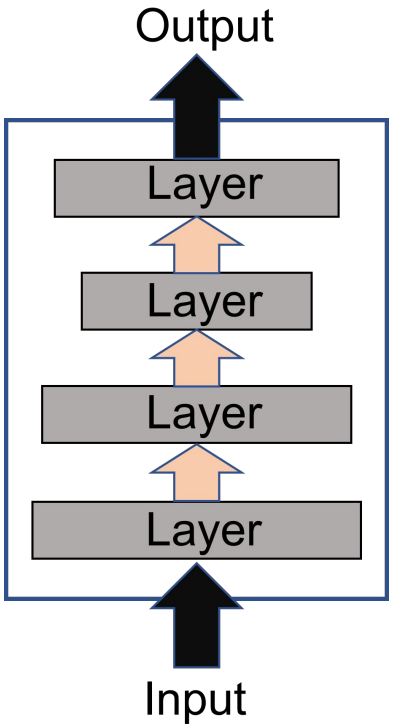
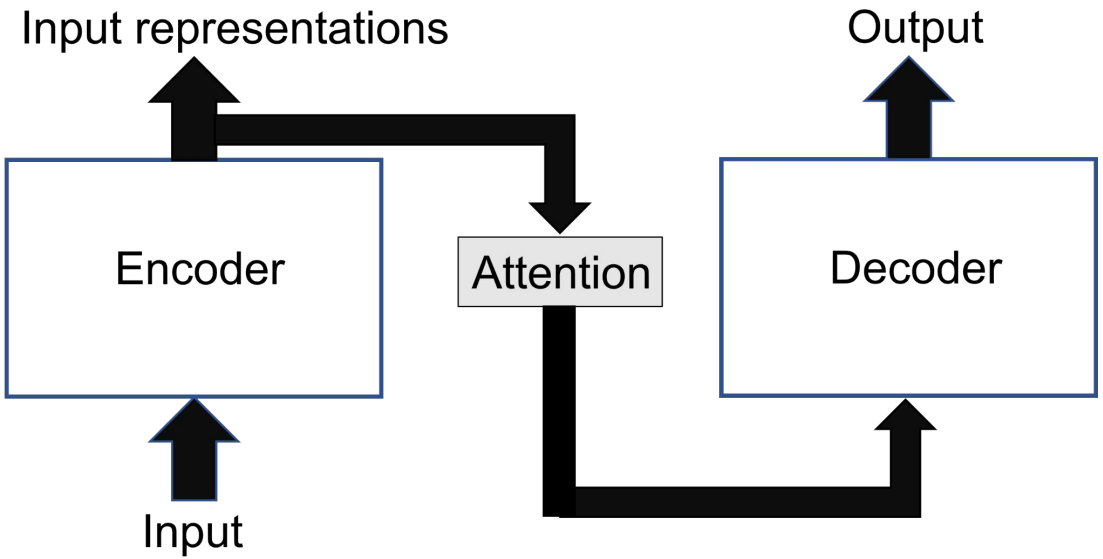
HW4P2 - Boot Camp

Swathi Jadav, Moayad Elamin

Attention-based End-to-End Speech-to-Text Deep Neural Network

- Implement an attention-based system to solve a sequence-to-sequence problem.





Encoder

- An encoder mainly derives feature vectors from the input sequence of speech vectors.
- The input speech feature vectors have - strong structural continuity with adjacent vectors , as well as longer-term Contextual dependencies.
- They usually contain CNN’s and RNN’s.

Decoder

- A decoder mainly uses feature vectors produced by the encoder to output a probability distribution over the output sequence.

Attention

- Each output character relates to some portion of the original input audio.
- To generate each output, the decoder receives, as input, a “context”, comprising a weighted sum of the sequence of representation vectors computed by the encoder. The weights with which the vectors are combined must be such that the context pays most “attention” to the most relevant part of the input.

Baseline Architecture

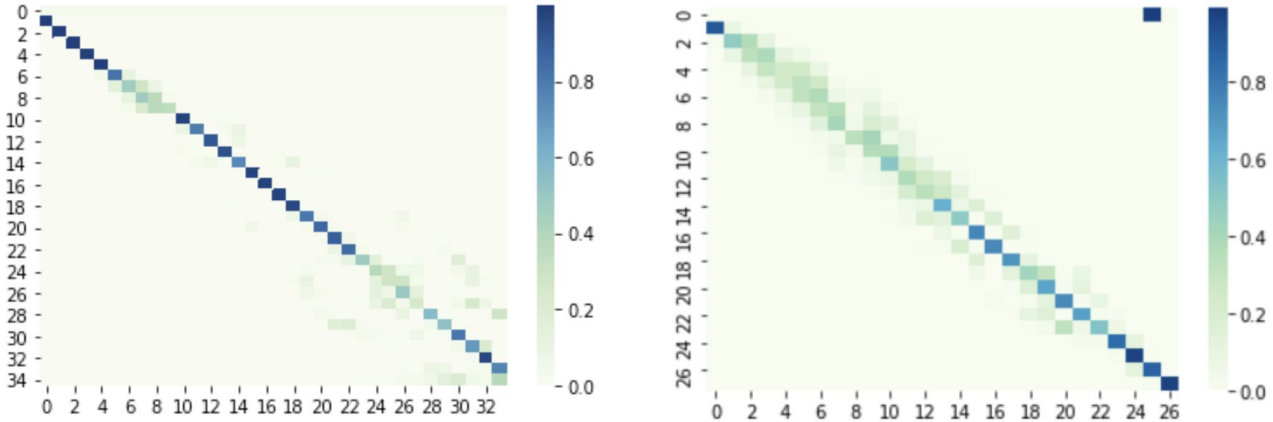
- The baseline architecture for this homework is the ***"Listen, Attend and Spell"***, *William Chan, Navdeep Jaitly,Quoc V. Le, Oriol Vinyals.*
- Get Started Early !!!

Starter Notebook

- Dataset
- Dataloader
- Model:
  - Implementing the baseline model
  - Encoder (Listener) : CNN’s, LSTM, pBLSTM
  - Attention Mechanism (Attend): Single-head Attention, Multi-head Attention
  - Decoder (Speller): LSTM Architecture
- Training
- Inference

Things to keep in mind….

- Use the Toy dataset : Make sure the attention plot is diagonal
- Use Mixed Precision Training for improving the training time per epoch
- Two ways of Transcription : Word Based and Character Based. We suggest using Character based
- Processing the Transcripts : Use the **Vocab_Map** for index to char and Vice-versa
- Use the Built-In Pytorch libraries for Pad Packed sequence and Pack Pad Sequence (Use Enforce-Sorted =False if you don't want to sort the inputs every batch)
- Try Using CNN's instead of pBLSTM - **don't downsample below a factor of 8**



Things to keep in mind….

- Single-Head Attention
- Multi-Head Attention
- Inference: select the most probable output sequence
  - Greedy Search
  - Random Search
  - Beam Search -iteratively expands out the K most probable paths
- Levenshtein distance : Don't use <EOS> and <SOS>
- Weight Initialization Techniques
- Regularization - Dropouts, Locked Dropouts, Weight Tying, Embedding Dropout
- Data Augmentation : Time and Frequency Masking
- Optimizers: Adam and AdamW , Weight Decay
- LR : 1e-3 Try different schedulers
- Pre-Train the Listener : Auto encoder architecture, helps faster training
- Pre- Train the speller : train transcripts as a train data and train your model like you would train a language model (e.g. HW4P1)