TTS

В работе была реализована архитектура FastSpeech2 на основе ноутбука с семинара. Было оставлено то же выравнивание. Нормализация, извлечение фонем и извлечение MEL-спектрограм также остались такими же, как в FastSpeech.

Для получения новых регуляторов репозиторий с FastSpeech был немного изменен: в файл audio/tools.py добавлены функции get_pitch, get_energy и get_emotion и в data/ljspeech.py, hparams.py и т.п. добавлены вспомогательные функции и пути, чтобы при запуске preprocess.py создавались такие же папки для данных, как и ./mels.

Get_pitch считает f0 с помощью pyworld.dio, get_energy считает энергию как l2 норма амплитуды по sfft окнам. Get_emotion предсказывает для каждой точки эмоцию по предобученной модели.

Почти все параметры модели остались такими же, как в FastSpeech. Изменения: использование pre-norm в трансформере, предсказывание log-duration, добавляется VarianceAdaptor, в котором дополнительные регуляторы преобразуются в размер входа так: диапазон значений делится на 256 частей, признаки преобразуются в ohe. Сами значения векторов pitch и energy изначально делились на их среднее, чтобы понизить порядок для лучшего обучения.

Во время предсказания векторы pitch и energy умножаются на соответствующий коэффициент, как мы хотим изменить параметры, а для эмоций используется значение целевой эмоции с коэффициентом 0.9 и 0.1 от предсказания.

Итоговый чекпоинт модели получился после 200 эпох. При этом были сложности с обучением лосса для VarianceAdaptor, нормализация помогла для обучения pitch и energy, с эмоциями также остались сложности, при попытке изменить значения градиент начинал взрываться после 1 эпохи. В конце генерируемые записи для каждых значений эмоций отличаются, но не

значительно. Качество звука всегда хорошее, так как mel_loss всегда хорошо падает.

Графики лоссов тут: ссылка, все три крывые отвечают за одну модель, которая обучалась в несколько подходов.