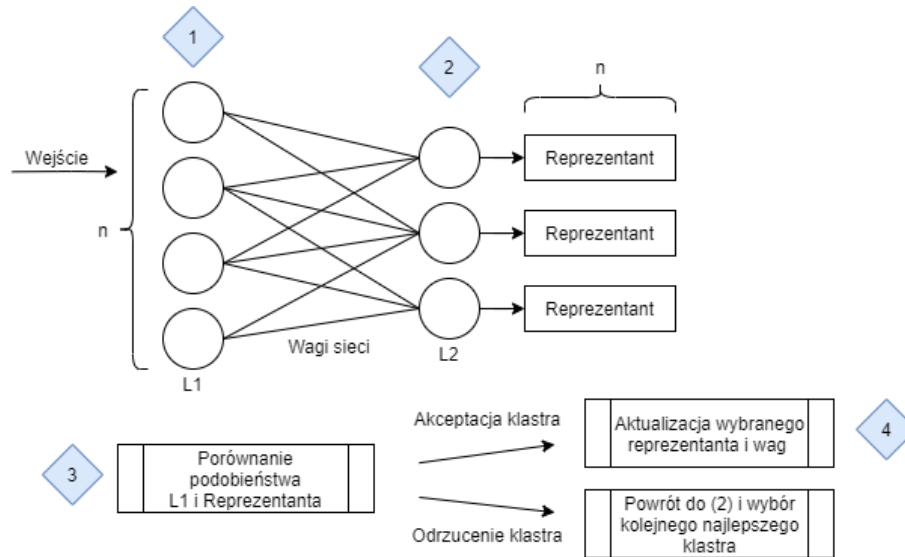
Problem klasyfikacji bez nadzoru polega na grupowaniu podobnych do siebie elementów do tych samych klas. Podobieństwo elementów zależy od przyjętego kryterium wyrażonego przy pomocy funkcji podobieństwa (metryki).

## 2 Opis sieci Art2

Pomysł sieci Art2 został przedstawiony po raz pierwszy przez Stephena Grossberga i Gail Carpenter [1]. Wykorzystana przez nas implementacja opiera się na wersji opracowanej w książce „Fundamentals of neural networks: architectures, algorithms and applications” [2].

Sieć Art2 składa się z czterech głównych mechanizmów:

1. Normalizacji wektora wejściowego w warstwie L1, do wektora jednostkowego
2. Przekształcenia liniowego warstwy L1 na warstwę L2
3. Wyboru klastra o największej wartości aktywacji i sprawdzenia podobieństwa między jego reprezentantem a warstwą L1
4. Przeprowadzenia aktualizacji wag sieci



Rysunek 1: Schematyczny diagram działania sieci Art2

Omówimy pokrótce każdy z tych elementów.

## 2.1 Warstwa L1

Wejście jest normalizowane do wektora jednostkowego.

## 2.2 Warstwa L2

Wartości tej warstwy otrzymujemy ze wzoru:

$$L_2 = L_1^T \cdot W$$

Gdzie  $W$  to macierz wag łączących warstwę L1 i L2.

## 2.3 Wybór wzorca podobnego

Po obliczeniu warstwy L2, szukamy indeksu  $J$  elementu warstwy o największej wartości.

Jeżeli wykorzystujemy sieć do klasyfikacji zwracamy  $J$  jako numer przypisanego klastra dla obserwacji wejściowej.

W przypadku trenowania sieci, sprawdzamy poziom zgodności między reprezentantem klastra  $J$ , a aktywacją warstwy L1. Jeżeli poziom jest zadowalający, następuje aktualizacja wag sieci między  $J$ -tym klastrem a warstwą L1 oraz aktualizacja reprezentanta klastra  $J$ .

## 2.4 Aktualizacja sieci

Wagi sieci wchodzące do J-tego klastra od każdego z neuronów warstwy L1 uaktualniamy według wzoru:

$$W_{iJ} = L_1 \alpha d + W_{iJ}(1 + \alpha d(d-1))$$

Reprezentant J uaktualniany jest w analogiczny sposób:

$$R_J = L_1 \alpha d + R_J(1 + \alpha d(d-1))$$

Gdzie  $\alpha$  oraz  $d$  są parametrami sieci odpowiedzialnymi za prędkość nauki.

## 3 Przeprowadzone eksperymenty

Zaprezentowana sieć została przetestowana z użyciem trzech zbiorów danych. Jej skuteczność była mierzona funkcją zdefiniowaną następująco. Rozważmy zbiór danych o liczności  $n$ , każdą z  $m$  klas danych wejściowych  $T$  i każdy z  $m$  klastrów  $P$ , znalezionych przez sieć. Jeżeli klastrów jest mniej, przyjmujemy, że liczność nadmiarowych wynosi zero. Oznaczmy przez  $C(t, p)$  licznosc zbioru rekordów, których prawdziwa klasa to  $t$ , zaś zostały przypisane do klastra  $p$ . Znajdujemy bijekcję  $f : T \rightarrow P$  maksymalizującą wartość  $S = \sum_{i=1 \dots m} C(i, f(i))$  i za skuteczność sieci uznajemy  $acc = S/n$ .

### 3.1 Proste dane testowe

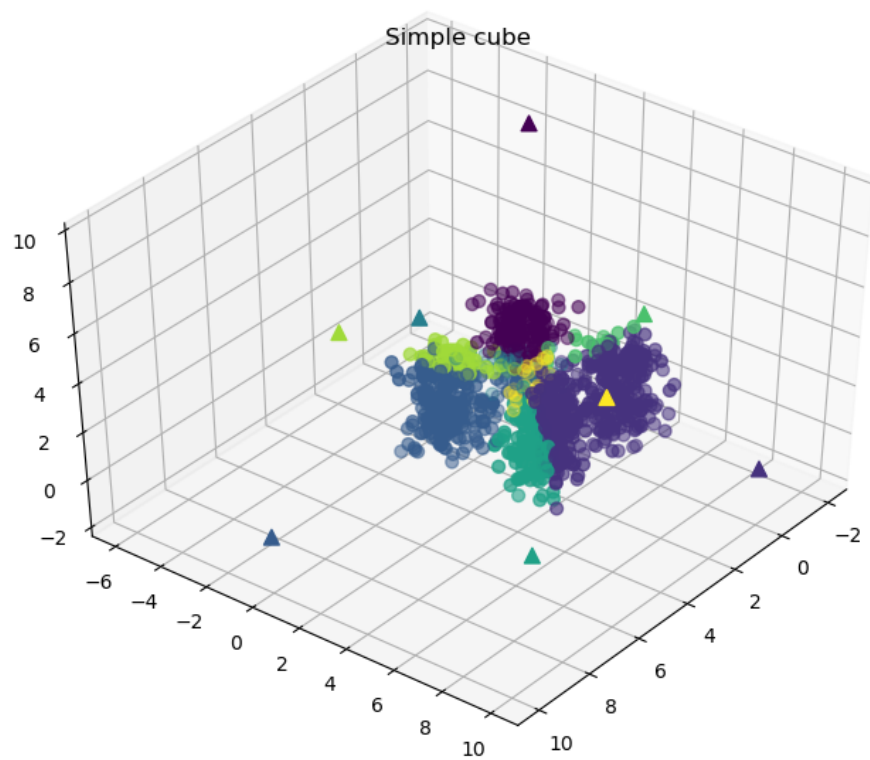
Dwa podzbiory danych reprezentujące sześciokąt w przestrzeni dwuwymiarowej oraz sześcian w przestrzeni trójwymiarowej zostały podzielone na zbiór treningowy rozmiaru 90% zestawu oraz testowy rozmiaru 10%.

Procentowe skuteczności otrzymane na zbiorach zostały przedstawione w tabeli nr 1.

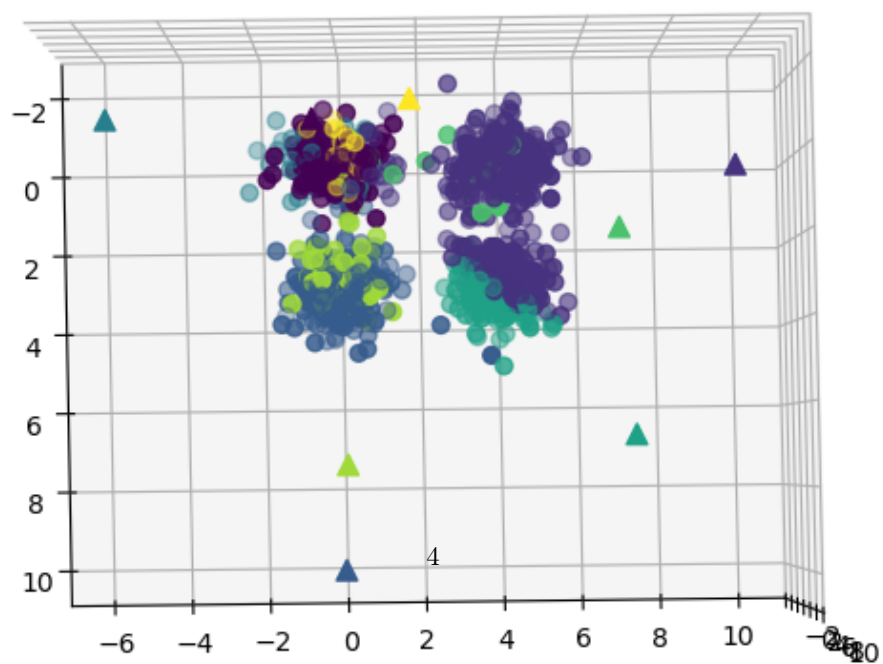
| zbiór | cube | cube-nonmatching | hexagon |
|-------|------|------------------|---------|
| train | 54%  | 47%              | 47%     |
| test  | 55%  | 52%              | 55%     |

Tablica 1: Podsumowanie procentowej skuteczności sieci w zadaniu klasteryzacji na zbiorach cube, cube-nonmatching i hexagon.

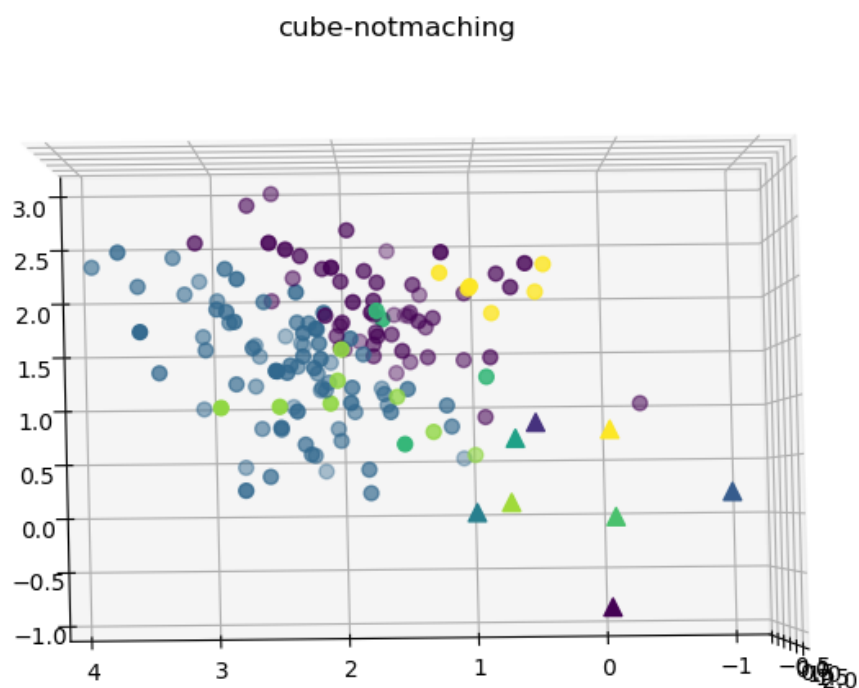
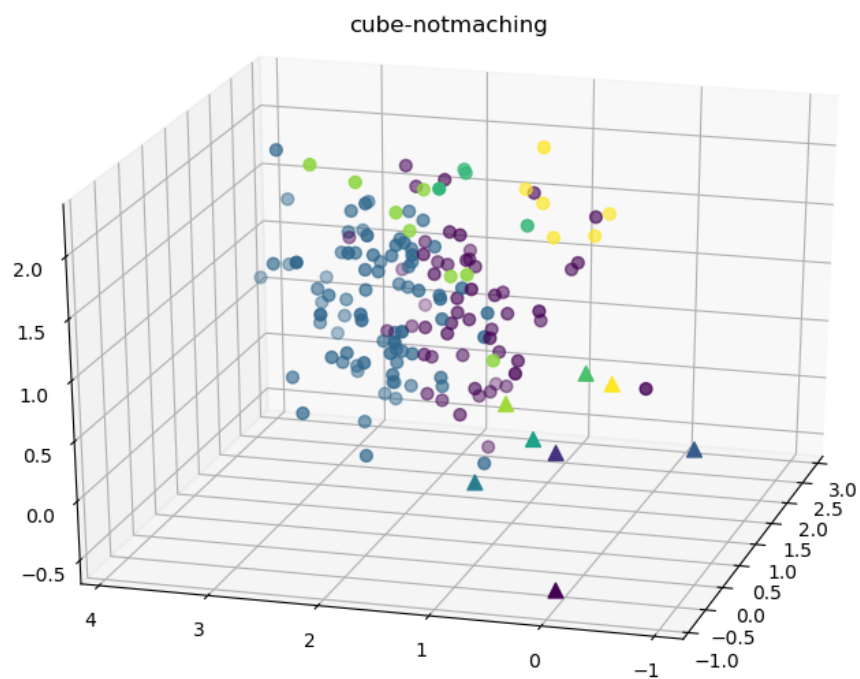
Na wykresie zbioru Hexagon, rysunek nr 4, widać charakterystykę działania sieci art2. Obserwacje obecne w pewnym stożku wektora będącego reprezentantem klastra, są przypisywane do tego klastra.



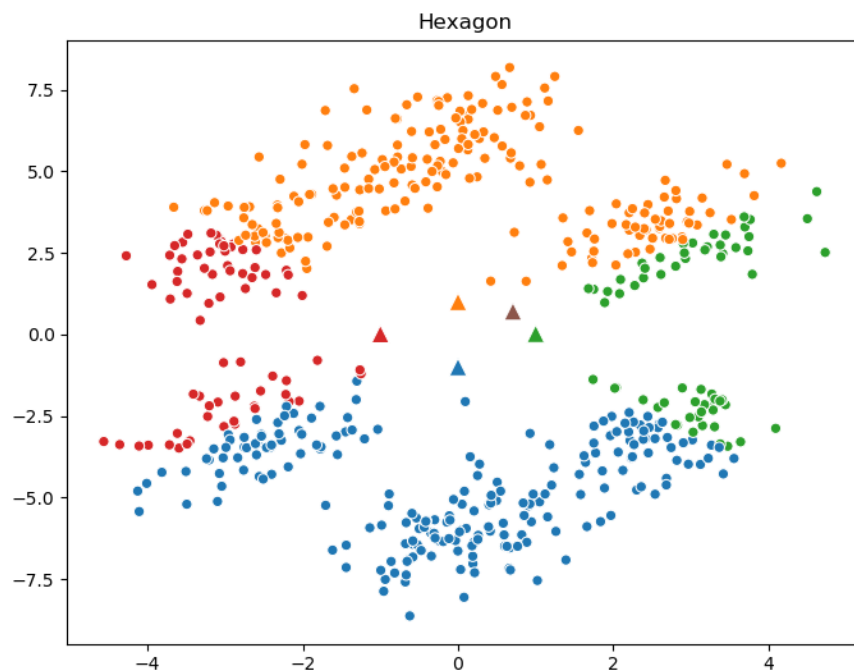
Simple cube



Rysunek 2: Wykresy klastryzacji zbioru cube. Trójkąty są reprezentantami klastrow.



Rysunek 3: Wykresy klastryzacji zbioru cube-nonmatching. Trójkąty są reprezentantami klastków.

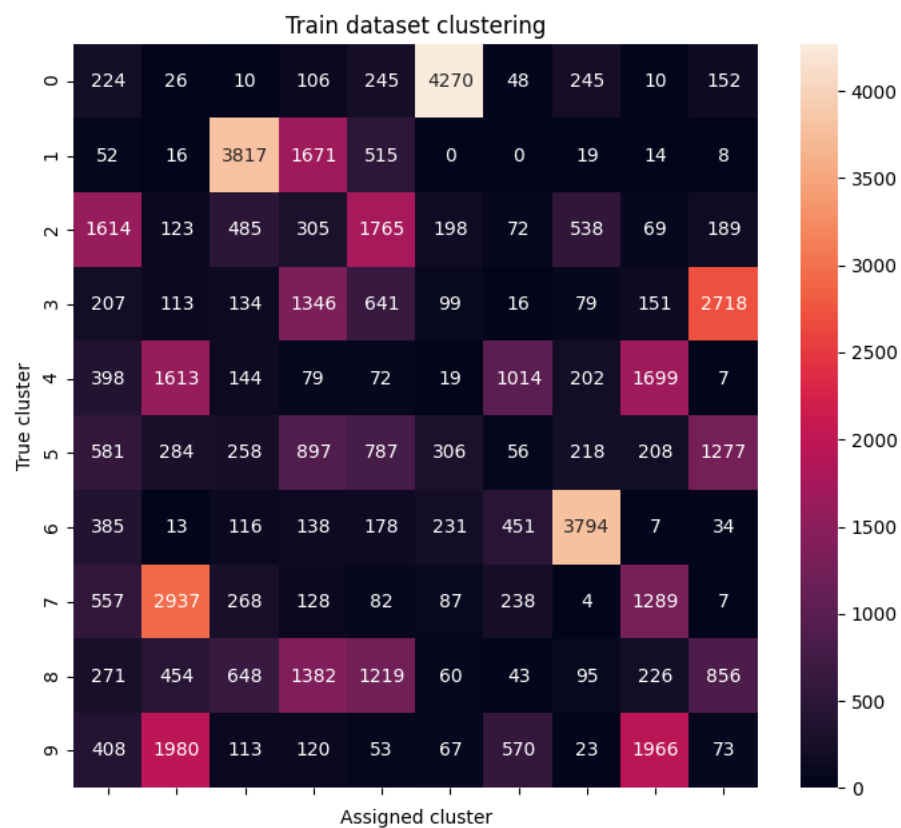


Rysunek 4: Wykres klastryzacji zbioru hexagon. Trójkąty są reprezentantami klastrów.

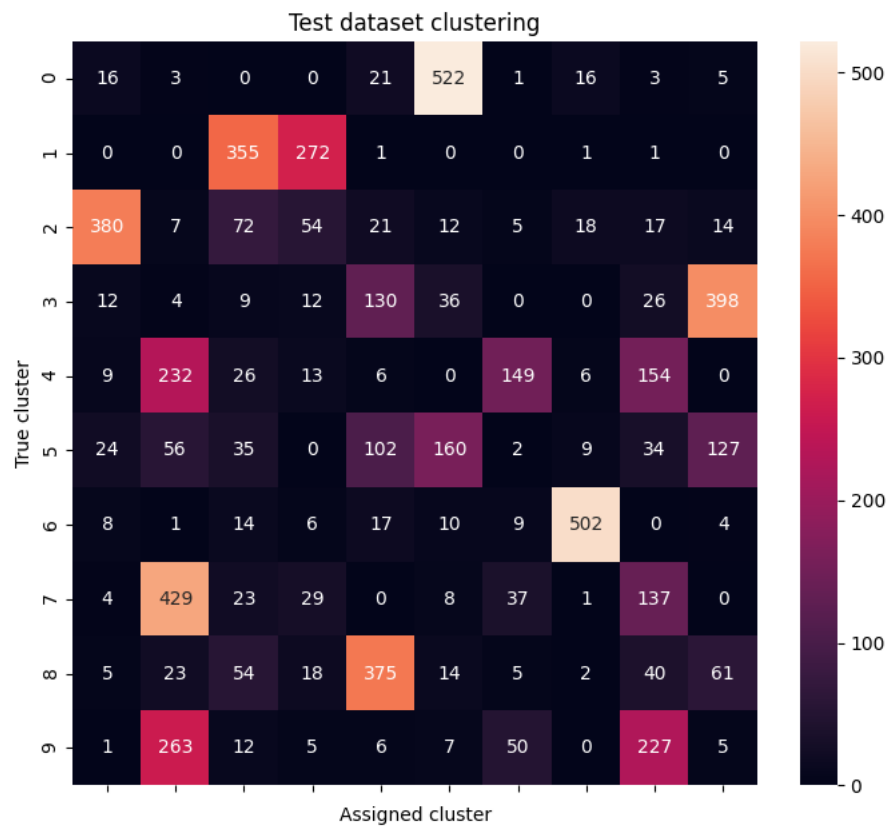
### 3.2 MNIST

Na zbiorze danych MNIST, zawierającym graficzny zapis cyfr oraz odpowiadającą im cyfrę jako klasę obiektu, zostały przeprowadzone kolejne testy. W pierwszym z nich zbiór został podzielony w proporcji 90% danych treningowych oraz 10% danych testowych.

Uzyskana skuteczność wyniosła 45% na zbiorze treningowym oraz 55% na zbiorze testowym. Rozkład elementów pomiędzy oryginalnymi klasami a przydzielonymi klastrami został przedstawiony w formie heatmapy.

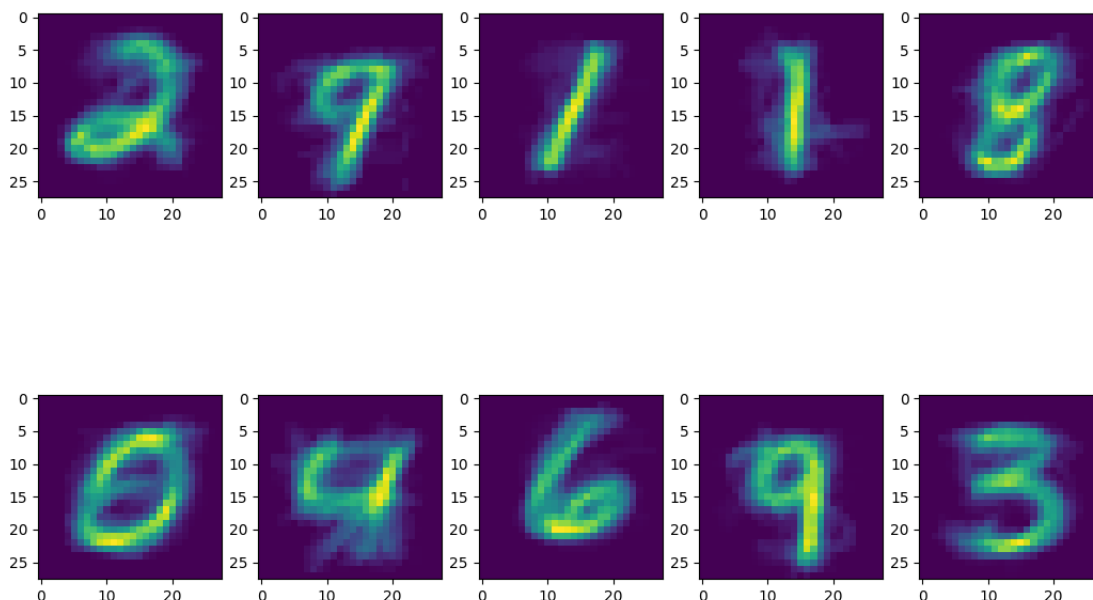


Rysunek 5: Schemat przedstawiający liczbę elementów danej klasy przydzielonych do danego klastra w zbiorze treningowym



Rysunek 6: Schemat przedstawiający liczbę elementów danej klasy przydzielonych do danego klastra w zbiorze testowym





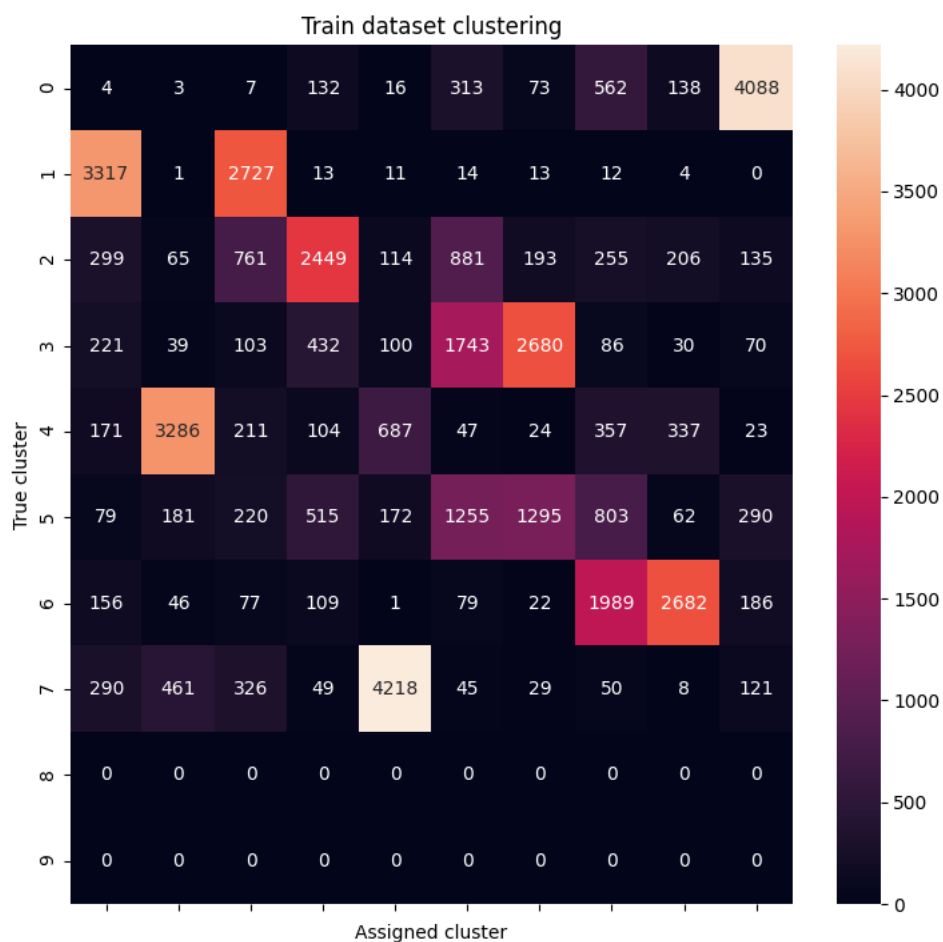
Rysunek 7: Graficzna wizualizacja wag top-down dla każdego z neuronów identyfikujących klastry

Możemy zauważyć na podstawie heatmap, że dla większości klas istnieje jeden klaster, który wyraźnie dominuje pośród innych przydzielonych tym punktom. Takie zjawisko obserwujemy m.in. dla cyfr „0”, „1”, „6”. W tych przypadkach także dana klasa jest dominująca podśród elementów w danym klastrze, co może świadczyć o tym, że ta zależność została poprawnie zidentyfikowana przez sieć. Jednocześnie obserwujemy takie cyfry, które nie są tak dobrze rozróżnialne np. „4”, „7” oraz „9”. Wszystkie z nich mają duży udział w dwóch tych samych klastrach. Podobny problem zachodzi także w przypadku cyfr „5” oraz „8”, co można łatwo wytłumaczyć, obserwując ich podobny kształt.

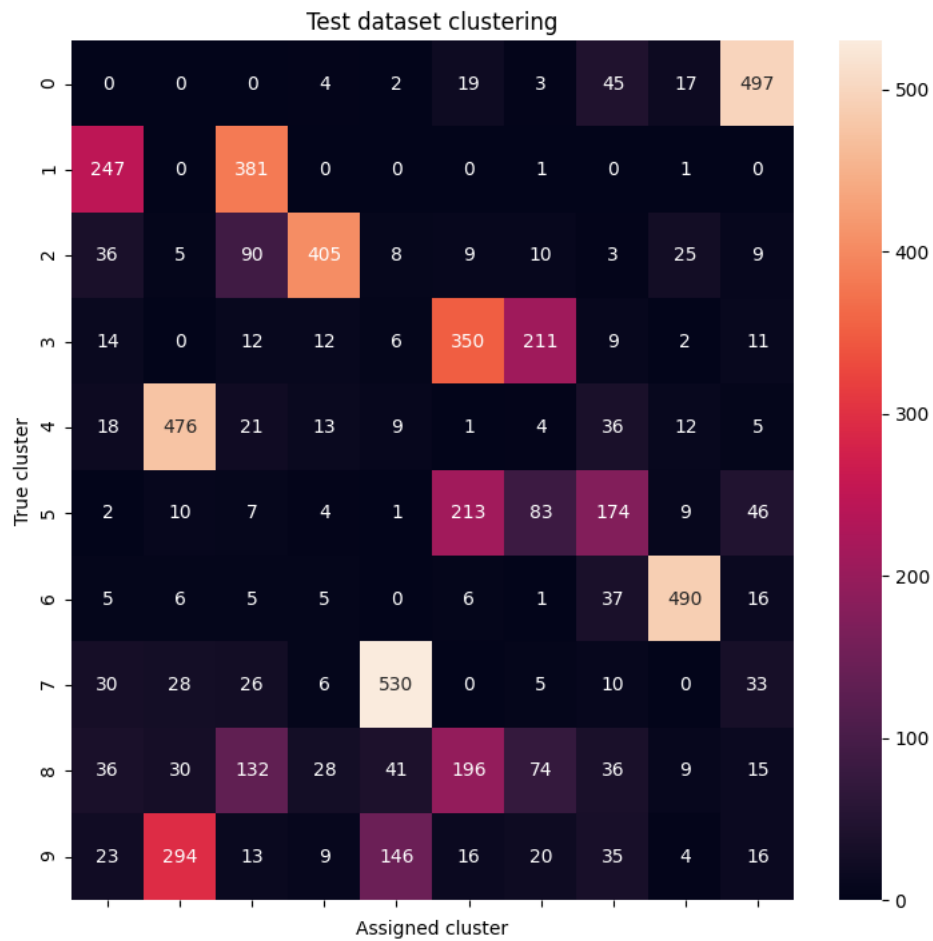
Korzystając z wizualizacji wag, widzimy, że nie istnieje klaster, reprezentowany przez cyfrę „7” - za to powstały dwa klastry reprezentowane przez „1” oraz dwa dla „9”, z których zawsze jedna jest pochyla. Może to stanowić jednocześnie przyczynę i wynik klastrowania siódmek wraz z jedynkami lub dziewiątkami.

Drugi test przeprowadzony z wykorzystaniem zbioru danych MNIST przebiegał podobnie, jednak z danych treningowych zostały wybrane jedynie rekordy klas o etykietach 0 – 7. Dane testowe pozostały niezmiennie. Efekt takiego na-

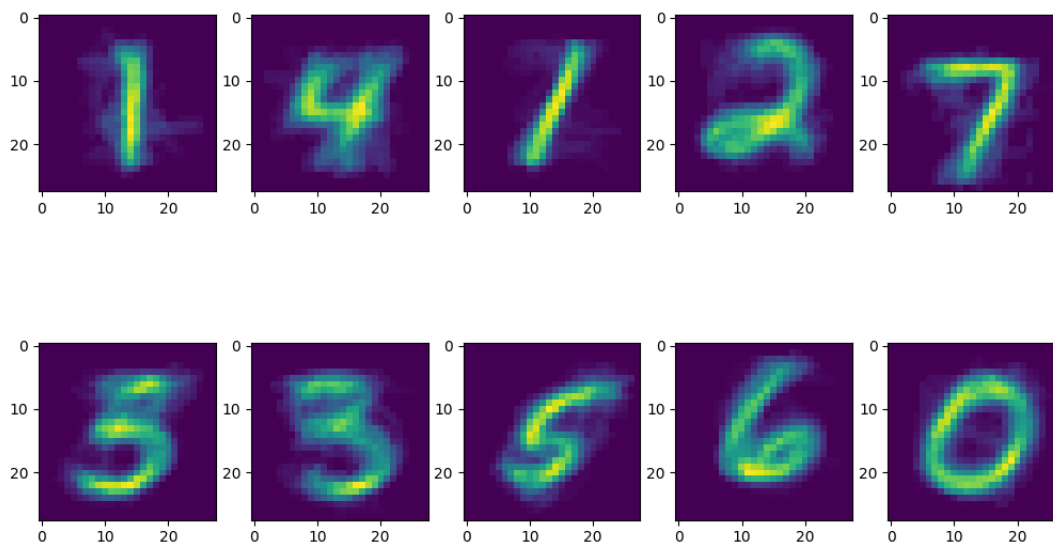
uczania przedstawiają poniższe schematy.



Rysunek 8: Schemat przedstawiający liczbę elementów danej klasy przydzielonych do danego klastra w niepełnym zbiorze treningowym



Rysunek 9: Schemat przedstawiający liczbę elementów danej klasy przydzielonych do danego klastra w zbiorze testowym po niepełnym treningu



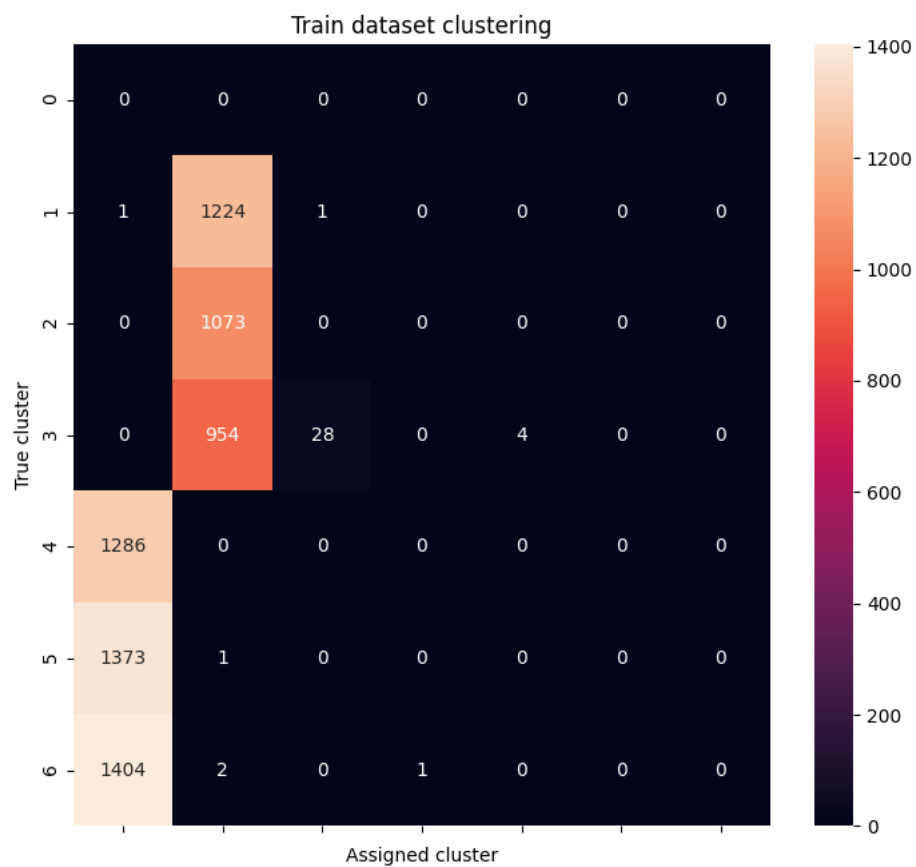
Rysunek 10: Graficzna wizualizacja wag top-down dla każdego z neuronów identyfikujących klastry

W procesie trenowania zostało utworzone 10 klastrów, odpowiadających jedynie 8 klasom, co obserwujemy także przy pomocy graficznej wizualizacji wag sieci. Na skutek tego w fazie testów na pełnym zbiorze zauważamy, że sposób klastrowania istniejących 8 klas pozostał proporcjonalny do zbioru treningowego. Sieć podjęła też próbę dopasowania elementów z klas „8” oraz „9” do istniejących klastrów. Cyfry „8” zostały przydzielone do klastrów złożonych głównie z piątek oraz trójek, a także, co ciekawe, do klastra zdominowanego przez jedynki, co może wynikać z tego, że część jedynek w zbiorze była zapisana w sposób pochyły. Z kolei dane z klasy „9” zostały przydzielone w znacznej większości do klastrów zawierających cyfry „4” oraz „7”, co wydaje się zrozumiałe z uwagi na ich podobny wygląd. Osiągnięta wartość funkcji skuteczności to 55% na zbiorze treningowym oraz 56% na testowym.

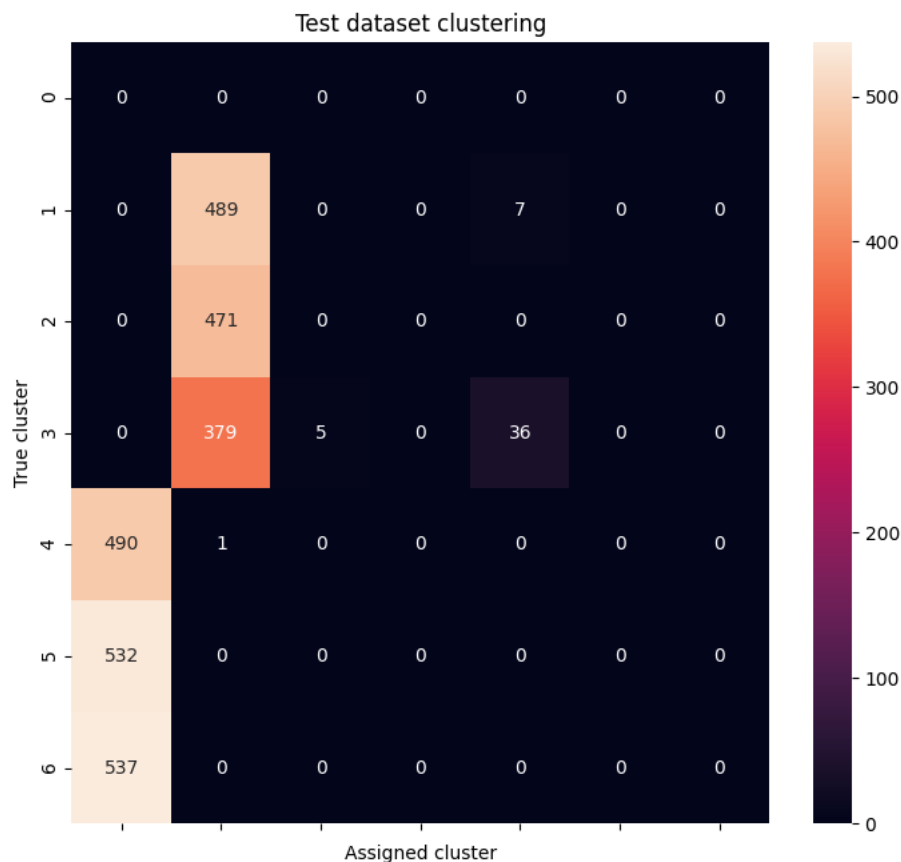
### 3.3 Zbiór danych Human Activity Recognition Using Smartphones

Sieć została także przetestowana przy pomocy zbioru Human Activity Recognition Using Smartphones, zawierającego dane zebrane ze smartfonów użytkowników wykonujących 6 podstawowych czynności, takich jak leżenie, siedzenie czy chodzenie. Czynności te zostały potraktowane jako klasy, które będziemy porównywać z efektem klastrowania. W przypadku tego zbioru danych istnieje

odgórny podział na dane treningowe i testowe w liczności odpowiednio 7352 oraz 2947. Wyniki klastrowania zostały przedstawione poniżej.



Rysunek 11: Schemat przedstawiający liczbę elementów danej klasy przydzielonych do danego klastra w zbiorze treningowym



Rysunek 12: Schemat przedstawiający liczbę elementów danej klasy przydzielonych do danego klastra w zbiorze testowym

Sieć dała niesatysfakcjonujące wyniki, będąc w stanie wyróżnić jedynie dwa klastry czynności, łącząc chodzenie, wchodzenie po schodach oraz schodzenie w nich w jeden zbiór oraz siedzenie, leżenie, stanie w drugi. Oczywiście można zauważyć bardzo prostą zależność - sieć odróżniła czynności statycznych od tych opisujących ruch, co jest poprawnym podziałem. Dane zostały też sklastrowane dość jednoznacznie - dla każdej klasy jest widoczna wyraźna dominacja jednego z klastrów. Ciekawym elementem jest jedyny wyraźny podzbiór klasy "schodzenie po schodach", odznaczający się zarówno w zbiorze treningowym, jak i testowym. Być może odpowiada on zbieganiu po schodach lub innemu ruchowi, który swoim dynamizmem odróżnia się od pozostałych danych.

Jednak żadna z przetestowanych konfiguracji parametrów sieci nie spowodowała zwiększenia jej czułości w wystarczającym stopniu, by rozróżnić wszystkie

sześć klas. Uzyskana skuteczność to 36% zarówno na zbiorze treningowym, jak i testowym.

## Literatura

- [1] Gail A Carpenter and Stephen Grossberg. Art 2: Self-organization of stable category recognition codes for analog input patterns. *Applied optics*, 26(23):4919–4930, 1987.
- [2] Laurene V Fausett. *Fundamentals of neural networks: architectures, algorithms and applications*. Pearson Education India, 2006.