Сегментация изображения на основе нейросети U-NET

В документе приводятся комментарии к решению задачи сегментации изображения митохондрий на основе нейросети UNET.

**Использованные инструменты и источники:**

1. DigitalSreeni channel: https://www.youtube.com/watch?v=csFGTLT6\_WQ
2. ELECTRON MICROSCOPY DATASET <https://www.epfl.ch/labs/cvlab/data/data-em/>
3. Tensorflow 2.5.0, Numpy и др. вспомогательные библиотеки
4. Google Colab
5. Spyder IDE with Python 3.8 (после тренировки нейросети в Google Colab)
6. UNET: <https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/>

Нейросеть U-NET может быть применена для сегментации изображений. В данной работе рассматривается сегментация медицинских изображений из набора ELECTRON MICROSCOPY DATASET. Тренировочные и тестовые изображения были разделены на блоки 256x256.

**Подготовка данных.**

Набор данных включает в себя многостраничные tif – изображения (training.tif,training\_groundtruth.tif, testing.tif, testing\_groundtruth.tif). Их нужно разбить на последовательности изображений. Для этого применяется программа Fiji. После открытия изображния следует перейти в File->Save As->Image Sequence. После чего выбрать наименование и сохранить в нужную. Директорию. Разрешение полученных изображений 1024x768. Для использования, их нужно разделить на более мелкие фрагменты (например, на 256x256). Подобная процедура описана в скрипте: create\_train\_and\_test\_sets\_256\_256.py. После его исполнения в директории проекта появятся несколько поддиректорий, каждая из которых содержит 1980 файлов изображений:

1. test\_256\_256 – тестовые изображения.
2. test\_groundtruth\_256\_256 – соответствующие маски для тестовых изображений.
3. train\_256\_256 – тренировочные изображения.
4. train\_groundtruth\_256\_256 – соответствующие маски для тренировочных изображений.

Стоит отметить, что размеры тренировочной выборки и тестовой равны (1980 изображений). Для тренировки этого недостаточно. Поэтому после того, как все изображения считаны и сохранены в numpy - массивы, применяется изменение размера тренировочных и тестовых данных (см. mitochondria\_nn\_script.py)

**Тренировка нейросети.**

U-model описана в скрипте: mitochondria\_nn\_model.py. Модель создана при помощи библиотеки tensorflow. Параметры подбирались эмпирически. Критерием являлась метрика Intersection Over Union (см. mitochondria\_nn\_script.py).

При отсутствии видеокарты для тренировки сети следует использовать Google Colab. Но в данном случае имеется ряд моментов:

1. Изображения для тренировки и тестирования следует поместить на Google Disk, предварительно заархивировав в ZIP – формат. Для извлечения имеется пакет zipfile.
2. Google Disk располагается в /content/drive
3. Саму нейросеть лучше выполнить в виде отдельного модуля и также поместить на Google Disk. Для его подключения следует добавить путь в переменную sys.path (см. mitochondria\_nn\_script.py).
4. После того, как тренировочные и тестовые изображения будут преобразованы в тензоры, их можно сохранить в бинарный формат с помощью pickle, чтобы в последствии повторно не выполнять процедуру по конвертации.
5. В процессе тренировки использовались следующие обратные вызовы: EarlyStopping (завершение тренировки, когда потери на валидации перестают уменьшаться), TensorBoard (для визуализации графиков обучения), ModelCheckpoint (для сохранения лучшей модели).
6. Если изменяется среда выполнения, то все переменные сбрасываются. Поэтому лучше сразу выбирать среду выполнения с GPU.
7. Для просмотра графиков обучения использовался TensorBoard. Для его запуска в отдельной ячейке следует набрать:

%load\_ext tensorboard

%tensorboard --logdir logs