

Введение **в нейронные сети**

Тамерлан Таболов, Антон Семёнкин
6 ноября 2017

Оглавление

- Предыстория
- Области применения искусственных нейронных сетей
- Принцип работы ИНС
- Пример работы ИНС
- Математическая модель ИНС
- Обучение нейронных сетей
- Backpropagation
- Виды нейронных сетей
- Свёрточные нейронные сети

Предыстория

- 1943 год — Мак-Каллок и Питтс. Первая компьютерная модель нейронной сети на основе математических алгоритмов и теории деятельности головного мозга.
- 1949 год — канадский физиолог и психолог Хебб высказывает идеи о характере соединения нейронов мозга и их взаимодействии. Он первым предположил, что обучение заключается в первую очередь в изменениях силы синаптических связей.
- 1954 год — В MIT Фарли и Кларк разработали имитацию сети Хебба.

Предыстория

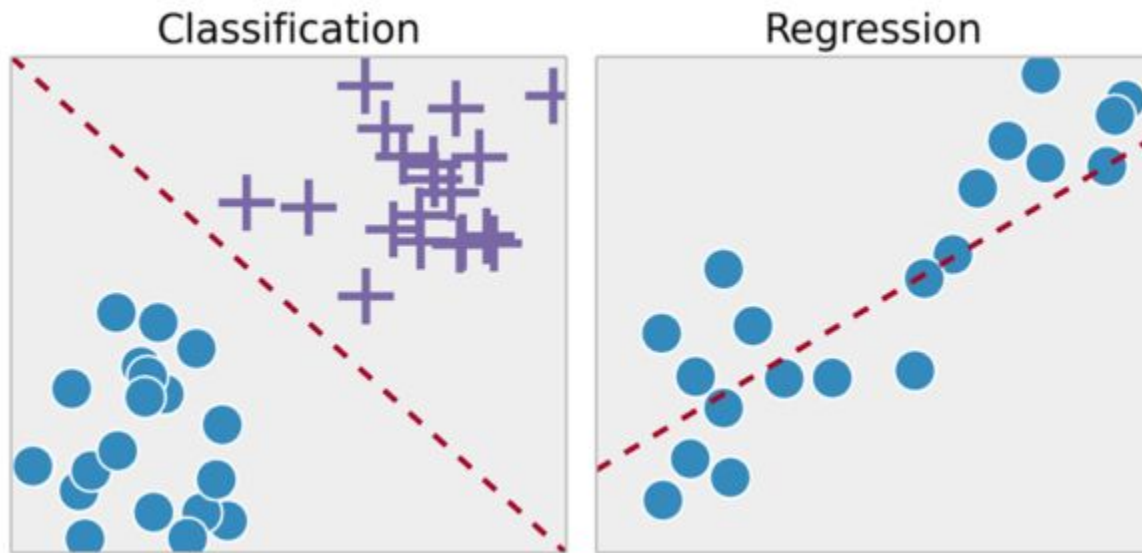
- 1969 год — публикация Минского и Пейперта:
 - Однослойные нейронные сети не способны вычислять XOR.
 - Компьютерам не хватает мощностей для обработки огромного объёма вычислений, необходимого для больших нейронных сетей.

Интерес к нейронным сетям угасает

- 1975 год — Вербос: метод обратного распространения ошибки.
Эффективное решение задачи обучения нейронных сетей.
- Многочисленные споры о том, может ли такое обучение быть на самом деле реализовано в головном мозге.

Области применения ИНН

- Классические задачи классификации и восстановления регрессии.



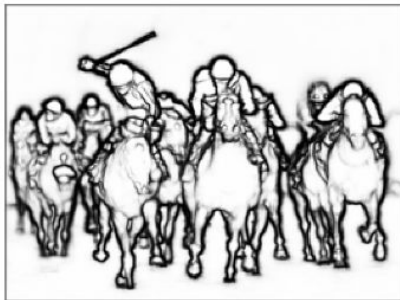
Области применения ИИИ

- Задачи, связанные с обработкой изображений. (CNN)

Input



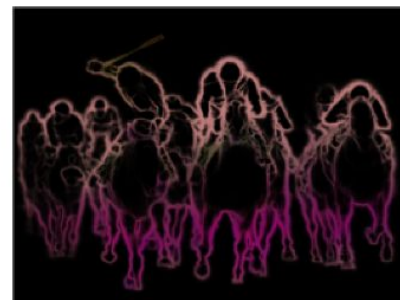
Boundaries



Semantic Segmentation



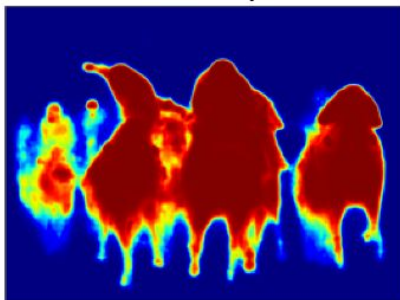
Semantic Boundaries



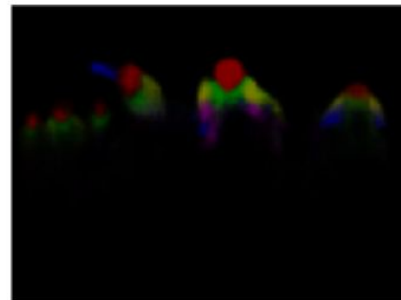
Surface Normals



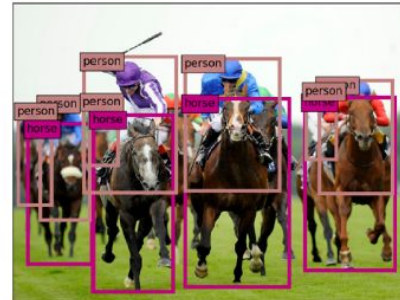
Saliency



Human Parts



Detection

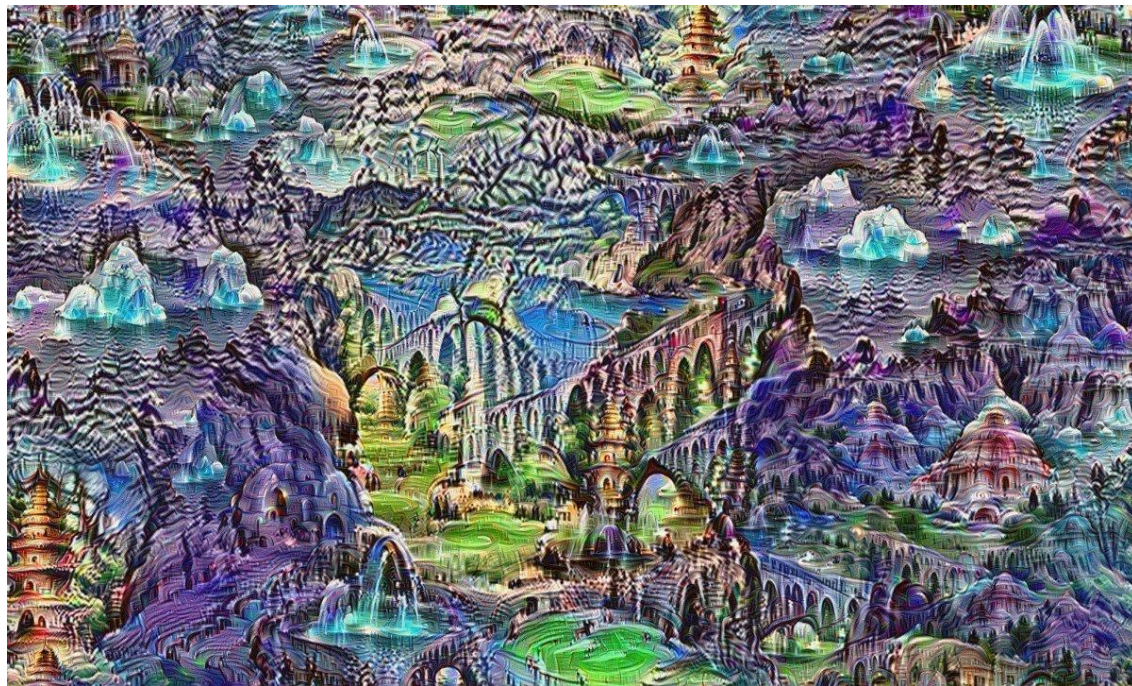


Области применения ИНН

- Задачи обработки последовательностей (RNN):
 - Обработка текстов на естественном языке
 - Обработка видео
 - Обработка аудио
 - Анализ временных рядов
 - Другие последовательности

Области применения ИИИ

- Задачи генерации (GAN):



Принцип работы ИНН

- Нейроны
 - Входные
 - Промежуточные (Скрытые)
 - Выходные
- Связи

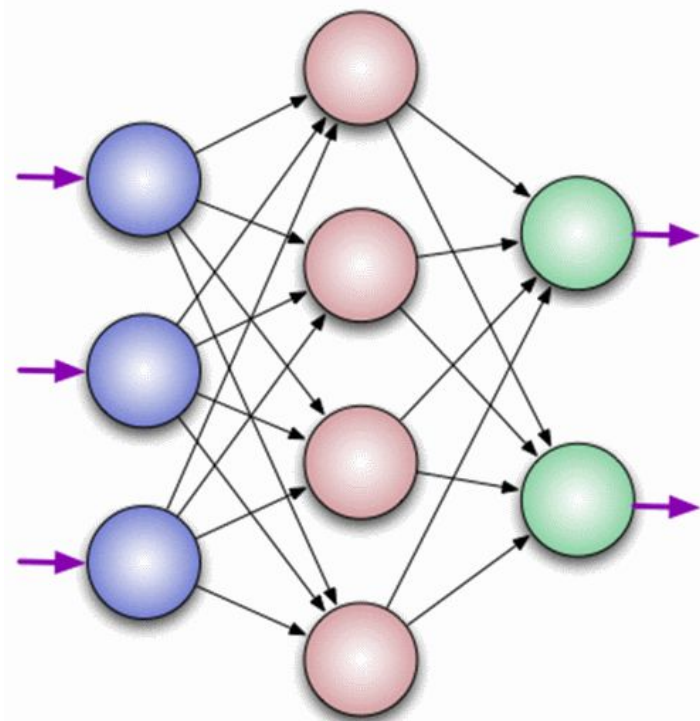
-Нейроны имеют несколько входов и ровно один выход.

-Нейроны образуют слои.

-Нейроны соседних слоёв соединены связями.

-Каждая связь имеет вес.

-Последний слой состоит из выходных нейронов и представляет “ответ” нейронной сети



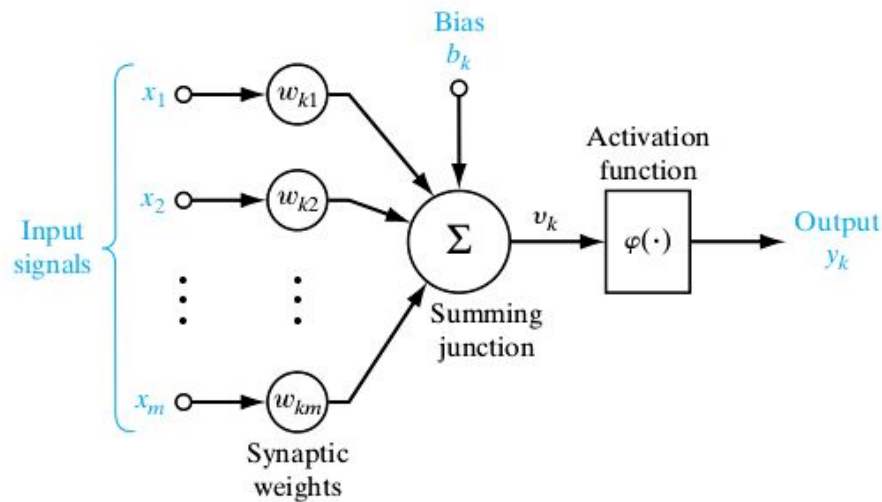
Принцип работы ИНН

-На вход подаётся информация об объекте.

-Информация распространяется в направлении от входа к выходу.

-Каждый нейрон суммирует значения, подаваемые на вход с предыдущих нейронов с весами, задаваемыми связями, добавляет смещение (bias) и применяет некую “функцию активации”.

-После прохождения информации через все слои, на выход подаётся “ответ” нейронной сети.

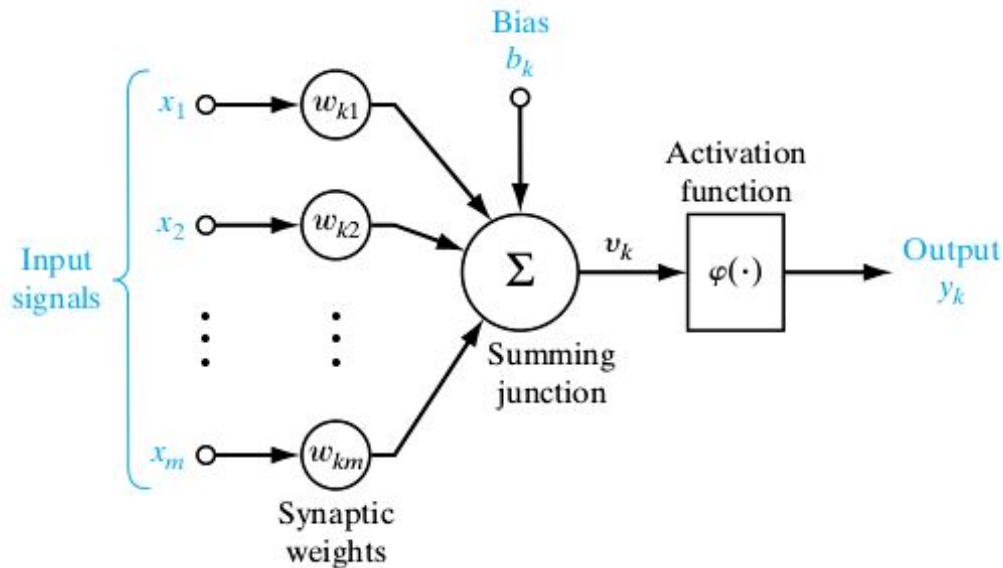


Принцип работы ИНН

Функция активации (функция сжатия):

Функция, применяемая нейроном к взвешенной сумме значений нейронов предыдущего слоя и “сжимающая” её в некоторый диапазон.

Добавляет нелинейность в работу нейронной сети

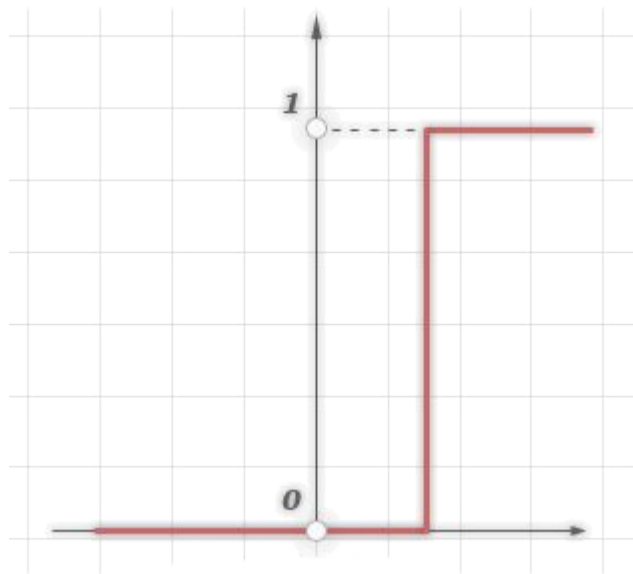


Принцип работы ИНН

Функция активации (функция сжатия)

Пороговая функция или функция единичного скачка

- Самое простое, что можно придумать
- Никак не учитывает величину отклонения от порога
- Почти не применяется на практике

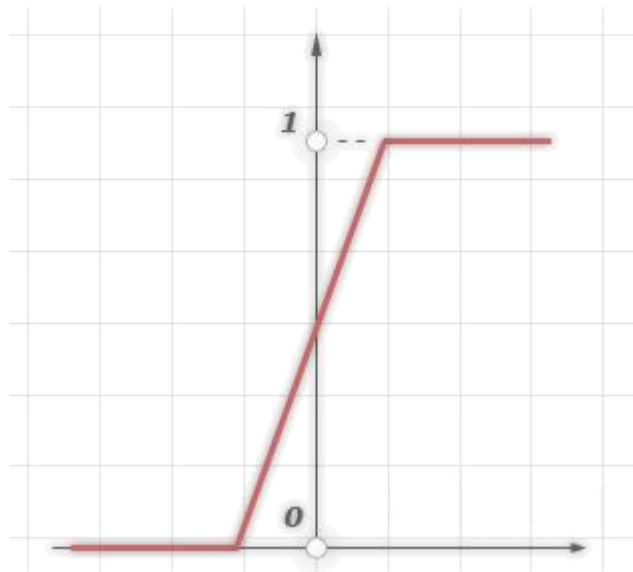


Принцип работы ИНН

Функция активации (функция сжатия)

Линейный порог или гистерезис

- Вариант чуть лучше
- Лучше учитывает значение входа



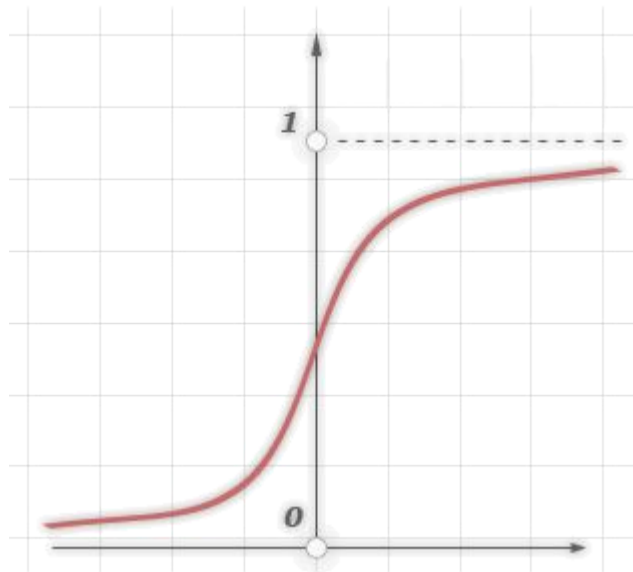
Принцип работы ИНН

Функция активации (функция сжатия)

Сигмоида

- Усиливает слабые сигналы и уменьшает сильные
- Простая, гладкая производная
- α — параметр наклона

$$OUT = \frac{1}{1 + \exp(-\alpha Y)}$$



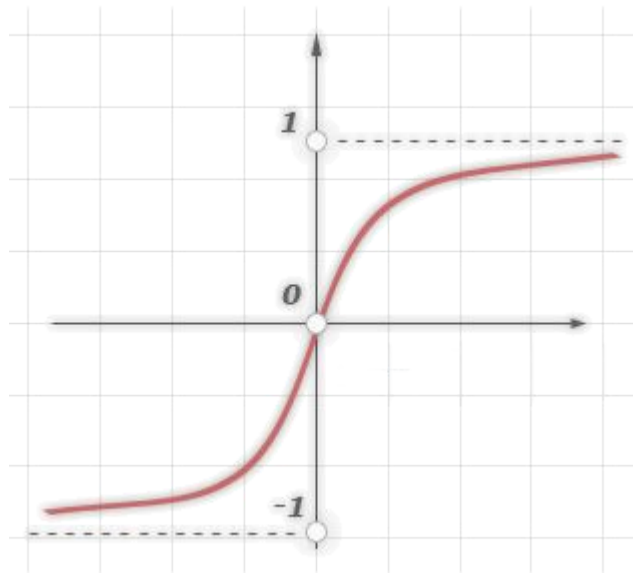
Принцип работы ИНН

Функция активации (функция сжатия)

Гиперболический тангенс

- Усиливает слабые сигналы и уменьшает сильные
- Простая, гладкая производная
- α — параметр наклона
- Диапазон от -1 до 1

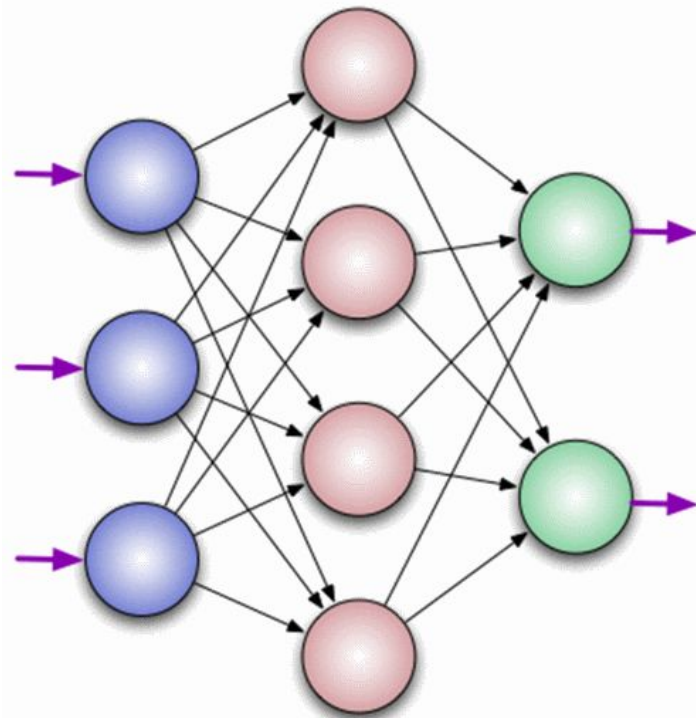
$$OUT = th \left(\frac{Y}{\alpha} \right)$$



Принцип работы ИНН

Общий алгоритм:

1. Получаем данные на вход
2. Переходим на следующий слой
3. Для каждого нейрона считаем соответствующую взвешенную сумму
4. Добавляем bias
5. Применяем функцию активации
6. Repeat from 2 until vikhondoi sloi
7. Получаем ответ на выходном слое, как-то его интерпретируем



Пример работы ИНН

Многослойная нейронная сеть для распознавания рукописных цифр.

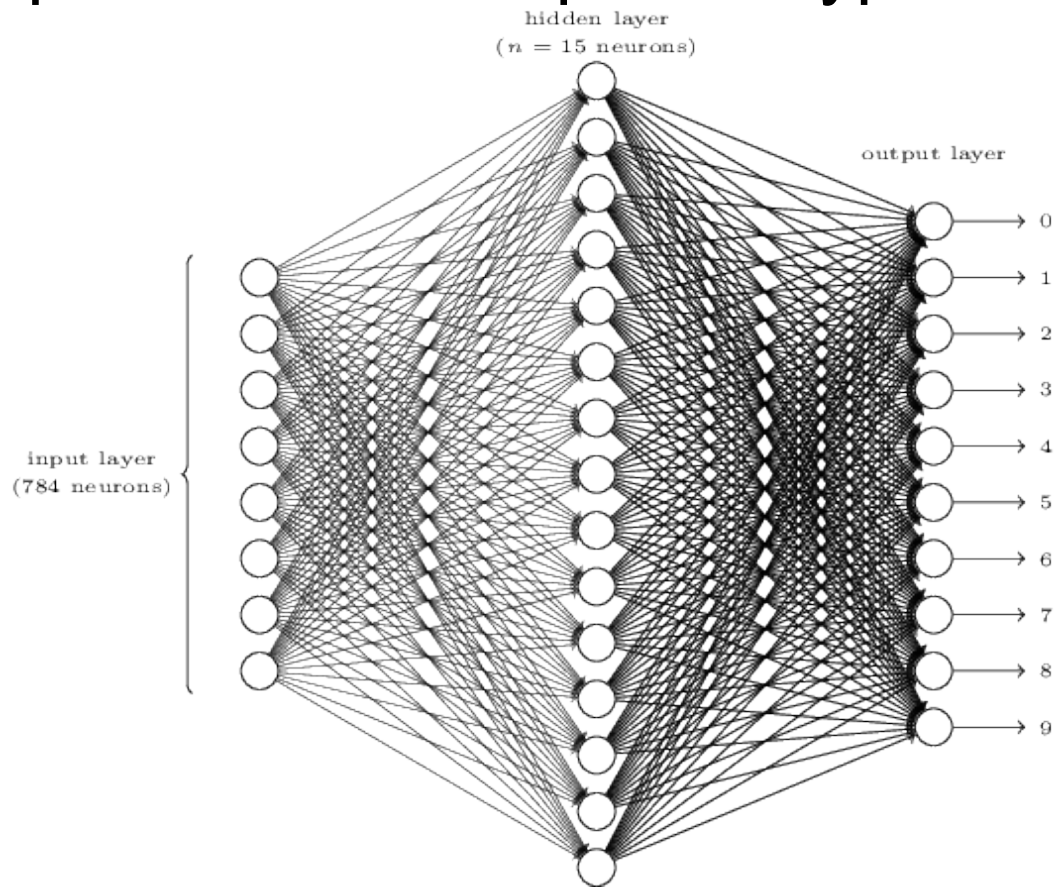
Вход: Изображение рукописной цифры от 0 до 9 размера 28x28 пикселей.

Задача: Распознать цифру и вывести её.

Примеры входа:



Пример работы ИНН. Архитектура сети

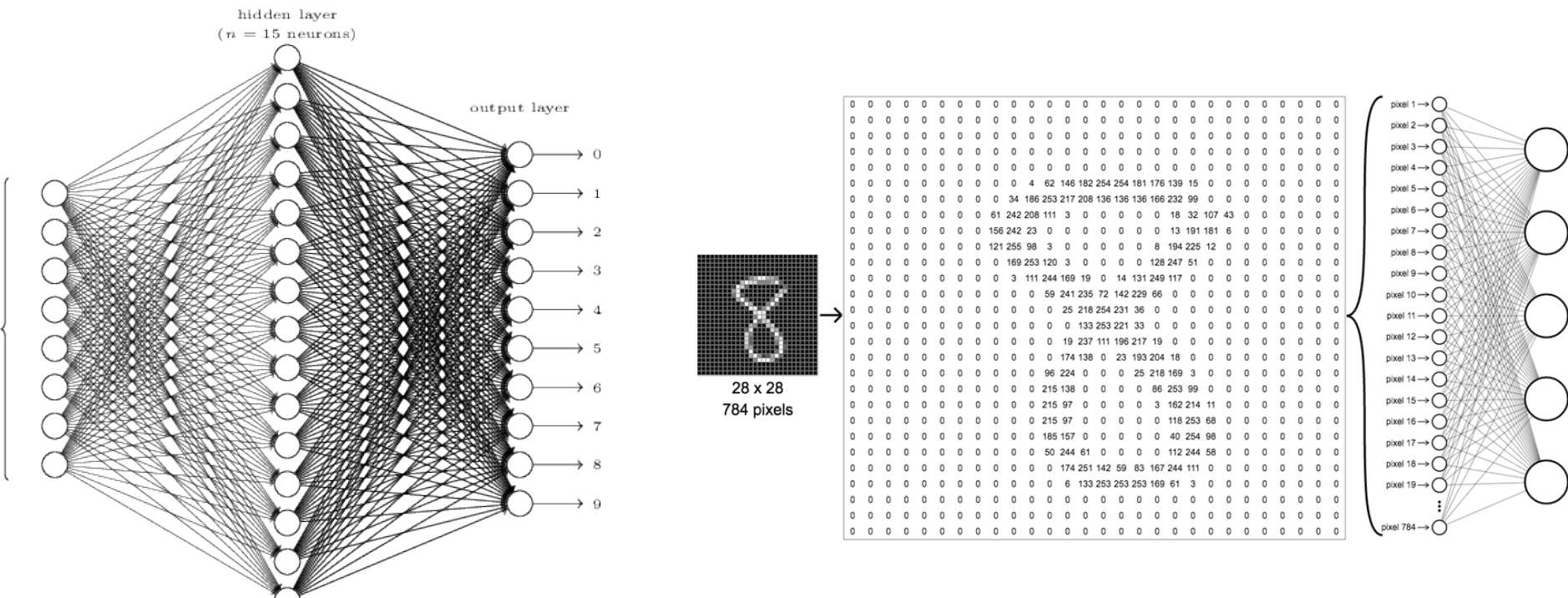


Пример работы ИНН

- $28 \times 28 = 784$ входных нейрона
- Один скрытый слой из 15-ти нейронов
- 10 выходных нейронов, по одному на каждую цифру
- Функция активации — сигмоида
- Идеальный ответ — 1 на нейроне нужной цифры, 0 на остальных нейронах.
- Ответ сети — цифра, нейрон которой имеет наибольшую активацию ($\text{argmax}(\{\text{Выходные нейроны}\})$)

Пример работы ИНН. Архитектура сети

- Точность >90%



Математическая модель ИНН

Простая нейронная сеть однозначно определяется:

- Весами
- Смещениями (bias)
- Функцией активации

W — матрица весов

w_{ij} — вес связи, соединяющей j -ый нейрон предыдущего слоя с i -ым нейроном данного слоя

b — вектор смещений слоя

f — функция активации

$a^{(L)}$ — вектор активаций (вектор выходов нейронов) слоя L обозначим

Математическая модель ИНН

В такой модели вектор активации слоя L задаётся простым выражением:

$$a^{(L)} = f \left(W \cdot a^{(L-1)} + b \right)$$

Обучение нейронных сетей

- Обучающая выборка
- Построение архитектуры
- Выбор функции потерь (MSE, root MSE, atan)
- Итерации по обучающей выборке
- Эпохи обучения

Обучение нейронных сетей

- Метод коррекции ошибки
- Генетический алгоритм
- Метод обратного распространения ошибки
- Метод упругого распространения

Обратное распространение ошибки

$$1) \delta_o = (\text{OUT}_{\text{ideal}} - \text{OUT}_{\text{actual}}) * f'(\text{IN})$$

$$2) \delta_H = f'(\text{IN}) * \sum(w_i * \delta_i)$$

$$f'(\text{IN}) = \begin{cases} f_{\text{sigmoid}} = (1 - \text{OUT}) * \text{OUT} \\ f_{\text{tanh}} = 1 - \text{OUT}^2 \end{cases}$$

Обратное распространение ошибки

Градиент для связи между нейронами A и B:

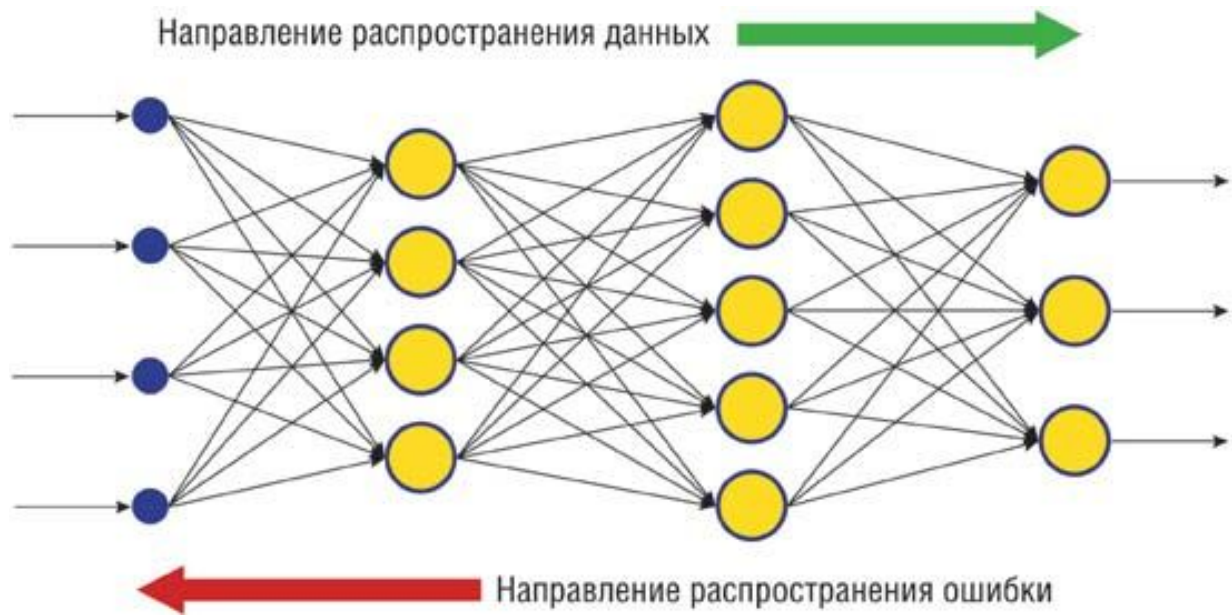
$$\text{GRAD}_B^A = \delta_B * \text{OUT}_A$$

Градиентный спуск для веса связи между нейронами A и B:

$$\Delta w_i = E * \text{GRAD}w + \alpha * \Delta w_{i-1}$$

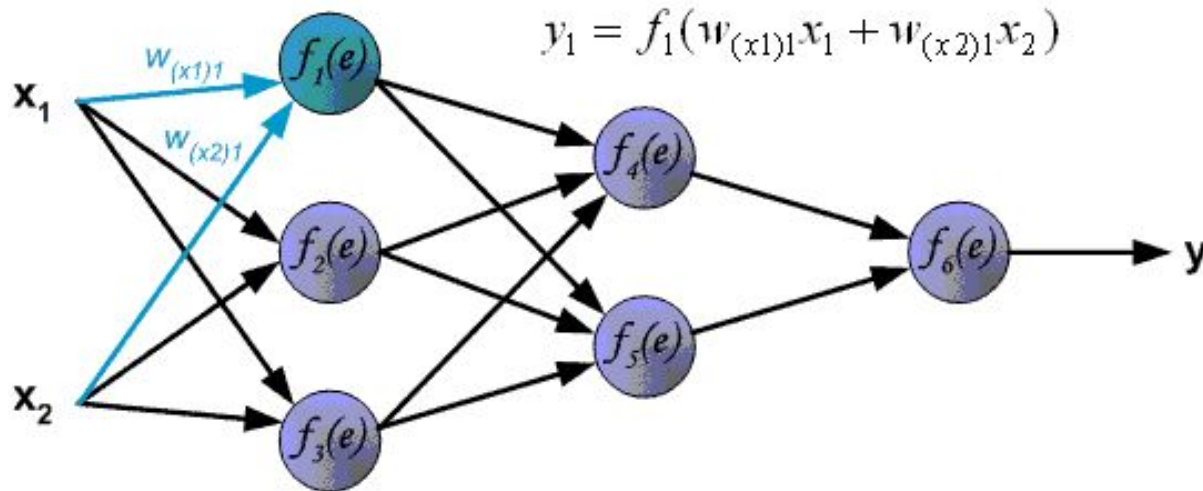
Обратное распространение ошибки

1. Рассчитываем ошибку на выходных нейронах
2. Рассчитываем ошибку на всех предыдущих слоях, распространяя ее на выходящие связи
3. Пересчитываем веса
4. ???
5. PROFIT



Обратное распространение ошибки

FP



Немного о гиперпараметрах

Примеры:

- Момент и скорость обучения
- Количество скрытых слоев
- Количество нейронов в каждом слое
- Наличие нейронов смещения

Проблемы в обучении НС

- Нехватка данных
- Переобучение
- Большой разброс значений в данных

Трюки для обучения НС

- Data Augmentation
- Dropout
- Batch normalization

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;

Parameters to be learned: γ, β

Output: $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad // \text{ mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{ mini-batch variance}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{ normalize}$$

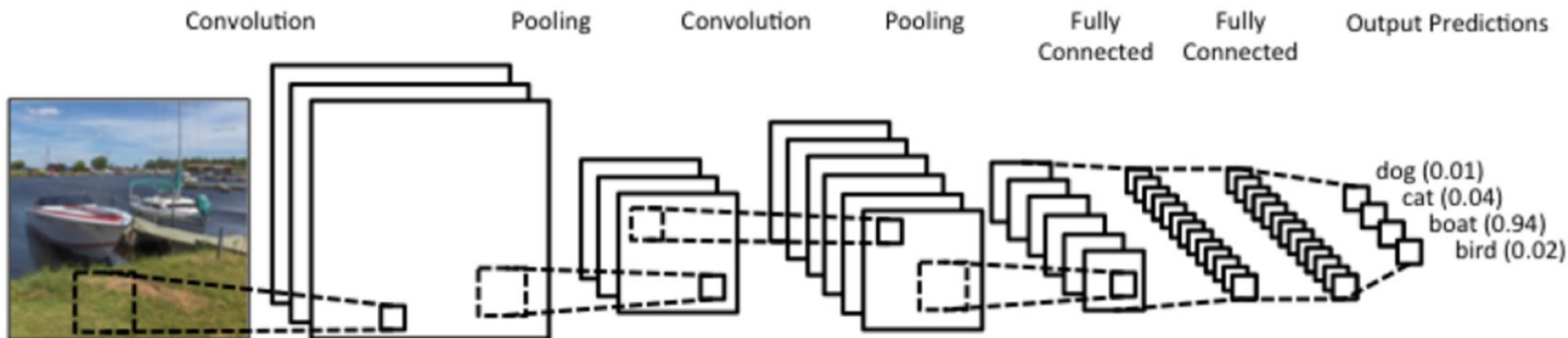
$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) \quad // \text{ scale and shift}$$

Виды нейронных сетей

- Нейронная сеть Хопфилда
- Машина Больцмана
- Цепи Маркова
- Автокодировщик
- Рекуррентные нейронные сети
- Сверточные нейронные сети

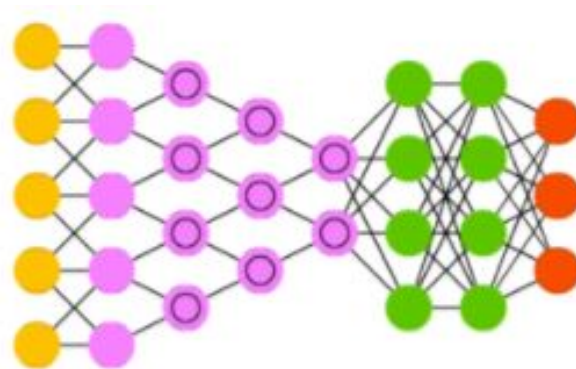
Сверточные НС

Основные области: компьютерное зрение, анализ аудиосигналов, семантический анализ



Сверточные НС

- Входной слой
- Слой свертки
- Слой субдискретизации (pooling)
- Полносвязный слой



Литература и ссылки

1. <http://neuronus.com/history/5-istoriya-nejronnykh-setej.html>
2. Ben Krose, Valter van de Smagt. Introduction to neural networks.
3. Саймон Хайкин. Нейронные сети полный курс.
4. Васенков Д.В. "Методы обучения нейронных сетей"
5. <https://habrahabr.ru/post/312450/>
6. <https://habrahabr.ru/post/313216/>
7. <https://tproger.ru/translations/neural-network-zoo-1/>
8. <https://basegroup.ru/community/articles/rprop>
9. <http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/>
10. <https://habrahabr.ru/post/309508/>