Announcements

- Please fill our survey!
- 3 thing that you like about the course
- 3 things that could be improved
Readings

- Attention Is All You Need
- The Illustrated Transformer
- The Annotated Transformer
- Language Modeling with Transformers and PyTorch
Recap - 2 Layer MLP

Output layer (σ node)

hidden units (σ node)

Input layer (vector)

\[ y = \text{softmax}(z) \]
\[ z = Uh \]
\[ y \text{ is a vector} \]

\[ h = \sigma(Wx + b) \]

Could be ReLU
Or tanh

\[ x_1 \]
\[ x_n \]
\[ +1 \]
Deep MLP
Recurrent Neural Networks - RNNs

I want to sleep
Encoder-Decoder Models

I want to sleep

Encoder

Je vuex dormir

Decoder
Limitations

- Long Range Dependencies
- Gradient vanishing / explosion
- Long time to converge
- Expensive computation
I’m want to watch Wicked! How does the weather in NYC look next week?

It looks sunny with some light rain during the weekend.

Oh! But I don’t have a rain jacket :( Is there a store nearby?

There’s a marshall’s a mile away. They have the navy blue jacket you have been eyeing for a while!
I'm want to watch Wicked! How does the weather in NYC look next week?

It looks sunny with some light rain during the weekend.

Oh! But I don’t have a rain jacket :( Is there a store nearby?

There’s a marshall’s a mile away. They have the navy blue jacket you have been eyeing for a while!

Ok! Looks like I can actually go! Book the tickets for next Wed!
I'm want to watch *Wicked*! How does the weather in NYC look next week?

It looks sunny with some light rain during the weekend.

Oh! But I don't have a rain jacket :( Is there a store nearby?

There's a marshall's a mile away. They have the navy blue jacket you have been eyeing for a while!

Ok! Looks like I can actually go! Book the tickets for next Wed!
Gradient vanishing / explosion
Gradient vanishing / explosion
Limitations

- Long Range Dependencies
- Gradient vanishing / explosion
- Long time to converge
- Expensive computation
Transformer Model

Attention is all you need (Vaswani et.al, 2017)
Wide Applications

https://blogs.nvidia.com/blog/2022/03/25/what-is-a-transformer-model/
Real World Impact

- Machine Translation
- Search Engines
- Smart Assistants
- Auto Transcription
- Summarization Engines
- Health Record Analysis
- and many more ....
Questions?
Visual Attention

What toppings are on the hot dog?

Differential Attention for Visual Question Answering (Patro et.al, 2018)
Cross Attention in NMT

I want to sleep

Encoder

Decoder

Attention in NMT

Economic growth has slowed down in recent years.

Das Wirtschaftswachstum hat sich in den letzten Jahren verlangsamt.

La croissance économique s’est ralentie ces dernières années.

Self Attention

I want to sleep
Self Attention - No more recurrence

I want to sleep
Self Attention - No more recurrence

I want to sleep
Self Attention - No more recurrence

I want to sleep
Self Attention - No more recurrence

I want to sleep
I want to watch Wicked in NYC next year.
Self Attention - Word Embedding

I want to sleep
you sleep
where

Embedding $E$

I → I
want → want
to → to
sleep → sleep
Self Attention - Projection Layer

\[ X W^Q \rightarrow E \]
\[ X W^K \rightarrow E \]
\[ X W^V \rightarrow E \]

Query
Key
Value
Embedding

I
want
to
sleep
Self Attention - Projection Layer

Query \( W^Q \)
Key \( W^K \)
Value \( W^V \)

Embedding \( E \)
I
want
to
sleep

Vidhisha Balachandran
Self Attention - Attention Scores

<table>
<thead>
<tr>
<th>Query</th>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6</td>
<td>0.05</td>
<td>0.1</td>
</tr>
<tr>
<td>0.25</td>
<td>I</td>
<td>want</td>
</tr>
<tr>
<td>to</td>
<td>sleep</td>
<td></td>
</tr>
</tbody>
</table>

Embedding E
Self Attention

Query

<table>
<thead>
<tr>
<th>I</th>
<th>want</th>
<th>to</th>
<th>sleep</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>K1</td>
<td>V1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>K2</td>
<td>V2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>K3</td>
<td>V3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>KN</td>
<td>VN</td>
<td></td>
</tr>
</tbody>
</table>

Key

Value

E

I | want | to | sleep

Q1

K1 | V1
K2 | V2
KN | VN
Questions?
### Self Attention - Scaled Dot Product

<table>
<thead>
<tr>
<th>Query</th>
<th>Key</th>
<th>SDP</th>
<th>Value</th>
<th>Scaled Dot Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>K1</td>
<td>(95)</td>
<td>V1</td>
<td>SDP = $\frac{(QK^T)}{\sqrt{d^k}}$</td>
</tr>
<tr>
<td>want</td>
<td>K2</td>
<td>(13)</td>
<td>V2</td>
<td></td>
</tr>
<tr>
<td>to</td>
<td>K3</td>
<td>(35)</td>
<td>V3</td>
<td></td>
</tr>
<tr>
<td>sleep</td>
<td>KN</td>
<td>(72)</td>
<td>VN</td>
<td></td>
</tr>
</tbody>
</table>

**Diagram:**

- **Query (Q):** I want to sleep
- **Key (K):** K1, K2, K3, KN
- **Value (V):** V1, V2, V3, VN
Self Attention - SoftMax

$$\text{score} = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right)$$

<table>
<thead>
<tr>
<th>Query</th>
<th>Key</th>
<th>SDP</th>
<th>Score</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>K1</td>
<td>(95)</td>
<td>(0.6)</td>
<td>V1</td>
</tr>
<tr>
<td>want</td>
<td>K2</td>
<td>(13)</td>
<td>(0.05)</td>
<td>V2</td>
</tr>
<tr>
<td>to</td>
<td>K3</td>
<td>(35)</td>
<td>(0.1)</td>
<td>V3</td>
</tr>
<tr>
<td>sleep</td>
<td>KN</td>
<td>(72)</td>
<td>(0.25)</td>
<td>VN</td>
</tr>
</tbody>
</table>

Vidhisha Balachandran
Self Attention - Soft (Relative) Values

<table>
<thead>
<tr>
<th>Query</th>
<th>Key</th>
<th>SDP</th>
<th>Score</th>
<th>Value</th>
<th>RelValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>K1</td>
<td></td>
<td></td>
<td>V1</td>
<td></td>
</tr>
<tr>
<td>want</td>
<td>K2</td>
<td></td>
<td></td>
<td>V2</td>
<td></td>
</tr>
<tr>
<td>to</td>
<td>K3</td>
<td></td>
<td></td>
<td>V3</td>
<td></td>
</tr>
<tr>
<td>sleep</td>
<td>KN</td>
<td></td>
<td></td>
<td>VN</td>
<td></td>
</tr>
</tbody>
</table>

RelValue = Score * Value
# Self Attention - Attended Repr

<table>
<thead>
<tr>
<th>Query</th>
<th>Key</th>
<th>SDP</th>
<th>SoftMax</th>
<th>Value</th>
<th>RelValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>K1</td>
<td></td>
<td></td>
<td>V1</td>
<td></td>
</tr>
<tr>
<td>want</td>
<td>K2</td>
<td></td>
<td></td>
<td>V2</td>
<td></td>
</tr>
<tr>
<td>to</td>
<td>K3</td>
<td></td>
<td></td>
<td>V3</td>
<td></td>
</tr>
<tr>
<td>sleep</td>
<td>KN</td>
<td></td>
<td></td>
<td>VN</td>
<td></td>
</tr>
</tbody>
</table>

\[ \sum = M1 \]
Self Attention - Attended Contextual Rep
Questions?
I want to sleep.
Problem with Self Attention

- **Self Attention can focus heavily on the same word!**

  Query

  I  I

  Q1

  Values

  I  want  to  sleep
Problem with Self Attention

- Self Attention can focus heavily on the same word!

<table>
<thead>
<tr>
<th>Query</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td>want</td>
</tr>
<tr>
<td></td>
<td>to</td>
</tr>
<tr>
<td></td>
<td>sleep</td>
</tr>
</tbody>
</table>

- Single representation

I like Harry Potter v/s I like Harry Potter

(Book) (Movie)
Multi-Headed Self Attention

H (no: of heads) Different versions of Q,K,V
Each different repr -> Different attended repr
Multi-Headed Self Attention

![Diagram of Multi-Headed Self Attention](image)

- **E**: Input
- **I**: Input
- **want**
- **to**
- **sleep**

**Notation**:
- **Q**: Query
- **K**: Key
- **V**: Value
- **M_H**: Mixture of multi-head outputs
- **Concat**: Concatenation of outputs
Multi-Headed Self Attention

I want to sleep
Multi-Headed Self Attention

Multi-Headed Self Attention + Feed Forward

I want to sleep
Multi-Headed Self Attention

Layer N: Multi-Headed Self Attention + Feed Forward

Layer 2: Multi-Headed Self Attention + Feed Forward

Layer 1: Multi-Headed Self Attention + Feed Forward

I want to sleep
Questions?
Revisiting Self Attention

Query (I)

<table>
<thead>
<tr>
<th>Key</th>
<th>SDP</th>
<th>SM</th>
<th>Value</th>
<th>RelValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>K1</td>
<td></td>
<td>V1</td>
<td></td>
</tr>
<tr>
<td>want</td>
<td>K2</td>
<td></td>
<td>V2</td>
<td></td>
</tr>
<tr>
<td>to</td>
<td>K3</td>
<td></td>
<td>V3</td>
<td></td>
</tr>
<tr>
<td>sleep</td>
<td>KN</td>
<td></td>
<td>VN</td>
<td></td>
</tr>
</tbody>
</table>

$$\sum = \text{M}$$

I want to sleep
Revisiting Self Attention

Query (I)

<table>
<thead>
<tr>
<th>Query</th>
<th>Key</th>
<th>SDP</th>
<th>SM</th>
<th>Value</th>
<th>RelValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>K1</td>
<td></td>
<td></td>
<td>V1</td>
<td></td>
</tr>
<tr>
<td>want</td>
<td>K2</td>
<td></td>
<td></td>
<td>V2</td>
<td></td>
</tr>
<tr>
<td>to</td>
<td>K3</td>
<td></td>
<td></td>
<td>V3</td>
<td></td>
</tr>
<tr>
<td>sleep</td>
<td>KN</td>
<td></td>
<td></td>
<td>VN</td>
<td></td>
</tr>
</tbody>
</table>

\[ \sum = M \]

I want to sleep

Sleep to I want
Revisiting Self Attention

\[
\begin{array}{|c|c|c|c|c|}
\hline
\text{Key} & \text{SDP} & \text{SM} & \text{Value} & \text{RelValue} \\
\hline
K1 & & & V1 & \\
K2 & & & V2 & \\
K3 & & & V3 & \\
KN & & & VN & \\
\hline
\end{array}
\]

\[
\sum = M
\]

I want to sleep

Sleep to I want
Revisiting Self Attention

Query (I)

<table>
<thead>
<tr>
<th>Key</th>
<th>SDP</th>
<th>SM</th>
<th>Value</th>
<th>RelValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\sum = M
\]

Sleep to I want

<table>
<thead>
<tr>
<th>Key</th>
<th>SDP</th>
<th>SM</th>
<th>Value</th>
<th>RelValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sleep to I want

<table>
<thead>
<tr>
<th>Key</th>
<th>SDP</th>
<th>SM</th>
<th>Value</th>
<th>RelValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Revisiting Self Attention

Query (I)

<table>
<thead>
<tr>
<th>Key</th>
<th>SDP</th>
<th>SM</th>
<th>Value</th>
<th>RelValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>want</td>
<td>K1</td>
<td></td>
<td>V1</td>
<td></td>
</tr>
<tr>
<td>to</td>
<td>K2</td>
<td></td>
<td>V2</td>
<td></td>
</tr>
<tr>
<td>sleep</td>
<td>K3</td>
<td></td>
<td>V3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>KN</td>
<td></td>
<td>VN</td>
<td></td>
</tr>
</tbody>
</table>

\[ \sum = M \]

I want to sleep

Query (I)

<table>
<thead>
<tr>
<th>Key</th>
<th>SDP</th>
<th>SM</th>
<th>Value</th>
<th>RelValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sleep</td>
<td>K1</td>
<td></td>
<td>VN</td>
<td></td>
</tr>
<tr>
<td>to</td>
<td>K2</td>
<td></td>
<td>V3</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>K3</td>
<td></td>
<td>V1</td>
<td></td>
</tr>
<tr>
<td>want</td>
<td>KN</td>
<td></td>
<td>V2</td>
<td></td>
</tr>
</tbody>
</table>

\[ \sum = M \]

Sleep to I want
Revisiting Self Attention

<table>
<thead>
<tr>
<th>Query (I)</th>
<th>Key</th>
<th>SDP</th>
<th>SM</th>
<th>Value</th>
<th>RelValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>want</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>to</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sleep</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Σ =</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>M</strong></td>
</tr>
<tr>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>want</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>to</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sleep</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query (I)</th>
<th>Key</th>
<th>SDP</th>
<th>SM</th>
<th>Value</th>
<th>RelValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleep</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>to</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>want</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>want</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>to</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sleep</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Σ =</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>M</strong></td>
</tr>
</tbody>
</table>

Same representation for both sentences - But positions matter!
Positional Encoding

Position embeddings - each position number has an associated embedding

\[ p_1 \rightarrow w \uparrow \quad p_2 \rightarrow \text{want} \uparrow \quad p_3 \rightarrow \text{to} \uparrow \quad p_4 \rightarrow \text{sleep} \]
Positional Encoding

Sinusoidal Position embeddings - generalize to any sequence length

\[ p = f(i, t) \]
Questions?
Transformer Encoder

Layer N
Multi-Headed Self Attention + Feed Forward

Layer 2
Multi-Headed Self Attention + Feed Forward

Layer 1
Multi-Headed Self Attention + Feed Forward

I + w₁ p₁
want + w₂ p₂
to + w₃ p₃
sleep + w₄ p₄

N-Layer Transformer Encoder
Transformer Encoder - Decoder

Transformer Encoder

Self-Attention

I want to sleep

Transformer Decoder

Cross-Attention

<s> Je Vuex

dormir
What’s so great about Transformers?

- Parallelizable computation
  - Entire sequence, all queries, all attention heads computed in parallel
  - Benefits from fast matrix multiplication on GPUs

- Rich expressive power
  - Every token connected to every other token
  - Can form long range dependencies

- Depth not proportional to seq length
  - Reduces exploding/vanishing gradient problem
  - Converges faster
What’s so great about Transformers?

- Parallelizable computation - Entire sequence can be processed in parallel
What’s so great about Transformers?

- Parallelizable computation - Entire sequence can be processed in parallel
- Rich expressive power - long range dependencies
Impact - Wide Applications!

**Classification**

Text → Transformer Model → Label Pred

**Sentence Similarity**

Text1 [SEP] Text2 → Transformer Model → Similar/Dissimilar

**Question Answering**

Question [SEP] Context → Transformer Model → Answer

**Translation**

English Text → Transformer Encoder - Decoder → French Text
Larger Impact
Larger Impact

GLUE scores evolution over 2018-2019

- Single generic models
- 2018 Task-specific-SOTA
- Human performance

<table>
<thead>
<tr>
<th>Model</th>
<th>GLUE Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>BILSTM+ELMo</td>
<td>71</td>
</tr>
<tr>
<td>GPT</td>
<td>75.2</td>
</tr>
<tr>
<td>BERT</td>
<td>79.6</td>
</tr>
<tr>
<td>BERT Big</td>
<td>81.2</td>
</tr>
<tr>
<td>BigBird</td>
<td>82.2</td>
</tr>
</tbody>
</table>
Larger Impact

2018 (left) through 2019 (right)

2020 onwards

ELMo 110M 340M 465M 665M 330M 1.5B 340M 355M 1.5B 8.3B 11B 1.5B 17B 11B 2.6B 66M 9.4B 175B

Vidhisha Balachandran

Undergrad NLP 2022
Thank you!

vbalacha@cs.cmu.edu
Transformer Encoder-Decoder

ENCODER #1
- Add & Normalize
- Self-Attention
- Add & Normalize
- Feed Forward
- Add & Normalize

ENCODER #2
- Add & Normalize
- Self-Attention
- Add & Normalize
- Feed Forward
- Add & Normalize

DECODER #1
- Add & Normalize
- Encoder-Decoder Attention
- Add & Normalize
- Self-Attention

DECODER #2
- Softmax
- Linear
- Add & Normalize
- Feed Forward

X1

X2

POSITIONAL ENCODING
Results/Impact

- Improves results, Establishes SOTA in various tasks!
  - Machine Translation
  - Constituency Parsing
  - Language Modeling
  - and more!

- Computationally faster!
  - No sequential computation - Entire sequence processed in parallel