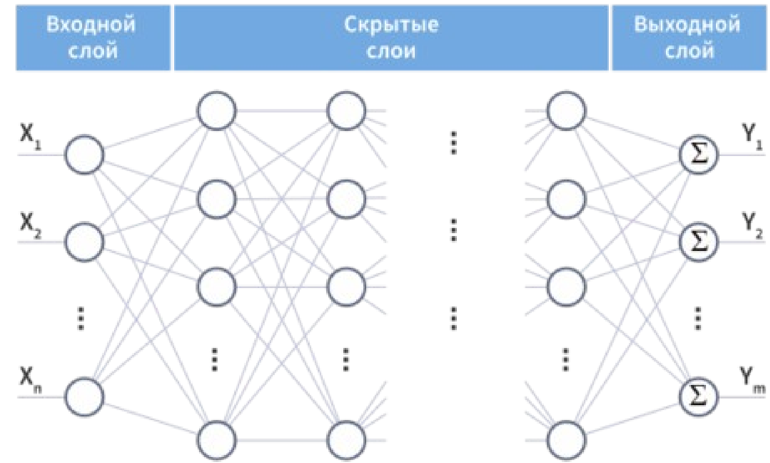
Машинное обучение в астрофизике

Лекция 3

Обратное распространение ошибок

Искусственные нейронные сети (ИНС)

- Многослойные (глубокие, deep) НС.
  - входной слой
  - несколько скрытых слоем (hidden layers)
  - выходной слой

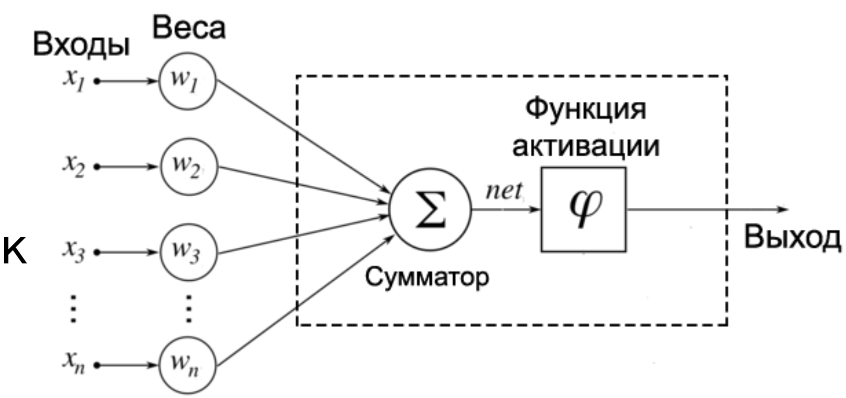


- Каждый узел — это искусственный нейрон, состоящий из сумматора и функции активации.
- Искусственный нейрон — это такая функция f: $R^n \to R$, которая преобразует несколько входных параметров в один выходной:

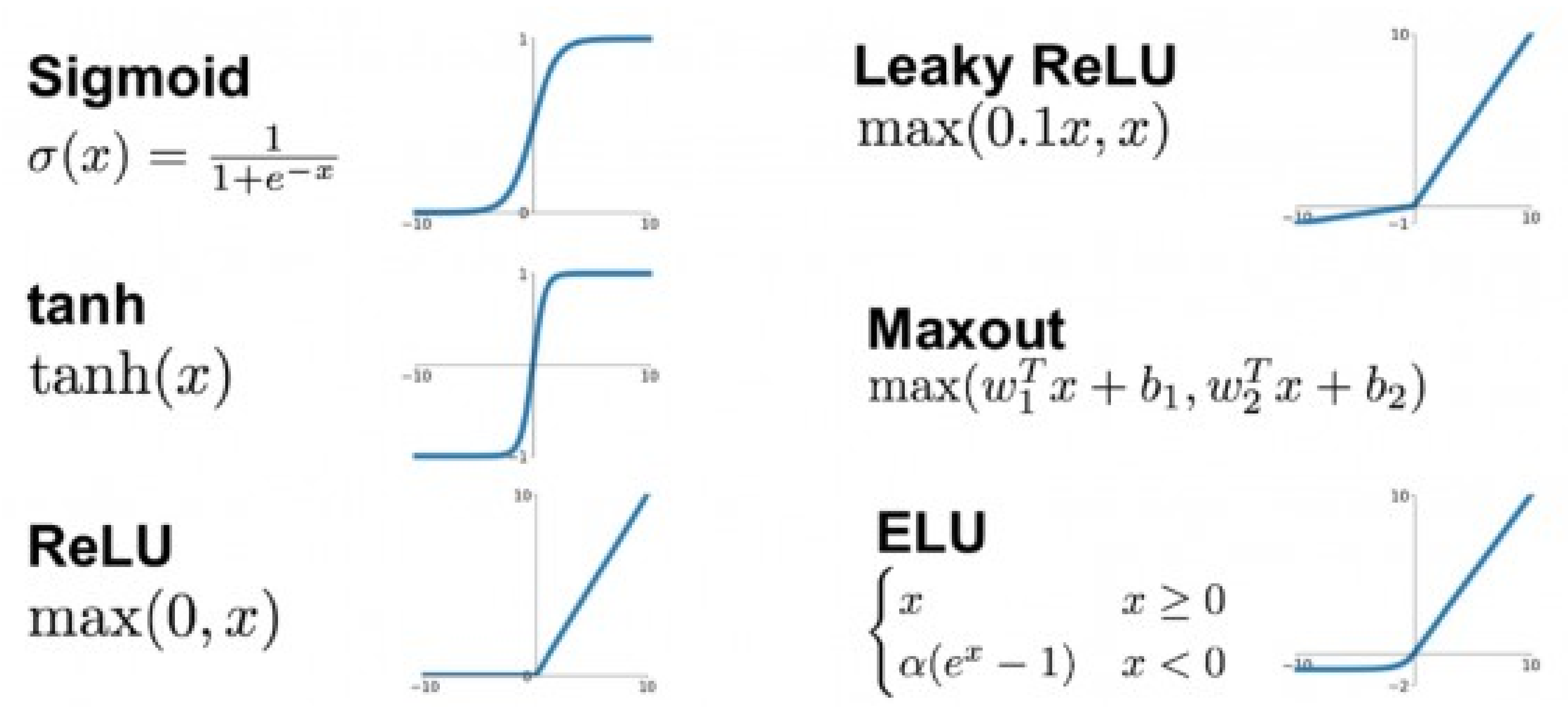
  $y=f(sum(i,w_i*x_i+b)), i=1..n$

  В матричном виде это можно записать как

  $y=f(W^T*X+b), W^T=\{w_1,..,w_n\}, X^T=\{x_1,..,x_n\},$



Функции активации



Важно, что функции активации — это нелинейные функции.

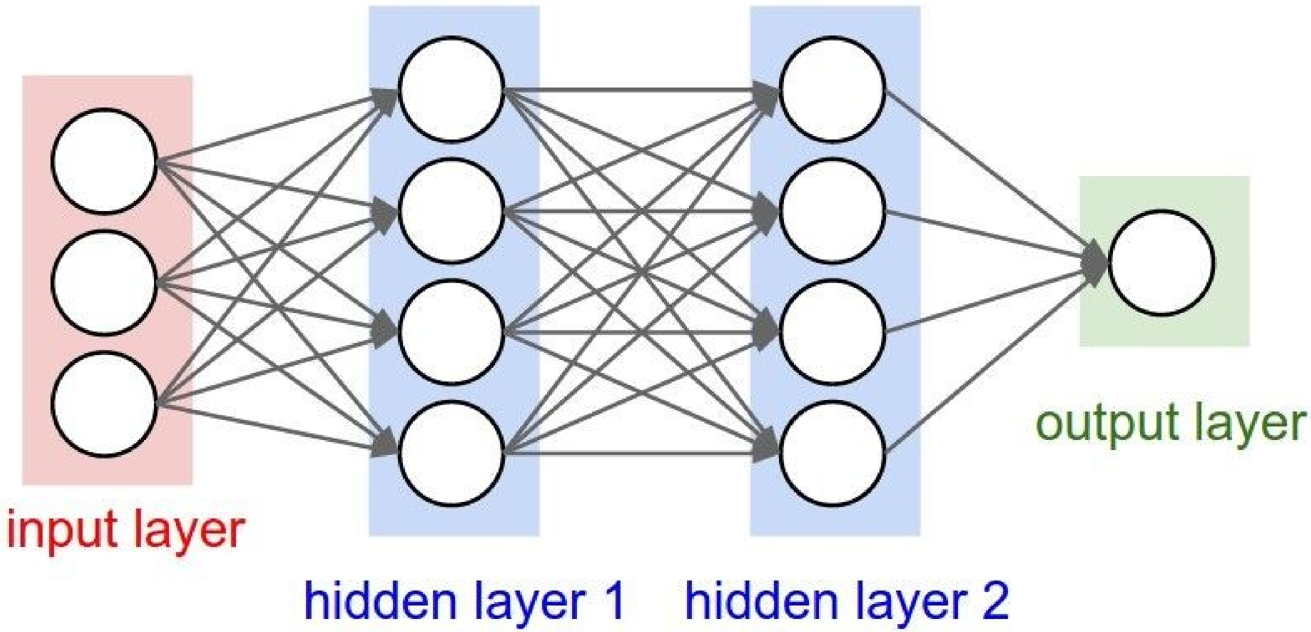
Функция ошибок

- Функция ошибок (часто называют функцией потерь, loss function) — это функция, которая позволяет оценить насколько полученный с помощью НС результат отличается от ожидаемого.
- Фактически это метрика, заданная в пространстве результатов.
- Таким образом, это отображение результатов M на действительные числа R:

  r: M → R

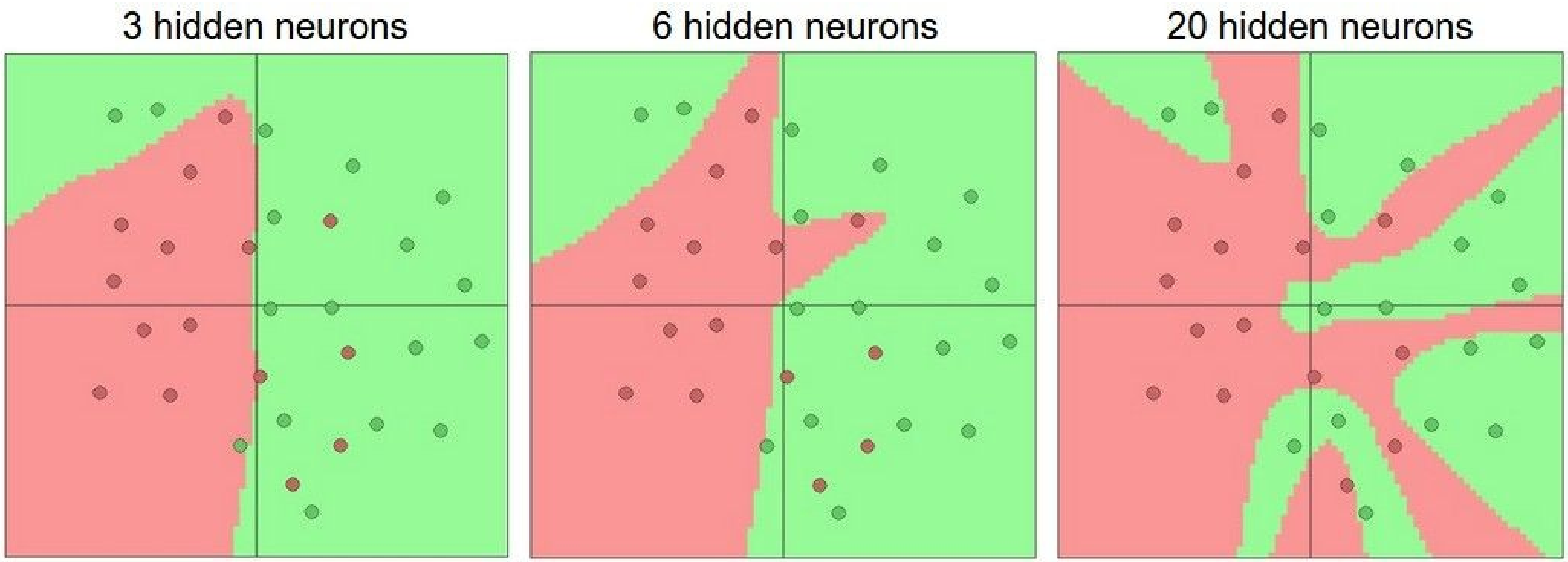
  - Как правило, в качестве пространства M у нас будет использоваться R^D, где D — размерность пространства.
- Наиболее употребительные меры — это
  - $L_1 = |x-y|$, где $x,y \in R^D$
  - $L_2 = (x-y)^2$

Архитектура НС



- Гиперпараметры НС — это параметры, изменение которых требует переобучения сети. Например
    - число слоев, число нейронов в слое;
    - функции активации.
- Параметры НС — это параметры, которые изменяются в процессе обучения НС. Например, w_{ij}, b_i.

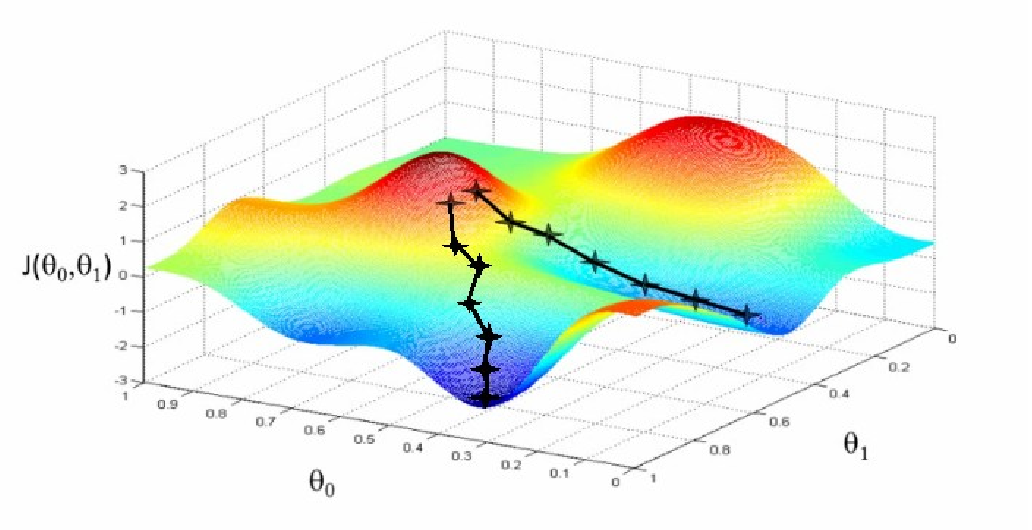
Выбор числа слоев и нейронов в слое



- Чем больше нейронов, тем более тонкие различия мы можем описать.

Обратное распространение ошибок

Метод градиентного спуска

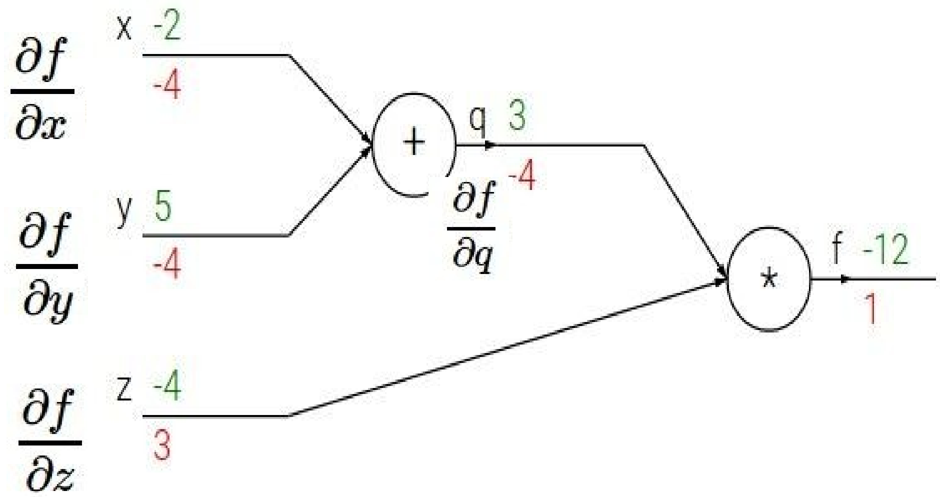


- Метод градиентного спуска основан на движении в сторону уменьшения значения функции в направлении обратному градиенту (df/dx)
- Фактически НС — это функция, которая представляет суперпозицию большого количества достаточно простых функций.
- В нашем случае НС представляет из себя класс функций, зависящий от параметров сети, среди которых надо выбрать оптимальный по некоторой метрике.

Градиентный спуск. Простой пример

- Функции f можно сопоставить следующий вычислительный граф:
- Зададим значения
  x = -2, y = 5, z = -4

$$f(x, y, z) = (x + y)z$$



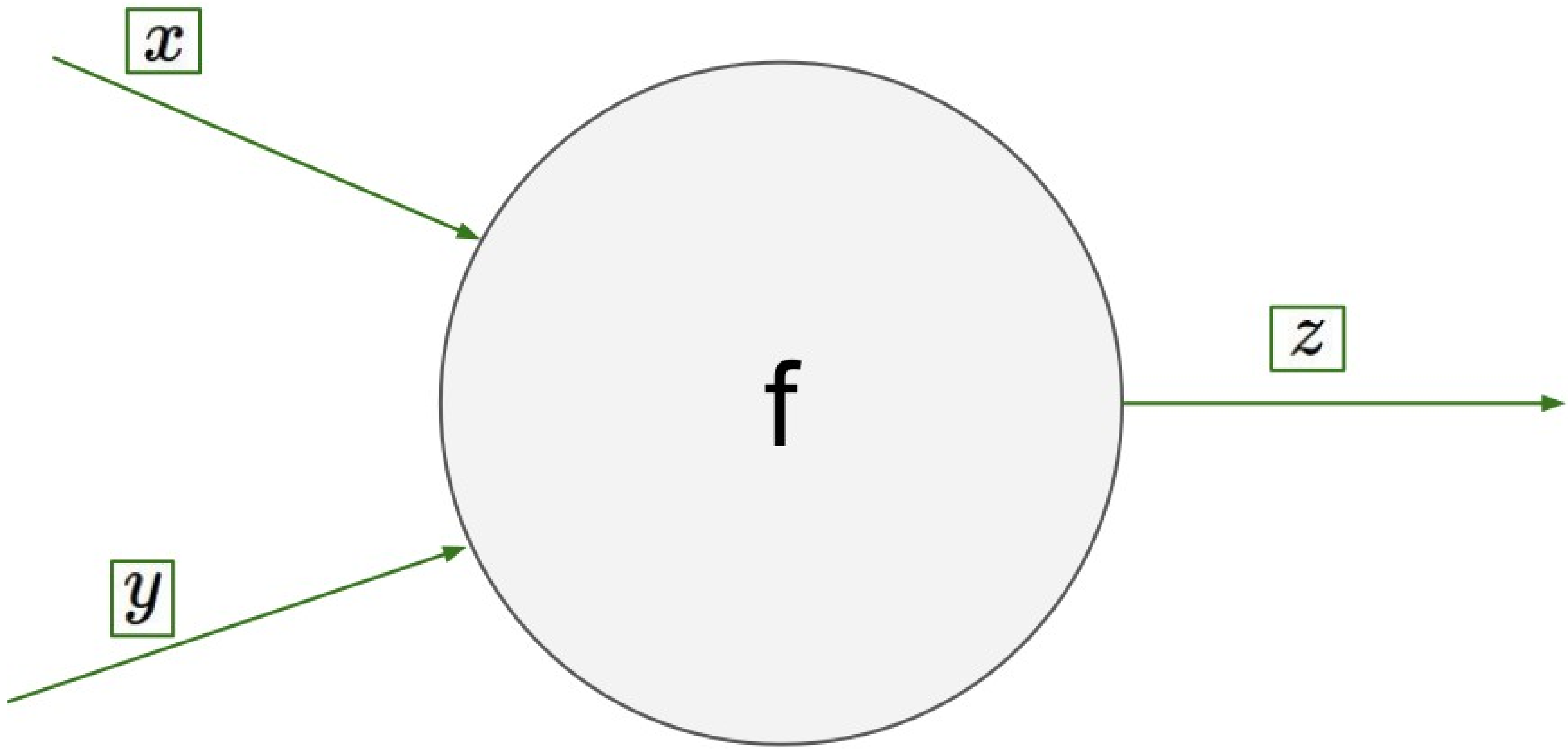
$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

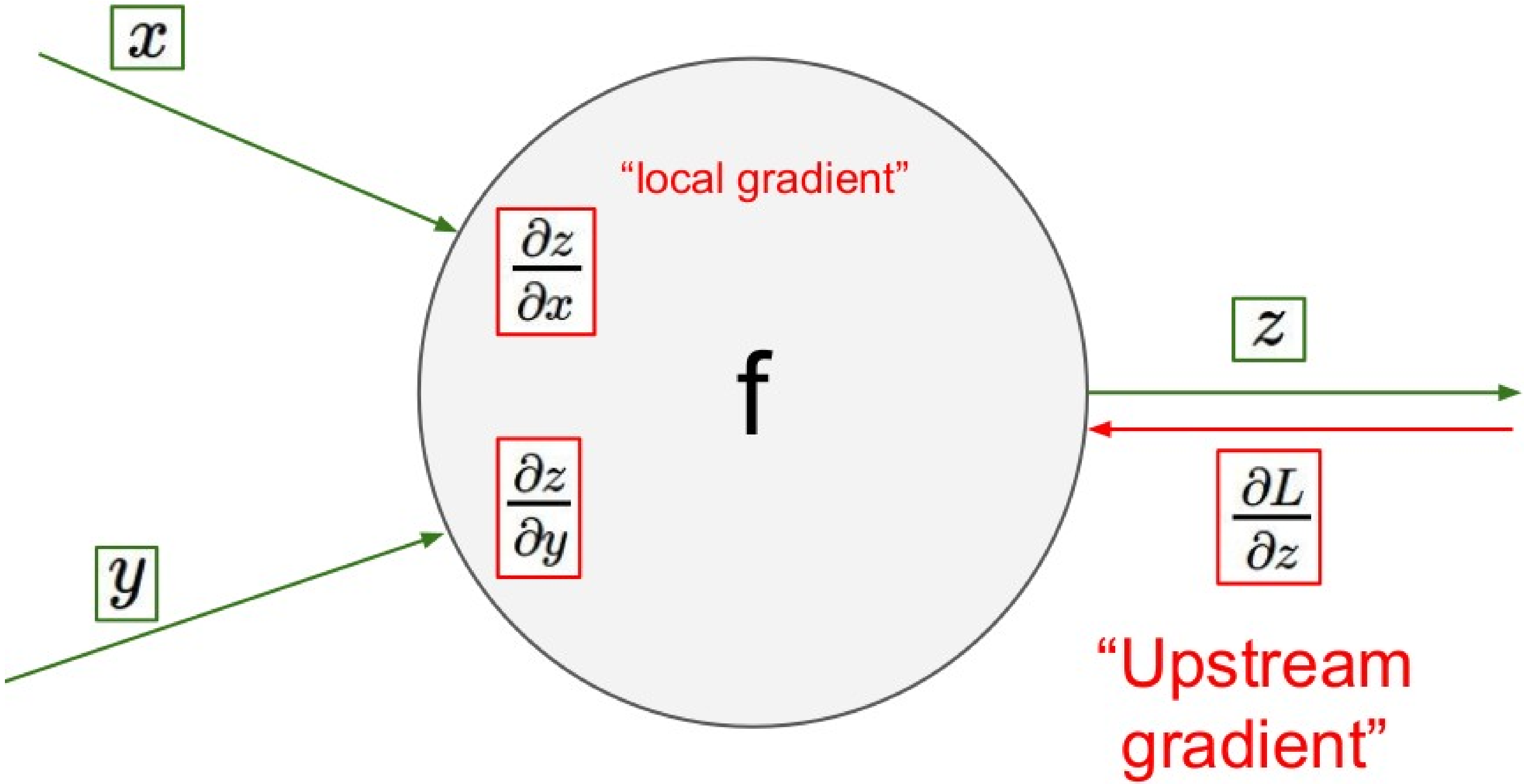
- Правило дифференцирования сложной функции

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial y}$$

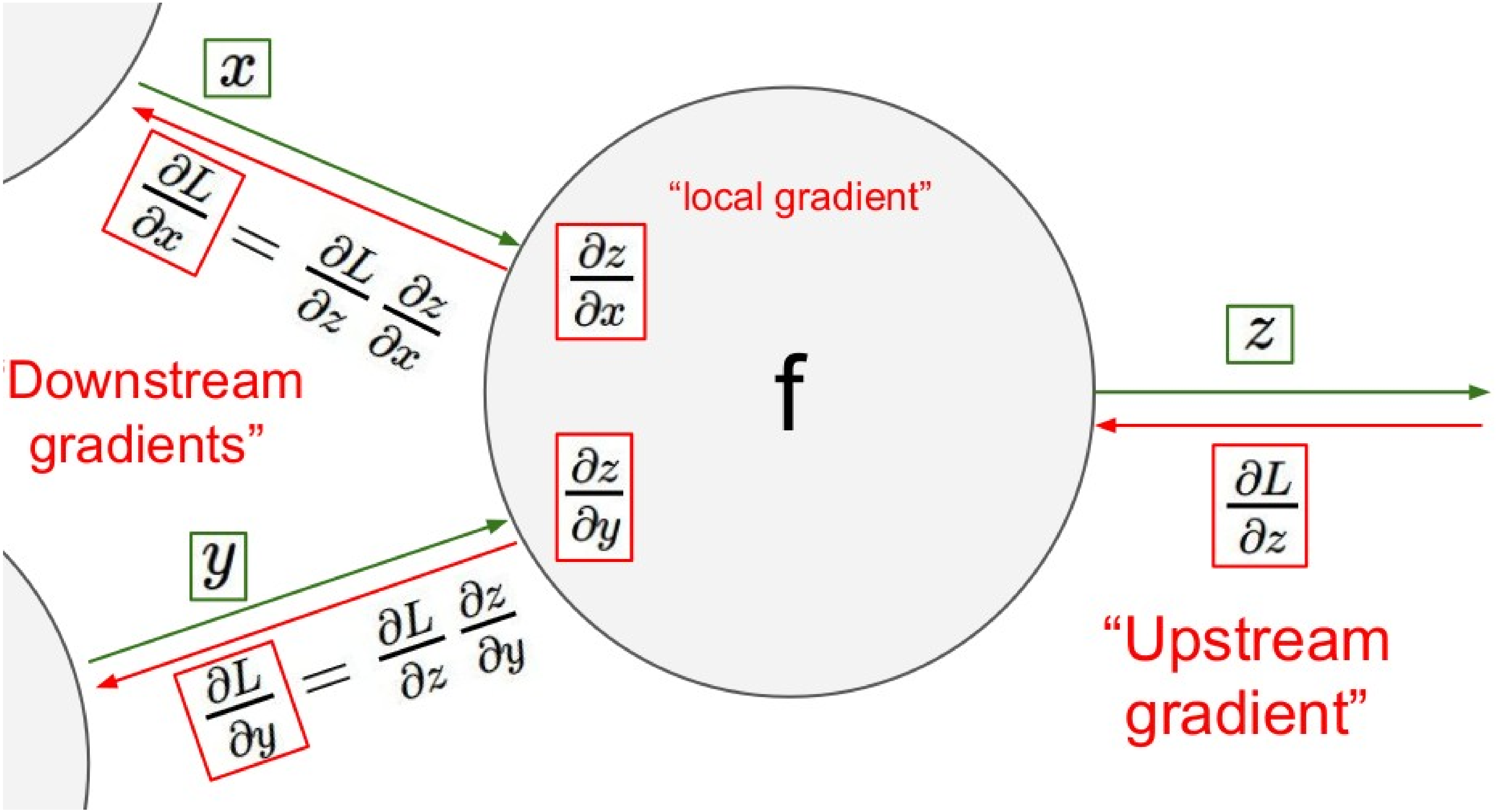
Градиентный спуск



Градиентный спуск



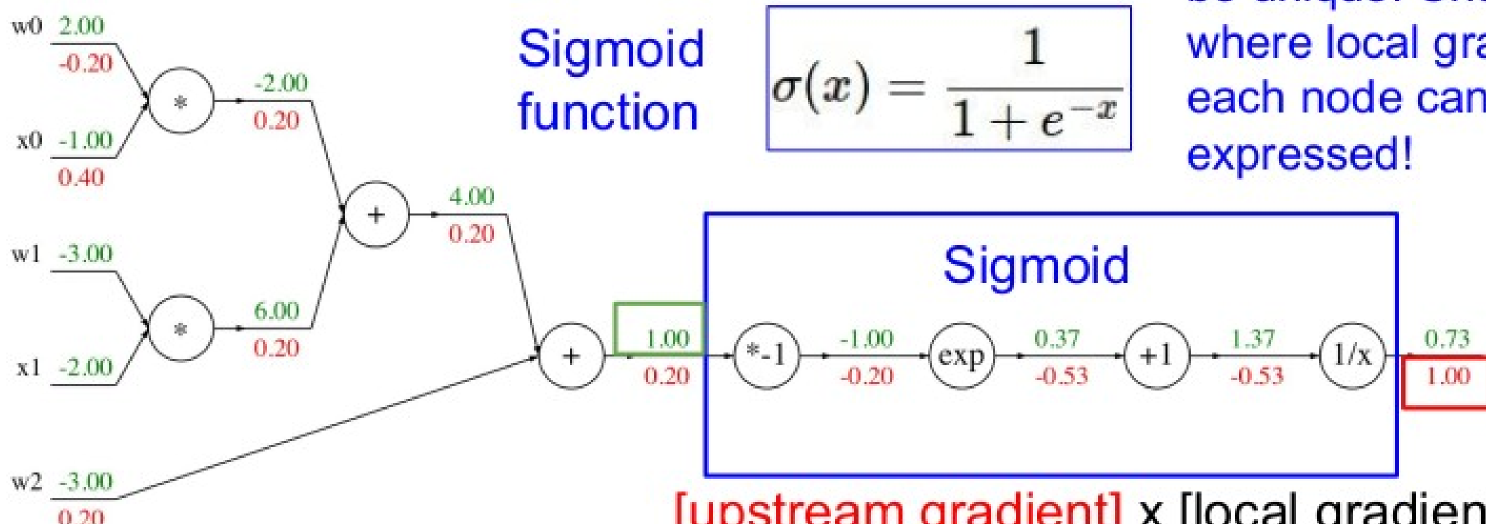
Градиентный спуск



Another example: $f(w,x) = \dfrac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$

Computational graph representation may not be unique. Choose one where local gradients at each node can be easily expressed!

Sigmoid function $\boxed{\sigma(x) = \dfrac{1}{1+e^{-x}}}$



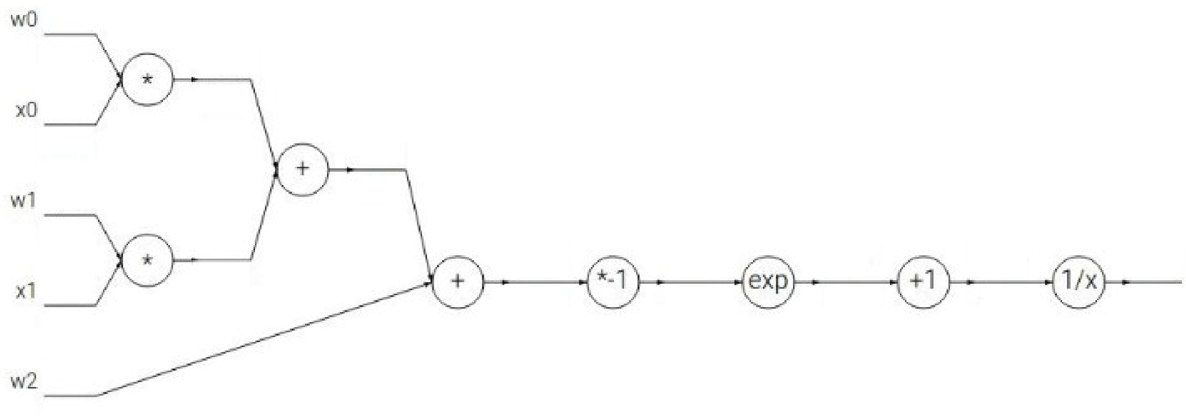
[upstream gradient] x [local gradient]

[1.00] x [(1 - 1/(1+e^{1})) (1/(1+e^{1}))] = 0.2

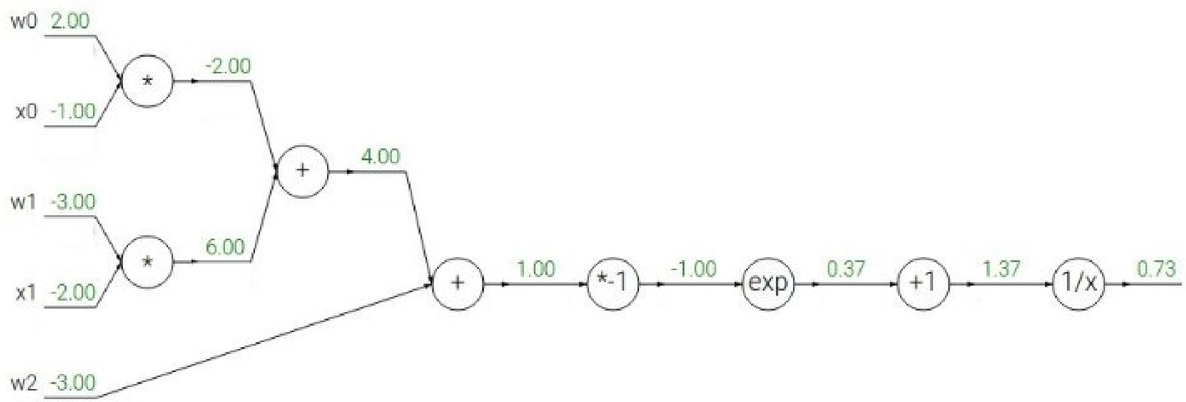
Sigmoid local gradient:

$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right) = (1-\sigma(x))\,\sigma(x)$$

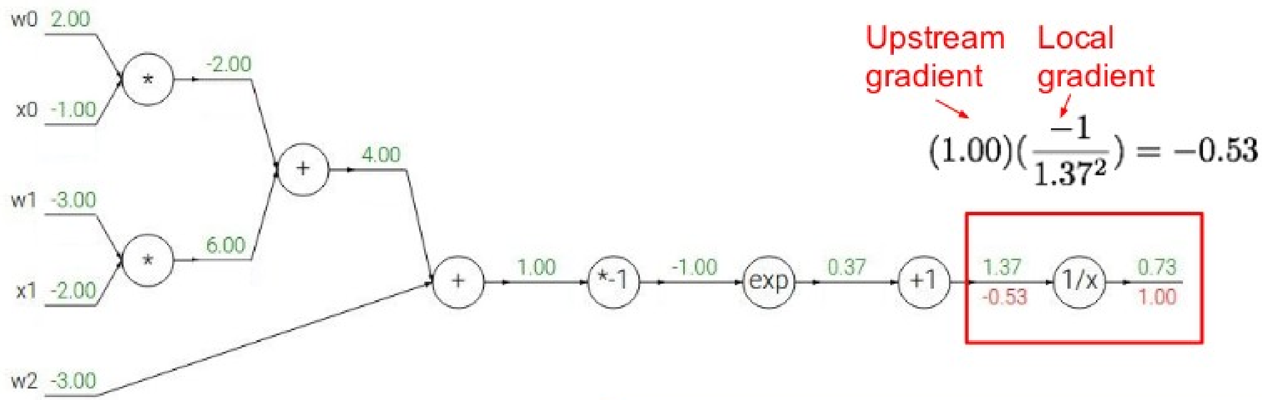
Another example: $f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$



Another example: $f(w,x) = \dfrac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$

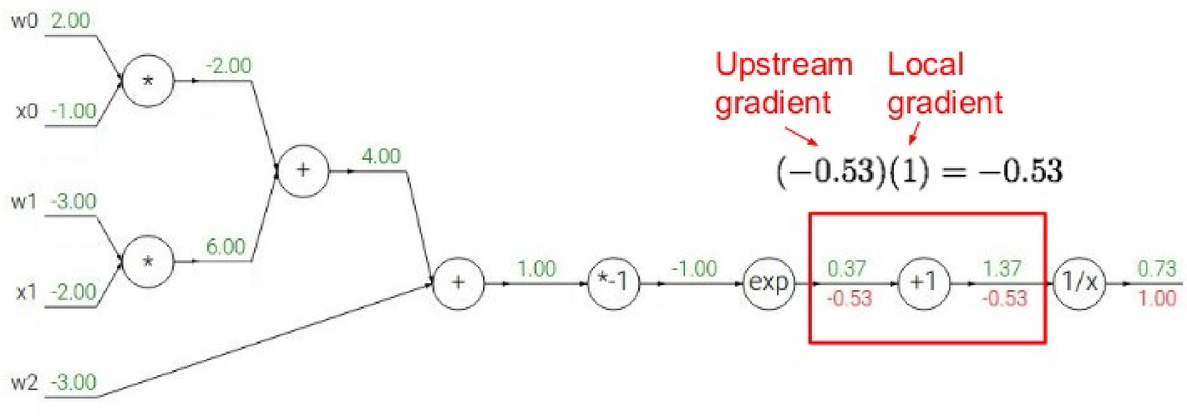


Another example: $f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$



$f(x) = e^x$	$\rightarrow$	$\frac{df}{dx} = e^x$	$\boxed{f(x) = \frac{1}{x}}$	$\rightarrow$	$\boxed{\frac{df}{dx} = -1/x^2}$
$f_a(x) = ax$	$\rightarrow$	$\frac{df}{dx} = a$	$f_c(x) = c + x$	$\rightarrow$	$\frac{df}{dx} = 1$

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



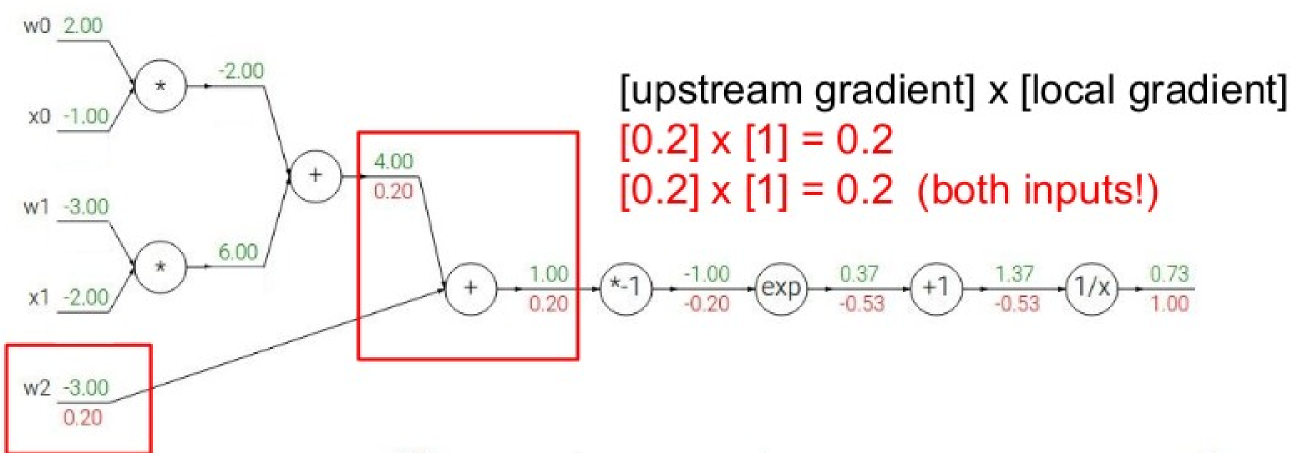
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$\boxed{f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1}$$

Another example: $f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$

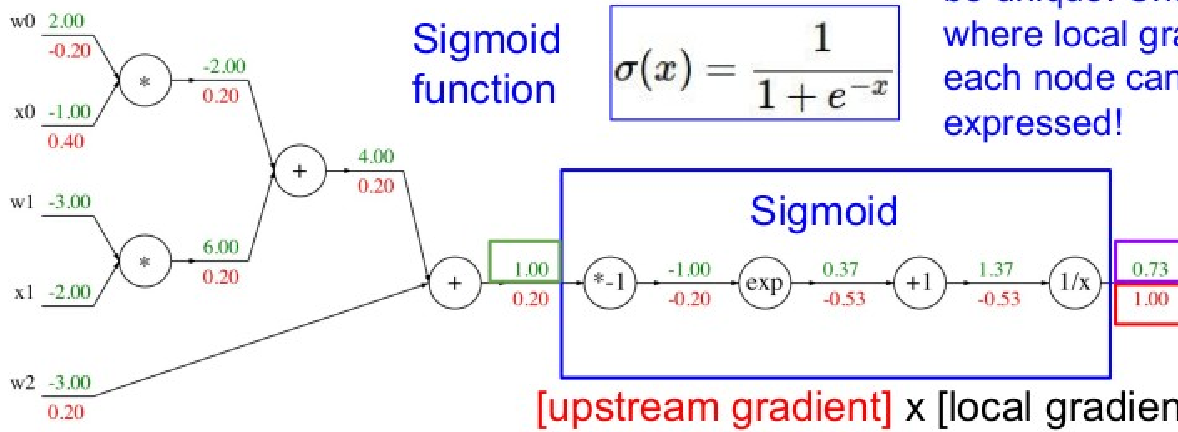


$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$	$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$
$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$	$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$

Another example: $f(w,x) = \dfrac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$

Computational graph representation may not be unique. Choose one where local gradients at each node can be easily expressed!

Sigmoid function $\boxed{\sigma(x) = \dfrac{1}{1+e^{-x}}}$



[upstream gradient] x [local gradient]

[1.00] x [(1 - 0.73) (0.73)] = 0.2

Sigmoid local gradient: $$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right) = (1-\sigma(x))\,\sigma(x)$$

Градиентный спуск

- Мы научились распространять ошибку. Таким образом новые значения весов w_i будут следующими:

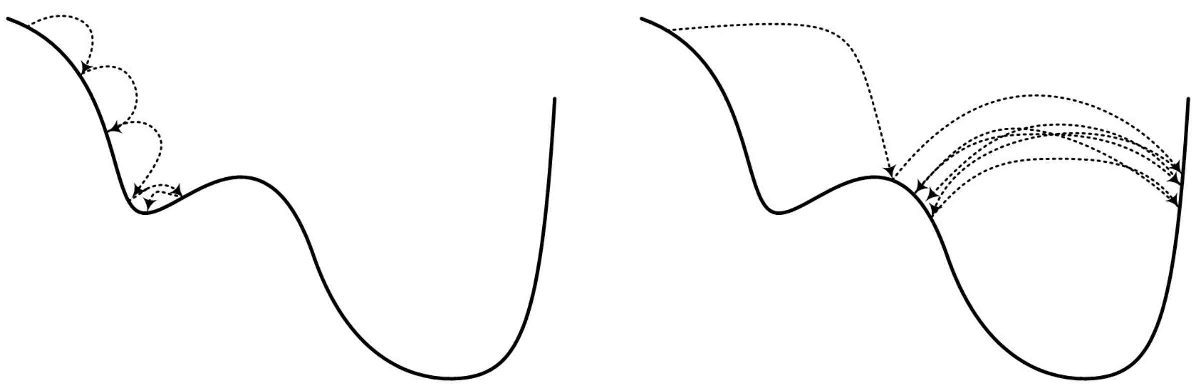
  $$w^{(k+1)} = w^{(k)} - \lambda \nabla L(w^{(k)}) ,$$

  где λ – скорость обучения

- Проблем со скорость обучения

  a) маленькая — долго и застревает в небольших локальных минимумах;

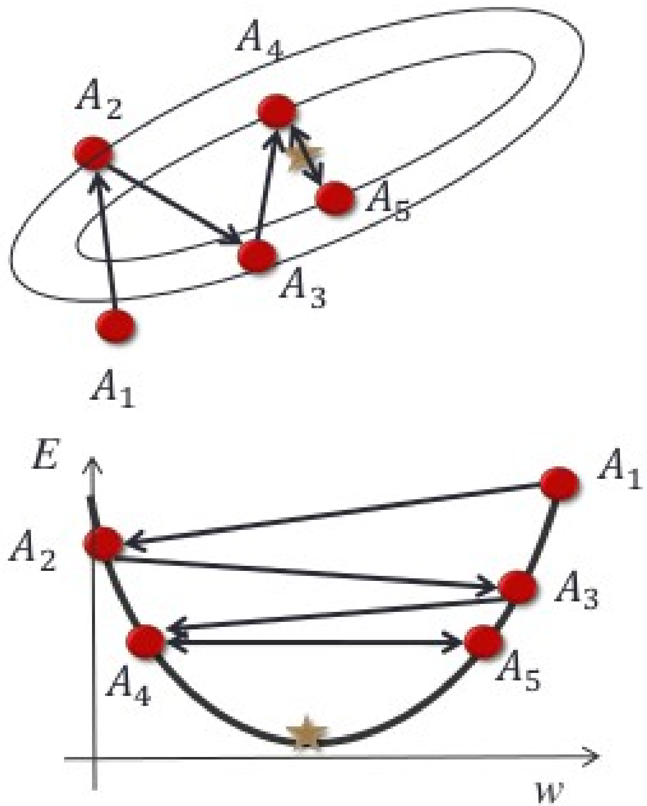
  b) большая — может проскочить минимум или вовсе застрять около.



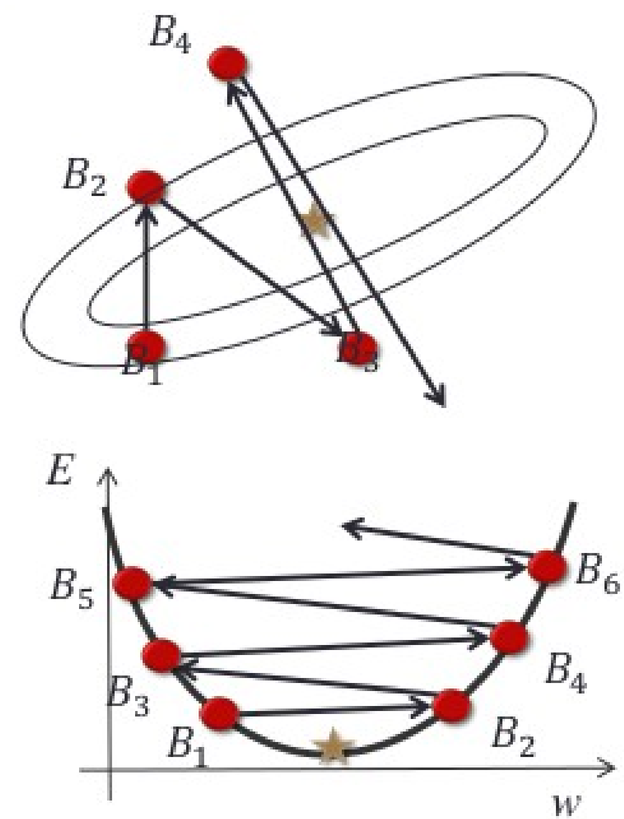
(а) (б)

Влияние скорости обучения

При большой скорости обучения алгоритм не может спуститься

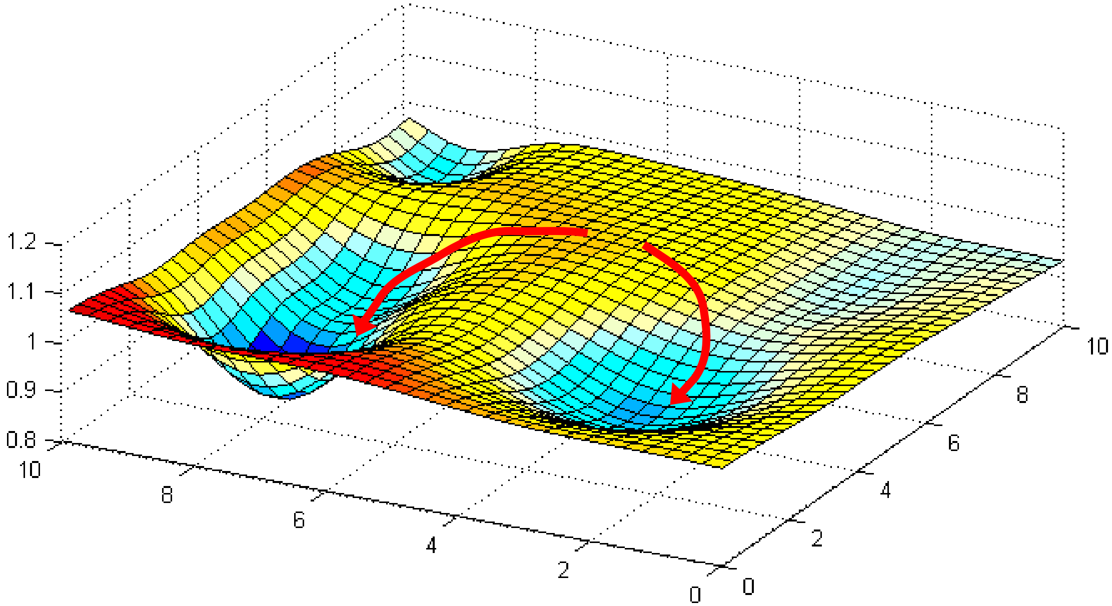


При большой скорости обучения алгоритм расходится



Проблема локальных минимумов

- Можно попробовать стартовать из разных точек и потом выбрать наилучший минимум из полученных.
  - Гарантии, что полученный минимум — это глобальный нет, но это лучше чем ничего :-)



Материалы лекций:

(https://theory.sinp.msu.ru/doku.php/ml_lectures)