Rozpoznawanie liter z użyciem sieci Hopfielda

Jakub Domogała Martyna Olszewska

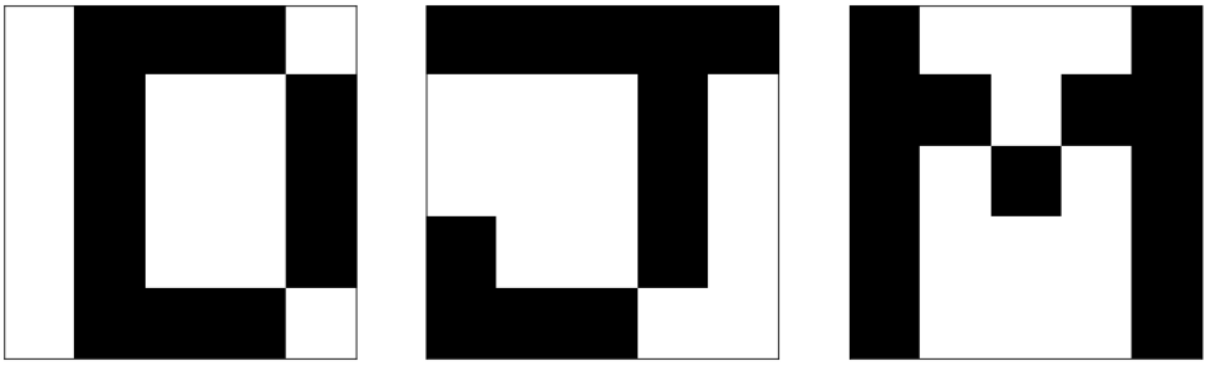
Projekt 2 na przedmiot Algorytmy inspirowane biologicznie

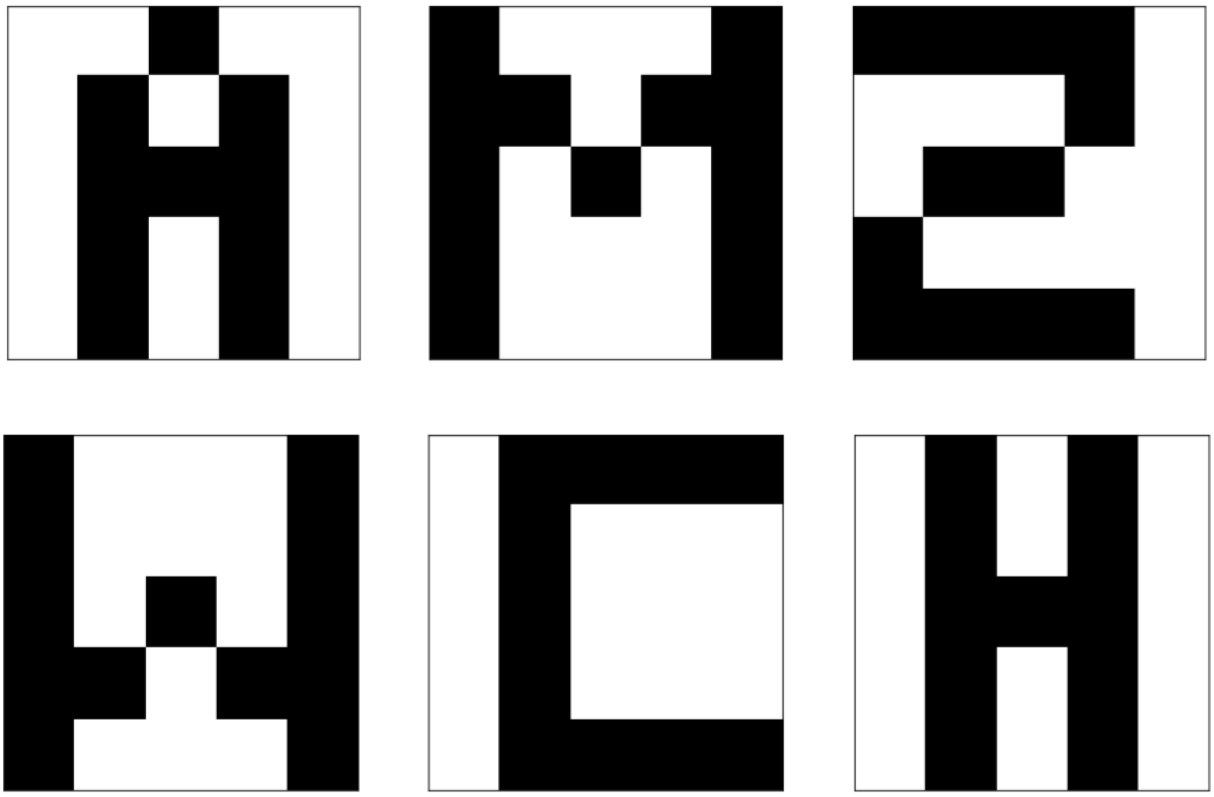
OMÓWIENIE

Naszym celem było stworzenie programu, który miał za zadanie nauczyć model rozpoznawać litery na, mając wcześniej podane wzorce.

OPIS DANYCH

Naszymi danymi były siatki 5x5 złożone z kwadratów. Na tej siatce możliwe jest narysowanie wzoru oraz liter testowych. Do przetestowania naszego modelu wykorzystaliśmy różne ilości siatek, które różniły się od wzorców w 1, 2 lub 3 pikselach, czyli odpowiednio dla tych wartości siatki miały losowo zmienione kwadratów. Zdecydowaliśmy się na taki wybór wartości ponieważ, dla większej ilości nawet dla nas było ciężko rozpoznać prawidłowy wzorzec. Stworzyliśmy kilka zestawów danych uczących, kilka z nich miało podobne litery, a kilka zupełnie różne. Poniżej przykład naszych wzorowych liter.





UŻYTE OPROGRAMOWANIE

Do stworzenia programu użyliśmy języka Python. Postanowiliśmy sami napisać implementacje sieci Hopfielda dlatego nie korzystamy z żadnej biblioteki, która dostarcza potrzebne funkcje. Korzystamy z bibliotek matplotlib do rysowania naszych siatek z literami oraz biblioteki numpy do obliczenia bardziej zaawansowanych działań matematycznych.

ETAPY TWORZENIA MODELU

Pierwszym etapem było stworzenie wzorców. Następnie wypełniamy macierz wag o wielkości NxN gdzie N to ilość neuronów, czyli w naszym przypadku będzie to 25, bo siatka jest wielkości 15x5.

Do obliczenia wartości do wypełnienia macierzy skorzystaliśmy z reguły Hebba:

$$w_{ij} = \frac{1}{N} \sum_{k=1}^{K} x_i^{(k)} x_j^{(k)}$$

Gdzie K to liczba wszystkich wektorów uczących, czyli u nas było to liczba wzorów 3. N to Liczba neuronów. x(i) i x(j) to odpowiednie wartości wektorów uczących, które mają wartości 1 lub -1. Na głównej przekątnej wartości równają się 0.

Skorzystaliśmy z dystansu Hamminga, aby porównywać literę testową i odpowiednie wzory i wybrać ten którego ta odległość jest najmniejsza czyli jest najbardziej podobny do przekształconego testu zgodnie z wzorami aktywacji pojedynczego neuronu:

$$1 \quad \text{, gdy} \quad \sum_{j=1}^{N} w_{ij} y_j(k) + x_i(k) > 0$$

$$-1 \quad \text{, gdy} \quad \sum_{j=1}^{N} w_{ij} y_j(k) + x_i(k) < 0$$

WYNIKI

Aby sprawdzić poprawność naszego modelu wykonaliśmy po 100 testów dla każdego z zestawów liter z różną ilością zmienionych pikseli 1, 2, lub 3.

piksele	A	M	Z
1	**86**, 4 ,10	84, **6**,10	77,4,**19**
2	**85**,3,12	79,**7**,14	82,11,**7**
3	**84**,7,9	79,**12**,9	83,3,**14**
	D	J	M
1	**54**,6,40	51,**15**,34	51,11,**38**
2	**54**,13,33	47,**11**,42	44,16,**40**
3	**51**,12,37	52,**10**,38	43,10,**47**
	W	C	H
1	**9**,76,15	6,**73**,21	3,81,**16**
2	**10**,74,16	6,**80**,14	8,80,**12**
3	**8**,77,15	7,**73**,20	9,77,**14**

Jak widać wyniki nie są zbyt satysfakcjonujące. Dla pierwszego i trzeciego zestawu liter dominuje wybór jednej litery i to tylko w jednym przypadku tej właściwej. W drugim zestawie wyniki są już bardziej wyrównane. Nie widać zbytniej różnicy w wynikach kiedy mamy zwiększoną ilość zmienionych pikseli