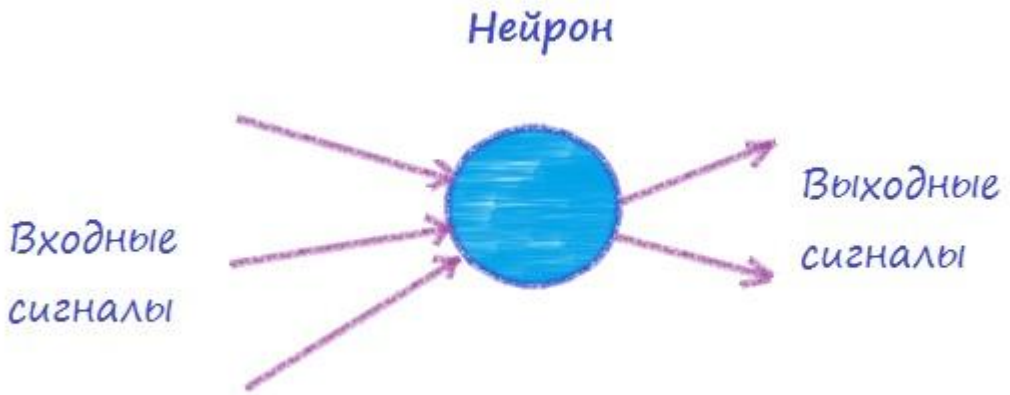
Нейронные сети

нейронная сеть — сеть представляет из себя совокупность нейронов, соединенных друг с другом определенным образом.

Нейрон

Нейрон представляет из себя элемент, который вычисляет выходной сигнал (по определенному правилу) из совокупности входных сигналов. То есть основная последовательность действий одного нейрона такая:

- Прием сигналов от предыдущих элементов сети
- Комбинирование входных сигналов
- Вычисление выходного сигнала
- Передача выходного сигнала следующим элементам нейронной сети



Связь нейронов

Выходной сигнал нейрона 1 равен 5. Вес связи между нейронами равен 2. Таким образом, чтобы определить входной сигнал нейрона 2, приходящий от нейрона 1, необходимо умножить значение этого сигнала на вес связи (5*2).

Связь нейронов

если сигналов много - они все суммируются.

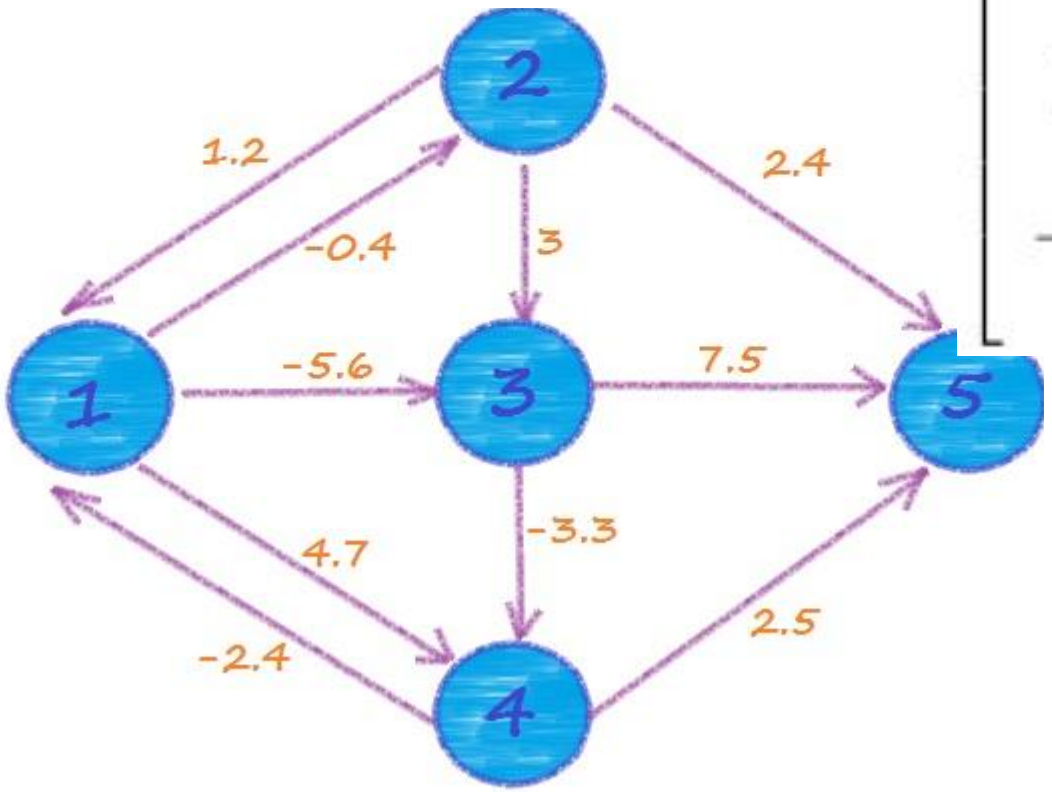
В итоге на входе нейрона мы получаем следующее:

$$net_j = \sum_{i=1}^{N} x_i * w_{ij}$$

В этой формуле:
netj - это результат комбинирования всех входных сигналов для нейрона (комбинированный ввод нейрона)
N — количество элементов, передающих свои выходные сигналы на вход сигнала j.
wij — вес связи, соединяющей нейрон i с нейроном j.
Суммируя все взвешенные входные сигналы, мы получаем комбинированный ввод элемента сети.

W – весовая матрица

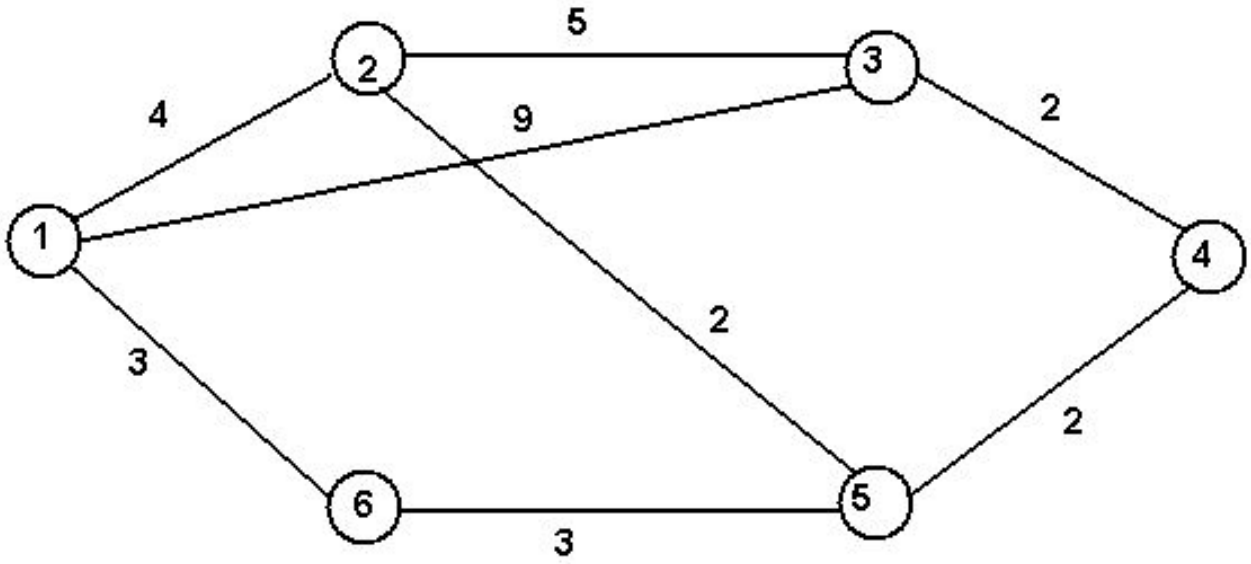
Чаще всего структура связей между нейронами представляется в виде матрицы W , которую называют весовой матрицей. Элемент матрицы , как и в формуле, определяет вес связи, идущей от элемента i к элементу j



$$\begin{bmatrix} 0 & -0.4 & -5.6 & 4.7 & 0 \\ 1.2 & 0 & 3 & 0 & 2.4 \\ 0 & 0 & 0 & -3.3 & 7.5 \\ -2.4 & 0 & 0 & 0 & 2.5 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

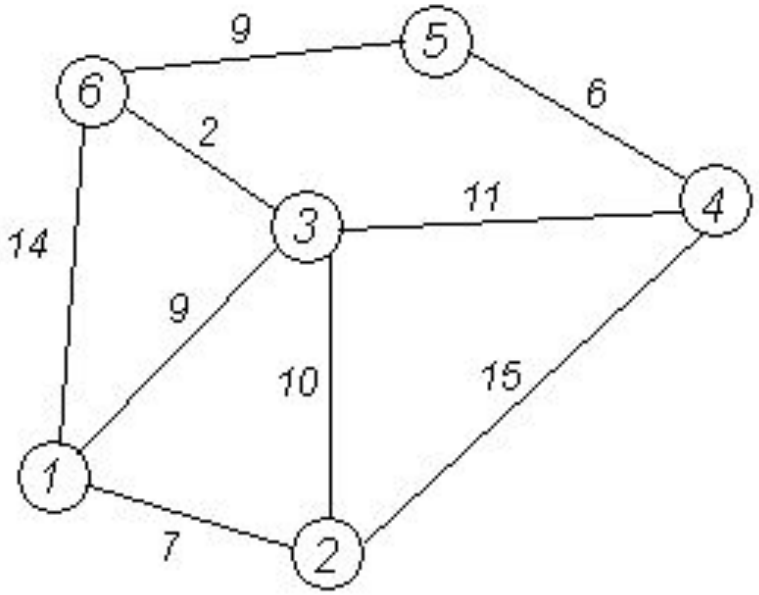
Задание 1

Напишите весовую матрицу для предложенных графов



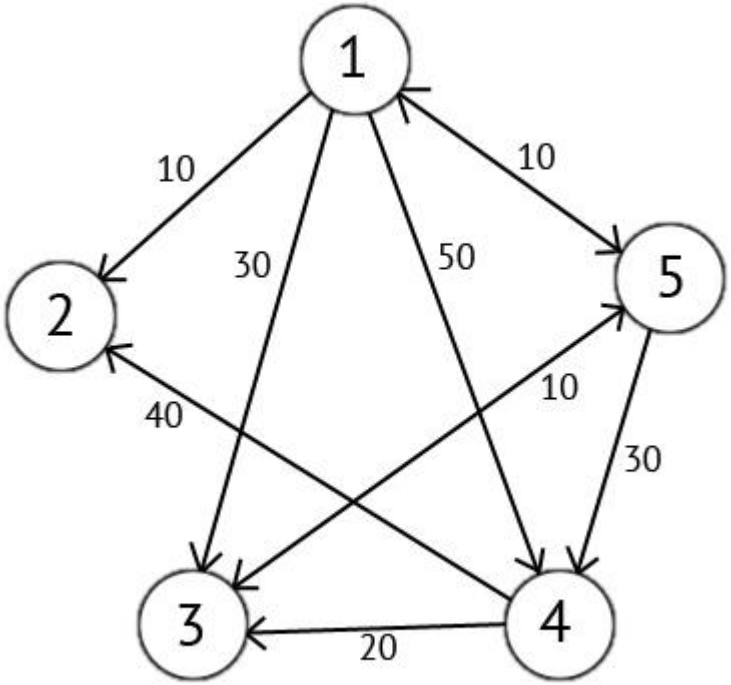
Задание 1

Напишите весовую матрицу для предложенных графов



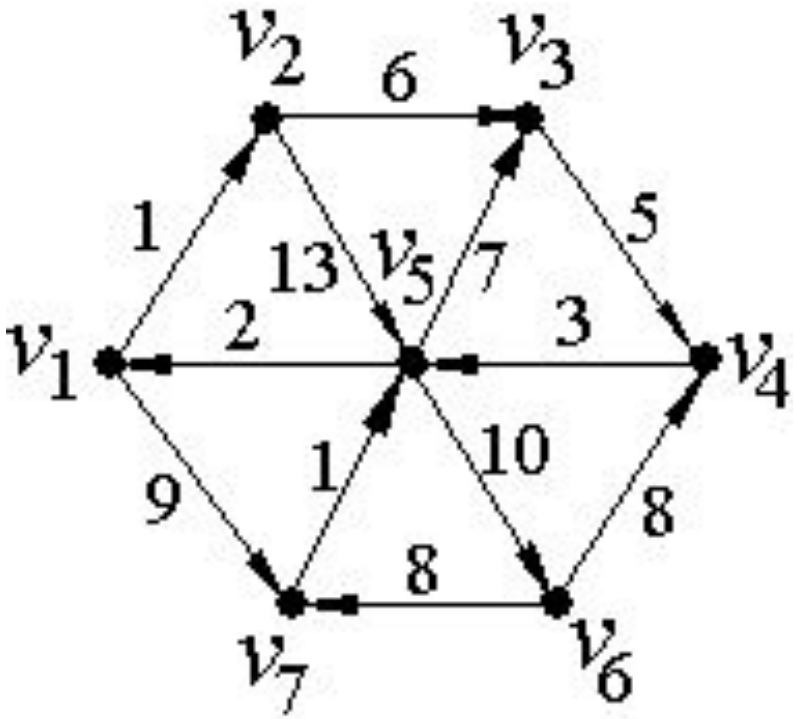
Задание 1

Напишите весовую матрицу для предложенных графов



Задание 1

Напишите весовую матрицу для предложенных графов



Задание 2

Восстановите граф, зная матрицу весов

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	1	1	0	1	0
x_2	0	1	0	0	1	0
x_3	0	0	0	0	0	0
x_4	0	0	1	0	0	0
x_5	0	0	0	1	0	0
x_6	1	0	0	0	1	1

Задание 2

Восстановите граф, зная матрицу весов

	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

Задание 2

Восстановите граф, зная матрицу весов

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_1	0	1	0	0	1	0	0
x_2	0	1	0	1	0	0	0
x_3	0	0	0	1	0	0	0
x_4	0	0	0	0	1	0	0
x_5	0	0	0	0	1	0	0
x_6	0	0	1	0	0	0	1
x_7	0	0	0	1	0	1	0

Задание 2

Восстановите граф, зная матрицу весов

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

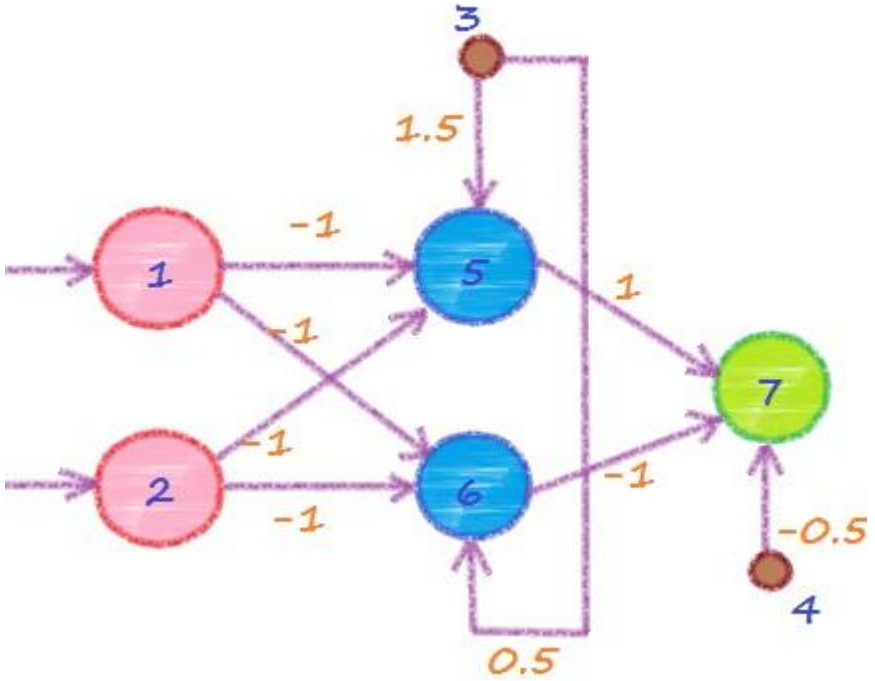
Функция активности элемента (активационная функция нейрона)

Для каждого элемента сети имеется определенное правило, в соответствии с которым из значения комбинированного ввода элемента вычисляется его выходное значение. Это правило называется функцией активности. А само выходное значение называется активностью нейрона. В роли функций активности могут выступать абсолютно любые математические функции, приведу в качестве примера несколько из наиболее часто использующихся:

- пороговая функция — если значение комбинированного ввода ниже определенного значения (порога), то активность равна нулю, если выше — единице.
- логистическая функция.

Пример

При помощи нейронной сети вычислить отношение XOR. То есть на вход мы будем подавать разные варианты сигналов, а на выходе должны получить результат операции XOR для поданных на вход значений



Пример

- Элементы 1 и 2 являются входными, а элемент 7 — выходным. Нейроны 5 и 6 называются скрытыми, поскольку они не связаны с внешней средой. Таким образом, мы получили три слоя — входной, скрытый и выходной. Элементы 3 и 4 называют элементами смещения. Их выходной сигнал (активность) всегда равен 1. Для вычисления комбинированного ввода в этой сети мы будем использовать правило суммирования взвешенных связей, а в качестве функции активности будет выступать пороговая функция. Если комбинированный ввод элемента меньше 0, то активность равна 0, если ввод больше 0, то активность — 1.
- Давайте подадим на вход нейрона 1 — единицу, а на вход нейрона 2 — ноль. В этом случае на выходе мы должны получить 1 (0 XOR 1 = 1). Рассчитаем выходное значение вручную для демонстрации работы сети.

Пример

Комбинированный ввод элемента 5: net_5 = 1 * (-1) + 0 * (-1) + 1 * 1.5 = 0.5.

Активность элемента 5: 1 (0.5 > 0).

Комбинированный ввод элемента 6: net_6 = 1 * (-1) + 0 * (-1) + 1 * 0.5 = -0.5.

Активность элемента 6: 0.

Комбинированный ввод элемента 7: net_7 = 1 * (1) + 0 * (-1) + 1 * (-0.5) = 0.5.

Активность элемента 7, а в то же время и выходное значение сети равно 1.

Задание 3

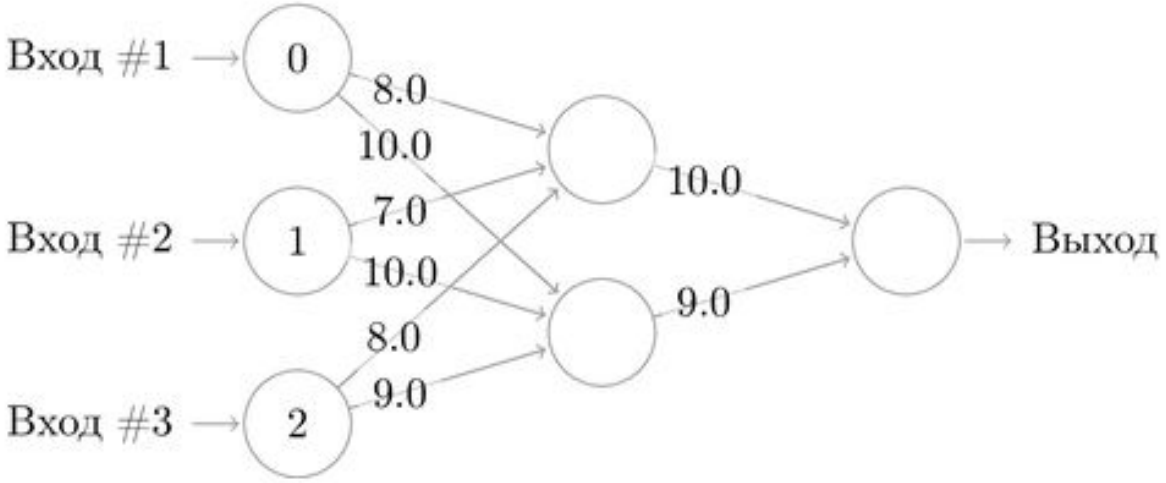
Для предложенной ранее сети, рассчитывающей значение операции XOR для остальных возможных активаций.

Особенность задания 3

- В данном случае все значения весовых коэффициентов нам были известны заранее, но главной особенностью нейронных сетей является то, что они могут сами корректировать значения веса всех связей в процессе обучения сети.

Задание 3.1

Применение сигмоидельной функции

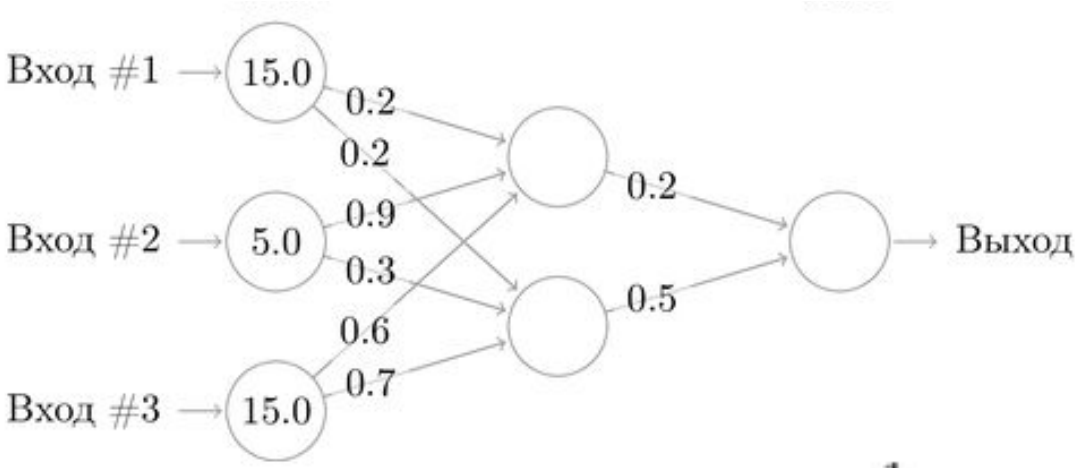


Вот так вот функция активации выглядит

$$f(x) = \frac{1}{1 + e^{-x}}$$

Задание 3.2

Применение сигмоидельной функции



Вот так вот функция активации выглядит

$$f(x) = \frac{1}{1 + e^{-x}}$$

Обучение нейронной сети

Цель обучения: корректировка весовых коэффициентов связей сети.
Одним из самых типичных способов является **управляемое обучение**.
Для его проведения нам необходимо иметь набор входных данных, а также соответствующие им выходные данные.

Обучение нейронной сети

Мы подаем на вход сети данные, после чего сеть вычисляет выходное значение. Мы сравниваем это значение с имеющимся у нас (напоминаю, что для обучения используется готовый набор входных данных, для которых выходной сигнал известен) и в соответствии с разностью между этими значениями корректируем весовые коэффициенты нейронной сети. И эта операция повторяется по кругу много раз. В итоге мы получаем обученную сеть с новыми значениями весовых коэффициентов.

Правило корректировки весов

Дельта правило (правило Видроу-
а)

$$\delta = y_0 - y$$

y0 - это ожидаемый (истинный) вывод сети
y - это реальный вывод (активность) выходного элемента

Правило корректировки весов

Дельта правило (правило Видроу-Хоффа)
Корректировка внутренних весов

$$\Delta w_{jk} = \eta * \delta_k * x_j$$

Где η — норма обучения. Это число мы сами задаем перед началом обучения. x_j — это сигнал, приходящий к элементу *k* от элемента *j*. А δ_k — ошибка элемента *k*.

Алгоритм обратного распространения ошибок

$$\delta_j = f^{'}(net_j) * \sum_k \delta_k * w_{kj}$$

Функция $f(x)$ — это функция активности элемента. Давайте использовать логистическую функцию, для нее:

$$f^{'}(net_j) = f(net_j) * (1 - f(net_j))$$

Подставляем в предыдущую формулу и получаем величину ошибки:

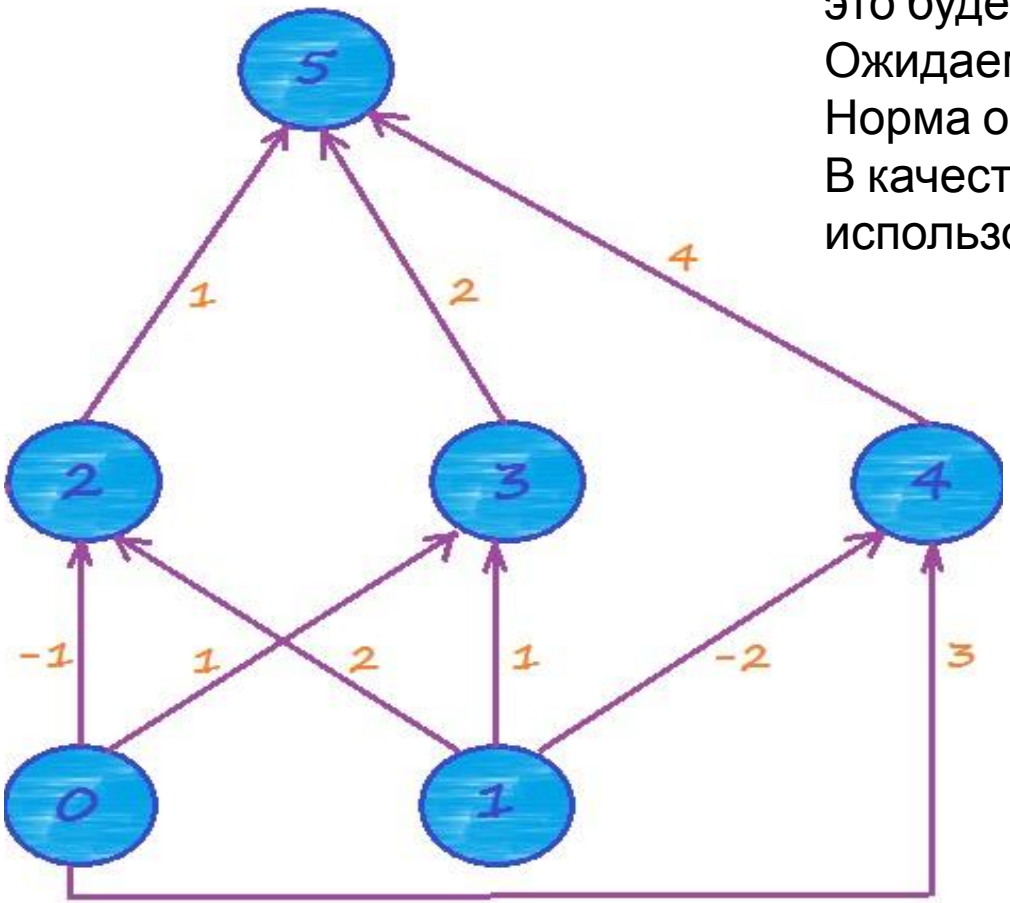
$$\delta_j = f(net_j) * (1 - f(net_j)) * \sum_k \delta_k * w_{kj}$$

В этой формуле:

- δ_j — ошибка элемента с индексом j
- k — индекс, соответствующий слою, который посылает ошибку «обратно»
- net_j — комбинированный ввод элемента
- $f(net_j)$ — активность элемента

Алгоритм обратного распространения ошибок Пример

На вход мы должны подать образец, пусть это будет *(0.2, 0.5)*.
Ожидаемый выход сети — *0.4*.
Норма обучения пусть будет равна *0.85*.
В качестве функции активности мы будем использовать логистическую функцию:



$$f(x) = \frac{1}{1 + e^{-x}}$$

Алгоритм обратного распространения ошибок Пример

Вычислим комбинированный ввод элементов *2*, *3* и *4*:

$$net_2 = (-1) * 0.2 + 2 * 0.5 = 0.8$$

$$net_3 = 1 * 0.2 + 1 * 0.5 = 0.7$$

$$net_4 = (-2) * 0.5 + 3 * 0.2 = -0.4$$

Активность этих элементов равна:

$$f(net_2) = 0.69$$

$$f(net_3) = 0.67$$

$$f(net_4) = 0.40$$

Комбинированный ввод пятого элемента:

$$net_5 = 0.69 + 0.67 * 2 + 0.4 * 4 = 3.63$$

Активность пятого элемента и в то же время вывод нейронной сети равен:

$$f(net_5) = 0.974$$

Рассчитываем корректировки

$$\delta_5 = (0.4 - 0.974)) * f(net_5) * (1 - f(net_5))$$

$$\delta_5 = (0.4 - 0.974) * 0.974 * (1 - 0.974)$$

$$\delta_5 = -0.014$$

Тогда ошибки для элементов *2*, *3* и *4* равны соответственно:

$$\delta_j = \sum_k \delta_k * w_{kj} * f(net_j) * (1 - f(net_j))$$

$$\delta_2 = -0.014 * 0.69 * (1 - 0.69) = -0.0029$$

$$\delta_3 = -0.028 * 0.67 * (1 - 0.67) = -0.0061$$

$$\delta_4 = -0.056 * 0.40 * (1 - 0.40) = -0.0134$$

Здесь значения *-0.014, -0.028* и *-0.056* получаются в результате прохода ошибки выходного элемента *—0.014* по взвешенным связям в направлении к элементам *2*, *3* и *4* соответственно.

Пересчитываем веса

- И, наконец-то, рассчитываем величину, на которую необходимо изменить значения весовых коэффициентов. Например, величина корректировки для связи между элементами *0* и *2* равна произведению величины сигнала, приходящего в элементу *2* от элемента *0*, ошибки элемента *2* и нормы обучения (все по дельта-правилу, которое мы обсудили в начале статьи):

$$\Delta w_{02} = 0.2 * -0.0029 * 0.85$$

$$\Delta w_{jk} = \eta * \delta_k * x_j$$

Где η — норма обучения. Это число мы сами задаем перед началом обучения. x_j — это сигнал, приходящий к элементу *k* от элемента *j*. А δ_k — ошибка элемента *k*.

Пересчитываем веса

Аналогичным образом производим расчеты и для остальных элементов:

$$\Delta w_{03} = 0.2 * -0.0061 * 0.85$$

$$\Delta w_{04} = 0.2 * -0.0134 * 0.85$$

$$\Delta w_{12} = 0.5 * -0.0029 * 0.85$$

$$\Delta w_{13} = 0.5 * -0.0061 * 0.85$$

$$\Delta w_{14} = 0.5 * -0.0134 * 0.85$$

$$\Delta w_{25} = 0.69 * -0.014 * 0.85$$

$$\Delta w_{35} = 0.67 * -0.014 * 0.85$$

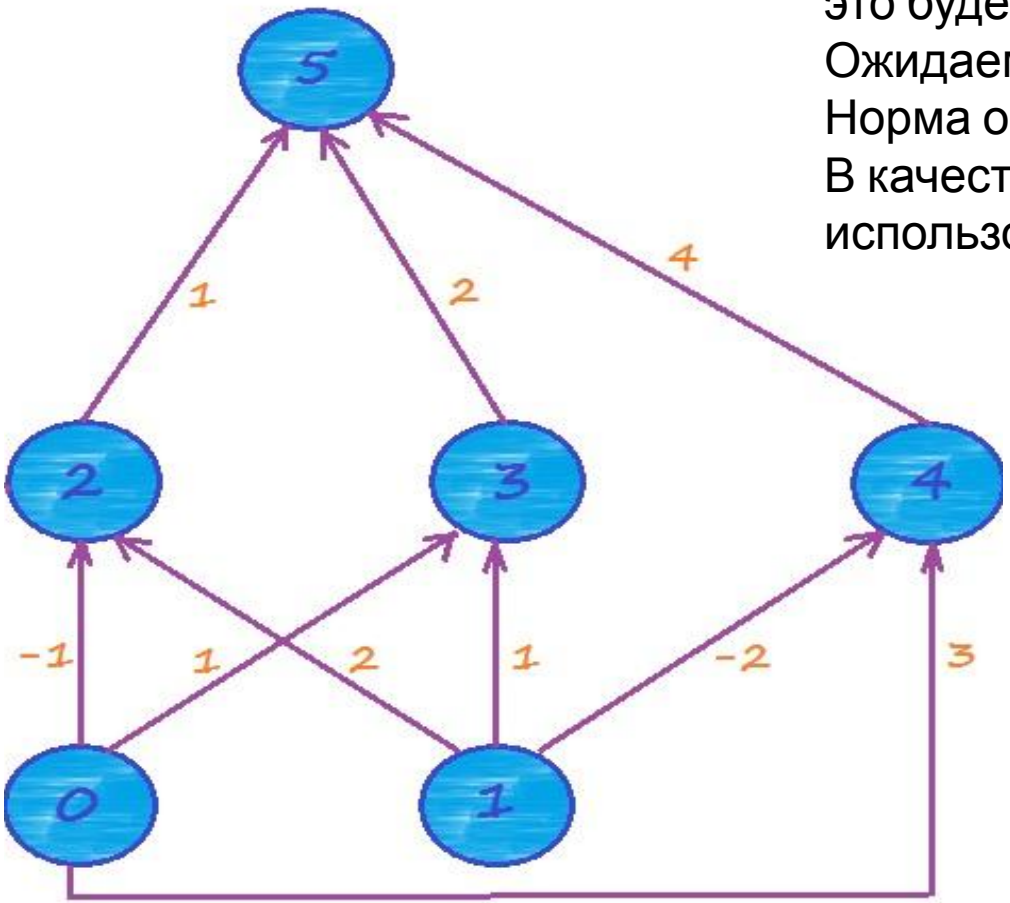
$$\Delta w_{45} = 0.40 * -0.014 * 0.85$$

Алгоритм обратного распространения ошибок Пример

На этом обратный проход по сети закончен, цель достигнута Именно так и протекает процесс обучения по алгоритму обратного распространения ошибок. Мы рассмотрели этот процесс для одного набора данных, а чтобы получить полностью обученную сеть таких наборов должно быть, конечно же, намного больше, но алгоритм при этом остается неизменным, просто повторяется по кругу много раз для разных данных

Задание 4

Применить алгоритм обратного распространения ошибок.



На вход мы должны подать образец, пусть это будет *(0.5, 0.2)*.
Ожидаемый выход сети — *0.4*.
Норма обучения пусть будет равна *0.85*.
В качестве функции активности мы будем использовать логистическую функцию:

$$f(x) = \frac{1}{1 + e^{-x}}$$