

ДОПОЛНИТЕЛЬНЫЕ МАТЕРИАЛЫ ПО КУРСУ МРЗВИС

Установка и настройка среды для разработки на Python и TensorFlow

СОДЕРЖАНИЕ

1	Установка Python 3	2
1.1	Установка на Windows с помощью Chocolatey	2
1.2	Установка на Windows	3
1.3	Установка на Ubuntu	3
2	Менеджер модулей Python	3
3	Виртуальная среда Python (Опционально)	3
3.1	Создание виртуальной среды	4
3.2	Активация виртуальной среды	4
4	Модули Python полезные при выполнении лабораторных работ .	4
5	Введение в Keras	5
	Список использованных источников	6

На момент написания данных методических материалов (2019 г.) Python является одним из самых популярных языков программирования в области машинного обучения. Для Python доступно большое количество качественных библиотек и фреймворков машинного обучения.

1 УСТАНОВКА PYTHON 3

Python - современный мультипарадигменный язык программирования. На данный момент используются 2 мажорные версии Python (2 и 3). Исходные коды написанные для разных версий не совместимы. В данной работе предлагается использовать более новую версию, т.е. 3.

1.1 Установка на Windows с помощью Chocolatey

Chocolatey [1] - пакетный менеджер для Windows. Многие пользователи Ubuntu знают, что пакетный менеджер это удобный способ устанавливать необходимые программы.

Python 3 устанавливается следующей командой:

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

1.2 Установка на Windows

Если Chocolatey не установлен на компьютер процесс установки следующий:

- 1) Нужно скачать файл инсталлятора с оф. 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.

1.3 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`).

2 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*
```

3 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].

3.1 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
```

3.2 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. По этому при запуске новых процессов (запуск новой консоли) активацию нужно повторить.

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

4 МОДУЛИ PYTHON ПОЛЕЗНЫЕ ПРИ ВЫПОЛНЕНИИ ЛАБОРАТОРНЫХ РАБОТ

Перечислим модули Python использующееся 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 у вас установлен OpenCV для Python и Scikit-learn, то у вас будет NumPy.
- cv2 : Библиотека, которая является оберткой над OpenCV, библиотекой компьютерного зрения с открытым исходным кодом

5 ВВЕДЕНИЕ В KERAS

После того как среда разработки настроена можно приступить к дальнейшему изучению библиотек машинного обучения. Для этого можно выполнить данный урок [5] или обратиться к поисковым системам.

СПИСОК ИСПОЛЬЗОВАННЫХ 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://www.pyimagesearch.com/2018/09/10/keras-tutorial-how-to-get-started-with-keras-deep-learning-and-python/> — Дата доступа: 12.09.2019.