Streszczenie Heloł<3

Raport

Sieć Hopfielda

Miłosz Pączkiewicz

Maciej Jakubiak

2018

# Skuteczność sieci

Został przeprowadzony szereg eksperymentów sieci za pomocą zbiorów reprezentujących obrazki. Podczas testów dla każdego zbioru generowanych były zaszumione obrazki (losowo spośród wzorców) o różnym poziomie zaszumienia. Poziom zaszumienia został określony przez parametr o wartości <0;1> i jego wartość oznacza prawdopodobieństwo, że dany piksel w obrazku zostanie zmieniony. Im większa wartość poziomu zaszumienia, tym bardziej zaszumiony był obrazek. Każdy z testów polegał na tym, że na wejściu przekazywaliśmy zaszumiony obrazek i sprawdzaliśmy czy obrazek na wyjściu jest równy obrazkowi, który został zaszumiony.

## Skuteczność dla dostarczonych zbiorów

Dla każdego zbioru generowaliśmy obrazki o siedmiu różnych stopniach zaszumienia: 0.05, 0.1, 0.15, 0.25, 0.3, 0.4 i 0.5. Dla każdego poziomu zaszumienia generowanych było po 1000 obrazków. Testy były przeprowadzane zarówno dla modelu synchronicznego jak i asynchronicznego co w sumie daje 14 000 losowo wygenerowanych zaszumionych obrazków dla każdego zbioru. Wyniki są następujące:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nazwa zbioru** | **Rozmiar zbioru** | **Współczynnik zaszumienia** | **Skuteczność modelu synchroniczny** | **Skuteczność modelu asynchronicznego** |
| animals-14x9 | 6 | 0.05 | 6% | 29.7% |
| animals-14x9 | 6 | 0.1 | 23.2% | 26.8% |
| animals-14x9 | 6 | 0.15 | 17% | 23.3% |
| animals-14x9 | 6 | 0.25 | 10.3% | 18.7% |
| animals-14x9 | 6 | 0.3 | 8.1% | 17.9% |
| animals-14x9 | 6 | 0.4 | 1.7% | 7.8% |
| animals-14x9 | 6 | 0.5 | 0.2% | 1.2% |
| large-25x25 | 6 | 0.05 | 70.4% | 73.3% |
| large-25x25 | 6 | 0.1 | 57.9% | 67.8% |
| large-25x25 | 6 | 0.15 | 51.5% | 61.7% |
| large-25x25 | 6 | 0.25 | 43.7% | 50.3% |
| large-25x25 | 6 | 0.3 | 38.8% | 49.5% |
| large-25x25 | 6 | 0.4 | 31.9% | 40.2% |
| large-25x25 | 6 | 0.5 | 1.2% | 1.6% |
| large-25x25.plus | 11 | 0.05 | 6.7% | 7% |
| large-25x25.plus | 11 | 0.1 | 6.9% | 6.6% |
| large-25x25.plus | 11 | 0.15 | 6.5% | 7.5% |
| large-25x25.plus | 11 | 0.25 | 7.5% | 5.9% |
| large-25x25.plus | 11 | 0.3 | 5.6% | 5.1% |
| large-25x25.plus | 11 | 0.4 | 1.8% | 1.6% |
| large-25x25.plus | 11 | 0.5 | 0% | 0% |
| letters-8x12 | 26 | 0.05 | 0% | 2% |
| letters-8x12 | 26 | 0.1 | 0% | 0.2% |
| letters-8x12 | 26 | 0.15 | 0% | 0% |
| letters-8x12 | 26 | 0.25 | 0% | 0% |
| letters-8x12 | 26 | 0.3 | 0% | 0% |
| letters-8x12 | 26 | 0.4 | 0% | 0% |
| letters-8x12 | 26 | 0.5 | 0% | 0% |
| letters-14x20 | 26 | 0.05 | 0% | 0% |
| letters-14x20 | 26 | 0.1 | 0% | 0% |
| letters-14x20 | 26 | 0.15 | 0% | 0% |
| letters-14x20 | 26 | 0.25 | 0% | 0% |
| letters-14x20 | 26 | 0.3 | 0% | 0% |
| letters-14x20 | 26 | 0.4 | 0% | 0% |
| letters-14x20 | 26 | 0.5 | 0% | 0% |
| letters-abc-8x12 | 3 | 0.05 | 0% | 2.7% |
| letters-abc-8x12 | 3 | 0.1 | 0% | 0.5% |
| letters-abc-8x12 | 3 | 0.15 | 0% | 0% |
| letters-abc-8x12 | 3 | 0.25 | 0% | 0.1% |
| letters-abc-8x12 | 3 | 0.3 | 0% | 0% |
| letters-abc-8x12 | 3 | 0.4 | 0% | 0% |
| letters-abc-8x12 | 3 | 0.5 | 0% | 0% |
| small-7x7 | 5 | 0.05 | 46.9% | 55.5% |
| small-7x7 | 5 | 0.1 | 44.4% | 47.5% |
| small-7x7 | 5 | 0.15 | 43.7% | 42.1% |
| small-7x7 | 5 | 0.25 | 36.8% | 34.4% |
| small-7x7 | 5 | 0.3 | 31.5% | 30.1% |
| small-7x7 | 5 | 0.4 | 20.6% | 20.6% |
| small-7x7 | 5 | 0.5 | 7.6% | 8.2% |

Dla porównania przy zerowym zaszumieniu skuteczność sieci była następująca dla modelu synchronicznego:

|  |  |
| --- | --- |
| **Zbiór** | **Skuteczność** |
| animals-14x9 | 34.7% |
| large-25x25 | 84.2% |
| large-25x25.plus | 8% |
| letters-8x12 | 0 |
| letters-14x20 | 0 |
| letters-abc-8x12 | 0 |
| small-7x7 | 61.9% |

Zakładając, że przy 50% zaburzeniu wejścia uzyskujemy obraz losowy można wysnuć następujące wnioski dla:

* animals-14x9.csv, large-25x25.plus.csv, letters-8x12.csv, letters-14x20.csv, letters-abc-8x12.csv - sieć zbiegała do punktów statycznych niebędących żadnym obrazkiem ze zbioru uczącego.
* large-25x25.csv, small-7x7.csv - sieć czasami zbiegała do punktu będącego jednym z obrazów uczących. Ten wniosek wynika stąd, że jeśli szukany był obrazek X, a na wejściu był losowy obraz to uzyskując X przyznawany był punkt. Być może dla innych losowych obrazów sieć zbiegała również do X, ale tego nie da się stwierdzić z powyższych danych.

Patrząc na wyniki można stwierdzić, że:

* Zbiory smal-7x7, large-25x25 uczą się w miarę poprawnie – w large-25x25 tylko 1 z 6 obrazów wejściowych nie był punktem stabilnym, a w smal-7x7 2 z 5 obrazków nie były punktami stabilnymi
* Zbiory large-25x25.plus, animals-14x9 uczyły się częściowo poprawnie – w large-25x25.plus jeden wzór z 15 był punktem stabilnym, a w animals-14x9 tylko 2 z 6 wzorów były punktami stabilnymi
* Pozostałe uczyły się niepoprawnie

Dlaczego uczyły się niepoprawnie? Ponieważ jak napisali: Liou, C.-Y.; Yuan, S.-K. (1999) w "Error Tolerant Associative Memory", Biological Cybernetics. 81: 331–342 obrazy uczące powinny znajdować w stabilnym wierzchołku (takim, z którego przekształcenie prowadzi do tego wierzchołka) N wymiarowej kostki, gdzie każdy wierzchołek reprezentuje jedną możliwą kombinację wartości -1 i 1 w wektorze o zadanej długości. Żeby to osiągnąć to wszystkie wzorce uczące powinny być jak najróżniejsze od siebie i nie powinno być ich więcej niż 13,8% \* długość wektora.

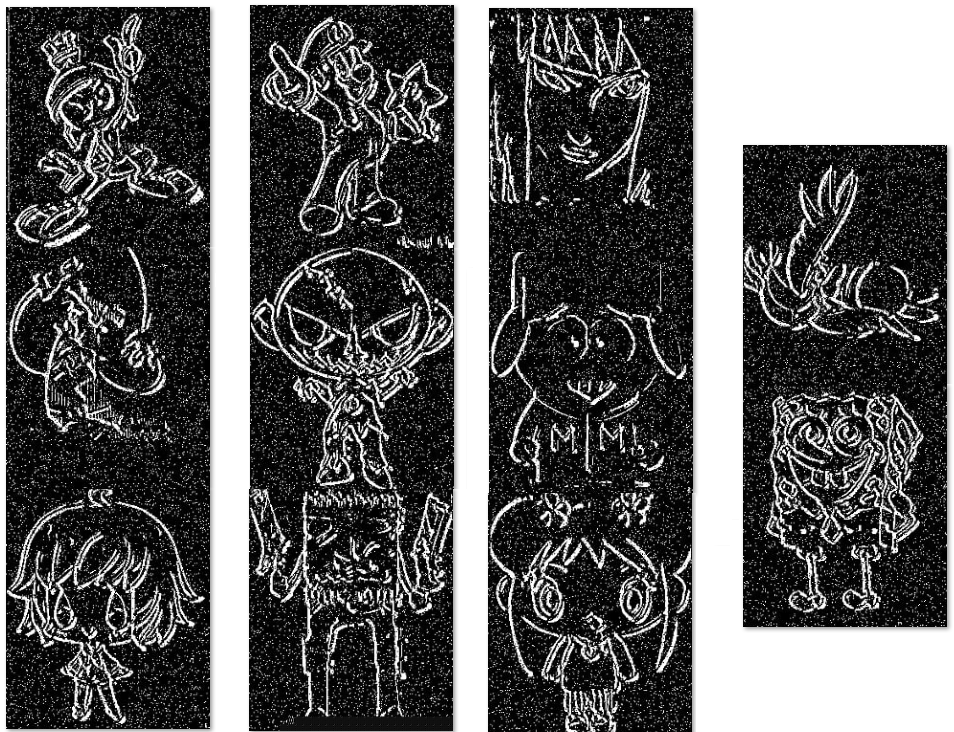
# Eksperymenty na dużych zbiorach

Stworzyliśmy dwa zbiory zawierające duże bitmapy – jeden zbiór zawierał 11 obrazków 149x149, a drugi zawierał 11 obrazków 250x300.

Obie sieci zbiegały do jednego punktu stabilnego dla dowolnego wejścia.

Ciężko było przeprowadzić symulacje na drugim zbiorze, ponieważ macierz 75000 x 75000 zajmowałaby w pamięci około 41,9 GB, a nie dysponujemy sprzętem o takiej ilości RAM-u. Całe szczęście po zmianie typu danych przechowywanych w macierzy z float64 do byte udało się zmniejszyć objętość macierzy do 5GB i przeprowadzić obliczenia.

Przykładowy test zawierający 11 zaszumionych obrazków rozmiaru 250x300:



Dla każdego z obrazków sieć zbiegała do następującego obrazka:



# Porównanie modelu synchronicznego i asynchronicznego

Model synchroniczny najczęściej zwracał gorsze wyniki niż model asynchroniczny, jednak nie jest to regułą. Wadą modelu asynchronicznego jest wielokrotnie dłuższy czas działania - model synchroniczny wymagał mniej niż 10 iteracji żeby osiągnąć stabilny stan, podczas gdy model asynchroniczny potrzebował kilkuset iteracji.

# Porównanie zbiorów gdzie elementy są podobne i niepodobne o podobnej liczności

Porównamy wyniki dla dostarczonych zbiorów: large-25x25.csv i animals-14x9.csv

W zbiorze large-25x25 znajdowały się takie obrazki jak: szachownica, poziome linie, domek, fragment alfabetu, słońce i strzała. W zbiorze zwierząt były takie obiekty jak: ryba, żółw, koń, pies, twarz i kaczka. W pierwszym zbiorze ciężko znaleźć podobieństwa między obiektami, natomiast w zbiorze zwierząt pies, żółw i kaczka mają nogi (poziome linie w podobnych miejscach), tułów (ciemny obszar na środku obrazka), i głowę (ciemny obszar po prawej stronie obrazka).



Patrząc na wyniki działania pies, żółw, koń i kaczka zbiegają do zdegenerowanego zwierzątka (nie przypomina zwierzątka), które jest podobne do każdego z nich. Twarz i ryba są poprawnie rozpoznawane. Dokładność sieci dla zwierząt z szumem na poziomie 5% wynosi około 30%

W zbiorze large-25x25 wszystkie obrazki poza strzałą są stanami stabilnymi, a dokładność sieci dla tych obiektów z szumem na poziomie 5% wynosi około 70%

# Maksymalny zbiór wektorów uczących 5x5, gdzie każdy wektor uczący jest stanem stabilnym nauczonej sieci

Nietrudno zauważyć, że zbiór składający się z 2^25 różnych wektorów (wszystkie możliwe obrazki 5x5) nauczy sieć Hopfielda tak, że każdy wzorzec z tego zbioru będzie stabilny (potrzebne jest dodatkowe założenie, że jeśli wynikiem obliczenia dla którejś pozycji jest 0 to przyjmujemy, że wynikiem jest to co było na wejściu). Wtedy ucząc taką sieć otrzymamy macierz W 25x25 składającą się z samych 0 i wtedy dla każdego X, WX = [0,0,0,0,0] -> X.

# Zbiór uczący i wektor testowy prowadzący do oscylacji

* Zbiór uczący:
  + X1 = [+1, +1, +1, +1, +1]
  + X2 = [+1, -1, -1, +1, -1]
  + X3 = [-1, +1, -1, -1, -1]
* Nauczona macierz (W):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | -1 | 1 | 3 | 1 |
| -1 | 0 | 1 | -1 | 1 |
| 1 | 1 | 0 | 1 | 3 |
| 3 | -1 | 1 | 0 | 1 |
| 1 | 1 | 3 | 1 | 0 |

* Wektor prowadzący do cyklu:
  + X = [+1, -1, -1, +1, +1]
* Obliczenie

WX = [+4, -2, +4, +4, -2]  
X’ = [+1, -1, +1, +1, -1]  
WX’ = [+4, -2, -2, +4, +4]  
X’’ = [+1, -1, -1, +1, +1]  
WX’’ = [+2, -2, +4, +4, -2]  
X’’’ = [+1, -1, +1, +1, -1] = X’

* Dlaczego prowadzi do cyklu

Zauważmy, że H(X2, X) = 1 oraz H(-X3, X) = 1 oraz wiedząc, że jeśli Y jest stanem stabilnym to –Y też jest stanem stabilnym, to widać że wektor X jest tak samo podobny do X2 i do –X3, więc sieć nie może zdecydować, do którego z nich X jest bardziej podobny, więc obliczenie wpada w cykl.

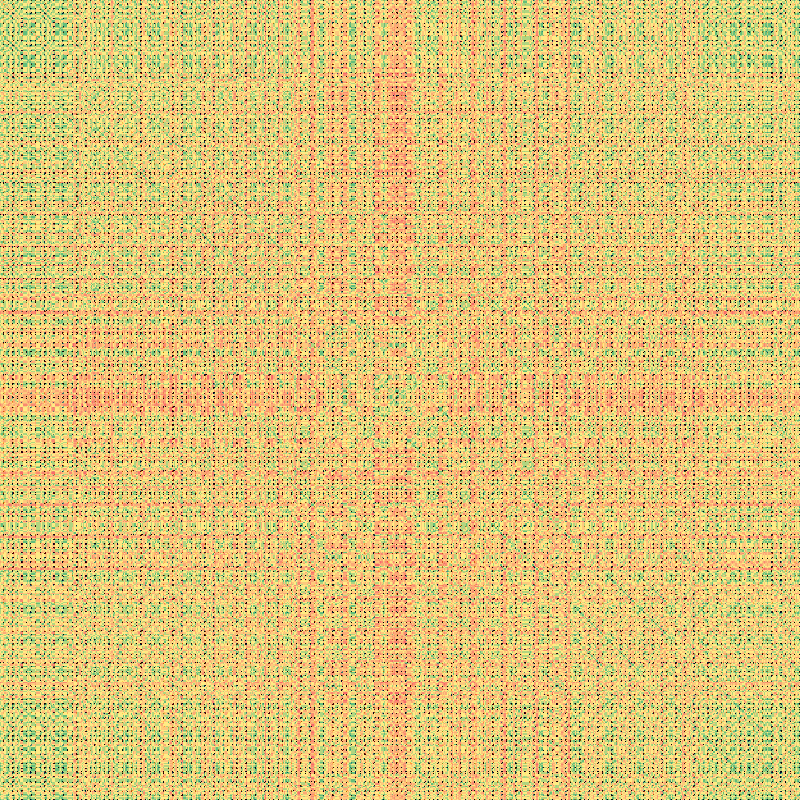
# Wizualna reprezentacja macierzy wag dla poszczególnych zbiorów

Przeprowadziliśmy wizualizacje macierzy wag dla wzorów zaprezentowanych w danych testowych. Wygląda ona następująco:

animals-14x9



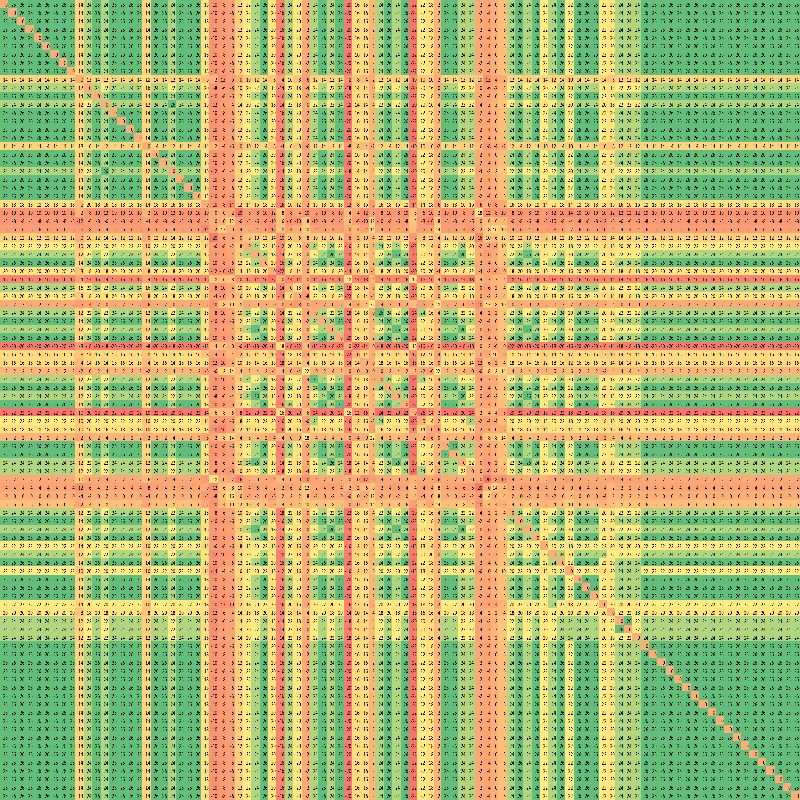
large-25x25



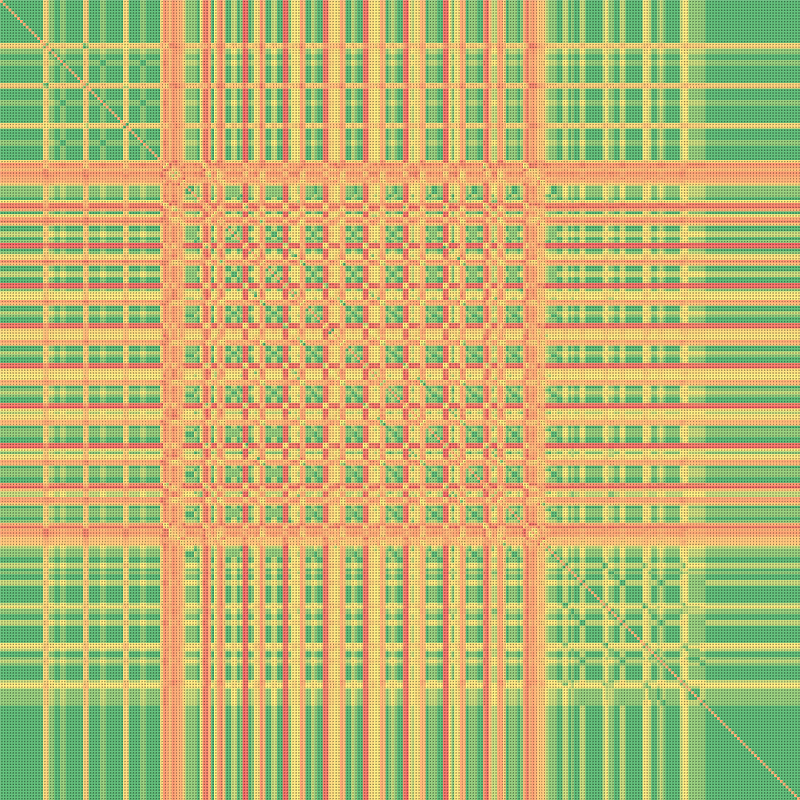
large-25x25.plus



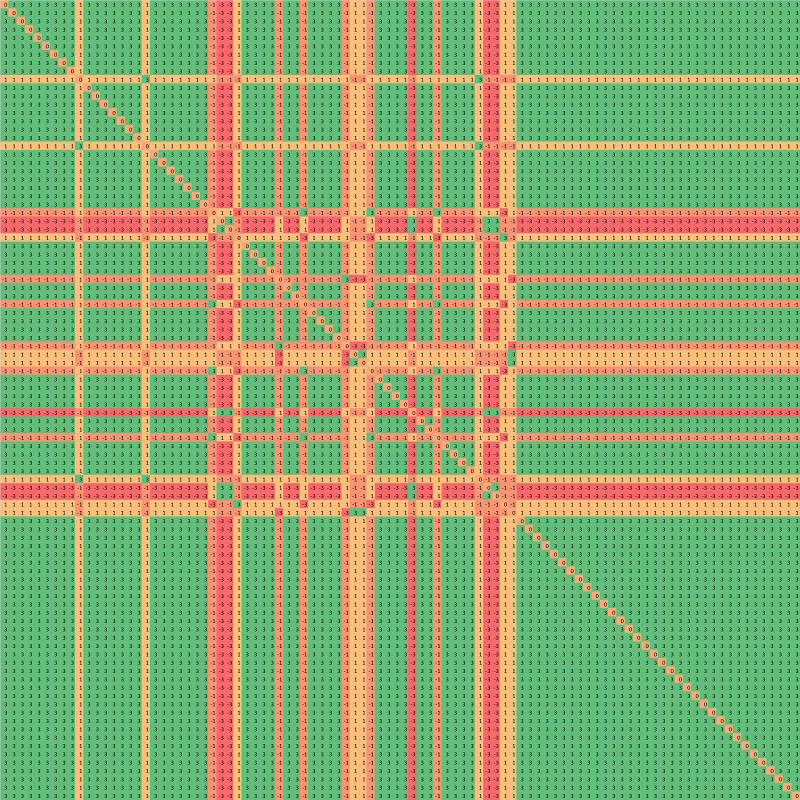
letters-8x12



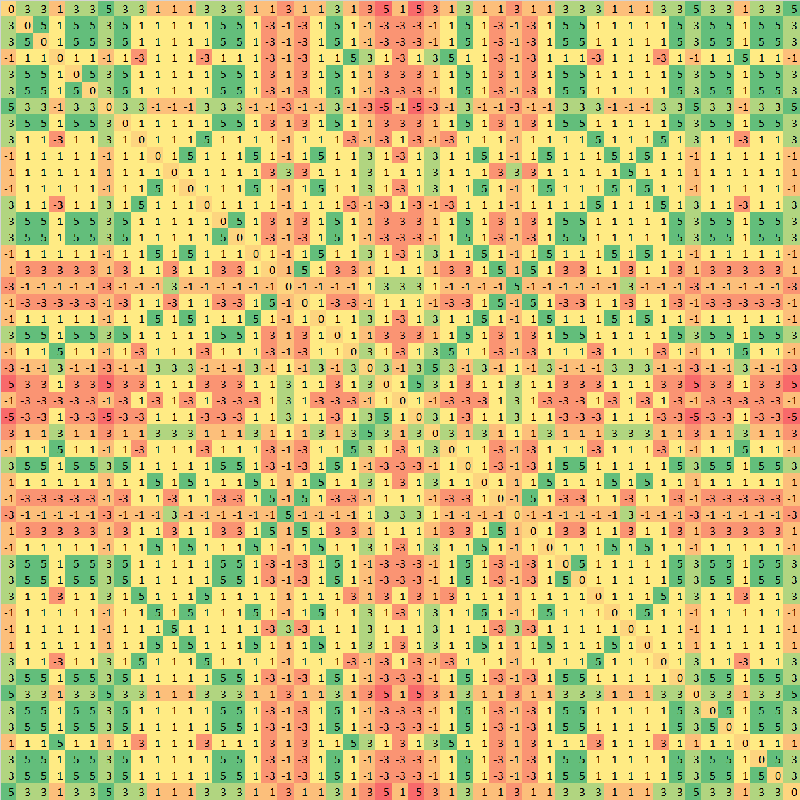
letters-14x20



letters-abs-8x12



small-7x7



Patrząc na powyższe rysunki można wysnuć wniosek, że im bardziej regularna macierz (na przykład dla letters) i można dopatrzeć się wzorów, powtórzeń i regularności, tym mniejsza jest skuteczność rozpoznawania wzorów. Jeśli macierz zawiera równomiernie rozłożone wartości ujemne i dodatnie to należy się spodziewać, że będzie lepiej rozpoznawała wzory (macierze inne niż dla letters).