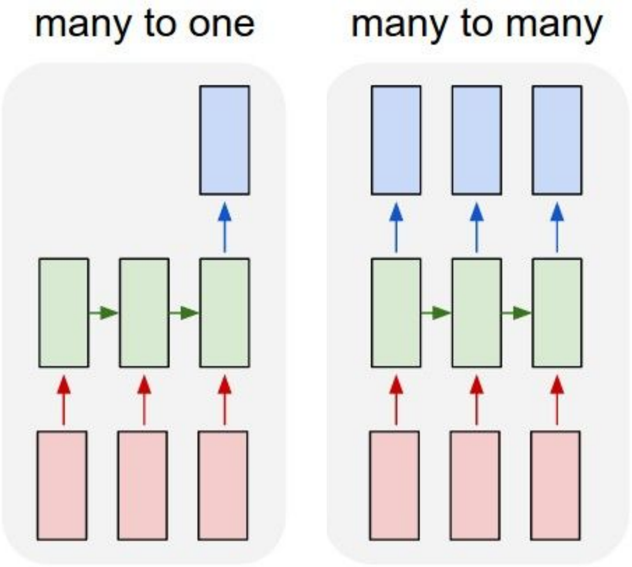
HW3 Bootcamp: P2

Kinori Rosnow

Data and Task

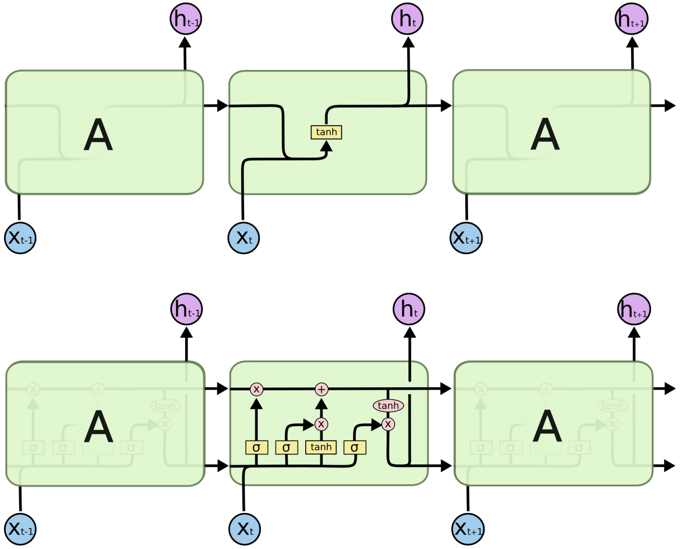
- Features:
  - Same as hw1p2 (array of utterances with 40D timesteps)
  - No P2 multiple choice
- Labels:
  - Order? Alignment?
  - List of lists - cast to numpy array
- Must generate sequences of phonemes
  - 41 phonemes and 1 blank character
- Loss: CTCLoss
- Metric: mean Levenshtein distance
  - Can import



https://i.stack.imgur.com/b4sus.jpg

Modeling and RNN+Variants

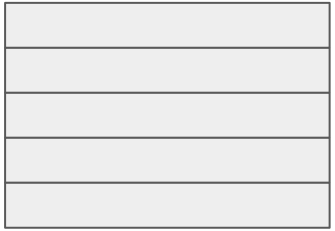
- Can use other types of layers
  - Hint: Convolutional Layers
  - Input shape: (batch, in_channel, length)
- RNN, LSTM, GRU, etc.
  - Capture sequential dependencies
  - Input shape: (length, batch, feature) or (batch, length, feature)
- No attention



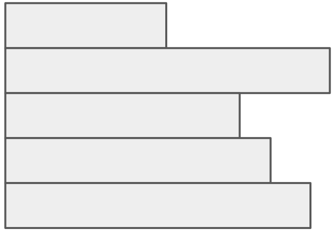
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Batch of Variable Length Inputs: Padding

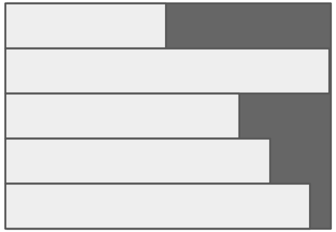
Equal Length



Variable Length

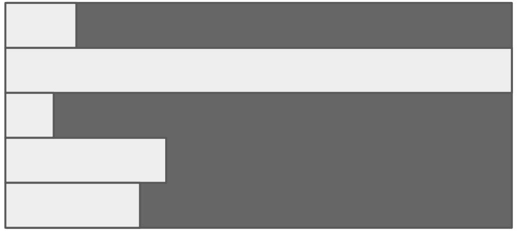


Padded Equal Length

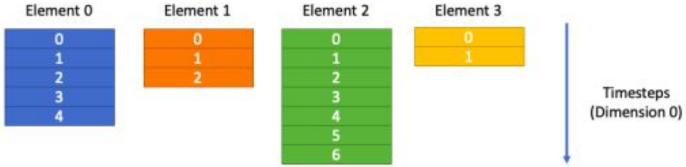


- HW1, HW2: equal length inputs
- HW3: variable length sequences
- Method 1: Pad
  - Inefficient with space
- Method 2: Packing

Another Problematic Example

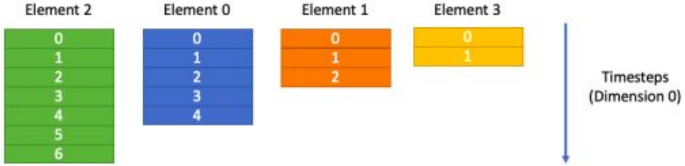


Batch of Variable Length Inputs: Packed Sequence



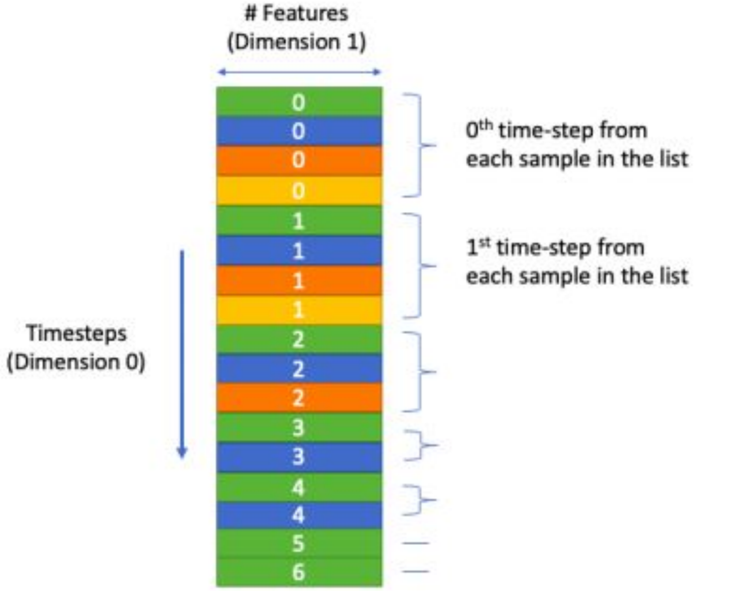
List of Tensors to be packed. Each has same number of features but different time steps.

Figure 2: List of tensors we want to pack



Tensors sorted in descending order based on the number of time steps in each sample.

Figure 3: First we sort the list in a descending order based on number of timesteps in each


# Features
(Dimension 1)
0
0
0
0
1
1
1
1
2
2
2
3
3
4
4
5
6
0th time-step from each sample in the list
1st time-step from each sample in the list
Timesteps
(Dimension 0)

Final 2d Packed Tensor

Figure 4: Final Packed 2d Tensor

Reference: Fall 20 HW3P1 Writeup

Packed Sequences

- pad_sequence()
  - Pads to equal length for batching
- pack_padded_sequence()
  - Packs batch of padded sequences
  - Requires sequences + sequence lengths
- X = pad_packed_sequence()
  - Unpacks back to a batch of padded sequences
  - Outputs sequences + sequence lengths
- Collate Function
  - Dataloader argument
  - Helpful when altering data for batch

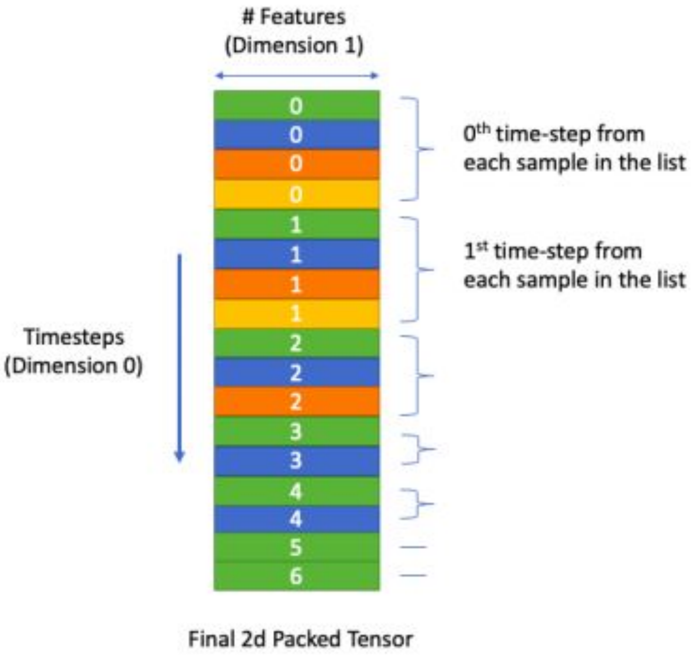
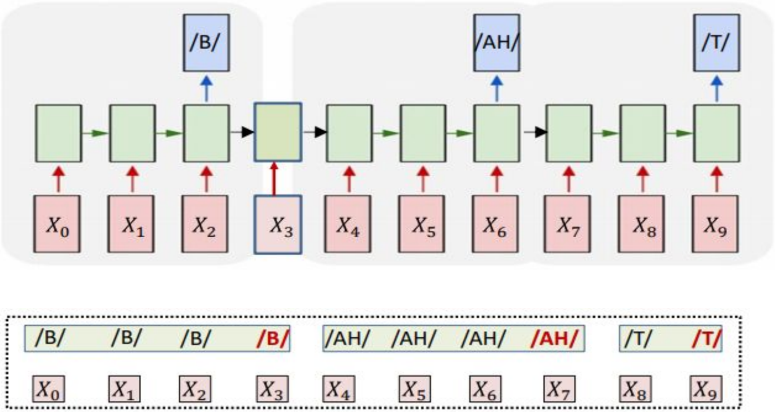

# Features
(Dimension 1)
0
0
0
0
1
1
1
1
2
2
2
3
3
4
4
5
6
0th time-step from each sample in the list
1st time-step from each sample in the list
Timesteps
(Dimension 0)
Final 2d Packed Tensor

Figure 4: Final Packed 2d Tensor

Output Processing

Output = probability distribution at each timestep

- Order Synchronous, not time synchronous
- Greedy Search
- Beam Search
  - Marginal performance improvement
  - Linked in the writeup
- Feel free to use your own!


/B/
/AH/
/T/
X_0 X_1 X_2 X_3 X_4 X_5 X_6 X_7 X_8 X_9
/B/ /B/ /B/ /B/
/AH/ /AH/ /AH/ /AH/
/T/ /T/
X_0 X_1 X_2 X_3 X_4 X_5 X_6 X_7 X_8 X_9

Advice

- Watch lecture and recitations
- Read the entire write-up and understand the problem before starting
- Look at the example submission
- Check tensor shapes
  - batch_first=True
- Use what you’ve learned from previous P2s

Some Helpful References

- PyTorch documentation — PyTorch 1.8.0 documentation - https://pytorch.org/docs/stable/index.html
- Homework_3_1.pdf (cmu.edu) - http://deeplearning.cs.cmu.edu/F20/document/homework/Homework_3_1.pdf
- http://colah.github.io/posts/2015-08-Understanding-LSTMs/