****

**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](#__RefHeading___1)

[Глава 1. 1. Принцип работы сигнального метода обучения Хебба 4](#__RefHeading___2)

[Глава 3. Реализация 6](#__RefHeading___3)

[Глава 4. Код программы 7](#__RefHeading___4)

[Итоги лабораторной работы 12](#__RefHeading___5)

[Список литературы 13](#__RefHeading___6)

# 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 трехслойной нейронной сети с тремя входными нейронами, двумя скрытыми нейронами и одним выходным.

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:

Weights are:

neuron I = 1.0, neuronJ=2.0, wIJ=0.09892421377649323

neuron I = 1.0, neuronJ=2.1, wIJ=0.05556480721739849

neuron I = 1.1, neuronJ=2.0, wIJ=4.019884019404537

neuron I = 1.1, neuronJ=2.1, wIJ=4.650410297704985

neuron I = 1.2, neuronJ=2.0, wIJ=3.1459842126538553

neuron I = 1.2, neuronJ=2.1, wIJ=3.571201704539695

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: 0.0016922557490675662

Estimating result: 0.02017613226608715

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

# Глава 4. Код программы

}import random

import numpy as np

class Hebb:

def \_\_init\_\_(self):

self.weights = []

self.learning\_coff = 0.3

self.training\_result = 0 # Ожидаемая активность выходного нейрона при полном соответствии образа

self.neural\_network = self.init\_neural\_network()

def init\_neural\_network(self):

i = [Neuron(f"1.{idx}") for idx in range(3)] # Входной слой из 3 нейронов

j = [Neuron(f"2.{idx}") for idx in range(2)] # Промежуточный слой из 2 нейронов

for neuron\_i in i:

neuron\_i.fill\_weights(j, self.weights) # Установление между ними весов

return [i, j]

def set\_input(self, entry):

for neuron\_id, neuron in enumerate(self.neural\_network[0]):

neuron.set\_in(entry[neuron\_id]) # Определяем вход нейрона

def train(self, input\_data):

print(f"Training neural network for image: {input\_data}")

self.set\_input(input\_data)

epochs = 1000

for i in range(epochs):

if i % 100 == 0:

print(f"Epoch №{i+1}")

print(f"Weights are:")

for weight in self.weights:

print(f"{weight}")

self.iterate()

self.update\_weights()

if i == 999:

self.training\_result = self.neural\_network[-1][0].out

self.reset\_all\_neurons()

# Обнуляем активность всех нейронов

def reset\_all\_neurons(self):

for layer in self.neural\_network:

for neuron in layer:

neuron.out = 0

# Обновить веса

def update\_weights(self):

for weight in self.weights:

weight.w\_ij += self.learning\_coff \* weight.neuron\_i.out \* weight.neuron\_j.out

def iterate(self):

local\_neural\_network = list(self.neural\_network)

layer = local\_neural\_network.pop(0)

next\_layer = local\_neural\_network.pop(0) if local\_neural\_network else None

while next\_layer is not None and layer is not None:

for neuron\_i in layer:

neuron\_i.out = 0

neuron\_i.get\_result()

neuron\_synapses = [synapse for synapse in self.weights if synapse.neuron\_i == neuron\_i]

for neuron\_synapse in neuron\_synapses:

neuron\_j = neuron\_synapse.neuron\_j

neuron\_j.in\_value = neuron\_i.out

neuron\_j.w = neuron\_synapse.w\_ij

out = neuron\_j.out

out += neuron\_j.get\_result()

neuron\_j.out = out

layer = next\_layer

next\_layer = local\_neural\_network.pop(0) if local\_neural\_network else None

def get\_result(self, input\_data):

self.set\_input(input\_data)

self.iterate()

print(f"Calculating result for: {input\_data}")

neuron = self.neural\_network[-1][0]

print(f"Result neuron is: {neuron}")

return neuron.out

def get\_training\_result(self):

return self.training\_result

class Neuron:

def \_\_init\_\_(self, name):

self.name = name

self.w = random.uniform(0, 0.1) # Начальный вес нейрона от 0 до 0.1

self.in\_value = 0

self.out = 0

def set\_in(self, in\_value):

self.in\_value = in\_value

def fill\_weights(self, next\_layer, weights):

for next\_neuron in next\_layer:

weights.append(Weight(self, next\_neuron, (random.uniform(0.1, 0.4) \* 0.3)))

def get\_result(self):

self.out = self.in\_value \* self.w

return self.out

class Weight:

def \_\_init\_\_(self, neuron\_i, neuron\_j, w\_ij):

self.neuron\_i = neuron\_i

self.neuron\_j = neuron\_j

self.w\_ij = w\_ij

def \_\_str\_\_(self):

return f"neuron I = {self.neuron\_i.name}, neuronJ={self.neuron\_j.name}, wIJ={self.w\_ij}"

if \_\_name\_\_ == "\_\_main\_\_":

hebb = Hebb()

origin\_image = [0, 1, 1]

hebb.train(origin\_image)

training\_result = hebb.get\_training\_result()

input\_image = [1, 0, 0]

result = hebb.get\_result(input\_image)

eps = abs(training\_result \* 0.1)

print(f"Actual result: {result}")

print(f"Estimating result: {training\_result}")

if abs(training\_result - result) <= eps:

print(f"Input image {input\_image} is similar to origin image {origin\_image}")

else:

print(f"Input image {input\_image} are different with origin image {origin\_image}")

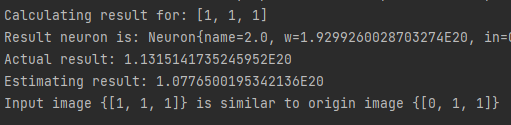


Рисунок 4.1. Результат выполнения программы

# 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