Тятя! Тятя! Наши сети притащили мертвеца!

Ульянкин Филипп

Аннотация

В этой виньетке собрана коллекция ручных задачек про нейросетки на пару томных вечеров. Вместе с Машей можно попробовать по маленьким шажкам с ручкой и бумажкой раскрыть у себя в теле несколько чакр и немного глубже понять модели глубокого обучения.

Вместо введения

Однажды Маша услышала про какой-то Машин лёрнинг. Она сразу же смекнула, что именно она та самая Маша, кому этот лёрнинг должен принадлежать. Ещё она смекнула, что если хочет владеть лёрнингом по праву, ни одна живая душа не должна сомневаться в том, что она шарит. Поэтому она постоянно изучает что-то новое.

Её друг Миша захотел стать адептом Машиного лёрнинга, и спросил её о том, как можно за вечер зашарить алгоритм обратного распространения ошибки. Тогда Маша открыла свою первую книгу по глубокому обучению и прочитала в ней:

> Благодаря символическому дифференцированию вам никогда не придется заниматься реализацией агоритма обратного распространения вручную. Поэтому не будем тратить время на его формулировку[1].

Маше такая логика показалась странной. Поэтому она взяла книгу с более глубокой математикой. Там она прочитала, что:

Николенко

Тогда Маша взяла Библию глубокого обучения[2] и поняла, что по ней за один вечер точно не разберёшься. Слишком серьёзно всё написано. Для вечерних разборок нужно что-то более инфантильное.

У Маши оставался один выход: поскрести по лёрнингу и собрать инфантильную коллекцию ручных задачек, прорешивая которую новые адепты Машиного лёрнинга могли бы открывать у себя во чакру за чакрой. Так и появилась эта виньетка.

[1] Франсуа Шолле, Глубокое обучение на Python, стр. 77

[2] Goodfellow I., Bengio Y., Courville A. Deep learning. – MIT press, 2016.

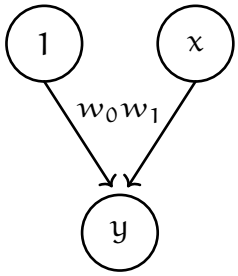
Содержание

1 Всего лишь функция

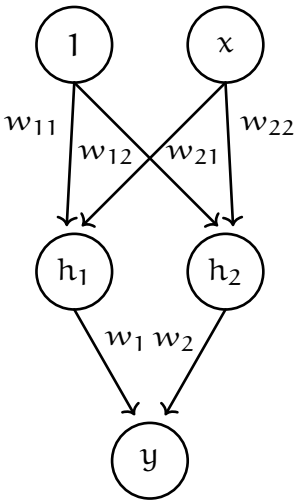
> Ты всего лишь машина, только имитация жизни. Робот сочинит симфонию? Робот превратит кусок холста в шедевр искусства?
>
> Из фильма «Я, робот» (2004)

Упражнение 1 (от регрессии к нейросетке)

Однажды вечером, по пути с работы[3] Маша зашла в свою любимую кофейню на Тверской. Там, на стене, она обнаружила очень интересную картину:



Хозяин кофейни, Добродум, объяснил Маше, что это Покрас-Лампас так нарисовал линейную регрессию,[4] 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:



а. Как такая функция будет выглядеть в виде формулы?

б. Правда ли, что y будет нелинейно зависеть от x?

в. Если нет, как это исправить и сделать зависимость нелинейной?

[3] она работает рисёрчером.

[4] эксклюзивный заказ был

Решение:

Когда мы переписывали картинку в виде уравнения регрессии, мы брали вход из кругляшей, умножали его на веса, написанные около стрелок и искали сумму.

Сделаем ровно то же самое для Машиной картинки. Буквы h 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 складываться из ашек:

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

Раскрываем h-ки и получаем для y итоговое уравнение:

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

Когда мы раскрыли скобки, мы получили ровно ту же самую линейную регрессию. Правда мы зачем-то довольно сложно параметризовали γ_1 и γ_2 через шесть параметров.

Чтобы сделать зависимость нелинейной, нужно немного преобразить каждую из h_i, взяв от них какую-нибудь нелинейную функцию. Например, сигмоиду:

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

Тогда формула преобразиться:

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

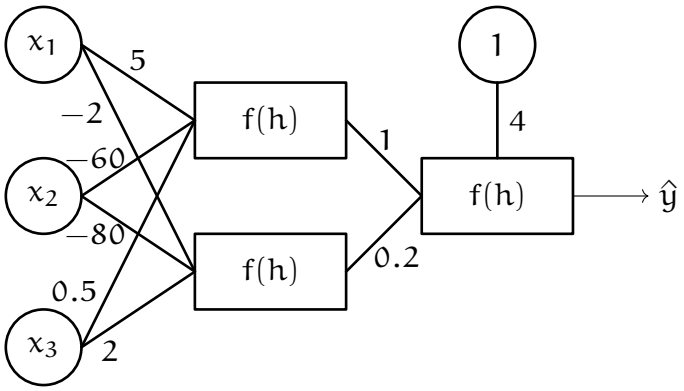
~~Смерти~~ Линейности больше нет. **Только что на ваших глазах произошло чудо. Регрессия превратилась в нейросеть.** Можно использовать вместо сигмоиды любую другую функцию. Например,

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

Она называется релу.[5] Она простая и обычно для нелинейности её хватает. Но об этом поговорим позже.

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 бариста на смене, x_3. В качестве функции активации Добродум использует ReLU.



а. В эти выходные за барной[6] стойкой стоит Агнесса. Митинга не предвидится, температура будет в районе 20 градусов. Сколько человек придёт в кофейню к Добродуму?

б. На самом деле каждая нейросетка — 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:

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

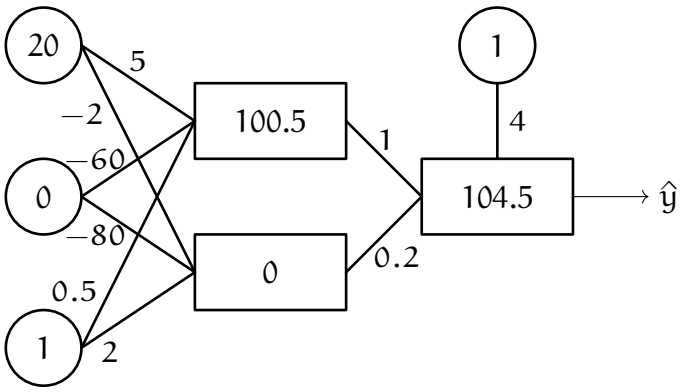
Дальше результат скрытых нейронов идёт во второй слой:

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

Это и есть итоговый прогноз.

[5] внезапное название

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



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)

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. Нарисуй.

Решение:

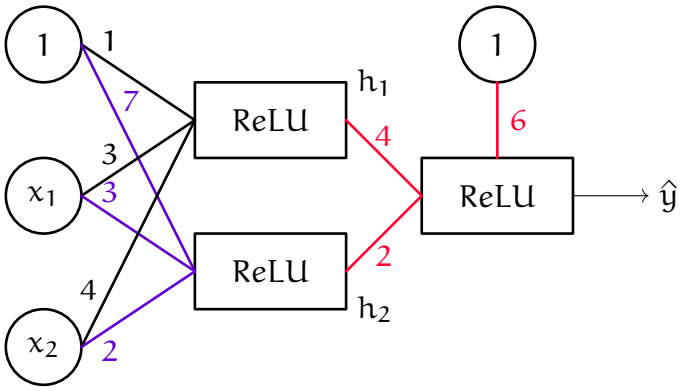
Начнём рисовать картинку с конца. На выход выплёвывается либо 0, либо комбинация из двух входов:

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

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:

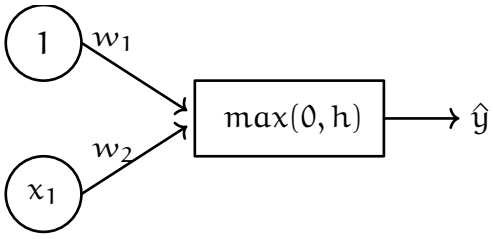


Упражнение 4 (армия регрессий)

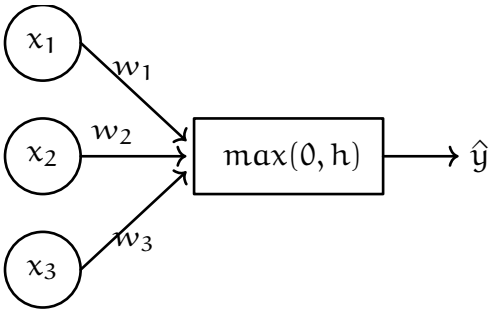
Парни очень любят Машу,[7] а Маша с недавних пор любит собирать персептроны и думать по вечерам об их весах и функциях активации. Сегодня она решила разобрать свои залежи из персептронов и как следует упорядочить их.

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 $y = 0$ (так работает логичская функция: отрицание).

Картинка 1



Картинка 2



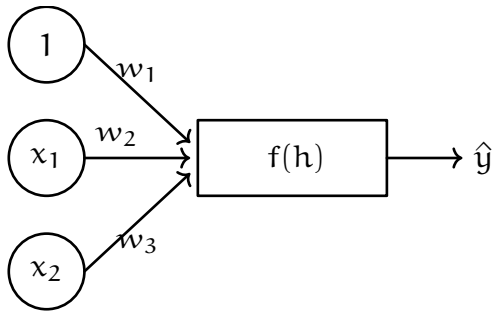
б. В тумбочке, среди носков, Маша нашла персептрон, с картинки 2, Маша хочет подобрать такие веса w_i, чтобы персептрон превращал x из таблички в соответствующие y:

x_1	x_2	x_3	y
1	1	2	0.5
1	−1	1	0

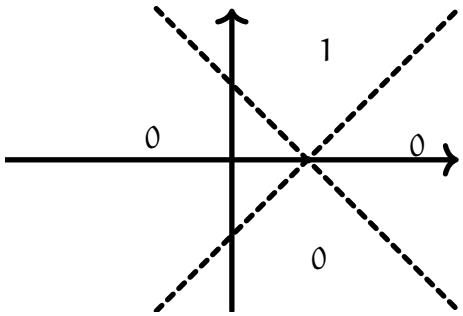
[7] когда у тебя есть лёрнинг, они так и лезут

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

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.

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 $w_2 \leqslant -1$. Нейрон готов.

б. Снова выписываем несколько уравнений:

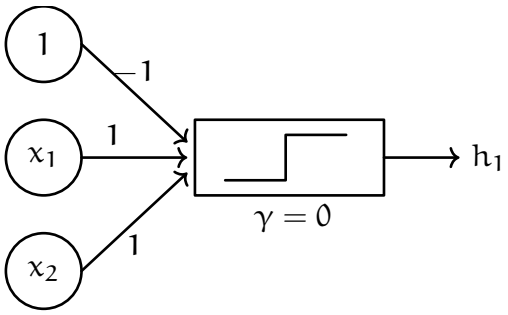
$$max(0, w_1 + w_2 + 2 \cdot w_3) = 0.5$$
$$max(0, w_1 - w_2 + w_3) = 0$$

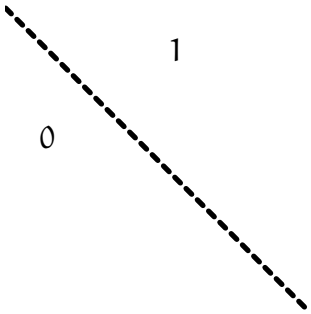
Тут решений может быть довольно много. Первое, что приходит в голову — это занулить w_1 и w_3 в первом уравнении, а w_2 поставить 0.5. Тогда во втором уравнении мы сразу же будем оказываться в отрицательной области и ReLU заботливо будет отдавать нам 0.

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

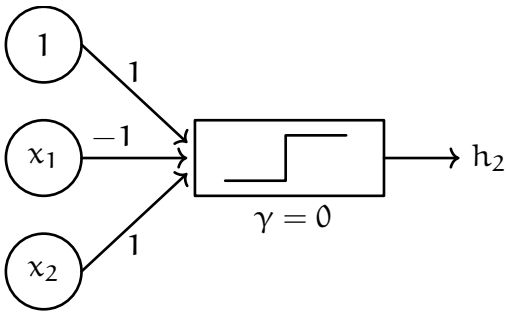
$$f(h) = \begin{cases} 1, h > 0 \\ 0, h \leqslant 0 \end{cases}.$$

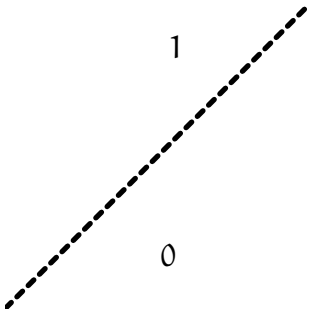
Один нейрон — это одна линия, проведённая на плоскости. Эта линия отделяет один класс от другого. Например, линию $x_1 + x_2 - 1 = 0$ мог бы описать нейрон





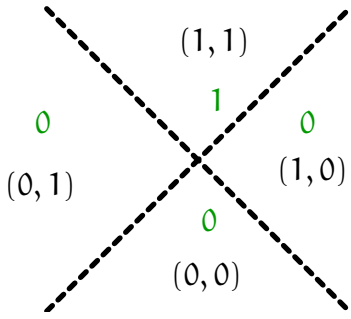
Порог γ для кусочной функции в каком-то смысле дублирует константу. Будем всегда брать его нулевым. Видим, что если мы получили комбинацию x_1, x_2 и 1, большую, чем ноль, мы оказались справа от прямой. Если хочется поменять метку 0 и 1 сторонами, можно умножить все коэффициенты на -1. **Наш персептрон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

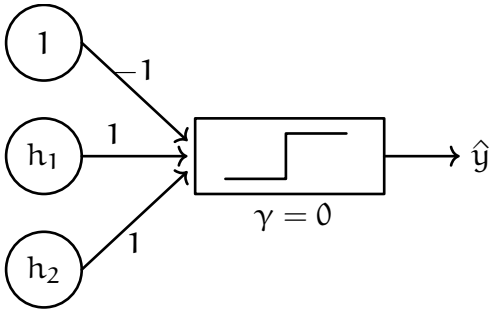




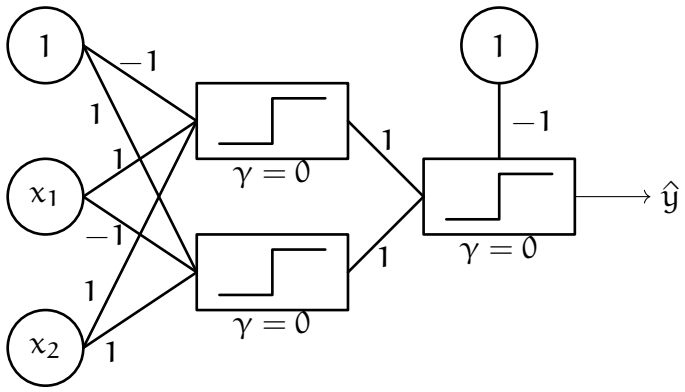
Итак, первый персептрон выбрал нам позицию относительно первой прямой, второй относительно второй. Остаётся только соединить эти результаты в один. Нейрон для скрепки должен реализовать для нас логическую функцию, которую задаёт табличка ниже. Там же нарисованы примеры весов, которые могли бы объединить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

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 апроксимацией 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

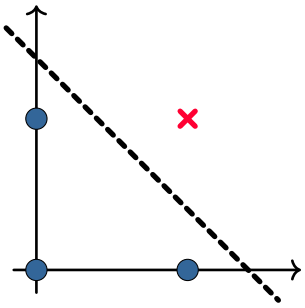
x_1	x_2	x_1 XoR x_2
1	1	0
1	0	1
0	1	1
0	0	0

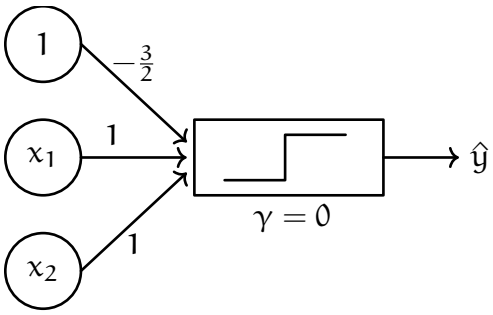
Первые два столбика идут на вход, третий получается на выходе. Первая операция — логическое "и вторая — "или". Операция из третьей таблицы называется "исключающим или"(XoR). Если внимательно приглядеться, то можно заметить, что XoR — это то же самое что и $[x_1 \neq x_2]$[8].

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:

[8]Тут квадратные скобки обозначают индикатор. Он выдаёт 1, если внутри него стоит правда и 0, если ложь.

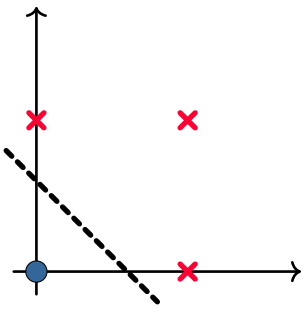


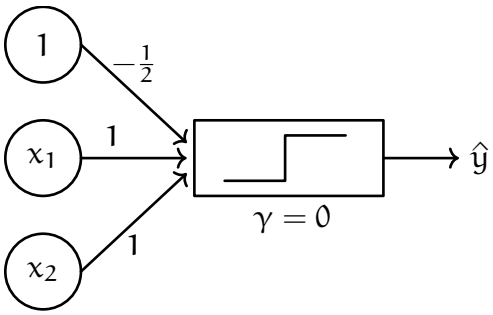


Если нарисовать все наши четыре точки на плоскости, становится ясно, что мы хотим отделить точку $(1, 1)$ от всех остальных. Сделать это можно практически любой линией. Например, в нейроне выше задана линия $x_2 = 1.5 - x_1$. Подойдёт и любая другая, отделяющая крест от точек. Пропустим ради приличия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

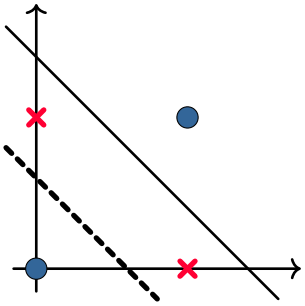
$$[-1.5 + 1 + 1 > 0] = [0.5 > 0] = 1$$
$$[-1.5 + 0 + 0 > 0] = [-1.5 > 0] = 0$$
$$[-1.5 + 0 + 1 > 0] = [-0.5 > 0] = 0$$
$$[-1.5 + 1 + 0 > 0] = [-0.5 > 0] = 0$$

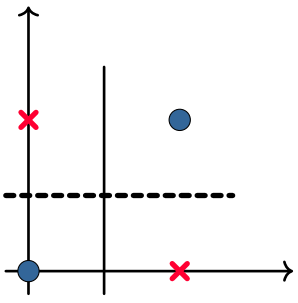
С объединением та же ситуация, только на этот раз линия должна пройти чуть ниже. Подойдёт $x_2 = 0.5 - x_1$.





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 задачки. Надо посмотреть первым слоем нейросетки, где мы оказались относительно каждой из линий, а вторым слоем соединить результаты.





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

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

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?

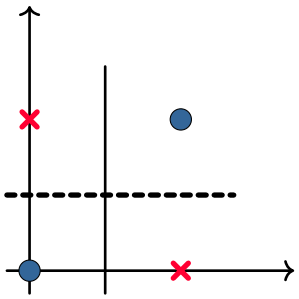
Решение:

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:

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

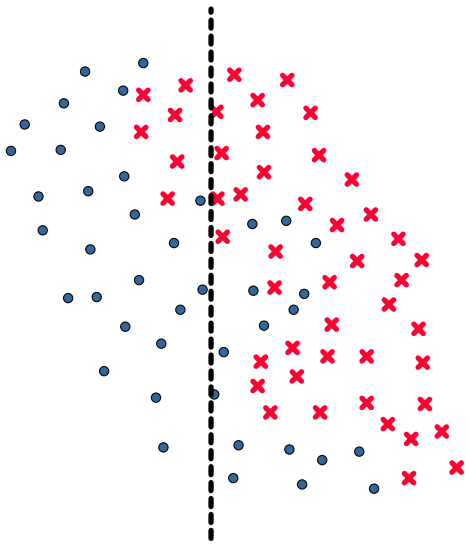
Получаем два решения. Прямую $x_2 = 0.5$ и прямую $x_1 = 0.5$.

Упражнение 7 (универсальный классификатор)

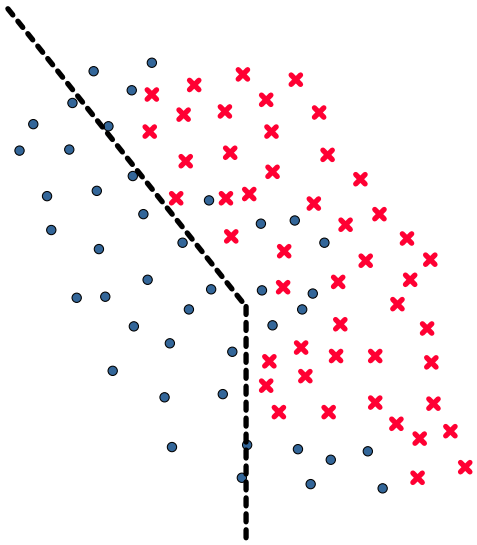
Маша задумалась о том, можно ли с помощью нейронной сетки с одним скрытым слоем и ступенчатой функцией активации решить абсолютно любую задачу классификации на два класса со сколь угодно большой точностью. Ей кажется, что да. Как это можно сделать?

Решение:

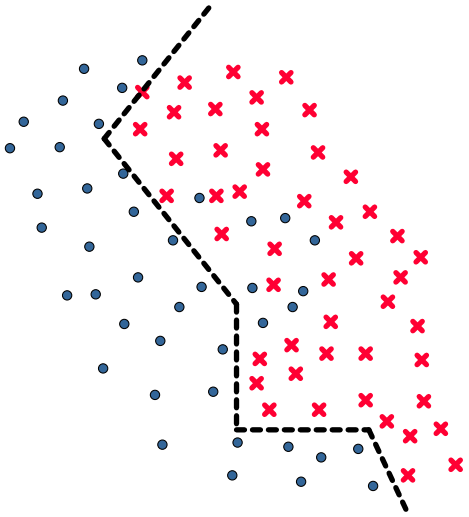
Сейчас мы докажем, что нейросеть может решить любую задачу классификации с любой точностью. Для начала попробуем решить задачку с картинки ниже одним нейроном:



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:



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 точек, и прогнозировать нужно 0:

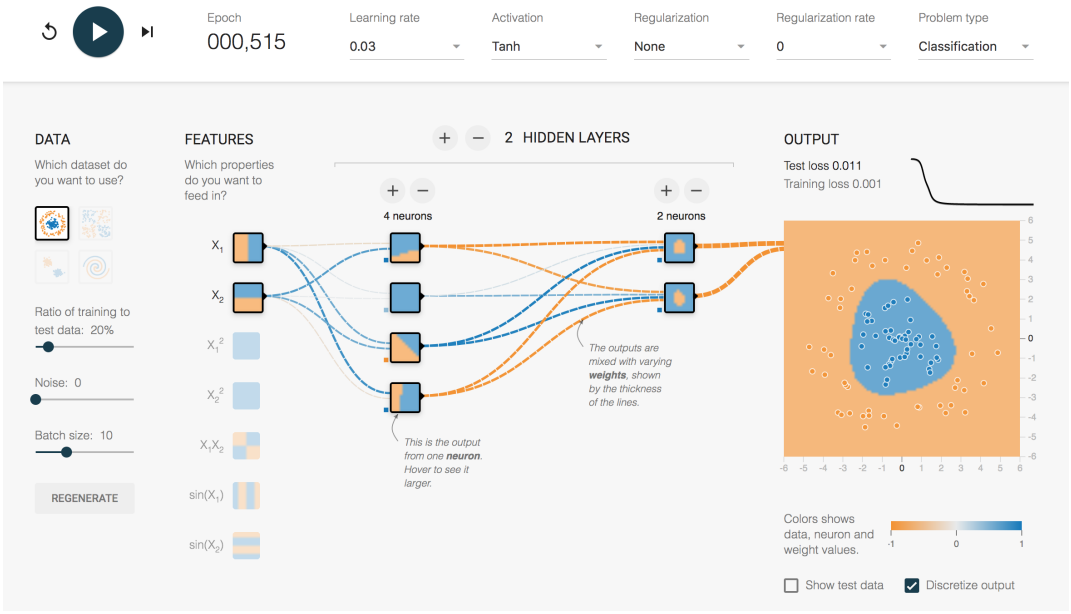
$$\hat{y} = [x_1 + x_2 + x_3 + x_4 + x_5 - 4.5 > 0].$$

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 играться с простенькими нейросетками и обучила для решения задачи классификации трёхслойного монстра.

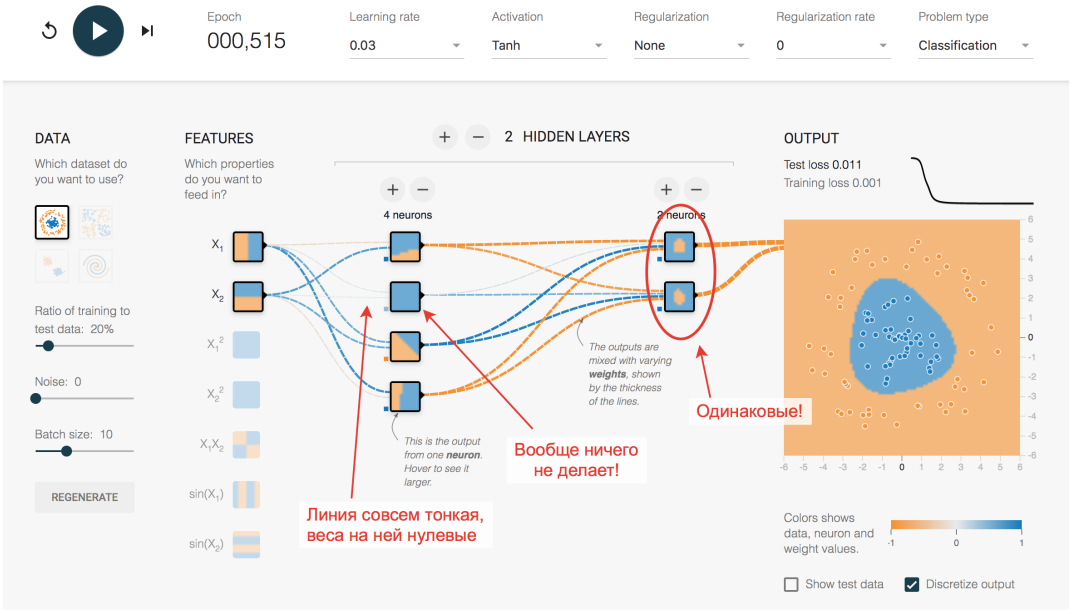
Голубым цветом обозначен первый класс, рыжим второй. Внутри каждого нейрона визуализирована та разделяющая поверхность, которую он выстраивает. Так, первый слой ищет разделяющую линию. Второй слой пытается из этих линий выстроить более сложные фигуры и так далее. Чем ярче связь между нейронами, тем больше 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.

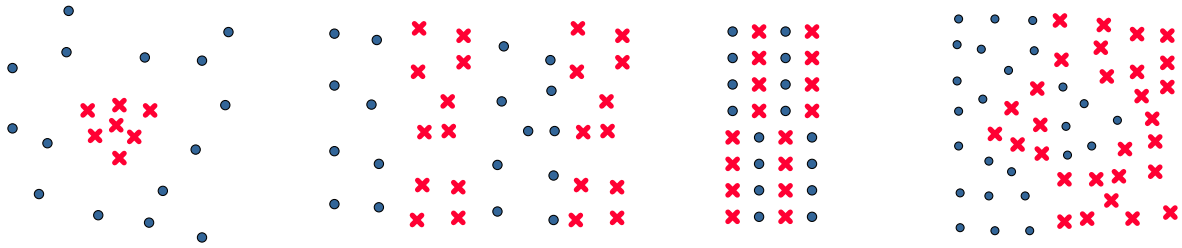


Для решения такой простой задачи классификации подойдёт более простая модель. Сколько минимально нужно нейронов, чтобы её решить вам и Маше предстоит выяснить в следующей

задаче.

Упражнение 9 (минималочка)

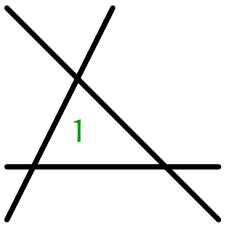
Шестилетняя сестрёнка ворвалась в квартиру Маши и разрисовала ей все обои:



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?

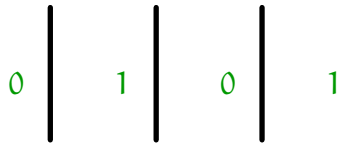
Решение:

а. Нам с помощью нейросетки надо выделить треугольник. Всё, что внутри будет относится к первому классу.



Получается 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. Посмотрите теперь на предыдущую задачу, сходите на сайт с демкой и постройте оптимальную нейросетку.

б. Понятно, что первый слой должен построить нам три линии. Это три нейрона.



Второй слой должен принять решение: в какой из полос мы оказались.

в. Перед нами две XoR задачи. На первом слое будем строить четыре линии.

г. Первый слой строит пять линий.

Упражнение 10 (универсальный регрессор)

Маша доказала Паше, что у неё всё в полном порядке с логикой. Теперь она собирается доказать ему, что с помощью однослойной нейронной сетки можно приблизить любую непрерывную функцию от одного аргумента $f(x)$ со сколь угодно большой точностью[9].

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:

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:

Картинка с двумя приближениями для функции

Попробуем смоделировать одну ступеньку.

Картинка ступеньки

Если x, для которого мы ищем $f(x)$ попадает в неё, мы будем приближать $f(x)$ этой ступенькой. Ступенька состоит из двух линий. Выходит, что она будет описываться двумя нейронами. Если мы внутри ступеньки, значит $a \leqslant x \geqslant a + h$. Пара нейронов должна сравнить x с a и $a + h$ и на основе этого принять решение.

Картинка двух нейронов

Можно записать попадание x в ступеньку в виде нейрона также как мы делали это в задачке с таблицами истинности:

$$1 - [x \leqslant a] - [x \geqslant a + h]$$

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.

Будем так действовать для каждой ступеньки. Мы попадём только в одну из них. Значит внутренний слой выплюнет на нас 1 только из одной ветки. Остаётся только решить

ко ко ко

2 50 оттенков градиентного спуска

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

3 Backpropagation

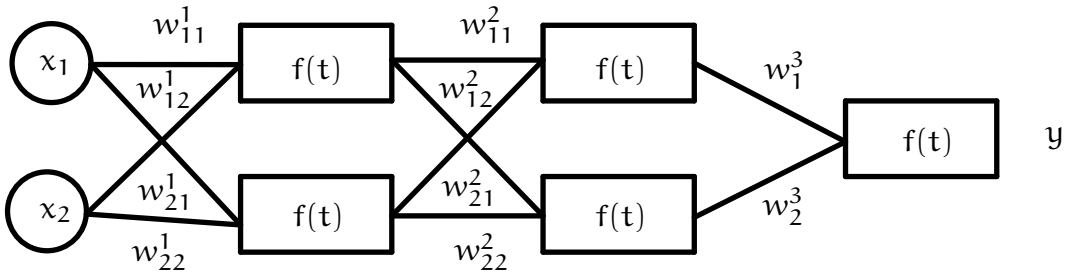
Что происходит, когда мы суём пальцы в розетку? Нас бьёт током! Мы делаем ошибку, и она распространяется по нашему телу назад.

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.[10]

Упражнение 2 (придумываем backpropagation)

Дана нейросетка:



а. Перепишите её как сложную функцию.

б. Запишите эту функцию в матричном виде.

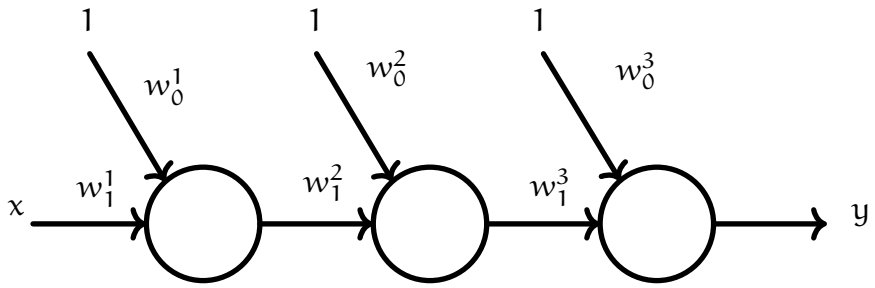
в. Предположим, что $L(W_1, W_2, W_3) = \frac{1}{2} \cdot (y - \hat{y})^2$ — функция потерь, где W_i — веса i—го слоя. Найдите производную функции L по всем весам W_i.

г. 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)

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

[10] По мотивам книги Николенко "Глубокое обучение"(стр. 79)



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 $f(t) = \mathrm{ReLU}(t) = \max(0; t)$? 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.

а. 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.

б. Найдите для первого слоя производную выхода по входу. При обратном движении по нейросетке, в первый слой пришёл накопленный градиент $(-1, 0)$. Каким будет новое накопленное значение градиента, которое выплюнет 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 выплюнет 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 выплюнет 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 выплюнет 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

Упражнение 1 У

Бандерлога три наблюдения[11], первое наблюдение — кит, остальные — муравьи. Киты кодируются $y_i = 1$, муравьи — $y_i = 0$. В качестве регрессоров Бандерлог берёт номера наблюдений $x_i = i$. После этого Бандерлог оценивает логистическую регрессию с константой.

а. Выпишите эмпирическую функцию риска, которую минимизирует Бандерлог;

б. При каких оценках коэффициентов логистической регрессии эта функция достигает своего минимума?

Упражнение 2

[11]Про другие приключения Бандерлога читай тут: https://github.com/bdemeshev/mlearn_pro/blob/master/mlearn_pro.pdf

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

a) 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 относится 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?

Упражнение 3 Б

андерлог чуть внимательнее присмотрелся к своему третьему наблюдению и понял, что это не кит, а бобёр. Теперь ему нужно решать задачу классификации на три класса. Он решил использовать для этого нейросеть с softmax-слоем на выходе. Предположим, что сетка обучилась и на двух новых наблюдениях, перед самым softmax-слоем она выплюнула $1, 2, 5$ и $2, 5, 1$.

а. Чему равны вероятности получить кита, муравья и бобра для обеих ситуаций?

б. Пусть первым был кит, а вторым бобёр. Чему будет равна logloss-ошибка?

Упражнение 4 И

ногда в функцию Softmax добавляют дополнительный параметр T, который называют температурой. Тогда она приобретает вид

$$f(z) = \frac{e^{\frac{z_i}{T}}}{\sum_{k=1}^{K} e^{\frac{z_k}{T}}}$$

Обычно это делается, когда с помощью нейросетки нужно сгенерировать какой-нибудь новый объект. Пусть у нас есть три класса. Наша нейросеть выдала на последнем слое числа $1, 2, 5$.

а. Какое итоговое распределение вероятностей мы получим, если $T = 10$?

б. А если $T = 1$?

в. А если $T = 0.1$?

г. Какое распределение получится при $T \to 0$?

д. А при $T \to \infty$?

е. Предположим, что объектов на порядок больше. Например, это реплики, которые Алиса может сказать вам в ответ на какую-то фразу. Понятное дело, что вашей фразе будет релевантно какое-то подмножество ответов. Какое значение температуры сэмплирования T смогут сделать реплики Алисы непредсказуемыми? А какие сделают их однотипными?

Упражнение 5 Ф

ункция $f(t) = \frac{e^t}{1+e^t}$ называется сигмоидой[12].

а. Что происходит при $t \to +\infty$? А при $t \to -\infty$?

б. Как связаны между собой $f(t)$ и $f(-t)$?

в. Как связаны между собой $f'(t)$ и $f'(-t)$?

г. Как связаны между собой $f(t)$ и $f'(t)$?

д. Найдите $f(0)$, $f'(0)$ и $\ln f(0)$.

е. Найдите обратную функцию $f^{-1}(t)$

ж. Как связаны между собой $\frac{d \ln f(t)}{dt}$ и $f(-t)$?

з. Постройте графики функций $f(t)$ и $f'(t)$.

и. Разложите $h(\beta_1, \beta_2) = \ln f(y_i(\beta_1 + \beta_2 x_i))$ в ряд Тейлора до второго порядка в окрестности точки $\beta_1 = 0$, $\beta_2 = 0$.

5 Регуляризаторы

6 Всего лишь кубики LEGO

7 Итоговый тест в стел Носко

а. Вопрос про батчнормализацию первым слоем вместо нормализации в предобработке.

[12] В этом всём тоже замешан один Бандерлог.