

# Тятя! Тятя! Наши сети притащили мертвеца!

Ульянкин Филипп

## Аннотация

Однажды вечером Маша читала книгу.

Благодаря символическому дифференцированию вам никогда не придется заниматься реализацией алгоритма обратного распространения вручную. Поэтому не будем тратить время на его формулировку<sup>1</sup>.

Тут ещё пара таких цитат и рофл из Гудфелоу

**Доколе можно такое терпеть?** Если ты не понимаешь как самостоятельно реализовать самую топорную версию модели, а потом ввести в неё несколько простеньких улучшений — ты не понимаешь эту модель.

Написать по-нормальному abstract!

## Содержание

1	Всего лишь функция	2
2	50 оттенков градиентного спуска	14
3	Backpropagation	14
4	Активация	16
5	Регуляризаторы	17
6	Всего лишь кубики LEGO	17
7	Итоговый тест в стел Носко	17

Все должно быть в концепте маши кому принадлежит лёрнинг, все задачи должны ложиться в историю про неё, решения должны быть органично оторваны от личности Маши.

<sup>1</sup>Франсуа Шолле, Глубокое обучение на Python, стр. 77

# 1 Всего лишь функция

Ты всего лишь машина, только имитация жизни. Робот сочинит симфонию? Робот превратит кусок холста в шедевр искусства?

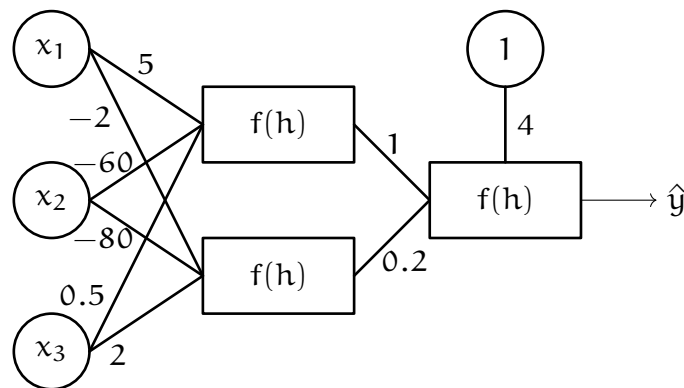
Из фильма «Я, робот» (2004)

## Упражнение 1 (от регрессии к нейросети)

Тут лёгкая задачка про то как регрессия превращается в нейронку + пункт про то что без функции активации будет то же самое на примере 2 регрессоров

## Упражнение 2 (из картинки в формулу)

Добродум хозяин кофейни на Тверской. Он хочет понять насколько сильно будет заполнена кофейня в следующие выходные. Для этого по старым данным он обучил нейросетку. На вход она принимает три фактора: температуру за окном,  $x_1$ , факт наличия на Тверской митинга,  $x_2$  и пол баристы на смене,  $x_3$ . В качестве функции активации Добродум использует ReLU.



- В эти выходные за барной стойкой стоит Агнесса. Митинга не предвидится, температура будет в районе 20 градусов. Сколько человек придёт в кофейню к Добродуму?
- На самом деле каждая нейросеть — это просто-напросто какая-то нелинейная сложная функция. Запишите нейросеть Добродума в виде функции.

### Решение:

Будем постепенно идти по сетке и делать вычисления. Подаём все значения в первый нейрон, получаем:

$$h_1 = \max(0, 5 \cdot 20 + (-60) \cdot 0 + 0.5 \cdot 1) = \max(0, 100.5) = 100.5$$

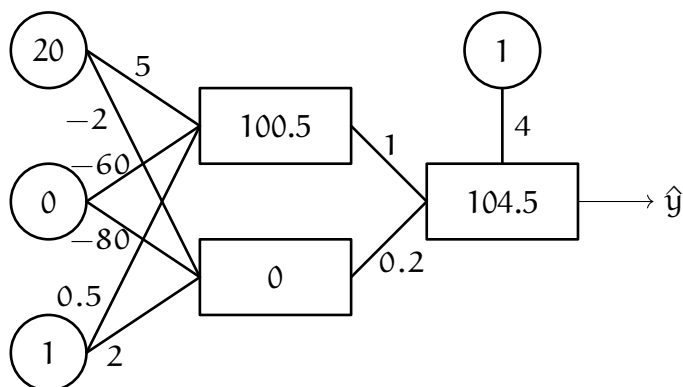
Ровно то же самое делаем со вторым нейроном:

$$h_2 = \max(0, -2 \cdot 20 + (-80) \cdot 0 + 2 \cdot 1) = \max(0, -38) = 0$$

Дальше результат скрытых нейронов идёт во второй слой:

$$\hat{y} = \max(0, 1 \cdot 100.5 + 0.2 \cdot 0 + 4 \cdot 1) = 104.5$$

Это и есть итоговый прогноз.



Теперь по мотивам наших вычислений запишем нейронку как функцию. Начинать будем с конца:

$$\hat{y} = f(1 \cdot h_1 + 0.2 \cdot h_2 + 4 \cdot 1)$$

Подставляем вместо  $h_1$  и  $h_2$  вычисления, которые происходят на первом слое нейронки:

$$\begin{aligned} \hat{y} &= f(1 \cdot f(5 \cdot x_1 - 60 \cdot x_2 + 0.5 \cdot x_3) + 0.2 \cdot f(-2 \cdot x_1 - 80 \cdot x_2 + 2 \cdot x_3) + 4 \cdot 1) = \\ &= \max(0, \max(0, 5 \cdot x_1 - 60 \cdot x_2 + 0.5 \cdot x_3) + 0.2 \cdot \max(0, -2 \cdot x_1 - 80 \cdot x_2 + 2 \cdot x_3) + 4). \end{aligned}$$

Обучение нейронной сетки на самом деле эквивалентно обучению такой сложной нелинейной функции.

### Упражнение 3 (из формулы в картинку)

Теперь в обратную сторону. Пусть у нас есть вот такая функция.

$$y = \max(0, 4 \cdot \max(0, 3 \cdot x_1 + 4 \cdot x_2 + 1) + 2 \cdot \max(0, 3 \cdot x_1 + 2 \cdot x_2 + 7) + 6)$$

Нарисуйте эту функцию в виде нейросетки.

## Решение:

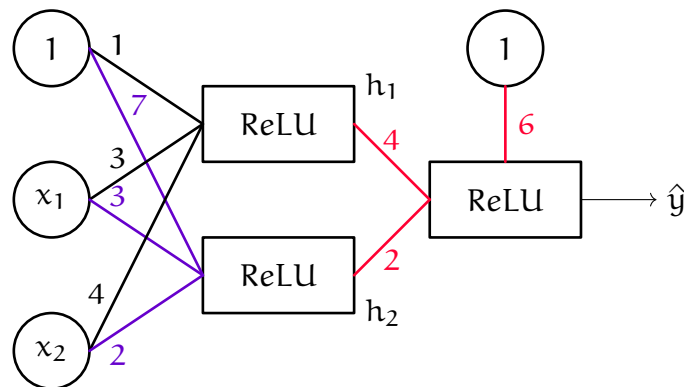
Начнём рисовать картинку с конца. На выход выплёвывается либо 0, либо комбинация из двух входов:

$$\hat{y} = \text{ReLU}(4 \cdot h_1 + 2 \cdot h_2 + 6)$$

Каждый из входов — это снова либо 0, либо комбинация из двух входов.

$$y = \max(0, 4 \cdot \underbrace{\max(0, 3 \cdot x_1 + 4 \cdot x_2 + 1)}_{h_1} + 2 \cdot \underbrace{\max(0, 3 \cdot x_1 + 2 \cdot x_2 + 7)}_{h_2} + 6)$$

Получается, что на первом слое находится два нейрона, которые передают свои выходы в третий:

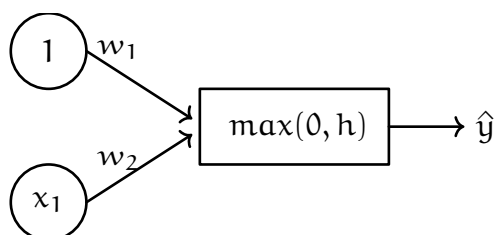


## Упражнение 4 (армия регрессий)

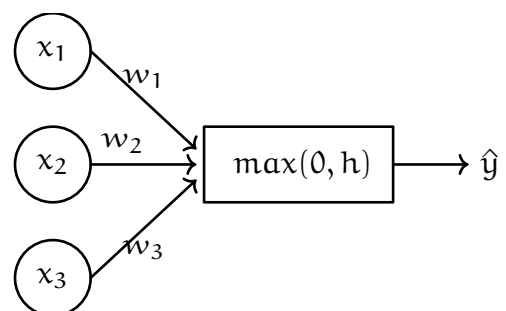
Парни очень любят Машу, а Маша с недавних пор любит собирать персептроны и думать по вечерам об их весах и функциях активации. Сегодня она решила разобрать свои залежи из персептронов и как следует упорядочить их.

- а. В ящике стола Маша нашла персептрон с картинки 1 Маша хочет подобрать веса так, чтобы он реализовывал логическое отрицание, то есть превращал  $x_1 = 0$  в  $y = 1$ , а  $x_1 = 1$  в  $y = 0$ .

Картинка 1



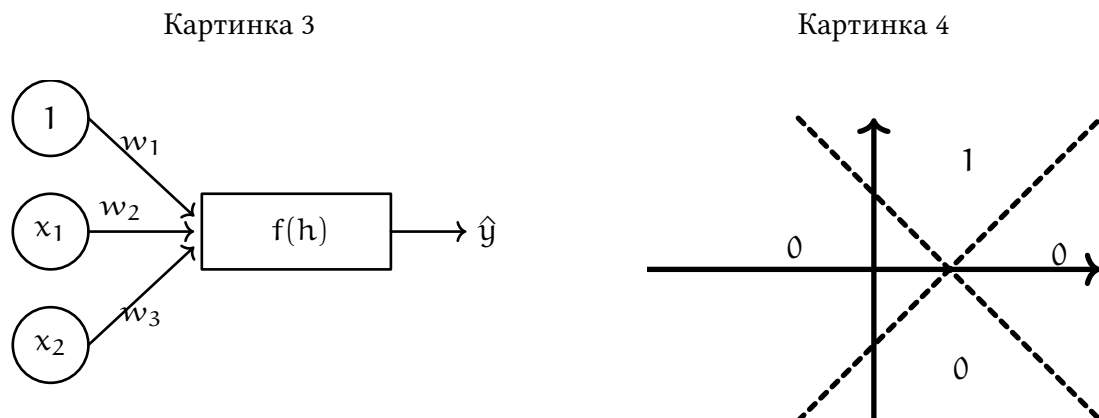
Картинка 2



- б. В тумбочке среди носков Маша нашла персептрон, с картинки 2, Маша хочет подобрать такие веса  $w_i$ , чтобы персептрон превращал  $x$  из таблички в соответствующие  $y$ :

$x_1$	$x_2$	$x_3$	$y$
1	1	2	0.5
1	-1	1	0

- в. Оказывается, что в ванной всё это время валялась куча персептронов с картинки 3 с неизвестной функцией активации (надо самому выбирать).



Маша провела на плоскости две прямые:  $x_1 + x_2 = 1$  и  $x_1 - x_2 = 1$ . Она хочет собрать из персептронов нейросетку, которая будет классифицировать объекты с плоскости так, как показано на картинке 4.

- г. В коробке на кухне заваялось три персептрона, у каждого два входа с константой и пороговая функция активации. Маша хочет реализовать с их помощью функцию

$$y = \begin{cases} 1, & \text{если } x_2 \geq |x_1 - 3| + 2; \\ 0, & \text{иначе} \end{cases}.$$

## Решение:

- а. Начнём с первого пункта. Чтобы было легче запишем нейрон в виде уравнения:

$$\hat{y} = \max(0, w_1 + w_2 \cdot x_1).$$

Нам нужно, чтобы

$$\max(0, w_1 + w_2 \cdot 1) = 0$$

$$\max(0, w_1 + w_2 \cdot 0) = 1$$

Из второго уравнения сразу получаем, что  $w_1 = 1$ , а  $w_2$  на второе уравнение никак не влияет. Для того, чтобы в первом уравнении получить ноль, нужно взять  $w_2 \leq -1$ . Нейрон готов.

б. Снова выписываем несколько уравнений:

$$\max(0, w_1 + w_2 + 2 \cdot w_3) = 0.5$$

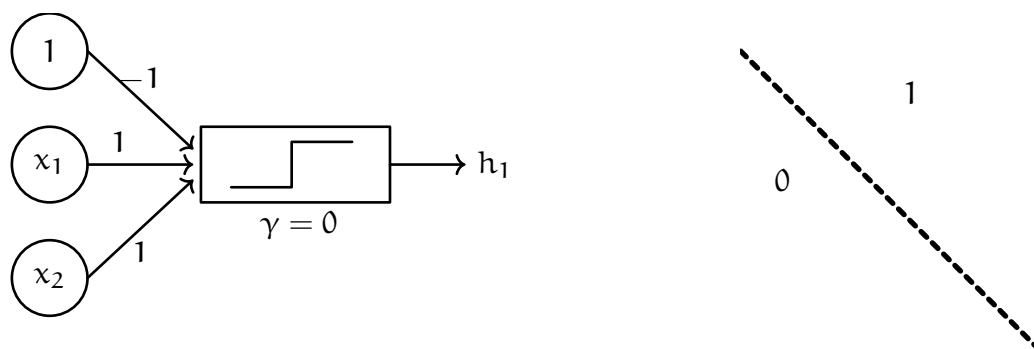
$$\max(0, w_1 - w_2 + w_3) = 0$$

Тут решений может быть довольно много. Первое, что приходит в голову — это занулить  $w_1$  и  $w_3$  в первом уравнении, а  $w_2$  поставить 0.5. Тогда во втором уравнении мы сразу же будем оказываться в отрицательной области и ReLU заботливо будет отдавать нам 0.

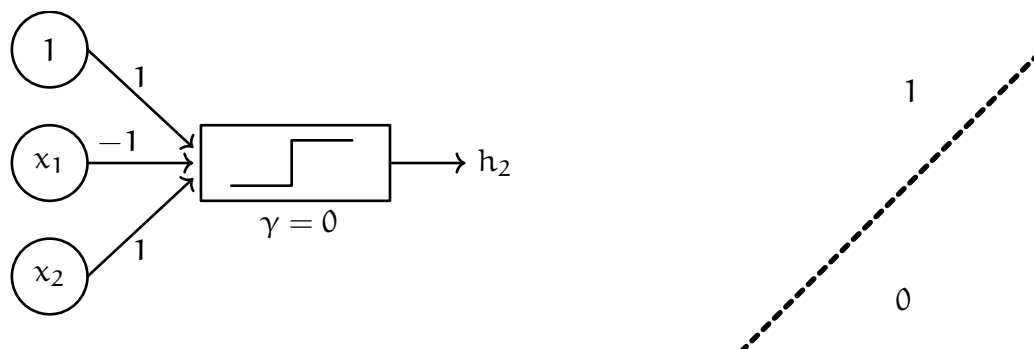
в. Давайте для разнообразия возьмём в качестве  $f(h)$  пороговую функцию потерь

$$f(h) = \begin{cases} 1, h > 0 \\ 0, h \leq 0 \end{cases}.$$

Один нейрон — это одна линия, проведённая на плоскости. Эта линия отделяет один класс от другого. Например, линию  $x_1 + x_2 - 1 = 0$  мог бы описать нейрон

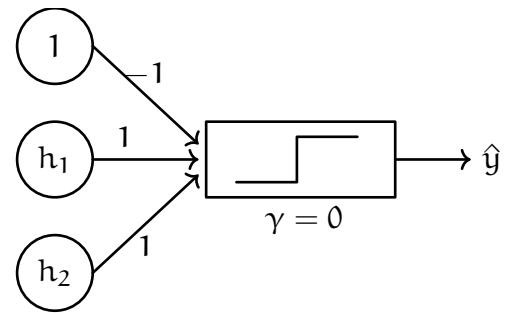
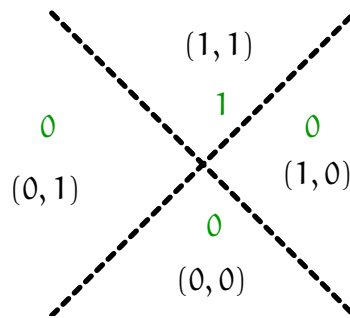


Порог  $\gamma$  для кусочной функции в каком-то смысле дублирует константу. Будем всегда брать его нулевым. Видим, что если мы получили комбинацию  $x_1$ ,  $x_2$  и 1, большую, чем ноль, мы оказались справа от прямой. Если хочется поменять метку 0 и 1 сторонами, можно умножить все коэффициенты на  $-1$ . **Наш перцептрон понимает по какую сторону от прямой мы оказались**, то есть задаёт одну линейную разделяющую поверхность. По аналогии для второй прямой мы можем получить:

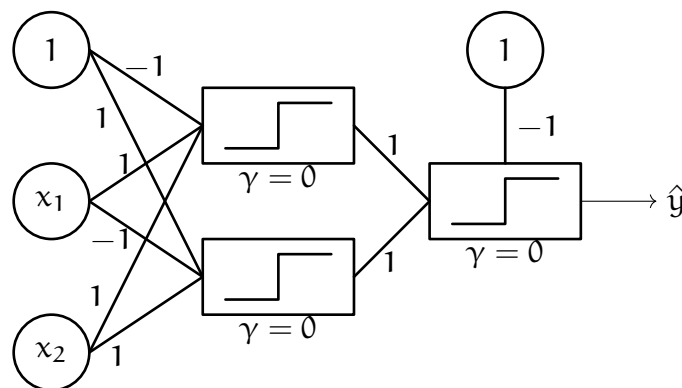


Итак, первый перцептрон выбрал нам позицию относительно первой прямой, второй относительно второй. Остаётся только соединить эти результаты в один. Нейрон для скрепки должен реализовать для нас логическую функцию, которую задаёт табличка ниже. Там же нарисованы примеры весов, которые могли бы объединить выхлоп первого слоя в итоговый прогноз.

$h_1$	$h_2$	$\hat{y}$
1	1	1
1	0	0
0	1	0
0	0	0



Теперь мы можем нарисовать итоговую нейронную сеть, решающую задачу Маши. Она состоит из двух слоёв. Меньше не выйдет, так как каждый персептрон строит только одну разделяющую линию.



Кстати говоря, если бы мы ввели для нашей нейросетки дополнительный признак  $x_1 \cdot x_2$ , у нас бы получилось обойтись только одним персептроном. В нашей ситуации **нейросетка сама сварила на первом слое признак  $x_1 \cdot x_2$ , которого ей не хватало.**

г. Вбить решение

## Упражнение 5 (логические функции)

Есть теорема, которая говорит о том, что с помощью нейросетки можно аппроксимировать почти любую функцию. Попробуйте с помощью нейросеток с минимально возможным числом нейронов описать логический функции, заданные следующими таблицами истинности:

$x_1$	$x_2$	$x_1 \cap x_2$
1	1	1
1	0	0
0	1	0
0	0	0

$x_1$	$x_2$	$x_1 \cup x_2$
1	1	1
1	0	1
0	1	1
0	0	0

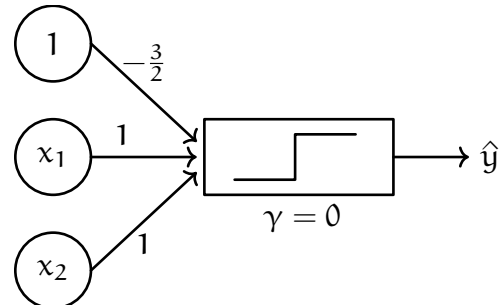
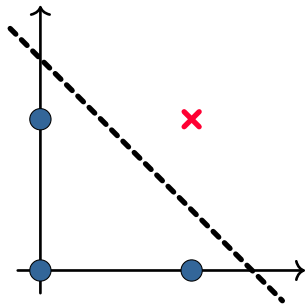
$x_1$	$x_2$	$x_1 \text{ XoR } x_2$
1	1	0
1	0	1
0	1	1
0	0	0

Первые два столбика идут на вход, третий получается на выходе. Операция из третьей таблицы называется исключающим или (XoR). Если внимательно приглядеться, то можно заметить, что XoR — это то же самое что и  $[x_1 \neq x_2]^2$ .

<sup>2</sup>Тут квадратные скобки обозначают индикатор. Он выдаёт 1, если внутри него стоит правда и 0, если ложь.

## Решение:

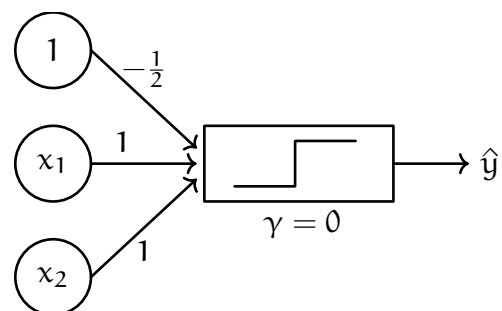
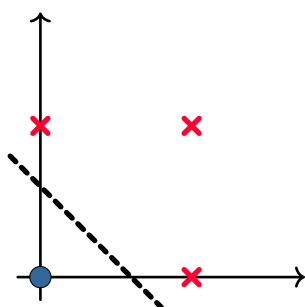
На самом деле в предыдущем упражнении мы уже построили нейрон для пересечения, когда нам нужно было оказаться два раза по правильную сторону прямой. Посмотрим на тот же нейрон под другим углом:



Если нарисовать все наши четыре точки на плоскости, становится ясно, что мы хотим отделить точку  $(1, 1)$  от всех остальных. Сделать это можно практически любой линией. Например, в нейроне выше задана линия  $x_2 = 1.5 - x_1$ . Подойдёт и любая другая, отделяющая крест от точек. Пропустим ради приличия точки через наш нейрон:

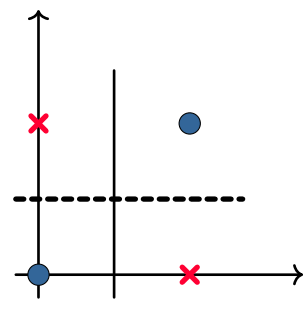
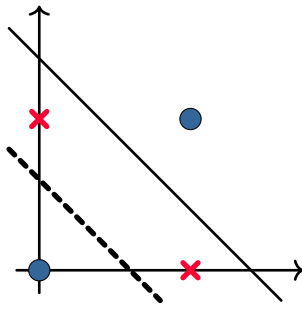
$$\begin{aligned}[-1.5 + 1 + 1 > 0] &= [0.5 > 0] = 1 \\[-1.5 + 0 + 0 > 0] &= [-1.5 > 0] = 0 \\[-1.5 + 0 + 1 > 0] &= [-0.5 > 0] = 0 \\[-1.5 + 1 + 0 > 0] &= [-0.5 > 0] = 0\end{aligned}$$

С объединением та же ситуация, только на этот раз линия должна пройти чуть ниже. Подойдёт  $x_2 = 0.5 - x_1$ .



С третьей операцией, исключающим или, начинаются проблемы. Чтобы разделить точки, нужно строить две линии. Сделать это можно многими способами. Но линий всегда будет две. То есть мы попадаем в ситуацию из прошлой задачи. Надо посмотреть первым слоем нейросетки где мы оказались относительно каждой из линий, а вторым слоем соединить результаты в окончательный ответ.





Если немного пофантазировать, можно даже записать эту нейросеть через объединение и пересечение:

$$\hat{y} = [1 \cdot (x_1 \cup x_2) - 1 \cdot (x_1 \cap x_2) - 0.5 > 0]$$

Узнали? Это нейросетка из предыдущей задачки! Нейрон  $(x_1 \cup x_2)$  выясняет по какую сторону от сплошной линии мы оказались, нейрон  $x_1 \cap x_2$  делает то же самое для пунктирной линии. А дальше мы просто объединяем результат.

### Упражнение 6 (ещё немного про XoR)

Маша заметила, что на XoR ушло очень много персептронов. Она поняла, что первые два персептрона пытаются сварить для третьего нелинейные признаки, которых нейросетке не хватает. Она решила самостоятельно добавить персептрону вход  $x_3 = x_1 \cdot x_2$  и реализовать XoR одним персептроном. Можно ли это сделать?

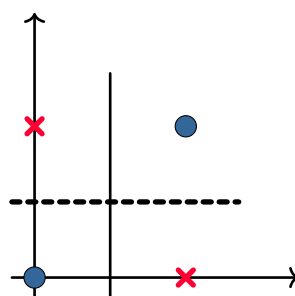
#### Решение:

Маша обратила внимание на очень важную штуку. Нам не хватает признаков, чтобы реализовать XoR за один нейрон. Поэтому первый слой нейросетки сам их для нас придумывает. Чем глубже нейросетку мы построим, тем более сложные и абстрактные признаки она будет выделять из данных и подавать дальше.

Если добавить ко входу  $x_3 = x_1 \cdot x_2$  мы сделаем за нейросетку часть её работы и сможем обойтись одним нейроном. Например, вот таким:

$$\hat{y} = [x_1 + x_2 - 2 \cdot x_1 \cdot x_2 - 0.5 > 0]$$

Такая линия как раз будет задавать две скрещивающиеся прямые.



Это легко увидеть, если немного поколдовать над уравнением:

$$x_1 + x_2 - 2x_1x_2 - 0.5 = 0$$

$$2x_1 + 2x_2 - 4x_1x_2 - 1 = 0$$

$$2x_1(1 - 2x_2) + 2x_2 - 1 = 0$$

$$(1 - 2x_2) \cdot (2x_1 - 1) = 0$$

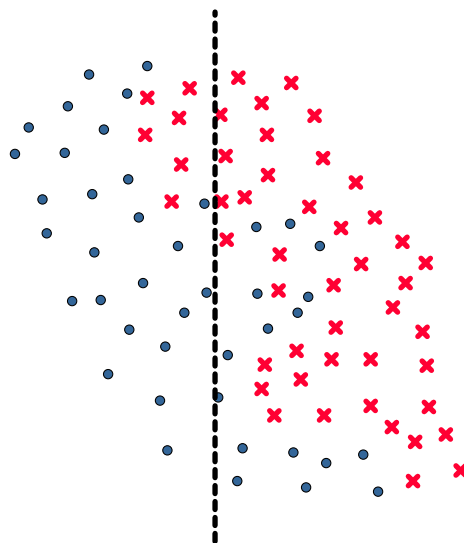
Получаем два решения. Прямую  $x_2 = 0.5$  и прямую  $x_1 = 0.5$ .

### Упражнение 7 (универсальный классификатор)

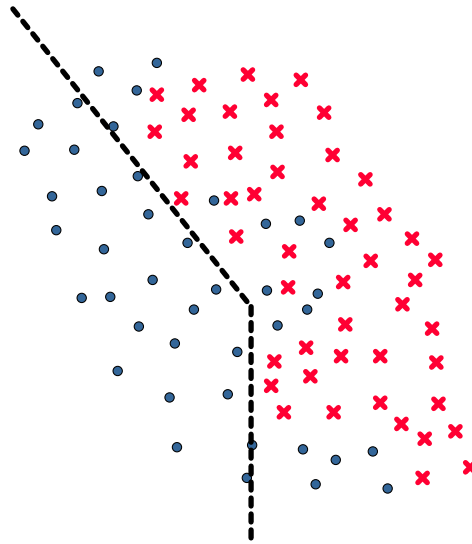
Маша задумалась о том, можно ли с помощью нейронной сетки с одним скрытым слоем и ступенчатой функцией активации решить абсолютно любую задачу классификации на два класса со сколь угодно большой точностью? Ей кажется, что да. Как это можно сделать?

#### Решение:

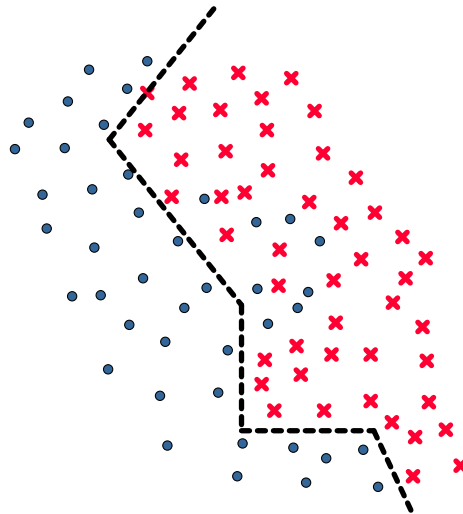
Сейчас мы докажем, что нейросеть может решить любую задачу классификации с любой точностью. Готовы? Для начала нарисуем какую-нибудь одну задачу классификации и попробуем решить её одним нейроном. Он построит нам одну линию так, чтобы мы пореже ошибались.



Такая разделяющая полоса будет давать нам какое-то число ошибок. Как улучшить результат работы такой нейронки? Ответ прост: Давайте возьмём на первом слое два нейрона. Каждый из них построит по линии. На втором слое возьмём один нейрон, который объединит результат работы первого слоя и скажет нам где именно мы оказались. Тогда получим такое решение:



Отлично! Разделяющая поверхность стала поинтереснее и мы стали лучше разбираться с тем в какой части плоскости мы оказались. Давайте на первом слое прикрутим ещё несколько персептронов, которые будут рисовать нам на плоскости линии:



На первом слое пять персептронов решают по какую сторону от каждой прямой мы оказались. На втором слое находится один единственный нейрон, который объединяет все решения в итоговый ответ. Всегда, когда мы оказываемся относительно прямой в зоне крестов, персептрон выдаёт на выход единицу. Если второй слой видит пять единиц, он прогнозирует крест, то есть 1. Если хотя бы одной единицы нет, значит мы оказались в зоне точки и прогнозировать нужно 0:

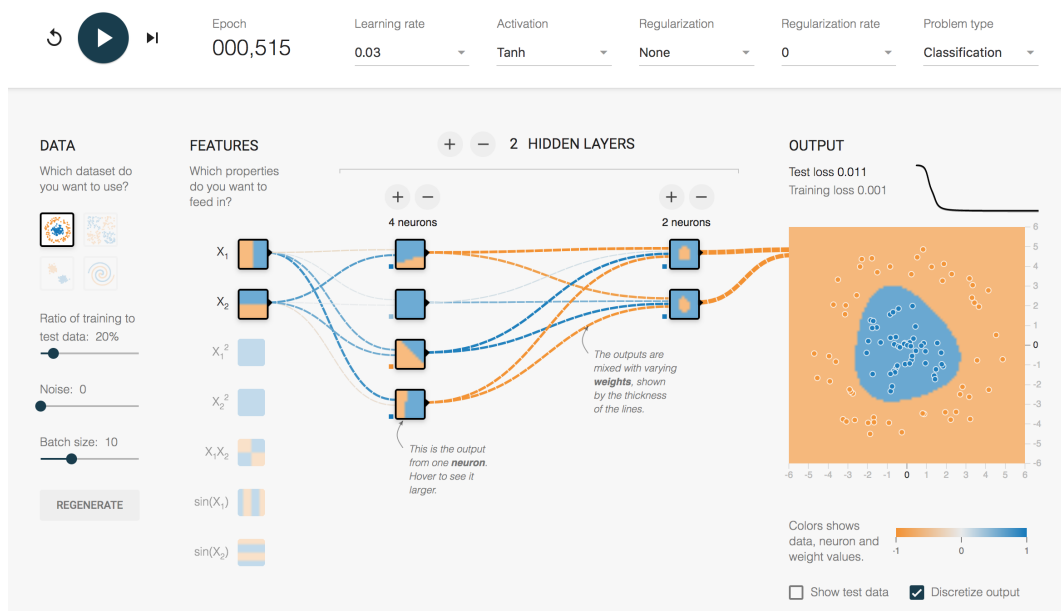
$$\hat{y} = [x_1 + x_2 + x_3 + x_4 + x_5 - 4.5 > 0].$$

Увеличивая число персептронов на первом слое и поднимая константу на втором, мы можем добиться любой точности при решении нашей задачи. Если в качестве функций активации использовать не пороговую, а, например, сигмоиду, то граница будет получаться гладкой. Принцип

работы при этом не поменяется.

## Упражнение 8 (избыток)

На сайте <http://playground.tensorflow.org> Маша стала играть с простенькими нейросетками и обучила для решения задачи классификации трёхслойного монстра.



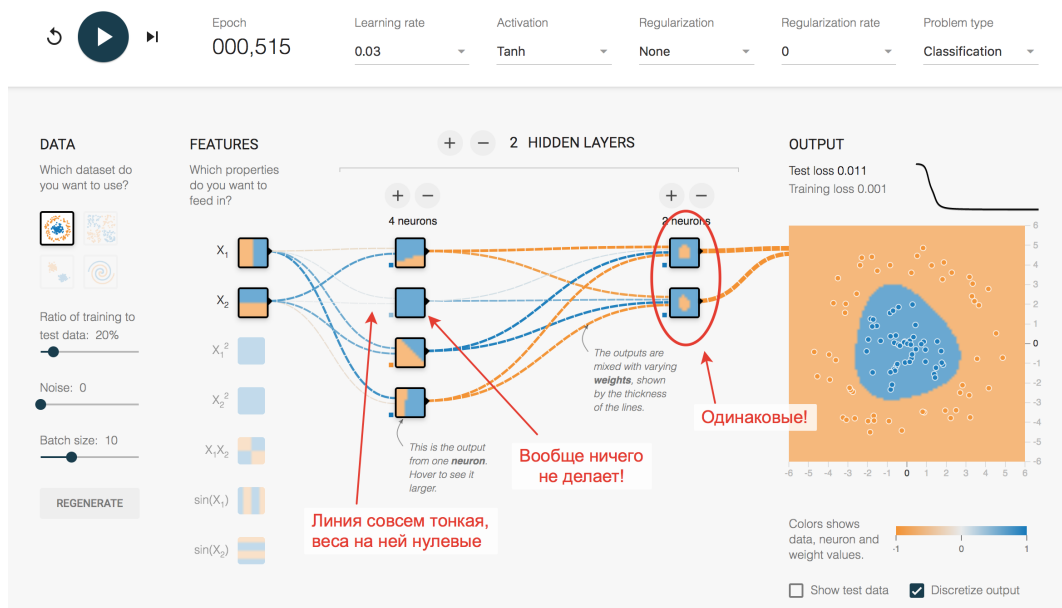
Голубым цветом обозначен первый класс, рыжим второй. Внутри каждого нейрона визуализирована та разделяющая поверхность, которую он выстраивает. Так, первый слой ищет разделяющую линию. Второй слой пытается из этих линий выстроить более сложные фигуры и так далее. Чем ярче связь между нейронами, тем больше весовой коэффициент, относящийся к ней. Синие связи — положительные, рыжие — отрицательные. Чем тусклее связь, тем он ближе к нулю.

Маша заметила, что с её получившейся архитектурой что-то не так. Какие проблемы вы в ней видите?

### Решение:

Нейросетка Маши оказалась избыточной. Во-первых, можно увидеть, что на первом слое есть нейрон, который вообще ничего не делает. Связи, которые идут к нему от входов настолько тусклые (коэффициенты при них равны нулю), что их даже не видно на картинке. От этого нейрона смело можно избавиться и сделать архитектуру проще.

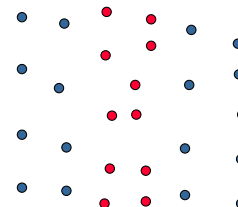
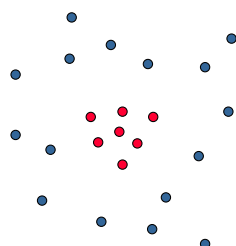
Во-вторых, можно заметить, что на последнем слое у нас есть два одинаковых нейрона. Один из них смело можно выбрасывать.



Для решения такой простой задачи классификации подойдёт более простая модель. Сколько минимально нужно нейронов, чтобы её решить вам и Маше предстоит выяснить в следующей задачке.

## Упражнение 9 (минималочка)

Сколько минимально нейронов необходимо для решения следующих задач классификации? Сколько слоёв минимально должно быть в нейросетке? Почему?



Добавить норм косинуса для точек как на семе

**Решение:**

## Упражнение 10 (универсальный регрессор)

Доказать, что с помощью однослойной нейронной сетки можно приблизить любую непрерывную функцию от одного аргумента  $f(x)$  со сколь угодно большой точностью<sup>3</sup>.

**Hint:** Вспомните, что любую непрерывную функцию можно приблизить с помощью кусочно-линейной функции (ступеньки). Осознайте как с помощью пары нейронов можно описать такую ступеньку. Соедините все ступеньки в сумму с помощью выходного нейрона.

<sup>3</sup><http://neuralnetworksanddeeplearning.com/chap4.html>

Решение:

## 2 50 оттенков градиентного спуска

## 3 Backpropagation

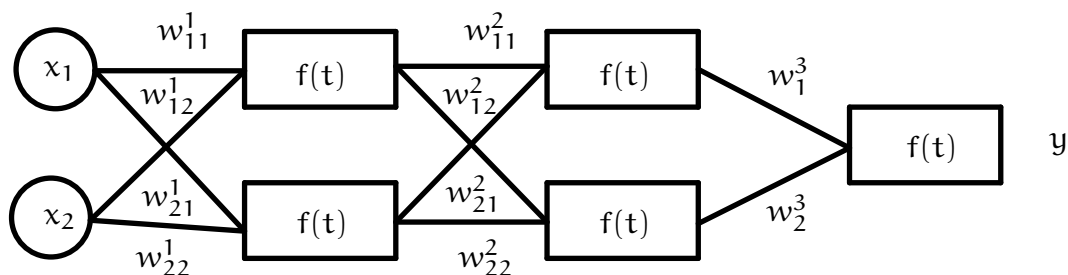
Что происходит, когда мы суём пальцы в розетку? Нас бьёт током! Мы делаем ошибку, и она распространяется по нашему телу назад.

### Упражнение 1 (граф вычислений)

Изобразите для функции  $f(x, y) = x^2 + xy + (x + y)^2$  граф вычислений. Найдите производные всех выходов по всем входам. Опираясь на граф выпишите частные производные функции  $f$ .<sup>4</sup>

### Упражнение 2 (придумываем backpropagation)

Дана нейросетка:

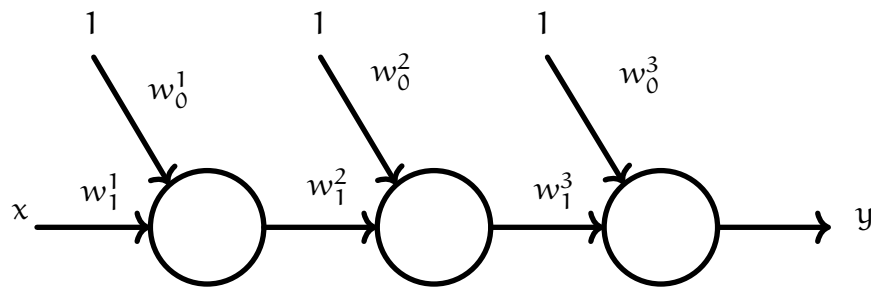


- Перепишите её как сложную функцию.
- Запишите эту функцию в матричном виде.
- Предположим, что  $L(W_1, W_2, W_3) = \frac{1}{2} \cdot (y - \hat{y})^2$  — функция потерь, где  $W_i$  — веса  $i$ -го слоя. Найдите производную функции  $L$  по всем весам  $W_i$ .
- Выглядит не очень оптимально, правда? Выпишите все производные в том виде, в котором их было бы удобно использовать для алгоритма обратного распространения ошибки, а затем, сформулируйте сам алгоритм.

### Упражнение 3 (backpropagation руками)

<sup>4</sup>По мотивам книги Николенко "Глубокое обучение" (стр. 79)

Как-то раз Вовочка решал задачу классификации. С тех пор у него в кармане завалялась нейросеть:



В качестве функции активации используется сигмоид:  $f(t) = \frac{e^t}{1+e^t}$ . Есть два наблюдения:  $x_1 = 1, x_2 = 5, y_1 = 1, y_2 = 0$ . Скорость обучения  $\gamma = 1$ . В качестве инициализации взяты нулевые веса. Как это обычно бывает, Вовочка обнаружил её в своих штанах после стирки и очень обрадовался. Теперь он собирается сделать два шага стохастического градиентного спуска, используя алгоритм обратного распространения ошибки. Помогите ему.

#### Упражнение 4 (ещё один backpropagation)

Пусть у нас есть нейронка:

$$y = f(X \cdot W_2) \cdot W_1$$

Как для функции потерь  $L(W_1, W_2) = (y - \hat{y})^2$  будет выглядеть алгоритм обратного распространения ошибки, если  $f(t) = \text{ReLU}(t) = \max(0; t)$ ? Найдите все выходы, все промежуточные производные. Опишите правило, по которому производная будет накапливаться, а также сам шаг градиентного спуска.

#### Упражнение 5 М

аша (ОПЯТЬ ОНА?!) собрала нейросеть:

$$y = \max \left( 0; X \cdot \begin{pmatrix} 1 & -1 \\ 0.5 & 0 \end{pmatrix} \right) \cdot \begin{pmatrix} 0.5 \\ 1 \end{pmatrix}$$

Теперь Маша внимательно смотрит на неё.

- Первый слой нашей нейросетки — линейный. По какой формуле делается forward pass? Предположим, что на вход пришло наблюдение  $x = (1, 2)$ . Сделайте через этот слой forward pass и найдите выход из слоя.
- Найдите для первого слоя производную выхода по входу. При обратном движении по нейросетке, в первый слой пришёл накопленный градиент  $(-1, 0)$ . Каким будет новое накоп-

ленное значение градиента, которое выплунет из себя линейный слой?

- в. Второй слой нейросетки — функция активации, ReLU. По какой формуле делается forward pass? На вход в него поступило значение  $(2, -1)$ . Сделайте через него forward pass.
- г. Найдите для второго слоя производную выхода по входу. При обратном движении по нейросетке во второй слой пришёл накопленный градиент  $(-1, -2)$ . Каким будет новое накопленное значение градиента, которое выплунет из себя ReLU?
- д. Третий слой нейросетки — линейный. По какой формуле делается forward pass? Пусть на вход поступило значение  $(2, 0)$ . Сделайте через него forward pass.
- е. Найдите для третьего слоя производную выхода по входу. При обратном движении по нейросетке, в третий слой пришёл накопленный градиент  $-2$ . Каким будет новое накопленное значение градиента, которое выплунет из себя линейный слой?
- ж. Мы решаем задачу Регрессии. В качестве функции ошибки мы используем MSE. Пусть для рассматриваемого наблюдения реальное значение  $y = 0$ . Найдите значение MSE. Чему равна производная MSE по входу (прогнозу)? Каким будет накопленное значение градиента, которое MSE выплунет из себя в предыдущий слой нейросетки, если изначально значение градиента инициализированно единицей?
- з. Пусть скорость обучения  $\gamma = 1$ . Сделайте для весов нейросети шаг градиентного спуска.

Посидела Маша, посидела, и поняла, что неправильно она всё делает. В реальности перед ней не задача регрессии, а задача классификации.

- а. Маша навинтила поверх второго линейного слоя сигмоиду. Как будет для неё выглядеть forward pass? Сделайте его. Найдите для сигмоиды производную выхода по входу.
- б. В качестве функции потерь Маша использует logloss. Как для этой функции потерь выглядит forward pass? Сделайте его. Найдите для logloss производную выхода по входу.
- в. Как будет выглядеть backward pass через logloss и сигмоиду? Прделайте его. Как изменится процедура градиентного спуска для остальной части сети?

## 4 Активация

Сюда перекинуть всё из task2

### Упражнение 1

Та, в чьих руках находится лёрнинг (это Маша), решила немного поэкспериментировать с выходами из своей сетки.

- а) Для начала Маша решила, что хочет решать задачу классификации на два класса и получать на выходе вероятность принадлежности к первому. Что ей надо сделать с последним слоем сетки?



- b) Теперь Маша хочет решать задачу классификации на  $K$  классов. Что ей делать с последним слоем?
- c) Новые вводные! Маша хочет спрогнозировать рейтинг фильма на "Кинопоиске". Он измеряется по шкале от 0 до 10 и принимает любое непрерывное значение. Как Маша может приспособить для этого свою нейронку?
- d) У Маши есть куча новостей. Каждая новость может быть спортивной, политической или экономической. Иногда новость может относиться сразу к нескольким категориям. Как Маше собрать нейросетку для решения этой задачи? Как будет выглядеть при этом функция ошибки?
- e) Маша пошла в кафе. А там куча народу. Сейчас она сидит за столиком, попивает ванильный топлёный кортадо и думает о нём, о лёрнинге. Сейчас мысль такая: как можно спрогнозировать число людей в кафе так, чтобы на выходе сетка всегда прогнозировала целое число. Надо ли как-то при этом менять функцию потерь?

## 5 Регуляризаторы

## 6 Всего лишь кубики LEGO

## 7 Итоговый тест в стел Носко

- a. Вопрос про батчнормализацию первым слоем вместо нормализации в предобработке.