

Практикум по нейронным сетям

(представление методической
разработки)

Семинар по ИТ
19 февраля 2019 г.

Курс «Машинное обучение»

- Лекции

(<https://sok.susu.ru/courses/MachineLearnig/lectures/>)

- Практикум

- Лабораторные работы*

- Промежуточное контрольное тестирование

** Michael A. Nielsen, "Neural Networks and Deep Learning",
Determination Press, 2015 (<http://neuralnetworksanddeeplearning.com/>)*

Инструментарий

Цель: помочь студентам получить навыки по созданию и обучению искусственных нейронных сетей

- IDLE Python (<https://docs.python.org/3/>)
- NumPy (<http://www.numpy.org/>)
- Theano (<http://www.deeplearning.net/software/theano/>)

Практикум

1. Нейронная сеть для распознавания рукописных цифр
2. Моделирование конфигурации нейронной сети и параметров скорости её обучения
3. Стоимостная функция на основе перекрестной энтропии, регуляризация, улучшение способа инициализации весов и смещений.
4. Выбор топологии нейронной сети и подбор параметров обучения
5. Функция активации Softmax
6. Сверточные нейронные сети (Theano)

Нейронная сеть для распознавания рукописных цифр

Написать компьютерную программу на языке Python 3 создающую и обучающую нейронную сеть для распознавания рукописных цифр с использованием метода градиентного спуска и базы данных MNIST.

Исходные данные:

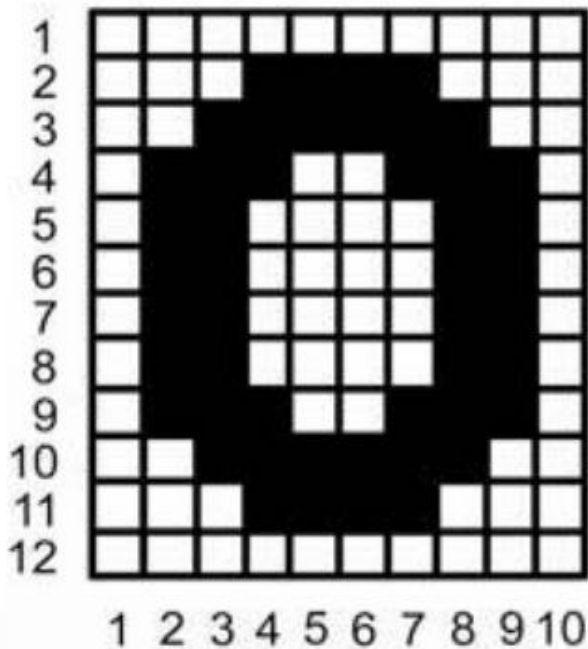
Количество слоев нейронной сети: 3.

Входные данные для нейронной сети: изображения размером 28 × 28 пикселей.

Среда программирования: Python 3.6 и выше.

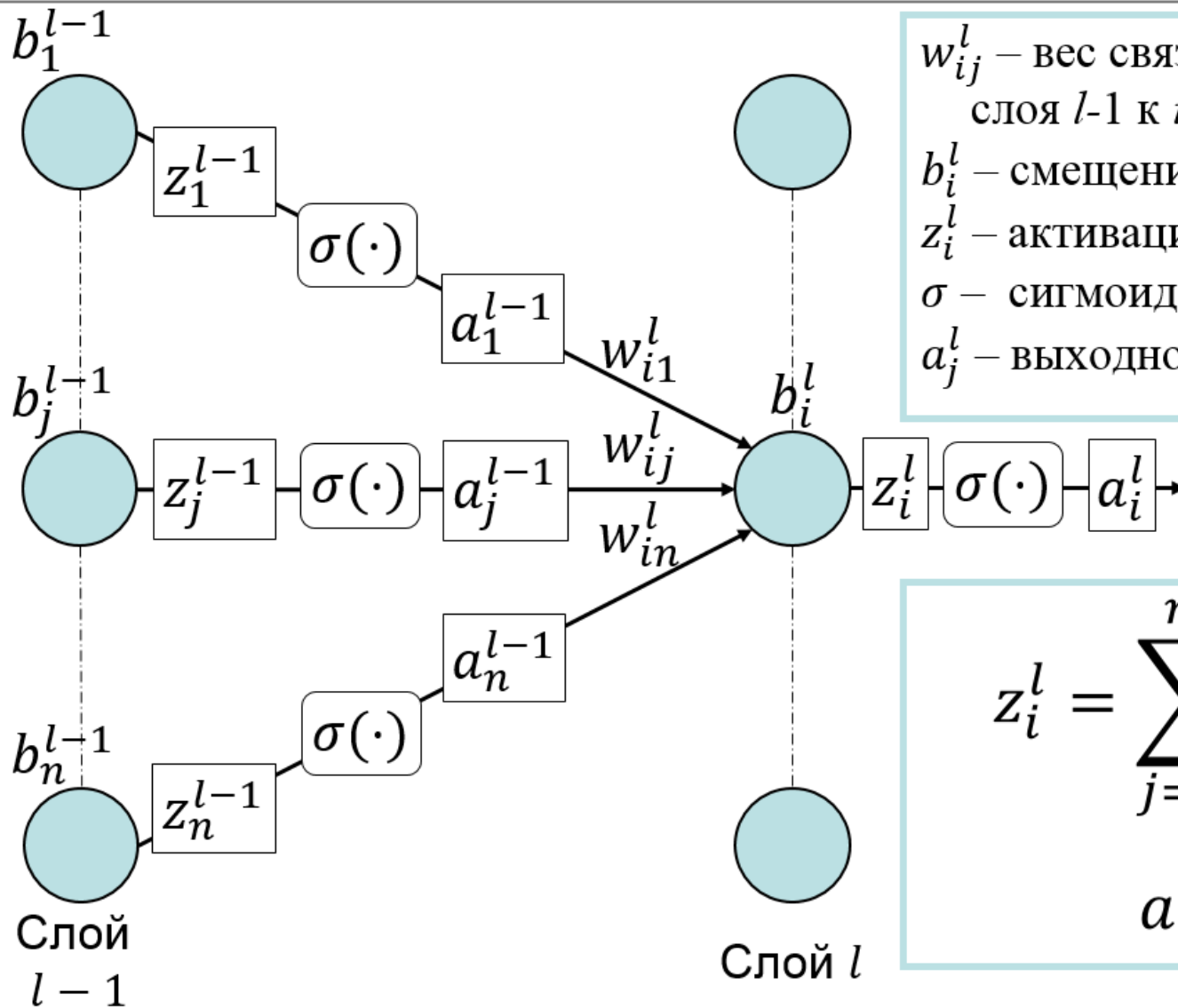
Используемая библиотека: NumPy.

0	4	1	9	2	1	3	1	4	3
5	3	6	1	7	2	8	6	9	4
0	9	1	1	2	4	3	2	7	3
8	6	9	0	5	6	0	7	6	1
8	7	9	3	9	8	5	9	3	3
0	7	4	9	8	0	9	4	1	4
4	6	0	4	5	6	1	0	0	1
7	1	6	3	0	2	1	1	7	9
0	2	6	7	8	3	9	0	4	6
7	4	6	8	0	7	8	3	1	5



0	0	0	0	0	0	0	0	0	0	<i>i</i>	1
0	0	0	1	1	1	1	0	0	0		2
0	0	1	1	1	1	1	1	0	0		3
0	1	1	1	0	0	1	1	1	0		4
0	1	1	0	0	0	0	1	1	0		5
0	1	1	0	0	0	0	1	1	0		6
0	1	1	0	0	0	0	1	1	0		7
0	1	1	0	0	0	0	1	1	0		8
0	1	1	1	0	0	1	1	1	0		9
0	0	1	1	1	1	1	1	0	0		10
0	0	0	1	1	1	1	0	0	0		11
0	0	0	0	0	0	0	0	0	0		12
<i>j</i>	1	2	3	4	5	6	7	8	9	10	

Обозначения



w_{ij}^l – вес связи от j -того нейрона слоя $l-1$ к i -тому нейрону слоя l
 b_i^l – смещение
 z_i^l – активационный потенциал
 σ – сигмоидная функция активации
 a_j^l – выходной сигнал

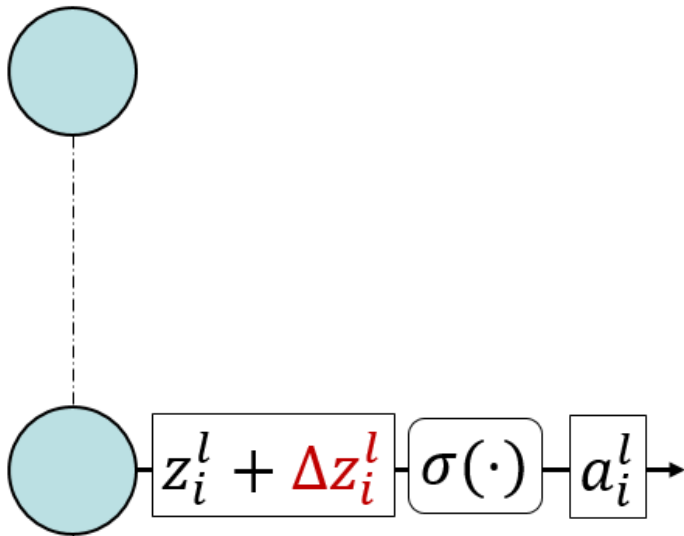
$$z_i^l = \sum_{j=1}^n w_{ij}^l a_j^{l-1} + b_i^l$$
$$a_i^l = \sigma(z_i^l)$$

Мера влияния нейрона на величину ошибки

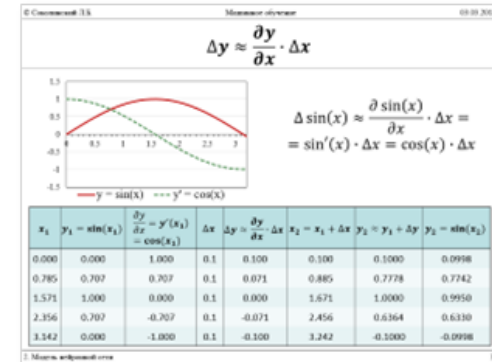
$$C = \frac{\|\vec{a}^L - \vec{y}\|^2}{2}$$

$$\Delta C \approx \frac{\partial C}{\partial z_i^l} \Delta z_i^l$$

$$\delta_i^l = \frac{\partial C}{\partial z_i^l}$$

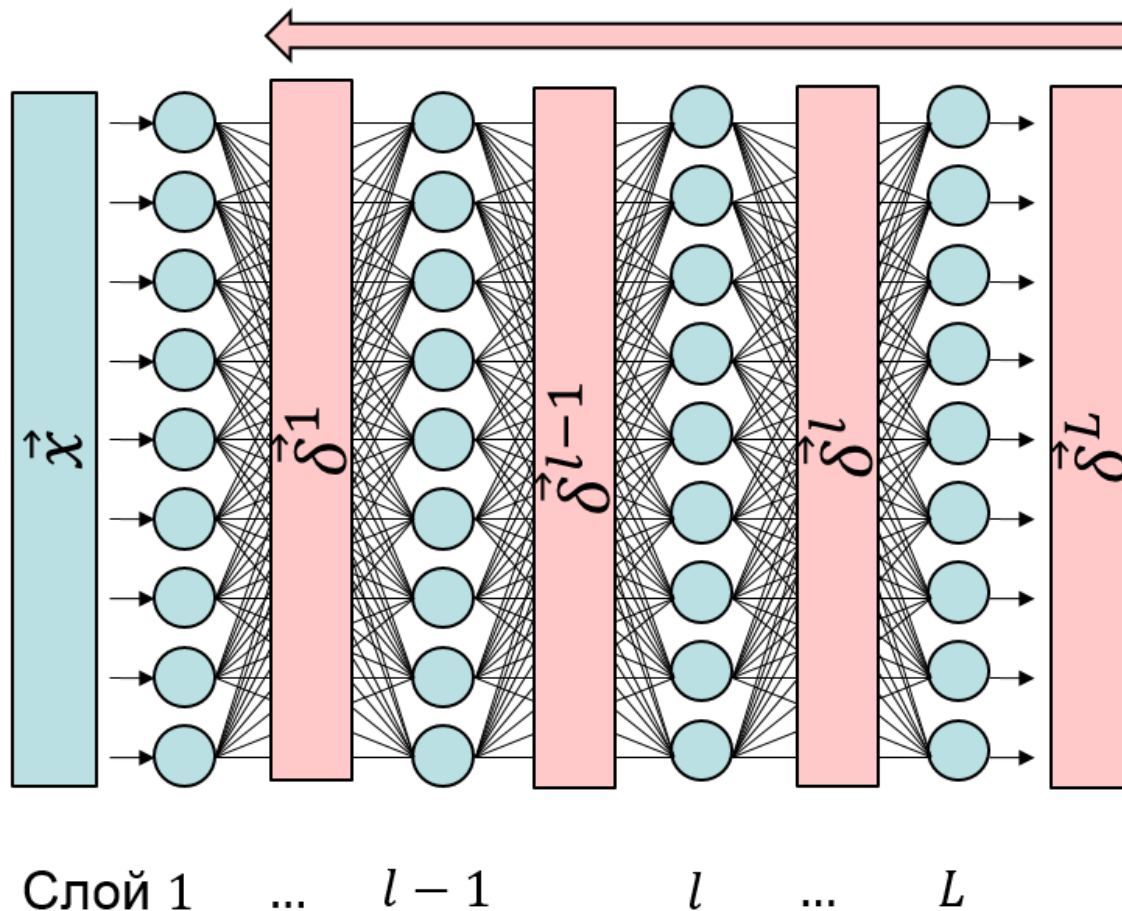


Слой l



Идея метода обратного распространения ошибки

Шаг 4. Вычислить в обратном порядке меру влияния δ^l на ошибку для каждого слоя



Формулы обратного распространения ошибки

$$\vec{\delta}^L = \nabla C(\vec{a}) \circ \vec{\sigma}'(\vec{z}^L) \quad (\text{BP1})$$

$$\vec{\delta}^l = \left((W^{l+1})^T \vec{\delta}^{l+1} \right) \circ \vec{\sigma}'(\vec{z}^l) \quad (\text{BP2})$$

$$\nabla C(\vec{b}^l) = \vec{\delta}^l \quad (\text{BP3})$$

$$\nabla C(W^l) = \vec{\delta}^{lT} \circ \begin{pmatrix} \vec{a}^{l-1} \\ \vdots \\ \vec{a}^{l-1} \end{pmatrix} \quad (\text{BP4})$$

Стохастический градиентный спуск

1. $\vec{w} := \overline{rnd}$; $\vec{b} := \overline{rnd}$; $epoch := 1$;
2. Последовательное разбиение V на непересекающиеся подвыборки V_1, \dots, V_M
3. $i := 0$;
4. $i := i + 1$;
5. $\nabla \mathbb{C}_{V_i}(\vec{w}) := \frac{1}{|V_i|} \sum_{(x,y) \in V_i} \nabla \mathbb{C}_{(x,y)}(\vec{w})$;
6. $\nabla \mathbb{C}_{V_i}(\vec{b}) := \frac{1}{|V_i|} \sum_{(x,y) \in V_i} \nabla \mathbb{C}_{(x,y)}(\vec{b})$;
7. $\vec{w} := \vec{w} - \eta \nabla \mathbb{C}_{V_i}(\vec{w})$;
8. $\vec{b} := \vec{b} - \eta \nabla \mathbb{C}_{V_i}(\vec{b})$;
9. **if** $i < M$ **goto** 4;
10. $shuffle(V)$; $epoch := epoch + 1$;
11. **if** $epoch \leq 10$ **goto** 2;

Цикл по подвыборкам

Цикл по эпохам обучения