

Modelowanie działania pojedynczego perceptronu

Zadanie: klasyfikacja dwóch zbiorów liniowo separowalnych na płaszczyźnie kartezjańskiej.

Funkcja aktywacji

```
unsigned int activationFunction(Point const & p)
{
    double a = vWeight[0] * p.getX() + vWeight[1] * p.getY() + vWeight[2];
    return a > 0 ? SHAPE::SQUARE : SHAPE::CIRCLE;
}
```

Uczenie

Perceptron uczy się wykonując następującą metodę:

```
void learn(std::vector<Point> vPoints, std::vector<unsigned int> vTargetFunction)
{
    for (int i = 0; i < vPoints.size(); i++)
    {
        unsigned int yi = activationFunction(vPoints[i]);
        int e = vTargetFunction[i] - yi;
        if (e != 0) ①
        {
            double x = vPoints[i].getX();
            double y = vPoints[i].getY();
            vWeight[0] += e * x;
            vWeight[1] += e * y;
            vWeight[2] += e;
        }
    }
}
```

① Jeżeli algorytm nie odpowiedział prawidłowo popraw wagi

Testowanie

```

    unsigned int test(std::vector<Point> vPoints, std::vector<unsigned int>
vTargetFunction)
    {
        unsigned int result = 0;
        for (int i = 0; i < vPoints.size(); i++)
        {
            unsigned int yi = activationFunction(vPoints[i]);
            int e = vTargetFunction[i] - yi;
            if (e != 0) result++;
            std::cout << e;
        }
        return result;
    }

```

Wyniki

Punkty potrzebne do wykonania zadanie generuje losowo w programie metodą rand(). Funkcja która określa rodzaj punktu (1 lub 0) to:

$$f(x) = 1.2x + 3$$

l. pkt. do nauki	l. pkt. testowych	l. błędnych odpowiedzi	funkcja perceptronu
8	2	0	$F(x) = 1.93642x + 0.67$
80	20	1	$F(x) = 1.18918x + 3.34$
800	200	3	$F(x) = 1.23999x + 2.59$
4000	1000	4	$F(x) = 1.22961x + 4.89$
8000	2000	33	$F(x) = 1.18356x + 2.86$

Wnioski

Około 100 punktów do nauki zapewnia dobrą funkcję do określenia rodzaju punktu.