Математические основы информационной безопасности

Груздев Дмитрий Николаевич

Нейронные сети

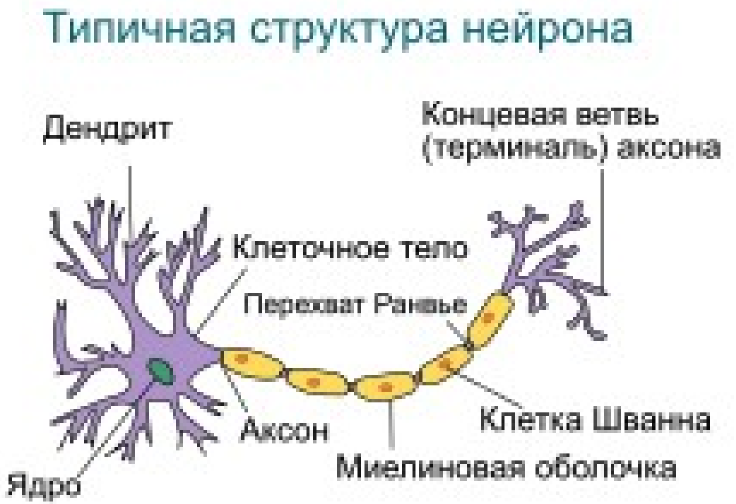
Модель нейрона

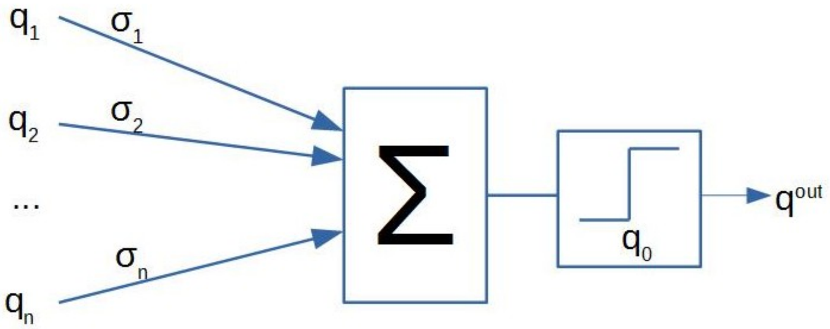
$q_i = \sigma_i * q_i^{in}$ – заряд с i-го дендрита

$Q = \Sigma q_i$ – образовавшийся заряд в нейроне

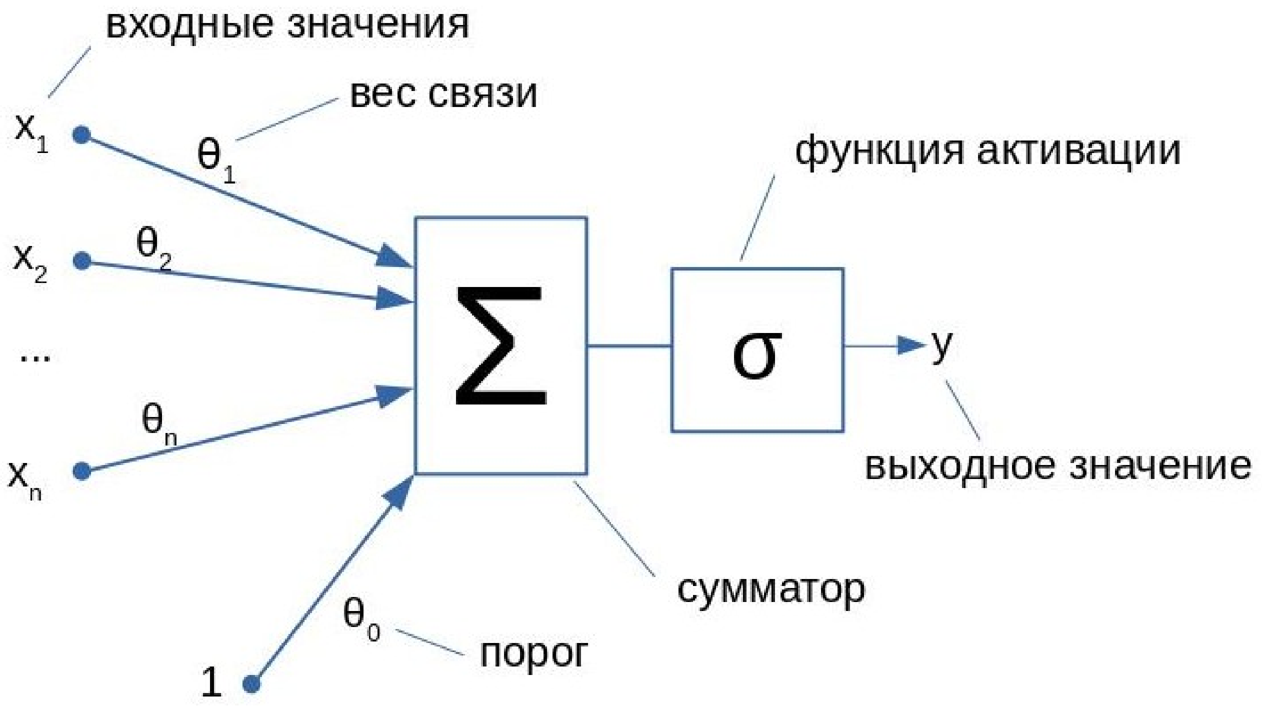
Если $Q \geq q_0$, то q^{out} – сигнал следующему нейрону

Нейрон проверяет $\Sigma \sigma_i * q_i^{in} \geq q^0 \Leftrightarrow \langle \sigma * q^{in} \rangle - q^0 \geq 0$. Т.е. нейрон – простой линейный классификатор.



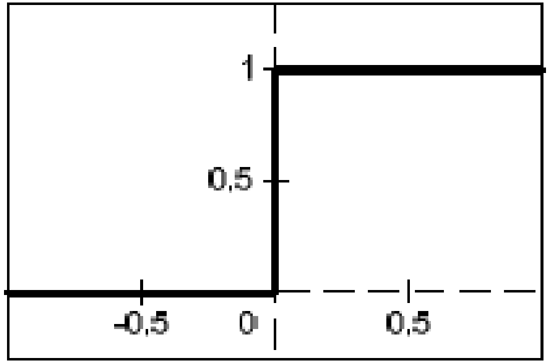


Математическая модель

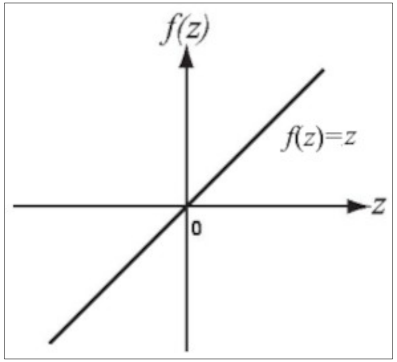


$$y = \sigma(<\theta, x>)$$

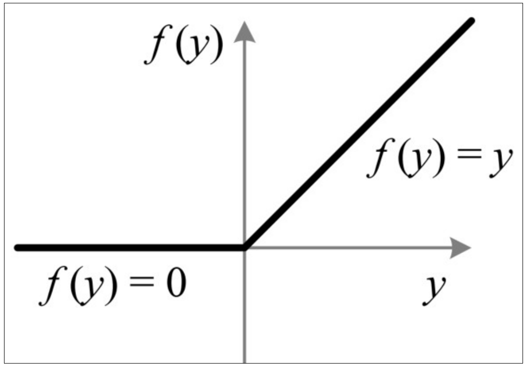
Функции активации



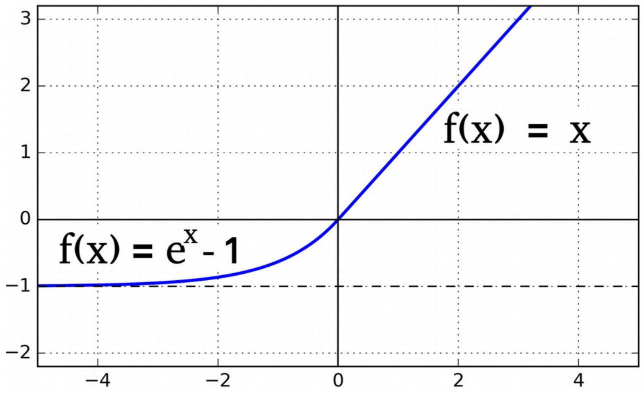
пороговая



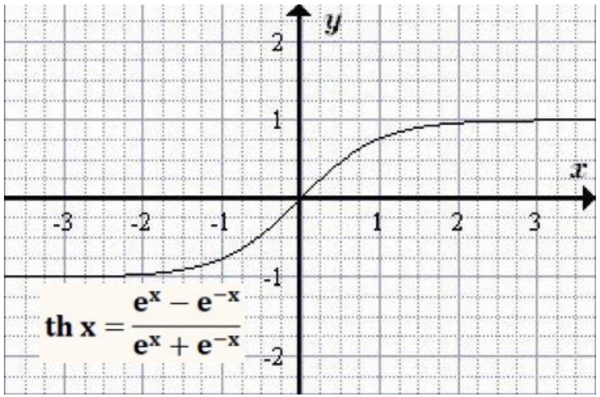
линейная



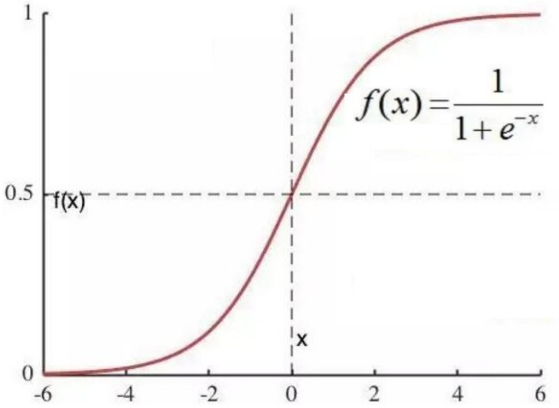
ReLU



ELU

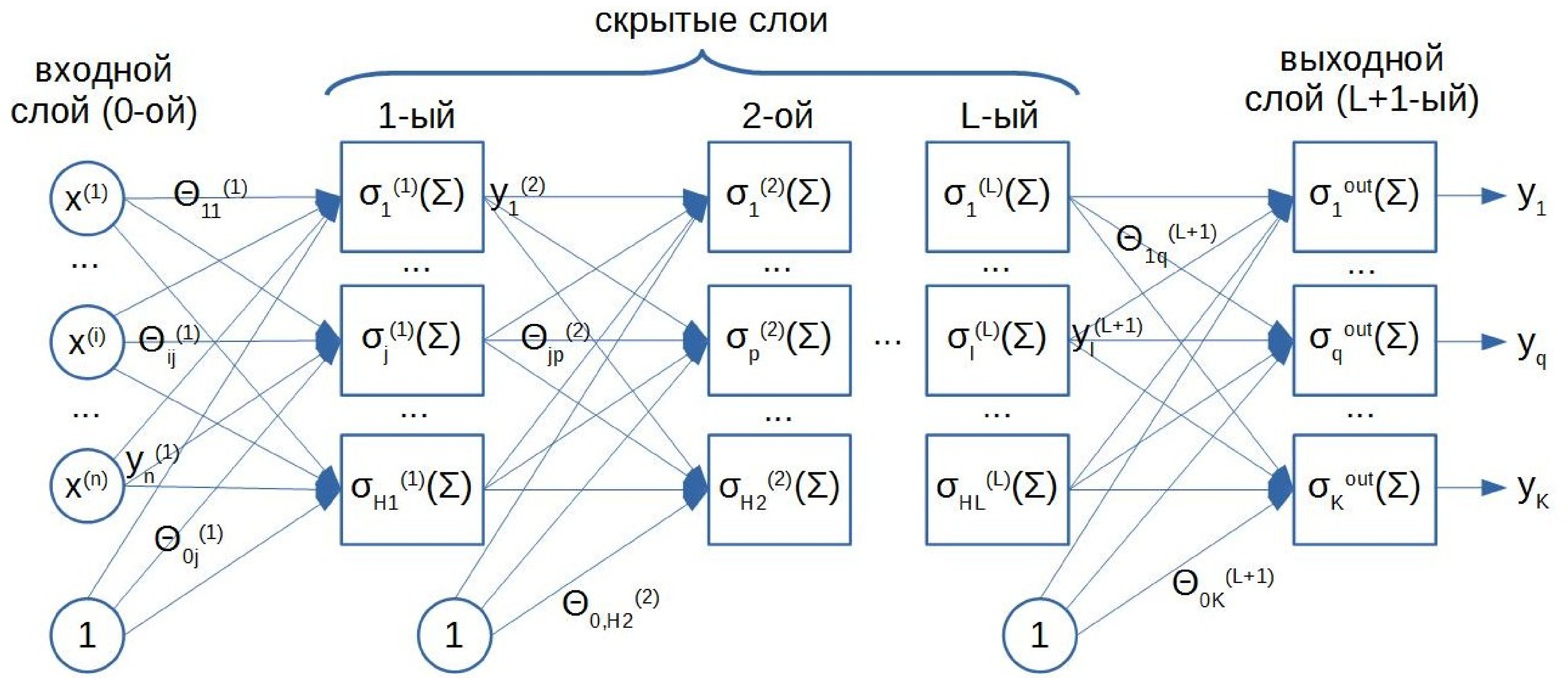


гиперболический тангенс



сигмойда

Нейронная сеть



$\Theta_{ij}^{(k)}$ – вес связи между i-м нейроном (k-1)-го слоя и j-м нейроном k-го слоя,
$y_i^{(k)}$ – значение выхода из i-го нейрона (k-1)-го слоя (причем $y_i^{(1)} = x^{(i)}$ и $y_0^{(j)} = 1$).

Нейронная сеть

$y_i^{(k+1)} = \sigma_i^{(k+1)}(\Sigma_{0 \le j \le Hk} \Theta_{ji}^{(k)} * y_j^{(k)})$

Количество обучаемых параметров = количество связей:

$$N = (n + 1) * H_1 + (H_1 + 1) * H_2 + \ldots + (H_L + 1) * K$$

У человека:

~ $1.6 * 10^{10}$ нейронов

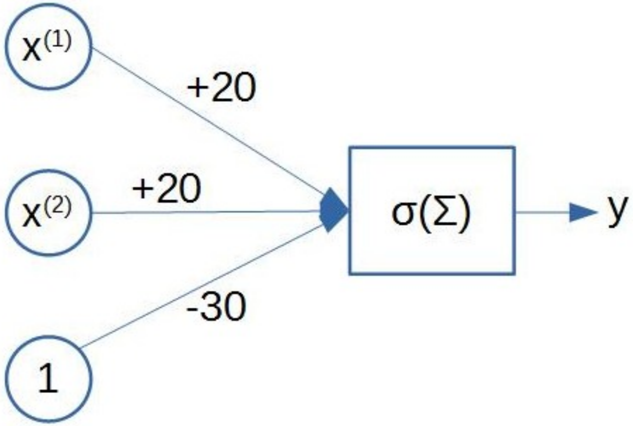
~ 10^4 связей у нейрона

У шароголового дельфина: $3.7*10^{10}$ нейронов

Google Translation Machine: $9*10^9$ связей

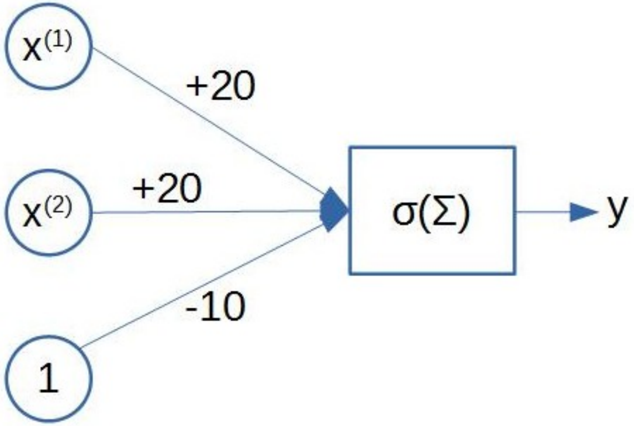
Бинарные функции

AND($x^{(1)}$, $x^{(2)}$)



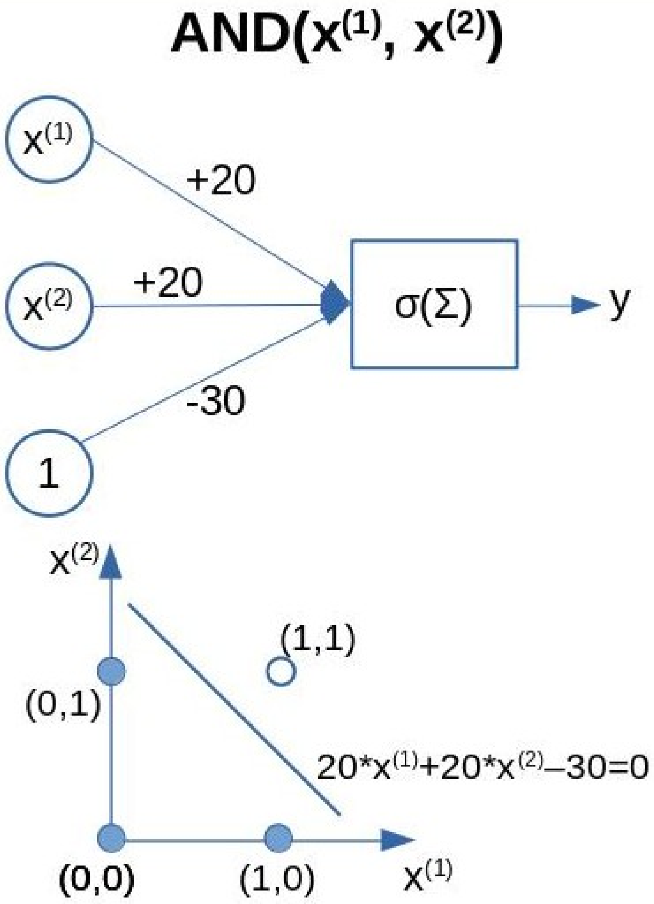
$x^{(1)}$	$x^{(2)}$	$\sigma(\Sigma)$
0	0	$\sigma(-30) \approx 0$
0	1	$\sigma(-10) \approx 0$
1	0	$\sigma(-10) \approx 0$
1	1	$\sigma(10) \approx 1$

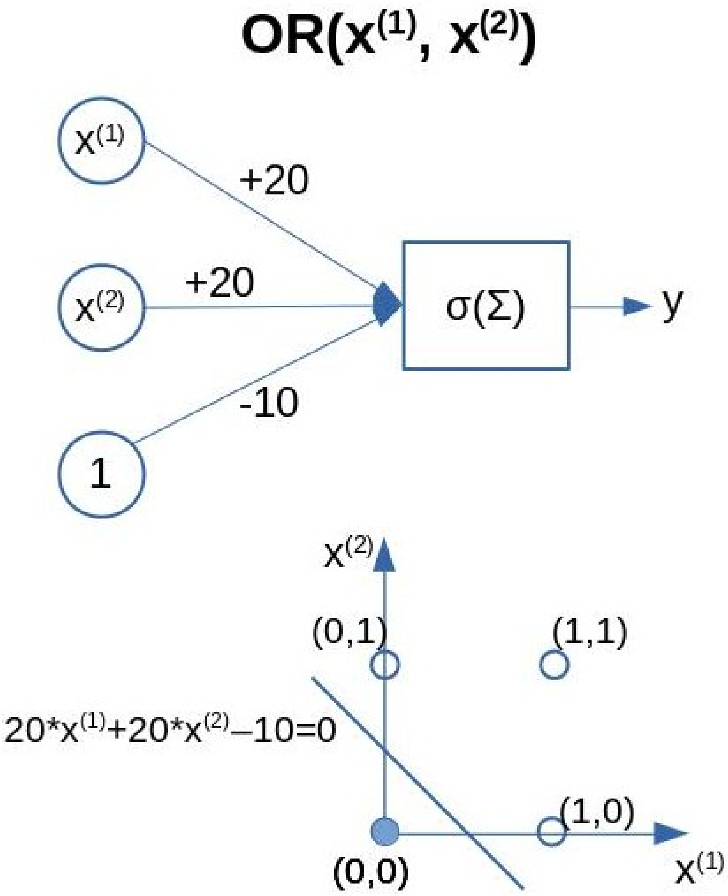
OR($x^{(1)}$, $x^{(2)}$)



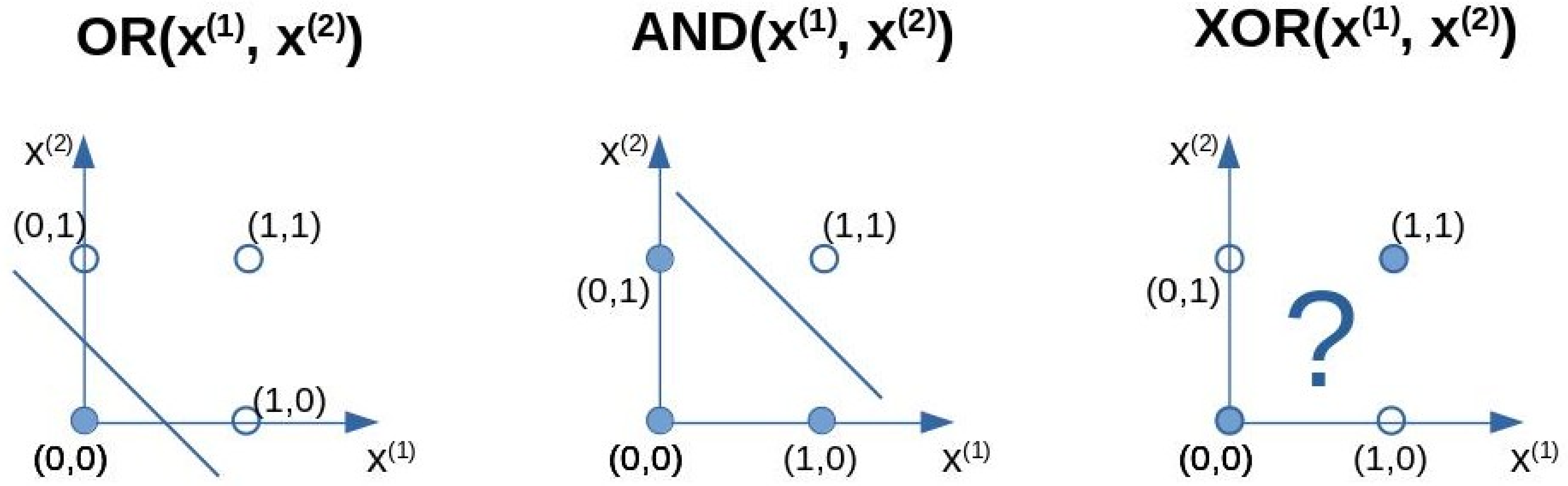
$x^{(1)}$	$x^{(2)}$	$\sigma(\Sigma)$
0	0	$\sigma(-10) \approx 0$
0	1	$\sigma(10) \approx 1$
1	0	$\sigma(10) \approx 1$
1	1	$\sigma(30) \approx 1$

Разделяющие прямые

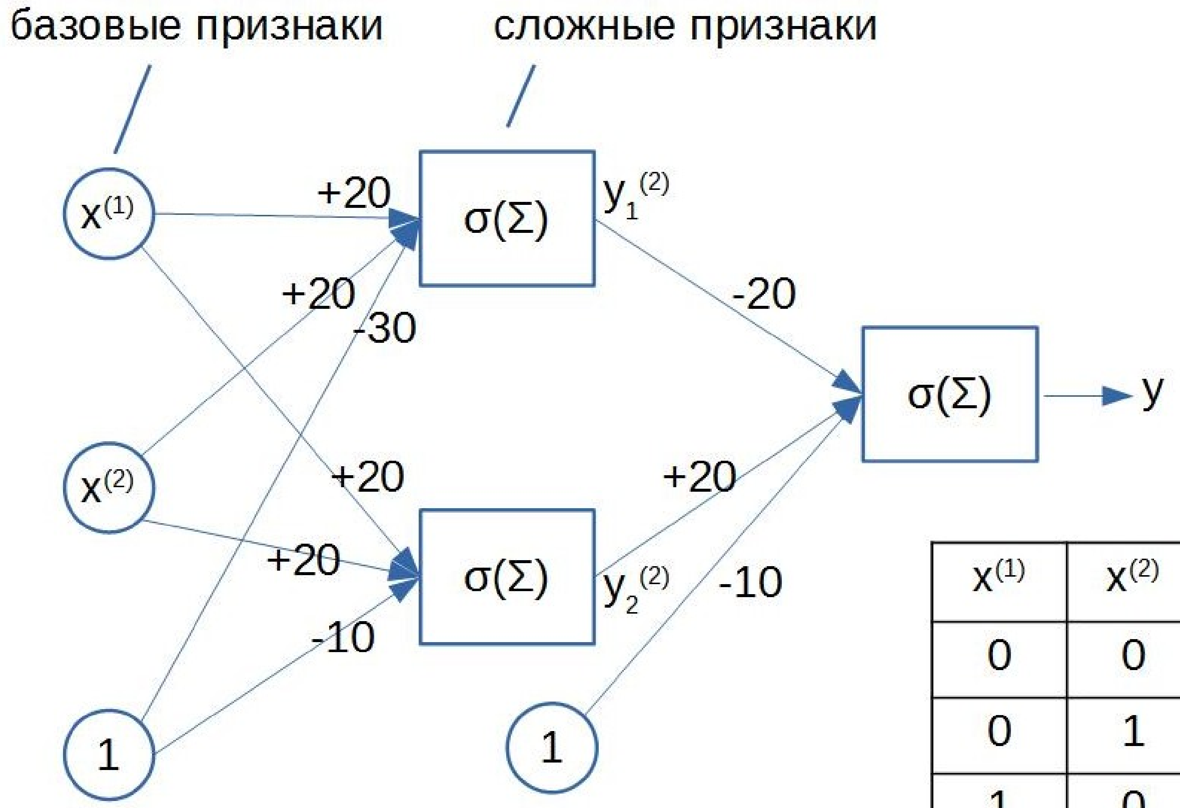




Неразделяемый случай



XOR



$x^{(1)}$	$x^{(2)}$	$y_1^{(2)}$	$y_2^{(2)}$	y
0	0	≈ 0	≈ 0	≈ 0
0	1	≈ 0	≈ 1	≈ 1
1	0	≈ 0	≈ 1	≈ 1
1	1	≈ 1	≈ 1	≈ 0

Обобщающая способность

С помощью линейных операций и одной нелинейной функции активации можно приблизить любую непрерывную функцию с любой желаемой точностью (Цыбенко 1989г).

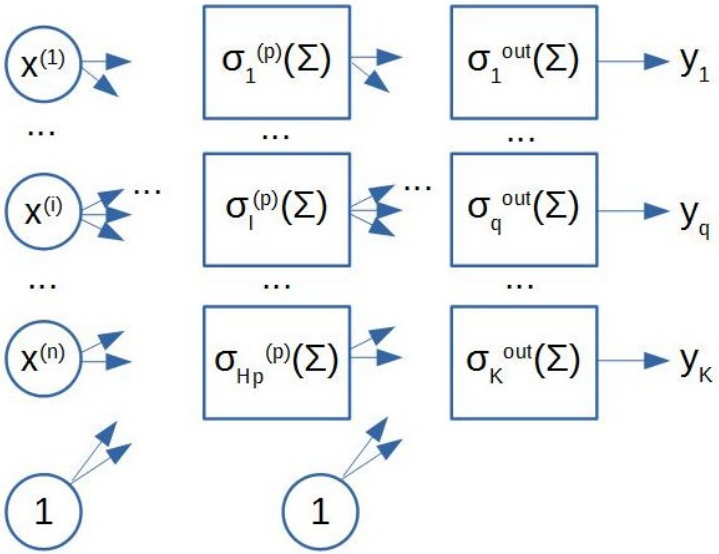
Если все функции активации линейны, то выходные значения представляются линейными комбинациями от входных параметров. Использование скрытых слоев не имеет смысла.

ILSVRC

ImageNet Large Scale Visual Recognition Challenge – соревнования по классификации объектов с 2010 года.

	Ошибки	Кол-во связей (млн)
AlexNet, 2012	18.2	60
ZFNet, 2013	16.0	62
VGG-19, 2014	8.4	144
GoogleNet, 2014	7.9	4
ResNet-152, 2015	4.5	2

Задачи регрессии



$(x_1, t_1), \dots, (x_m, t_m)$ – обучающая выборка

$x_i = (x_1^{(i)}, \dots, x_n^{(i)}),\ t_i = (t_1^{(i)}, \dots, t_k^{(i)})$

$y_i = A(x_i)$

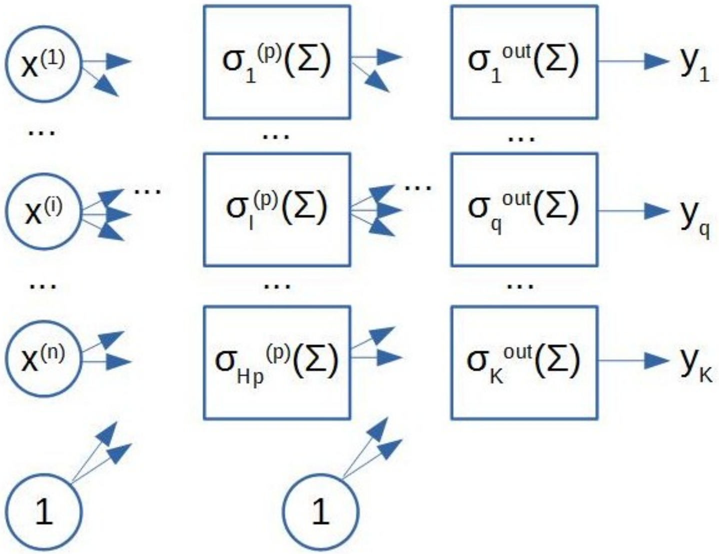
$\sigma_i^{out}(\Sigma)$ – линейные:

$y_i = \Sigma$

Функция ошибки – квадратичная:

$E = 0.5 * \Sigma(y_i - t_i)^2$

Задачи классификации



$(x_1, t_1), \dots, (x_m, t_m)$ – обучающая выборка

$x_i = (x_i^{(1)}, \dots, x_i^{(n)}),\ t_i \in K$

$t_i = (0_1, \dots, 0_{p-1}, 1_p, 0_{p+1}, \dots, 0_K)$, если $t_i = p$

$y_i = A(x_i)$

$\sigma_i^{out}(\Sigma)$ – softmax:

$y_i = \sigma_i^{out}(\Sigma) = e^{\Sigma i} / \Sigma e^{\Sigma j};\ \ \sigma_i^{out}(\Sigma)' = y_i * (1-y_i)$

Функция ошибки:

$E = -\Sigma t_i * \log(y_i);\ \ \partial E/\partial \Sigma_i = y_i - t_i$

Обучение нейросети

(x_1, t_1), (x_2, t_2), … , (x_m, t_m) – обучающая выборка

$x_i = (x_1^{(i)}, \ldots, x_n^{(i)})$, $t_i = (t_1^{(i)}, \ldots, t_k^{(i)})$

$A : \mathbb{R}^n \rightarrow \mathbb{R}^k$ – алгоритм, реализуемый нейросетью

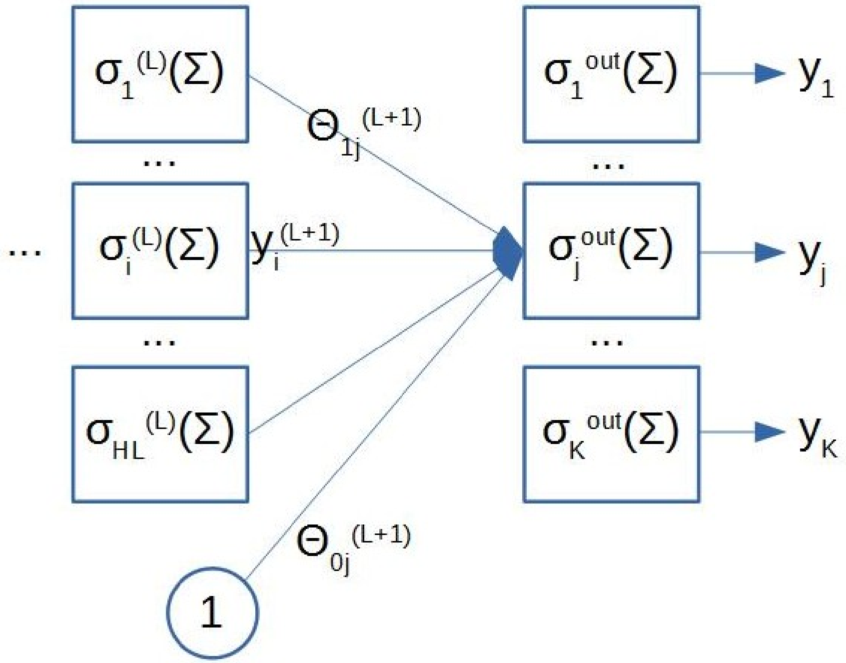
После вычисления $A(x_i) = y_i$ известны:

x_i – входные значения для нейросети;
y_i – выходные значения из нейросети;
t_i – правильные выходные значения из нейросети;
Σ – входные значения для нейронов каждого слоя;
$y_i^{(p)}$ – выходные значения из нейронов каждого слоя;
$\Theta_{xy}^{(z)}$-текущие значение весов связей.

Необходимо:

вычислить $\Delta\Theta_{xy}^{(z)} = -\alpha * \partial E / \partial \Theta_{xy}^{(z)}$ для каждой связи

Обратное распространение ошибки



$$E = E_{y1} + E_{y2} + \ldots + E_{yk}$$

$$\frac{\partial E}{\partial y_i^{(L+1)}} = \frac{\partial E_{y1}}{\partial y_i^{(L+1)}} + \ldots + \frac{\partial E_{yk}}{\partial y_i^{(L+1)}}$$

$$\frac{\partial E_{yj}}{\partial y_i^{(L+1)}} = \frac{\partial \sigma_j^{out}(\Sigma)}{\partial y_i^{(L+1)}} * E'_{yj}$$

$$\Sigma = \Sigma_{0 \le p \le HL} \Theta_{pj}^{(L+1)} y_p^{(L+1)}$$

$$\frac{\partial E_{yj}}{\partial y_i^{(L+1)}} = \frac{\partial \Sigma}{\partial y_i^{(L+1)}} * (\sigma_j^{out}(\Sigma))' * E'_{yj}$$

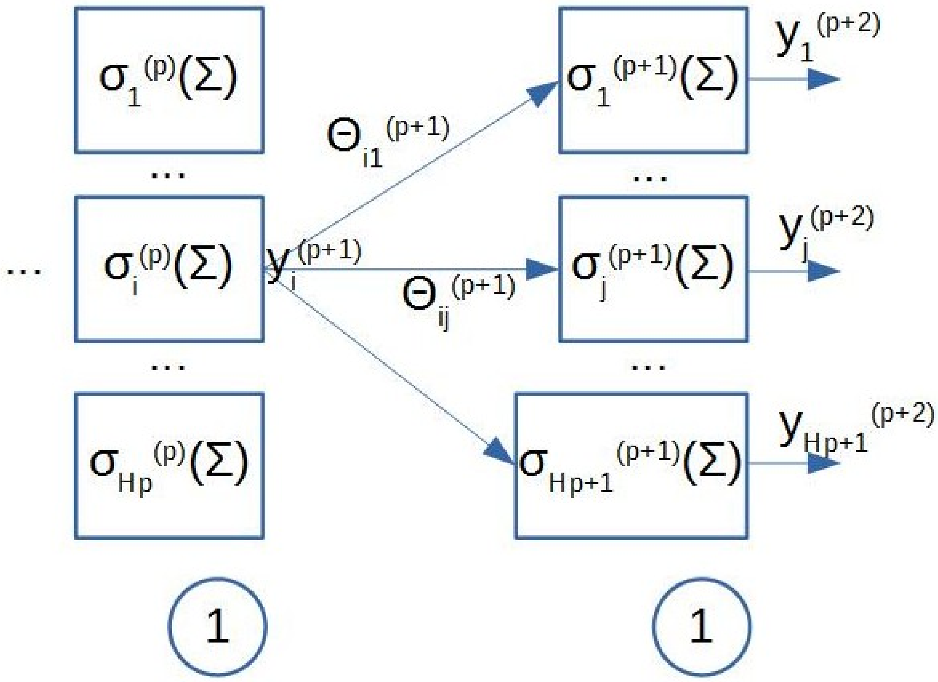
$$\frac{\partial E_{yj}}{\partial y_i^{(L+1)}} = \Theta_{ij}^{(L+1)} * (\sigma_j^{out}(\Sigma))' * E'_{yj}$$

Если $E = 0.5 *(t_j - y_j)^2$ и $\sigma_j^{out}(\Sigma) = 1/(1+e^{-\Sigma})$, то

$dE/dyj = y_j - t_j$ и $d\sigma_j^{out}(\Sigma)/d\Sigma = \sigma_j^{out}(\Sigma)(1 - \sigma_j^{out}(\Sigma)) = y_j*(1-y_j)$, поэтому

$\partial E_y/\partial y_i^{(L+1)} = \Theta_{ij}^{(L+1)} * y_j*(1-y_j) * (y_j - t_j)$

Обратное распространение ошибки



$$\frac{\partial E}{\partial y_1^{(p+2)}}, \ldots, \frac{\partial E}{\partial y_{Hp+1}^{(p+2)}}$$ - известны

$$\frac{\partial E}{\partial y_i^{(p+1)}}$$ - найти

$$\frac{\partial E}{\partial y_i^{(p+1)}} = \sum \frac{\partial E}{\partial y_j^{(p+2)}} * \frac{\partial y_j^{(p+2)}}{\partial y_i^{(p+1)}}$$

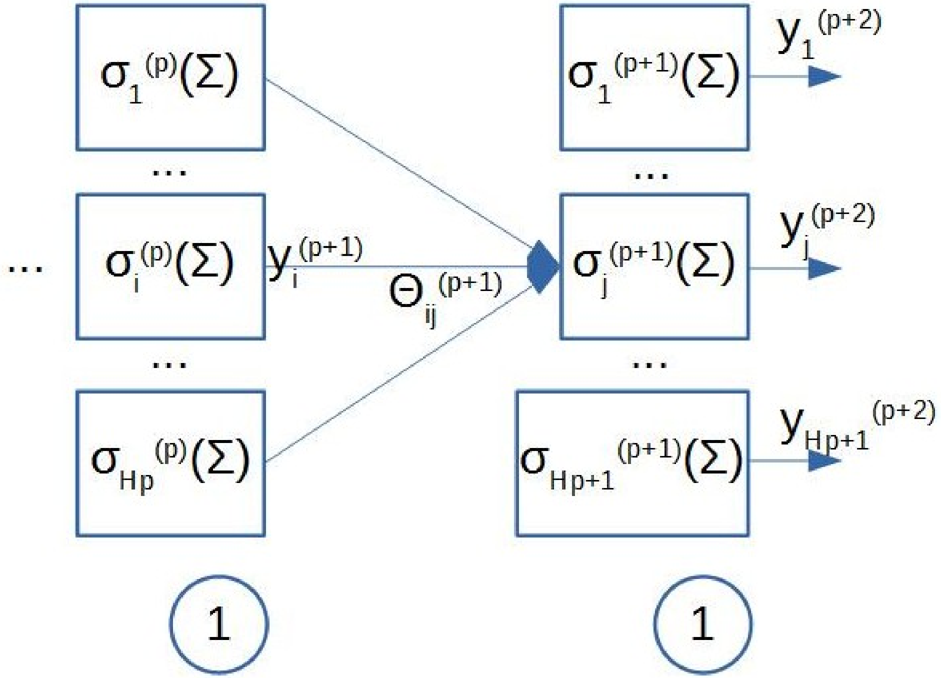
$$\frac{\partial y_j^{(p+2)}}{\partial y_i^{(p+1)}} = \frac{\partial \left(\sigma_j^{(p+1)}(\Sigma)\right)}{\partial y_i^{(p+1)}}$$

$$= \frac{\partial \Sigma}{\partial y_i^{(p+1)}} * \left(\sigma_j^{(p+1)}(\Sigma)\right)'$$

$$= \Theta_{ij}^{(p+1)} * \left(\sigma_j^{out}(\Sigma)\right)'$$

$$\frac{\partial E}{\partial y_i^{(p+1)}} = \sum \frac{\partial E}{\partial y_j^{(p+2)}} * \Theta_{ij}^{(p+1)} * \left(\sigma_j^{out}(\Sigma)\right)'$$

Изменение весов



$$\frac{\partial E}{\partial \Theta_{ij}^{(p+1)}} \text{ - найти}$$

$$\frac{\partial E}{\partial \Theta_{ij}^{(p+1)}} = \frac{\partial E}{\partial y_j^{(p+2)}} * \frac{\partial y_j^{(p+2)}}{\partial \Theta_{ij}^{(p+1)}}$$

$$\frac{\partial y_j^{(p+2)}}{\partial \Theta_{ij}^{(p+1)}} = \frac{\partial \left(\sigma_j^{(p+1)}(\Sigma)\right)}{\partial \Theta_{ij}^{(p+1)}}$$

$$= \frac{\partial \Sigma}{\partial \Theta_{ij}^{(p+1)}} * \left(\sigma_j^{(p+1)}(\Sigma)\right)'$$

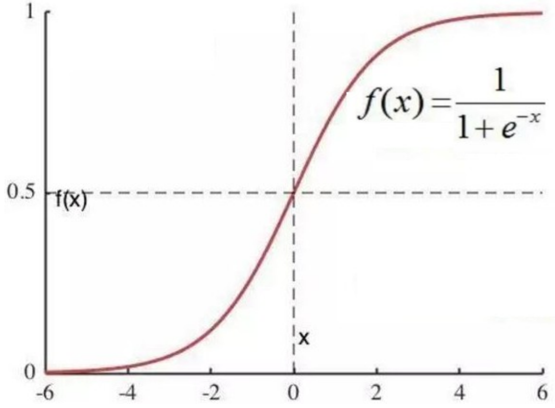
$$= y_i^{(p+1)} * \left(\sigma_j^{out}(\Sigma)\right)'$$

$$\frac{\partial E}{\partial \Theta_{ij}^{(p+1)}} = \frac{\partial E}{\partial y_j^{(p+2)}} * y_i^{(p+1)} * \left(\sigma_j^{out}(\Sigma)\right)'$$

Проблема затухания и взрыва градиента

$$\frac{\partial E}{\partial \Theta_{ij}^{(p+1)}} = \frac{\partial E}{\partial y_j^{(p+2)}} * y_i^{(p+1)} * (\sigma_j^{out}(\Sigma))'$$

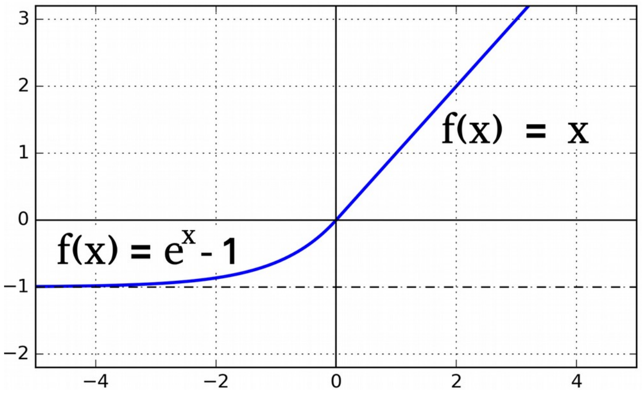
$$\frac{\partial E}{\partial y_i^{(p+1)}} = \sum \frac{\partial E}{\partial y_j^{(p+2)}} * \Theta_{ij}^{(p+1)} * (\sigma_j^{out}(\Sigma))'$$



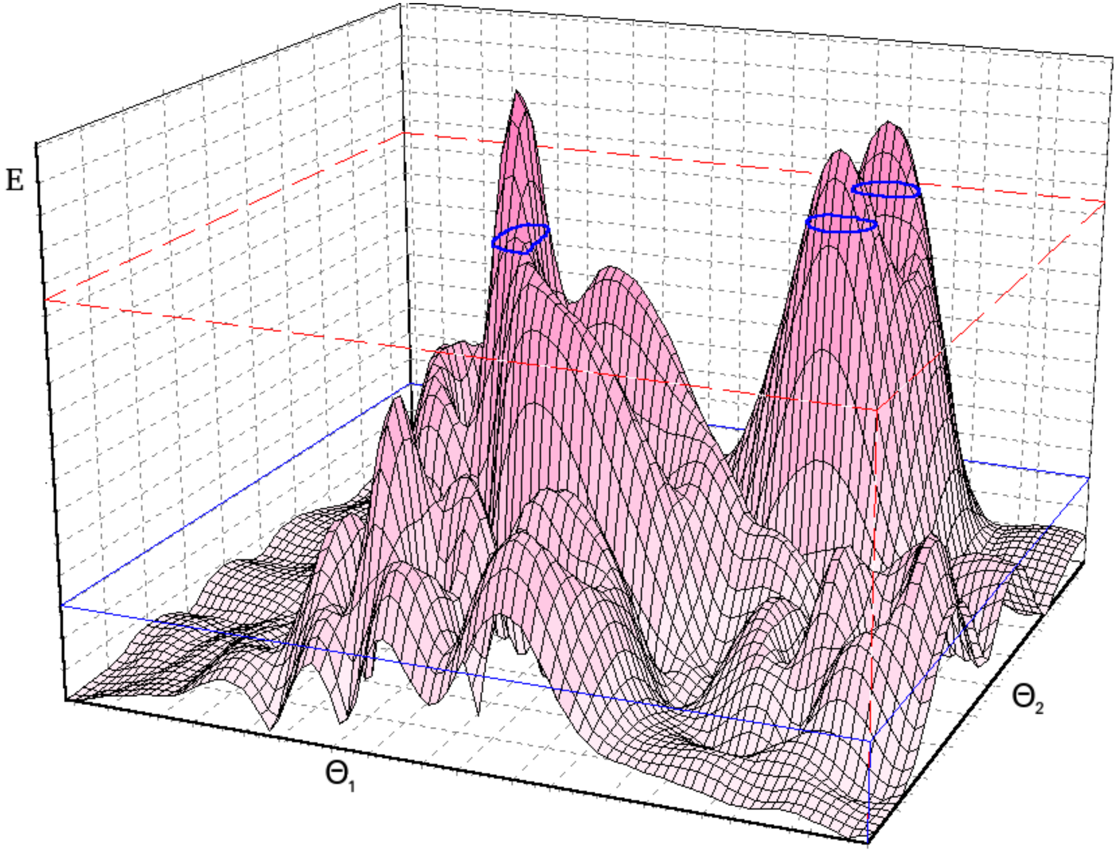
Если $|\theta_{ij}^{(p+1)} * (\sigma_j^{out}(\Sigma))'| < 1$, то возможно затухание градиента и замедление обучения.

Если $|\theta_{ij}^{(p+1)} * (\sigma_j^{out}(\Sigma))'| > 1$, то возможен взрыв градиента и нестабильность обучения.

Выход: использовать подходящую функцию активации и следить за весами связей.



Невоспроизводимость эксперимента



Если область, сходящаяся к хорошему минимуму, мала, то вероятность попадания в нее при случайной инициализации весов также мала.

tensorflow

https://sesc-infosec.github.io/