Intent classifier and slot tagger

Intent classifier

- What you can do:
  - Any model on BOW with n-grams and TF-IDF
  - RNN (LSTM, GRU, …)
  - CNN (1D convolutions)

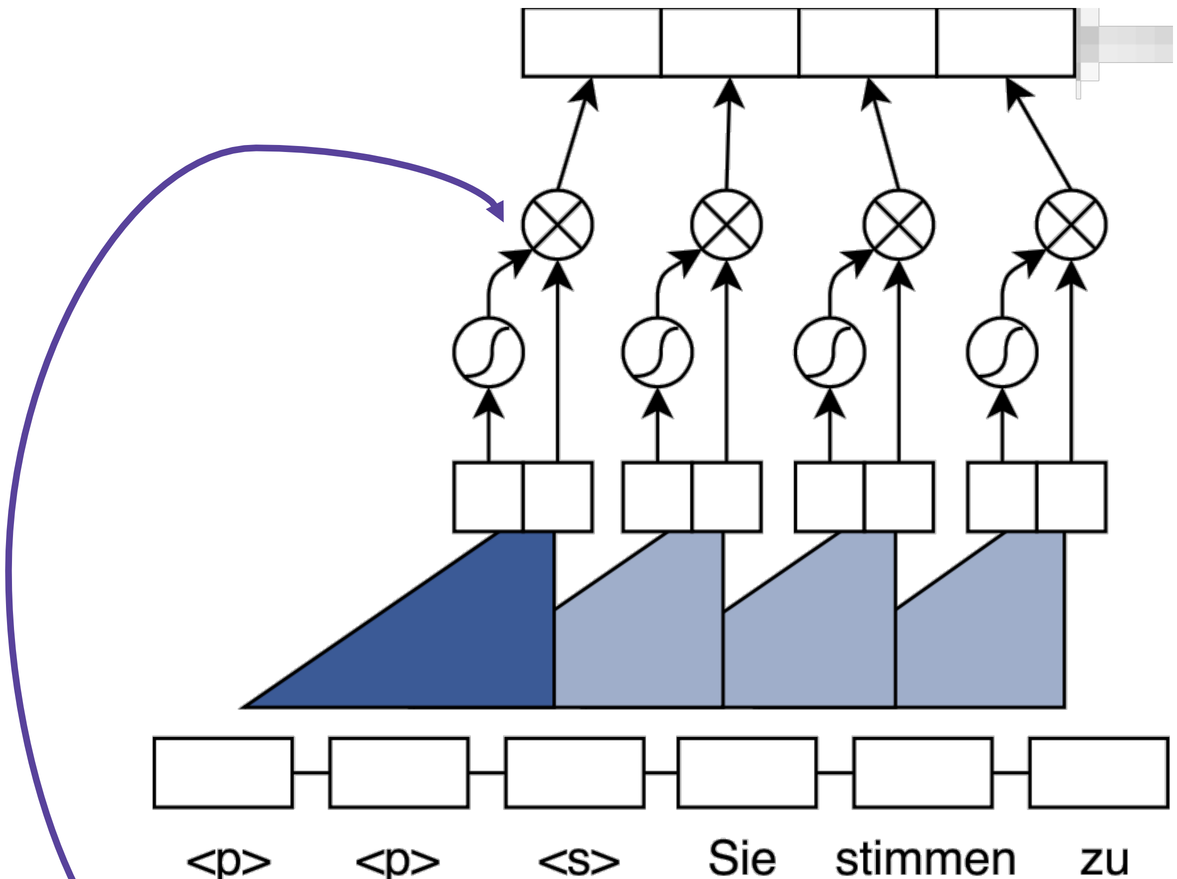
- CNNs can perform better on datasets where the task is essentially a key phrase recognition task as in some sentiment detection datasets.

https://arxiv.org/pdf/1702.01923.pdf

Slot tagger

- What you can do:
  - Handcrafted rules like regular expressions
  - CRF
  - RNN seq2seq
  - **CNN seq2seq**
  - Any seq2seq with attention

CNN for sequences: Gated Linear Unit



Stacking 6 layers with kernel size 5 results in an input field of 25 elements

$$h_l(\mathbf{X}) = (\mathbf{X} * \mathbf{W} + \mathbf{b}) \otimes \sigma(\mathbf{X} * \mathbf{V} + \mathbf{c})$$

https://arxiv.org/pdf/1612.08083.pdf https://arxiv.org/pdf/1705.03122.pdf

CNN for sequences: results

- They can sometimes beat LSTM in **language modeling**:

Model	Test PPL	Hardware
LSTM-1024 (Grave et al., 2016b)	48.7	1 GPU
GCNN-8	44.9	1 GPU
GCNN-14	37.2	4 GPUs

Table 3. Results for single models on the WikiText-103 dataset.

- … and **machine translation**:

WMT'14 English-French	**BLEU**
Wu et al. (2016) GNMT (Word 80K)	37.90
Wu et al. (2016) GNMT (Word pieces)	38.95
Wu et al. (2016) GNMT (Word pieces) + RL	39.92
ConvS2S (BPE 40K)	40.51

https://arxiv.org/pdf/1612.08083.pdf https://arxiv.org/pdf/1705.03122.pdf

CNN for sequences: speed benefit

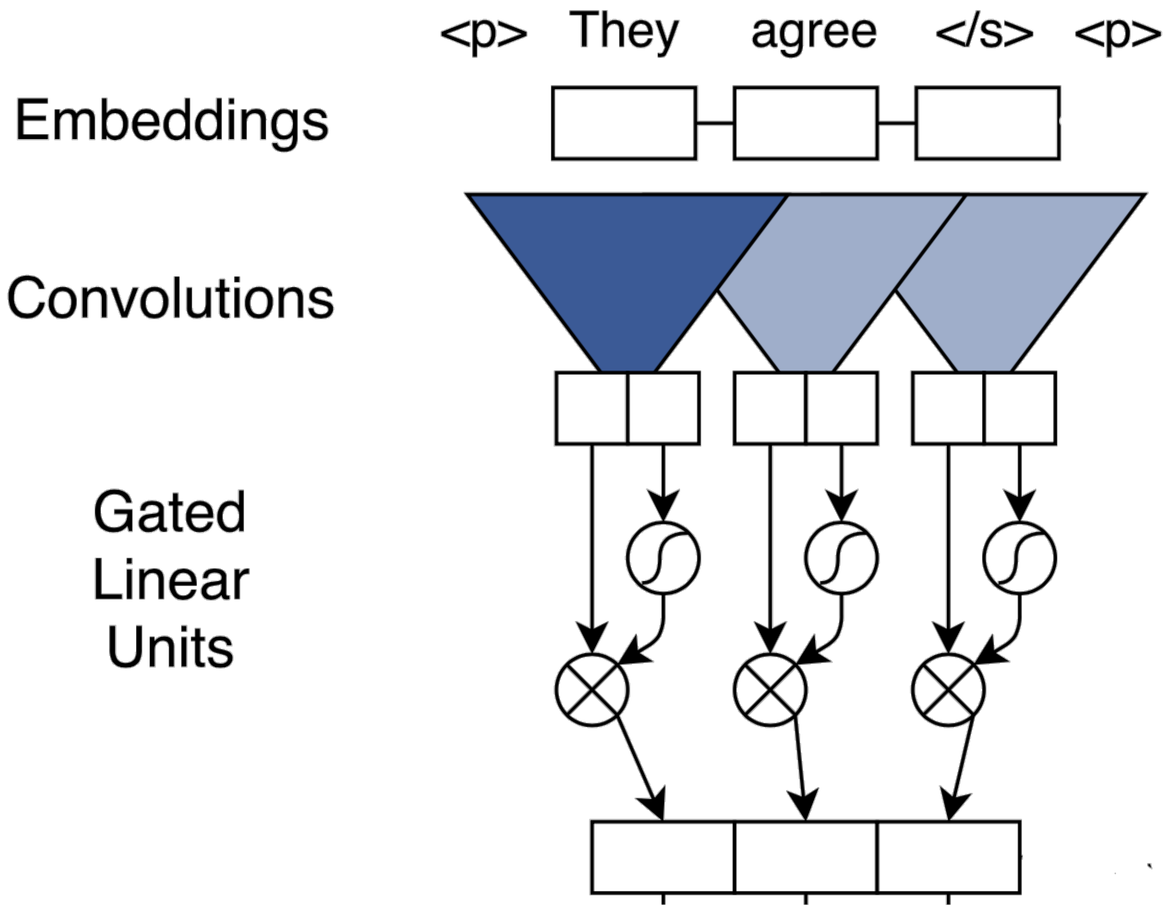
- They work faster than RNN:
  - During **training** we can process all time steps in parallel
  - During **testing** encoder can do the same
  - During **testing** we get higher throughput thanks to convolution optimizations in GPUs

	BLEU	Time (s)
GNMT GPU (K80)	31.20	3,028
GNMT CPU 88 cores	31.20	1,322
GNMT TPU	31.21	384
ConvS2S GPU (K40) $b = 1$	33.45	327
ConvS2S GPU (M40) $b = 1$	33.45	221
ConvS2S GPU (GTX-1080ti) $b = 1$	33.45	142
ConvS2S CPU 48 cores $b = 1$	33.45	142

Translation generation speed during testing

https://arxiv.org/pdf/1705.03122.pdf

CNN for sequences: how encoder looks like



- Bi-directional encoder is easy
- Works in parallel for all time steps

https://arxiv.org/pdf/1705.03122.pdf

ATIS dataset

- Airline Travel Information System
- Collected in 90s
- 4978 context independent utterances
- 17 intents, 127 slot labels
- State-of-the-art: 1.79% intent error, 95.9 slots F1

Utterance	show	flights	from	Seattle	to	San	Diego	tomorrow
Slots	O	O	O	B-fromloc	O	B-toloc	I-toloc	B-depart_date
Intent	Flight							

http://www.isca-speech.org/archive/Interspeech_2016/pdfs/1352.PDF

https://www.microsoft.com/en-us/research/wp-content/uploads/2010/12/SLT10.pdf

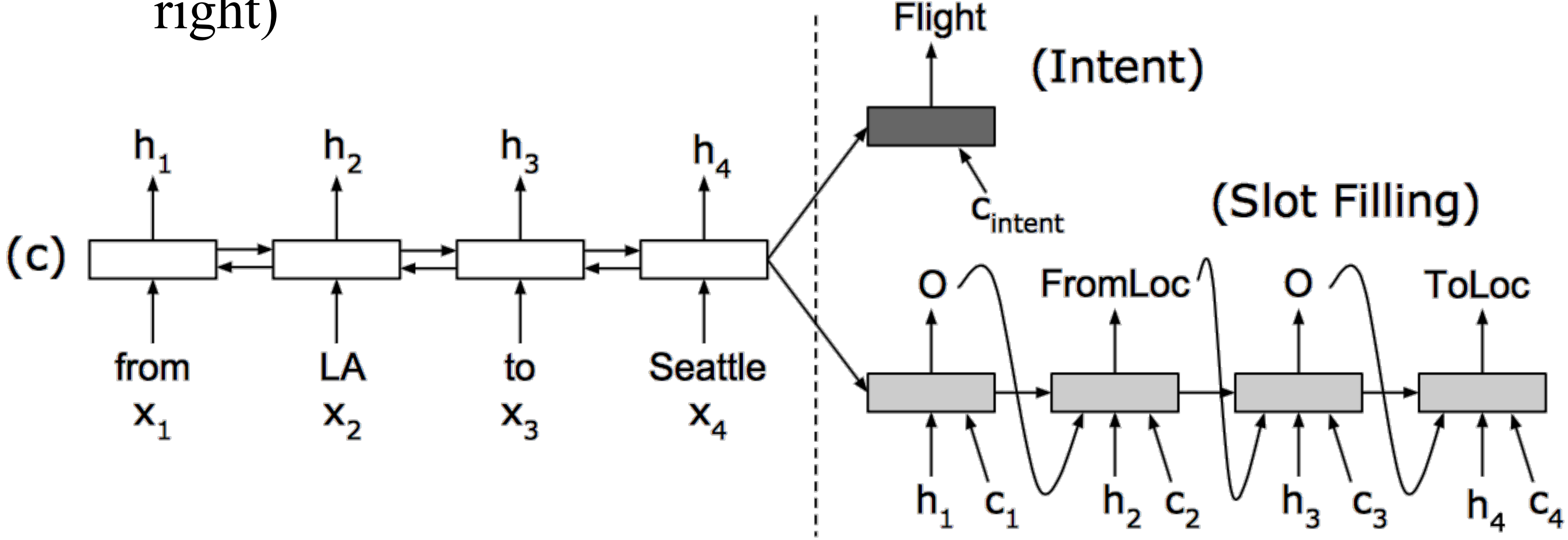
Joint training of intent classifier and slot tagger

- They both analyze the same sequence
- What if we learn representations suitable for both tasks?
- That results in more supervision and higher quality of both

http://www.isca-speech.org/archive/Interspeech_2016/pdfs/1352.PDF

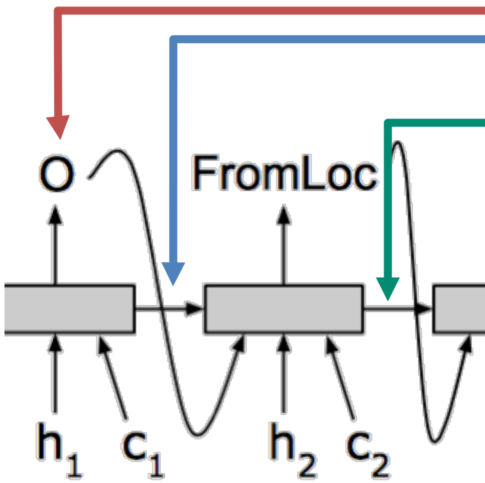
Joint training of intent classifier and slot tagger

- Encoder-decoder architecture for joint intent detection and slot filling
- Encoder is a bi-directional LSTM
- With aligned inputs (h_i on the right) and attention (c_i on the right)



http://www.isca-speech.org/archive/Interspeech_2016/pdfs/1352.PDF

Attention in decoder



Decoder state s_i: $s_i = f(s_{i-1}, y_{i-1}, h_i, c_i)$

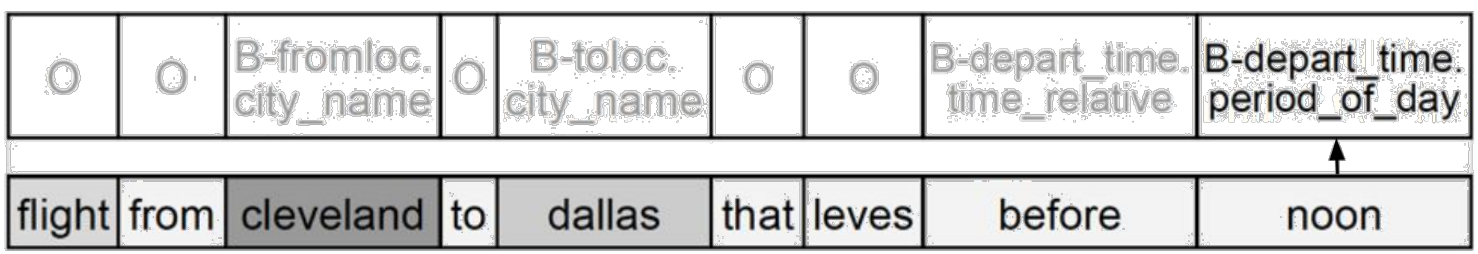
Attention c_i (g is a feed forward network):

$$c_i = \sum_{j=1}^{T} \alpha_{i,j} h_j \quad \alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k=1}^{T} \exp(e_{i,k})}$$

$$e_{i,k} = g(s_{i-1}, h_k)$$

Attention weights (the darker the higher) when predicting the slot label for the last word "noon":

$\alpha_{T,j}$

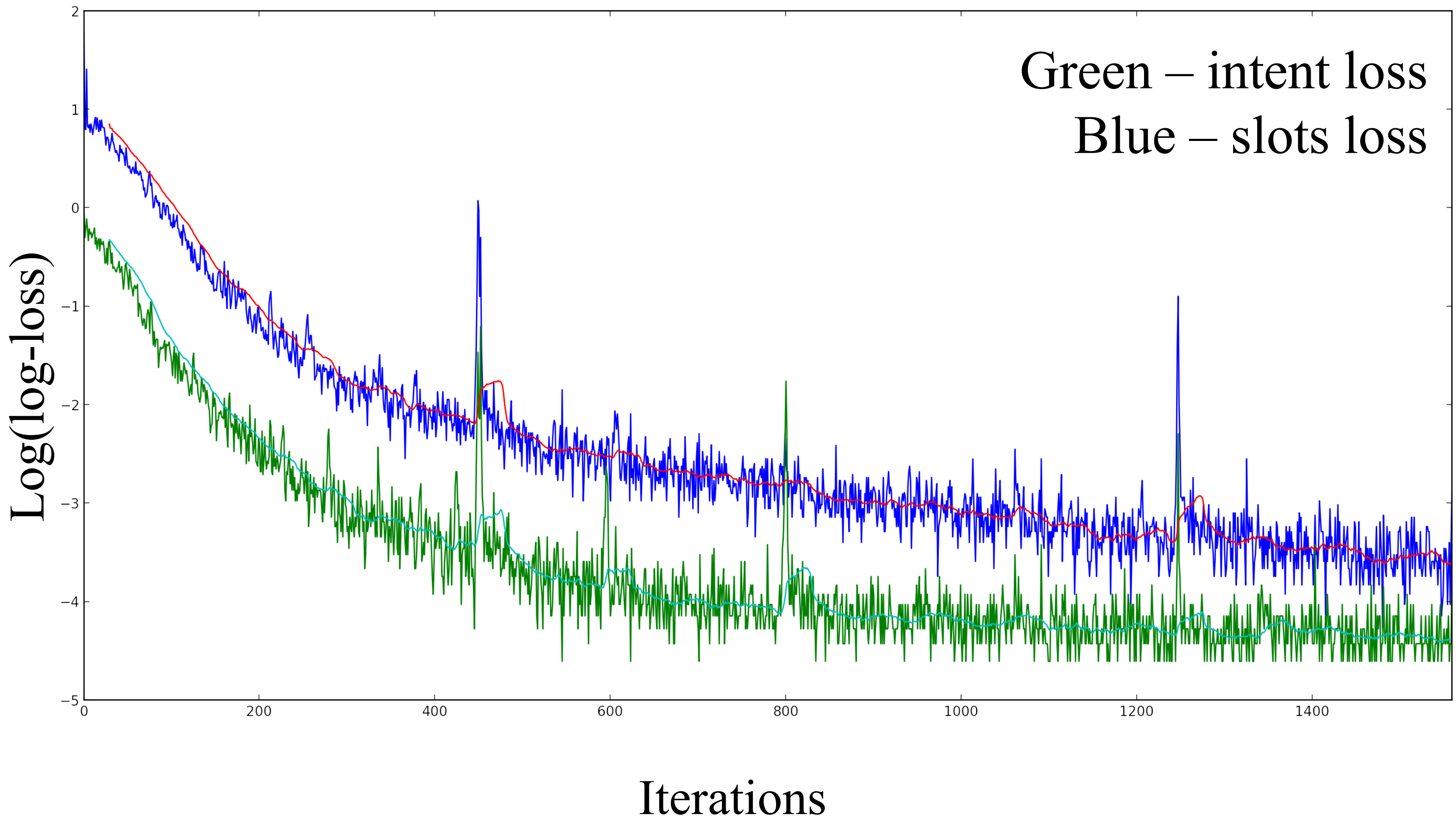


T

http://www.isca-speech.org/archive/Interspeech_2016/pdfs/1352.PDF

Joint training loss

- Final training loss is a sum of losses for intent and slots



Joint training results

- Better performance on ATIS dataset:

Training	Slots F1	Intent % error
Independent training for slot filling	95.78	-
Independent training for intent detection	-	2.02
Joint training for slot filling and intent detection	**95.87**	**1.57**

- Works faster than two separate models

http://www.isca-speech.org/archive/Interspeech_2016/pdfs/1352.PDF

Summary

- We’ve overviewed different options for intent classifier and slot tagger training
- People start to use CNN for sequence modeling and sometimes get better results than with RNN
- Joint training can be beneficial in terms of speed and performance
- In the next video we’ll take a look at context utilization in our NLU (intent classifier and slot tagger)