Recep Oğuz Araz
01.11.2020
Prof. Engin Erzin
ELEC390

Report VI: Making the Model Work

Batch Normalization (BatchNorm): Select a subset of non-linearities in the model and normalize the inputs to these non-linearities. Normalization refers to mean removal and unit variance imposing. (Ioffe et. al.) This will help model learn faster and eliminates the requirement for dropouts.

DS2: Inputs are normalized, Convolutional layer pre non-linearities are normalized and BiRNN input transformations are sequence-wise normalized. No information on FC layers.

- Tried various combinations including,
    - Only input normalization.
    - All non-linearities normalization. (Current try)
- Disadvantages:
    - Increased computation time for normalization.
    - 2*Input size parameters added for each BatchNorm
- Advantages:
    - CTC outputs not all-zero outputs!
    - Example:
      tfztftftcdüifiriririririfififififififufiuiuiuiuirfufufiuğublblblblblblflfczcücufafufırıtrtatıtnrnınririninininininti
      rtrtrutrıuauıaıtıtctctcacütutrtrıdrtatathcncğcğcğcğpğpğpğpğpğlplplsmuzvzvgkükgö ö ö ö ö ö

Update on CTC Loss: Following the BatchNorm,

- started observing not all zero outputs.
- decreasing loss over epochs.

However, the new model contains instabilities.

- NaN losses after a couple hundred iterations or couple of epochs. (zero_infinity parameter is not working as supposed to)
- Random outputs even after 3 epochs

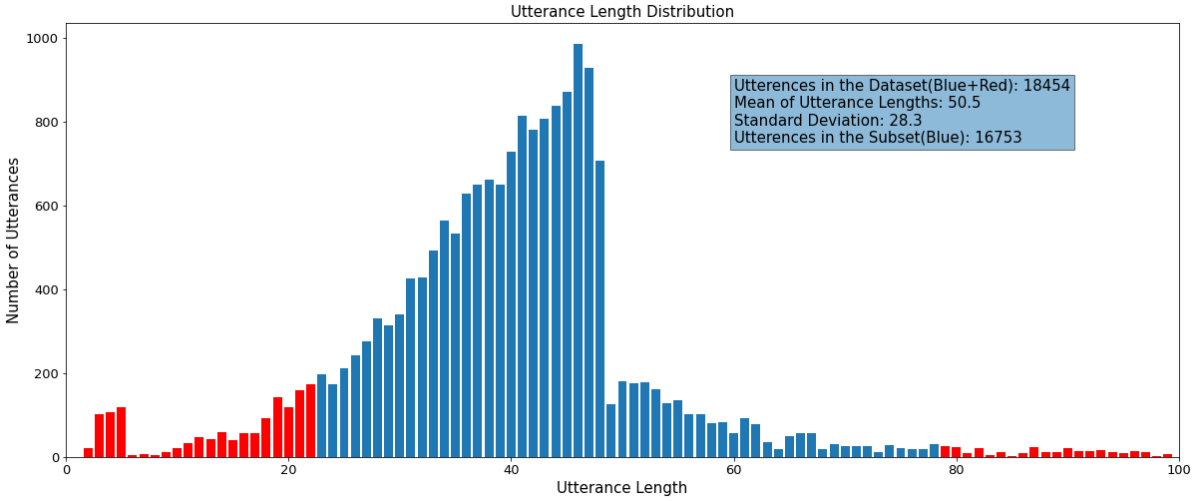
Infinite losses mainly occur when the input lengths are shorter than target lengths. (Source: PyTorch Discuss)

As we are zero padding extensively for batching various size utterances, this may create a problem.

- For example, shortest and longest utterances in the same batch

Recep Oğuz Araz
01.11.2020
Prof. Engin Erzin
ELEC390

Solution:



Use sentences that have length 1 standard deviation around the mean length.

With this new subset, the NaN problem stopped appearing. (It may be due to other changes?)

Subset will not cause any problems during evaluation.

- No batch during evaluation
- Sequence length not a model parameter
- Batch small and long utterances together respectively in the future
- When the CTC problems are solved, we can try the full dataset.

Language Based Spectrogram Calculations

DeepSpeech2 is Language agnostic in feature extraction.

We can go beyond and adapt our feature extraction process to the dynamics of Spoken Turkish.

By default, we were using:

- 512 N_{FFT} bins (257 non-negative bins as features).
- Hann window with 512 window length and 512/4 hop size

For 22050 sampling rate this corresponds to a 23msec window and a resulting wideband spectrogram.

We can use a narrowband spectrogram to achieve less dependent (maybe even less correlated) features.
Model is outputting grapheme transcriptions for each frame, by using Spoken Turkish dynamics. We can decrease training time.

Need a reference for research done on Spoken Turkish

One research claims Turkish syllables are spoken between 200-350msec.

Recep Oğuz Araz
01.11.2020
Prof. Engin Erzin
ELEC390

Another research claims that Turkish syllables contain 1-4 “sounds” with mainly 2-3 sounds.

- ⇨ 1 Turkish sounds duration is between 80msec-140msec. Much larger than current window length.

Our dataset: as people are reading written pieces, people are speaking fast. ➔ 50msec-100msec

So, we can double the frame length by first doubling the N_{FFT} calculations.

By counting, we see that,

$$\#total\ parameters\ \approx\ 2 * N_{FFT}^2$$

Decoding Model Outputs

Model outputs tensor of size $N_{alphabet}$ for each frame (for each example in N_{Batch}), of estimated probabilities (logsoftmax).

For now, we use argmax for decoding the outputs and remove repeating characters. (Best Path Decoding)

0ej0ej0j0jl0e0e0j0ilj0'lflmpmpmpmpmps

Beam Search is on the way.