



Нейросети в РС

Андрей Зимовнов (Яндекс, ВШЭ)



Введение в нейросети



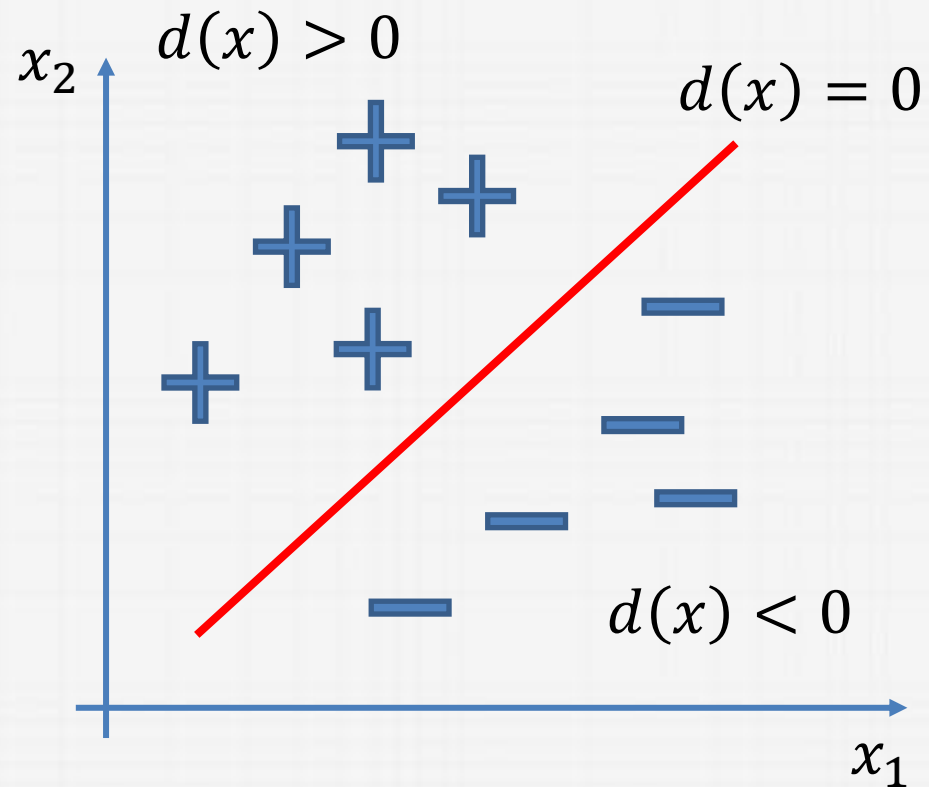
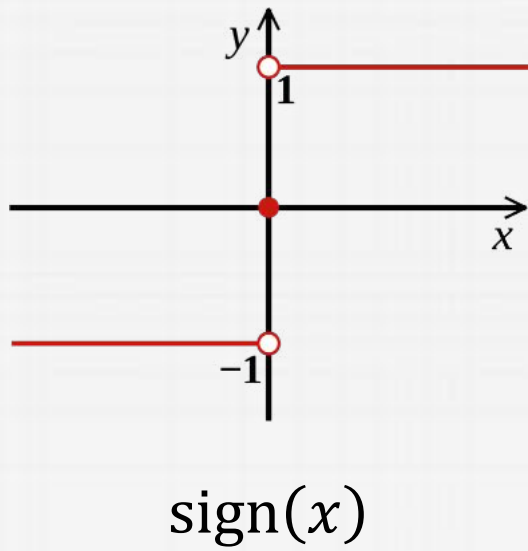
Линейная классификация

Признаки: $x = (x_1, x_2)$

Целевая переменная: $y \in \{+1, -1\}$

Функция принятия решения: $d(x) = w_0 + w_1x_1 + w_2x_2$

Алгоритм: $a(x) = \text{sign}(d(x))$

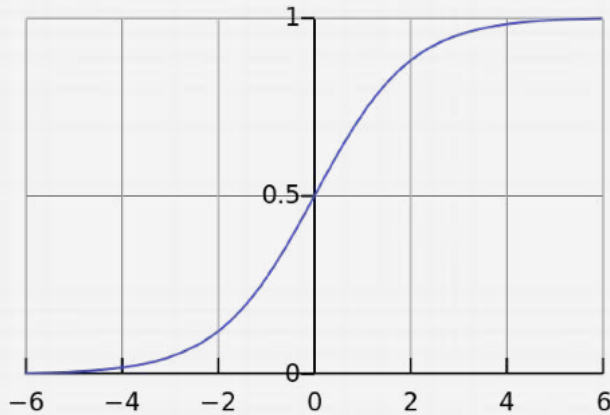


Логистическая регрессия

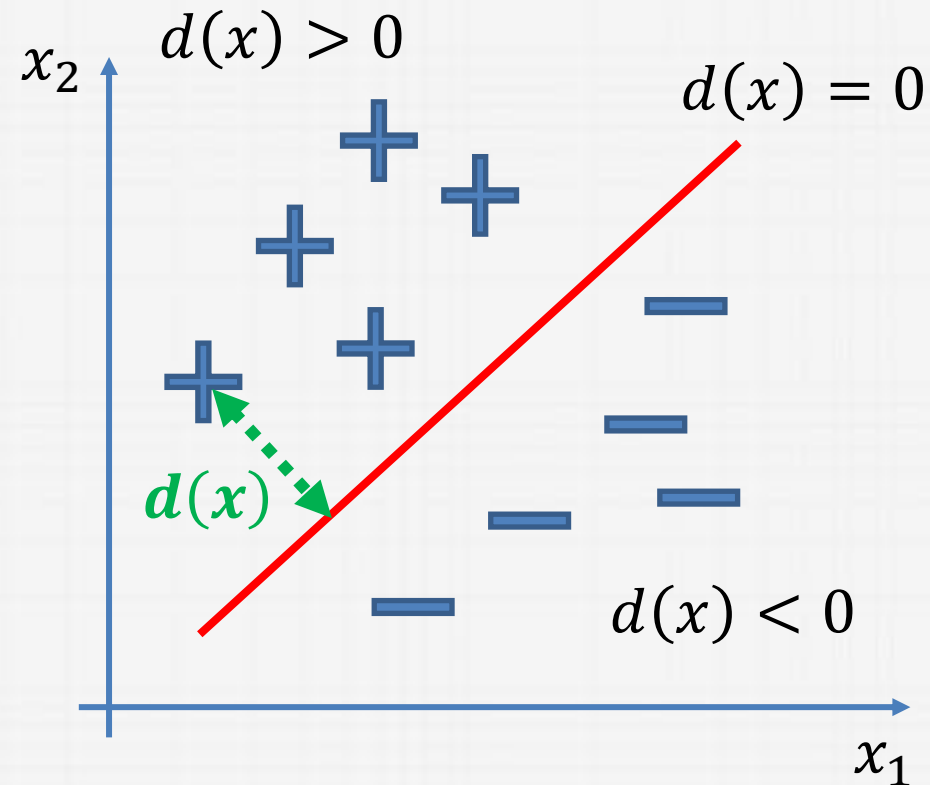
Предсказывает вероятность положительного класса (+1)

Функция принятия решения: $d(x) = w_0 + w_1x_1 + w_2x_2$

Алгоритм: $a(x) = \sigma(d(x))$



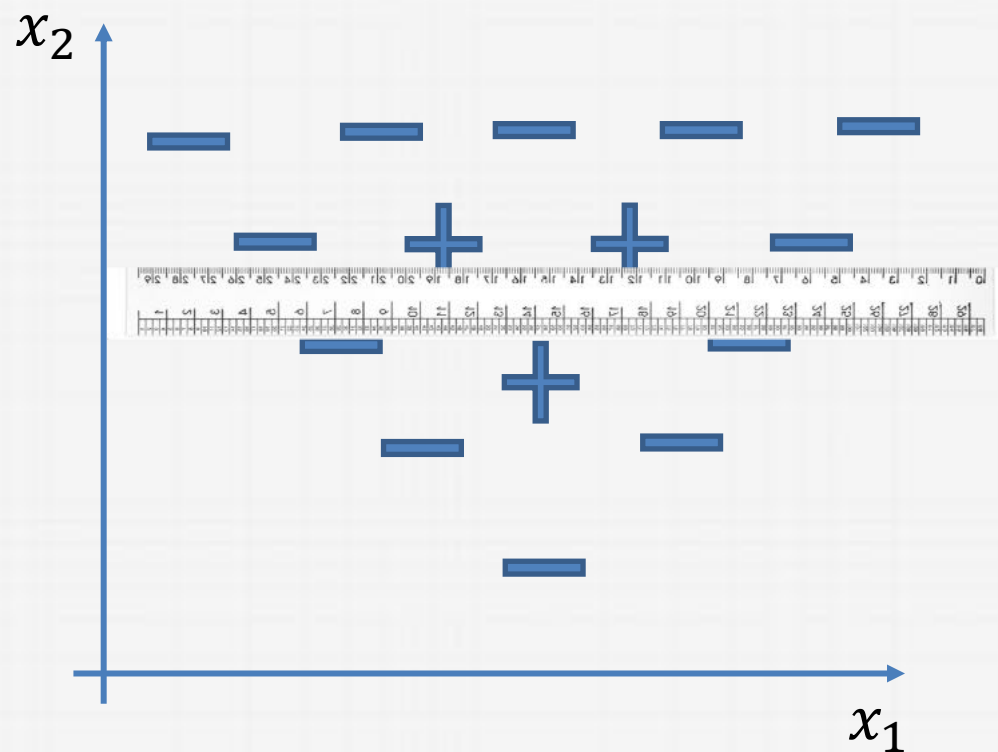
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



А что же делать тут?

Признаки: $x = (x_1, x_2)$

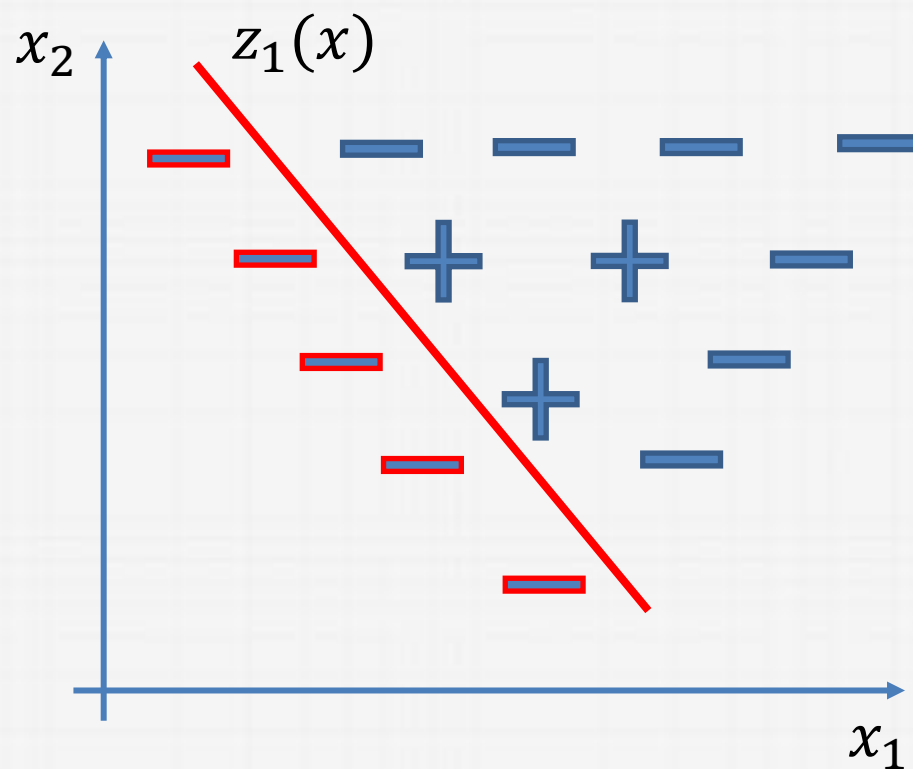
Целевой признак: $y \in \{+1, -1\}$



Решим подзадачу

Признаки: $x = (x_1, x_2)$

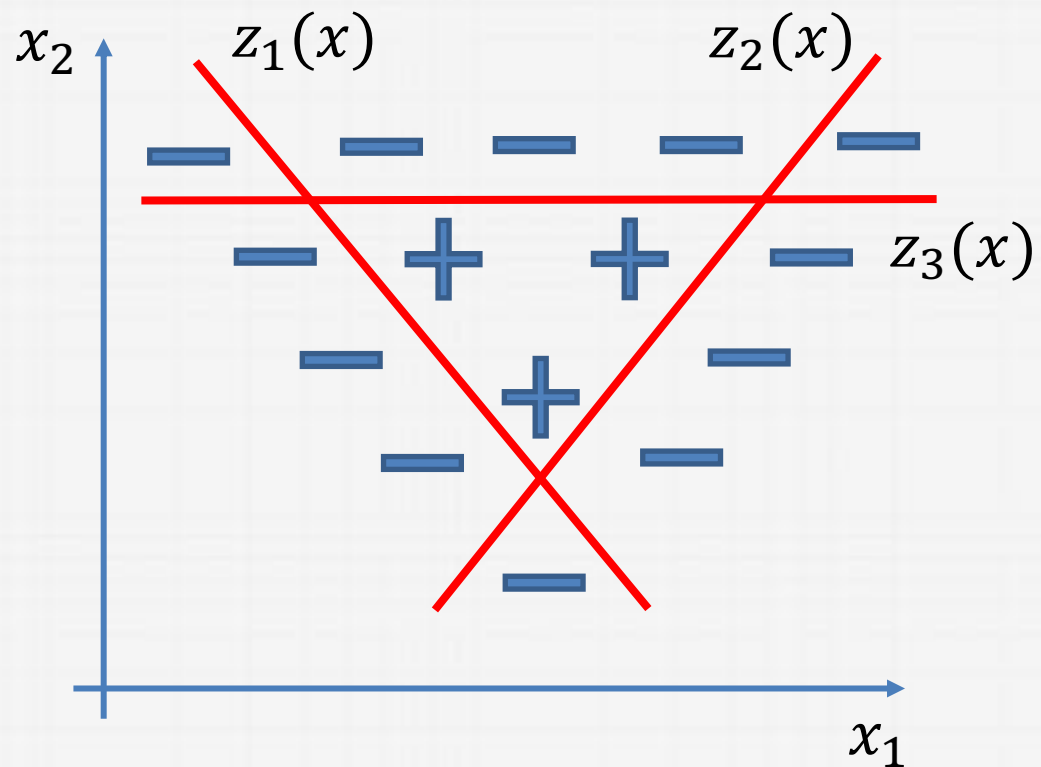
Целевой признак: $y \in \{+1, -1\}$



Своя линия для каждой подзадачи

Признаки: $x = (x_1, x_2)$

Целевой признак: $y \in \{+1, -1\}$



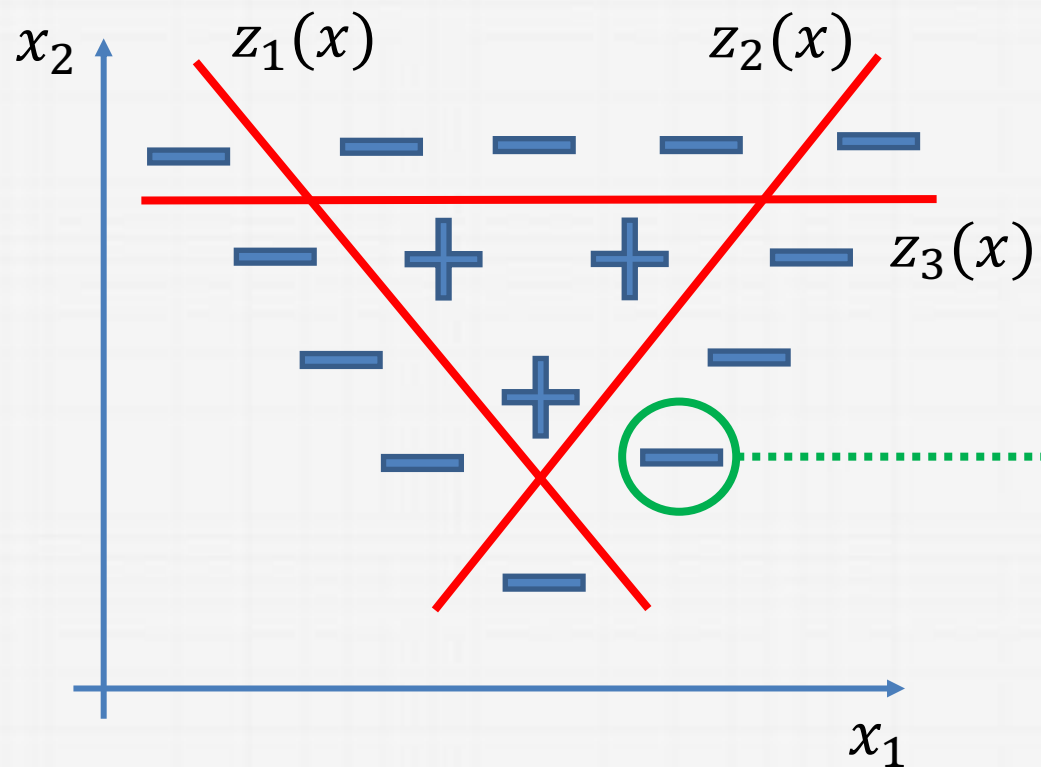
$$z_i = \sigma(w_{0,i} + w_{1,i}x_1 + w_{2,i}x_2)$$

Допустим, мы нашли
3 таких линии...

Используем предсказания линий

Признаки: $x = (x_1, x_2)$

Целевой признак: $y \in \{+1, -1\}$



Новые признаки:

$z_1(x)$	$z_2(x)$	$z_3(x)$	y
0.6	0.3	0.8	-1
0.7	0.7	0.7	+1

$(x_1, x_2) \rightarrow (z_1, z_2, z_3)$

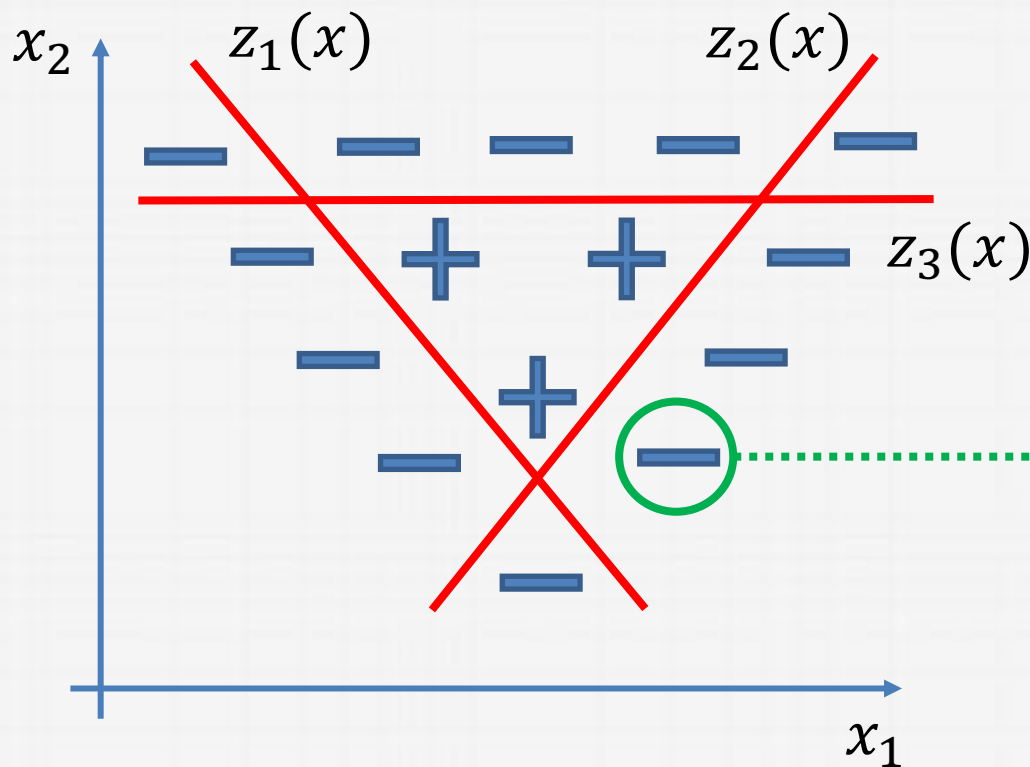
Что дальше?

$$z_i = \sigma(w_{0,i} + w_{1,i}x_1 + w_{2,i}x_2)$$

Финальная модель

Признаки: $x = (x_1, x_2)$

Целевой признак: $y \in \{+1, -1\}$



$$z_i = \sigma(\mathbf{w}_{0,i} + \mathbf{w}_{1,i}x_1 + \mathbf{w}_{2,i}x_2)$$

Новые признаки:

$z_1(x)$	$z_2(x)$	$z_3(x)$	y
0.6	0.3	0.8	-1
0.7	0.7	0.7	+1

$$(x_1, x_2) \rightarrow (z_1, z_2, z_3)$$

Строим финальную линейную модель:

$$a(x) = \sigma(\mathbf{w}_0 + \mathbf{w}_1 z_1(x) + \mathbf{w}_2 z_2(x) + \mathbf{w}_3 z_3(x))$$



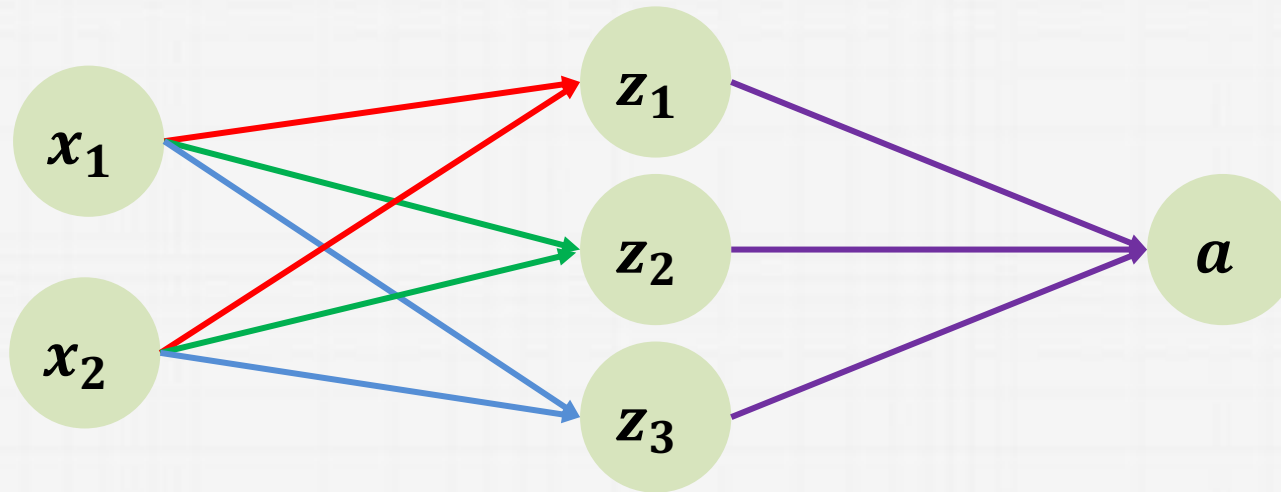
Параметров линий пока нет

Но понятно как все будет работать, если найти эти параметры:

$$z_i = \sigma(w_{0,i} + w_{1,i}x_1 + w_{2,i}x_2)$$

$$a(x) = \sigma(w_0 + w_1z_1(x) + w_2z_2(x) + w_3z_3(x))$$

Запишем наши вычисления в виде графа:

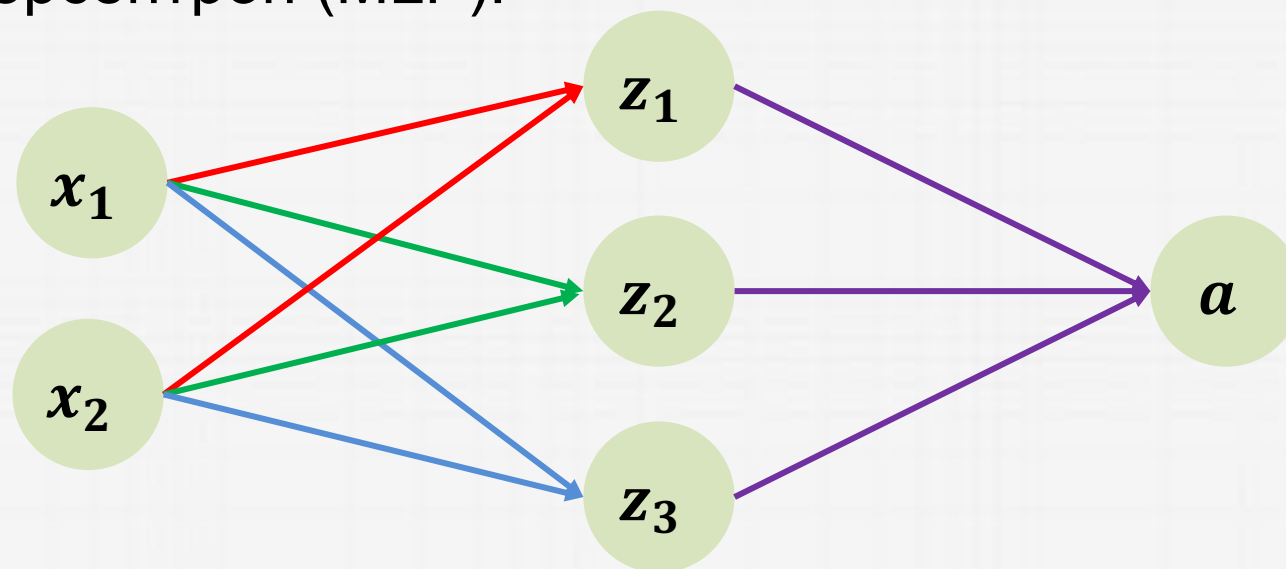


Вершины: вычисляемые переменные ($x_1, x_2, z_1, z_2, z_3, a$)

Ребра: зависимости (нам нужен x_1 и x_2 чтобы получить z_1)

У этого графа есть имя!

Многослойный персептрон (MLP):



Входной слой

Скрытый слой

Выходной слой

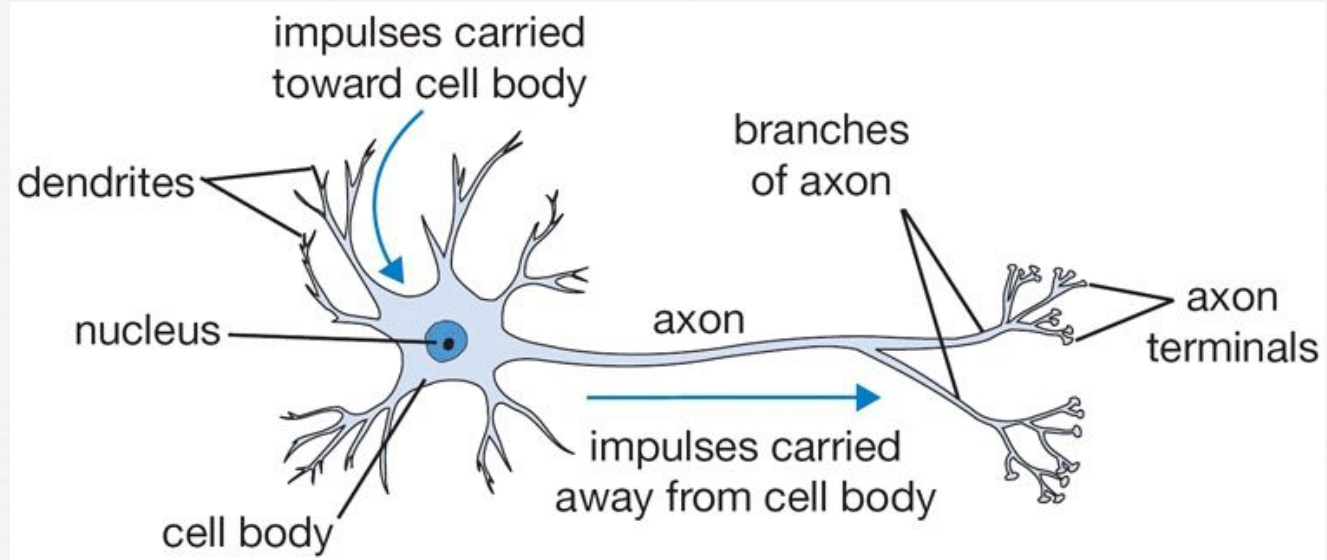
Признаки

Каждая вершина называется **нейроном**:

1. Линейная комбинация входов
2. Нелинейная функция **активации** (пример: $\sigma(x)$)

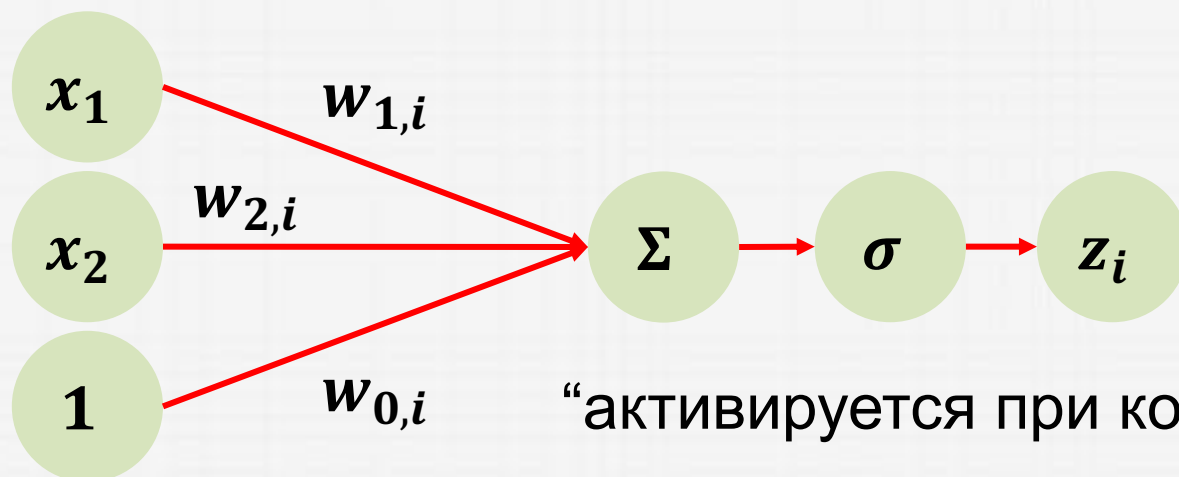
Почему нейрон?

Нейрон человека:



$$z_i = \sigma(w_{0,i} + w_{1,i}x_1 + w_{2,i}x_2)$$

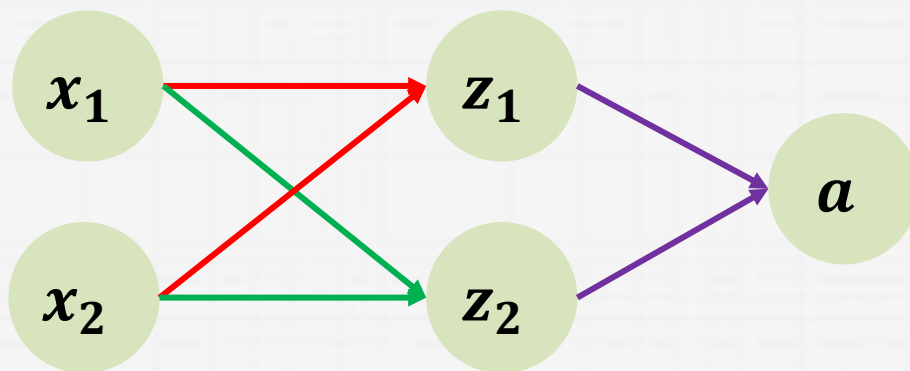
Математический нейрон:



“активируется при корреляции”

Нужны нелинейности в нейронах!

Давайте попробуем выкинуть $\sigma(x)$:



$$z_1 = w_{1,1}x_1 + w_{2,1}x_2$$

$$z_2 = w_{1,2}x_1 + w_{2,2}x_2$$

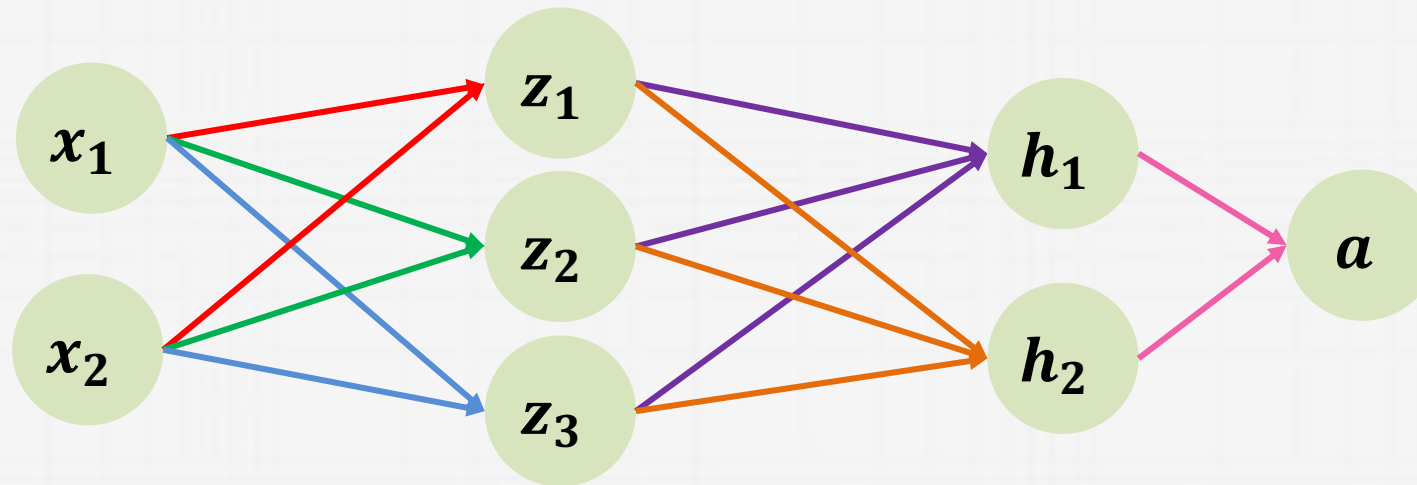
$$a = w_1z_1 + w_2z_2$$

Наш алгоритм становится линейной функцией!

$$a = (w_1w_{1,1} + w_2w_{1,2})x_1 + (w_1w_{2,1} + w_2w_{2,2})x_2$$

MLP – это простейший пример нейросети

MLP может иметь много скрытых слоев:



Архитектура MLP:

Кол-во слоев

Кол-во нейронов в каждом слое

Какую активацию использовать

Скрытый слой в MLP называют:

- Dense layer (плотный)
- Fully-connected layer (полно-связный)



Как найти параметры MLP?

Мы знаем как выучить параметры логистической регрессии – **SGD!**

Давайте здесь **сделаем то же самое**, ведь финальная функция дифференцируемая!

Быстрый и эффективный способ вычисления градиента для любого дифференцируемого графа вычислений называется **back-propagation** (обратное распространение ошибки)

Застревает в локальных минимумах

**На практике – не беда!
Просто перезапустим!**

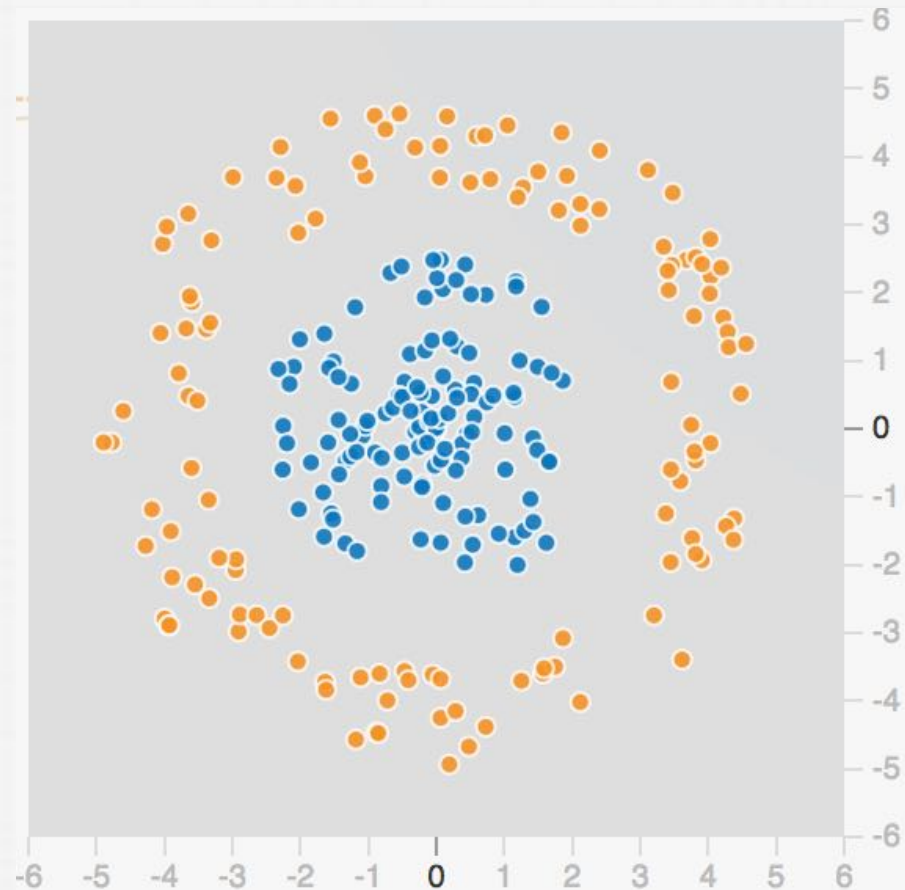


Bob chillin at a local optima



Демо в TensorFlow Playground

<http://playground.tensorflow.org>



Можно решать сложные задачи

DATA

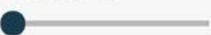
Which dataset do you want to use?



Ratio of training to test data: 50%



Noise: 0



Batch size: 9



REGENERATE

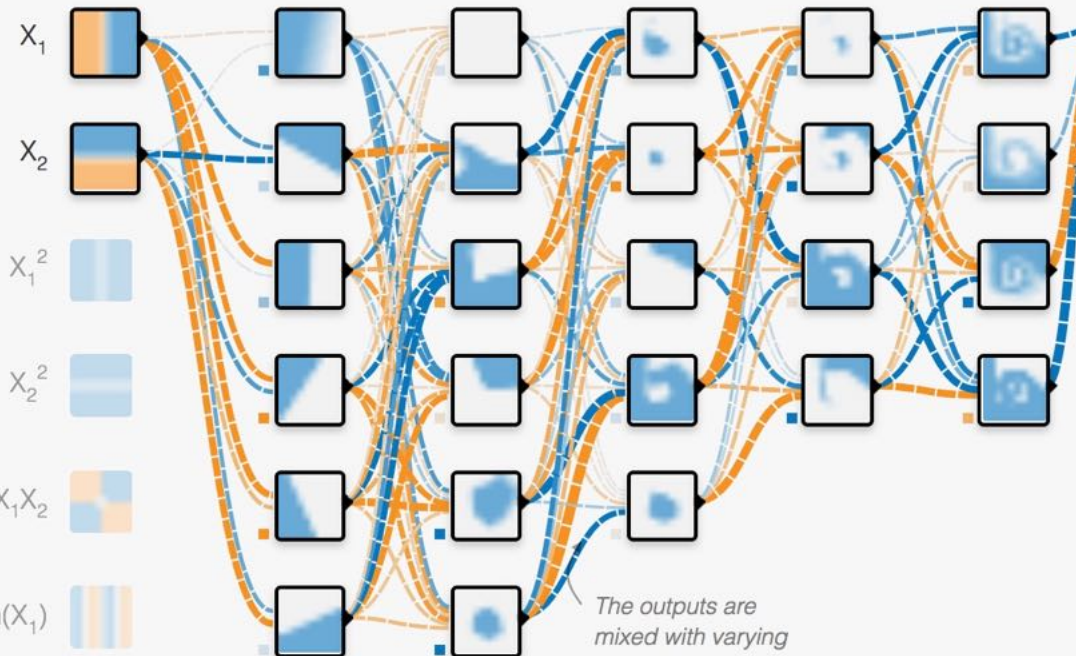
FEATURES

Which properties do you want to feed in?



5 HIDDEN LAYERS

6 neurons 6 neurons 5 neurons 4 neurons 4 neurons

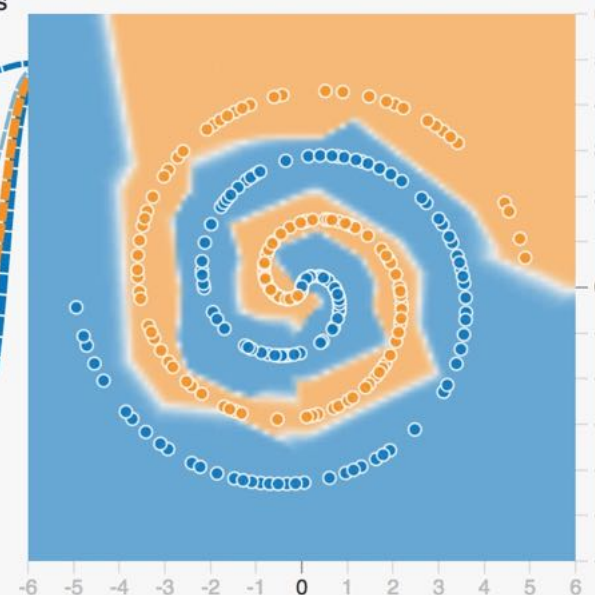


This is the output from one **neuron**. Hover to see it larger.

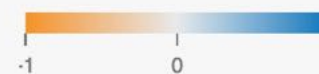
The outputs are mixed with varying **weights**, shown by the thickness of the lines.

OUTPUT

Test loss 0.011
Training loss 0.000



Colors shows data, neuron and weight values.



☐ Show test data

☐ Discretize output

* с функцией активации ReLU

От SVD к нейросетям

Что такое embedding

~100 тыс. слов

good	movie	very	a	did	like
------	-------	------	---	-----	------

very →

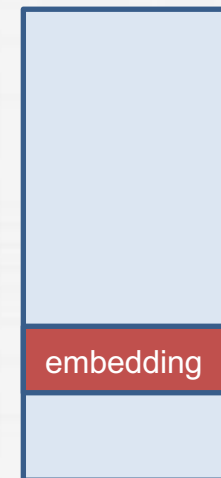
0	0	1	0	0	0
---	---	---	---	---	---

Dense слой

300 чисел

	1	
--	---	--

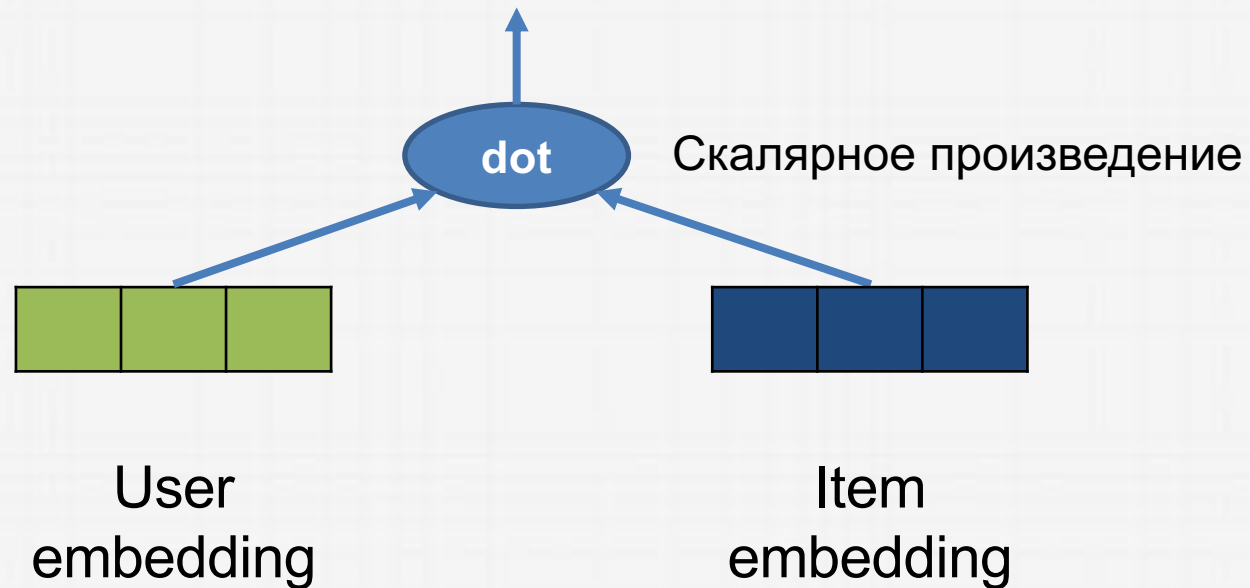
x



Матрица *имбедингов*

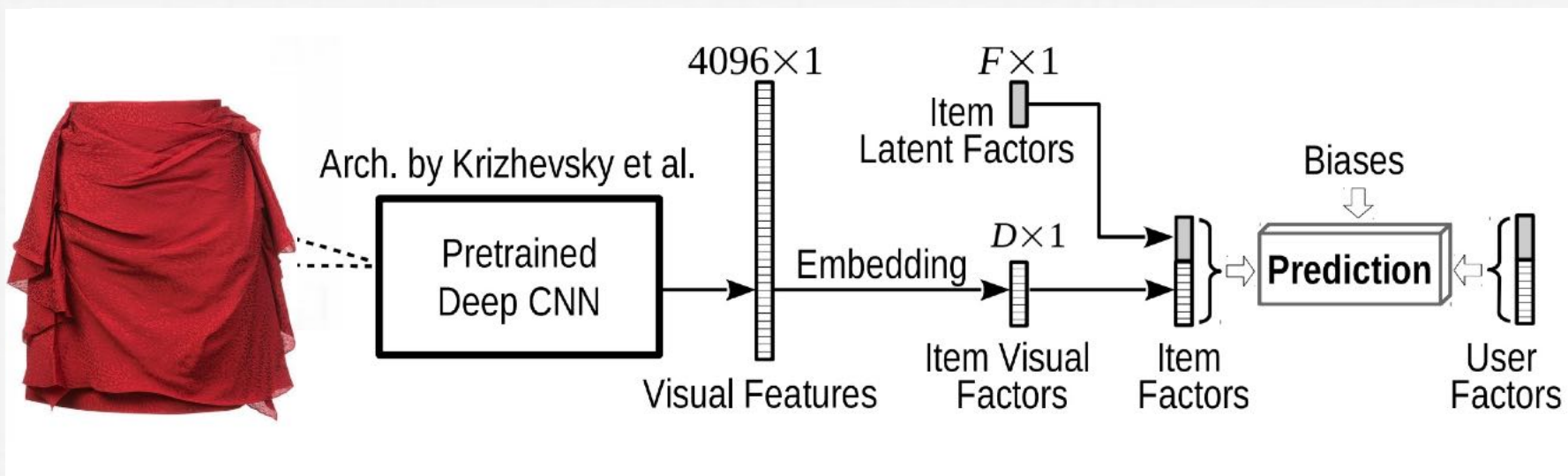
Funk SVD как нейросеть

Обучаем при помощи SGD



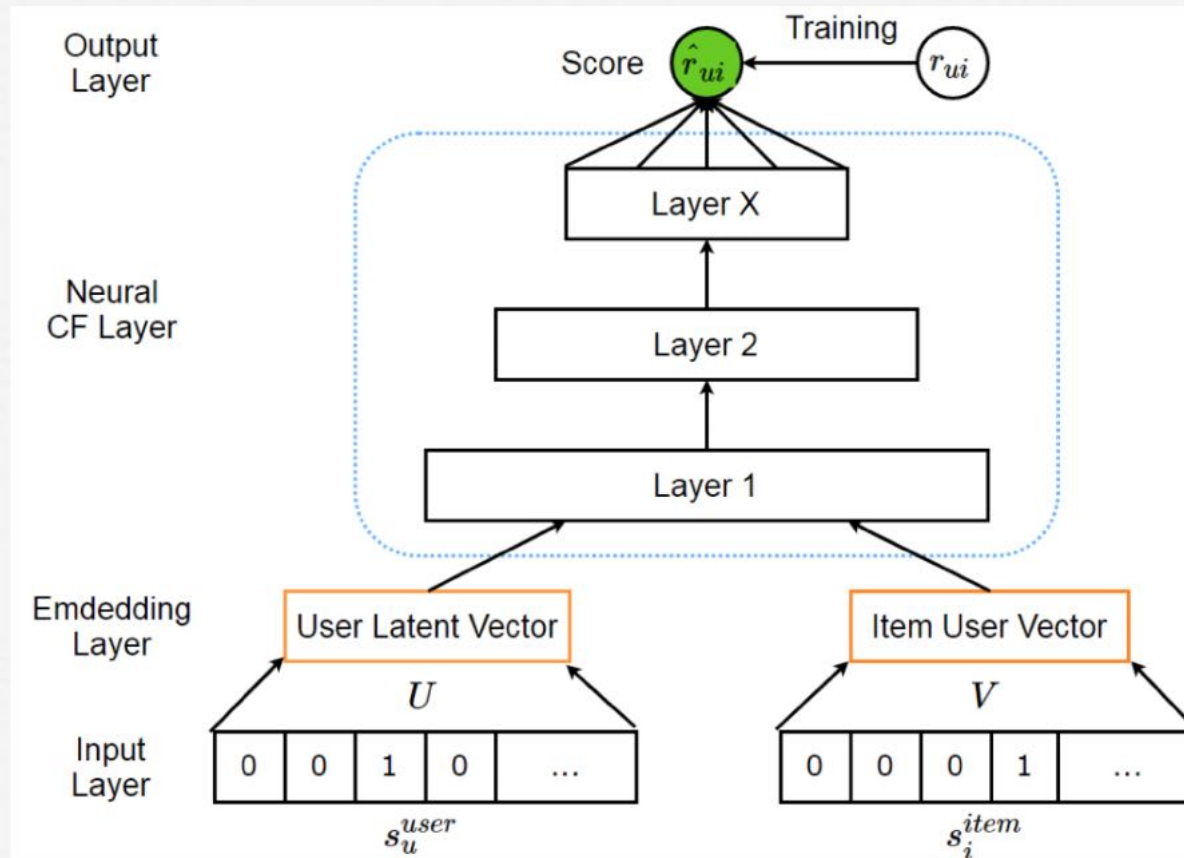
Content-boosted CF

Добавим в профиль товара имбединг его контента!



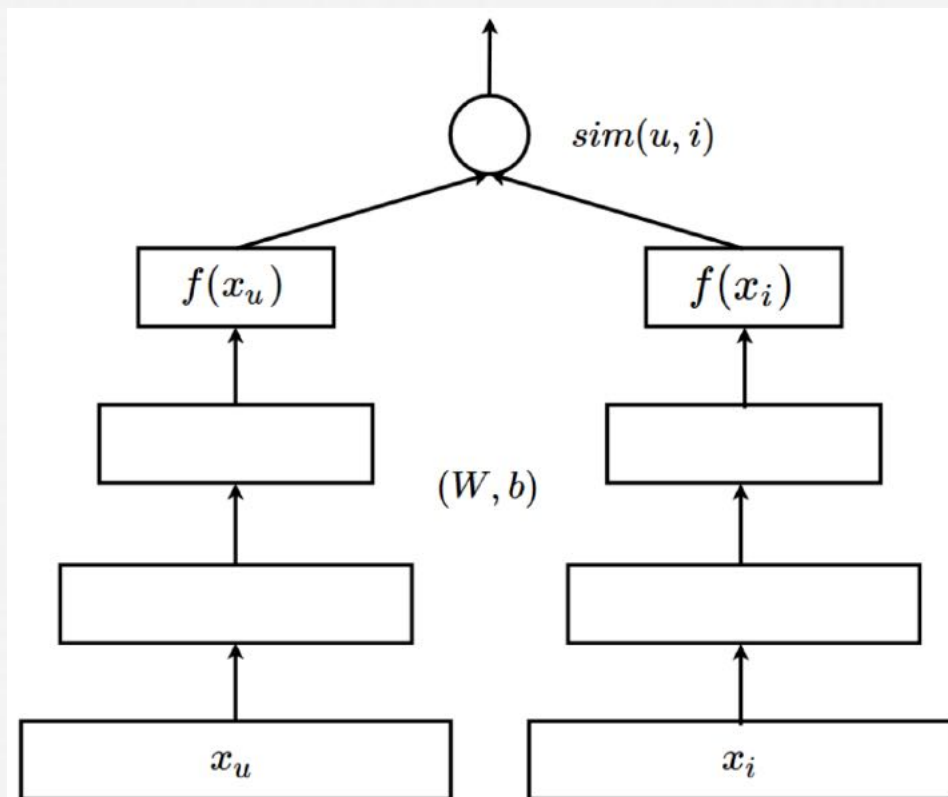
Контент бывает разный, например, картинки

Neural Collaborative Filtering



В онлайн
считать
затратно

Работает в онлайне!



Признаки
пользователя

Признаки
товара

Например:
мешки слов



DL фреймворк – это удобно!

Можно использовать крутые оптимайзеры (Adam, ...)

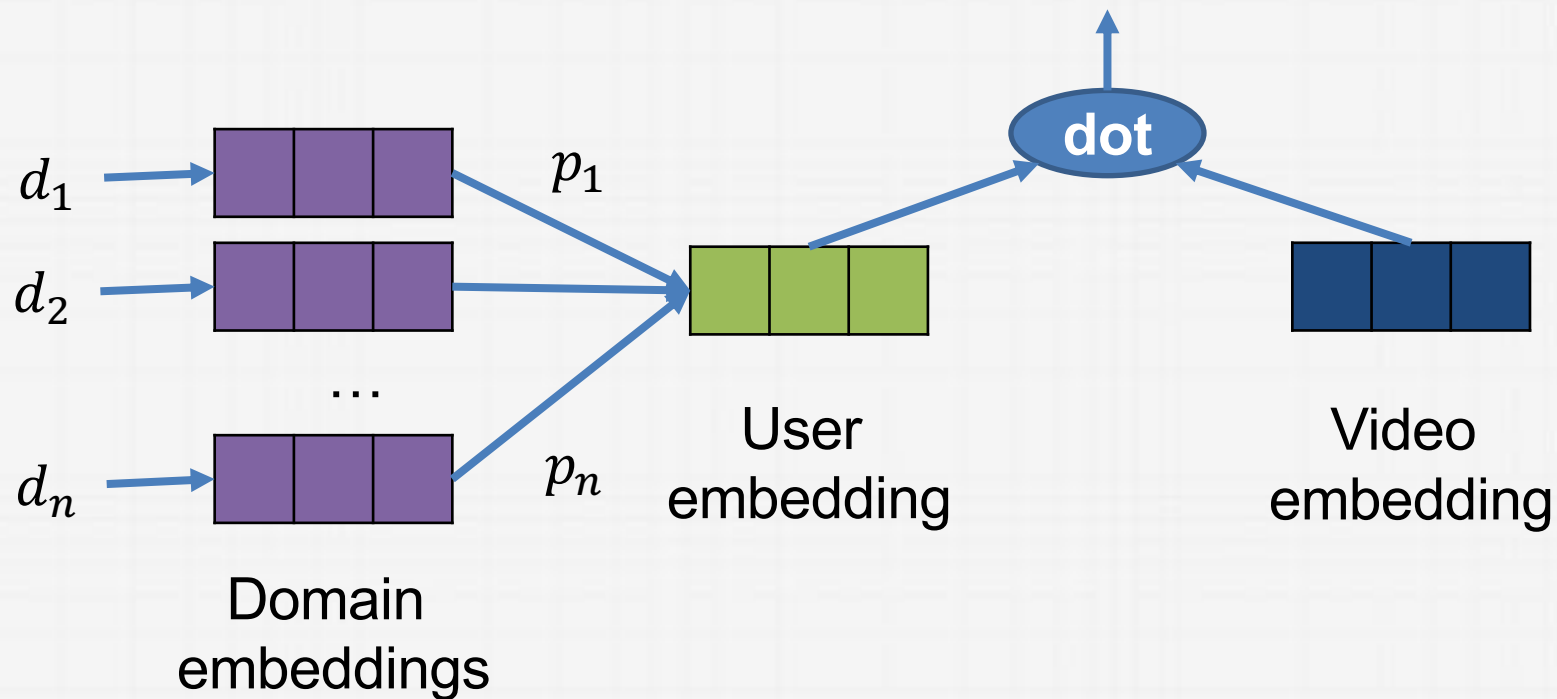
Можно имбеддить картинки, текст, ...

Описываем какие взаимосвязи имеют смысл.

Детали выучит сеть.

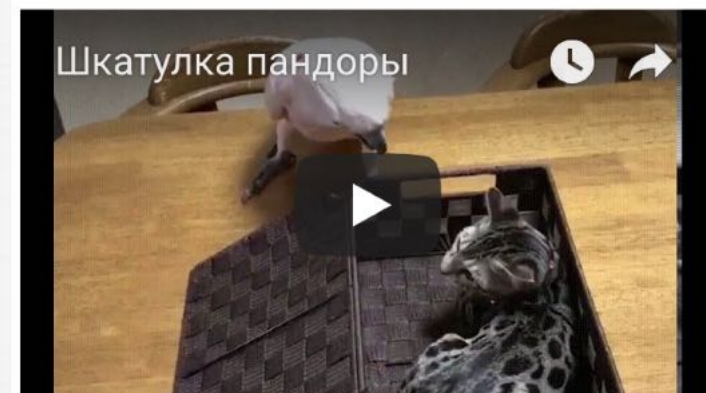
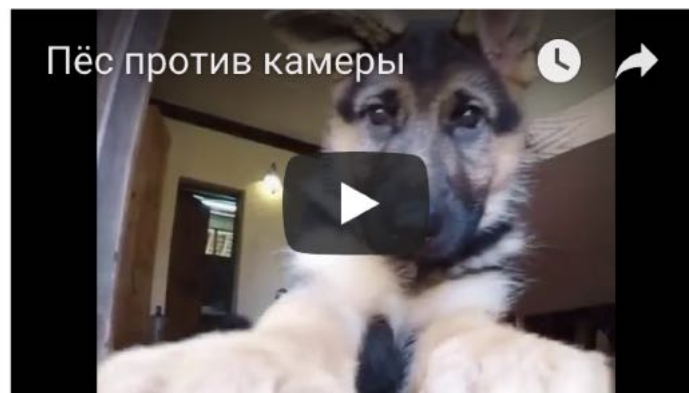
Задача: домены → видео

Хотим помочь холодному старту рекомендаций видео

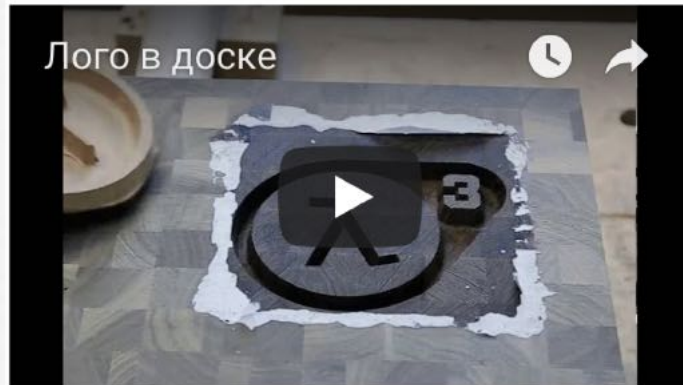


Знаем куда ходит пользователь

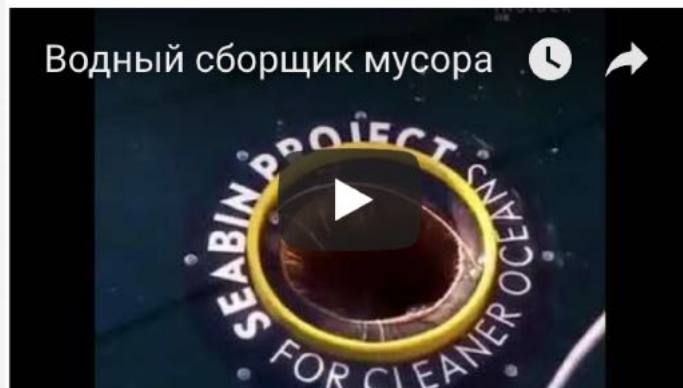
Похожие видео (по dot)



[0.94703776]



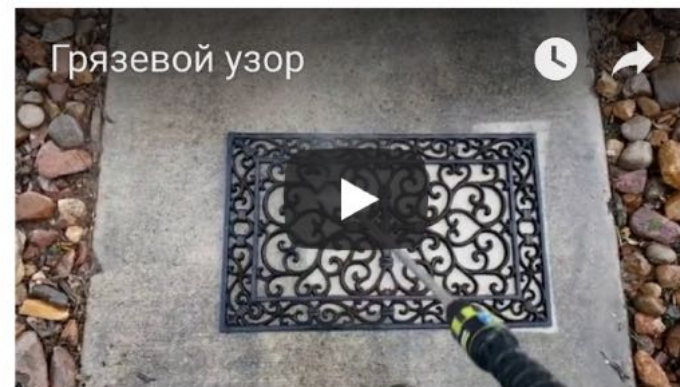
[0.9381]



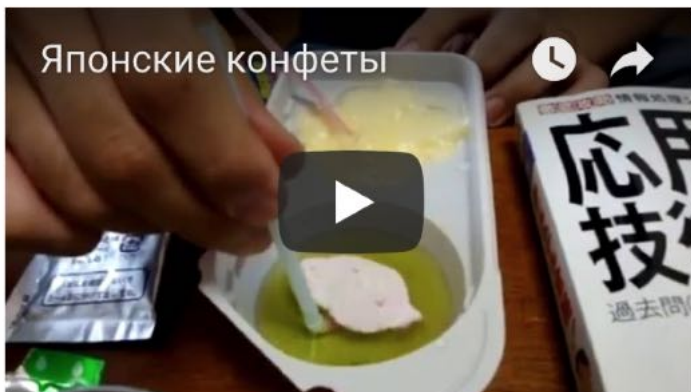
[0.92087275]



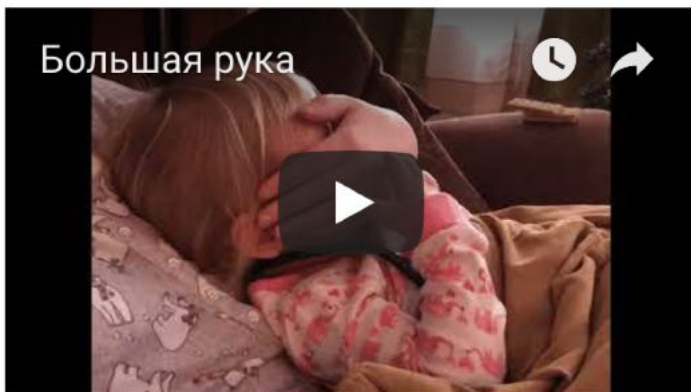
[0.9200871]



[0.89342886]



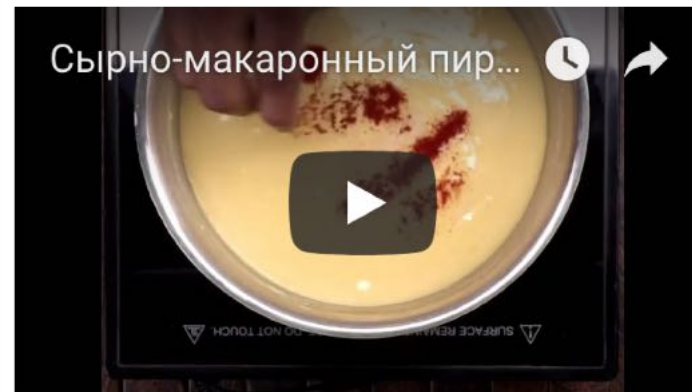
[0.89316946]

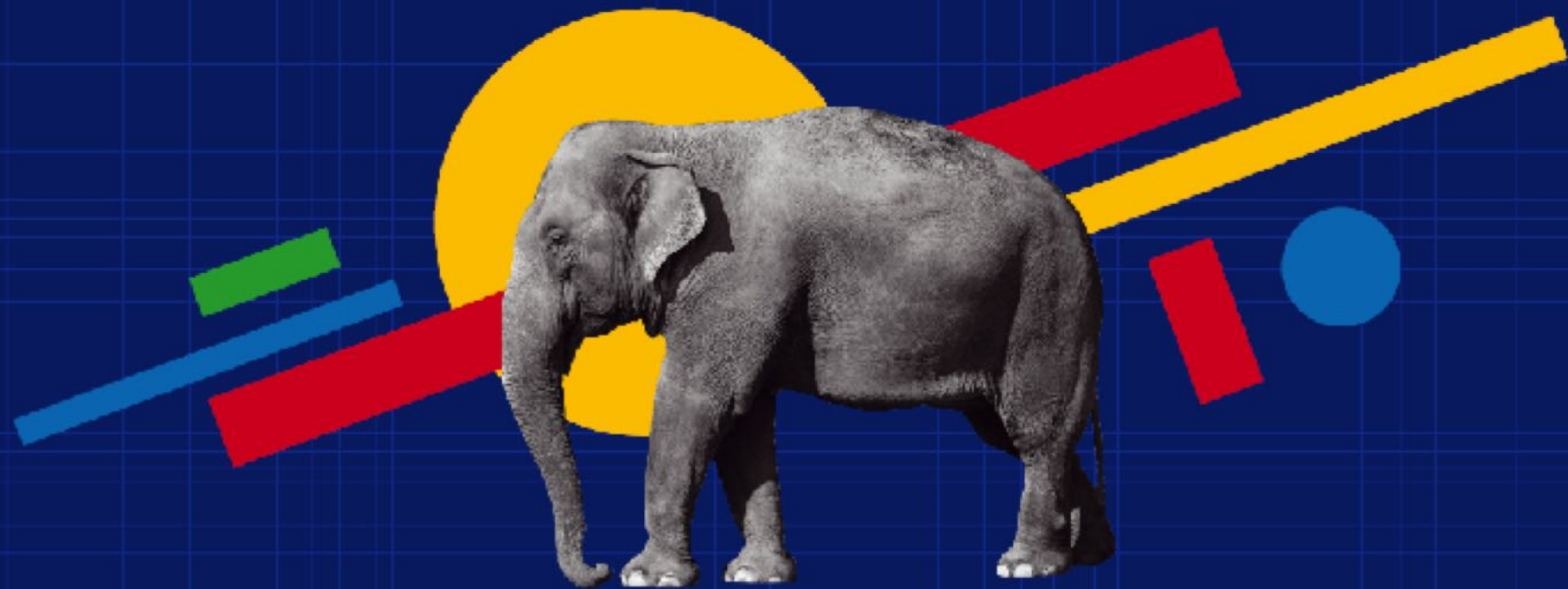


[0.8903276]



[0.88709587]





BIG DATA IS LOVE

NEWPROLAB.COM