Тятя! Тятя! Нейросети заменили продавца!

Пилиф Ульянкин

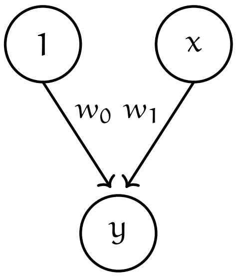
https://github.com/FUlyankin/neural_nets_prob

Листочек 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)*

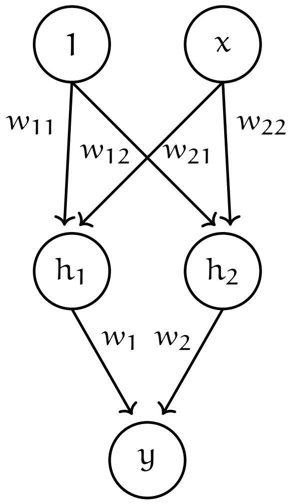
Упражнение 1 (от регрессии к нейросетке)

Однажды вечером, по пути с работы[1] Маша зашла в свою любимую кофейню на Тверской. Там, на стене, она обнаружила интересную картину:



Хозяин кофейни, Добродум, объяснил Маше, что это Покрас-Лампас нарисовал линейную регрессию, и её легко можно переписать в виде формулы: $y_i = w_0 + w_1 \cdot x_i$. Пока Добродум готовил кофе, Маша накидала у себя на бумажке новую картинку:

[1]она работает рисёрчером.



Как такая функция будет выглядеть в виде формулы? Правда ли, что y будет нелинейно зависеть от x? Если нет, как это исправить и сделать зависимость нелинейной?

Решение:

Когда Добрадум записывал картинку в виде уравнения, он брал вход из кругляшей, умножал его на веса, написанные около стрелок и искал сумму. Сделаем ровно то же самое для Машиной картинки. Величины h_i внутри кругляшей скрытого слоя будут считаться как:

$$h_1 = w_{11} \cdot 1 + w_{21} \cdot x$$
$$h_2 = w_{12} \cdot 1 + w_{22} \cdot x$$

Итоговый y будет получаться из этих промежуточных величин как

$$y = w_1 \cdot h_1 + w_2 \cdot h_2.$$

Подставим вместо h_i их выражение через x и получим уравнение, которое описывает картинку Маши

$$\begin{aligned} y &= w_1 \cdot h_1 + w_2 \cdot h_2 = \\ &= w_1 \cdot (w_{11} + w_{21} \cdot x) + w_2 \cdot (w_{12} + w_{22} \cdot x) = \\ &= \underbrace{(w_1 w_{11} + w_2 w_{12})}_{\gamma_1} + \underbrace{(w_1 w_{21} + w_2 w_{22})}_{\gamma_2} \cdot x. \end{aligned}$$

Когда мы раскрыли скобки, мы получили ровно ту же самую линейную регрессию. Правда мы зачем-то параметризовали γ_1 и γ_2 через шесть параметров. Чтобы сделать зависимость нелинейной, нужно преобразить каждую из h_i, взяв от них нелинейную функцию. Например, сигмоиду:

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

Тогда формула изменится:

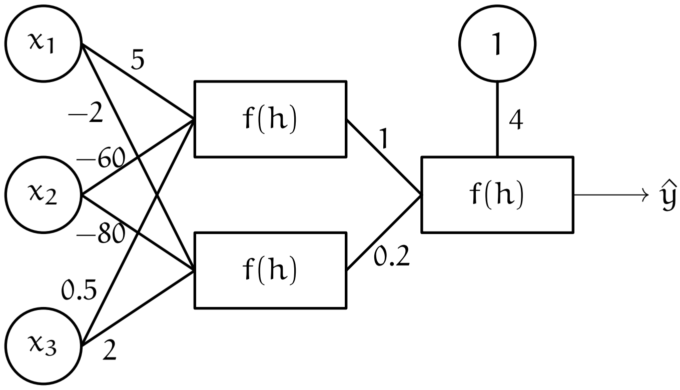
$$y = w_1 \cdot f(w_{11} + w_{21} \cdot x) + w_2 \cdot f(w_{12} + w_{22} \cdot x).$$

Линейности больше нет. **Только что на ваших глазах произошло чудо. Регрессия превратилась в нейросеть.** Функцию $f(h)$ называют **функцией активации.** Вместо сигмоиды можно использовать другие функции. В современных нейросетях чаще всего применяют ReLU (Rectified Linear Unit). Она нелинейная и просто вычисляется

$$\mathrm{ReLU}(h) = \max(0, h).$$

Упражнение 2 (из картинки в формулу)

Добродум хочет понять, насколько сильно будет заполнена кофейня в следующие выходные. Для этого 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.



а. В эти выходные за барной[2] стойкой стоит Агнесса. Митинга не предвидится, температура будет в районе 20 градусов. Спрогнозируйте, сколько человек придёт в кофейню к Добродуму?

б. На самом деле каждая нейросетка — это просто-напросто какая-то нелинейная 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:

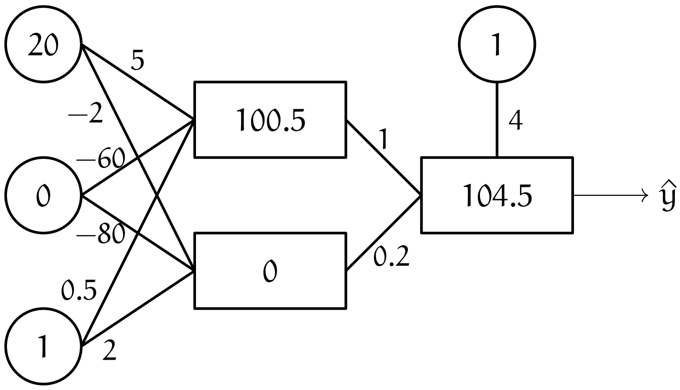
$$h_2 = \max(0, -2 \cdot 20 + (-80) \cdot 0 + 2 \cdot 1) = \max(0, -38) = 0$$

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.

[2]барной... конечно, кофейня у него...



Теперь по мотивам наших вычислений запишем нейронку в виде функции. 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 эквивалентно обучению такой нелинейной функции.

Упражнение 3 (из формулы в картинку)

Маша написала на бумажке функцию:

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

Теперь она хочет, чтобы кто-нибудь из её адептов нарисовал её в виде нейросетки. Нарисуйте.

Решение:

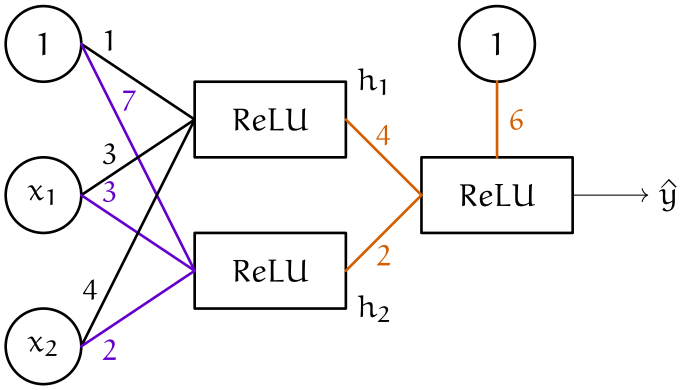
Начнём рисовать картинку с конца. На выход выплёвывается либо 0, либо комбинация из двух входов:

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

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:

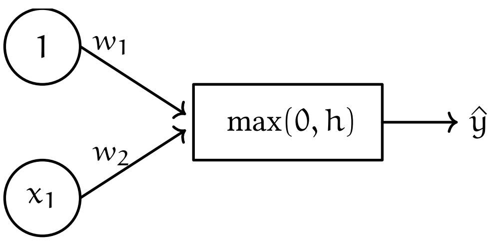


Упражнение 4 (армия регрессий)

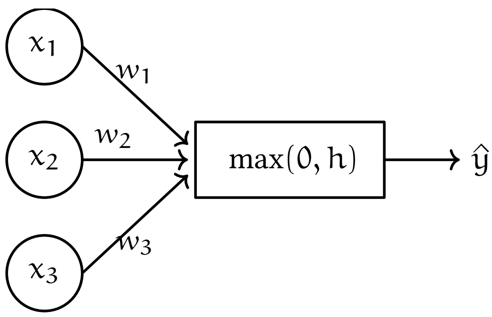
Парни очень любят Машу,[3] а Маша с недавних пор любит собирать персептроны и думать по вечерам об их весах и функциях активации. Сегодня она решила разобрать свои залежи из персептронов и как следует упорядочить их.

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



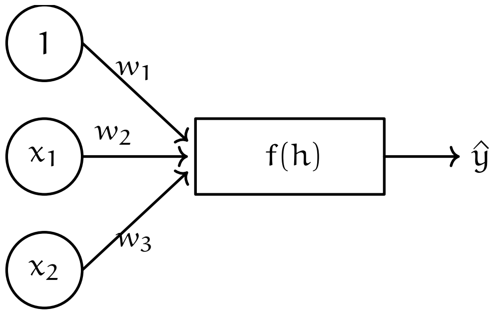
б. В тумбочке, среди носков, Маша нашла персептрон, с картинки 2, Маша хочет подобрать такие веса w_i, чтобы персептрон превращал x из таблички в соответствующие y:

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

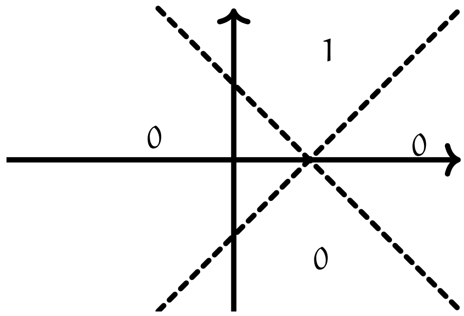
[3]когда у тебя есть лёрнинг, они так и лезут

в. Оказывается, что в ванной всё это время валялась куча персептронов с картинки 3 с неизвестной функцией активации.

Картинка 3



Картинка 4



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. В качестве функции возьмите единичную ступеньку (Функцию Хевисайда)

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

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

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

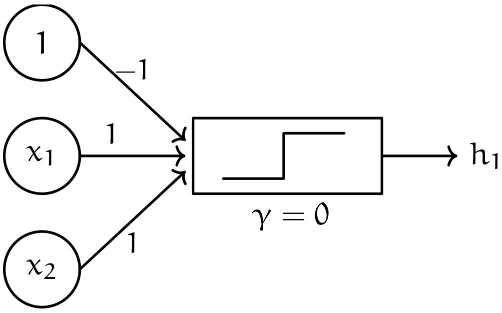
На второе уравнение w_2 никак не влияет, а $w_1 = 1$. Для того, чтобы в первом уравнении получить ноль, нужно взять любое $w_2 \leqslant -1$.

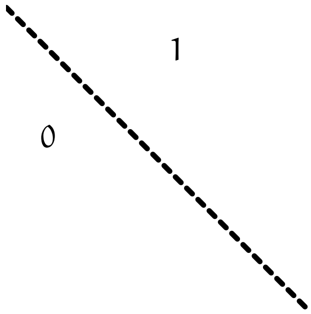
б. Снова выписываем несколько уравнений:

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

Тут решений может быть довольно много. Одно из них — 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.

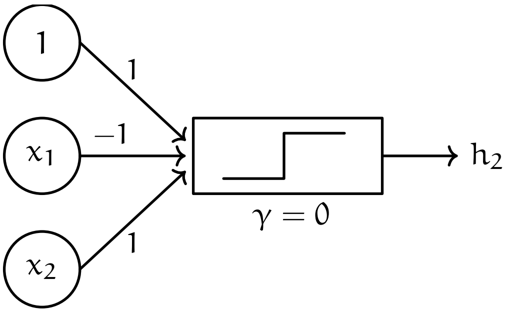
в. Один нейрон — это одна линия, проведённая на плоскости. Эта линия отделяет один класс от другого. Например, линию $x_1 + x_2 - 1 = 0$ мог бы описать нейрон

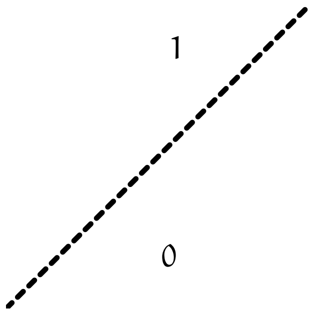




Порог γ для кусочной функции в каком-то смысле дублирует константу. Они взаимосвязаны. Будем всегда брать его нулевым. Видим, что если мы получили комбинацию x_1, x_2 и 1, большую, чем ноль, мы оказались справа от прямой. Если хочется поменять метки 0 и 1 местами, можно умножить все коэффициенты на -1.

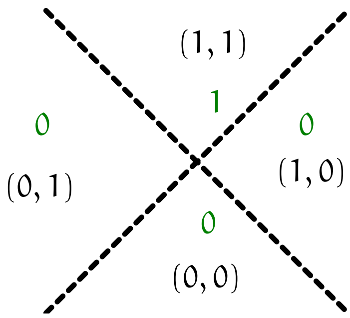
Наш персептрон понимает по какую сторону от прямой мы оказались, то есть задаёт одну линейную разделяющую поверхность. По аналогии для второй прямой мы можем получить следующий нейрон.

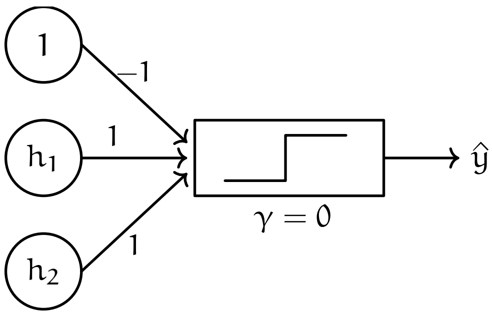




Итак, первый персептрон выбрал нам позицию относительно первой прямой, второй относительно второй. Остаётся только объединить эти результаты. Нейрон для скрепки должен реализовать для нас логическую функцию, которую задаёт табличка ниже. Там же нарисованы примеры весов, которые могли бы объединить выхлоп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

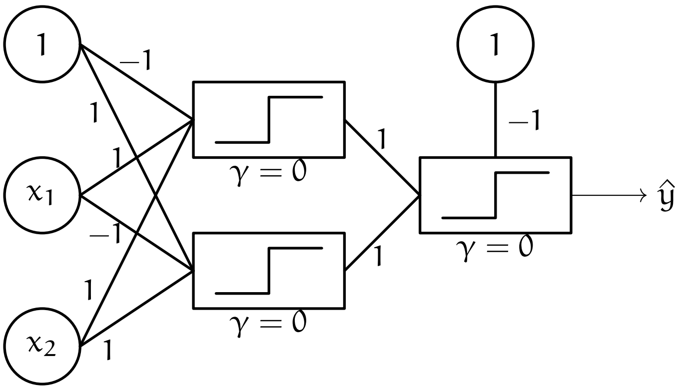
h_1	h_2	$\hat{y}$
1	1	1
1	0	0
0	1	0
0	0	0





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.

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.** Другими словами говоря, нейросетка своим первым слоем превратила сложное пространство признаков в более простое, а затем вторым слоем, решила в нём задачу классификации.



Упражнение 5 (логические функции)

Маша вчера поссорилась с Пашей. Он сказал, что у неё нет логики. Чтобы доказать Паше обратное, Маша нашла теорему, которая говорит о том, что с помощью нейросетки можно аппроксимировать почти любую функцию, и теперь собирается заняться аппроксимацией логических функций. Для начала она взяла самые простые, 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

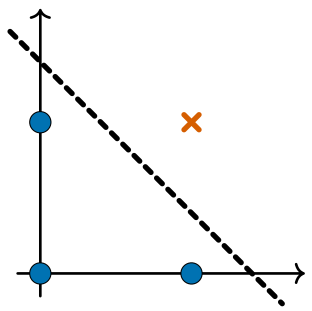
x_1	x_2	$x_1 \cup x_2$
1	1	1
1	0	1
0	1	1
0	0	0

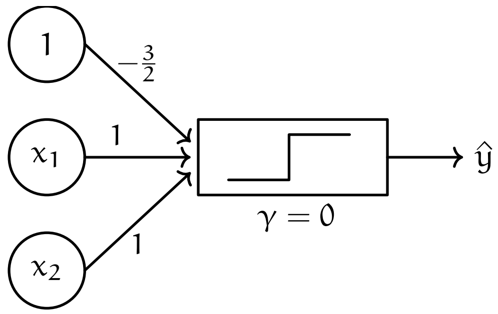
x_1	x_2	x_1 XoR x_2
1	1	0
1	0	1
0	1	1
0	0	0

Первые два столбика идут на вход, третий получается на выходе. Первая операция — логическое «и», вторая — «или». Операция из третьей таблицы называется «исключающим или», (XoR). Если внимательно приглядеться, то можно заметить, что XoR — это то же самое что и $[x_1 \neq x_2]$[4].

Решение:

В предыдущем упражнении мы уже построили нейрон для пересечения. Он располагался на последнем слое нейросети. Посмотрим на тот же нейрон под другим углом.



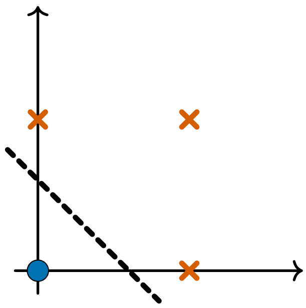


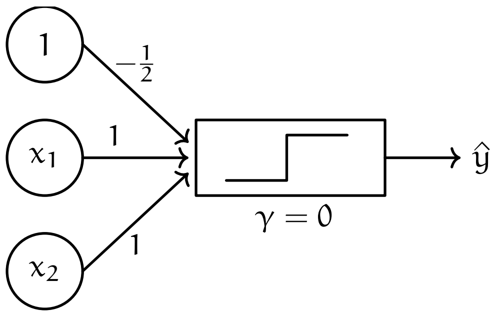
[4]Тут квадратные скобки обозначают индикатор. Он выдаёт 1, если внутри него стоит правда и 0, если ложь. Такая запись называется скобкой Айверсона. Попробуйте записать через неё единичную ступеньку Хевисайда.

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

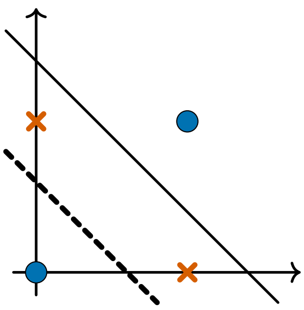
$$
\begin{aligned}
[-1.5 + 1 + 1 > 0] &= [0.5 > 0] = 1 \\
[-1.5 + 0 + 0 > 0] &= [-1.5 > 0] = 0 \\
[-1.5 + 0 + 1 > 0] &= [-0.5 > 0] = 0 \\
[-1.5 + 1 + 0 > 0] &= [-0.5 > 0] = 0
\end{aligned}
$$

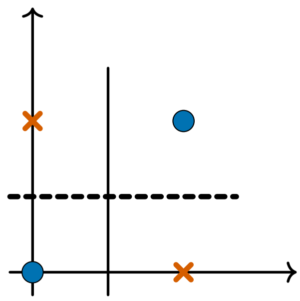
С объединением та же ситуация, только на этот раз линия должна пройти чуть ниже. Подойдёт $x_2 = 0.5 - x_1$.





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





Можно даже записать эту нейросеть через объединение и пересечение:

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

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. Получается, что XoR можно описать двухслойной нейронной сетью.

Упражнение 6 (ещё немного про XoR)

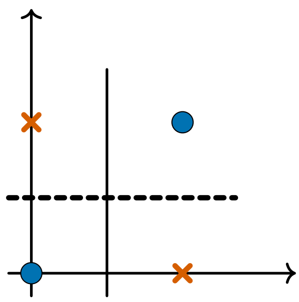
Маша заметила, что на XoR ушло очень много персептронов. Она поняла, что первые два персептрона пытаются построить для третьего нелинейные признаки, которых нейросетке не хватает. Она решила самостоятельно добавить персептрону вход $x_3 = x_1 \cdot x_2$ и реализовать XoR одним персептроном. Можно ли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



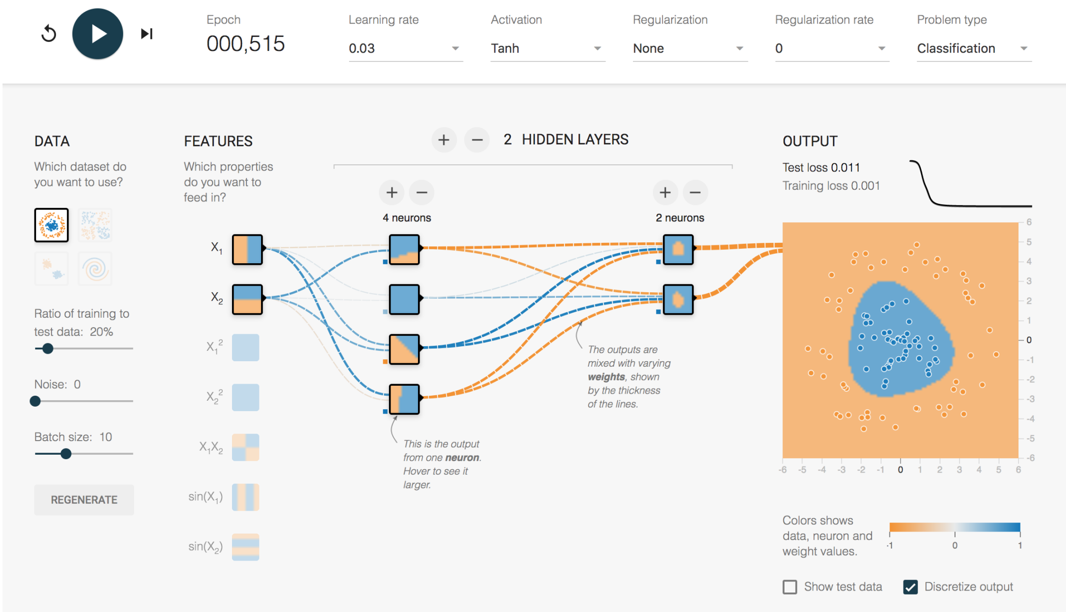
$$\begin{aligned}
&x_1 + x_2 - 2x_1x_2 - 0.5 = 0 \\
&2x_1 + 2x_2 - 4x_1x_2 - 1 = 0 \\
&2x_1(1 - 2x_2) + 2x_2 - 1 = 0 \\
&(1 - 2x_2) \cdot (2x_1 - 1) = 0
\end{aligned}$$

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

Упражнение 7 (избыток)

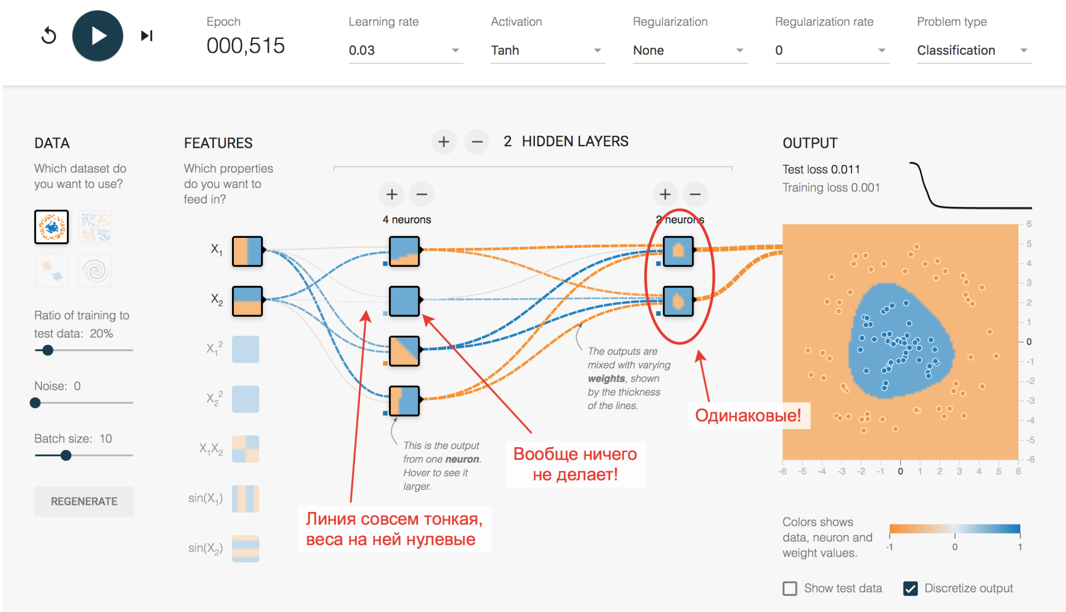
На сайте `http://playground.tensorflow.org` Маша стала играться с простенькими нейросетками и обучила для решения задачи классификации трёхслойного монстра.

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, относящейся к ней. Синие связи — положительные, рыжие — отрицательные. Чем тусклее связь, тем он ближе к нулю. Маша заметила, что с её архитектурой что-то не так. Какие у неё проблемы?



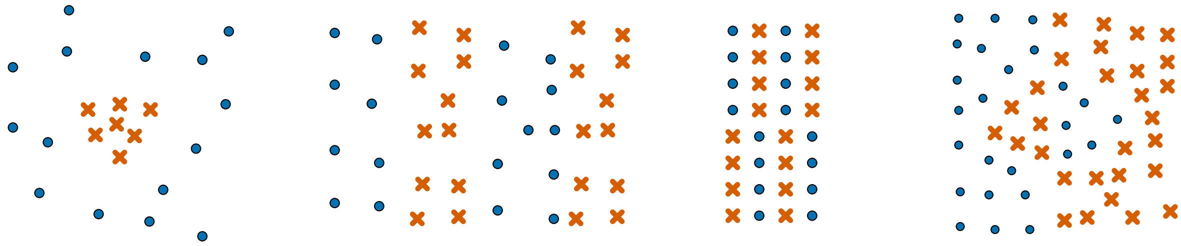
Решение:

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.



Упражнение 8 (минималочка)

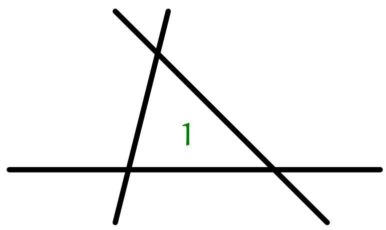
Шестилетняя сестрёнка ворвалась в квартиру Маши и разрисовала ей все обои:

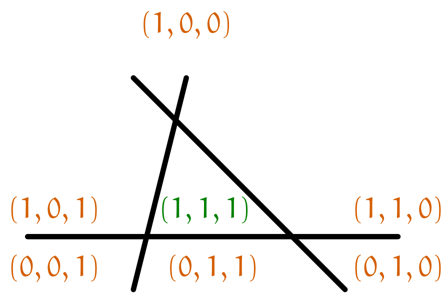


Маша по жизни оптимистка. Поэтому она увидела не дополнительные траты на ремонт, а четыре задачи классификации. И теперь в её голове вопрос: сколько минимально нейронов нужно, чтобы эти задачи решить?

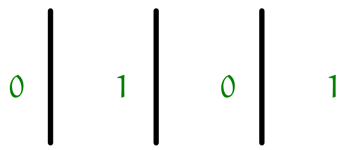
Решение:

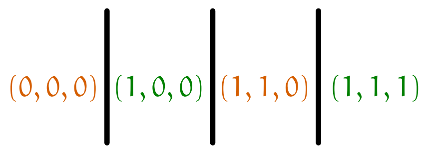
а. Нам надо выделить треугольник. Всё, что внутри будет относится к первому классу. Получается, что на первом слое надо три нейрона. Каждый из них настроим так, что если мы попадаем внутрь треугольника, он выдаёт 1. Тогда на втором слое будет достаточно одного нейрона, который удостоверится, что все три результата с первого слоя оказались равны 1. Вернитесь к предыдущей задаче, сходите на сайт с демкой и постройте оптимальную нейросетку.





б. Первый слой должен построить нам три линии. Это три нейрона. Второй слой должен принять решение в какой из полос мы оказались. Будем считать, что если мы попали направо, нейрон выдаёт единицу. Если мы попали налево, ноль. В качестве функции активации используем единичную ступеньку.





Вопрос в том, хватит ли нам на втором слое одного нейрона для того, чтобы обработать все четыре возможные ситуации. Нам нужно, чтобы выполнялись следующие условия

$$
\begin{cases}
f(1 \cdot w_1 + 1 \cdot w_2 + 1 \cdot w_3) = 1 \\
f(1 \cdot w_1 + 0 \cdot w_2 + 0 \cdot w_3) = 0 \\
f(1 \cdot w_1 + 1 \cdot w_2 + 01 \cdot w_3) = 0 \\
f(0 \cdot w_1 + 0 \cdot w_2 + 0 \cdot w_3) = 0
\end{cases}.
$$

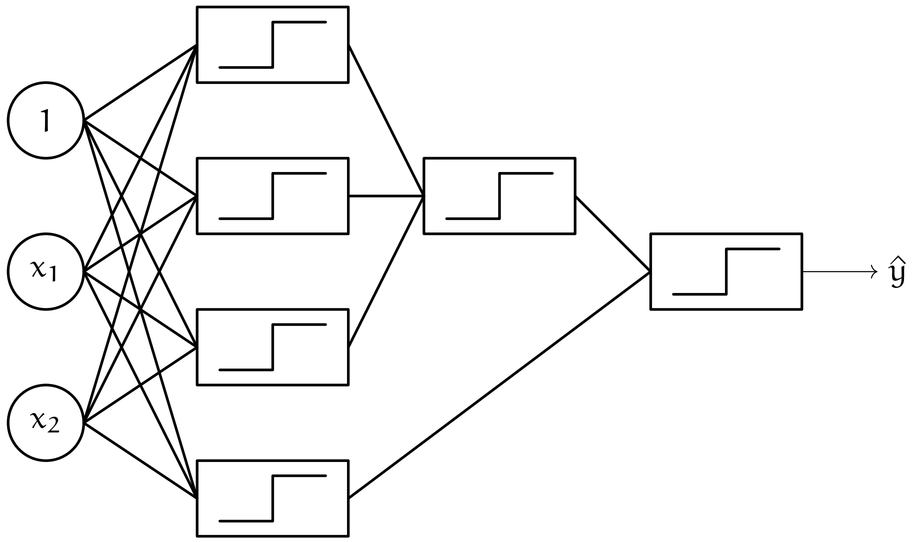
Для того, чтобы со вторым уравнением всё было хорошо, возьмём $w_1 = 1$. Тогда вес w_2

надо взять отрицательным, а w_3 положительным, например, $w_2 = -2$, а $w_3 = 4$. Тогда один нейрон на внешнем слое решит нашу задачу. Выходит, что всего надо задействовать 4 нейрона.

в. Оценим число нейронов сверху. Перед нами две XoR задачи, которые лежат рядом с друг-другом. Для решения каждой надо 3 нейрона. Чтобы объединить получившиеся решения нужен ещё один нейрон. Получается трёхслойная сетка с 7 нейронами.

   Если мы попробуем подойти к задаче также, как в предыдущем пункте, на втором слое мы получим несовместимую систему из уравнений. То есть третьего слоя точно не избежать.

   Можно первым слоем построить 3 линии, вторым решить задачу из предыдущего пункта, а на третьем добавить информацию о том, выше горизонтальной линии мы оказались или ниже. Тогда мы потратим 6 нейронов. Нейросетка получится неполносвязной.



г. Думайте, рассуждайте, а автор умывает руки.

Упражнение 9 (универсальный регрессор)

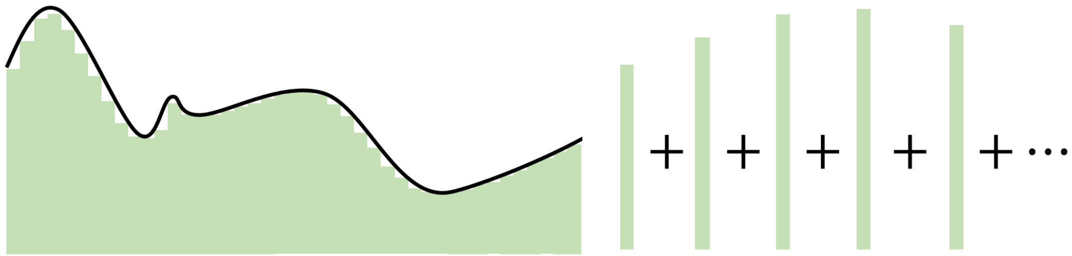
Маша доказала Паше, что у неё всё в полном порядке с логикой. Теперь она собирается доказать ему, что с помощью нейронной сетки можно приблизить любую непрерывную функцию от одного аргумента $f(x)$ со сколь угодно большой точностью[5].

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.

Решение:

Не нужно воспринимать эту задачку как строгое доказательство. Скорее, это «показательство». Мы хотим приблизить функцию $f(x)$ с какой-то точностью. Будем делать это с помощью кусочно-линейных ступенек. Чем выше точность, тем больше будем рисовать ступенек.

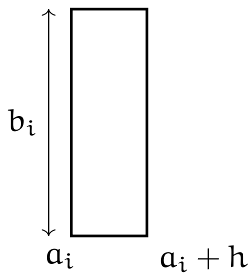
[5] http://neuralnetworksanddeeplearning.com/chap4.html



Высоту ступеньки определяют по-разному. Чаще всего, как значение функции в середине выбранного отрезка $b_i = f(\frac{a_i+a_{i+1}}{2})$. Тогда всю функцию целиком можно приблизить суммой

$$f(x) \approx \sum_{i=1}^{n} f\left(\frac{a_i + a_{i+1}}{2}\right) \cdot [a_i \geqslant x < a_{i+1}].$$

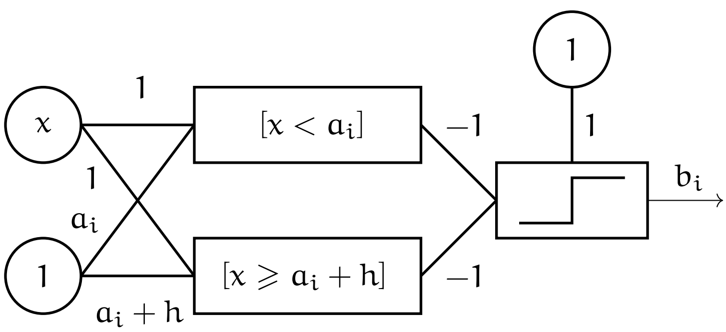
Давайте попробуем описать с помощью нейрона одну из ступенек. Пусть высота этой ступеньки равна b_i. Шагать по оси x мы будем с фиксированным шагом h, поэтому $a_{i+1} = a_i + h$.



Если x, для которого мы ищем $f(x)$, попадает в полуинтервал, на котором задана наша ступенька, мы будем приближать $f(x)$ этой ступенькой. Ступенька состоит из двух линий. Выходит, что она будет описываться двумя нейронами. Если мы внутри ступеньки, значит $a_i \leqslant x < a_i + h$. Пара нейронов должна сравнить x с a_i и $a_i + h$ и на основе этого принять решение. Можно записать попадание x в ступеньку следующим образом:

$$1 - [x < a_i] - [x \geqslant a_i + h]$$

Если оба условия — неправда, получаем 1. Мы в ступеньке. Если хотя бы одно из них выполнено — мы вылетаем за ступеньку. Оба сразу выполниться они не могут. Нарисуем это в виде нейрона. В качестве функции активации используем единичную ступеньку.



Нарисуем такую сетку для каждой ступеньки. Если мы попали в ступеньку, сетка будет выплёвывать со второго слоя единичку. Там мы будем умножать её на b_i и посылать на внешний слой. Мы всегда будем попадать только в одну из ступенек, значит только один из слоёв выдаст нам 1. Все остальные выдадут 0. На внешнем слое нам остаётся только просуммировать всё, что к нам пришло и выдать ответ. Чем больше ступенек мы добавляем в модель, тем точнее наша апроксимация.

Упражнение 10 (число параметров)

Та, кому принадлежит машин лёрнинг собирается обучить полносвязную нейронную сеть для решения задачи регрессии, На вход в ней идёт 12 переменных, в сетке есть 3 скрытых слоя. В первом слое 300 нейронов, во втором 200, в третьем 100. Сколько параметров предстоит оценить Маше?

Решение:

Нам нужно решить комбинаторную задачку. Связи в нашей сетке проведены между всему нейронами. Не забываем учесть, что у каждого нейрона есть константа. Получается, что всего параметров будет

$$(12+1)\cdot 300 + (300+1)\cdot 200 + (200+1)\cdot 100 + (100+1)\cdot 1.$$