****

**Московский авиационный институт**

**(Национальный исследовательский университет)**

**Институт № 3**

**Кафедра 311**

**Интеллектуальные системы**

**Лабораторная работа № 2**

**«Сигнальный метод обучения Хебба»**

**Выполнил студент  
Веденеев Максим Кириллович**

**Группа М3З-501-БК**

**Дата 06.10.2023 г.**

**Принял преподаватель  
Кос Оксана Игоревна**

Оглавление

[Цель лабораторной работы 3](#_Toc105775365)

[Глава 1. Алгоритм обучения Хебба 4](#_Toc105775366)

[Глава 2. Метод сигнального обучения Хебба 5](#_Toc105775367)

[Глава 3. Реализация 6](#_Toc105775368)

[Глава 4. Код программы 7](#_Toc105775369)

[Итоги лабораторной работы 9](#_Toc105775370)

[Список литературы 10](#_Toc105775371)

# Цель лабораторной работы

Изучить:

1. Принцип работы сигнального метода обучения Хебба.
2. Реализовать его.

# Глава 1. 1. Принцип работы сигнального метода обучения Хебба

Сигнальный метод обучения Хебба – это один из ключевых принципов обучения нейронных сетей, предложенный Дональдом Хеббом в 1949 году. Он основан на идее усиления связей между нейронами, которые активируются одновременно.

Принцип работы Сигнального метода обучения Хебба можно разделить на несколько этапов:

**Инициализация:** Начальные веса связей между нейронами устанавливаются случайным образом или другими способами в зависимости от конкретной задачи.

**Представление данных:** Нейронная сеть принимает на вход данные, которые могут быть представлены в виде векторов или матриц. Каждый элемент данных соответствует активации нейрона во входном слое.

**Распространение сигнала:** Сигнал проходит через нейроны сети от входного слоя к выходному. Каждый нейрон вычисляет свою активацию на основе входных данных и текущих весов связей.

**Обновление весов:** Важная часть метода Хебба заключается в обновлении весов связей между нейронами. Если два нейрона активируются одновременно и совместно способствуют активации друг друга, то вес связи между ними увеличивается. Если активация одного нейрона способствует ингибированию другого, то вес связи уменьшается. Формула для обновления весов может быть представлена как:

ΔW\_ij = η \* x\_i \* y\_j

Где:

ΔW\_ij - изменение веса связи между нейронами i и j.

η (это коэффициент обучения) - параметр, который регулирует скорость обучения.

x\_i - активация нейрона i.

y\_j - активация нейрона j.

Этот процесс повторяется для множества входных данных многократно. Со временем веса связей могут изменяться так, что сеть будет более эффективно реагировать на конкретные шаблоны в данных.

Завершение обучения: Обучение может завершиться после фиксированного числа итераций или при достижении желаемой производительности на задаче.

Сигнальный метод обучения Хебба прост в реализации и может использоваться для обучения нейронных сетей на простых задачах. Однако у него есть ограничения, и он может быть неэффективным или даже неустойчивым на более сложных задачах. В более современных нейронных сетях обычно используются более сложные алгоритмы обучения, такие как обратное распространение ошибки (backpropagation), которые позволяют эффективнее обучать глубокие и сложные модели.

# Глава 3. Реализация

Пример реализован на двухслойной нейронной сети с тремя входными нейронами и одним выходным. Схема представлена на рисунке 3.1.

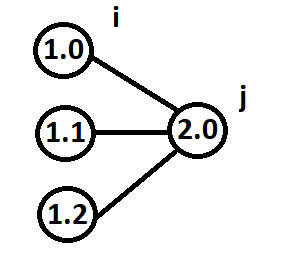


Рисунок 3.1 – Схема нейронной сети

Сеть будет обучаться распознавать образ {0, 1, 1}. Количество итераций обучения: 1000. Коэффициент обучения: 0.3.

Ожидается увеличение весов синапсов между нейронами 1.1 – 2.0 и 1.2 – 2.0.

Полученный результат после выполнения обучения:

Synapse{neuronI=1.0, neuronJ=2.0, wIJ=0.19546801745222864}

Synapse{neuronI=1.1, neuronJ=2.0, wIJ=490.49718474784527}

Synapse{neuronI=1.2, neuronJ=2.0, wIJ=314.97430498576955}

Попытаемся с помощью сети определить соответствие противоположного образа ({1, 0, 0}) и изначального образа ({0, 1, 1}). Допустимым расхождением в результатах активности выходного нейрона примем 10%.

Получим результат:

Actual result: 0.012302886200383629

Estimating result: 157.37855371666797

Input image {[1, 0, 0]} are different with origin image {[0, 1, 1]}

# Глава 4. Код программы

* Метод обучения нейронной сети

public void train(int[] input) {  
 System.out.println("Training neural network for image: " + Arrays.toString(input));  
 setInput(input);  
 for (int i = 1; i <= 1000; i++) {  
 System.out.printf("====== ITERATION #%d ======%n", i);  
 iterate();  
 updateSynapses();  
 if (i == 1000) {  
 trainingResult = neuralNetwork.peekLast().get(0).out;  
 }  
 resetAllNeurons();  
 }  
}

* Метод обновления весов синапсов

private void updateSynapses() {  
 // Обновить вес синапсов  
 System.out.println("====== UPDATE SYNAPSES WEIGHT ======");  
 for (Synapse synapse : synapses) {  
 synapse.wIJ += a \* synapse.neuronI.out \* synapse.neuronJ.out;  
 System.out.printf("====== CURRENT SYNAPSE (i): %s ======%n", synapse);  
 }  
}

* Метод итерации обучения

public void iterate() {  
 LinkedList<List<Neuron>> localNeuralNetwork = new LinkedList<>(neuralNetwork);  
 List<Neuron> layer = localNeuralNetwork.poll();  
 List<Neuron> nextLayer = localNeuralNetwork.poll();  
 // Вычислить выходы слоя i, поставить их сумму на вход слою j  
 while (nextLayer != null && layer != null) {  
 for (Neuron neuronI : layer) {  
 neuronI.out = 0; // Обнулить выход нейрона  
 neuronI.getResult(); // Вычислить выход нейрона  
  
 List<Synapse> neuronSynapses = synapses.stream() // Найти синапсы нейрона  
 .filter(s -> s.neuronI.equals(neuronI))  
 .collect(Collectors.toList());  
  
 for (Synapse neuronSynapse : neuronSynapses) {  
 Neuron neuronJ = neuronSynapse.neuronJ;  
 neuronJ.in = neuronI.out; // Поставляем выход i на вход j  
 neuronJ.w = neuronSynapse.wIJ; // Ставим вес нейрону j = весу синапса  
 double out = neuronJ.out; // Сохраняем имеющийся выход нейрона j  
 out += neuronJ.getResult(); // Вычисляем выход нейрона j  
 neuronJ.out = out; // Обновляем значение выхода нейрона j  
 }  
 }  
 layer = nextLayer;  
 nextLayer = localNeuralNetwork.poll();  
 }  
}

* Метод получения результата обработки входного изображения

public double getResult(int[] input) {  
 setInput(input);  
 iterate();  
 System.out.println("Calculating result for: " + Arrays.toString(input));  
 Neuron neuron = neuralNetwork.peekLast().get(0);  
 System.out.println("Result neuron is: " + neuron);  
  
 return neuron.out;  
}

* Метод инициализации нейронной сети

private LinkedList<List<Neuron>> initNeuralNetwork() {  
 List<Neuron> i = new LinkedList<>();  
 for (int idx = 0; idx < 3; idx++) { // Входной слой из 3 нейронов  
 i.add(new Neuron("1." + idx));  
 }  
  
 List<Neuron> j = new LinkedList<>();  
 for (int idx = 0; idx < 1; idx++) { // Промежуточный слой из 2 нейронов  
 j.add(new Neuron("2." + idx));  
 }  
 i.forEach(neuron -> neuron.createSynapsesWithNextLayer(j, synapses)); // Создаём синапсы между нейронами 1 и 2 слоя  
  
 return new LinkedList<List<Neuron>>() {{  
 add(i);  
 add(j);  
 }};  
}

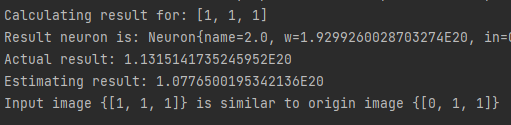


Рисунок 4.1. Результат выполнения программы

# Итоги лабораторной работы

Мы изучили и научились реализовывать обучение нейронных сетей без учителя. (Алгоритм + характерные черты)

# Список литературы

1. [[pandia.org](https://pandia.ru/text/79/352/46535.php)] – Обучение без учителя. Сигнальный метод Хебба (аналогично