10-418/10-618 Machine Learning for Structured Data

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Learning to Search

Matt Gormley
Lecture 4
Sep. 12, 2022

Reminders

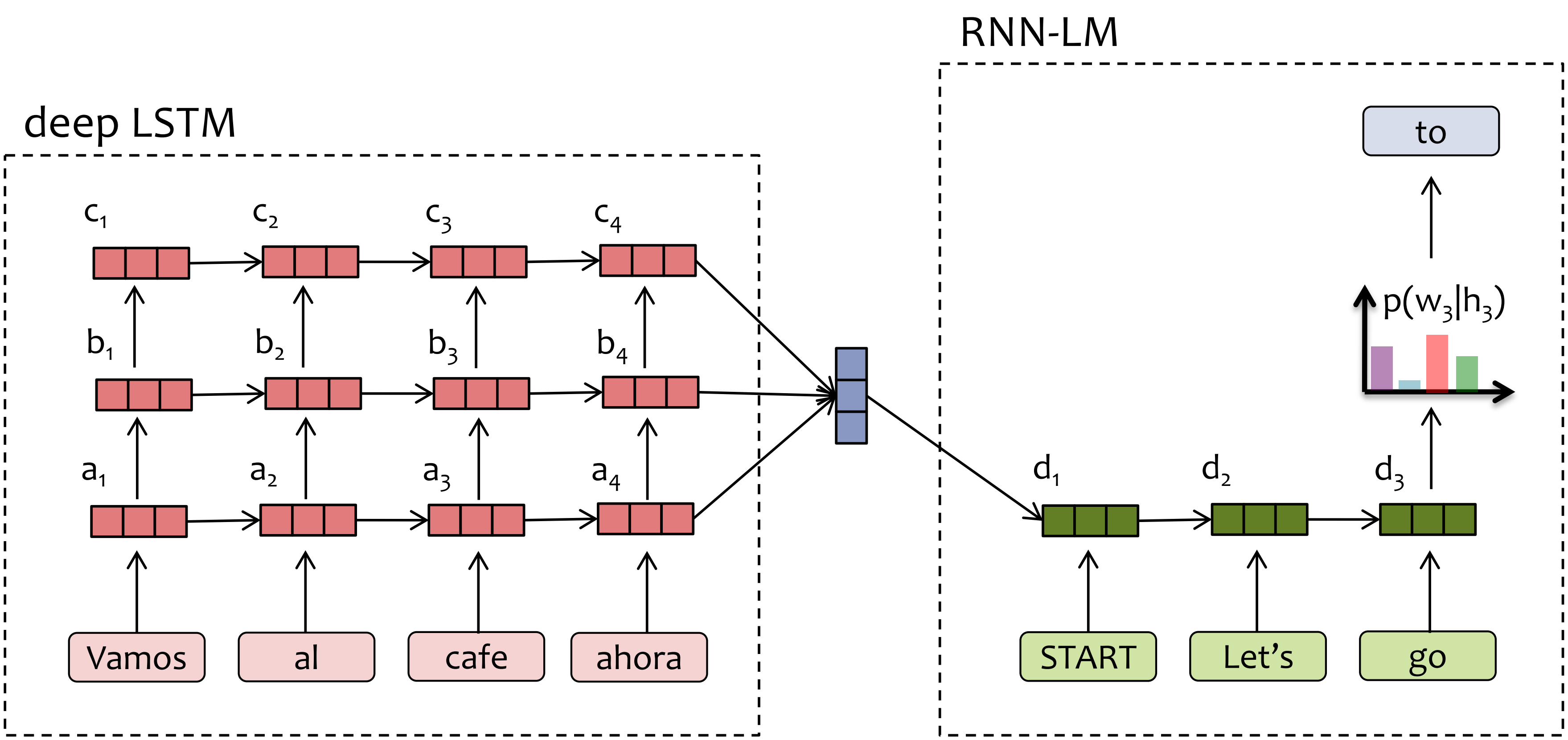
- **Homework 1: Neural Networks for Sequence Tagging**
  - **Out: Wed, Sep 7** (later today!)
  - **Due: Fri, Sep 16 at 11:59pm**
  - Two parts:
    1. written part to Gradescope (Written slot)
    2. programming part to Gradescope (Programming slot)

EXAMPLE SEQ2SEQ ARCHITECTURES

Example Architectures

deep LSTM + RNN-LM

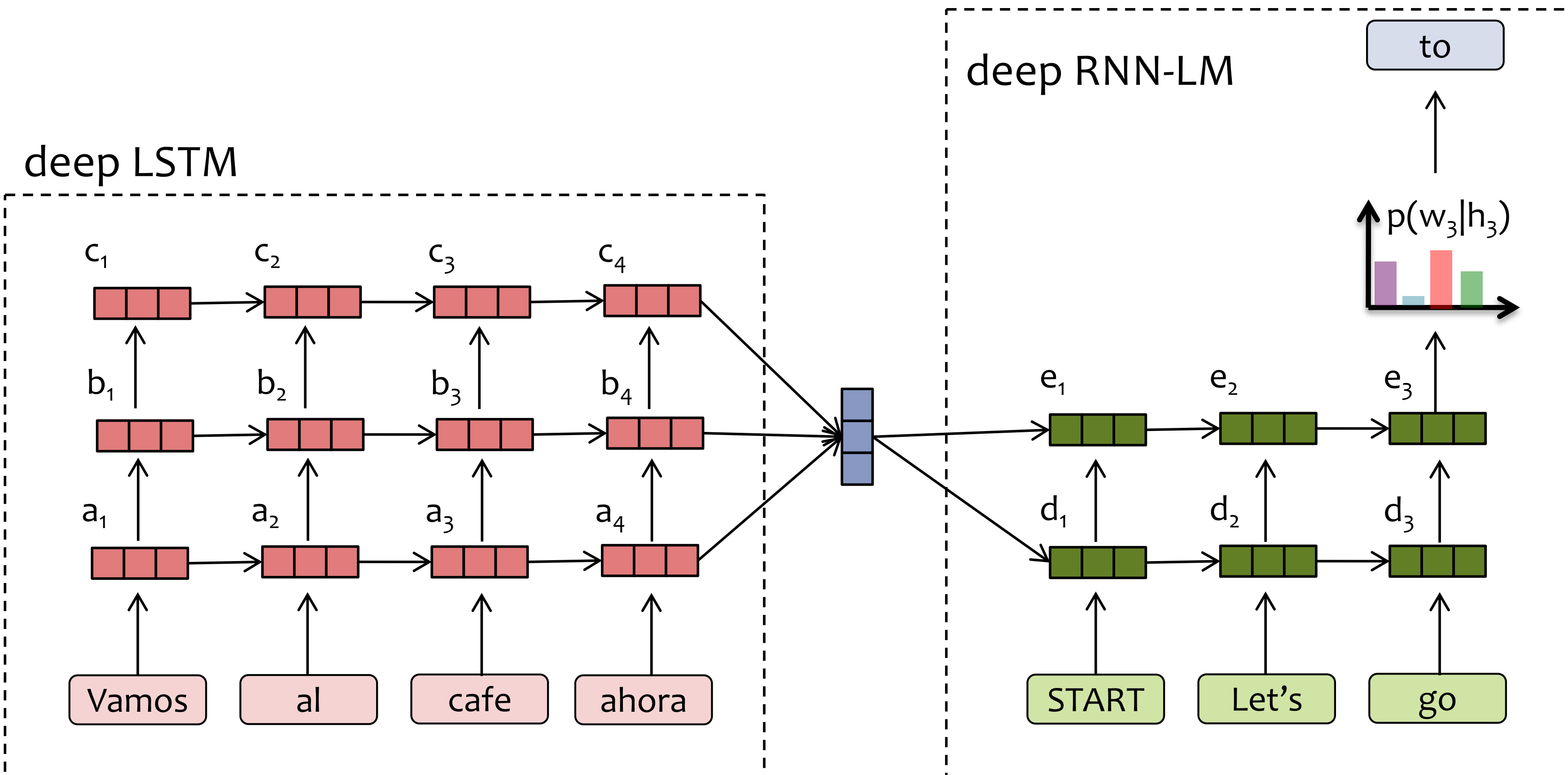
- *Encoder*: three-layer unidirectional LSTM
- *Decoder*: a one-layer RNN-LM



Example Architectures

deep LSTM + deep RNN-LM

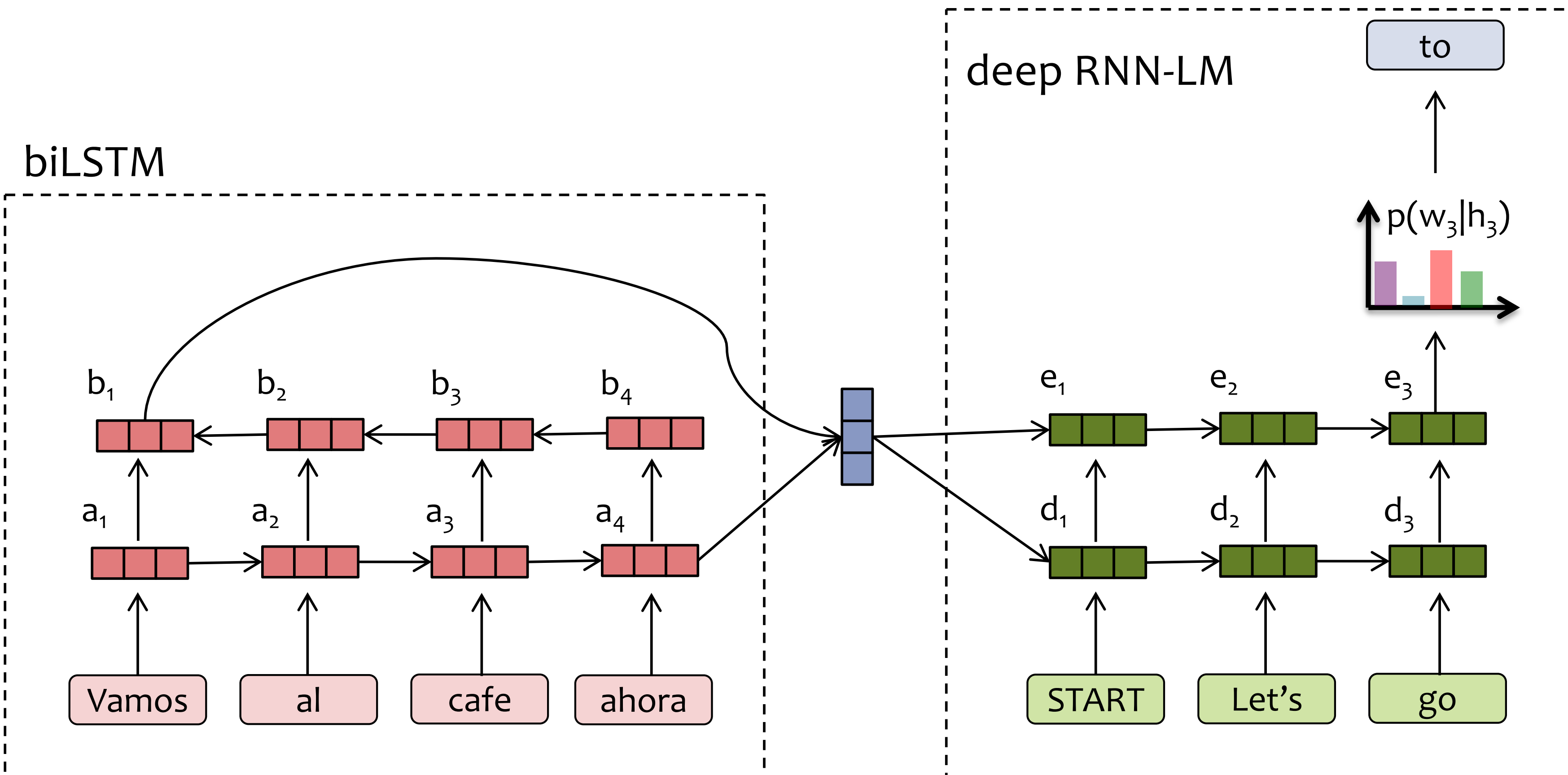
- *Encoder*: three-layer unidirectional LSTM
- *Decoder*: a two-layer RNN-LM



Example Architectures

biLSTM + deep RNN-LM

- *Encoder*: two-layer bidirectional LSTM
- *Decoder*: a two-layer RNN-LM



LEARNING A SEQ2SEQ MODEL

Comparing RNN, RNN-LM, seq2seq

Whiteboard:

- Objective functions for RNN, RNN-LM, and seq2seq models
- Training a seq2seq model

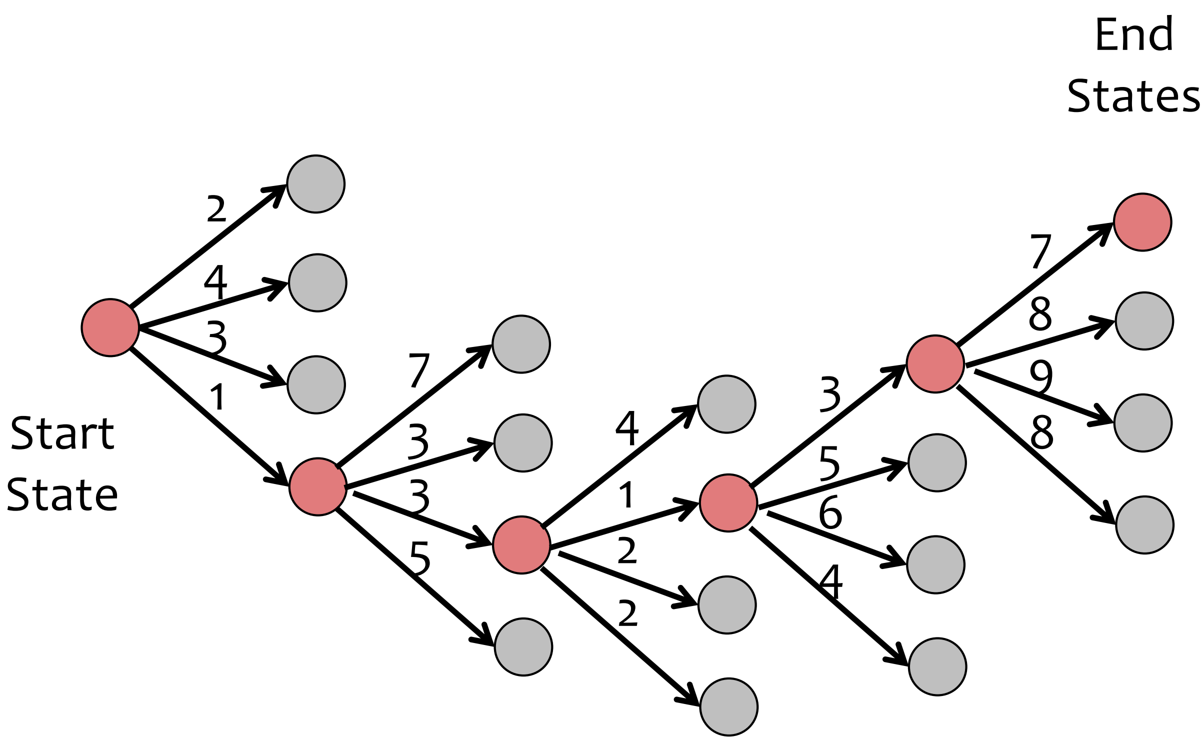
DECODING FOR SEQ2SEQ MODELS

Decoding for seq2seq Models

At test time, how do we obtain predictions from our model?

- The two most common approaches:
  - Greedy search
  - Beam search
- Many alternatives:
  - Ancestral sampling (assuming we have a locally normalized model)
  - Nucleus sampling
  - Top-k sampling

Background: Greedy Search



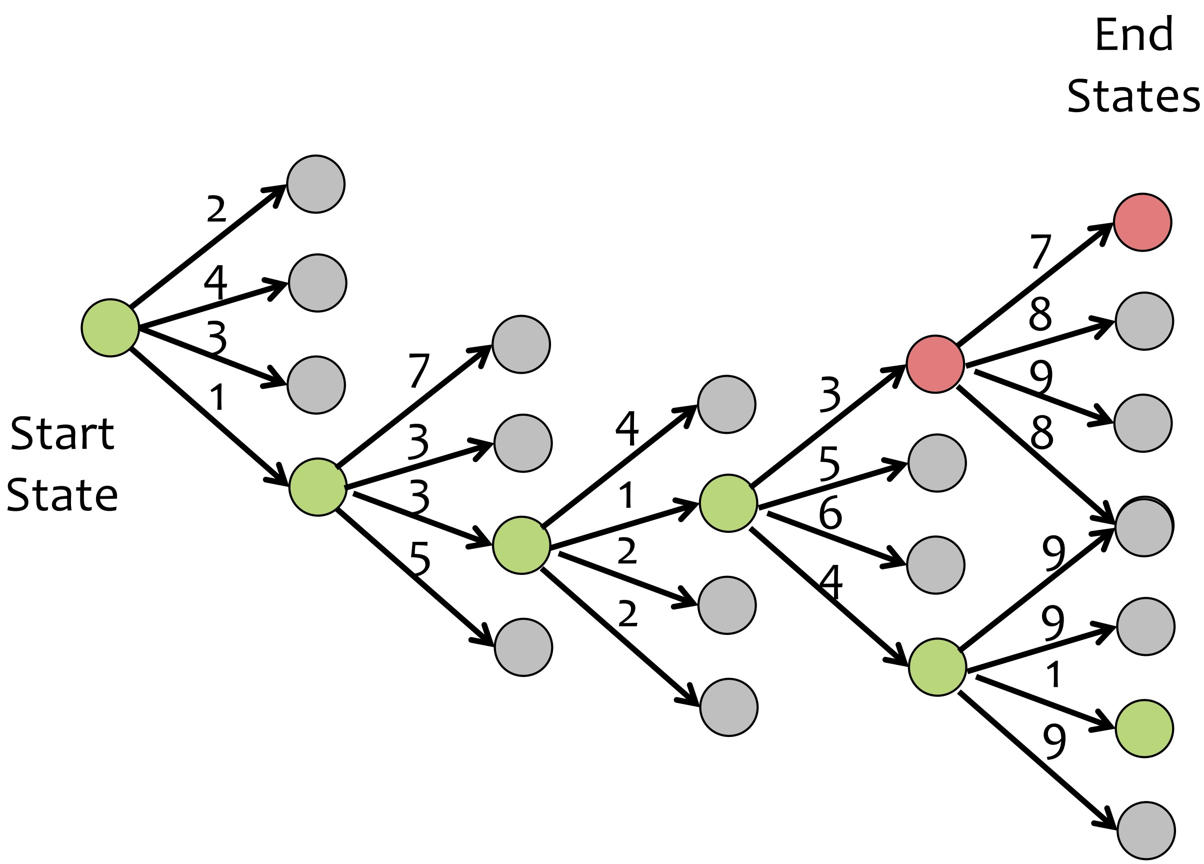
Goal:

- Search space consists of nodes and weighted edges
- Goal is to find the lowest (total) weight path from root to a leaf

Greedy Search:

- At each node, selects the edge with lowest (immediate) weight
- **Heuristic** method of search (i.e. does *not* necessarily find the best path)
- Computation time: **linear** in max path length

Background: Greedy Search



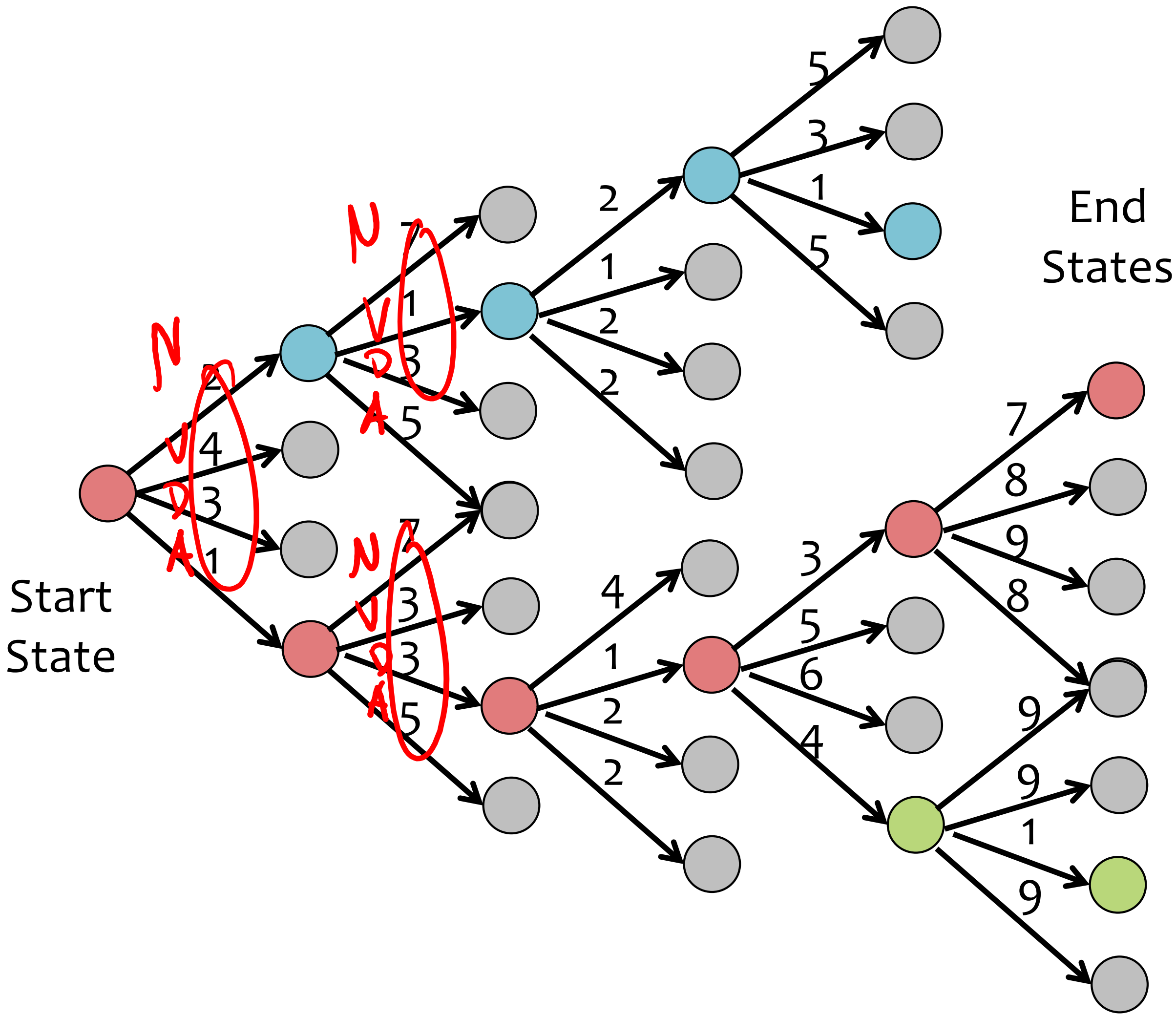
Goal:

- Search space consists of nodes and weighted edges
- Goal is to find the lowest (total) weight path from root to a leaf

Greedy Search:

- At each node, selects the edge with lowest (immediate) weight
- **Heuristic** method of search (i.e. does *not* necessarily find the best path)
- Computation time: **linear** in max path length

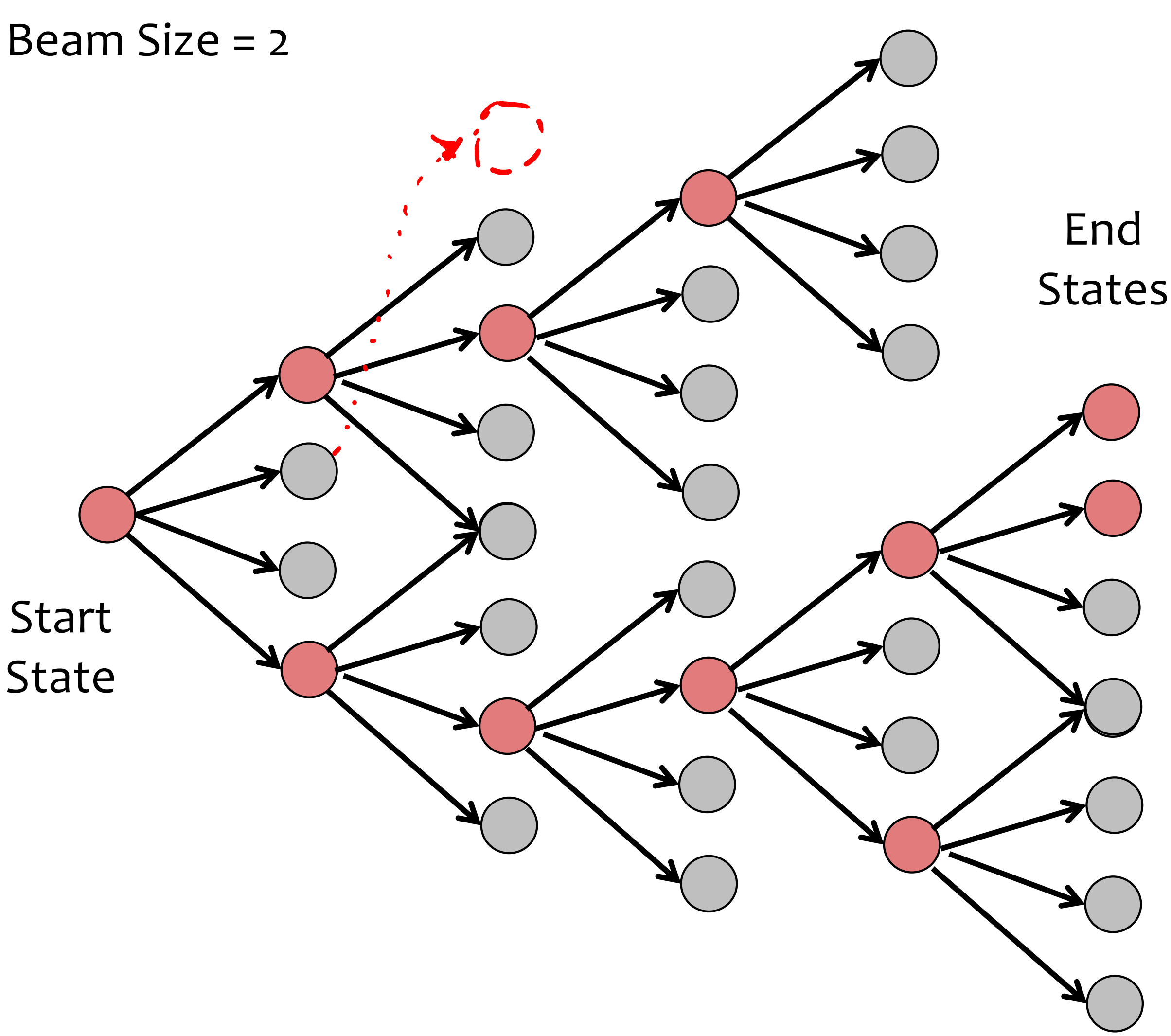
Background: Greedy Search



Goal:
- Search space consists of nodes and weighted edges
- Goal is to find the lowest (total) weight path from root to a leaf

Greedy Search:
- At each node, selects the edge with lowest (immediate) weight
- **Heuristic** method of search (i.e. does *not* necessarily find the best path)
- Computation time: **linear** in max path length

Background: Beam Search



Goal:

- Search space consists of nodes and weighted edges
- Goal is to find the lowest (total) weight path from root to a leaf

Beam Search:

- The "beam" is current set of best k nodes
- Let the expansion set be all neighbors of nodes in the beam
- At each time step, selects the set of k nodes in the expansion set with lowest (immediate) weight
- **Heuristic** method of search (i.e. does *not* necessarily find the best path)
- Computation time: **linear** in max path length

Decoding for seq2seq Models

At test time, how do we obtain predictions from our model?

- The two most common approaches:
  - Greedy search
  - Beam search
- Many alternatives:
  - Ancestral sampling (assuming we have a locally normalized model)
  - Nucleus sampling
  - Top-k sampling

Important Observation

- maximum likelihood training (MLE) **assumes** that our inference strategy will return the **highest probability sequence**
- at **test time,** our **inference** strategies are all **heuristic** (i.e. they will make mistakes)

APPLICATIONS OF SEQ2SEQ

seq2seq for MT

Basic Architecture:

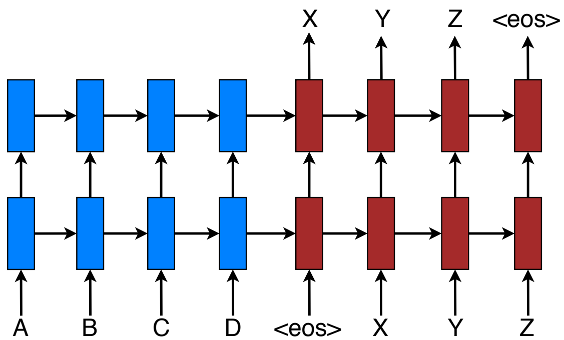


Figure 1: **Neural machine translation** – a stacking recurrent architecture for translating a source sequence A B C D into a target sequence X Y Z. Here, <eos> marks the end of a sentence.

Results from Sutskever et al. (2014)

Method	test BLEU score (ntst14)
Bahdanau et al. [2]	28.45
Baseline System [29]	33.30
Single forward LSTM, beam size 12	26.17
Single reversed LSTM, beam size 12	30.59
Ensemble of 5 reversed LSTMs, beam size 1	33.00
Ensemble of 2 reversed LSTMs, beam size 12	33.27
Ensemble of 5 reversed LSTMs, beam size 2	34.50
Ensemble of 5 reversed LSTMs, beam size 12	**34.81**

Table: performance on WMT'14 English to French test set

Visualization from Sutskever et al. (2014)

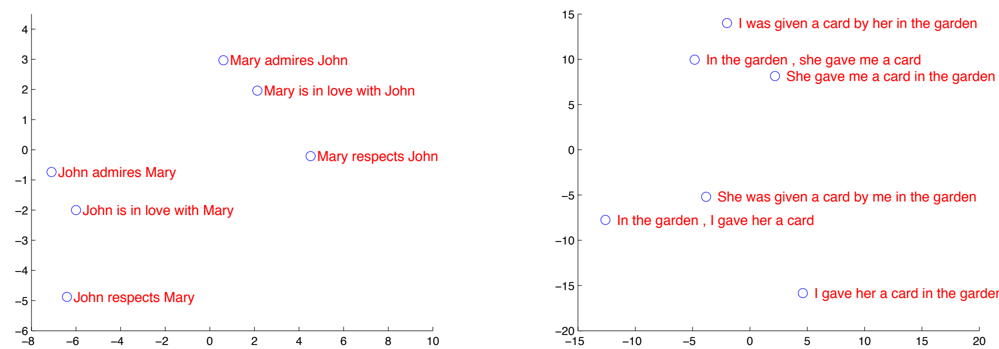


Figure 2: The figure shows a 2-dimensional PCA projection of the LSTM hidden states that are obtained after processing the phrases in the figures. The phrases are clustered by meaning, which in these examples is primarily a function of word order, which would be difficult to capture with a bag-of-words model. Notice that both clusters have similar internal structure.

Figure from Luong et al. (2015)

seq2seq for ASR

Listen Attend and Spell

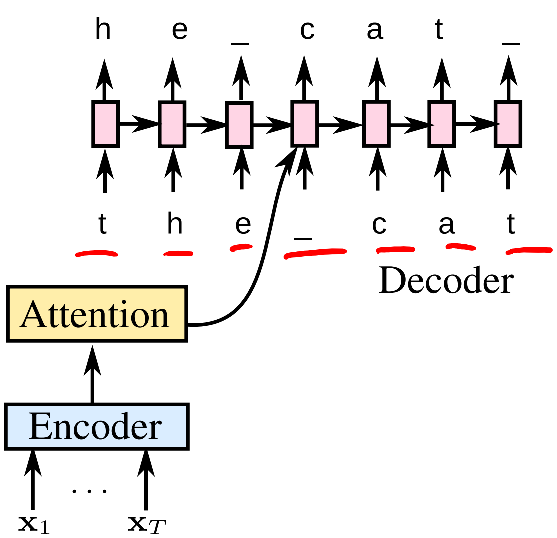


Figure 1: *LAS model.*

$$\mathbf{h} = \text{Listen}(\mathbf{x})$$
$$P(y_i|\mathbf{x}, y_{<i}) = \text{AttendAndSpell}(y_{<i}, \mathbf{h})$$

Figure from Irie et al. (2019)

seq2seq for ASR

Listen Attend and Spell

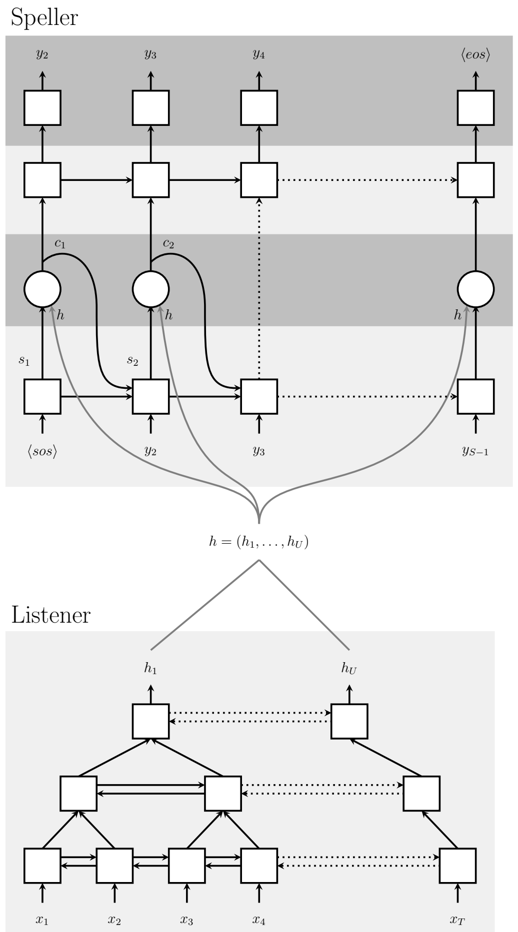


Fig. 1: Listen, Attend and Spell (LAS) model: the listener is a pyramidal BLSTM encoding our input sequence $\mathbf{x}$ into high level features $\mathbf{h}$, the speller is an attention-based decoder generating the $\mathbf{y}$ characters from $\mathbf{h}$.

Figure from Chan et al. (2016)

Results from Park et al. (2019)

Table 3: *LibriSpeech 960h WERs (%).*

Method	**No LM**		**With LM**	
	clean	**other**	**clean**	**other**
HMM				
Panayotov et al., (2015) [19]			5.51	13.97
Povey et al., (2016) [29]			4.28	
Han et al., (2017) [30]			3.51	8.58
Yang et al. (2018) [31]			2.97	7.50
CTC/ASG				
Collobert et al., (2016) [32]	7.2			
Liptchinsky et al., (2017) [33]	6.7	20.8	4.8	14.5
Zhou et al., (2018) [34]			5.42	14.70
Zeghidour et al., (2018) [35]			3.44	11.24
Li et al., (2019) [36]	3.86	11.95	2.95	8.79
LAS				
Zeyer et al., (2018) [23]	4.87	15.39	3.82	12.76
Zeyer et al., (2018) [37]	4.70	15.20		
Irie et al., (2019) [24]	4.7	13.4	3.6	10.3
Sabour et al., (2019) [38]	4.5	13.3		

seq2seq for ASR

Listen Attend and Spell

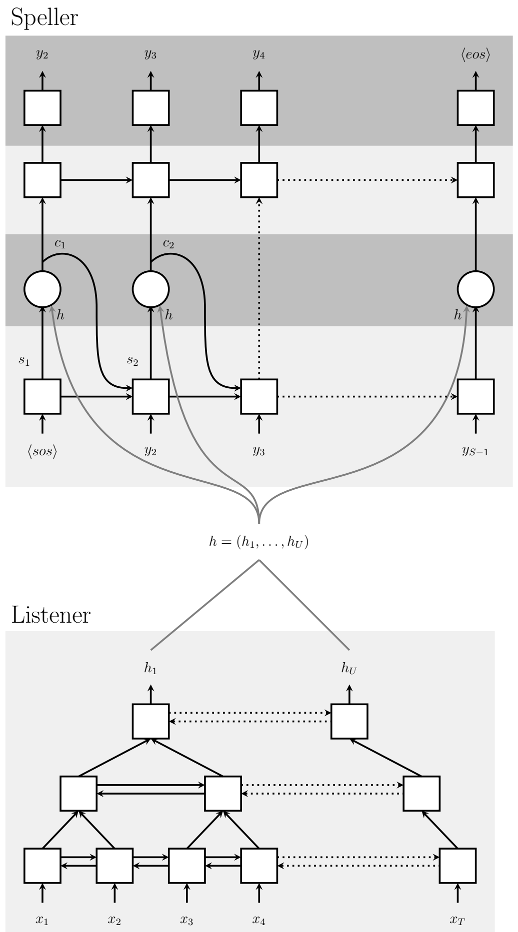


Fig. 1: Listen, Attend and Spell (LAS) model: the listener is a pyramidal BLSTM encoding our input sequence **x** into high level features **h**, the speller is an attention-based decoder generating the **y** characters from **h**.

Figure from Chan et al. (2016)

Results from Park et al. (2019)

Table 3: *LibriSpeech 960h WERs (%).*

Method	No LM clean	No LM other	With LM clean	With LM other
HMM				
Panayotov et al., (2015) [19]			5.51	13.97
Povey et al., (2016) [29]			4.28	
Han et al., (2017) [30]			3.51	8.58
Yang et al. (2018) [31]			2.97	7.50
CTC/ASG				
Collobert et al., (2016) [32]	7.2			
Liptchinsky et al., (2017) [33]	6.7	20.8	4.8	14.5
Zhou et al., (2018) [34]			5.42	14.70
Zeghidour et al., (2018) [35]			3.44	11.24
Li et al., (2019) [36]	3.86	11.95	2.95	8.79
LAS				
Zeyer et al., (2018) [23]	4.87	15.39	3.82	12.76
Zeyer et al., (2018) [37]	4.70	15.20		
Irie et al., (2019) [24]	4.7	13.4	3.6	10.3
Sabour et al., (2019) [38]	4.5	13.3		
Our Work				
LAS	4.1	12.5	3.2	9.8
LAS + SpecAugment	**2.8**	**6.8**	**2.5**	**5.8**

Park et al. (2019) used the **LAS model** from prior work, and introduced a **data augmentation** method that gave state-of-the-art performance on LibriSpeech 960h and Switchboard 300h tasks

Image Captioning

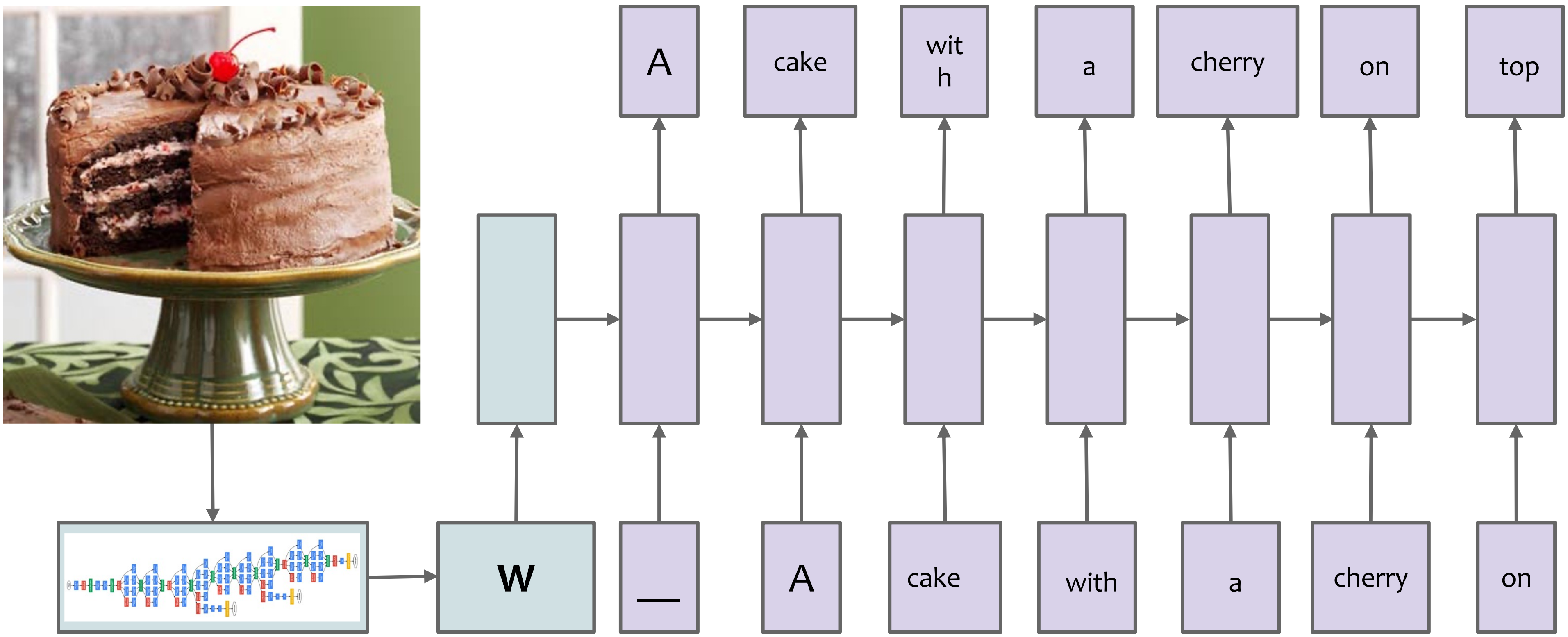
$$p(\text{English} \mid \text{French})$$

$$\downarrow$$

$$p(\text{English} \mid \text{Image})$$

1. Vinyals, O., et al. "Show and Tell: A Neural Image Caption Generator." *CVPR* (2015).
2. Mao, J., et al. "Deep captioning with multimodal recurrent neural networks (m-rnn)." *ICLR (2015).*
3. Karpathy, A., Li, F., "Deep visual-semantic alignments for generating image descriptions." *CVPR (2015).*

Image Captioning



$$\theta^\star = \arg\max_\theta p(S|I)$$

Slide from Vinyals & Jaitly (ICML Tutorial, 2017)

Image Captioning

Human: A close up of two bananas with bottles in the background.

BestModel: A bunch of bananas and a bottle of wine.

InitialModel: A close up of a plate of food on a table.

Slide from Vinyals & Jaitly (ICML Tutorial, 2017)

Image Captioning

Human: A woman holding up a yellow banana to her face.

BestModel: A woman holding a banana up to her face.

InitialModel: A close up of a person eating a hot dog.

Slide from Vinyals & Jaitly (ICML Tutorial, 2017)

Image Captioning

Human: A man outside cooking with a sub in his hand.

BestModel: A man is holding a sandwich in his hand.

InitialModel: A man cutting a cake with a knife.

Slide from Vinyals & Jaitly (ICML Tutorial, 2017)

Image Captioning

Human: Someone is using a small grill to melt his sandwich.

BestModel: A person is cooking some food on a grill.

InitialModel: A pizza sitting on top of a white plate.

Slide from Vinyals & Jaitly (ICML Tutorial, 2017)

Image Captioning

Human: A blue , yellow and red train travels across the tracks near a depot.

BestModel: A blue and yellow train traveling down train tracks.

InitialModel: A train that is sitting on the tracks.

Slide from Vinyals & Jaitly (ICML Tutorial, 2017)

Q1: What questions do you have?

Learning Objectives

418.mlcourse.org

Sequence to Sequence Models

618.mlcourse.org

You should be able to…

1. Apply an RNN to time-series structured prediction tasks
2. Employ an RNN-LM for various structured prediction tasks through prompting
3. Explain the difference between RNNs, RNNLMs, encoder-decoder models, and seq2seq models
4. Implement and train a basic seq2seq model

IMITATION LEARNING

Imitation Learning vs. RL

Paradigm	**Data**
Supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N \qquad \mathbf{x} \sim p^*(\cdot)$ and $y = c^*(\cdot)$
$\hookrightarrow$ Regression	$y^{(i)} \in \mathbb{R}$
$\hookrightarrow$ Classification	$y^{(i)} \in \{1, \ldots, K\}$
$\hookrightarrow$ Binary classification	$y^{(i)} \in \{+1, -1\}$
$\hookrightarrow$ Structured Prediction	$\mathbf{y}^{(i)}$ is a vector
Unsupervised	$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N \qquad \mathbf{x} \sim p^*(\cdot)$
Semi-supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N_1} \cup \{\mathbf{x}^{(j)}\}_{j=1}^{N_2}$
Online	$\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), (\mathbf{x}^{(3)}, y^{(3)}), \ldots\}$
Active Learning	$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$ and can query $y^{(i)} = c^*(\cdot)$ at a cost
Imitation Learning	$\mathcal{D} = \{(s^{(1)}, a^{(1)}), (s^{(2)}, a^{(2)}), \ldots\}$
Reinforcement Learning	$\mathcal{D} = \{(s^{(1)}, a^{(1)}, r^{(1)}), (s^{(2)}, a^{(2)}, r^{(2)}), \ldots\}$

Autonomous Driving via Imitation Learning

- *Goal*: learn to **drive a car** around a **dirt track** at **high speed** without crashing
- *Approach 1*: (Williams et al., 2016; 2017)
  - model-predictive control (MPC)
  - expensive, accurate sensors required:
    - Global Positioning System (GPS)
    - Inertial Measurement Unit (IMU)
  - effective, but limited applicability
- *Approach 2*: (Pan et al., 2018)
  - imitation learning with deep CNN defining the policy
  - low-cost, on-board sensors:
    - monocular camera
    - wheel speed sensors
  - learn from expert demonstrations to reduce risk of crash

Figure from Pan et al. (2018)

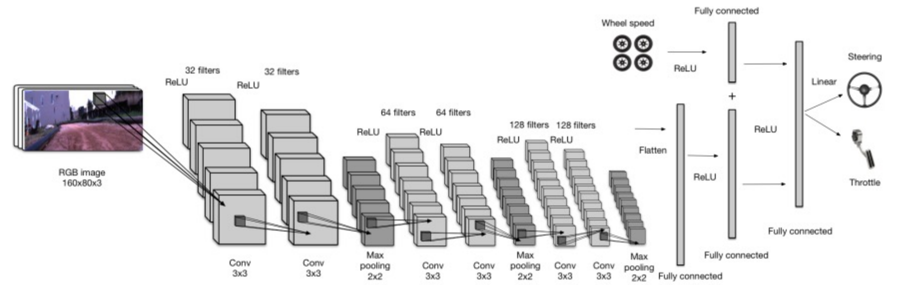
Autonomous Driving via Imitation Learning

Autonomous Driving via Imitation Learning

Imitation Learning

state (sensors)

policy (neural network)



action (left / right)

agent (car)

Figures from Pan et al. (2018)

Imitation Learning

Whiteboard:

– Fully supervised imitation learning
– The pitfall of fully supervised imitation learning
– DAgger for imitation learning