

Распознавание искусственным интеллектом

Агенда

1. Проверка домашнего задания N°1
2. Проверка домашнего задания N°2

Проверка домашнего задания №1

colab.research.google.com

The Neural Network Zoo - The Asimov Institute

simple_keras.ipynb - Colaboratory

CO

simple_keras.ipynb

File Edit View Insert Runtime Tools Help

CODE TEXT CELL CELL

RAM Disk

EDITING

[1] # Необходимые библиотеки

from keras.applications.resnet50 import ResNet50, preprocess_input, decode_
from keras.preprocessing import image
import urllib.request
import numpy as np

Using TensorFlow backend.

Нейронная сеть с архитектурой ResNet50, обученная на imagenet dataset
model = ResNet50(weights="imagenet").

[3] # Путь до изображения
img_url="https://i.ytimg.com/vi/OC03qYvqQ8w/hqdefault.jpg"

[4] # Загружаем файл из интернета с картинкой и выполняем предобработку
tf,*_ =urllib.request.urlretrieve(img_url, "pic.jpg")
img = image.load_img(tf, target_size=(224, 224))
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)
x = preprocess_input(x)

[5] # Обучение модели, вывод классов
preds = model.predict(x)
decode_predictions(preds, top=5)

Downloading data from https://s3.amazonaws.com/deep-learning-models/image-m
10960/35363 [=====] - 0s 2us/step
[('n02123597', 'Siamese_cat', 0.36827978),
('n02127052', 'lynx', 0.22720693),
('n02124075', 'Egyptian_cat', 0.079959266),
('n02123045', 'tabby', 0.04382691),
('n02123159', 'tiger_cat', 0.02701163)]]

i.ytimg.com



classes









ANACONDA®



Pipenv

Проверка домашнего задания №2

Алгоритм обратного распространения ошибки

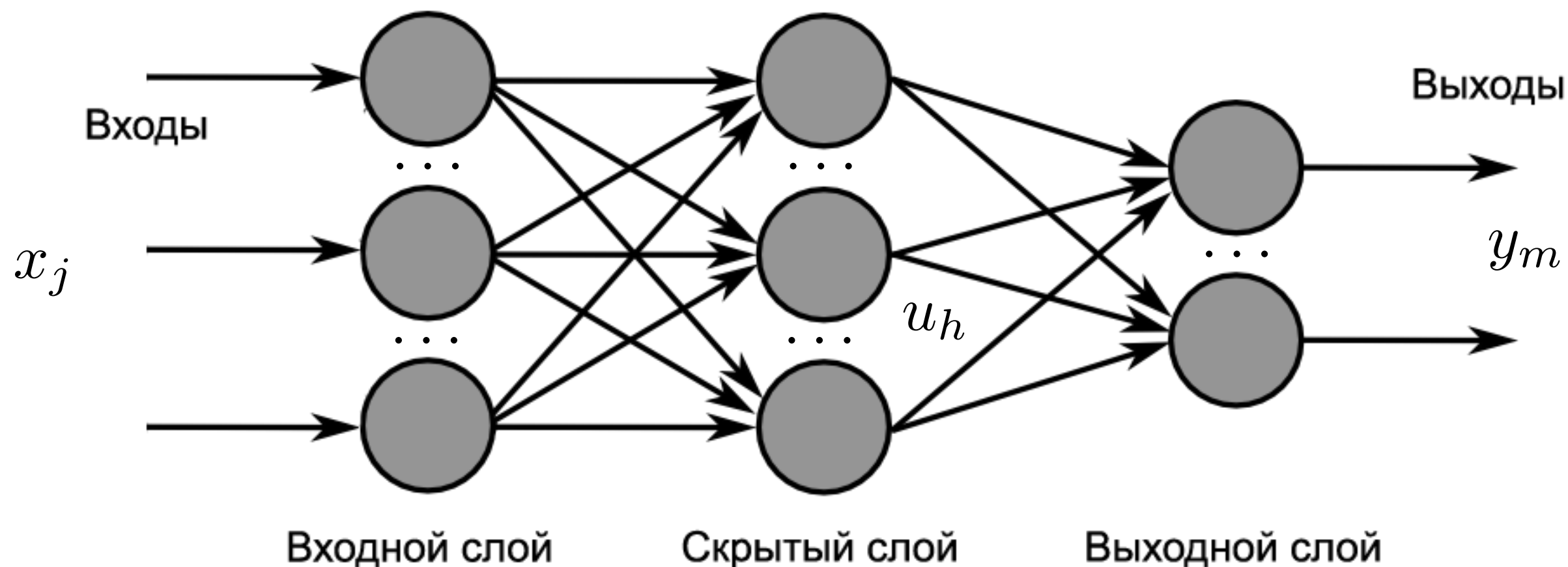
$$\vec{\omega}^{(\tau+1)} = \vec{\omega}^{(\tau)} - \eta \nabla_{\vec{\omega}} E[y(\vec{x}^{(\tau)}, \vec{\omega}^{(\tau)}), \hat{y}^{(\tau)}]$$

Стохастический градиентный спуск

$$E^{\tau} = \frac{1}{2} \sum_m (y_m^{\tau} - \hat{y}_m^{\tau})^2$$

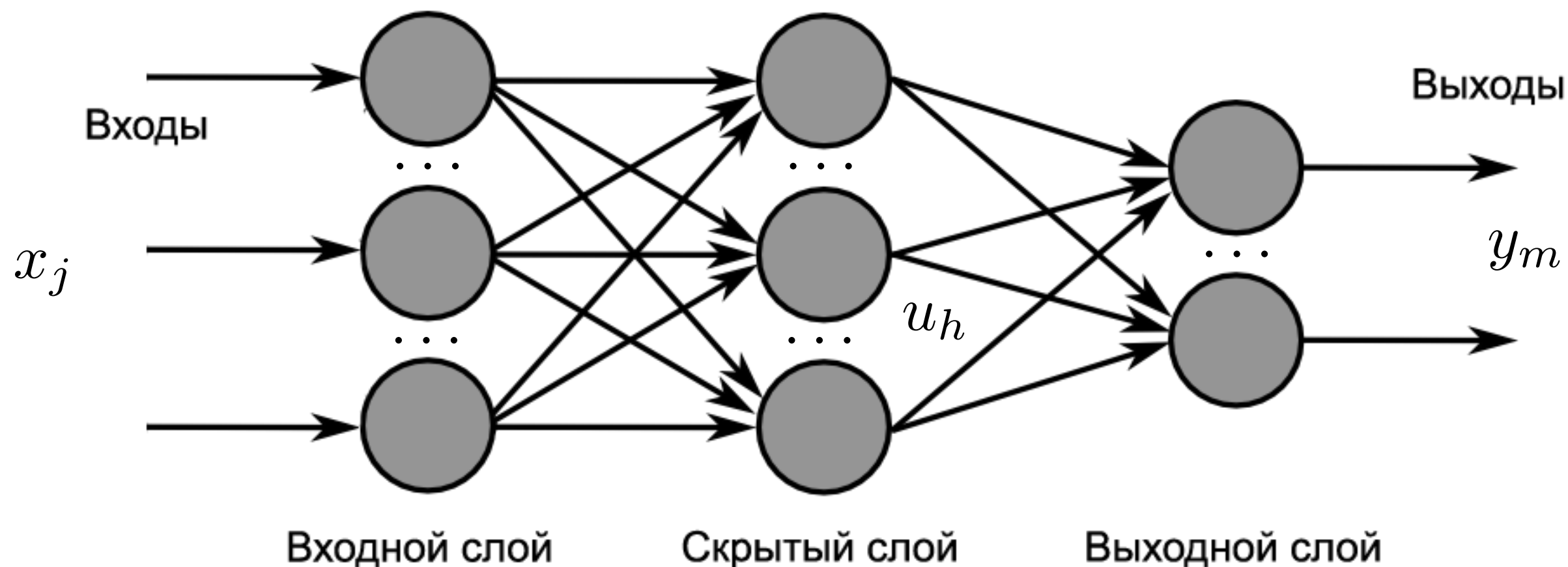
Функция ошибки

Алгоритм обратного распространения ошибки



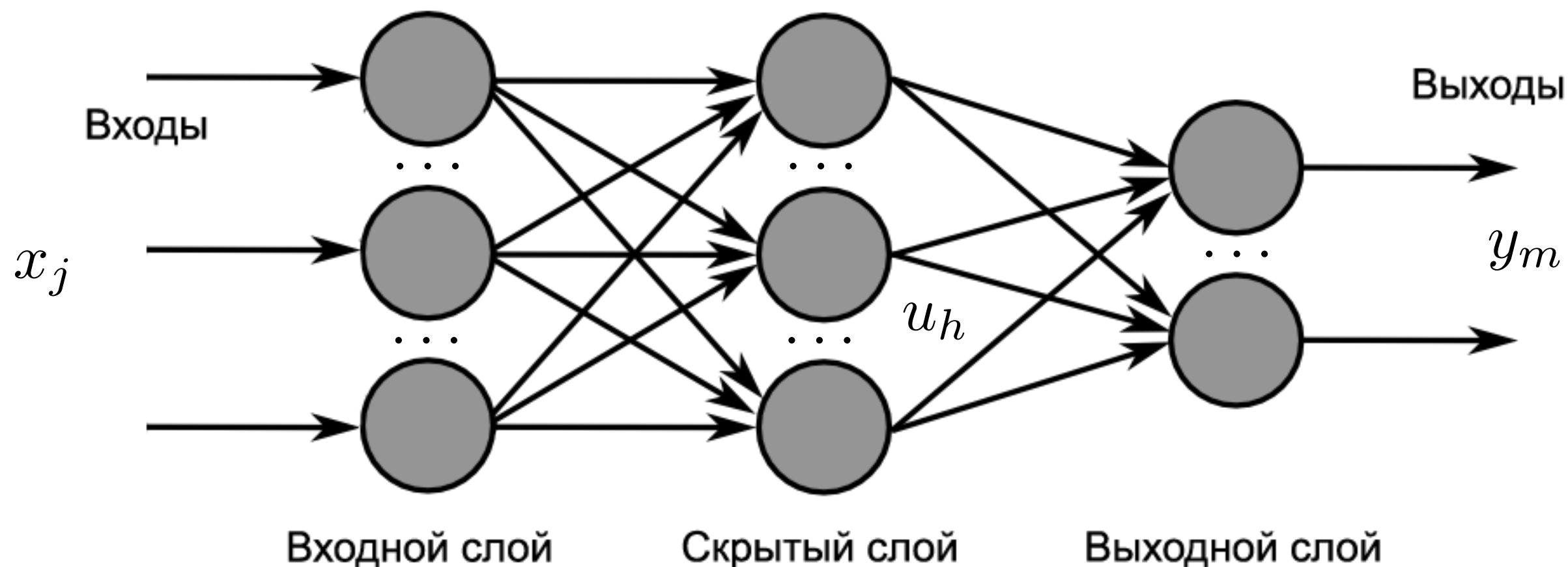
$\{\omega_{j,h}, \omega_{h,m}\}$ — параметры

Алгоритм обратного распространения ошибки



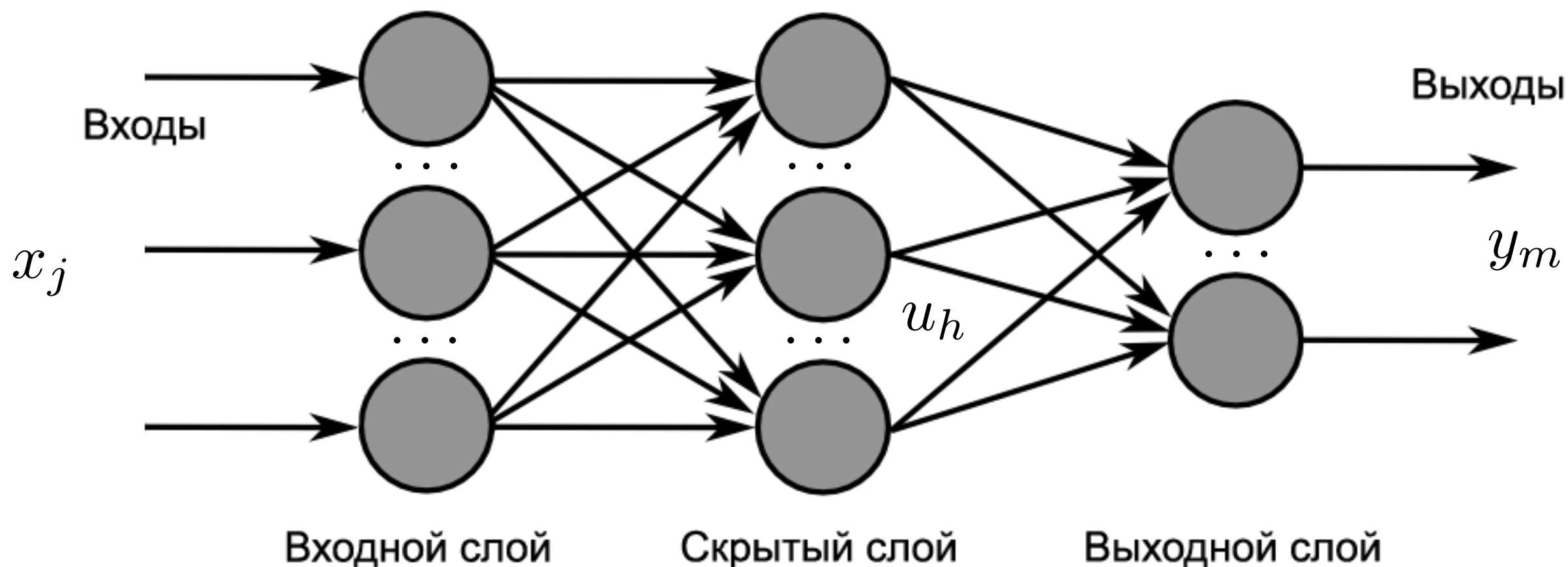
$$y_m = \sigma_m \left(\sum_h \omega_{h,m} u_h \right)$$

Алгоритм обратного распространения ошибки



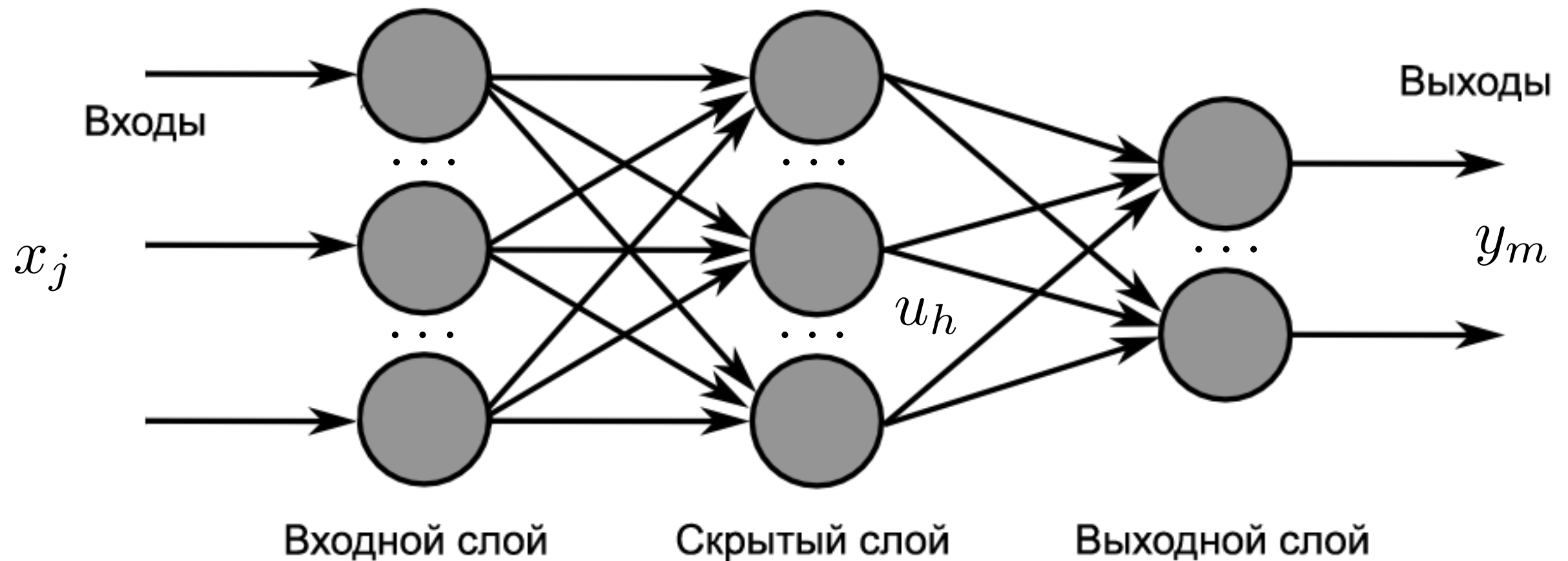
$$u_h = \sigma_h \left(\sum_j \omega_{j,h} x_j \right)$$

Алгоритм обратного распространения ошибки



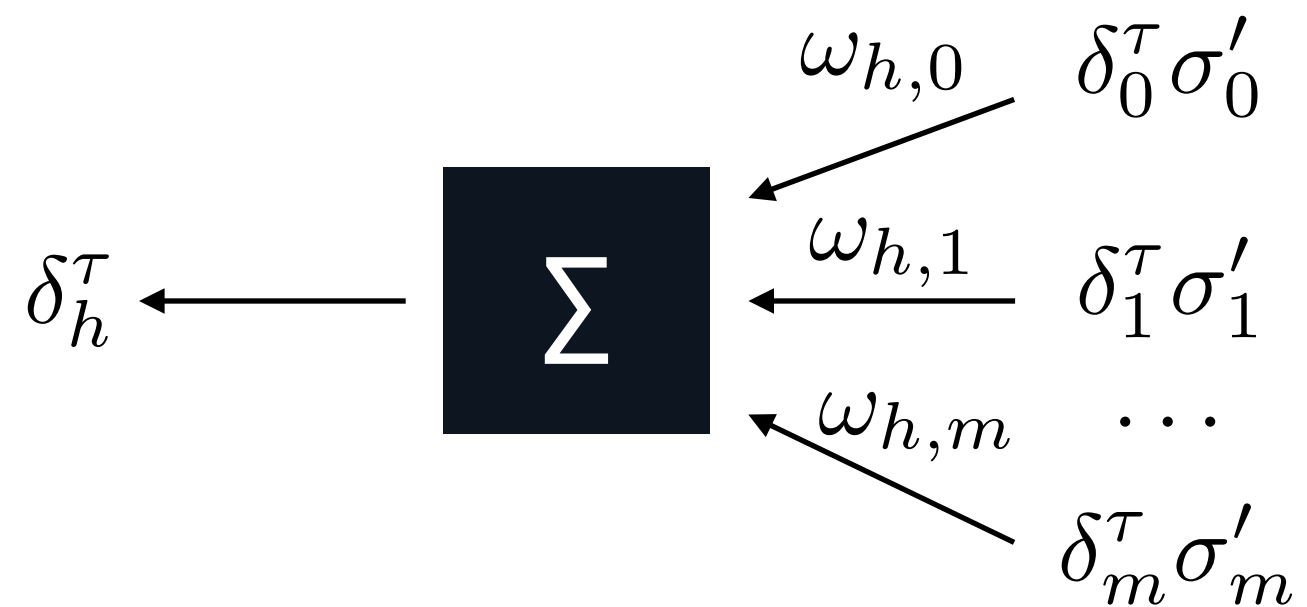
$$\frac{\partial E^\tau}{\partial \omega_{h,m}} = \frac{\partial E^\tau}{\partial y_m} \frac{\partial y_m}{\partial \omega_{h,m}} = \underbrace{(y_m^\tau - \hat{y}_m^\tau)}_{\delta_m^\tau} \sigma'_m u_h$$

Алгоритм обратного распространения ошибки



$$\frac{\partial E^\tau}{\partial \omega_{j,h}} = \frac{\partial E^\tau}{\partial u_h} \frac{\partial u_h}{\partial \omega_{j,h}} = \underbrace{\sum_m \delta_m^\tau \sigma'_m \omega_{h,m}}_{\delta_h^\tau} \sigma'_h x_j$$

Алгоритм обратного распространения ошибки



Алгоритм обратного распространения ошибки

Способ вычисления градиента в методе
градиентного спуска

Алгоритм обратного распространения ошибки

Шаг 1. Инициализируем веса.

Шаг 2. Для одного вектора выполняем прямой проход. u_h, y_m, δ_m

Шаг 3. Выполняем обратный проход. δ_h

Шаг 4. Корректируем веса.

Шаг 5. Проверяем условие остановки (точка минимума)

Алгоритм обратного распространения ошибки

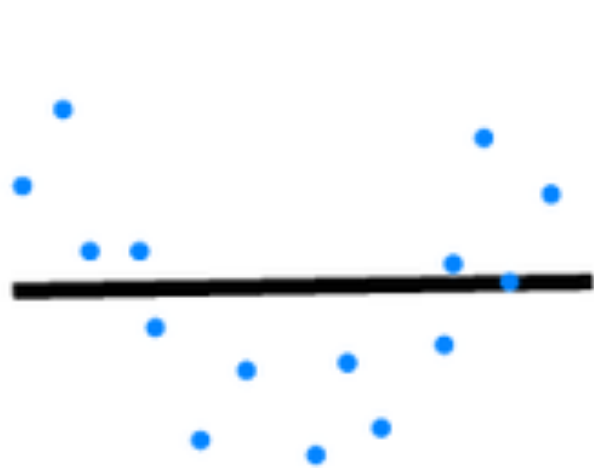
Плюсы

1. Эффективность
2. Обобщение
3. Потокное обучение
4. Распараллеливание

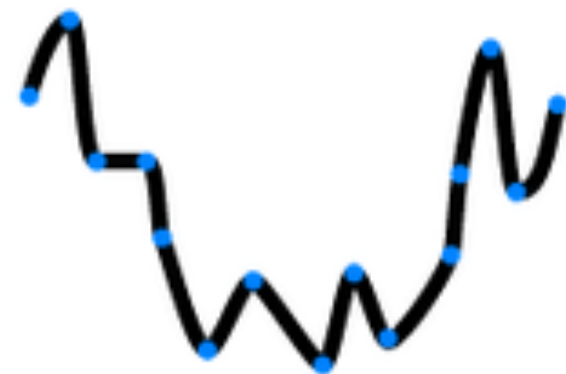
Минусы

1. Медленная сходимость
2. Рельеф функции
3. Оверфиттинг
4. Конфигурация

Алгоритм обратного распространения ошибки



Андерфиттинг



Оверфиттинг

Алгоритм обратного распространения ошибки

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

Алгоритм обратного распространения ошибки

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

Удобней при обучении глубоких сетей

Алгоритм обратного распространения ошибки

$$\text{step}(x) = \{1, x > 0; \quad 0, x < 0\}$$

Не подходит для многослойной сети

Алгоритм обратного распространения ошибки

$$\text{ReLU}(x) = \{x, x \geq 0; \quad 0, x < 0\}$$

Нет насыщения

Алгоритм обратного распространения ошибки

$$\text{LReLU}(x) = \{x, x \geq 0; \quad ax, x < 0\}$$

Обучение на отрицательных областях

Алгоритм обратного распространения ошибки

$$\text{ELU}(x) = \{x, x \geq 0; \quad a(e^x - 1), x < 0\}$$

Обучение на отрицательных областях +
насыщение

Алгоритм обратного распространения ошибки

$$\tilde{E} = E + \lambda \sum_i |\omega_i|$$

$$\tilde{E} = E + \lambda \sum_i \omega_i^2$$

Регуляризация и дропаут

WeChat

