**<머신러닝 Q & A list>**

1. **RNN의 shape의 궁금하다.**

[개인공부]

Hello를 예측한다고 가정하자.

Input\_data의 shape이 (3,5,4)라고 할 때,

3 = batch size (H,e,l,o로 구성된 아무 단어 3개)

5 = sequence length (Hello의 문자열 길이)

4 = input size (H,e,l,o 총 4개의 letter 존재)

이를 one-hot-vector로 나타낸다.



\*중요

sample\_idx = [char\_dic[c] for c in sample]

x\_data = [sample\_idx[:-10]]

x\_one\_hot = [np.eye(input\_size)[x] for x in x\_data]

y\_data = [sample\_idx[10:]]

# tensor

X = torch.FloatTensor(x\_one\_hot)

Y = torch.LongTensor(y\_data)

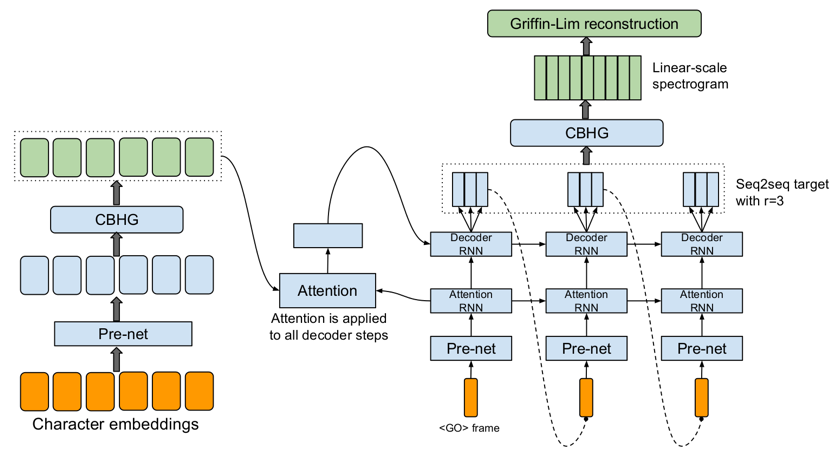


X와 Y의 shape이 일치할 수 있도록 해야 한다.

가령 , x의 sequence length는 15이고 이때 input\_size에 해당하는 letter들은 18개가 존재한다.

이때 대응되는 y의 sequence 역시 15개 여야한다.

1. **TacoTron 구조**

****

**[발표 중 질문]**

Q1. new data에 adaptable하다: <text,audio>로 pair가 이뤄지기 때문에 가능

(unseen, new domain이든)

Q2.기본적으로 encoder-decoder 모델은

Ecoder의 마지막 LSTM의 hidden state가

Decoder의 첫번째 LSTM의 hidden state로 들어간다.

Q3. CBHG에서 Conv1D Bank는 여러개의 kernel사이즈를 가지고 있다. (encoder에서는 16개, decoder에서는 8개)

이를 통해서 여러개의 local한 feature들을 뽑아 낼 수 있다. (이는 convolution의 기본적인 특성)

Q4. GRU는 RNN의 일종으로 LSTM을 간소화시킨 모델이고 Bi-directional 이기 때문에 양방향으로 학습한다고 보면 됩니다.

Q5. L1 Distance는 각 절댓값들의 합을 의미합니다.

1. DCGAN

[파이토치 튜토리얼]

DCGAN을 통해서는 기존 GAN방식에서 CNN이 추가되었다고 생가하면 된다.

1. Torchaudio

[파이토치 튜토리얼]

Q. Spectrogram 의 shape 이 3인데 각각은 무엇을 의미하나요?

🡪 Shape = (x,y,z): 채널,음압,시간

**파이토치의 torchaudio를 통해서**

**우리의 음성신호를 데이터로 입력받고 이를**

**Melspectrogram형식으로 변환가능하다.**

1. Google mount

from google.colab import drive

drive.mount('/content/drive')

filename = '/content/drive/My Drive/Colab Notebooks/data/label.csv'

1. Normalization

Batch-Normalization : **Batch** 단위로 Normalization 이 이뤄진다. Instance-level을 보존한다.

**Instance-Normalization:** 자세히는 모르겠으나, 그림으로 파악하는게 좋고, instance-level을 보존하지 못한다. (**각 Feature에 대해서 Normalization)**

1. Voice conversion

특정 발화를 content의 변화없이 target speaker의 목소리로 변환시키는 것

(ex: 여성이 말한 내용을 남성voice로 말하게 함)

이때, one-shot conversion이란, 1source(여성) 와 1target(남성)이 매칭될 때, one-shot이라고 부른다.

1. [One-shot]논문 요약

Content와 speaker의 정보가 나눠진다고 판단한다. 각각의 인코더를 주고, 디코더에서 두 개를 받아들인다.

1. Conv 1d bank

One-shot conversion 맨 앞에서 사용되는데 이때 kernel 사이즈가 1 ~ 8로 이뤄진다. 그리고, 마지막에 dim-reduction을 위해서 conv1d가 사용된다. 이를 통해 long term 에 대해 데이터를 받아들이고자 하였다.

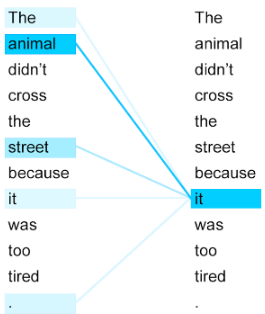
1. **One-shot**

하나의 example에 대해서

Source와 target이 존재한다고 이해하면 된다.

1. Multi-head-Attention & self attention

Encoder와 Decoder의 head들을 병렬적으로 구성하였다.



🡪 Self attention을 통해서 시간 축에 대한 병렬적인 수행이 가능하다. 가령, 기존의 RNN은 순차적으로 The, animal, … , tired까지 하나씩 이전 입력이 요구되었으나, transformer 같은 경우에는 it이라는 쿼리에 모든 입력값들이 대응된다. 이를 self-attention이라고 한다.

이때, 각각의 V, K, Q가 여러 개의 Head에 걸쳐져서 concat이 되고, Linear하게 만들어지는데, 이것이 바로 multi-head attention. 이때 head 한 개의 작동과정을 self-attention이라고 생각하면 되겠다.



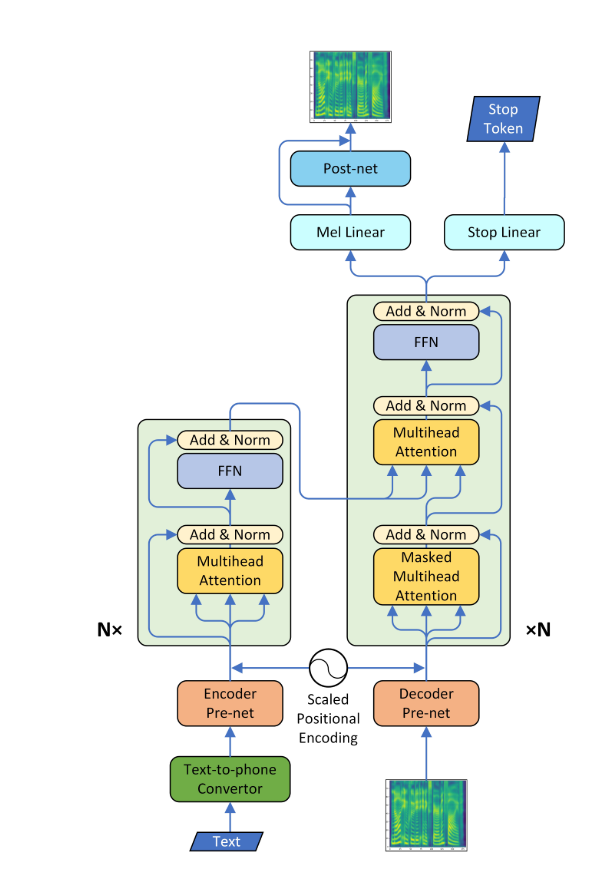
1. Masked-multi-head attention

Transformer에서 decoder에서 mask를 쓰는 이유는 미래 정보를 반영하지 않기 위함이라고 한다.

Q. 그런데 residual net을 통해서, 자기자신이Input으로 들어오기 때문에, 미래정보가 반영되는데 그러면 mask를 하는게 의미 없는게 아닌가?

A: Input에서 Mask를 취해주면, 가능하다.

1. Transformer의 구조



1. Heuristic이란 무엇인가??
2. Robust란 무엇인가?
3. Speech command recognition with torchaudio

: torchaudio에 있는 dataset에 대한 이용이 가능하다. 해당하는 음성데이터에 대해 trainloader를 만들고, target을 만들어서 이것이 어떤 범주에 속하는지 분석이 가능하다.