Омский Государственный ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Онлайн-курс

ВВЕДЕНИЕ В ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ

5. Задача регрессии для нейронных сетей

Автор: Шевляков Артём Николаевич

Задачи, решаемые нейросетями
Задача регрессии

Задачи предсказания

Постановка задачи

Есть множество объектов *M* с известными значениями признака *Y*. Найти (предсказать, оценить) значение признака *Y* для нового объекта *A*. Признак Y называется **целевым**.

Предсказываемый признак Y может быть

количественным	**меткой класса**
задача предсказания называется **задачей регрессии**	задача предсказания называется **задачей классификации**

Каждая из этих задач требует особой архитектуры нейронной сети.

Задача регрессии

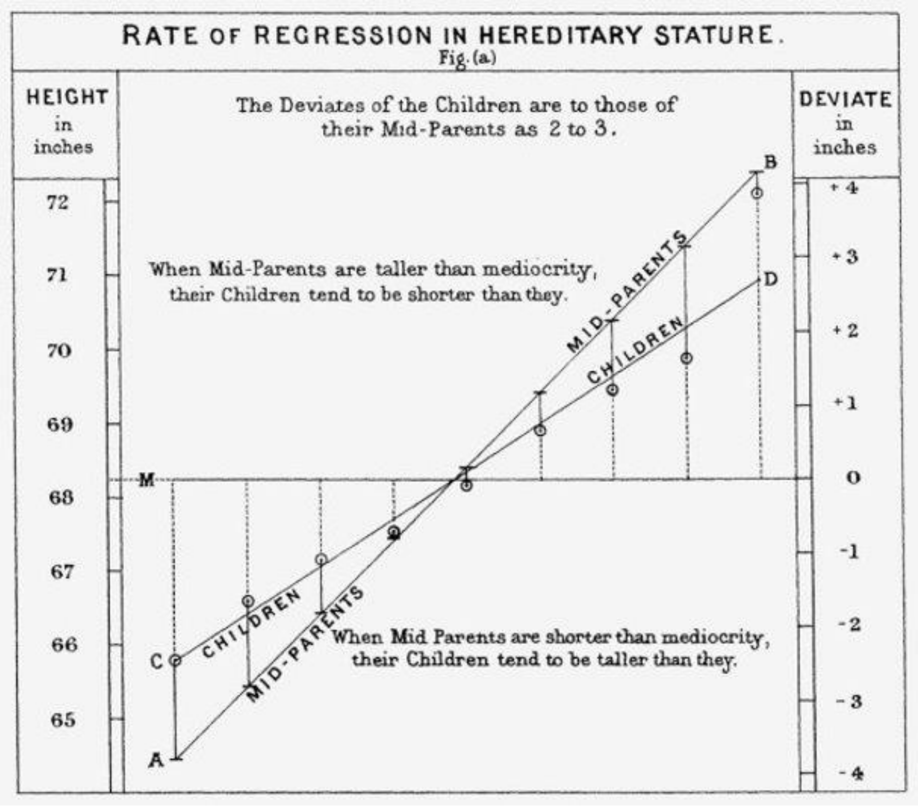
Примеры:

- Предсказать возраст человека по его фото (анализу крови).
- Подсчитать количество людей (машин) на фото.
- Предсказать курс доллара (температуру воздуха) на завтра.
- Предсказать рост ребенка по росту его родителей (это же классика!).

Откуда слово «регрессия»

Когда исследовали зависимость роста сына от роста его отца, то было замечено «рост сына приближался (**регрессировал**) к среднему росту мужчин».

Поскольку это была исторически первая задача предсказания, то эффект регрессии дал название целому классу задач.



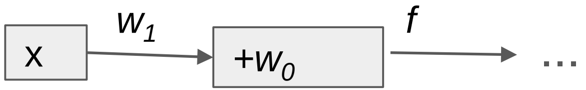
Какую НС использовать для регрессии?

Нужно построить НС и так подогнать ее веса, чтобы функция $F_{NN}(x)$ давала более-менее точное значение для любого объекта x (x может быть не только числом, а числовым вектором или объектом более сложной природы: фото, текст итд).

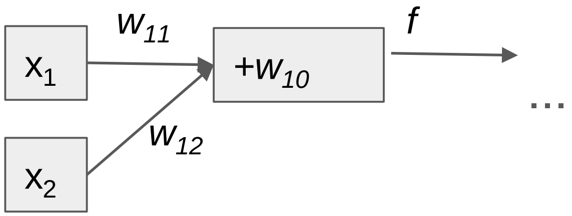
Какую НС использовать для регрессии?

Нужно построить НС и так подогнать ее веса, чтобы функция $F_{NN}(x)$ давала более-менее точное значение для любого объекта x (x может быть не только числом, а числовым вектором или объектом более сложной природы: фото, текст итд).

Входной слой НС соответствует размерности объектов x:



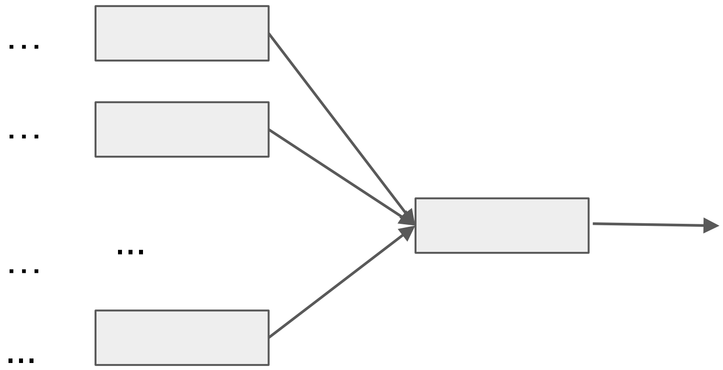
если объекты x – это числа



если объекты x – это векторы длины 2. ИТД

А что насчет выходного слоя?

Для задачи регрессии выходной слой состоит из одного нейрона



это число (равное $F_{NN}(x)$) – и есть ответ НС

Онлайн-курс

ВВЕДЕНИЕ В ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ

5. Задача регрессии для нейронных сетей

План решения задачи регрессии с помощью нейросети

План

(аналогичный план имеет место и для задачи классификации).

1. Взять **тренировочную выборку** (ТВ), то есть набор объектов с известными значениями целевого признака Y. Нейронная сеть в дальнейшем должна восстановить зависимость между нецелевыми признаками и целевым признаком.
2. Задать основные параметры нейронной сети: количество слоёв, количество нейронов на каждом слое, тип связи между слоями и т.д.

3. Выписать функцию $F_{NN}(x)$. Это выражение будет содержать вхождения весов w_i.

План

4. Пусть y_i – точное значение целевого признака Y для i-го объекта из тренировочной выборки, $F_{NN}(X_i)$ – значение функции НС для i-го объекта из тренировочной выборки.

5. Составим квадрат разности $(F_{NN}(X_i)\text{-}y_i)^2$.
А затем и сумму по всем объектам тренировочной выборки:

$$L(w)=(F_{NN}(X_1)-y_1)^2+(F_{NN}(X_2)-y_2)^2+\ldots+(F_{NN}(X_m)-y_m)^2$$

– это суммарная ошибка НС на тренировочной выборке.
$L(w)$ называется функцией потерь для задачи регрессии.

6. Функция потерь $L(w)$ содержит вхождения букв w_i (весов НС).
Относительно этих переменных мы находим точку минимума функции $L(w)$.

План

7. Точка минимума определяет оптимальные веса НС.

8. Присваиваем весам НС найденные оптимальные значения.

9. (Боевое применение НС). Пусть теперь объект А не принадлежит тренировочной выборке. Тогда значение целевого признака Y для него вычисляется (предсказывается) по формуле $F_{NN}(A)$ с использованием найденных ранее оптимальных значений весов.

Тренировочная выборка (ТВ)

В нашем курсе мы предполагаем, что все объекты для задачи предсказания описываются набором числовых признаков.
Иными словами, тренировочную выборку можно представить в виде таблицы.

Объект	Признак P_1	...	Признак P_n	Целевой признак Y
A_1	p_{11}	...	p_{1n}	y_1
A_2	p_{21}	...	p_{2n}	y_2
...	...	...	...	...
A_m	p_{m1}	...	p_{mn}	y_m

Архитектура нейронной сети

Тренировочная выборка (ТВ) строго определяет размер входного слоя. **Если у объектов ТВ *n* нецелевых признаков, то размер входного слоя равен *n*.**

Количество внутренних слоев – на ваше усмотрение. Слишком много (да-да: переобучение) и слишком мало слоев плохо.

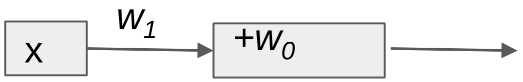
Выходной слой из одного нейрона.

Пример

Пусть имеется ТВ:

Объекты	x	Y
A_1	-1	1
A_2	0	0
A_3	1	1
A_4	2	4

Определимся с архитектурой НС:



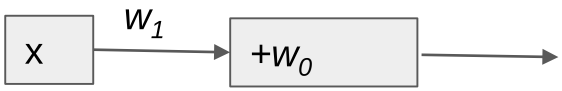
она настолько простая,

что нету даже ФА!

Пример

Посмотрим, сумеет ли такая простая НС найти правильную зависимость между x и Y.

Выпишем выражение $F_{NN}(x)$: $F_{NN}(x)=w_1x+w_0$



Объекты	x	$F_{NN}(A_i)$	Y
A_1	-1	$-w_1+w_0$	1
A_2	0	w_0	0
A_3	1	w_1+w_0	1
A_4	2	$2w_1+w_0$	4

Пример

Составляем квадраты ошибки:

Объекты	x	$F_{NN}(A_i)$	Y	$(F_{NN}(A_i)-y_i)^2$
A_1	-1	$-w_1+w_0$	1	$(-w_1+w_0-1)^2$
A_2	0	w_0	0	$(w_0-0)^2$
A_3	1	w_1+w_0	1	$(w_1+w_0-1)^2$
A_4	2	$2w_1+w_0$	4	$(2w_1+w_0-4)^2$

Составляем функцию потерь:

$L(w)=(-w_1+w_0-1)^2+(w_0-0)^2+(w_1+w_0-1)^2+(2w_1+w_0-4)^2.$

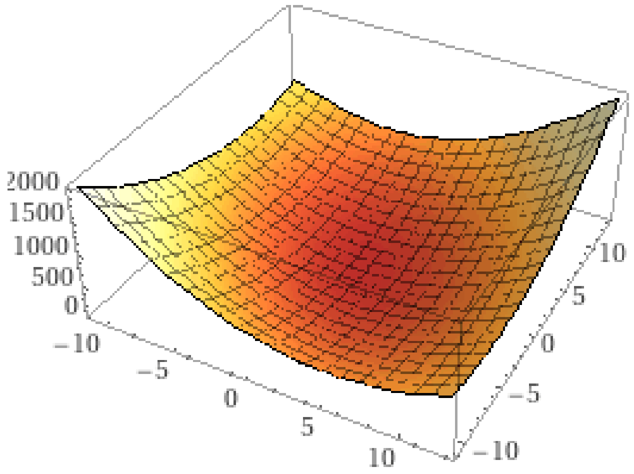
Пример

Надо искать минимум функции потерь

$L(w)=(-w_1+w_0-1)^2+(w_0-0)^2+(w_1+w_0-1)^2+(2w_1+w_0-4)^2$

(то есть нужно найти оптимальные значения w_0, w_1).

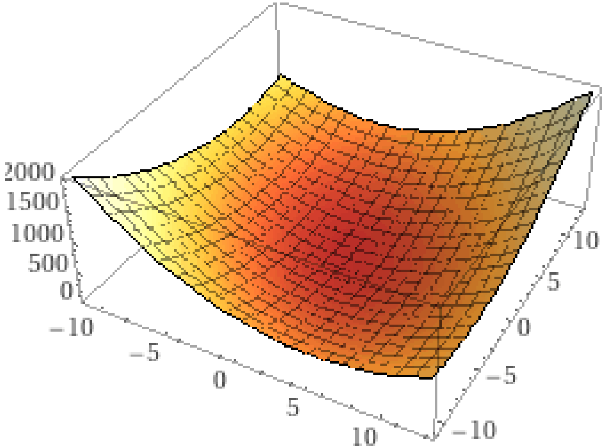
$L(w)=6w_1^2+4w_0^2+4w_1w_0-16w_1-12w_0+18$



Пример

$L(w)=6w_1^2+4w_0^2+4w_1w_0-16w_1-12w_0+18$

Точку минимума можно найти с помощью ГС. Она равна $w_1=w_0=1$. Локальный минимум тут единственный, поэтому ГС придёт к нему из любой начальной точки.



Пример

У нас получилась НС с функцией $F_{NN}(x)=x+1$.

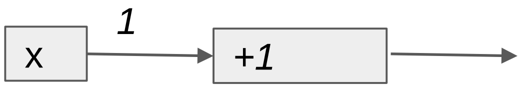
А как это использовать на практике?

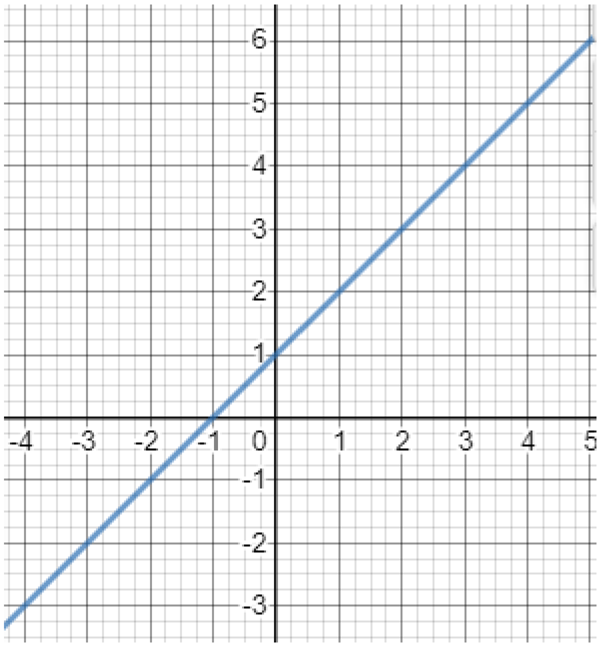
Допустим, к нам попадает новый объект A_5 со значением признака $x=3$.

Тогда НС выдаст предсказание: $F_{NN}(3)=3+1=4$.

Конечно, мы ожидали ответ $3^2=9$, но архитектура нашей НС принципиально не может уловить такую сложную для нее зависимость.

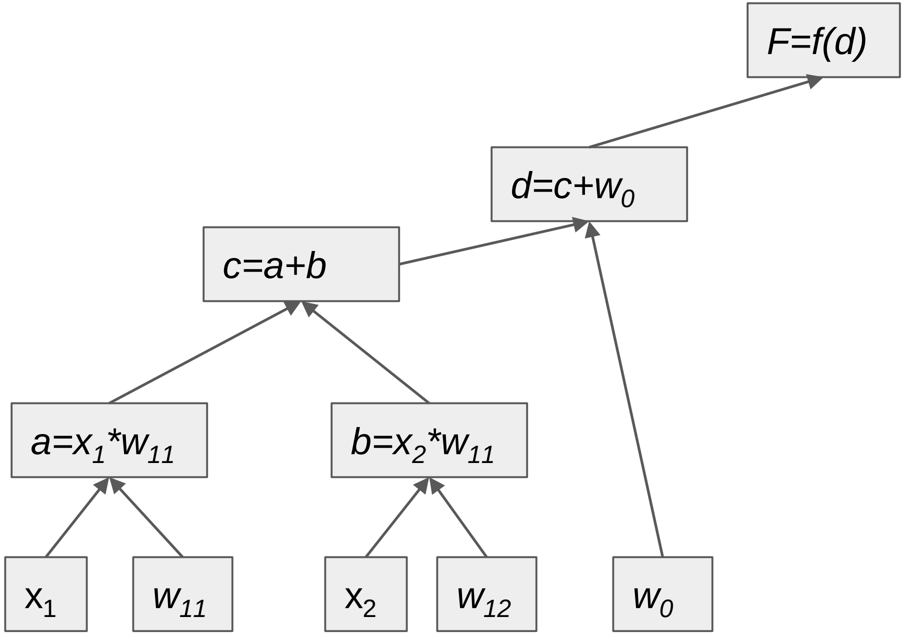
Можно взять НС с более сложной архитектурой и попытаться восстановить зависимость с помощью нее.

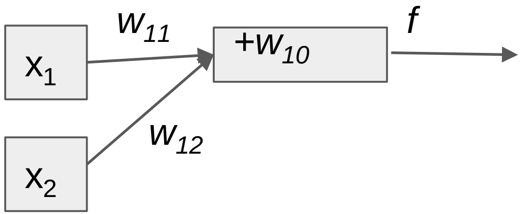




Комментарии к примеру

Вычисление градиента во время ГС осуществляется с помощью графов вычислений, поскольку сама нейросеть представляет собой граф.





Онлайн-курс

ВВЕДЕНИЕ В ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ

5. Задача регрессии для нейронных сетей

Автор: Шевляков Артём Николаевич

Улучшения градиентного спуска для нейросетей

ГС – универсальный метод поиска минимума, но...

функция потерь $L(w)$ имеет весьма специфический вид, можно этим воспользоваться.

Итак, $L(w)$ для задачи регрессии – это сумма квадратов

$$L(w)=(F_{NN}(X_1)-y_1)^2+(F_{NN}(X_2)-y_2)^2+\ldots+(F_{NN}(X_m)-y_m)^2$$

Каждое слагаемое этой суммы соответствует некоторому элементу ТВ.

Это свойство позволяет улучшить алгоритм ГС.

Стохастический ГС (СГС)

1. Случайным образом выберем из суммы $L(w)=(F_{NN}(X_1)-y_1)^2+(F_{NN}(X_2)-y_2)^2+\ldots+(F_{NN}(X_m)-y_m)^2$

   одно слагаемое $F=(F_{NN}(X_i)-y_i)^2$

1. **Осуществим одну итерацию ГС для функции F.** Это приведёт к получению нового набора весов нашей НС. Возвращаемся на п.1

Прим.
ГС на п.2 может осуществляться как по классической формуле, так и по «продвинутым» формулам (Momentum, Adagrad, Adam...)

Плюсы: СГС, как правило, находит более глубокие минимумы, чем классический ГС. Мы фактически заменяем минимизацию большой функции на минимизацию нескольких простых функций.

1ая итерация СГС

Итерация СГС начинается со случайного выбора слагаемого в $L(w)$.

Допустим, что было выбрано первое слагаемое:

$L_1(w)=(-w_1+w_0-1)^2$ – делаем для него один шаг классического ГС:

$dL/dw_1=-2(-w_1+w_0-1)$, $dL/dw_0=2(-w_1+w_0-1)$

$dL/dw_1(0,0)=2$ $dL/dw_0(0,0)=-2$

$a_1=(0,0)-0.1*(2,-2)=(-0.2, 0.2)$ – вот новая точка спуска.

После этого начинается вторая итерация СГС...

2ая итерация СГС

Допустим, что было выбрано второе слагаемое:

$L_2(w)=(w_0-0)^2$ – делаем для него один шаг классического ГС:

$dL/dw_1=0,$ $dL/dw_0=2w_0$

$dL/dw_1(-0.2,0.2)=0$ $dL/dw_0(-0.2,0.2)=0.4$

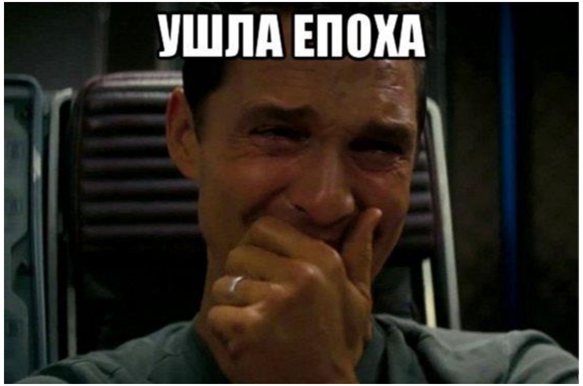
$a_2=(-0.2,0.2)-0.1*(0,0.4)=(-0.2,\ 0.16)$ – вот новая точка спуска.

После этого начинается третья итерация СГС… и так далее

Эпоха – что это?

Обычно поступают так: в случайном порядке перебирают все слагаемые в функции потерь (их число совпадает с объемом тренировочной выборки) и для выбранного слагаемого вычисляют градиент и производят сдвиг по формуле ГС. То есть делают СГС.

Когда будут просмотрены все слагаемые в функции потерь, то говорят, что прошла одна эпоха обучения. Число итераций в эпохе равно размеру ТВ.



Потом начинается следующая эпоха и перебор слагаемых (в другом случайном порядке) повторяется.

Количество эпох является одним из входных параметров обучения НС.

СГС по мини-батчам

Мы рассмотрели две крайности в способах минимизации функции потерь:

- **«обычный» ГС** (когда минимизируется вся функция потерь);
- **СГС** (когда на каждой итерации спуска минимизируется одно случайное слагаемое).

Но как всегда бывает: **истина посередине**, то есть оптимальна промежуточная стратегия – **СГС по мини-батчам**.

СГС по мини-батчам

Фиксируется размер батча (группы) *k*. После этого на каждом шаге происходит выбор случайных k слагаемых из функции потерь и вычисляется градиент по сумме этих k слагаемых.

При *k=1* – это обычный СГС.
При k={объем тренировочной выборки} – это минимизация всей функции потерь. То есть это - классический ГС функции потерь.

Пример. СГС по мини-батчам

Возьмем известную ТВ:

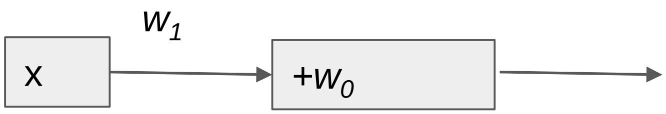
Объекты	x	Y
A_1	-1	1
A_2	0	0
A_3	1	1
A_4	2	4

Будем использовать прежнюю НС с функцией
$F_{NN}(x)=w_1x+w_0$

Обычная функция потерь:

$$L(w)=(-w_1+w_0-1)^2+(w_0-0)^2+(w_1+w_0-1)^2+(2w_1+w_0-4)^2$$

Параметры ГС:
шаг $h=0.1$, начальная точка спуска $a_0=(0,0)$, $k=2$.



СГС по мини-батчам. 1ая итерация

$L(w)=(-w_1+w_0-1)^2+(w_0-0)^2+(w_1+w_0-1)^2+(2w_1+w_0-4)^2$ – это функция потерь по всей ТВ.

Пусть на 1й итерации были случайно выбраны 1й и 2й объекты ТВ. Мы получим «обрезанную» функцию потерь:

$L_1(w)=(-w_1+w_0-1)^2+(w_0-0)^2$

И делаем один шаг ГС для этой функции потерь. Для этого находим ЧП и их значения в точке $a_0=(0,0)$.

$$\frac{\partial L_1}{\partial w_1} = -2(-w_1 + w_0 - 1),$$

$$\frac{\partial L_1}{\partial w_0} = 2(-w_1 + w_0 - 1) + 2w_0,$$

$$\frac{\partial L_1}{\partial w_1}(a_0) = 2, \ \frac{\partial L_1}{\partial w_0}(a_0) = -2$$

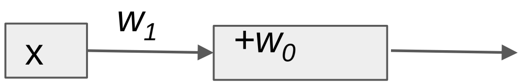
Пример СГС

Объекты	x	Y
A_1	-1	1
A_2	0	0
A_3	1	1
A_4	2	4

Применим СГС для уже знакомой нам ТВ и НС.

Функция потерь равна:

$$L(w)=(-w_1+w_0-1)^2+(w_0-0)^2+(w_1+w_0-1)^2+(2w_1+w_0-4)^2.$$



Будем ее минимизировать с помощь СГС. Начальная точка (0,0), шаг h=0.1

СГС по мини-батчам. 1ая итерация

Тогда новая точка спуска равна:

$$a_1 = (0, 0) - 0.1(2, -2) = (-0.2, 0.2)$$

СГС по мини-батчам. 2ая итерация

$L(w)=(-w_1+w_0-1)^2+(w_0-0)^2+(w_1+w_0-1)^2+(2w_1+w_0-4)^2$ – это функция потерь по всей ТВ.

Пусть теперь на 2й итерации были случайно выбраны 3й и 4й объекты ТВ. Мы получим «обрезанную» функцию потерь:

$L_2(w)=(w_1+w_0-1)^2+(2w_1+w_0-4)^2$

И делаем один шаг ГС для этой функции потерь. Для этого находим ЧП и их значения в точке $a_1=(-0.2,0.2)$.

$$\frac{\partial L_2}{\partial w_1} = 2(w_1 + w_0 - 1) + 4(2w_1 + w_0 - 4), \qquad \frac{\partial L_2}{\partial w_1}(a_1) = -18.8,$$

$$\frac{\partial L_2}{\partial w_0} = 2(w_1 + w_0 - 1) + 2(2w_1 + w_0 - 4) \qquad \frac{\partial L_2}{\partial w_0}(a_1) = -10.4$$

СГС по мини-батчам. 2ая итерация

Тогда новая точка спуска равна:

$$a_2 = (-0.2, 0.2) - 0.1(-18.8, -10.4) = (-0.12, 1.24)$$

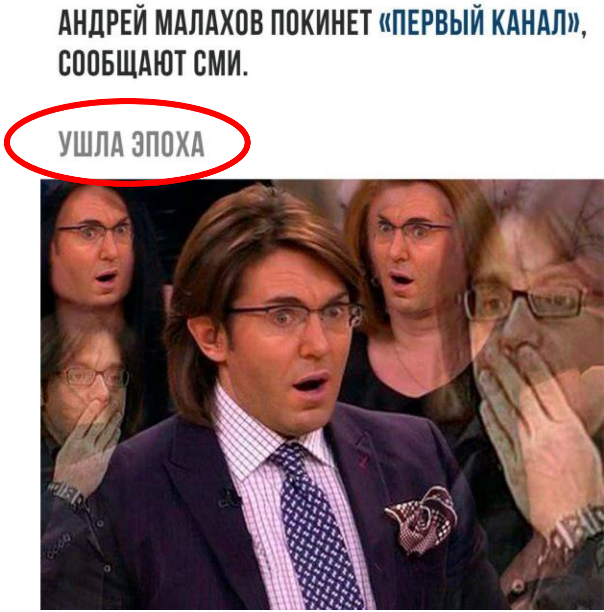
$$\frac{\partial L_2}{\partial w_1}(a_1) = -18.8,$$

$$\frac{\partial L_2}{\partial w_0}(a_1) = -10.4$$

И так далее.

Эпоха для мини-батчей

Пусть k – размер мини-батча. Тогда объекты ТВ случайным образом разбиваются на m/k батчей (m — объем ТВ). Далее делается m/k шагов ГС. Эпоха обучения на этом заканчивается.



После этого происходит новое случайное разбиение ТВ на m/k батчей и т.д.

Онлайн-курс

ВВЕДЕНИЕ В ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ

5. Задача регрессии для нейронных сетей

Выводы

Выводы:

- Мы дали определение задачи регрессии и обсудили план ее решения.
- Был рассмотрен пример тренировки НС для задачи регрессии.
- Мы обсудили новые способы улучшения ГС для функции потерь в задаче регрессии: СГС и СГС по мини-батчам.

Онлайн-курс

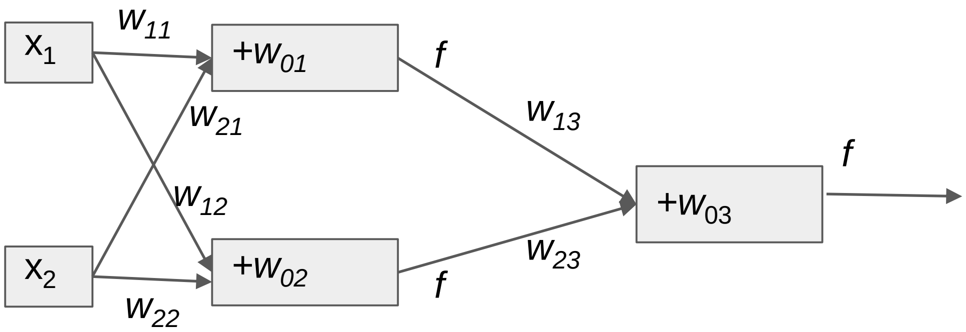
ВВЕДЕНИЕ В ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ

5. Задача регрессии для нейронных сетей

Тренировка более сложной нейронной сети

Вы действительно этого хотите?

Вот вам нейросеть посложнее:

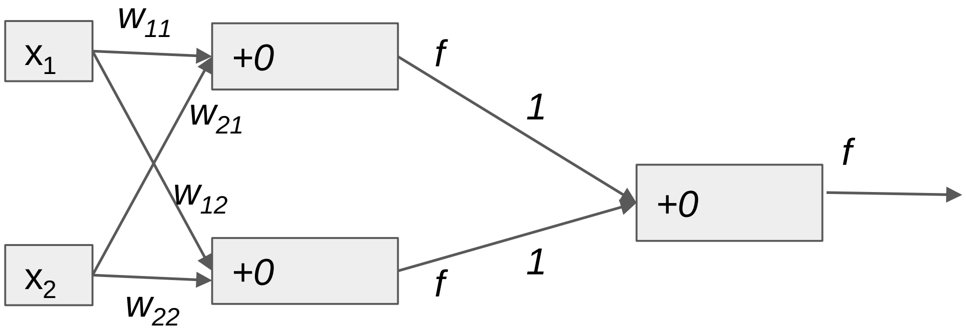


$$F_{NN}(x_1, x_2) = f(w_{13} f(w_{11} x_1 + w_{21} x_2 + w_{01}) + w_{23} f(w_{12} x_1 + w_{22} x_2 + w_{02}) + w_{03})$$

$$f(x) = Relu(x)$$

Немножко упростим задачу

Зафиксируем некоторые веса,
и будем тренировать лишь оставшиеся.



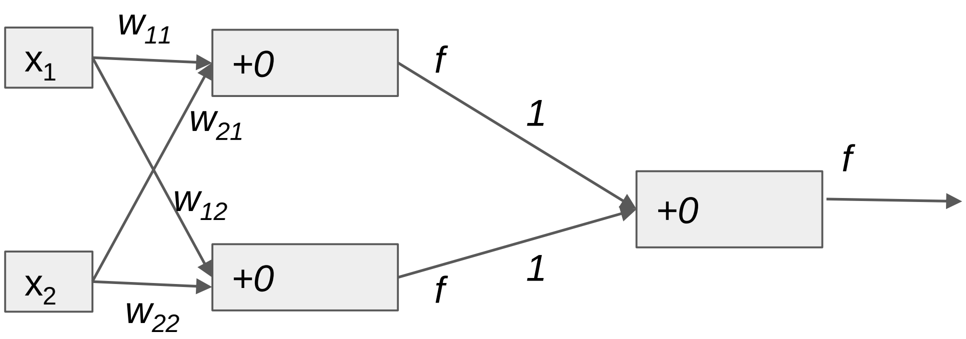
$$F_{NN}(x_1, x_2) = f(f(w_{11}x_1 + w_{21}x_2) + f(w_{12}x_1 + w_{22}x_2))$$

$$f(x) = Relu(x)$$

ТВ: xor-функция

Будем стремиться хотя бы к тому, чтобы на ТВ нейросеть не сильно бы ошибалась

x_1	x_2	Y
0	0	0
0	1	1
1	0	1
1	1	0



$$F_{NN}(x_1, x_2) = f(f(w_{11}x_1 + w_{21}x_2) + f(w_{12}x_1 + w_{22}x_2))$$

$$L(w) = (F_{NN}(0,0) - 0)^2 + (F_{NN}(0,1) - 1)^2 + (F_{NN}(1,0) - 1)^2 + (F_{NN}(1,1) - 0)^2$$

Расчёты #1

$$F_{NN}(x_1, x_2) = f(f(w_{11}x_1 + w_{21}x_2) + f(w_{12}x_1 + w_{22}x_2))$$

$$L(w) = (F_{NN}(0,0) - 0)^2 + (F_{NN}(0,1) - 1)^2 + (F_{NN}(1,0) - 1)^2 + (F_{NN}(1,1) - 0)^2$$

Текущие значения весов (точка *a_0*):

$$w_{11} = 0.5, w_{12} = -0.5, w_{21} = -0.5, w_{22} = 0.5,$$

$$F(0,0) = 0,$$

$$F(0,1) = f(f(w_{21}) + f(w_{22})),$$

$$F(1,0) = f(f(w_{11}) + f(w_{12})),$$

$$F(1,1) = f(f(w_{11} + w_{21}) + f(w_{12} + w_{22}))$$

$$\partial F(0,0)/\partial w_{11} = 0,$$

$$\partial F(0,1)/\partial w_{11} = 0,$$

$$\partial F(1,0)/\partial w_{11} = f'(f(w_{11}) + f(w_{12}))f'(w_{11}),$$

$$\partial F(1,1)/\partial w_{11} = f'(f(w_{11} + w_{21}) + f(w_{12} + w_{22}))f'(w_{11} + w_{21})$$

Расчёты #2

$$L(w) = (F_{NN}(0,0) - 0)^2 + (F_{NN}(0,1) - 1)^2 + (F_{NN}(1,0) - 1)^2 + (F_{NN}(1,1) - 0)^2$$

$$\partial L/\partial w_{11} = 2(\partial F(0,0)/\partial w_{11})(F(0,0) - 0) + 2(\partial F(0,1)/\partial w_{11})(F(0,1) - 1) + 2(\partial F(1,0)/\partial w_{11})(F(1,0) - 1) + 2(\partial F(1,1)/\partial w_{11})(F(1,1) - 0)$$

В текущей точке получаем:

$$w_{11} = 0.5, w_{12} = -0.5, w_{21} = -0.5, w_{22} = 0.5$$

$$\begin{aligned} F(0,0) &= 0, \\ F(0,1) &= 0.5, \\ F(1,0) &= 0.5, \\ F(1,1) &= 0, \\ \partial F(0,0)/\partial w_{11} &= 0, \\ \partial F(0,1)/\partial w_{11} &= 0, \\ \partial F(1,0)/\partial w_{11} &= 1, \\ \partial F(1,1)/\partial w_{11} &= 0 \end{aligned}$$

$$\partial L/\partial w_{11}(a_0) = 0 + 0 + 2 * 1 * (-0.5) + 0 = -1$$

Расчёты #3

$$\partial L/\partial w_{11}(a_0) = 0 + 0 + 2 * 1 * (-0.5) + 0 = -1$$

Можно уже узнать новое значение веса w_{11}
(не дожидаясь предварительных вычислений по другим весам).

$$w_{11} := 0.5 - 0.1(-1) = 0.6$$

То есть НС хочет увеличивать этот вес!

Теперь повторим все вычисления про вес w_{12}.

Расчёты #4

$$L(w) = (F_{NN}(0,0) - 0)^2 + (F_{NN}(0,1) - 1)^2 + (F_{NN}(1,0) - 1)^2 + (F_{NN}(1,1) - 0)^2$$

$$\partial L/\partial w_{12} = 2(\partial F(0,0)/\partial w_{12})(F(0,0) - 0) + 2(\partial F(0,1)/\partial w_{12})(F(0,1) - 1) + 2(\partial F(1,0)/\partial w_{12})(F(1,0) - 1) + 2(\partial F(1,1)/\partial w_{12})(F(1,1) - 0)$$

$$F(0,0) = 0,$$
$$F(0,1) = 0.5,$$
$$F(1,0) = 0.5,$$
$$F(1,1) = 0,$$
$$\partial F(0,0)/\partial w_{12} = 0,$$
$$\partial F(0,1)/\partial w_{12} = 0,$$
$$\partial F(1,0)/\partial w_{12} = f'(f(w_{11}) + f(w_{12}))f'(w_{12}),$$
$$\partial F(1,1)/\partial w_{12} = f'(f(w_{11} + w_{21}) + f(w_{12} + w_{22}))f'(w_{12} + w_{22})$$

Расчёты #5

В текущей точке получаем:

$$w_{11} = 0.5, w_{12} = -0.5, w_{21} = -0.5, w_{22} = 0.5$$

$$\begin{aligned}
F(0,0) &= 0, \\
F(0,1) &= 0.5, \\
F(1,0) &= 0.5, \\
F(1,1) &= 0, \\
\partial F(0,0)/\partial w_{12} &= 0, \\
\partial F(0,1)/\partial w_{12} &= 0, \\
\partial F(1,0)/\partial w_{12} &= 0, \\
\partial F(1,1)/\partial w_{12} &= 0
\end{aligned}$$

Тогда ЧП функции потерь равна:

$$\partial L/\partial w_{12} = 2(\partial F(0,0)/\partial w_{12})(F(0,0) - 0) + 2(\partial F(0,1)/\partial w_{12})(F(0,1) - 1) + 2(\partial F(1,0)/\partial w_{12})(F(1,0) - 1) + 2(\partial F(1,1)/\partial w_{12})(F(1,1) - 0)$$

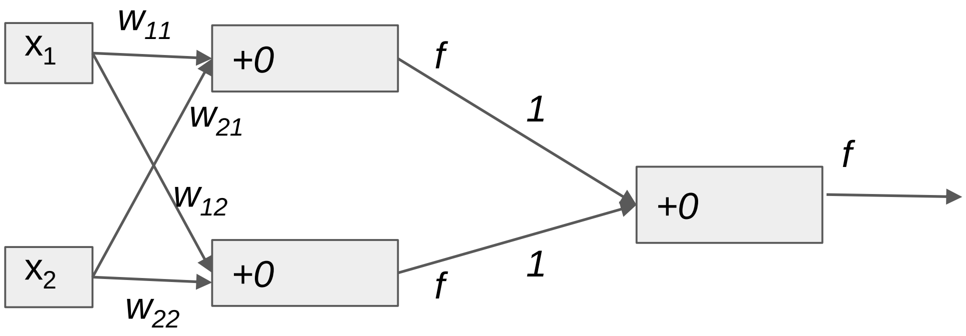
$$\partial L/\partial w_{12} = 0 + 0 + 0 + 0 = 0$$

Расчёты #6

Так как ЧП функции потерь по весу w_{12} равна 0, то значение этого веса не изменится.

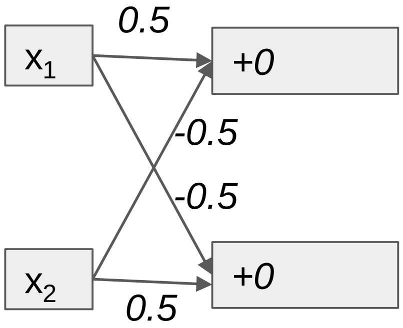
$$w_{12} := -0.5 - 0.1 * 0 = -0.5$$

С весами w_{21}, w_{22} все аналогично. Вес w_{22} станет равным 0.6, вес w_{21} останется равным -0.5.

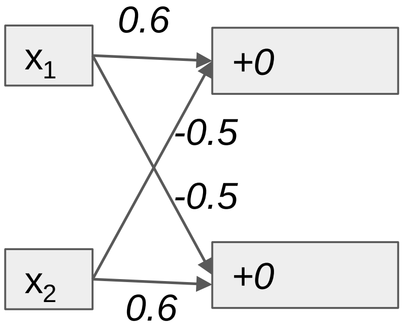


К каким значениям стремятся веса?

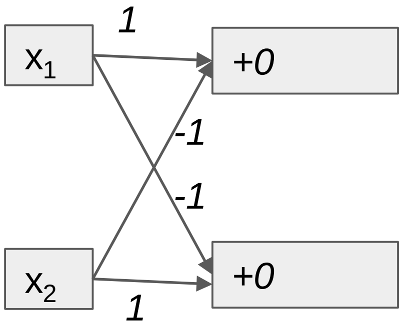
Было:



Стало:



В конце:



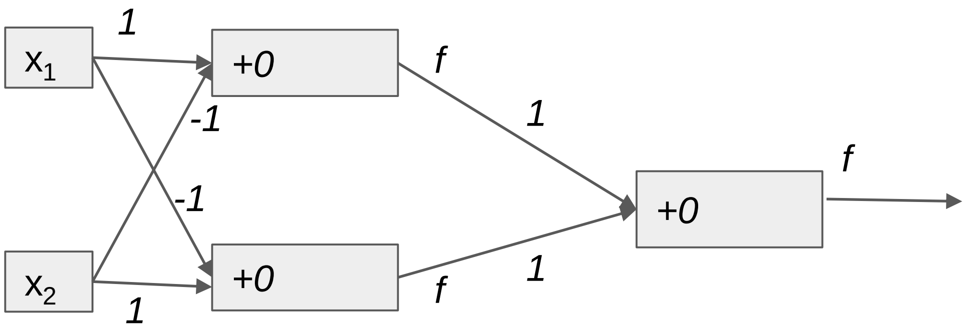
В итоге будет натренирована НС, вычисляющая функцию

$F_{NN}(x_1,x_2)=f(f(x_1-x_2)+f(x_2-x_1))$.

Посмотрим, насколько её значения согласуются с ответами из ТВ.

Точность на ТВ

$$F_{NN}(x_1,x_2)=f(f(x_1-x_2)+f(x_2-x_1))$$



x_1	x_2	Y	F_{NN}
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

$$F_{NN}(0,0)=f(f(0)+f(0))=0,$$

$$F_{NN}(0,1)=f(f(-1)+f(1))=1,$$

$$F_{NN}(1,0)=f(f(1)+f(-1))=1,$$

$$F_{NN}(1,1)=f(f(0)+f(0))=0,$$

А кому не лень, можете сделать 2ую итерацию ГС