

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Макаренко Елена Николаевна

Должность: Ректор

Дата подписания: 29.07.2022 17:52:52

Уникальный программный ключ:

c098bc0c1041cb2a4cf926cf171d6715d99a6ae00adc8e27b55cbe1e26bd7c78

Лабораторная работа 0 (Знакомство с фреймворком PyTorch)

Задания:

1. Установите библиотеку PyTorch, следуя следующей инструкции:
<https://pytorch.org/get-started/locally/>
2. Познакомьтесь с "пулом" моделей PyTorch:
<https://pytorch.org/vision/stable/models.html>
3. Просмотрите документацию по трассировке моделей PyTorch:
<https://pytorch.org/docs/stable/generated/torch.jit.trace.html>
4. Просмотрите документацию по кодированию моделей PyTorch с помощью TorchScript: https://pytorch.org/tutorials/beginner/Intro_to_TorchScript_tutorial.html
5. Оттрассируйте любую модель из пула torchvision
6. Сохраните оттрассированную модель в виде файла
7. Загрузите сохранённую модель, провалидируйте её на корректную конвертацию

Лабораторная работа 1 (Конвертация модели на фреймворке PyTorch в CoreML)

1. Выберите любую модель классификации из [зоопарка моделей](#) на PyTorch.
2. Конвертируйте её по [следующей](#) инструкции.
3. Произведите предсказание полученной моделью, с этим шагом поможет [следующая](#) инструкция.
4. Выберите одну из моделей YOLOv5, DeepLabV3, FCN на [PyTorchHub](#) и произведите с ней шаги 2 и 3.

Необходимо загрузить Jupyter Notebook.

Замечание к пункту 3: если у вас не MacOS, можете пропустить этот шаг.

Лабораторная работа 2 (Конвертация модели на фреймворке PyTorch в TensorFlow-Lite)

1. Выберите любую модель классификации из [зоопарка моделей](#) на PyTorch.
2. Конвертируйте её по [следующей](#) инструкции.
3. Произведите предсказание полученной моделью на ONNX с помощью **InferenceSession**.
4. Конвертируйте модель на ONNX в Keras, рекомендуется воспользоваться [следующим](#) решением (сохраните замороженный граф TensorFlow).

5. Модель из шага 4 конвертируйте в TensorFlow-Lite модель, воспользуйтесь [официальной](#) документацией. Проверьте её работу.
6. Выберите одну из моделей YOLOv5, DeepLabV3, FCN на [PyTorchHub](#) и произведите с ней шаги с 2 по 5.

Необходимо загрузить Jupyter Notebook.

Лабораторная работа 3 (Запуск приложения с встроенной нейронной сетью на Android или iOS устройстве)

1. Необходимо взять за основу готовую реализацию приложения для Android/iOS с задачей классификации (в лекции 3 можно найти ссылки на соответствующие репозитории)
2. В данном коде заменить модель на ResNet101 (можно использовать PyTorch или TensorFlow-Lite фреймворки)
3. Собрать решение и запустить на мобильном устройстве данную модель, оценить скорость работы архитектуры (добавить профилирование времени работы вызова функции с моделью)

Лабораторная работа 4 (Запуск приложения с встроенной нейронной сетью на Android или iOS устройстве)

1. Необходимо взять за основу готовую реализацию приложения для Android/iOS с задачей детекции (в лекции 3 можно найти ссылки на соответствующие репозитории)
2. В данном коде заменить модель на YOLOv5 или SSD (можно использовать PyTorch или TensorFlow-Lite фреймворки)
3. Собрать решение и запустить на мобильном устройстве данную модель, оценить скорость работы архитектуры (добавить профилирование времени работы вызова функции с моделью)

Лабораторная работа 5 (Запуск приложения с встроенной нейронной сетью на Android или iOS устройстве)

1. Необходимо взять за основу готовую реализацию приложения для Android/iOS с задачей распознавания речи (в лекции 3 можно найти ссылки на соответствующие репозитории или по ссылкам: [Android](#), [iOS](#))
2. В данном коде заменить модель на [данную реализацию](#) (можно использовать PyTorch или TensorFlow-Lite фреймворки (необходима конвертация исходной модели))
4. В коде необходимо заменить вывод текста, извлечённого из аудиозаписи на вывод первых 5-ти значений предсказанного моделью вектора признаков
4. Собрать решение и запустить на мобильном устройстве данную модель, оценить скорость работы архитектуры (добавить профилирование времени работы вызова функции с моделью)

Лабораторная работа 6 (Model serving)

Необходимо ознакомиться с материалом из лекции 5 и интегрировать модель [YOLOv5](#) в сервер на [BentoML](#).