

Рекомендации и подсказки

I. VScode

Основная среда разработки, рекомендуемая для использования при прохождении курса - [VS Code](#)

Если вы ещё с ней не работали, стоит установить и освоиться.

Для быстрой настройки среды под наш курс, импортируйте профиль `nn_python.code-profile`.

II. Conda

В основном в рамках курса, по крайней мере - в первом семестре, мы будем работать с языком `Python` и множеством его пакетов.

Для управления ими и воспроизводимости кода нам понадобится `Anaconda` - менеджер окружений и пакетов для языка `Python`.

Мы будем использовать его легковесную версию `Miniconda`, без GUI и большого количества предустановленных библиотек. Инструкции по установке доступны здесь: <https://docs.anaconda.com/miniconda/install/>

В основном пригодятся следующие команды:

Команда	Описание
<code>conda</code>	Показать краткое описание всех доступных команд
<code>conda env list</code>	Вывести список существующих окружений.
<code>conda env create -n env_name</code>	Создать новое пустое окружение с именем <code>env_name</code>
<code>conda env create -f env_description.yml</code>	Создать новое окружение из <code>yaml</code> -описания. Аргументом <code>-n name</code> можно передать новое имя, отличное от задаваемого в <code>env_description.yml</code>
<code>conda activate env_name</code>	Переключить окружение на <code>env_name</code> . <i>Нюанс: Нужно помнить, что команда действует только на текущую сессию терминала - т.е. можно открыть сессии с разными окружениями. В VScode, по умолчанию, новое окно терминала запускается в последнем активированном окружении.</i>
<code>conda deactivate</code>	Выйти из текущего окружения. Аналогично предыдущей, действует только на текущую сессию

Команда	Описание
	терминала.
<code>conda env remove [-n env_name \ -p env_path]</code>	Удалить указанное окружение.
<code>conda export [-n env_name \ -p env_path] > env_description.yml</code>	Экспортировать окружение с именем <code>env_name</code> или расположенное по адресу <code>env_path</code> . С помощью аргумента <code>-f file</code> можно сразу указать файл, в который будет экспортироваться окружение, другой вариант - направить поток вывода в конкретный файл: <code>> env_description.yml</code>
<code>conda install [-n env_name \ -p env_path] package_name=version</code>	Установить пакет <code>package_name</code> версии <code>version</code> в активное окружение, либо в указанное аргументом окружение.
<code>conda remove package_name</code>	Удалить пакет <code>package_name</code> из активного окружения. Можно указать другое окружение, как в предыдущей команде.
<code>conda list [-n env_name \ -p env_path]</code>	Показать список пакетов, установленных в активном окружении либо окружении, указанном аргументом.

Более подробную справку по каждой команде можно получить в терминале с помощью аргумента `-h`.

YAML-описания окружений довольно распространены в машинном обучении, поскольку позволяют легко воспроизвести рабочее окружение, и, следовательно, обеспечить стабильное исполнение кода.

YAML-описание окружения, с которым мы будем работать большую часть времени, выглядит так:

```
name: nn_env # Имя окружения
channels: # Каналы распространения
- defaults
- conda-forge
dependencies: # Зависимости
- python=3.12.1=h9e4cc4f_1_cpython # Можно явно указать версию пакета и
  версию сборки
- pytorch=2.5.1
- numpy # Либо не указывать: тогда установится последняя совместимая
  версия пакета
- scikit-learn
- tqdm
- pandas
- matplotlib
- seaborn
```

Нужно понимать, что при автоматизированном экспорте окружения зависимости описываются более подробно:

```
name: nn_env
channels:
- defaults
- https://conda.anaconda.org/conda-forge
dependencies:
- _libgcc_mutex=0.1=main
- _openmp_mutex=5.1=1_gnu
- blas=1.0=mkl
- bottleneck=1.4.2=py312ha883a20_0
# === 110 entries hidden ===
- zlib=1.2.13=h5eee18b_1
- zstd=1.5.6=hc292b87_0
prefix: /home/alexey/.conda/envs/nn_env
```

Это обстоятельство может приводить к проблемам со сборкой окружения и запуском программ на других машинах и платформах, поэтому лучше создавать уaml-файл вручную.