

## Python и TensorFlow

На текущий момент Python является одним из самых популярных языков программирования в области машинного обучения. Для Python доступно большое количество библиотек и фреймворков машинного обучения.

### **УСТАНОВКА PYTHON 3**

Python – современный мультипарадигмальный язык программирования. На данный момент используются 2 основные версии Python (2 и 3).

Исходные коды написанные для этих разных версий несовместимы. В данной работе предлагается использовать более новую версию, т.е. версию 3.

#### **1 Установка на Windows**

##### **1.1 Установка посредством Chocolatey**

Chocolatey [1] – пакетный менеджер для Windows. Для знакомых с Ubuntu, пакетный менеджер – удобный способ устанавливать необходимые программы.

Python 3 устанавливается следующей командой:

```
C:\> choco install python3
```

##### **1.2 Установка посредством платформы Anaconda**

См. <https://www.anaconda.com>

##### **1.3 Непосредственная установка**

Если Chocolatey не установлен на компьютер процесс установки следующий:

1) Нужно скачать файл инсталлятора с официального сайта Python [2]

<https://www.python.org/downloads/>

2) Запустить инсталлятор и, следуя инструкции, установить Python.

3) Чтобы Python был доступен из командной строки, добавить в переменную PATH путь к папке Scripts, которая находится в папке, куда установлен Python.

#### **3 Установка на Ubuntu**

После установки дистрибутива Ubuntu Python 2 и 3 уже установлен,

но по умолчанию активна 2-ая версия. Чтобы использовать 3-ю версию из командной строки, к командам нужно добавлять суффикс 3 (т.е. python → python3).

### **МЕНЕДЖЕР МОДУЛЕЙ PYTHON**

PIP [3] – менеджер модулей Python. PIP входит в стандартную поставку Python начиная с версии 3.4.

Команда установки модуля имеет следующий вид:

```
pip install <module_name>
```

Чтобы найти имя нужного модуля можно поискать в интернете или воспользоваться командой:

```
pip search tensor *
```

## ***ВИРТУАЛЬНАЯ СРЕДА PYTHON(ОПЦИОНАЛЬНО)***

Задача виртуальной среды Python – создание изолированной среды для проектов Python. Это может быть полезно, если на рабочей машине активно ведется разработка проектов под разные версии Python.

Далее приведен краткий порядок работы с виртуальной средой, для уточнения деталей смотреть в [4].

### **1 Создание виртуальной среды**

Начиная с версии Python 3.6 создание виртуальной среды не требует установки дополнительных скриптов и делается следующей командой  
`python3 -m venv/path/to/virtual/env`

### **2 Активация виртуальной среды**

Чтобы начать работать в виртуальной среде, ее нужно активировать. Для этого из командной строки нужно запустить скрипт активации. Команда активации будет отличаться в зависимости от оболочки командной строки, по ссылке можно найти примеры для популярных оболочек [4]. Для оболочки Git-Bash команда имеет следующий вид:

`source/path/to/virtual/env/Script/activate`

Нужно заметить, что активация виртуальной среды происходит за счет изменения переменных среды текущего процесса. Поэтому при запуске новых процессов (запуск новой консоли) активацию нужно повторить.

Многие современные IDE (например PyCharm) при создании нового проекта сами предлагают создать виртуальную среду.

## ***МОДУЛИ PYTHON ПОЛЕЗНЫЕ ПРИ ВЫПОЛНЕНИИ ЛАБОРАТОРНЫХ РАБОТ***

Модули Python используемые в области машинного обучения, которые могут быть полезны при выполнении лабораторных работ [5].

- **matplotlib** : Это пакет для Python, позволяющий рисовать графики.
- **sklearn** : Библиотека Scikit-learn позволит привести в бинарную форму метки, разделить данные для обучения/тестирования и создать отчет об обучении в терминале.
- **TensorFlow**: Библиотека с открытым исходным кодом, которая позволяет разрабатывать и обучать модели машинного обучения.
- **keras**: Открытая нейросетевая библиотека. Предоставляет более удобный интерфейс для работы с TensorFlow.
- **imutils**: Библиотека предоставляет ряд удобных функций для выполнения базовых функций обработки изображений, таких как перенос, поворот, изменение размера, отображение изображений Matplotlib.
- **numpy** : NumPy предназначен для числовой обработки с помощью Python. Если установлен OpenCV для Python и Scikit-learn, то будет и NumPy (работа с массивами `flatten`, `reshape` ...).
- **cv2** : Библиотека, которая является обёрткой над OpenCV, библиотекой компьютерного зрения с открытым исходным кодом (работа с изображениями `imread`, `imwrite`, `imshow`...).
- **torch**: Библиотека, для работы с тензорным потоком данных для нейронных сетей на процессоре и устройствах cuda. Поддерживается обратное распространение градиента.

# ***TENSORFLOW***

## **1 Базовые концепции**

Фреймворк использует абстракцию графа потока данных, где данные представляются многомерными массивами (“тензорами”), над ними выполняются операции соответствующие вершинам графа потока данных. Поддерживается автоматическое дифференцирование и обратное распространение градиента по графу потока данных. Работа моделями и графами потока данных организуется в виде сессий.

## **2 TensorBoard**

Позволяет визуализировать граф потока данных и результаты работы tensorflow моделей.

## **3 Установка для cpu/avx/cuda**

Для установки необходимо выполнить:

```
pip install tensorflow
```

для работы на устройствах, возможно потребуется скомпилировать библиотеки или найти уже скомпилированные wheel-файлы, которые поддерживают работу с avx, avx2, cuda

для установки cuda необходимо установить драйверы nvidia для cuda

и dnnkit (файлы cudnn и др.). Необходимо убедиться в совместимости (одинаковости) версий cuda.

## ***KERAS***

Для установки необходимо выполнить:

```
pip install keras
```

## ***Порядок выполнения работы.***

1. Установить python (3.x)
2. Установить и настроить tensorflow и все необходимые пакеты
3. Установить keras
4. Выбрать демонстрационный и запустить пример (cifar10 CNN, cifar10 ResNet)
5. Приобретение навыков создания модели нейронной сети
6. Приобретение навыков задания метода обучения и оптимизируемой функции

## ***Ответить на вопросы.***

Как задать модель нейронной сети. Какие есть интерфейсы и их параметры?

Как задать весовые коэффициенты нейронной сети?

Как задать полносвязный слой нейронной сети?

Как задать свёрточный слой нейронной сети?

Какие есть средства для работы с рекуррентными нейросетями?

Как задать функцию активации нейронной сети и какие поддерживаются в keras?

Чем отличается linear от ReLU, softplus?

Как задать функцию ошибки\ потерь нейронной сети?

Чем отличается mean\_squared\_error от cosine\_proximity, по каким формулам они вычисляются?

Как задать метод обучения нейронной сети?

Чем отличается SGD от rprop, Adadelta, Adam; nesterov от momentum?

Как указать обучающую выборку?

## ***Подготовить отчёт***

Отчёт должен содержать:

1. Титульную страницу, примеры команд и результаты, демонстрирующие успешность установки.
2. Демонстрационные примеры и результаты их работы.
3. Ответы на вопросы.
4. Список источников.

## ***СПИСОК ИСТОЧНИКОВ***

- [1] Оф. сайт Chocolatey. — Электронный ресурс. — Режим доступа: <https://chocolatey.org/> — Дата доступа: 12.09.2019.
- [2] Оф. сайт Python. — Электронный ресурс. — Режим доступа: <https://www.python.org/> — Дата доступа: 12.09.2019.
- [3] Оф. сайт PIP. — Электронный ресурс. — Режим доступа: <https://pypi.org/project/pip/> — Дата доступа: 12.09.2019.
- [4] Документация виртуальной среды Python. — Электронный ресурс. — Режим доступа: <https://docs.python.org/3/library/venv.html> — Дата доступа: 12.09.2019.
- [5] Пример глубокого обучения с помощью библиотеки Keras. — Электронный ресурс. — Режим доступа: <https://keras.io> — Дата доступа: 12.09.2019.