Language Models

Model the **probability of a sentence**; ideally be able to sample plausible sentences

*Slides from Louis-Philippe Morency*
Language Models

Model the **probability of a sentence**; ideally be able to sample plausible sentences

Why is this useful?

* Slides from Louis-Philippe Morency
Language Models

Model the **probability of a sentence**; ideally be able to sample plausible sentences

Why is this useful?

\[
\text{arg max } P(\text{wordsequence} | \text{acoustics}) = \frac{P(\text{acoustics} | \text{wordsequence}) \times P(\text{wordsequence})}{P(\text{acoustics})}
\]

\[
\text{arg max } P(\text{acoustics} | \text{wordsequence}) \times P(\text{wordsequence})
\]

* Slides from Louis-Philippe Morency
Language Models

Model the **probability of a sentence**; ideally be able to sample plausible sentences

Why is this useful?

\[
\begin{align*}
\arg\max_{\text{wordsequence}} P(\text{wordsequence} | \text{acoustics}) &= \\
\arg\max_{\text{wordsequence}} & \frac{P(\text{acoustics} | \text{wordsequence}) \times P(\text{wordsequence})}{P(\text{acoustics})}
\end{align*}
\]

* Slides from Louis-Philippe Morency
Simple **Language Models**: N-Grams

Given a word sequence: \( w_{1:n} = [w_1, w_2, ..., w_n] \)

We want to estimate \( p(w_{1:n}) \)

* Slides from Louis-Philippe Morency
Simple **Language Models**: N-Grams

Given a word sequence: \( w_{1:n} = [w_1, w_2, ..., w_n] \)

We want to estimate \( p(w_{1:n}) \)

Using **Chain Rule** of probabilities:

\[
p(w_{1:n}) = p(w_1)p(w_2|w_1)p(w_3|w_1, w_2) \cdots p(w_n|w_{1:n-1})
\]
Simple Language Models: N-Grams

Given a word sequence: \( w_{1:n} = [w_1, w_2, \ldots, w_n] \)

We want to estimate \( p(w_{1:n}) \)

Using **Chain Rule** of probabilities:

\[
p(w_{1:n}) = p(w_1)p(w_2|w_1)p(w_3|w_1, w_2) \cdots p(w_n|w_{1:n-1})
\]

**Bi-gram Approximation:**

\[
p(w_{1:n}) = \prod_{k=1}^{n} p(w_k|w_{k-1})
\]

**N-gram Approximation:**

\[
p(w_{1:n}) = \prod_{k=1}^{n} p(w_k|w_{k-N+1:k-1})
\]

* Slides from Louis-Philippe Morency*
Estimating **Probabilities**

N-gram conditional probabilities can be estimated based on raw concurrence counts in the observed sequences

**Bi-gram:**

\[
p(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}
\]

**N-gram:**

\[
p(w_n|w_{n-N-1:n-1}) = \frac{C(w_{n-N-1:n-1}w_n)}{C(w_{n-N-1:n-1})}
\]

* Slides from Louis-Philippe Morency
Neural-based Unigram Language Mode

P(next word is “dog”)  
Neural Network  
1-of-N encoding of “START”

P(next word is “on”)  
Neural Network  
1-of-N encoding of “dog”

P(next word is “the”)  
Neural Network  
1-of-N encoding of “on”

P(next word is “beach”)  
Neural Network  
1-of-N encoding of “the”

* Slides from Louis-Philippe Morency
Neural-based Unigram Language Mode

Problem: Does not model sequential information (too local)

* Slides from Louis-Philippe Morency
Neural-based Unigram Language Mode

Problem: Does not model sequential information (too local)

We need sequence modeling!

* Slides from Louis-Philippe Morency
Sequence Modeling
Why Model **Sequences**?

* slide from Dhruv Batra

**Image Credit:** Alex Graves and Kevin Gimpel

\[ x = \text{bringen sie bitte das auto zurück} \]

\[ y = \text{please return the car} \]
Multi-modal tasks

[ Vinyals et al., 2015 ]
Sequences where you don’t expect them …

Classify images by taking a series of “glimpses”

[ Gregor et al., ICML 2015 ]
[ Mnih et al., ICLR 2015 ]
Sequences where you don’t expect them …

Classify images by taking a series of “glimpses”

[ Gregor et al., ICML 2015 ]
[ Mnih et al., ICLR 2015 ]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Sequences in Inputs or Outputs?

Input: No sequence
Output: No seq.
Example: “standard” classification / regression problems

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Sequences in Inputs or Outputs?

**Input:** No sequence  
**Output:** No seq.  
**Example:** “standard” classification / regression problems

**Input:** No sequence  
**Output:** Sequence  
**Example:** Im2Caption

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Sequences in Inputs or Outputs?

**Input:** No sequence

**Output:** No seq.

**Example:** "standard" classification / regression problems

**Input:** No sequence

**Output:** Sequence

**Example:** Im2Caption

**Input:** Sequence

**Output:** No seq.

**Example:** sentence classification, multiple-choice question answering

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Sequences in Inputs or Outputs?

one to one

Input: No sequence
Output: No seq.
Example: “standard” classification / regression problems

one to many

Input: No sequence
Output: No seq.
Example: Im2Caption

many to one

Input: Sequence
Output: No seq.
Example: sentence classification, multiple-choice question answering

many to many

Input: Sequence
Output: Sequence
Example: machine translation, video captioning, open-ended question answering

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Key Conceptual Ideas

Parameter Sharing
— in computational graphs = adding gradients

“Unrolling”
— in computational graphs with parameter sharing

Parameter Sharing + “Unrolling”
— Allows modeling *arbitrary length sequences*
— Keeps number of parameters in check

* slide from Dhruv Batra*
Recurrent Neural Network

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
usually want to predict a vector at some time steps
Recurrent Neural Network

We can process a sequence of vectors $\mathbf{x}$ by applying a recurrence formula at every time step:

$$ h_t = f_W(h_{t-1}, x_t) $$

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Recurrent Neural Network

We can process a sequence of vectors $x$ by applying a recurrence formula at every time step:

$$h_t = f_W(h_{t-1}, x_t)$$

**Note:** the same function and the same set of parameters are used at every time step.

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
(Vanilla) **Recurrent** Neural Network

\[ h_t = f_W(h_{t-1}, x_t) \]
(Vanilla) **Recurrent** Neural Network

\[ h_t = f_W(h_{t-1}, x_t) \]

\[ h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h) \]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
(Vanilla) **Recurrent Neural Network**

\[ y_t = W_{hy} h_t + b_y \]

\[ h_t = f_W(h_{t-1}, x_t) \]

\[ h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t + b_h) \]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
RNN Computational Graph

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
RNN Computational Graph

\[ h_0 \xrightarrow{f_W} h_1 \xrightarrow{f_W} h_2 \]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
RNN Computational Graph

\[ h_0 \xrightarrow{f_W} h_1 \xrightarrow{f_W} h_2 \xrightarrow{f_W} h_3 \xrightarrow{\cdots} h_T \]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
RNN Computational Graph

Re-use the same weight matrix at every time-step

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
RNN **Computational Graph**: Many to Many

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
RNN Computational Graph: Many to Many

\[
\begin{align*}
W & \\
x_1 & \rightarrow f_W & h_0 \\
x_2 & \rightarrow f_W & h_1 \\
x_3 & \rightarrow f_W & h_2 \\
\vdots & \rightarrow f_W & h_T \\
\end{align*}
\]

\[
\begin{align*}
h_0 & \rightarrow f_W & h_1 \\
h_1 & \rightarrow f_W & h_2 \\
h_2 & \rightarrow f_W & h_3 \\
\vdots & \rightarrow f_W & h_T \\
\end{align*}
\]

\[
\begin{align*}
y_1 & \rightarrow L_1 & y_2 \\
y_2 & \rightarrow L_2 & y_3 \\
y_3 & \rightarrow L_3 & y_T \\
\vdots & \rightarrow L_T & \end{align*}
\]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
**RNN Computational Graph: Many to Many**

\[
\begin{align*}
    h_0 & \rightarrow f_W \rightarrow h_1 \\
    y_1 & \rightarrow L_1 \\
    h_1 & \rightarrow f_W \rightarrow h_2 \\
    y_2 & \rightarrow L_2 \\
    h_2 & \rightarrow f_W \rightarrow h_3 \\
    y_3 & \rightarrow L_3 \\
    h_3 & \rightarrow f_W \rightarrow h_T \\
    y_T & \rightarrow L_T \\
    h_T & \rightarrow L
\end{align*}
\]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
RNN Computational Graph: Many to One

\[ h_0 \rightarrow f_W \rightarrow h_1 \rightarrow f_W \rightarrow h_2 \rightarrow f_W \rightarrow h_3 \rightarrow \ldots \rightarrow h_T \]

\[ x_1 \rightarrow W \rightarrow \]

\[ x_2 \rightarrow \]

\[ x_3 \rightarrow \]

\[ y \rightarrow \]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
RNN Computational Graph: One to Many

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Sequence to Sequence: Many to One + One to Many

Many to one: Encode input sequence in a single vector

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
**Sequence to Sequence:** Many to One + One to Many

**Many to one:** Encode input sequence in a single vector

**One to many:** Produce output sequence from single input vector

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Example: Character-level Language Model

Vocabulary:
[‘h’, ‘e’, ‘l’, ‘o’]

Example training sequence:
“hello”
Example: Character-level Language Model

Vocabulary:

['h', 'e', 'l', 'o']

Example training sequence:

“hello”

\[ h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t + b_h) \]
Example: Character-level Language Model

Vocabulary:

[‘h’, ‘e’, ‘l’, ‘o’]

Example training sequence:

“hello”
Example: Character-level Language Model (Sampling)

Vocabulary:
['h', 'e', 'l', 'o']

At test time sample one character at a time and feed back to the model.

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
**Example:** Character-level Language Model *(Sampling)*

**Vocabulary:**

['h', 'e', 'l', 'o']

At test time sample one character at a time and feed back to the model.

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, *cs231n Stanford*
Example: Character-level Language Model (Sampling)

Vocabulary:
['h', 'e', 'l', 'o']

At test time sample one character at a time and feed back to the model
**Example:** Character-level Language Model (Sampling)

**Vocabulary:**
['h', 'e', 'l', 'o']

At test time sample one character at a time and feed back to the model.
Sampling vs. ArgMax vs. Beam Search

**Sampling**: allows to generate diverse outputs

**ArgMax**: could be more stable in practice

**Beam Search**: typically gets the best results
Beam Search

```
<START>
  / \  (25%)
-a  \   /
  cat  (30%)
    / \  (50%)
   meowed (50%)
   /
<END> (100%)
  \
    \  /
   dog (70%)
     / \  (80%)
    barked (20%)
    /
<END> (100%)
  \
    \  /
   cat (50%)
     / \  (50%)
    meowed (50%)
    /
<END> (100%)
  \
    \  /
   the (27%)
     / \  (75%)
    meowed (50%)
    /
<END> (100%)
  \
    \  /
   dog (73%)
     / \  (75%)
    barked (25%)
    /
<END> (100%)
```
Beam Search

A steam engine train travelling down train tracks.
A steam engine train travelling down tracks.
A steam engine train travelling through a forest.
A steam engine train travelling through a lush green forest.
A steam engine train travelling through a lush green countryside.
A train on a train track with a sky background.

Diverse Beam Search

A steam engine travelling down train tracks.
A steam engine train travelling through a forest.
An old steam engine train travelling down train tracks.
An old steam engine train travelling through a forest.
A black train is on the tracks in a wooded area.
A black train is on the tracks in a rural area.
**Teacher Forcing**

**Training** Objective: Predict the next word (cross entropy loss)

![Diagram showing the Teacher Forcing process with input, hidden, and output layers.](image)

**Testing:** Sample the full sequence

![Diagram showing the Testing process with Softmax and sampling of full sequence.](image)
**Teacher Forcing**

**Training** Objective: Predict the next word (cross entropy loss)

**Testing:** Sample the full sequence

Training and testing objectives are not consistent!
Teacher Forcing

Slowly move from Teacher Forcing to Sampling

![Diagram showing the transition from Teacher Forcing to Sampling]

Probability of sampling from the ground truth

[ Bengio et al., 2015 ]

* slide from Marco Pedersoli and Thomas Lucas
**Teacher Forcing**

<table>
<thead>
<tr>
<th>Approach vs Metric</th>
<th>BLEU-4</th>
<th>METEOR</th>
<th>CIDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>28.8</td>
<td>24.2</td>
<td>89.5</td>
</tr>
<tr>
<td>Baseline with Dropout</td>
<td>28.1</td>
<td>23.9</td>
<td>87.0</td>
</tr>
<tr>
<td>Always Sampling</td>
<td>11.2</td>
<td>15.7</td>
<td>49.7</td>
</tr>
<tr>
<td>Scheduled Sampling</td>
<td>30.6</td>
<td>24.3</td>
<td>92.1</td>
</tr>
<tr>
<td>Uniformed Scheduled Sampling</td>
<td>29.2</td>
<td>24.2</td>
<td>90.9</td>
</tr>
<tr>
<td>Baseline ensemble of 10</td>
<td>30.7</td>
<td>25.1</td>
<td>95.7</td>
</tr>
<tr>
<td>Scheduled Sampling ensemble of 5</td>
<td>32.3</td>
<td>25.4</td>
<td>98.7</td>
</tr>
</tbody>
</table>

Baseline: Google NIC captioning model

Baseline **with Dropout**: Regularized RNN version

**Always** sampling: Use sampling from the beginning of training

**Scheduled** sampling: Sampling with inverse Sigmoid decay

**Uniformed** scheduled sampling: Scheduled sampling but uniformly

* slide from Marco Pedersoli and Thomas Lucas
BackProp Through Time

Forward through entire sequence to compute loss, then backward through entire sequence to compute gradient

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Truncated BackProp Through Time

Run backwards and forwards through (fixed length) chunks of the sequence, instead of the whole sequence.

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Truncated BackProp Through Time

Run backwards and forwards through (fixed length) **chunks of the sequence**, instead of the whole sequence

Carry hidden states forward, but only **BackProp** through some smaller number of steps

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Truncated BackProp Through Time

Run backwards and forwards through (fixed length) **chunks of the sequence**, instead of the whole sequence.

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Learning to Write Like Shakespeare

THE SONNETS

by William Shakespeare

From fairest creatures we desire increase,
That thereby beauty's rose might never die,
But as the riper should by time decease,
His tender heir might bear his memory:
But thou, contracted to thine own bright eyes,
Feed'st thy light's flame with self-substantial fuel,
Making a famine where abundance lies,
Thyself thy foe, to thy sweet self too cruel:
Thou art thy own exclusiveness,
And only herald to the gaudy spring,
Within thine own bud buried thy content,
And tender churl mak'st waste in niggarding:
Pity the world, or else this glutton be,
To eat the world's due, by the grave and thee.

When forty winters shall besiege thy brow,
And dig deep trenches in thy beauty's field,
Thy youth's proud livery so gazed on now,
Will be a tatter'd weed of small worth held:
Then being asked, where all thy beauty lies,
Where all the treasure of thy lusty days;
To say, within thine own deep sunken eyes,
Were an all-eating shame, and thriftless praise.
How much more praise deserv'd thy beauty's use,
If thou couldst answer 'This fair child of mine
Shall sum my count, and make my old excuse,'
Proving his beauty by succession thine!
This were to be new made when thou art old,
And see thy blood warm when thou feel'st it cold.

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Learning to Write | Like Shakespeare ... after training a bit

at first:

train more

"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuyw fil on aseterlome
coiniogennc Phe lism thond hon at. MeiDimorotion in ther thize."

train more

Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.

train more

"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Learning to Write Like Shakespeare ... after training

PANDARUS:
Alas, I think he shall be come approached and the day
When little strain would be attain’d into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.

VIOLA:
Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father's world;
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master's ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.

KING LEAR:
O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder'd at the deeds,
So drop upon your lordship's head, and your opinion
Shall be against your honour.

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
static void do_command(struct seq_file *m, void *v)
{
    int column = 32 << (cmd[2] & 0x80);
    if (state)
        cmd = (int)(int_state ^ (in_8(&ch->ch_flags) & Cmd) ? 2 : 1);
    else
        seq = 1;
    for (i = 0; i < 16; i++) {
        if (k & (1 << i))
            pipe = (in_use & UMXTHREAD_UNCCA) +
                ((count & 0x00000000ffffffff) & 0x0000000f) << 8;
        if (count == 0)
            sub(pid, ppc_md.kexec_handle, 0x20000000);
        pipe_set_bytes(i, 0);
    }
    /* Free our user pages pointer to place camera if all dash */
    subsystem_info = &of_changes[PAGE_SIZE];
    rek_controls(offset, idx, &osoffset);
    /* Now we want to deliberately put it to device */
    control_check_polarity(&context, val, 0);
    for (i = 0; i < COUNTER; i++)
        seq_puts(s, "policy ");
}
Everybody got one
And all the pretty mommies want some
And what i told you all was
But you need to stay such do not touch
They really do not want you to vote
what do you condone
Music make you lose control
What you need is right here ahh oh
This is for you and me
I had to dedicate this song to you Mami
Now I see how you can be
I see u smiling i kno u hattig
Best I Eva Had x4
That I had to pay for
Do I have the right to take yours
Trying to stay warm

(2 Chainz - Extremely Blessed)
(Mos Def - Undeniable)
(Lil Wayne - Welcome Back)
(Common - Heidi Hoe)
(KRS One - The Mind)
(Cam’ron - Bubble Music)
(Missy Elliot - Lose Control)
(Wiz Khalifa - Right Here)
(Missy Elliot - Hit Em Wit Da Hee)
(Fat Joe - Bendicion Mami)
(Lil Wayne - How To Hate)
(Wiz Khalifa - Damn Thing)
(Nicki Minaj - Best I Ever Had)
(Ice Cube - X Bitches)
(Common - Retrospect For Life)
(Everlast - 2 Pieces Of Drama)

[ Malmi et al., KDD 2016 ]
Sunshine: First movie generated by AI

Sunshine, a short science fiction movie written entirely by AI, debuts exclusively on Ars today.