# Sieci Hopfielda

Analiza przykładów sieci ciągłych i dyskretnych

**Maria Harbaty i Wiktoria Grodzka**

**Wydział MiNI, Politechnika Warszawska**

**styczeń 2025**

*Projekt przygotowany w ramach kursu Pakiety Matematyczne.*

# Spis treści

# 1 Wprowadzenie

## 1.1 Abstrakt

W tym raporcie zostaną przedstawione wyniki pięciu eksperymentów dotyczących sieci Hopfielda - zarówno ciągłych jak i dyskretnych. W pierwszej części zostanie przedstawiona analiza równań różniczkowych opisujących stany neuronów w sieci ciągłej a następnie przeanalizujemy rozbieżności wyników dla różnych danych wejściowych.
W drugiej części zostaną przedstawione sieci dyskretne służące do przechowywania wzorców oraz analiza stabilności danych modeli. Te przykłady pomogą zobrazować ideę sieci Hopfielda oraz ukażą ich zastosowanie w pamięci asocjacyjnej.

## 1.2 Zmiany w raporcie

Jedyną zmianą w stosunku do planu projektu jest zastąpienie problemu komiwojażera dodatkową analizą sieci ciągłych.

# 2 Ciągłe sieci Hopfielda

Do tego zagadnienia możemy podejść od strony równań różniczkowych. Wiemy, że funkcje ukazujące stany neuronów w czasie są rozwiązaniami takowych równań, dlatego aby lepiej zrozumieć działanie sieci neuronowych warto przeanalizować wcześniej wspomniane równania różniczkowe.

## 2.1 Analiza modelu z dwoma neuronami

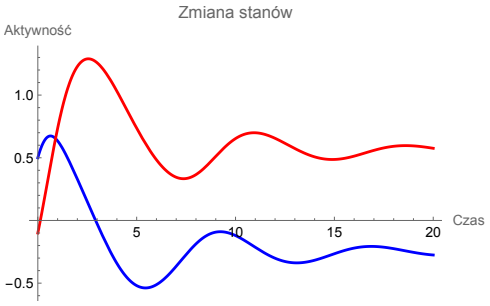
Najprostszym przykładem obrazującym oddziaływania w sieciach ciągłych będzie model składający się z dwóch neuronów. Należy zdefiniować macierz wag, która będzie $2 \times 2$, $\tau$ jako stałą oraz $I_i$ jako macierz $2 \times 1$. Równanie różniczkowe opisujące stany neuronów w czasie ma postać:

$$\tau \frac{dy_j}{dt} = -y_j + \sum_{j=1}^{N} w_{ij} v_j + I_i, \tag{1}$$

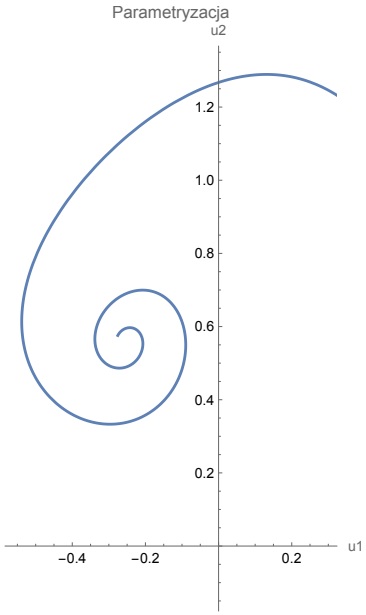
Mając już dwie rozwiązane numerycznie funkcje $u_1$ i $u_2$, możemy rozpocząć ich analizę. Ciekawe wyniki można otrzymać przy następujących danych początkowych:

$$\tau = 1,\ W = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix},\ I = \begin{bmatrix} 0.5 & 0.3 \end{bmatrix}.$$

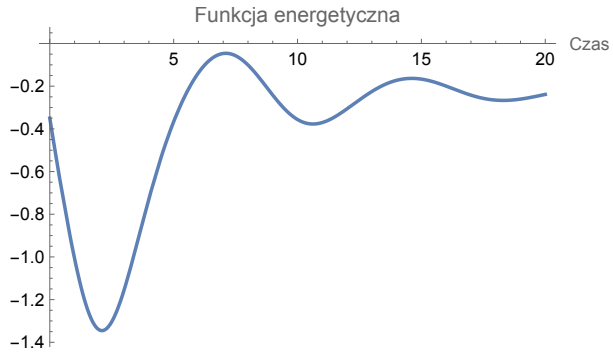
Poniższy wykres ukazuje zmianę stanów neuronów w czasie, gdzie wykres $u_1$ jest przedstawiony przez niebieską krzywą, a $u_2$ przez czerwoną.




Widać, że stany obu jednostek są ze sobą splątane. Można zauważyć, że przy zmianie aktywności drugiego neuronu, pierwszy będzie razem z nim ewaluował.
Następnie należy sprawdzić czy model dąży do stanu stabilnego. Na wykresie powyżej widać, że po pewnym czasie amplitudy maleją, więc można podejrzewać, że sieć jest blisko osiągnięcia pożądanego stanu. Można to również zobrazować za pomocą parametryzacji.




Ponownie widzimy, że wykres dąży do pewnego punktu. Jednak aby potwierdzić intuicje należy przeanalizować zmiany w funkcji energetycznej układu. Jeżeli będzie ona dążyła do 0, to będziemy pewni, że nasza sieć osiągnie stabilność. Kolejny wykres będzie przedstawiał zmiany funkcji energetycznej w czasie:

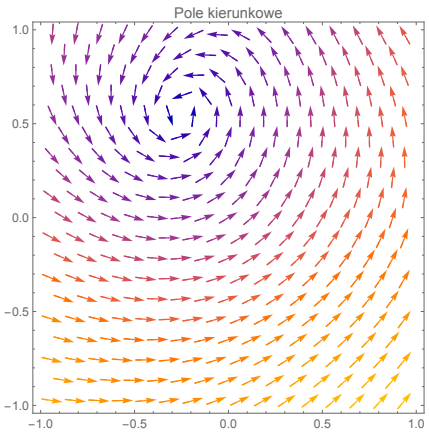



Otrzymujemy więc, że funkcja dąży do 0 wraz z upływem czasu, więc układ będzie osiągał stabilność. Dodatkowo przy analizie równań różniczkowych warto zbadać charaktetr punktów równowagi, czyli miejsc, kiedy pochodna funkcji się zeruje. Należy więc najpierw znaleźć pierwiastki równania (1), czyli:

$$0 = \frac{1}{\tau}(-y_j + \sum_{j=1}^{N} w_{ij} v_j + I_i)$$

Następnie należy policzyć macierz Jacobiego dla układu równań oraz podstawić wcześniej obliczone miejsca zerowe. Kolejnym krokiem jest policzenie wartości własnych tej macierzy i w zależności od znaku ich części rzeczywistych, punkt wcześniej wstawiony będzie stabilny albo niestabilny. Jeżeli wszystkie wartości własne będą miały ujemne części rzeczywiste, to punkt równowagi będzie stabilny, jednak jeżeli będzie występować dodatnia część rzeczywista, to punkt nie będzie stabilny.

W tym przykładzie macierz Jacobiego dla punktu $(-0.254228, 0.55645)$ posiada wartości własne $\lambda_1 = -0.158657 + 0.830152i$, $\lambda_2 = -0.158657 - 0.830152i$, więc wcześniej wymieniony punkt będzie stabilny. Można to również zobrazować na wykresie pola kierunkowego, gdzie wyraźnie widać, że wektory kierują się ku punktowi stabilnemu. Podobne zachowanie można zauważyć na wykresie, gdzie została przedstawiona parametryzacja funkcji $u_1$ oraz $u_2$.

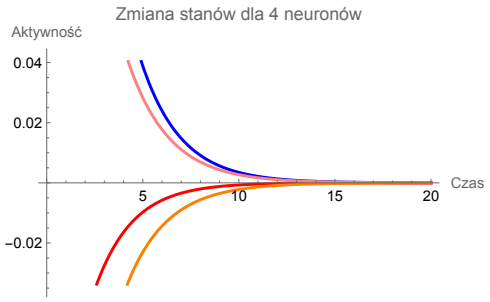



## 2.2 Zmiana pojedynczych danych

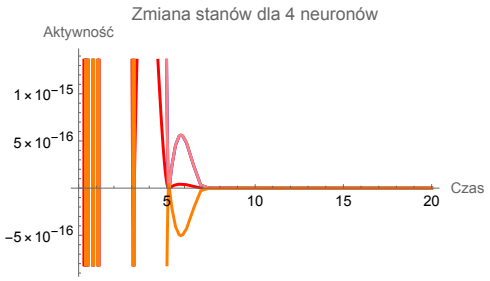
Przy tworzeniu sieci neuronowej ważne jest odpowiednie dobranie danych. Trzeba jednak najpierw zrozumieć znaczenie wprowadzanych przez nas parametrów.
Kolejnym przykładem będzie sieć składająca się z 4 neuronów. Niech macierz wag $W$oraz $I$ wyglądają następująco:

$$W = \begin{bmatrix} 0 & -\frac{1}{2} & \frac{1}{3} & -\frac{1}{4} \\ -\frac{1}{2} & 0 & \frac{1}{4} & -\frac{1}{3} \\ \frac{1}{3} & \frac{1}{4} & 0 & -\frac{1}{5} \\ -\frac{1}{4} & -\frac{1}{3} & -\frac{1}{5} & 0 \end{bmatrix}, I = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}.$$

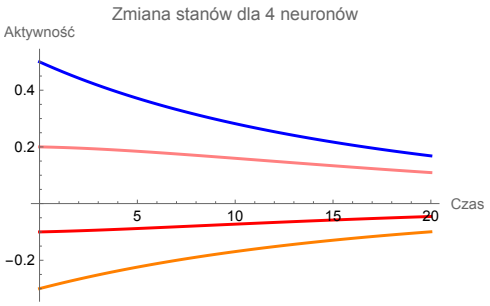
Należy się teraz zastanowić nad parametrem $\tau$. Najpierw niech $\tau = 1$, tak jak we wcześniejszym modelu. Wtedy wykres przedstawiający zmiany stanów neuronów w czasie wygląda następująco:




Widać, że oscylacje zachowań neuronów są znikome oraz wraz z czasem maleje ich aktywność. Co by się jednak wydarzyło przy przyjęciu innego parametru $\tau$? Niech teraz $\tau = 0.01$. Wykres przedstawiający zmiany stanów neuronów w czasie wygląda następująco:




Widać więc, że na początku występują intensywne oscylacje w zmianach stanów neuronów, które nie występowały przy wczesniejszym doborze $\tau$. Są to oczywiście zmiany na małej skali, ale jednak występują. Końcowo, ponownie sieć osiaga stan stabilny. Teraz niech $\tau = 10$.




Tutaj można zauważyć podobieństwo do pierwszego wykresu, gdzie $\tau = 1$, jednak zbieżnosc do 0 jest spowolniona.

### 2.3 Sieci ciągłe - wnioski

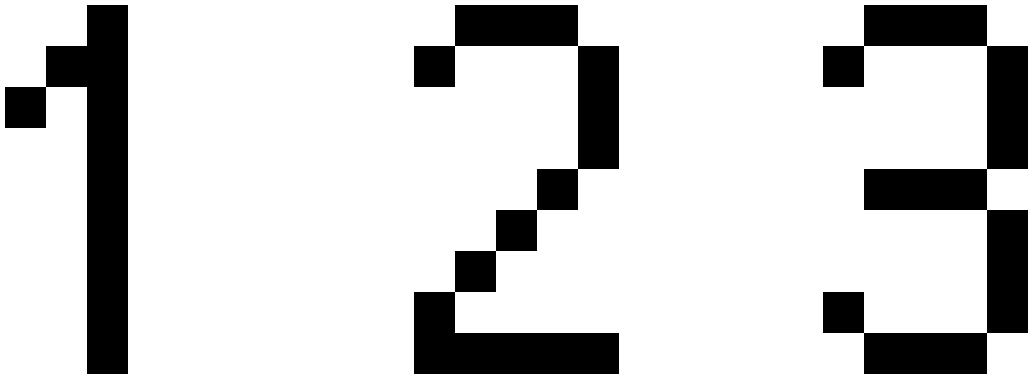
Podpunkt 2.1 przedstawia ilość metod, które mogą służyć w analizie stabilności danej sieci neuronowej. Parametryzacja oraz pole kierunkowe są analogiczne, jednak są najbardziej klarowne w $2D$, dlatego warto analizować wartości własne macierzy Jacobiego oraz zbieżnośc funkcji energetycznej.
Z wykresów 2.2 można wywnioskować, że $\tau$ odpowiada za 'wrażliwość' sieci neuronowej. Przy małym parametrze widać, że model jest bardziej podatny na oscylacje, jednak szybciej przyjmuje stan równowagi. Dla dużego $\tau$ nastepuje wolniejsza adaptacja stanu neuronów, jednak nie otrzymujemy dużych oscylacji. Wynika z tego, że przyjmowanie $\tau = 1$ jest dobrym wyborem i pozwala na optymalną szybkość stabilizacji bez większych błędów.

## 3 Dyskretne sieci Hopfielda

Przedstawimy działanie dyskretnych sieci Hopfielda na podstawie eksperymentu, w którym będziemy chciały nauczyć sieć o $n = 81$ neuronach rozpoznawania wzorców, które w tym przypadku będą kolejnymi cyframi. Naszym celem jest sprawdzenie jaki jest maksymalny stopień zaszumienia wprowadzanych danych, aby model zwrócił obraz prawidłowy.

### 3.1 Model z trzema wzorcami

W tym przypadku za wzorce przyjmujemy macierze o wymiarach $9 \times 9$, których elementami są wyłącznie liczby 1 i -1. Implementujemy odpowiednią funkcje, która wyrazy -1 zamienia na białe piksele, a wyrazy o wartości 1 zamienia na czarne piksele. W ten sposób wspomnniane wzorce można przedstawić w sposób jak powyższe obrazki.



Dla ułatwienia z macierzy wzorcowych $9\times9$ tworzymy listy o 81 elementach. Następnie tworzymy macierz wagową pamiętając, że elementy na przekątnej to 0 (co wynika z własności tej macierzy) i korzystając z poniższego wzoru dla $i \neq j$:

$$w_{ij} = \frac{1}{n} \sum_{k=1}^{p} a_i^k a_j^k$$

Tutaj $n$ to liczba neuronów, $p$ to liczba zapamiętywanych wzorców, a $a_i^k$ i $a_j^k$ to elementy listy $k$-tego wzorca o indeksach $i, j = 1, ..., 81$. Otrzymujemy macierz W o wymiarach $81 \times 81$.

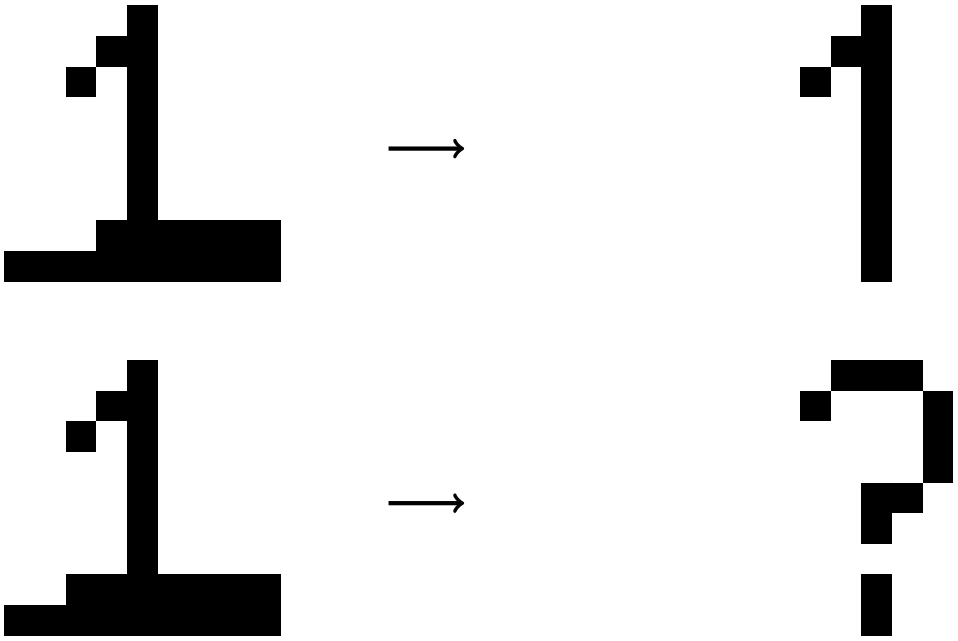
$$\mathrm{W} = \begin{bmatrix} 0 & \frac{1}{27} & \frac{1}{27} & -\frac{1}{81} & \dots \\ \frac{1}{27} & 0 & \frac{1}{27} & -\frac{1}{81} & \dots \\ \frac{1}{27} & \frac{1}{27} & 0 & -\frac{1}{81} & \dots \\ -\frac{1}{81} & -\frac{1}{81} & -\frac{1}{81} & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

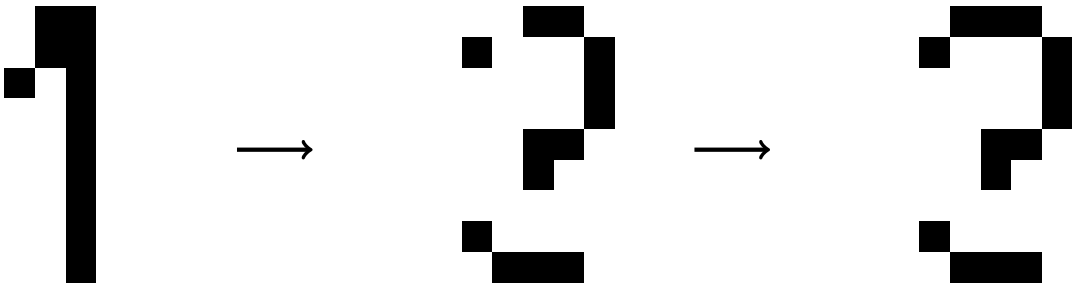
Zauważmy, że zgodnie z własnościami macierzy wagowej macierz W jest symetryczna.
Nazwijmy naszą zaszumioną macierz wejściową o wymiarach 9x9 *input*, następnie dla uproszczenia zamienimy ją na listę o 81 wyrazach. Roważamy poniższe funkcje dla $t = 1, ..., 10$, gdzie $t$ oznacza liczbę iteracji oraz tworzymy macierz zerową A o wymiarach $81 \times 10$ której kolumny to odpowiednio macierze wyjściowe po $(t-1)$-ej iteracji. Sprawdzimy, czy przy użyciu poniższych funkcji uda się zwrócić nauczony wzorzec i jeżeli tak to po ilu iteracjach.

$$A_i(t+1) = f\left(\sum_{j=0}^{n} w_{ij} A_j(t)\right)$$

$$f(y_j(t)) = \begin{cases} 1, & \text{gdy } \sum_{i=0}^{n} w_{ij} A_j(t) > 0, \\ -1, & \text{gdy } \sum_{i=0}^{n} w_{ij} A_j(t) \leq 0 \end{cases}$$

W ten sposób można sprawdzić maksymalną ilość pikseli, które można zamienić, aby algorytm zwrócił odpowiedni wzorzec oraz czy indeks piksela ma znaczenie w jego poprawności.
Niech po lewej stronie znajduje się macierz wejściowa, a strzałka oznacza jedną iterację. Po skrajnej prawej stronie mamy ostateczny obrazek wyjściowy, który nie ulega zmianie przy kolejnych iteracjach.



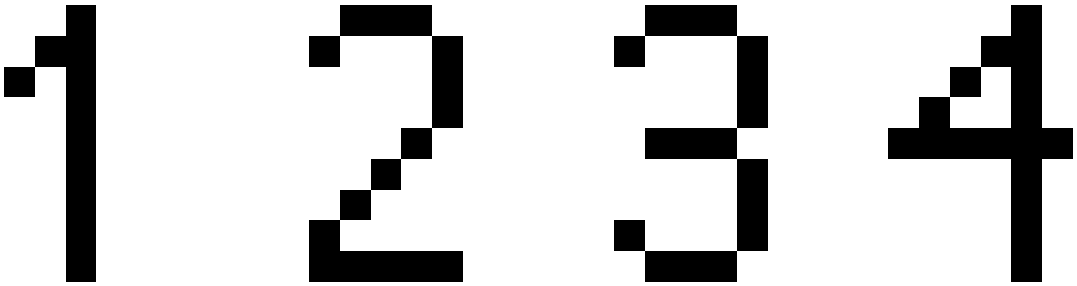


### 3.1.1 Wnioski

Z powyższych przykładów można zauważyć, że indeksy pikseli mają znaczenie. W pierwszym przypadku zmieniliśmy aż 13 elementów, a algorytm rozpoznał jedynkę po pierwszej interacji. Idąc tym tropem chciałyśmy zobaczyć czy analogiczna zamiana 14 elementów też zwróci poprawny wynik, jednak nie stało się tak w kolejnych iteracjach. Natomiast w trzecim przypadku zamieniłyśmy tylko jeden element, jednak algorytm nie był w stanie rozpoznać jedynki i próbował zwrócić macierz przedstawiającą trójkę. Możemy więc wnioskować, że miejsce występowania zaszumionych pikseli mają większe znaczenie niż ich ilość.

## 3.2 Model z czterema wzorcami

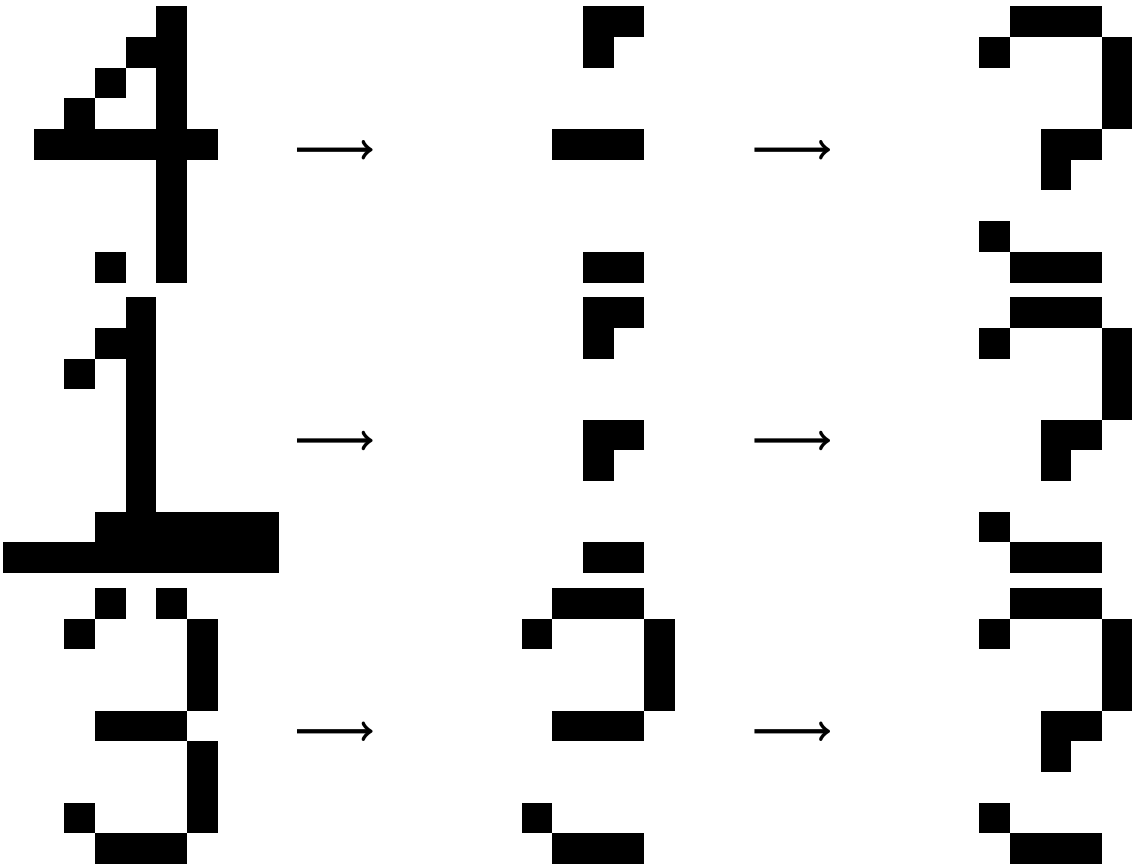
W analogiczny sposób możemy stworzyć sieć neuronową dyskretną przechowującą cztery wzorce.



Korzystamy z wyżej wspomnianego wzoru na macierz W dla $p = 4$ i otrzymujemy macierz wagową.

$$\mathrm{W} = \begin{bmatrix} 0 & \frac{4}{81} & \frac{4}{81} & 0 & -\frac{2}{81} & \dots \\ \frac{4}{81} & 0 & \frac{4}{81} & 0 & -\frac{2}{81} & \dots \\ \frac{4}{81} & \frac{4}{81} & 0 & 0 & -\frac{2}{81} & \dots \\ 0 & 0 & 0 & 0 & \frac{2}{81} & \dots \\ -\frac{2}{81} & -\frac{2}{81} & -\frac{2}{81} & \frac{2}{81} & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \end{bmatrix}$$

Tworzymy algorytm analogicznie do sieci o trzech wzorcach i obserwujemy co dzieje się z macierzami wejściowymi.



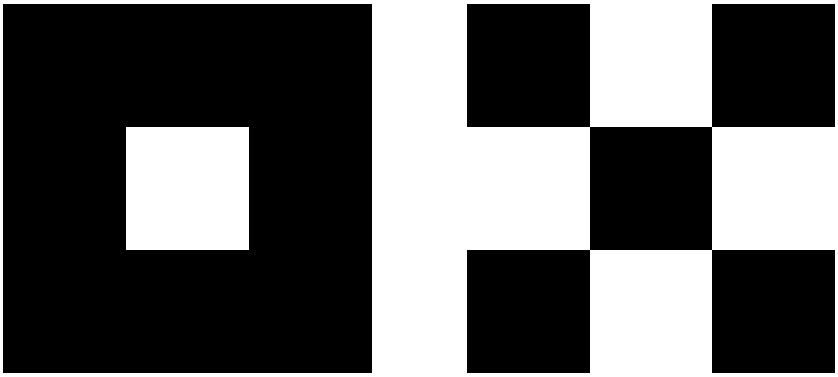
Po więcej niż dwóch iteracjach otrzymujemy ten sam obrazek wyjściowy co po dwóch iteracjach.

#### 3.2.1 Wnioski

W powyższym modelu można zauważyć zmniejszoną efektywność rozpoznawania przez sieć wzorców, ponieważ nawet przy wprowadzeniu poprawnego wzorca, nie został on zwrócony. Z tego przykładu można wyciągąć spostrzerzenie, że przy zwiększonej liczbie wzorców poprawność algorytmu maleje. Aby potwierdzić tą hipotezę zaimplementujemy kolejny przykład.

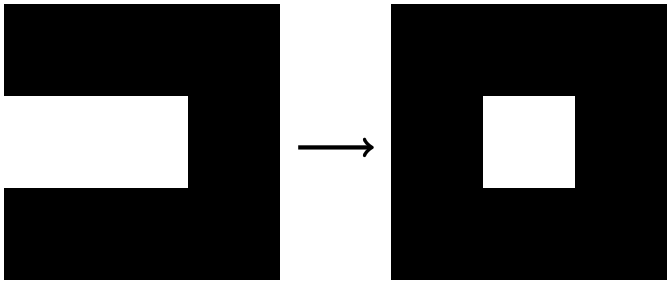
### 3.3 Model kółko i krzyżyk

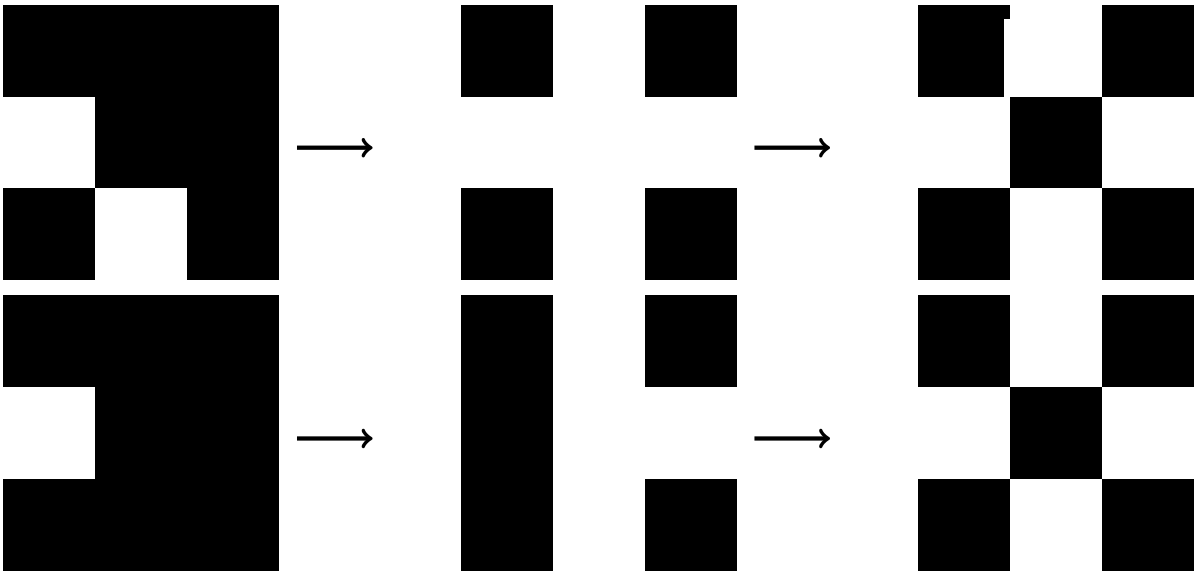
W tym podejściu spróbujemy nauczyć sieć składającą się z n=9 neuronów dwóch wzorców.



Ponownie korzystając ze wzoru dla $p = 2$ otrzymujemy poniższą macierz wagową.

$$\mathrm{W} = \begin{bmatrix} 0 & 0 & \frac{2}{9} & 0 & 0 & 0 & \frac{2}{9} & 0 & \frac{2}{9} \\ 0 & 0 & 0 & \frac{2}{9} & -\frac{2}{9} & \frac{2}{9} & 0 & \frac{2}{9} & 0 \\ \frac{2}{9} & 0 & 0 & 0 & 0 & 0 & \frac{2}{9} & 0 & \frac{2}{9} \\ 0 & \frac{2}{9} & 0 & 0 & -\frac{2}{9} & \frac{2}{9} & 0 & \frac{2}{9} & 0 \\ 0 & -\frac{2}{9} & 0 & -\frac{2}{9} & 0 & -\frac{2}{9} & 0 & -\frac{2}{9} & 0 \\ 0 & \frac{2}{9} & 0 & \frac{2}{9} & -\frac{2}{9} & 0 & 0 & \frac{2}{9} & 0 \\ \frac{2}{9} & 0 & \frac{2}{9} & 0 & 0 & 0 & 0 & 0 & \frac{2}{9} \\ 0 & \frac{2}{9} & 0 & \frac{2}{9} & -\frac{2}{9} & \frac{2}{9} & 0 & 0 & 0 \\ \frac{2}{9} & 0 & \frac{2}{9} & 0 & 0 & 0 & \frac{2}{9} & 0 & 0 \end{bmatrix}$$





### 3.4 Wnioski

Widzimy, że przy mniejszej liczbie wzorców i neuronów algorytm jest w stanie rozpoznać nawet znacznie zaszumione obrazy. Zauważmy, że w przypadku drugim i trzecim algorytm wybrał wzorzec krzyżyka zamiast kółka, choć w trzecim przypadku macierzy wejściowej bliżej do macierzy reprezentującej kółko.

### 3.5 Sieci dyskretne - Podsumowanie

Na podstawie przeprowadzonych eksperymentów możemy stwierdzić, że przy małej liczbie neuronów i przechowywanych wzorców sieć jest w stanie efektywnie rozpoznawać i zapamiętywać wektory uczące. Zauważmy, że tutaj macierz wagowa odgrywa znaczącą rolę. Choć z logicznego punktu widzenia widzimy, do którego wzorca powinna być sprowadzona macierz wejściowa, to nie zawsze tak jest. Istotną rolę odgrywają indeksy zaszumionych pikseli i często są one ważniejsze niż ilość zaszumionych elementów.

## 4 Wnioski

Jako że przeanalizowałyśmy modele sieci dyskretnych składających się z różnej liczby neuronów i wzorców, możemy wywnioskować, że im większa liczba neuronów i wzorców uczących w sieci tym mniejsza efektywność modelu. Możnaby wysnuć tezę, że taka sama zależność zachodzi w sieciach ciągłych Hopfielda.

Dzięki pokazanym przykładom sieci dwuneuronowej ciągłej i dziewięcioneuronowej dyskretnej jesteśmy w stanie zaobserwować charakter względnie stabilnych sieci Hopfielda. Z przedstawionych wykresów opisujących zachowanie sieci ciągłej względem czasu widzimy, że układ dąży do stabilności. Analogiczne zachowanie obserwujemy w przykładzie z kółkiem i krzyżykiem, gdzie wraz ze wzrostem liczby iteracji wynik dąży do zwrócenia wzorca.

Co ciekawe, w przykładzie sieci dyskretnej z czterema wzorcami uczącymi wraz z kolejnymi iteracjami algorytm również dąży do konkretnego obrazka, jednak nie jest nim żaden z zadanych wzorców. Dzieki tej obserwacji możemy zobaczyć jak istotną rolę ma macierz wagowa w rozważanym algorytmie.