****

**Московский авиационный институт**

**(Национальный исследовательский университет)**

**Институт № 3**

**Кафедра 311**

**Интеллектуальные системы**

**Лабораторная работа № 3**

**«Нейронная сеть Хопфилда»**

**Выполнил студент  
Плотников Антон Сергеевич**

**Группа М3З-301-БК**

**Дата 28.04.2021 г.**

**Принял преподаватель  
Кос Оксана Игоревна**

Оглавление

[Цель лабораторной работы 3](#_Toc88066520)

[Глава 1. Работа с матрицами 4](#_Toc88066521)

[Глава 2. Построение графиков функций 10](#_Toc88066522)

[Глава 5. Шаблоны в Excel 18](#_Toc88066523)

[Глава 6. Новое в Microsoft Excel 20](#_Toc88066524)

[Итоги лабораторной работы 22](#_Toc88066525)

[Список литературы 23](#_Toc88066526)

# Цель лабораторной работы

Изучить:

1. Принцип работы нейронной сети Хопфилда.
2. Реализовать его.

# Глава 1. Принцип нейронной сети Хопфилда

Нейро́нная сеть Хо́пфилда (англ. Hopfield network) (Рисунок 1.1) — полносвязная нейронная сеть с симметричной матрицей связей. В процессе работы динамика таких сетей сходится (конвергирует) к одному из положений равновесия. Эти положения равновесия определяются заранее в процессе обучения, они являются локальными минимумами функционала, называемого энергией сети (в простейшем случае — локальными минимумами отрицательно определённой квадратичной формы на n-мерном кубе). Такая сеть может быть использована как автоассоциативная память, как фильтр, а также для решения некоторых задач оптимизации. В отличие от многих нейронных сетей, работающих до получения ответа через определённое количество тактов, сети Хопфилда работают до достижения равновесия, когда следующее состояние сети в точности равно предыдущему: начальное состояние является входным образом, а при равновесии получают выходной образ.

Нейронная сеть Хопфилда устроена так, что её отклик на запомненные m эталонных «образов» составляют сами эти образы, а если образ немного исказить и подать на вход, он будет восстановлен и в виде отклика будет получен оригинальный образ. Таким образом, сеть Хопфилда осуществляет коррекцию ошибок и помех.

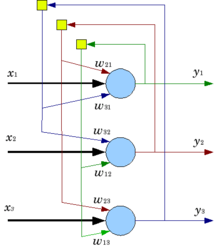


Рисунок 1.1 – Схема сети Хопфилда с тремя нейронами

# Глава 3. Реализация

Обучим сеть распознавать следующие образы:

@-----@  
@---@@-  
@--@---  
@@@----  
@--@---  
@---@@-  
@-----@

-@@@@@-  
-@---@-  
-@---@-  
-@---@-  
-@---@-  
-@---@-  
-@---@-

@@@@@@@  
---@---  
---@---  
---@---  
---@---  
---@---  
---@---

На вход будем подавать искаженный образ буквы К:

@@--@--  
@---@@-  
@--@---  
@-@----  
@--@---  
@-@-@@-  
@@----@

Алгоритм берёт случайный нейрон из матрицы и из входного вектора и пытается определить необходимость инверсии значения нейрона.

В случае, если алгоритм 200 раз подряд не смог найти нейрон для инверсии, то сеть принимает решение прервать поиск и сообщает, что исходный образ определить не удалось.

Результат выполнения программы:

Result found in 129 iterations

Result:

@-----@  
@---@@-  
@--@---  
@@@----  
@--@---  
@---@@-  
@-----@

# Глава 4. Листинг программы

* Метод обучения нейронной сети
* public void trainNetwork(Shape shape) {  
   shapes.add(shape);  
   Integer[] X = shape.toIntegers();  
    
   for (int i = 0; i < n; i++) {  
   for (int j = 0; j < n; j++) {  
   if (i == j) {  
   W[i][j] = 0;  
   } else {  
   W[i][j] += X[i] \* X[j];  
   }  
   }  
   }  
  }
* Метод распознавания образа
* public Shape recognize(Shape shape) {  
   System.out.println("Known shapes: ");  
   for (Shape knownShape : shapes) {  
   System.out.println(knownShape);  
   }  
    
   System.out.println("Input shape: ");  
   System.out.println(shape);  
    
   Integer[] Y = shape.toIntegers();  
   int maxTries = 200;  
   int errs = 0;  
   int iters = 0;  
   Shape modifiedShape = shape;  
    
   while (!shapes.contains(modifiedShape)) {  
   iters++;  
   boolean success = recstep(Y);  
   modifiedShape = new Shape(Y);  
   System.out.println("Modified shape: ");  
   System.out.println(modifiedShape);  
   if (success) {  
   errs = 0;  
   } else {  
   errs++;  
   }  
   if (errs >= maxTries) {  
   System.out.printf("Cannot recognize shape in %d iterations%n", iters);  
   return modifiedShape;  
   }  
   }  
    
   System.out.printf("Result found in %d iterations%n", iters);  
   return modifiedShape;  
  }
* Шаг распознавания образа
* private boolean recstep(Integer[] Y) {  
   int r = (int) ((Math.random() \* (Y.length)));  
   int net = 0;  
   for (int i = 0; i < Y.length; i++) {  
   net += Y[i] \* W[i][r];  
   }  
   System.out.println("net is " + net);  
   int signet = Integer.compare(net, 0);  
   System.out.println("signet is " + signet);  
   System.out.println("Y[r] is " + Y[r]);  
   if (signet != Y[r]) {  
   System.out.printf("Neuron %d: %d -> %d%n", r, Y[r], signet);  
   Y[r] = signet;  
    
   return true;  
   } else {  
   return false;  
   }  
  }

# Итоги лабораторной работы

Мы изучили и научились реализовывать нейронные сети Хопфилда.

# Список литературы

1. [[bitsofmind.org](https://bitsofmind.wordpress.com/2008/08/05/hopfield_net/)] – Нейронная сеть Хопфилда