

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Wydział: EEIA Rok akad.: 2017/2018

Grupa: 7D30 Semestr: VII

Dzień tygodnia: Niedziela

**Sprawozdanie z ćwiczenia z**

**Podstawy Sztucznej Inteligencji**

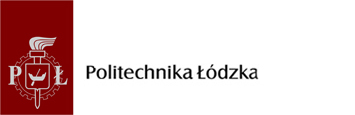
Temat ćwiczenia: **Perceptron**

|  |  |  |  |
| --- | --- | --- | --- |
| Data wykonania ćwiczenia | Podpis | Data oddania sprawozdania | Podpis |
| 01.12.2017 |  | 01.12.2017 |  |

Wykonał:

Radosław Subczyński

Nr inteksu 200137



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Perceptron – najprostsza [sieć neuronowa](https://pl.wikipedia.org/wiki/Sie%C4%87_neuronowa), składająca się z jednego bądź wielu niezależnych [neuronów McCullocha-Pittsa](https://pl.wikipedia.org/wiki/Neuron_McCullocha-Pittsa), implementująca algorytm [uczenia nadzorowanego](https://pl.wikipedia.org/wiki/Uczenie_nadzorowane) [klasyfikatorów binarnych](https://pl.wikipedia.org/w/index.php?title=Klasyfikacja_binarna&action=edit&redlink=1). Perceptron jest funkcją, która potrafi określić przynależność parametrów wejściowych do jednej z dwóch klas. Może być wykorzystywany tylko do klasyfikowania zbiorów [liniowo separowalnych](https://pl.wikipedia.org/w/index.php?title=Liniowa_separowalno%C5%9B%C4%87&action=edit&redlink=1).



private ArrayList<Point> generatePointList() {  
 ArrayList<Point> pointsList = new ArrayList<Point>();  
 for (int i = 1; i < 10; i++) {  
 for (int j = 1; j < 10; j++) {  
 if (i < j) {  
 pointsList.add(new Point(i, j, POINT.*GROUP1*));  
 } else {  
 pointsList.add(new Point(i, j, POINT.*GROUP2*));  
 }  
 }  
 }  
  
 pointsList.add(new Point(2, 4, POINT.*NOTHING*));  
 return pointsList;  
}

enum POINT {  
 *GROUP1*, *GROUP2*, *NOTHING*}

Wygenerowanie punktów do nauki wraz z flaga Grupa 1 i Grupa 2,

Nothing jest flagą która oznacza punkt nie należący niczego, i chcąc sprawdzić do czego należy.

private void addPoint() {  
 for (Point point : generatePointList()) {  
 g2.fillOval(point.getX(), point.getY(), 5, 5);  
 groupPoint(point);  
 }  
}

funkcja rysujące wykres na jFreimie

private void searchGroup(Point point) {  
 for (Point pointInList : pointListGroup1) {  
 if (pointInList.getX() == point.getX() && pointInList.getY() == point.getY()) {  
 System.*out*.print("Punkt nalezy do grupy 1 ");  
 return;  
 }  
 }  
  
 for (Point pointInList : pointListGroup2) {  
 if (pointInList.getX() == point.getX() && pointInList.getY() == point.getY()) {  
 System.*out*.print("Punkt nalezy do grupy 1 ");  
 return;  
 }  
 }  
 System.*out*.print("Nie znaleziono punktu");  
}

funkcja porównująca do której grupy należy dany punkt.który uczyliśmy program.