

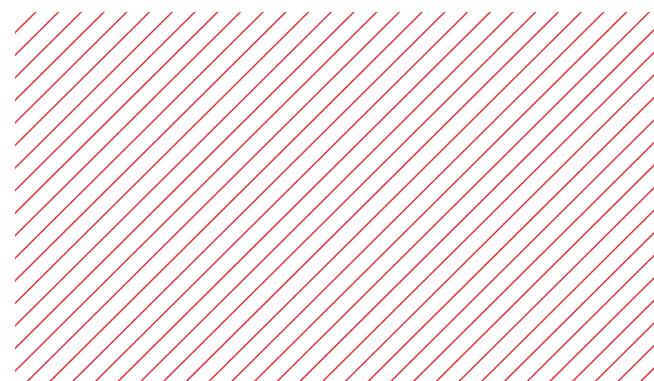
академия
больших
данных



Computer Vision Introduction and recap

Иван Капрухин

Ведущий программист-исследователь в
команде машинного зрения





Вопрос 1: читаемость

“Доктор, я не могу больше! Куда бы я ни посмотрел, о чём бы ни подумал, мне везде мерещится одно — шрифты, шрифты, шрифты!”

24. Сядьте, успокойтесь, выпейте чаю, съешьте ещё этих мягких французских булок

22. Сядьте, успокойтесь, выпейте чаю, съешьте ещё этих мягких французских булок

20. Сядьте, успокойтесь, выпейте чаю, съешьте ещё этих мягких французских булок

18. Сядьте, успокойтесь, выпейте чаю, съешьте ещё этих мягких французских булок

16. Сядьте, успокойтесь, выпейте чаю, съешьте ещё этих мягких французских булок

14. Сядьте, успокойтесь, выпейте чаю, съешьте ещё этих мягких французских булок

12. Сядьте, успокойтесь, выпейте чаю, съешьте ещё этих мягких французских булок



Вопрос 2

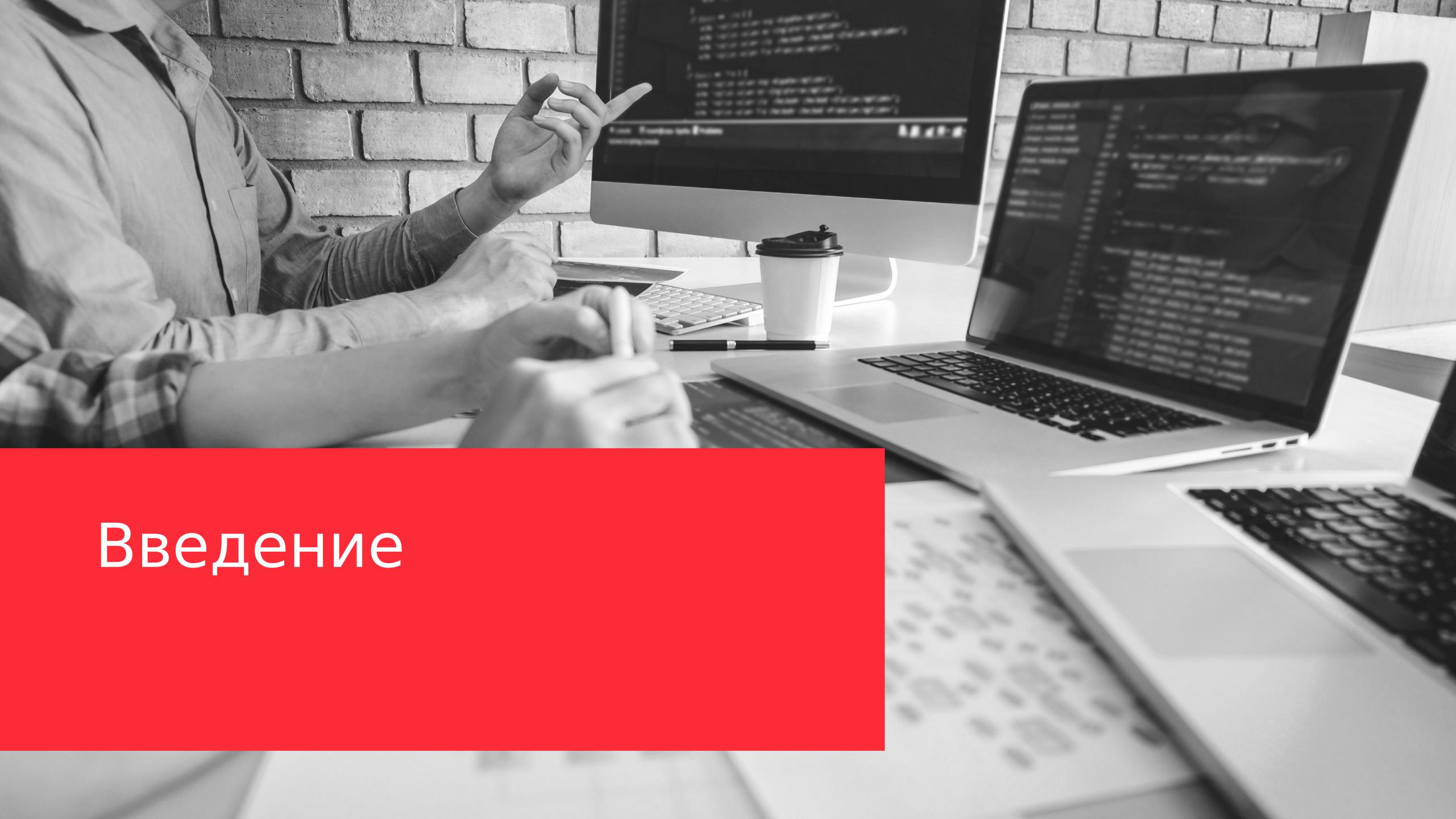
Приходилось ли вам использовать машинное обучение?



Вопрос 3

Приходилось ли вам использовать нейронные сети?

Введение

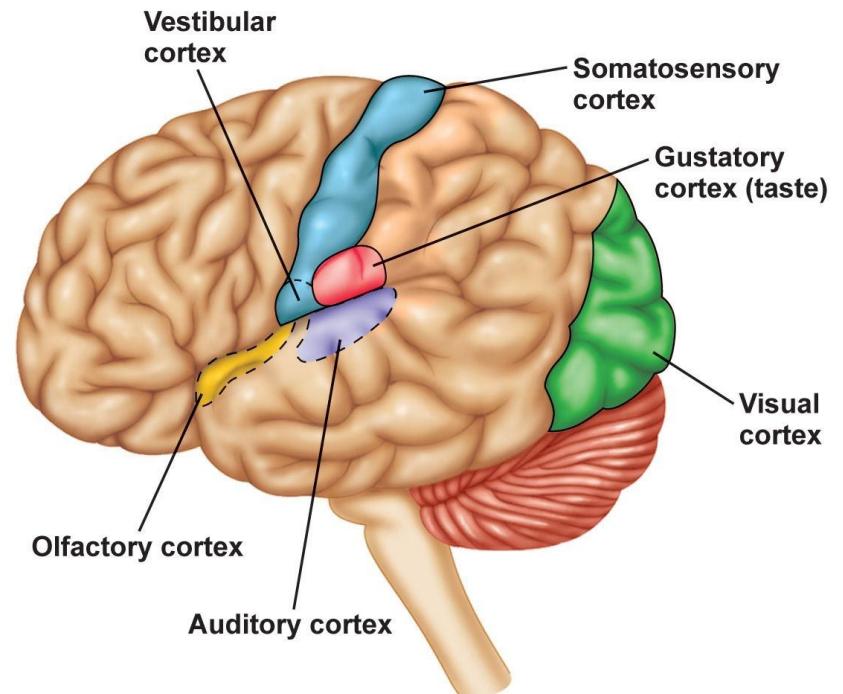


Что изучает Computer Vision

Два подхода к определению CV.

Биология: повторить поведение зрительной коры головного мозга человека.

Инженерия: автоматически извлекать высокоуровневую информацию из картинок и видео.



© 2011 Pearson Education, Inc.

* Huang T. Computer vision: Evolution and promise. 1996.

История развития CV

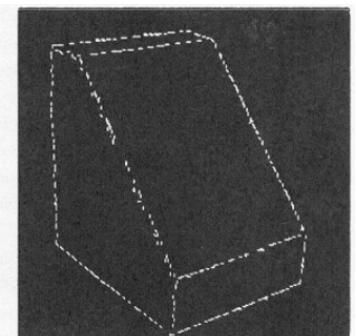
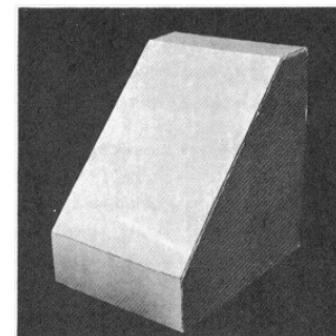
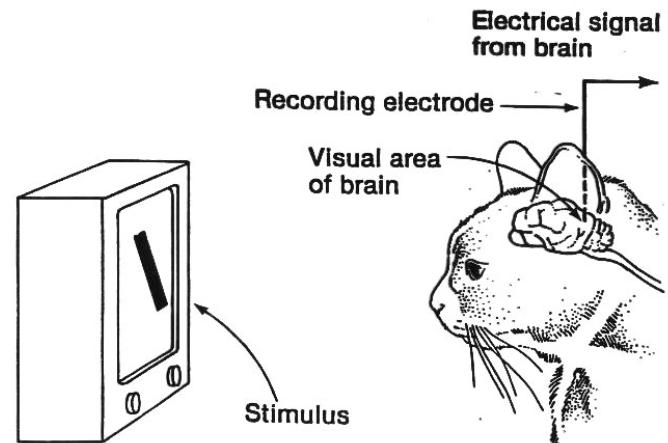
Появление CV: 1950–1970

1957: Цифровой сканер

1959: Изучение зрительной коры

1963: Реконструкция формы 3D моделей

1966: The Summer Vision Project



* Hubel D. H., Wiesel T. N. Receptive fields of single neurones in the cat's striate cortex. 1959.

* Roberts L. G. Machine perception of three-dimensional solids. 1963.

* Papert S. A. The summer vision project. 1966.

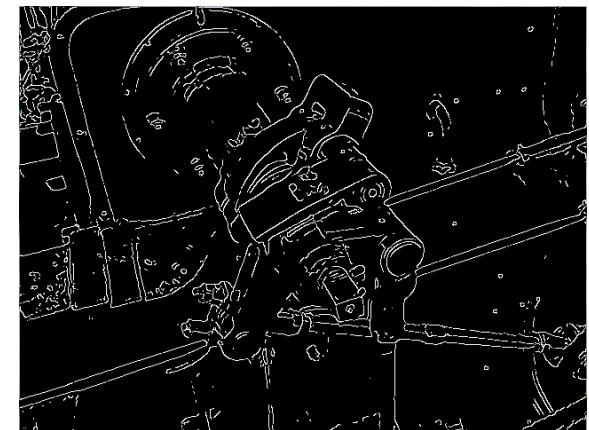
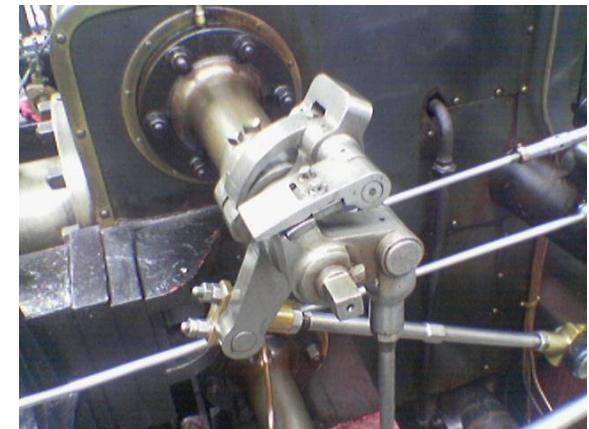
История развития CV

Классические алгоритмы CV: 1970–1990

1981: Оценка оптического потока (движения объектов на изображении)

1982: Теория анализа визуальной информации

1986: Детектор границ Кэнни



* Horn B. K. P., Schunck B. G. Determining optical flow. 1981.

* Marr D., Vision: A computational investigation into the human representation and processing of visual information. 1982.

* Bao P., Zhang L., Wu X. Canny edge detection enhancement by scale multiplication. 2005.

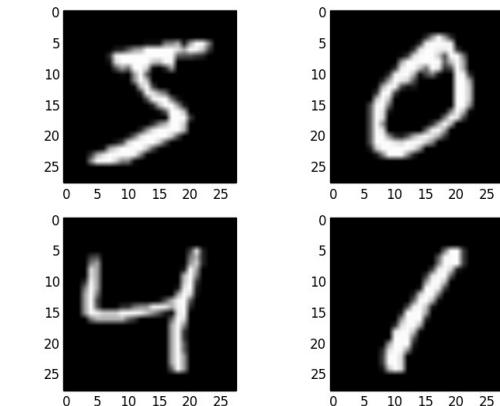
История развития CV

Машинное обучение в CV: 1990–2000

1991: Распознавание лиц с использованием собственных векторов



1998: CNN для распознавания рукописных цифр



* Turk M. A., Pentland A. P. Face recognition using eigenfaces. 1991.

* LeCun Y. et al. Gradient-based learning applied to document recognition. 1998.

История развития CV

Локальные дескрипторы: конец 2000–2010

1999: SIFT: локальные признаки, инвариантные к масштабу

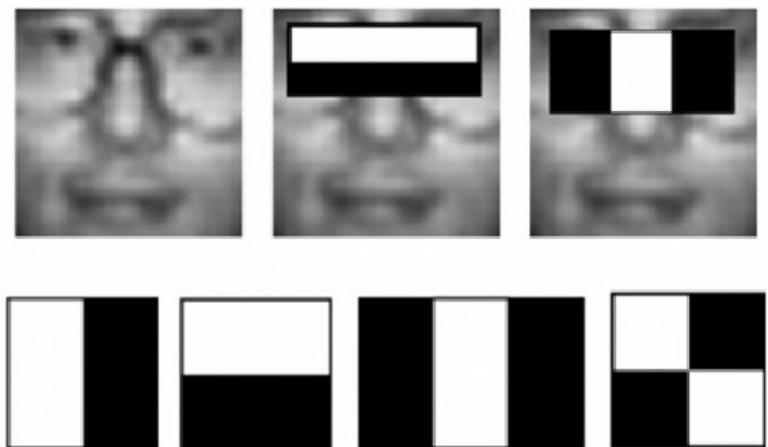
2001: Метод Виолы — Джонса для поиска лиц

2006: SURF: быстро вычислимые локальные признаки



Области применения:

- 3D реконструкция
- поиск объектов
- распознавание лиц



* Lowe D. G. Object recognition from local scale-invariant features. 1991.

* Viola P. et al. Robust real-time object detection. 2001.

* Bay H., Tuytelaars T., Van Gool L. Surf: Speeded up robust features. 2006.

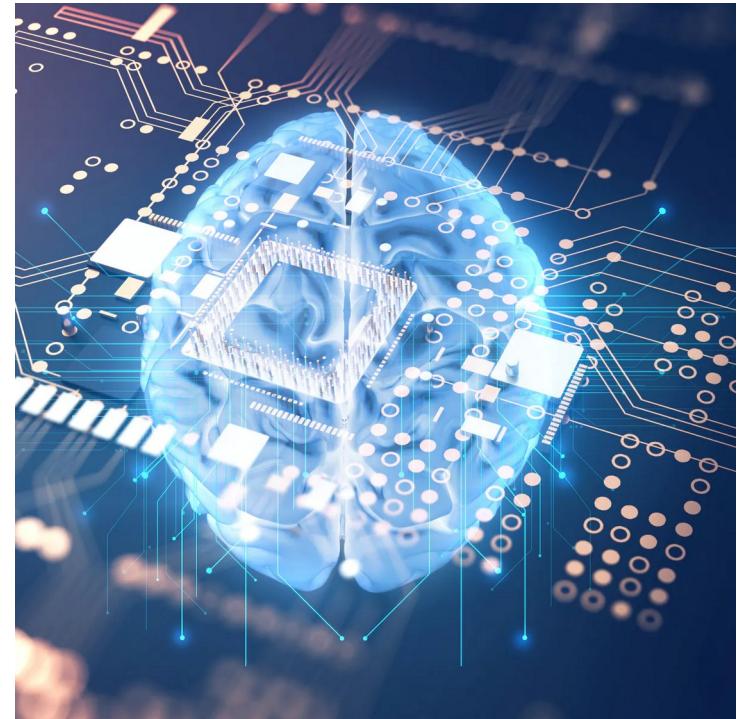
История развития CV

Глубокое обучение: с 2010

2012: AlexNet: прорыв в распознавании изображений из ImageNet

2014: VGG: больше слоев, выше качество

2015: ResNet: изменение архитектуры, качество выше, чем у человека



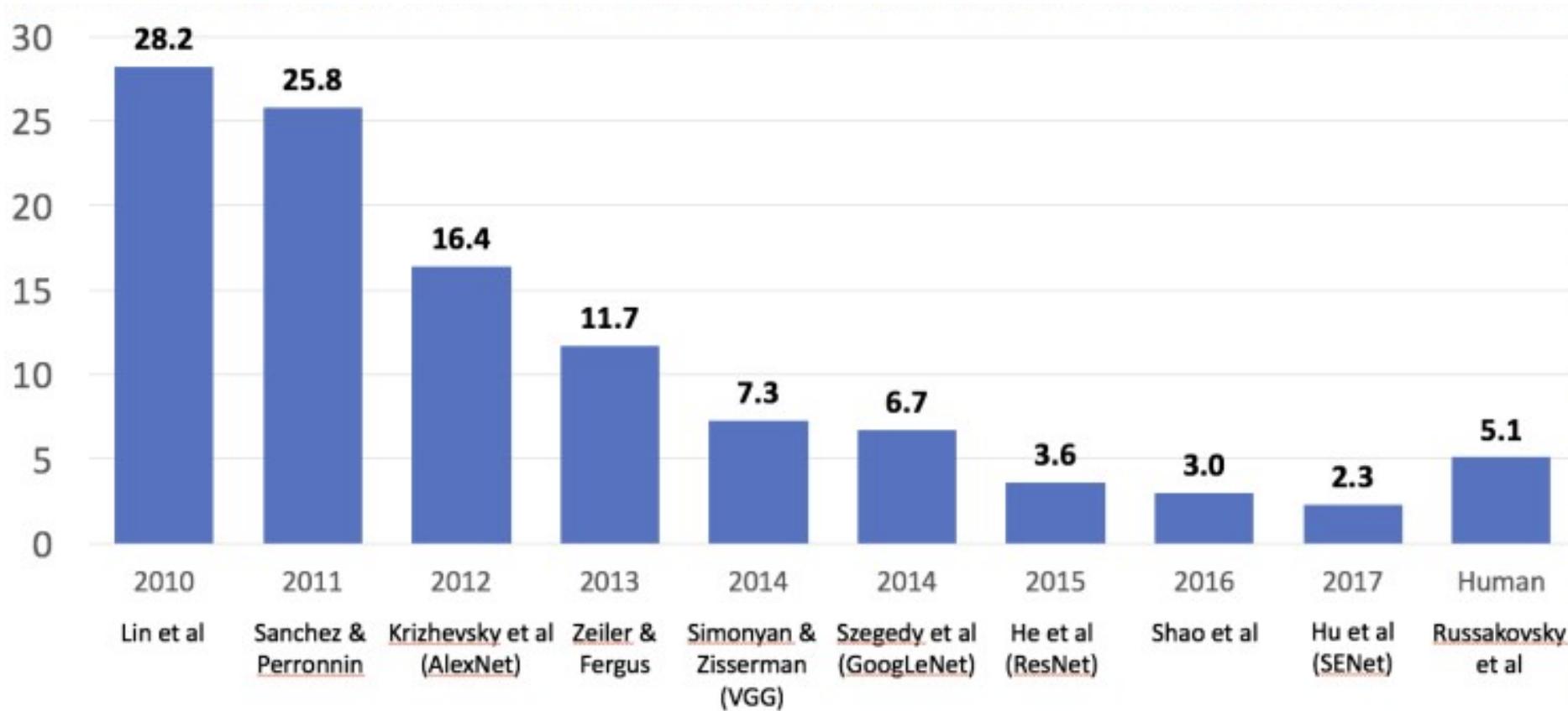
* Krizhevsky A., Sutskever I., Hinton G. E. Imagenet classification with deep convolutional neural networks. 2012.

** Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014.

*** He K. et al. Deep residual learning for image recognition. 2015.

История развития CV

Глубокое обучение: с 2010





Области науки и технологии

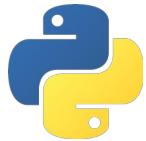
Математика:

- Линейная алгебра (векторы, матрицы, линейные отображения)
- Анализ (непрерывность, производная)
- Теория информации (энтропия, перекрёстная энтропия)
- Оптимизация (градиентный спуск)

Инженерия:

- Обработка сигналов (фильтры, свертки, кодирование изображений)
- Машинное обучение (кластеризация, классификация, нейронные сети)
- Программирование (больше работы с библиотеками, меньше алгоритмы)

Инструменты



Python: основной язык

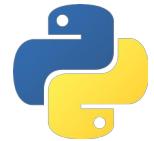
- Популярный язык для CV и DL
- Множество библиотек
- Легко читается

```
def crop_image(image, bbox):
    """Crop image to bounding box.

Args:
    image: Image in WHC format.
    bbox: Bounding box in XYWHA format (A for angle).

Returns:
    Crop in WHC format.
"""
if image.shape[-1] > 4:
    raise ValueError("Image channels should be at last dimension")
src_points = get_bbox_points(bbox).astype("float32")
dst_points = np.array([[0, bbox[BBOX_HEIGHT] - 1],
                      [0, 0],
                      [bbox[BBOX_WIDTH] - 1, 0],
                      [bbox[BBOX_WIDTH] - 1, bbox[BBOX_HEIGHT] - 1],
                      dtype="float32")]
M = cv2.getPerspectiveTransform(src_points, dst_points)
crop = cv2.warpPerspective(image, M, (int(bbox[BBOX_WIDTH]), int(bbox[BBOX_HEIGHT])))
return crop
```

Инструменты



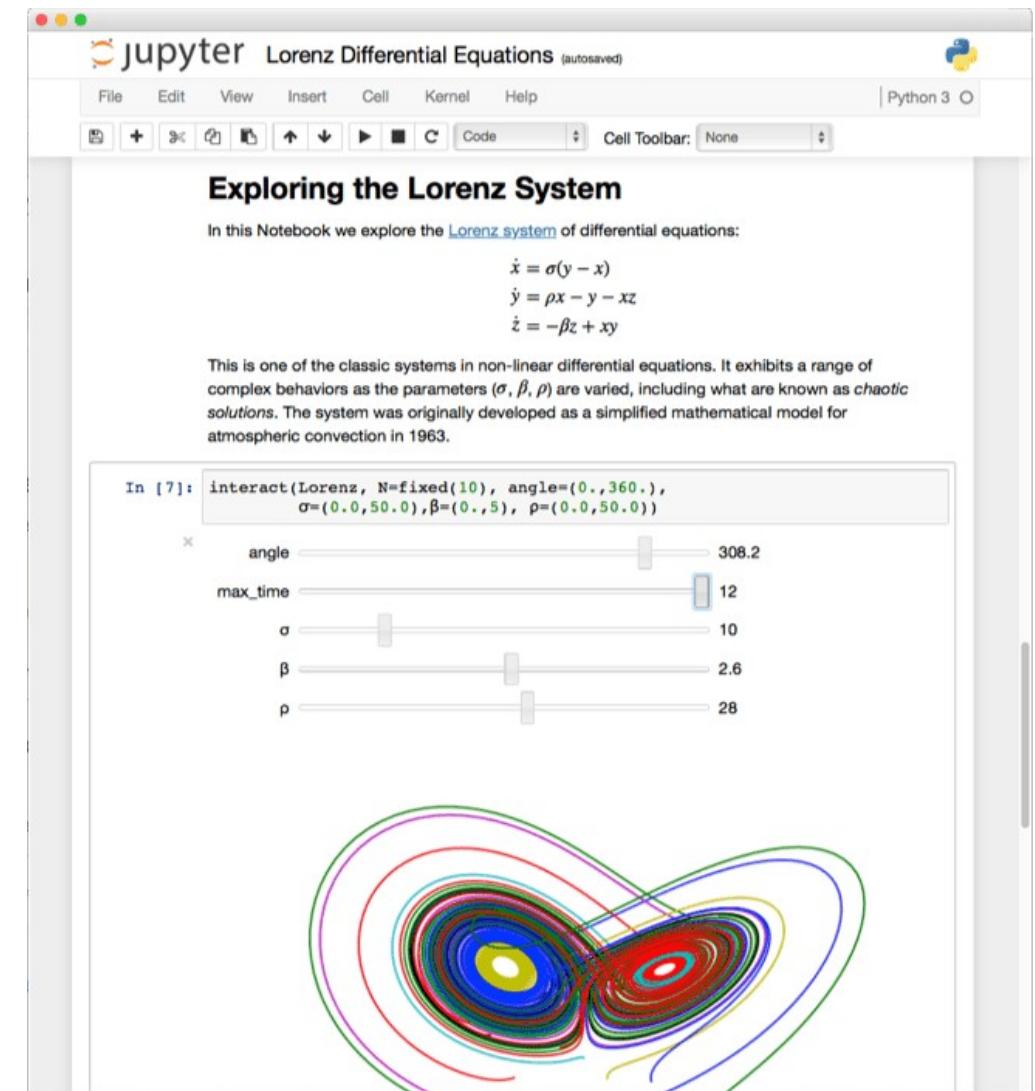
Python: основной язык

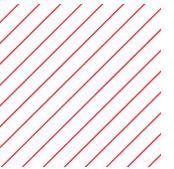
- Популярный язык для CV и DL
- Множество библиотек
- Легко читается



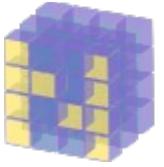
Jupyter: среда для прототипирования

- Документация и код
- Демонстрации
- Быстрая разработка





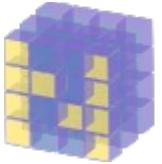
Инструменты



NumPy: линейная алгебра

- Матричные операции
- Работа с многомерными массивами (тензорами)

Инструменты



NumPy: линейная алгебра

- Матричные операции
- Работа с многомерными массивами (тензорами)

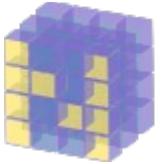


OpenCV: работа с изображениями

- Чтение и запись изображений
- Преобразования цвета
- Повороты, масштабирование, деформация, фильтры
- Алгоритмы



Инструменты



NumPy: линейная алгебра

- Матричные операции
- Работа с многомерными массивами (тензорами)

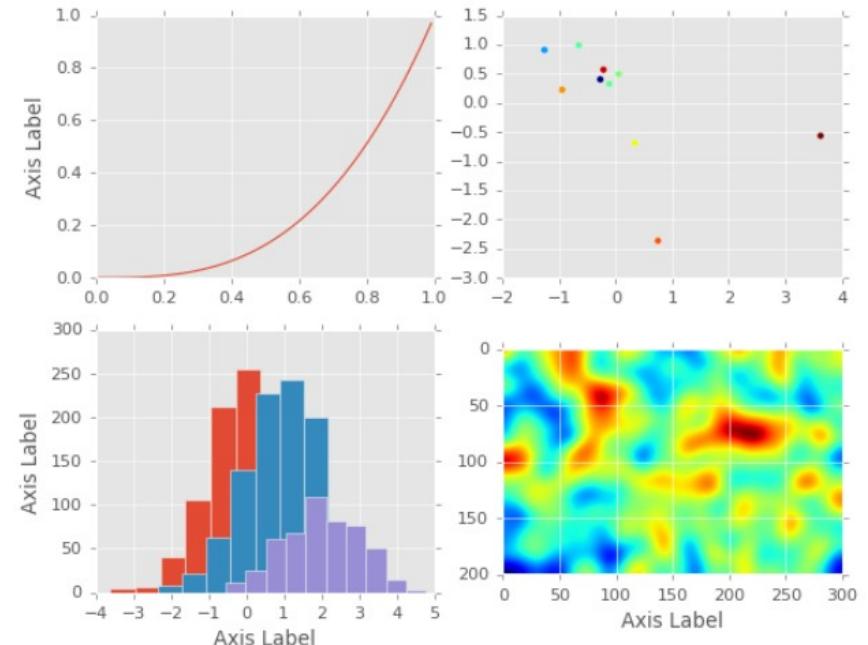


OpenCV: работа с изображениями

- Чтение и запись изображений
- Преобразования цвета
- Повороты, масштабирование, деформация, фильтры
- Алгоритмы



Matplotlib: визуализация

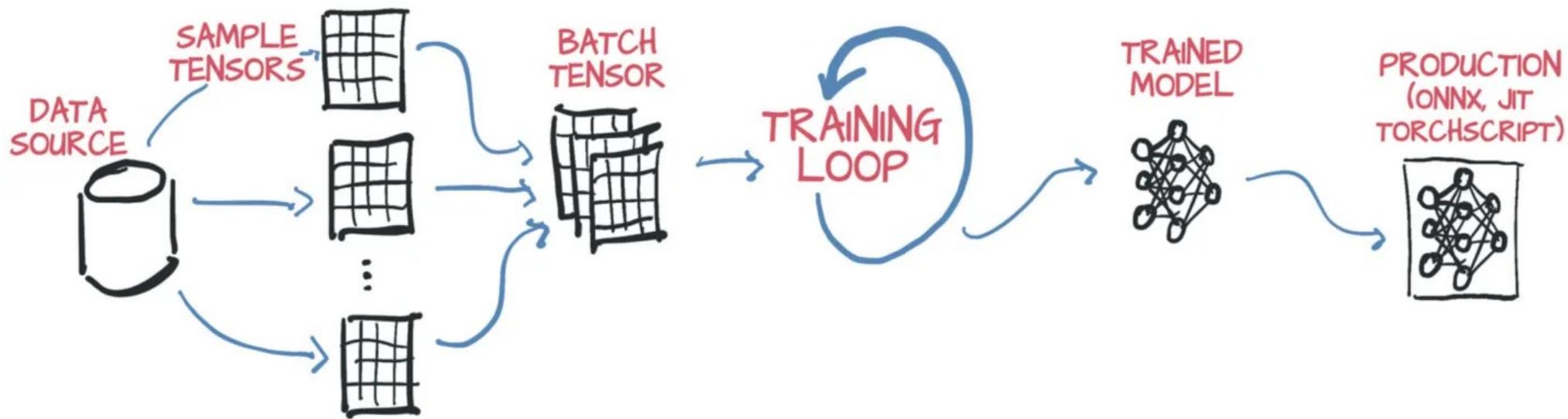


Инструменты



PyTorch: нейронные сети

- Популярный фреймворк для DL
- Множество реализованных сетей и блоков
- Оптимизация на GPU

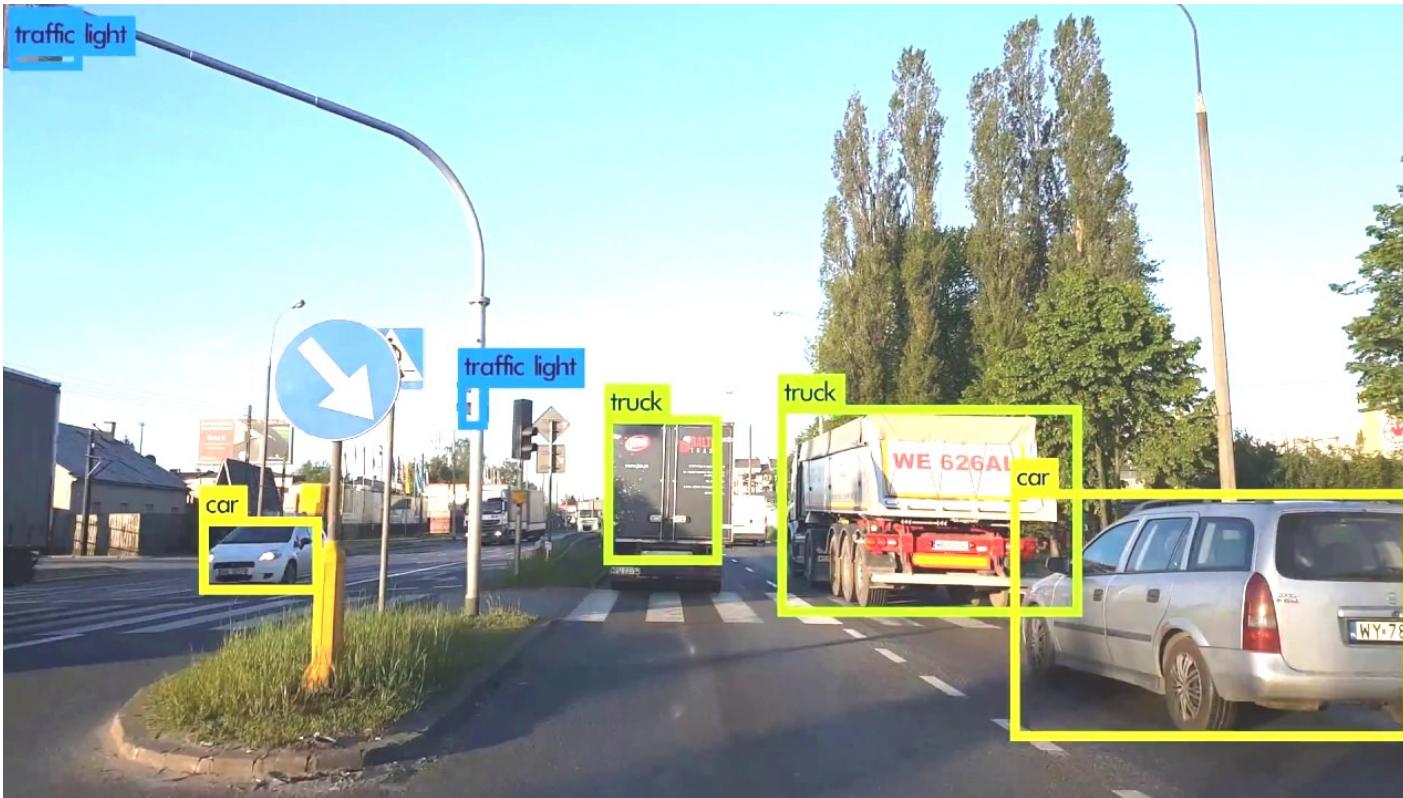


Күрс



О чем этот курс

Современные задачи CV: поиск и распознавание объектов



О чем этот курс

Современные задачи CV: сопоставление изображений



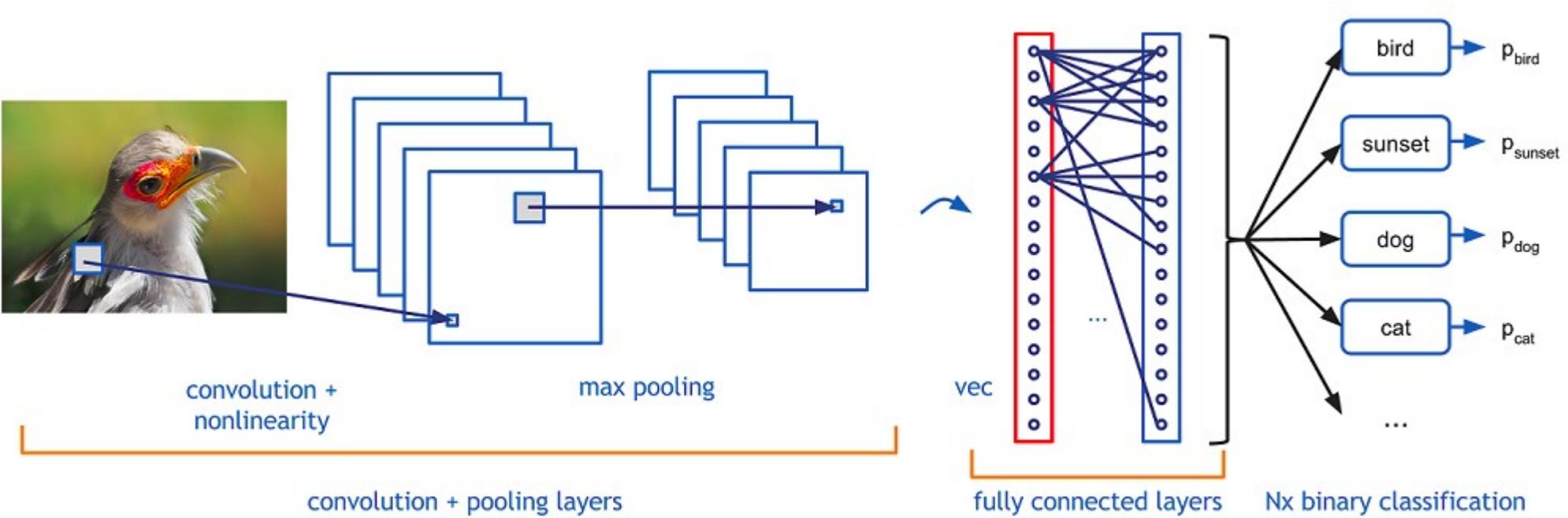
О чем этот курс

Современные задачи CV: генерация изображений



О чем этот курс

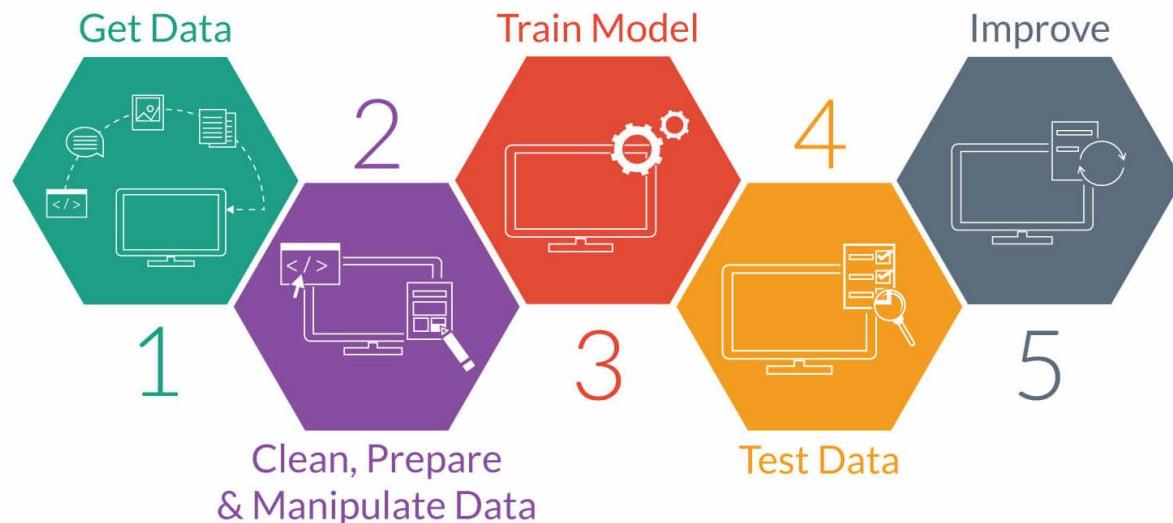
Решение CV задач при помощи нейронных сетей

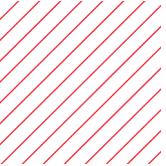


О чем этот курс

Хотим научить:

- Решать современные задачи CV методами глубокого обучения
- Устранять возникающие проблемы
- Понимать специфику использования нейронных сетей в production





Как устроен курс

Начало: 20 февраля

Конец: 11 июня (может измениться)

Каждый четверг: лекция и семинар

~17 лекций

3 конкурса

Кто ведет курс



Эдуард Тяントов

Руководитель направления машинного обучения в Почте и Портале



Даниил Лысухин

Программист-исследователь
в команде машинного зрения



Андрей Бояров

Ведущий программист-исследователь
в команде машинного зрения



Иван Карпухин

Ведущий программист-исследователь
в команде машинного зрения



Фёдор Киташов

Программист-исследователь
в команде машинного зрения

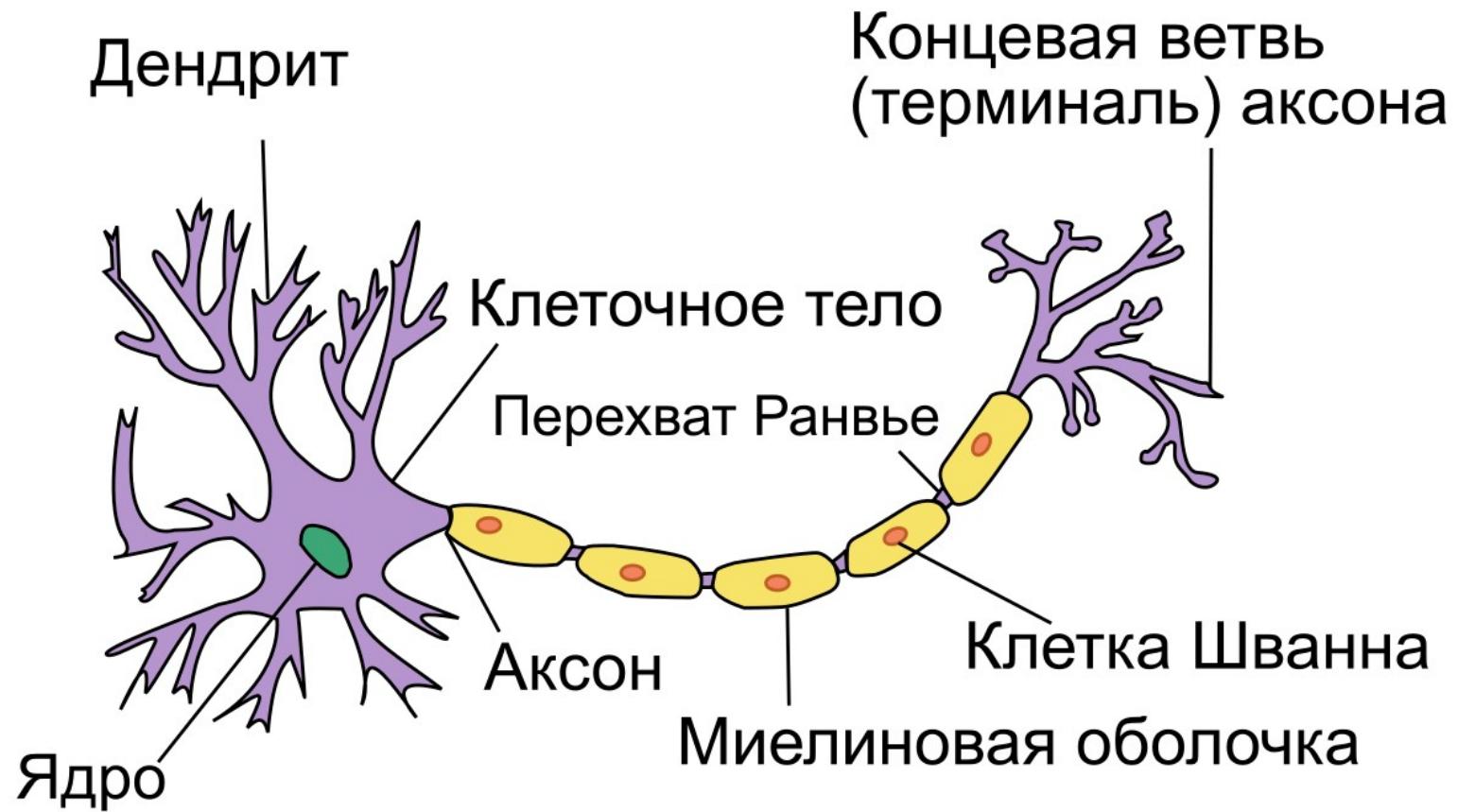


Борис Лесцов

Программист-исследователь
в команде машинного зрения

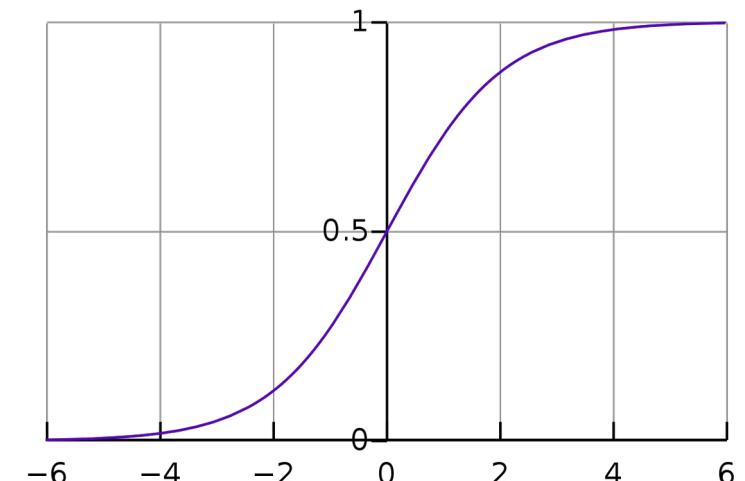
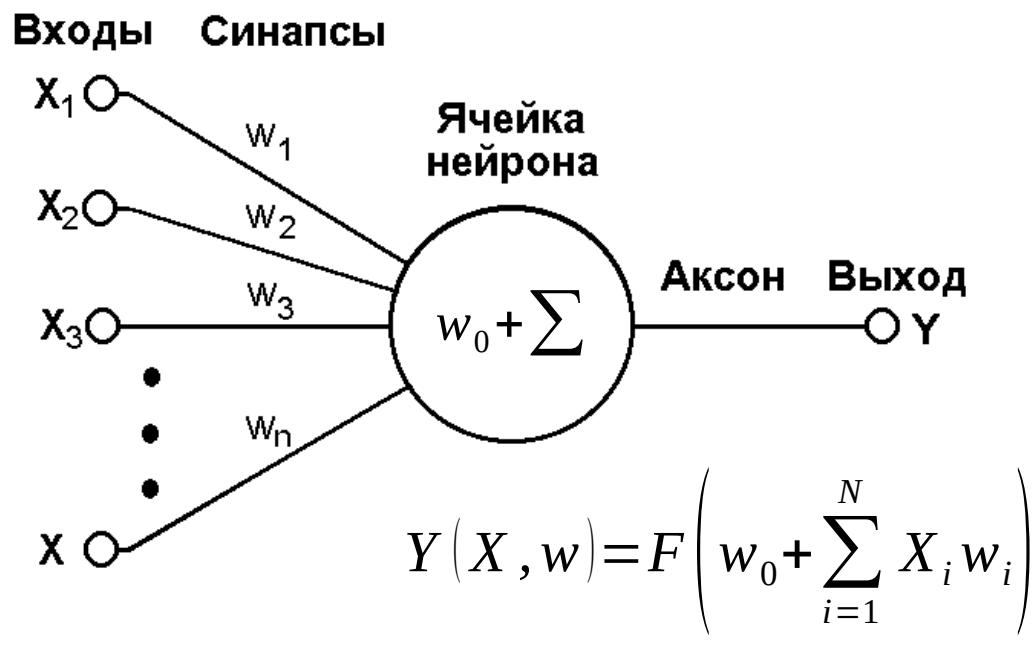
Нейрон

Нейрон в биологии



Модель нейрона

Если функция активации – сигмоида, то получается модель, похожая на логистическую регрессию.



$$F(S) = \frac{1}{1+e^{-S}}$$

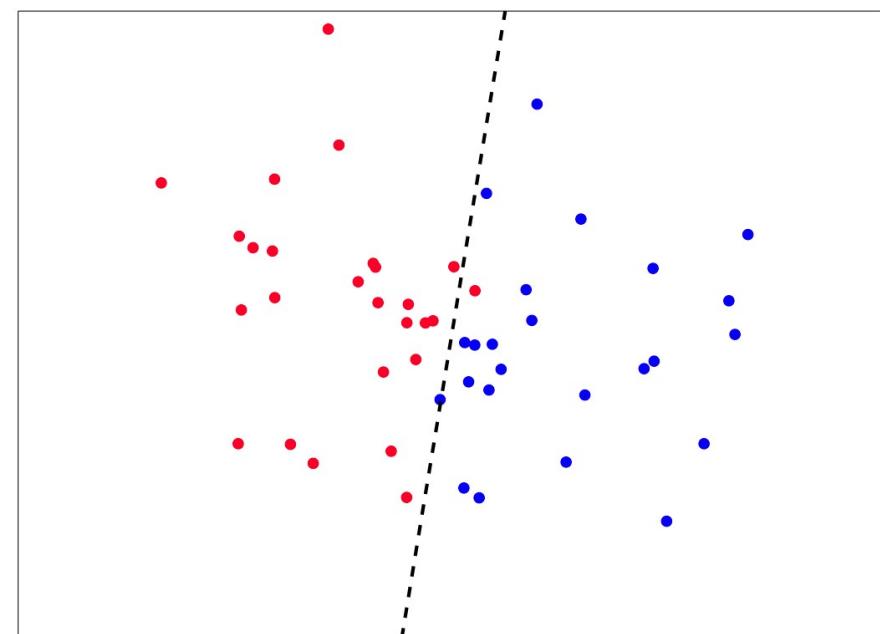
* Rosenblatt F. The perceptron, a perceiving and recognizing automaton Project Para. 1957.

Бинарная классификация

$$Y(X, w) = F\left(w_0 + \sum_{i=1}^N X_i w_i\right)$$

$$P(X, w) = \begin{cases} 0, & Y(X, w) \geq \text{threshold} \\ 1, & Y(X, w) < \text{threshold} \end{cases}$$

Если $F(S)$ монотонна, то разделяющая поверхность - гиперплоскость.



ФУНКЦИЯ ПОТЕРЬ

$$Y(X, w) = F\left(w_0 + \sum_{i=1}^N X_i w_i\right), \quad L(X) \in \{0, 1\}$$

$$P(X, w) = \begin{cases} 0, & Y(X, w) \geq \text{threshold} \\ 1, & Y(X, w) < \text{threshold} \end{cases}$$

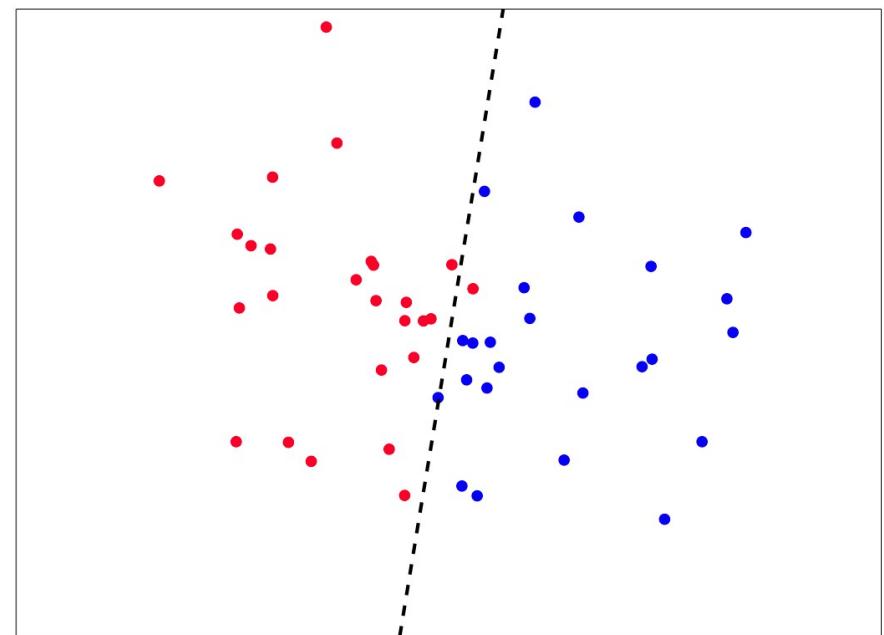
$$\text{Error}(w) = \sum_X \text{Error}(X, w)$$

- L2:

$$\text{Error}(X, w) = (Y(X, w) - L(X))^2$$

- Перекрестная энтропия (BCE):

$$\text{Error}(X, w) = \begin{cases} -\log Y(X, w), & L(X) = 0 \\ -\log(1 - Y(X, w)), & L(X) = 1 \end{cases}$$



Оптимизация

$$Y(X, w) = F\left(w_0 + \sum_{i=1}^N X_i w_i\right), \quad L(X) \in \{0, 1\}$$

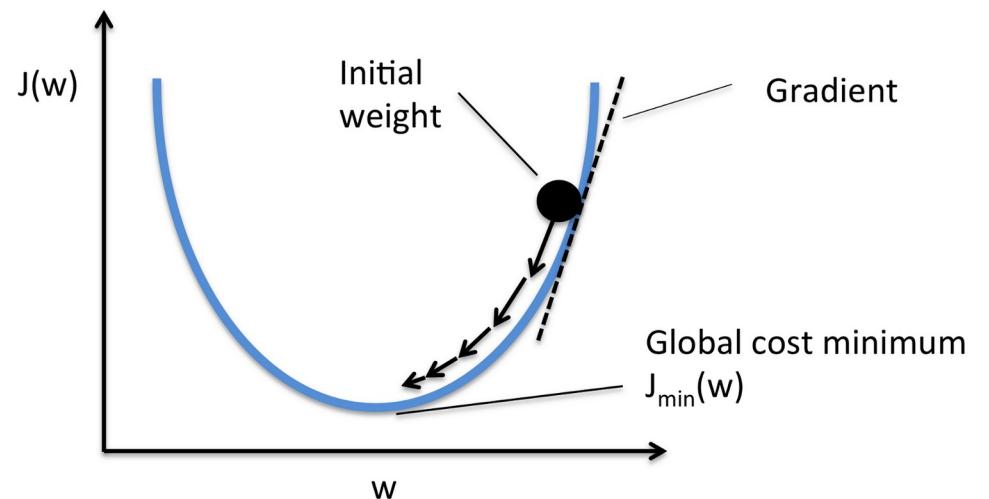
$$\text{Error}(w) = \sum_X \text{Error}(X, w)$$

$$\text{Error}(X, w) = \begin{cases} -\log Y(X, w), & L(X) = 0 \\ -\log(1 - Y(X, w)), & L(X) = 1 \end{cases}$$

$$w = \underset{w}{\operatorname{argmin}} \text{Error}(w)$$

Если F дифференцируема:

- Y и Error дифференцируемы
- можно использовать метод градиентного спуска для нахождения вектора весов w



Градиентный спуск

Одномерный случай:

$$f : \mathbb{R} \rightarrow \mathbb{R}$$

$$w = \underset{w}{\operatorname{argmin}} f(w)$$

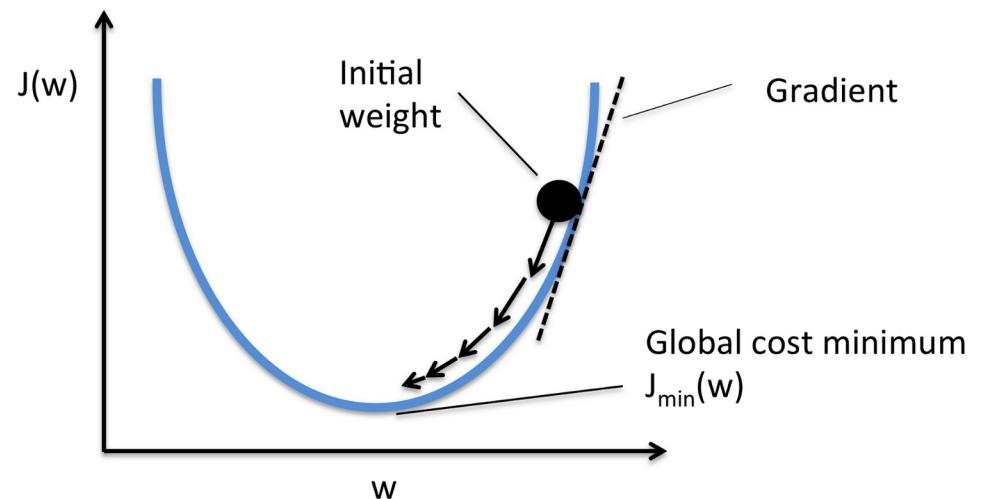
$$w_{i+1} = w_i - \lambda f'(w_i), \forall w_0$$

Многомерный случай:

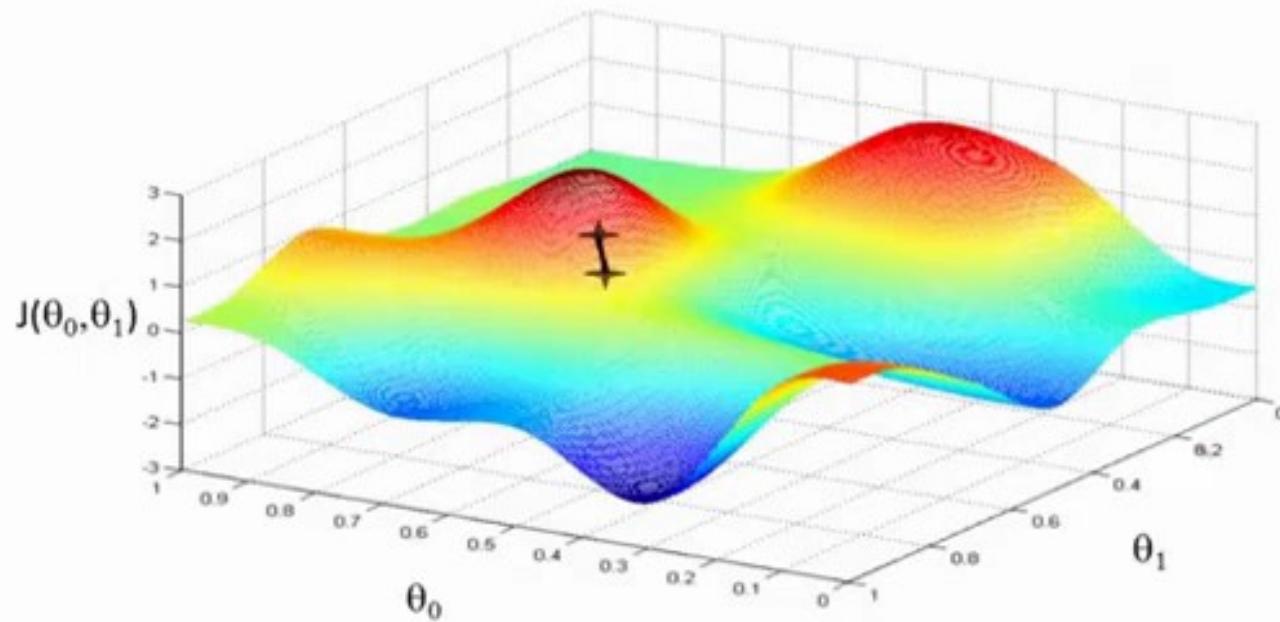
$$f : \mathbb{R}^N \rightarrow \mathbb{R}$$

$$W = \underset{W}{\operatorname{argmin}} f(W)$$

$$W_{i+1} = W_i - \lambda \nabla f(W_i), \forall W_0$$



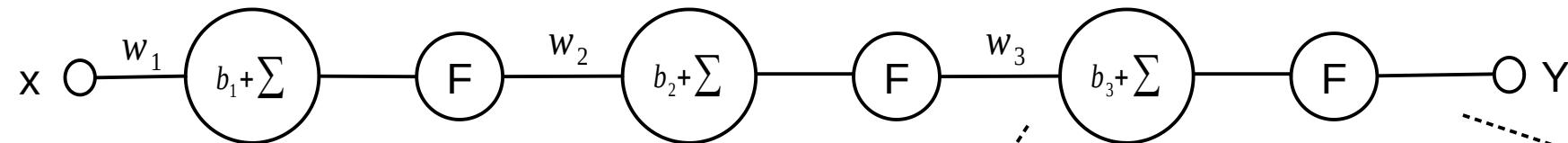
Градиентный спуск



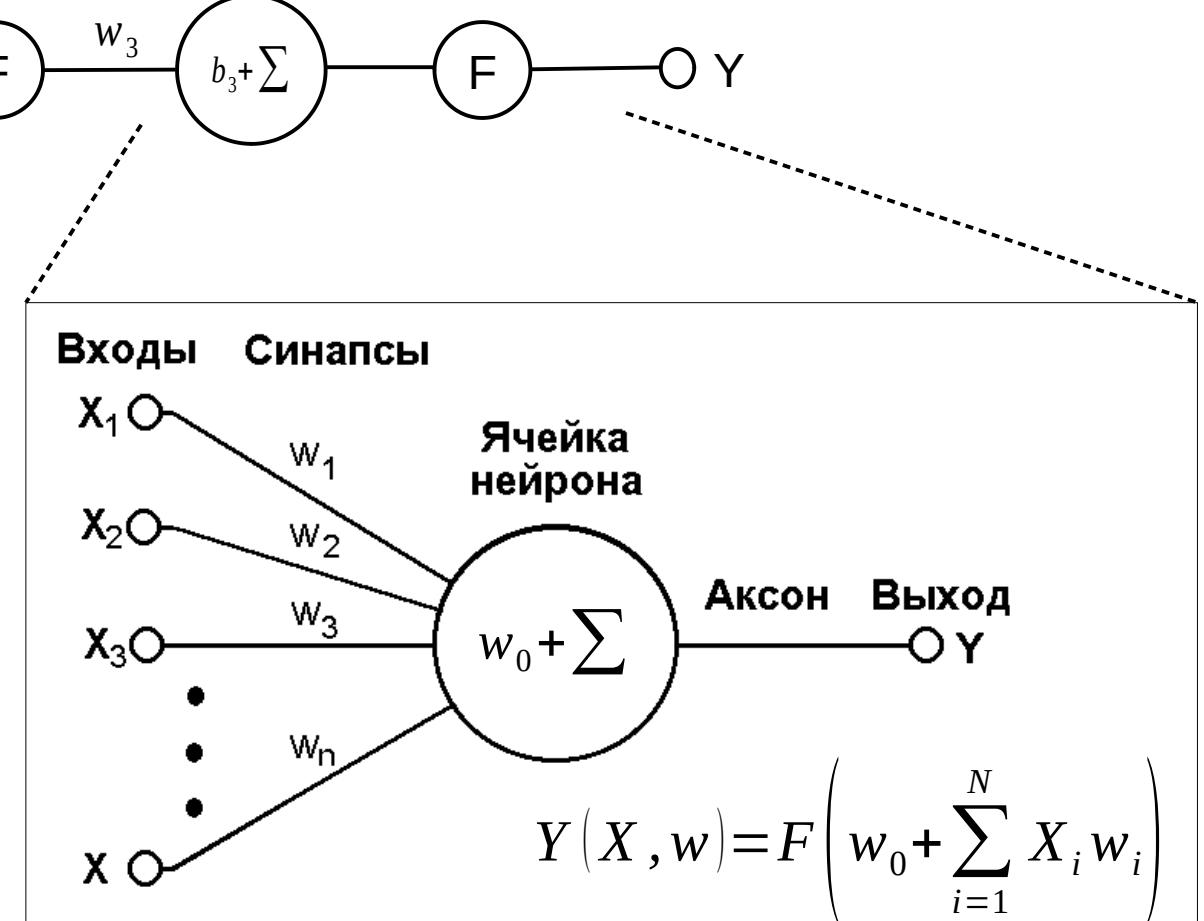
Andrew Ng

Цепочка нейронов

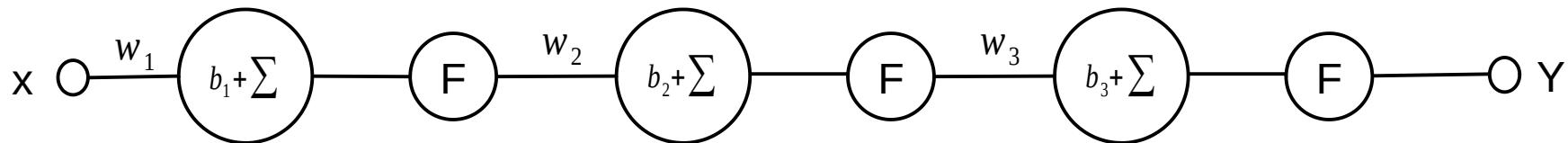
Модель



$$Y = Y_3 \left(Y_2 \left(Y_1(x, w_1, b_1), w_2, b_2 \right), w_3, b_3 \right)$$



Модель



$$Y = Y_3 \left(Y_2 \left(Y_1(x, w_1, b_1), w_2, b_2 \right), w_3, b_3 \right)$$

$$\begin{aligned} (f(w) + g(w))' &= f'(w) + g'(w) \\ f'(g(w)) &= f'(g(w))g'(w) \end{aligned}$$

$$Error'_{w_3} = \sum_x BCE'_Y(Y_3, L(x)) Y_3'_{w_3}(Y_2)$$

$$Error'_{w_2} = \sum_x BCE'_Y(Y_3, L(x)) Y_3'_{x}(Y_2) Y_2'_{w_2}(Y_1)$$

$$Error'_{w_1} = \sum_x BCE'_Y(Y_3, L(x)) Y_3'_{x}(Y_2) Y_2'_{x}(Y_1) Y_1'_{w_1}(x)$$

Оптимизация

$$Y = Y_3 \left(Y_2 \left(Y_1 \left(x, w_1, b_1 \right), w_2, b_2 \right), w_3, b_3 \right)$$

$$f'(g(x)) = f'(g(x))g'(x)$$

Прямой проход (forward pass):

$$f_{1x} = Y_1 \left(x, w_1, b_1 \right),$$

$$f_{2x} = Y_2 \left(f_{1x}, w_2, b_2 \right),$$

$$f_{3x} = Y_3 \left(f_{2x}, w_3, b_3 \right)$$

$$Error_x = BCE(f_{3x}, L(x))$$

$$Error = \sum_x Error_x$$

$$Error'_{w_3} = \sum_x BCE'_Y(Y_3, L(x)) Y_3'_{w_3}(Y_2)$$

$$Error'_{w_2} = \sum_x BCE'_Y(Y_3, L(x)) Y_3'_{x}(Y_2) Y_2'_{w_2}(Y_1)$$

$$Error'_{w_1} = \sum_x BCE'_Y(Y_3, L(x)) Y_3'_{x}(Y_2) Y_2'_{x}(Y_1) Y_1'_{w_1}(x)$$

Обратный проход (backward pass):

$$g_{BCE} = BCE'_Y(f_{3x}, L(x)),$$

$$g_3 = g_{BCE} Y_3'_{x}(f_{2x}),$$

$$g_2 = g_3 Y_2'_{x}(f_{1x}),$$

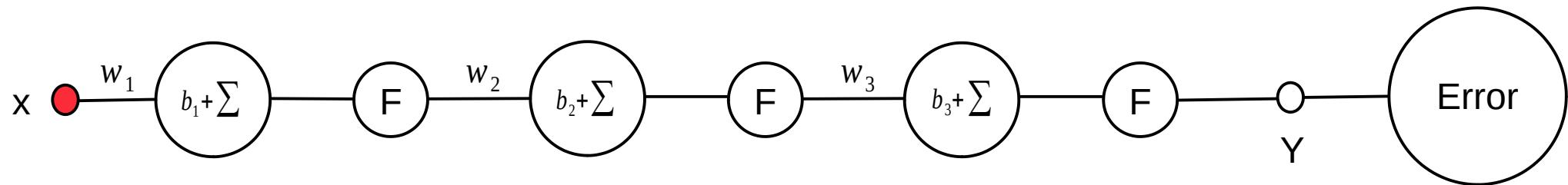
$$g_1 = g_2 Y_1'_{x}(x),$$

$$Error'_{x w_3} = g_{BCE} Y_3'_{w_3}(f_{2x})$$

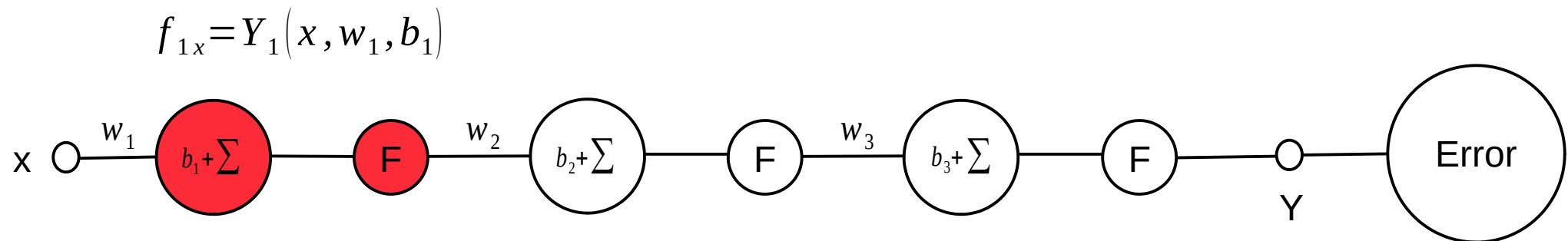
$$Error'_{x w_2} = g_3 Y_2'_{w_2}(f_{1x})$$

$$Error'_{x w_1} = g_2 Y_1'_{w_1}(x)$$

Прямой проход

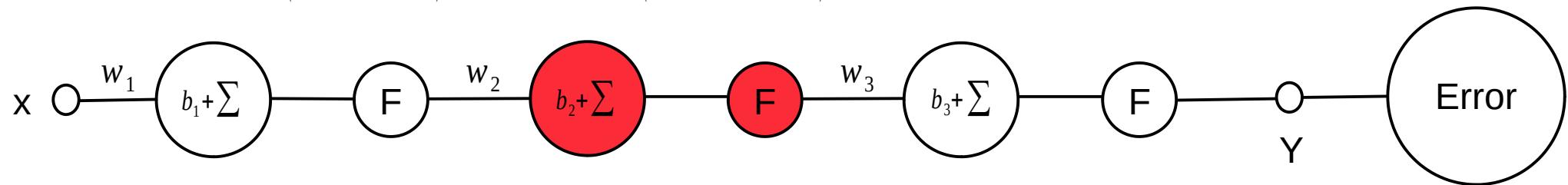


Прямой проход

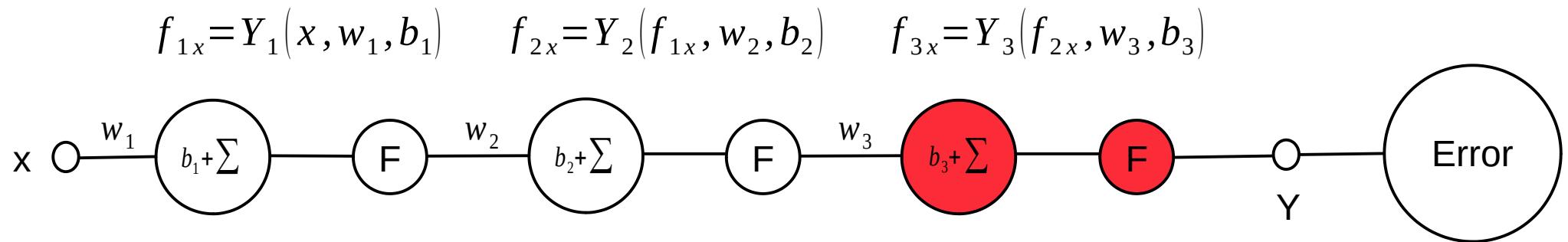


Прямой проход

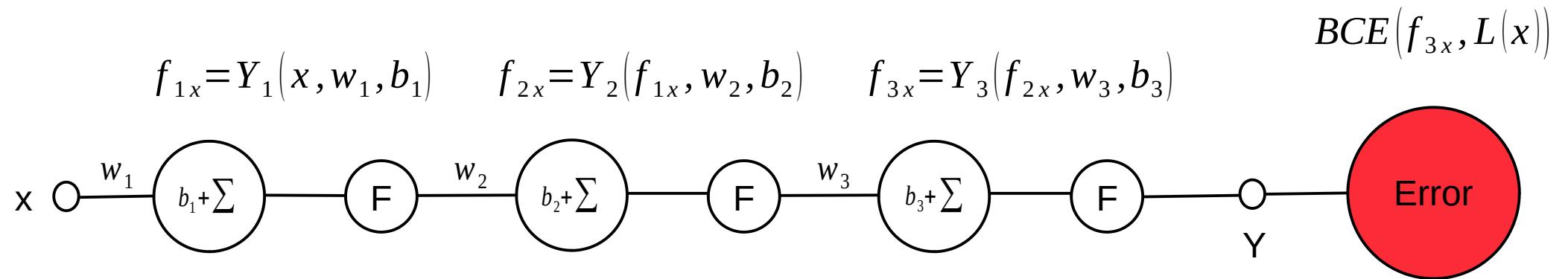
$$f_{1x} = Y_1(x, w_1, b_1) \quad f_{2x} = Y_2(f_{1x}, w_2, b_2)$$



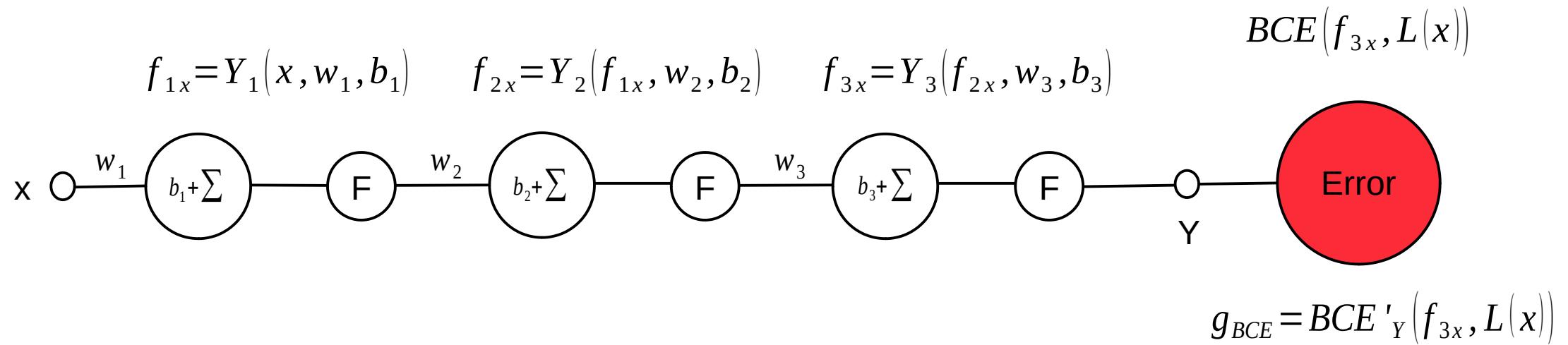
Прямой проход



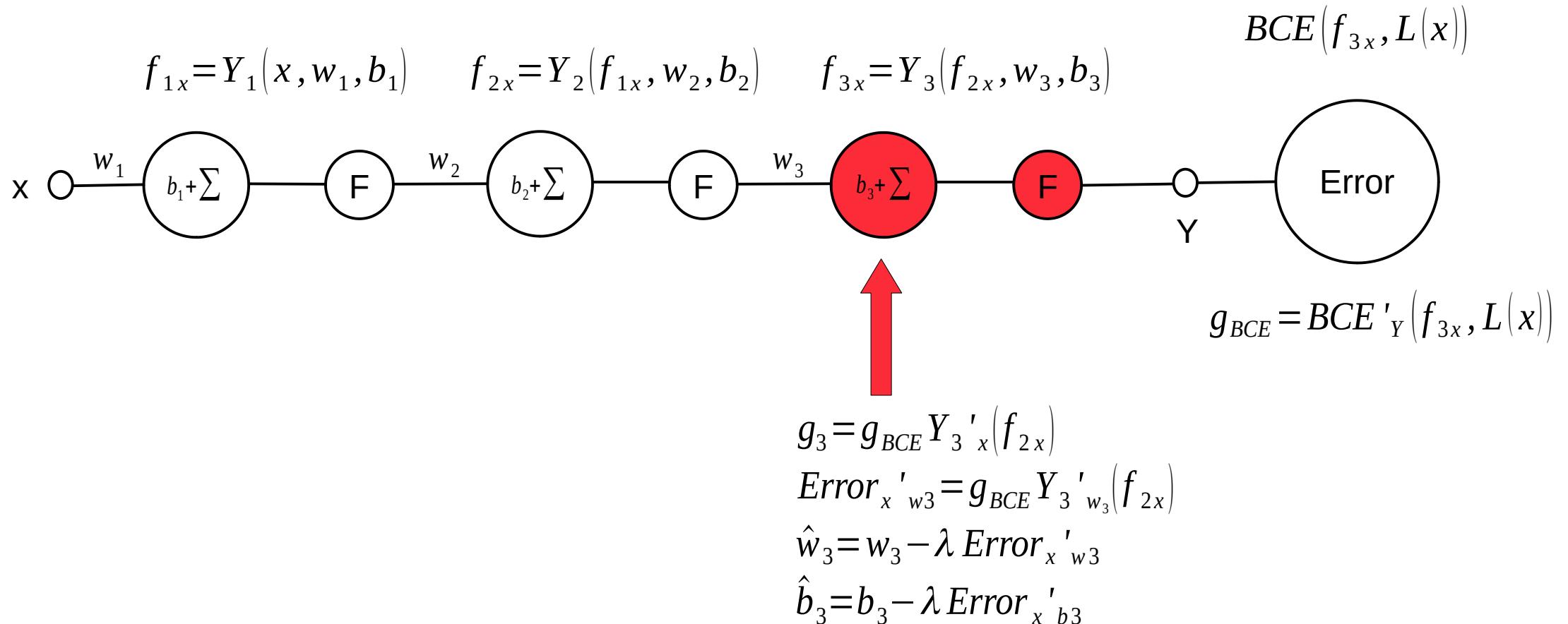
Прямой проход



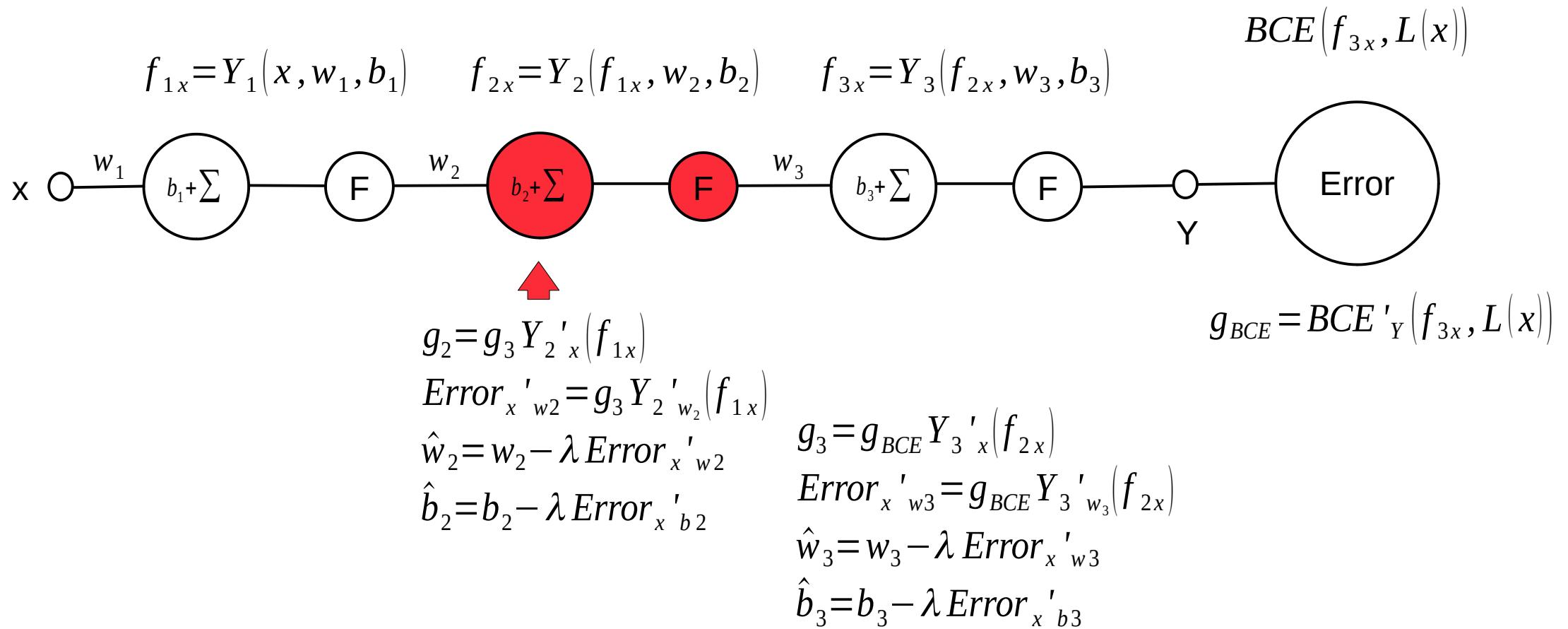
Обратный проход



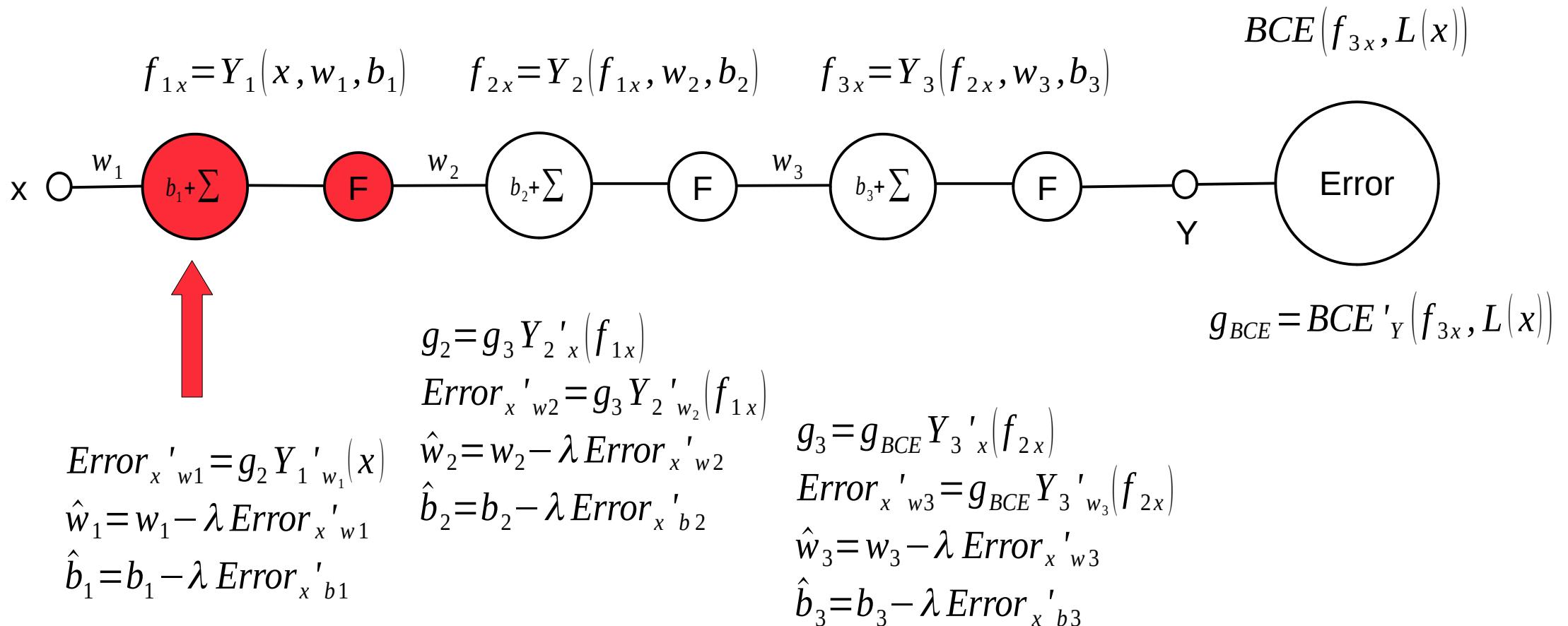
Обратный проход



Обратный проход



Обратный проход



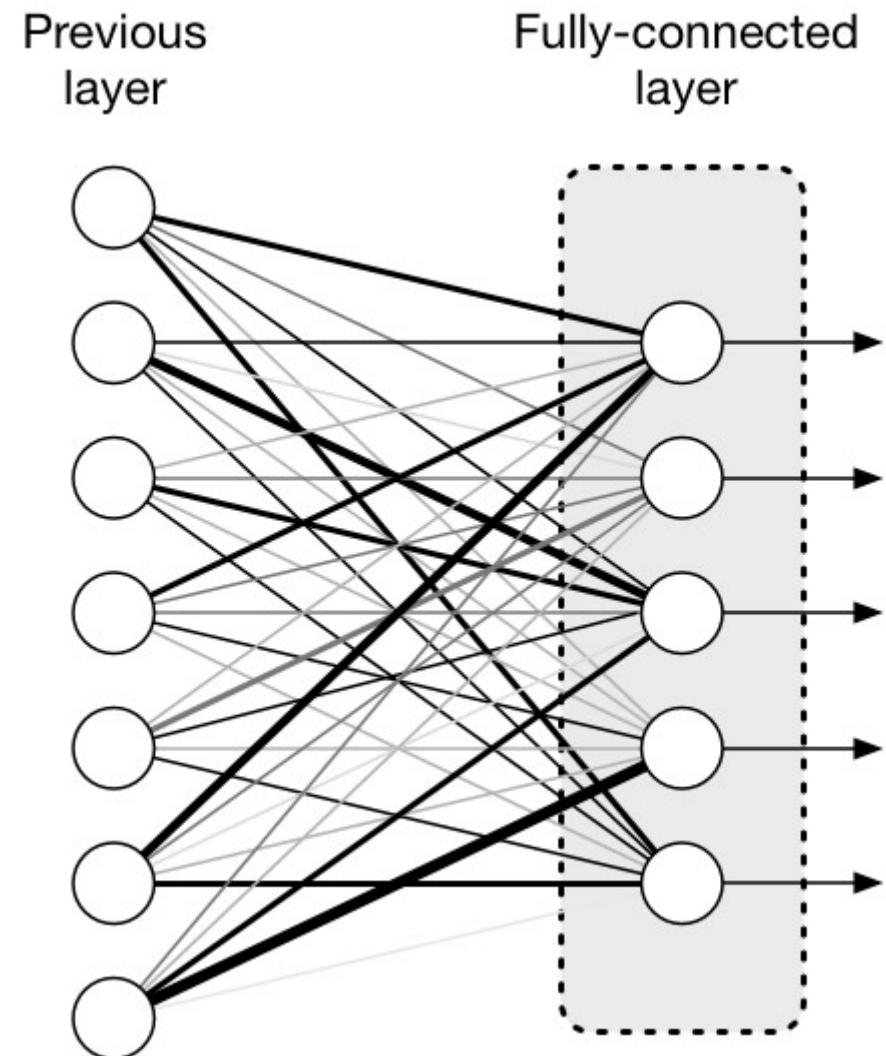
Полносвязные сети

Полносвязный слой

Полносвязный слой (Fully Connected layer, FC) объединяет несколько нейронов.

$$Y(X) = F(MX + B),$$

$$X \in \mathbb{R}^N, M \in \mathbb{R}^{N_{out} \times N_{in}}, B \in \mathbb{R}^{N_{out}}$$



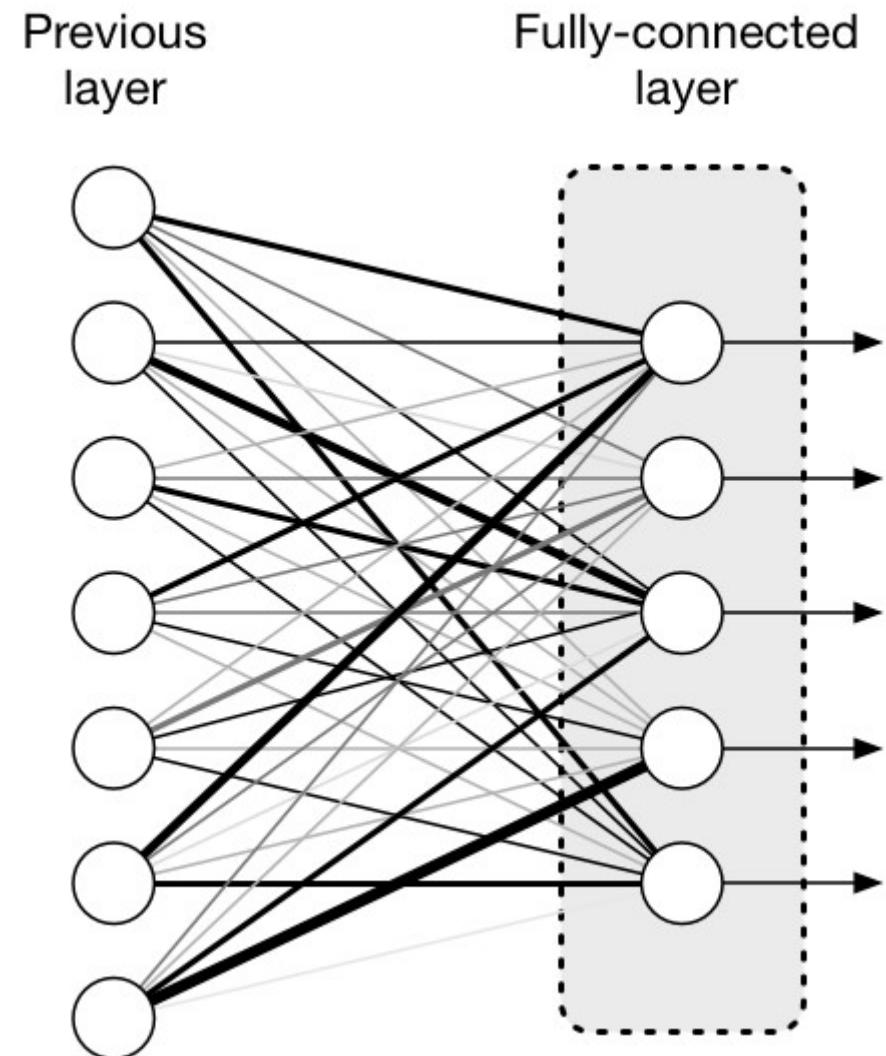
Полносвязный слой

Полносвязный слой (Fully Connected layer, FC) объединяет несколько нейронов.

$$Y(X) = F(MX + B),$$

$$X \in \mathbb{R}^N, M \in \mathbb{R}^{N_{out} \times N_{in}}, B \in \mathbb{R}^{N_{out}}$$

- Нейроны независимы



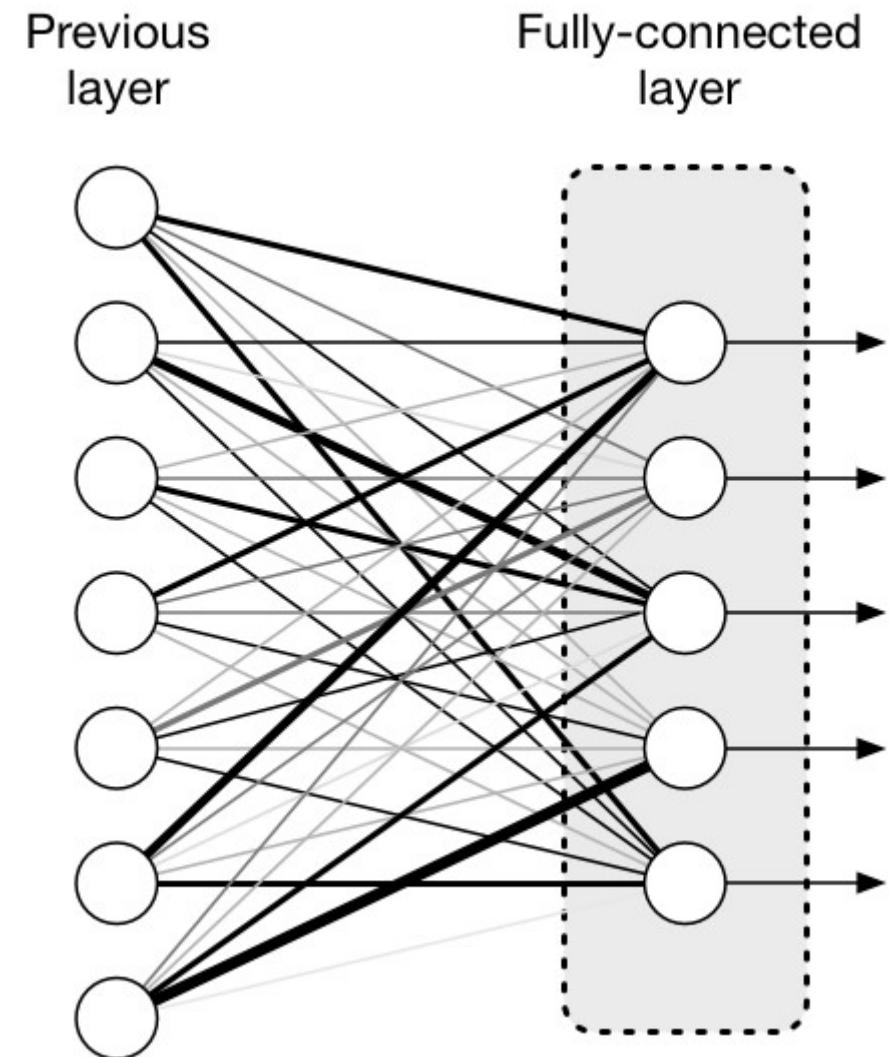
Полносвязный слой

Полносвязный слой (Fully Connected layer, FC) объединяет несколько нейронов.

$$Y(X) = F(MX + B),$$

$$X \in \mathbb{R}^N, M \in \mathbb{R}^{N_{out} \times N_{in}}, B \in \mathbb{R}^{N_{out}}$$

- Нейроны независимы
- Веса формируют матрицу размера (N_{out} , N_{in}) и вектор размера N_{out}



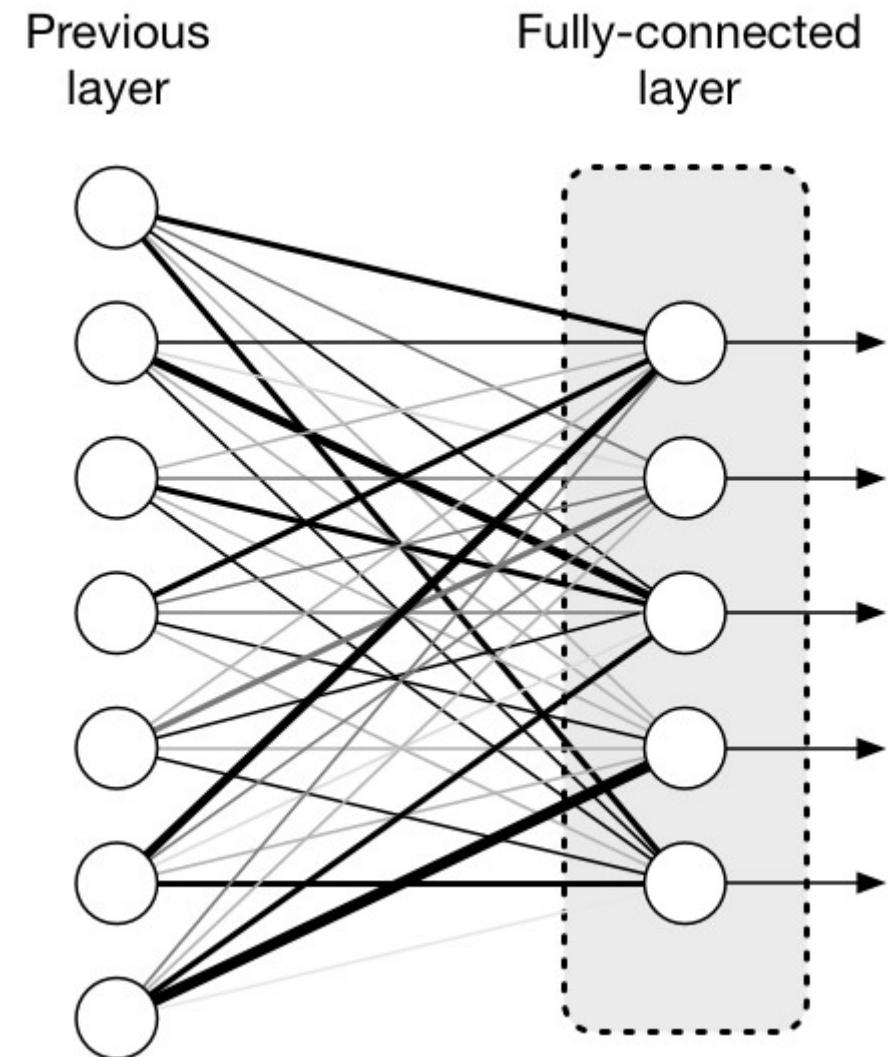
Полносвязный слой

Полносвязный слой (Fully Connected layer, FC) объединяет несколько нейронов.

$$Y(X) = F(MX + B),$$

$$X \in \mathbb{R}^N, M \in \mathbb{R}^{N_{out} \times N_{in}}, B \in \mathbb{R}^{N_{out}}$$

- Нейроны независимы
- Веса формируют матрицу размера (N_{out} , N_{in}) и вектор размера N_{out}
- Активационная функция применяется отдельно к каждому выходу слоя



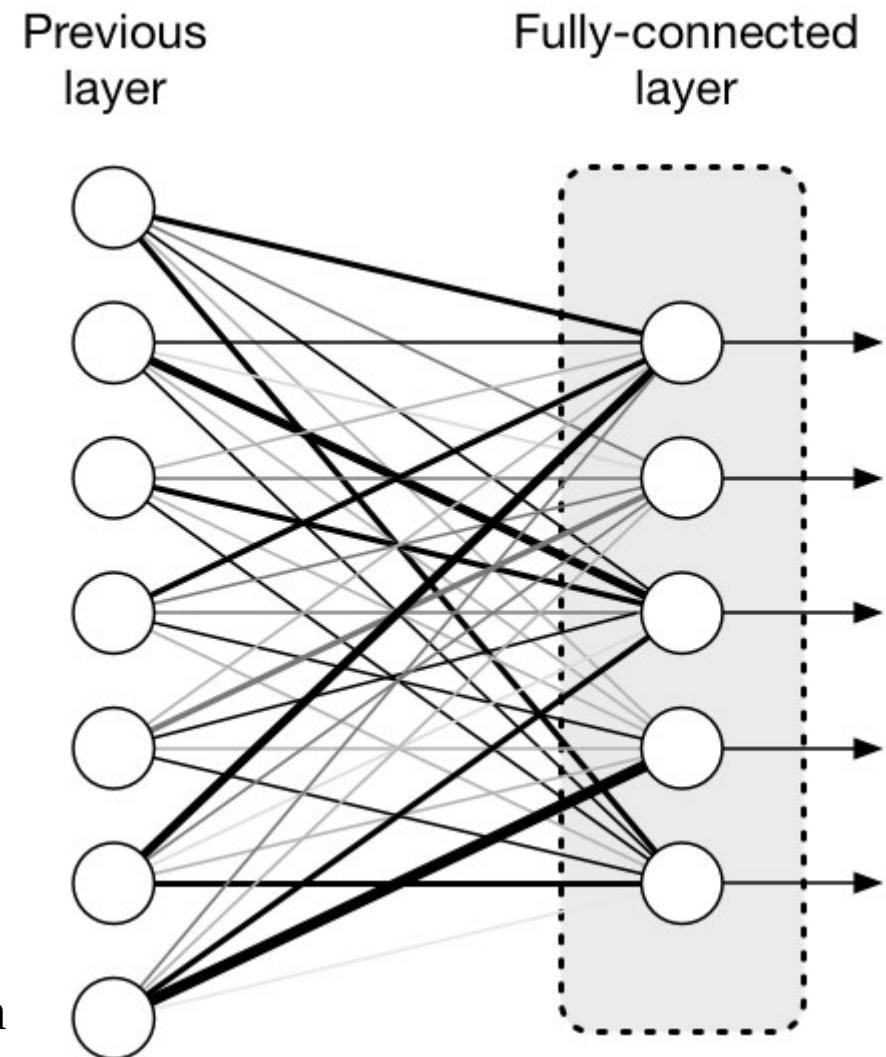
Полносвязный слой

Полносвязный слой (Fully Connected layer, FC) объединяет несколько нейронов.

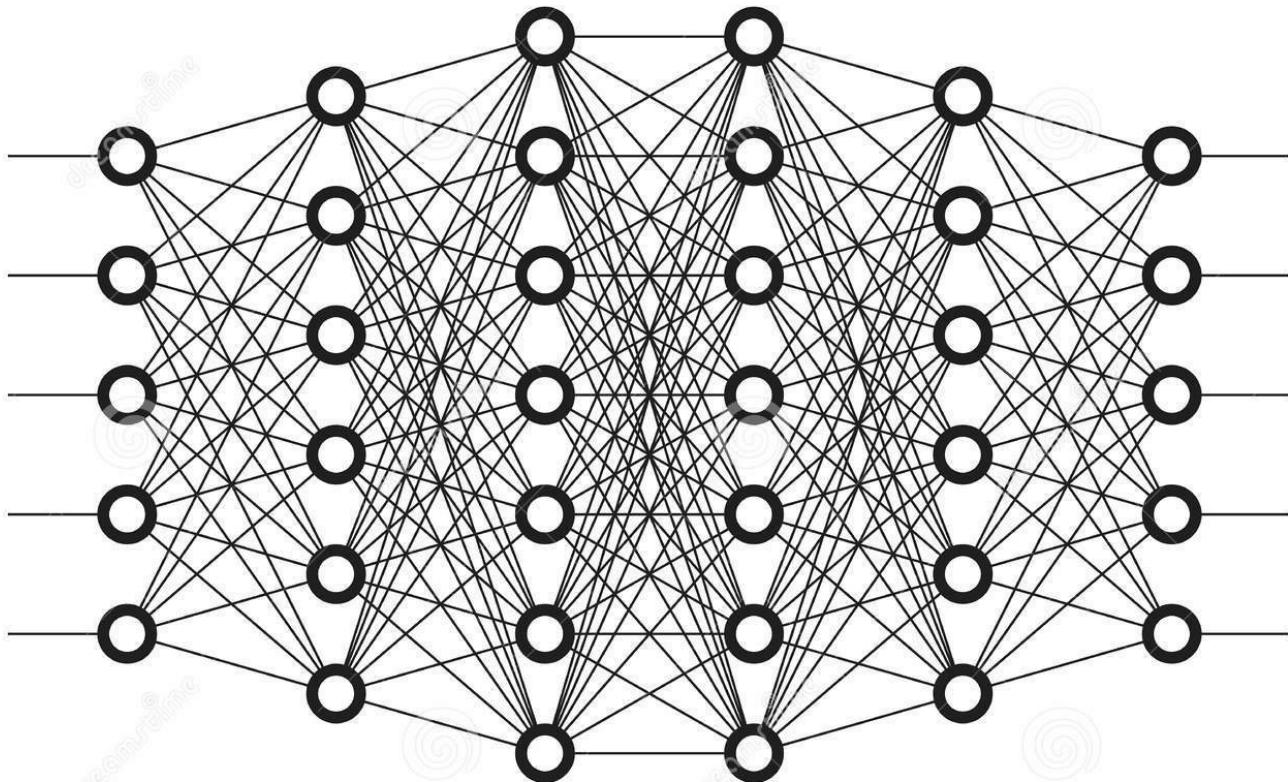
$$Y(X) = F(MX + B),$$

$$X \in \mathbb{R}^N, M \in \mathbb{R}^{N_{out} \times N_{in}}, B \in \mathbb{R}^{N_{out}}$$

- Нейроны независимы
- Веса формируют матрицу размера (N_{out} , N_{in}) и вектор размера N_{out}
- Активационная функция применяется отдельно к каждому выходу слоя
- При обратном проходе вычисляются векторы, а не числа



Полносвязная сеть



Задача классификация

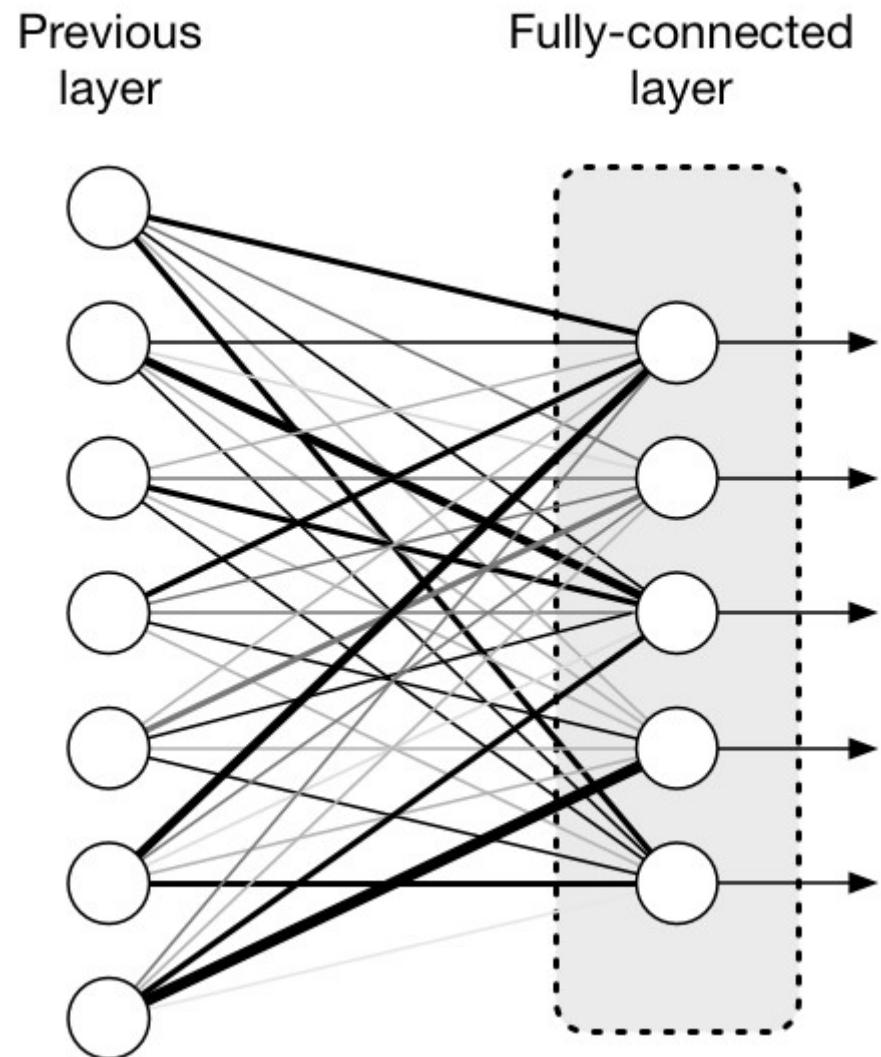
- Y - ненормированные логарифмы вероятностей классов (logits)
- Чтобы получить вероятности, добавим нормировочный слой SoftMax:

$$P_i = \text{SoftMax}(Y)_i = \frac{e^{Y_i}}{\sum_i e^{Y_i}}$$

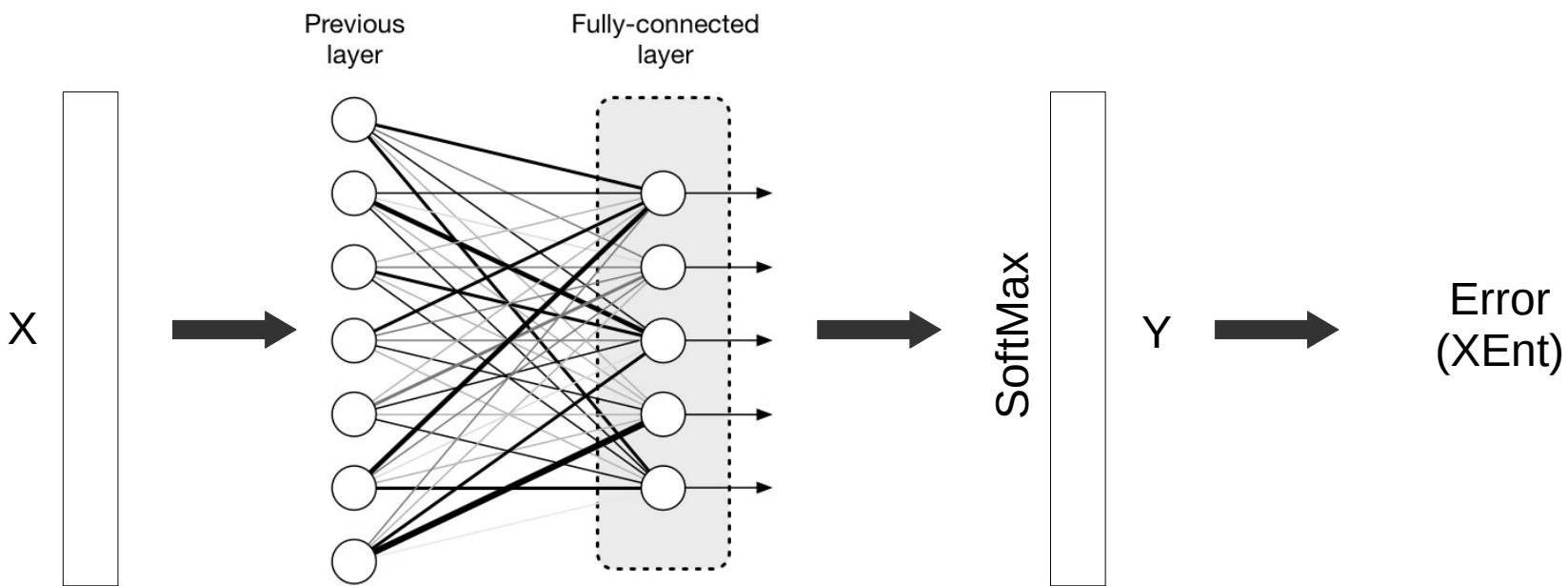
- Минимизируем перекрестную энтропию:

$$XEnt(Y, L(X)) = -\log P_{L(X)}$$

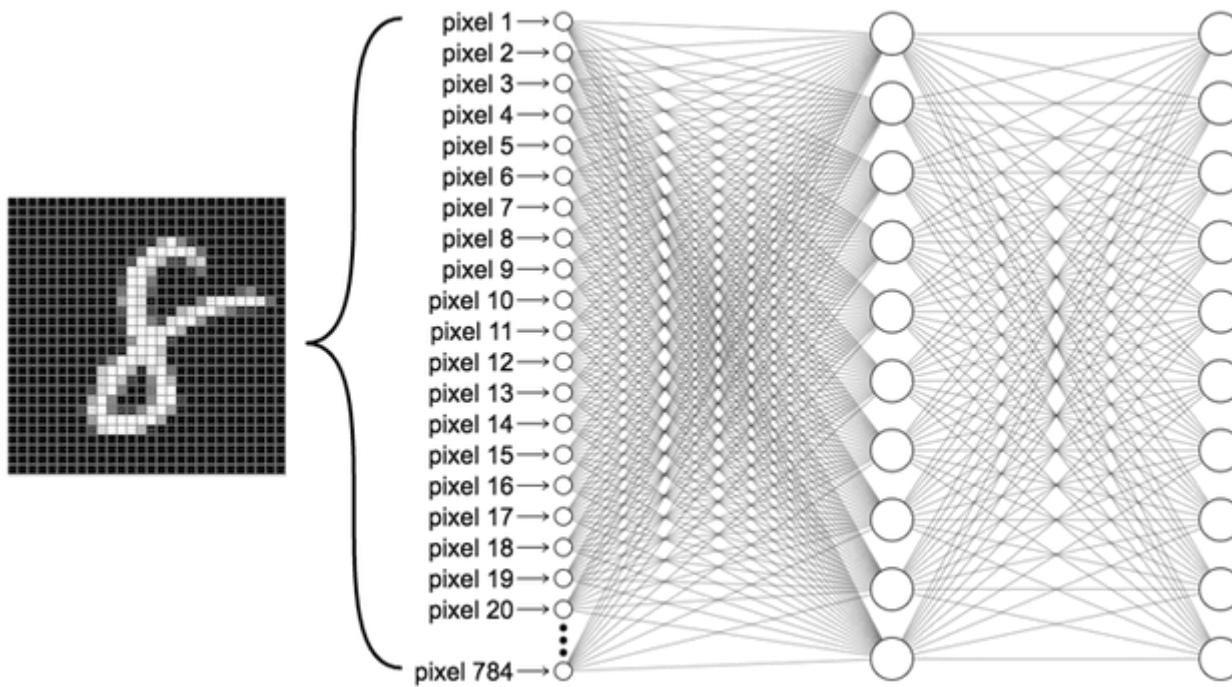
$$L(X) \in \overline{1, C}$$



Задача классификация

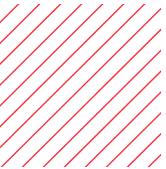


Полносвязная сеть



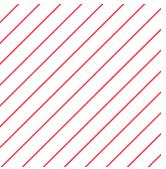
Pred	Label
0.01	0
0.01	1
0.01	2
0.22	3
0.01	4
0.01	5
0.01	6
0.01	7
0.70	8
0.01	9

Резюме



Резюме

- Computer Vision около 70 лет
- Переломный момент - появление DL в 201x
- Курс в первую очередь про DL



Резюме

- Computer Vision около 70 лет
 - Переломный момент - появление DL в 201x
 - Курс в первую очередь про DL
-
- Сеть состоит из множества дифференцируемых (почти всюду) слоев
 - Обучение сети – минимизация функции потерь методом градиентного спуска
 - Функция потерь для классификации – перекрестная энтропия
 - Обучение эффективно реализуется через серию прямых и обратных проходов

В следующий раз...

- Пайплайн машинного обучения
- Современные методы оптимизации, основанные на градиентном спуске
- Выбор функции активации
- Сравнение популярных функций потерь



Online курсы по ML

<https://www.coursera.org/learn/machine-learning>

<https://yandexdataschool.ru/edu-process/courses/machine-learning>

<https://ru.coursera.org/specializations/machine-learning-data-analysis>