Sequence Model
Announcements

1. Makeup exam Dec 11

2. We will release the last reading quiz today
Recap on Convolutional neural network

Learned feature representations in CNN

Edges | Textures | Patterns | Parts | Objects
---|---|---|---|---
[Image of Edges] | [Image of Textures] | [Image of Patterns] | [Image of Parts] | [Image of Objects]

[https://distill.pub/2017/feature-visualization/]
Objective today

Understanding neural network structures that are suitable for natural language (i.e., sequences of words)
Outline today

1. Word-2-Vec embedding and positional embedding

2. Attention model

3. Putting things together: the Transformer model
Example: autocompletion

e.g., I went to the climbing gym and I ___
Example: autocompletion

e.g., I went to the climbing gym and I ___

A Language model is a conditional probability model:
Example: autocompletion

e.g., I went to the climbing gym and I ___

A Language model is a conditional probability model:

\[ y_1 \sim P(Y = \cdot \mid x_1, \ldots, x_n) \in \mathbb{R}^{100k} \]
Example: autocompletion

e.g., I went to the climbing gym and I ___

A Language model is a conditional probability model:

\[ y_1 \sim P(Y = \cdot \; x_1, \ldots, x_n) \in \mathbb{R}^{100k} \]

\[ y_2 \sim P(Y = \cdot \; x_1, \ldots, x_n, y_1) \]
Example: autocompletion

e.g., I went to the climbing gym and I ___

A Language model is a conditional probability model:

\[ y_1 \sim P(Y = \cdot x_1, \ldots, x_n) \in \mathbb{R}^{100k} \]

\[ y_2 \sim P(Y = \cdot x_1, \ldots, x_n, y_1) \]

\[ y_m \sim P(Y = \cdot x_1, \ldots, x_n, y_1, \ldots y_{m-1}) \]
Word to Vector Embedding

ML models only take vectors of real numbers as inputs...

e.g., I went to the climbing gym and I
Word to Vector Embedding

ML models only take vectors of real numbers as inputs...

e.g., I went to the climbing gym and I

Size of the English vocabulary (e.g., 100k)

Embedding matrix

128
Word to Vector Embedding

ML models only take vectors of real numbers as inputs...

e.g., I went to the climbing gym and I

\[ u_I \in \mathbb{R}^{128} \]

Size of the English vocabulary (e.g., 100k)

Embedding matrix
Word to Vector Embedding

ML models only take vectors of real numbers as inputs...

e.g., I went to the climbing gym and I

\[ u_I \in \mathbb{R}^{128} \quad \text{and} \quad u_{went} \in \mathbb{R}^{128} \]

Size of the English vocabulary (e.g., 100k)

Embedding matrix
Word to Vector Embedding

ML models only take vectors of real numbers as inputs...

e.g., I went to the climbing gym and I

\[
\begin{align*}
u_l & \in \mathbb{R}^{128} \\
u_{went} & \in \mathbb{R}^{128} \\
u_{and} & \in \mathbb{R}^{128} \\
u_l & \in \mathbb{R}^{128}
\end{align*}
\]

Size of the English vocabulary (e.g., 100k)

<table>
<thead>
<tr>
<th>128</th>
<th>...</th>
</tr>
</thead>
</table>

Embedding matrix
Positional embedding

Order of the words and their positions matter...

e.g., When I say Transformer in ML, I do not mean the transformer in the movies

\[ u_{\text{transformer}} \in \mathbb{R}^{128} \]
Positional embedding

Order of the words and their positions matter...

e.g., When I say Transformer in ML, I do not mean the transformer in the movies
Positional embedding

Order of the words and their positions matter...

e.g., When I say Transformer in ML, I do not mean the transformer in the movies

\[ u_{\text{transformer}} \in \mathbb{R}^{128} \]
\[ + p_4 \in \mathbb{R}^{128} \]
\[ u_{\text{transformer}} + p_{13} \in \mathbb{R}^{128} \]

Create positional embedding using sin functions
Positional embedding

Order of the words and their positions matter...

e.g., When I say Transformer in ML, I do not mean the transformer in the movies

\[ u_{\text{transformer}} \in \mathbb{R}^{128} \]
\[ + p_4 \in \mathbb{R}^{128} \]

Create positional embedding using sin functions

\[ p_t = \begin{bmatrix} \sin(t/c_1) \\ \sin(t/c_2) \\ \vdots \\ \sin(t/c_{128}) \end{bmatrix} \]
Positional embedding

Order of the words and their positions matter...

e.g., When I say Transformer in ML, I do not mean the transformer in the movies

\[ u_{\text{transformer}} \in \mathbb{R}^{128} \]
\[ u_{\text{transformer}} + p_{13} \in \mathbb{R}^{128} \]

Create positional embedding using sin functions

\[ p_t = \begin{bmatrix} \sin(t/c_1) \\ \sin(t/c_2) \\ \vdots \\ \sin(t/c_{128}) \end{bmatrix} \]

Low frequency

High frequency
We turn words into vectors of real numbers

e.g., When I say Transformer in ML, I do not mean the transformer in the movies

\[ u_{\text{transformer}} + p_4 \]

\[ u_{\text{transformer}} + p_{13} \in \mathbb{R}^{128} \]
We turn words into vectors of real numbers

e.g., When I say Transformer in ML, I do not mean the transformer in the movies

\[ u_{\text{transformer}} + p_4 \]

\[ u_{\text{transformer}} + p_{13} \in \mathbb{R}^{128} \]

Feature of the word + feature of the position
Outline today

1. Word-2-Vec embedding and positional embedding

2. Attention model

3. Putting things together: the Transformer model
e.g., When I say Transformer in ML, I do not mean the transformer in the movies.

  e.g., When I say Transformer, I literally mean the transformer in the movies.
Motivation

e.g., When I say Transformer in ML, I do not mean the transformer in the movies

e.g., When I say Transformer, I literally mean the transformer in the movies

Contextual feature: feature of a word should depend on the context around it
Self-attention

I went to the climbing gym

Word-2-vec + positional

$x_1 \downarrow x_2 \downarrow x_3 \downarrow x_4 \downarrow x_5 \downarrow x_6 \in \mathbb{R}^{128}$
Self-attention

I went to the climbing gym

Word-2-vec + positional

Attention head: three matrices:

\( W_q, W_k, W_v \)
Self-attention

I went to the climbing gym

Word-2-vec + positional

Attention head: three matrices:

\[ W_q, W_k, W_v \]

Query \( q = W_q x \)  key \( k = W_k x \)  value \( v = W_v x \)
Self-attention

I went to the climbing gym

Word-2-vec + positional

Attention head: three matrices:

\[ W_q, W_k, W_v \]

\[ q = W_q x \quad k = W_k x \quad v = W_v x \]

Query key value
Self-attention

I went to the climbing gym

Word-2-vec + positional

Attention head:
three matrices:

$W_q, W_k, W_v$

$q = W_q x$
$k = W_k x$
$v = W_v x$

Query key value

$(q_1, k_1, v_1)$
$q_1 = W_q x_1$
$k_1 = W_k x_1$
$v_1 = W_v x_1$

$(q_5, k_5, v_5)$
$(q_6, k_6, v_6)$
Self-attention

I went to the climbing gym

Word-2-vec + positional

Attention head: three matrices:

\[ W_q, W_k, W_v \]

\[ q = W_q x \quad k = W_k x \quad v = W_v x \]

Query key value
Self-attention

I went to the climbing gym

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$x_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(q_1, k_1, v_1)$</td>
<td>$(q_5, k_5, v_5)$</td>
<td>$(q_6, k_6, v_6)$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Attention head:
three matrices:

$W_q, W_k, W_v$

$q = W_q x\quad k = W_q x\quad v = W_q x$

Query key value
Self-attention

I went to the climbing gym

Word-2-vec + positional

Attention head: three matrices:

\[ W_q, W_k, W_v \]

Query key value

\[ q = W_q x \quad k = W_q x \quad v = W_q x \]
Self-attention

I went to the climbing gym

Word-2-vec + positional

Attention head:
three matrices:

\[ W_q, W_k, W_v \]

\[ q = W_q x \quad k = W_q x \quad v = W_q x \]

Query key value
Self-attention

I went to the climbing gym

Word-2-vec + positional

Attention head: three matrices:

$W_q, W_k, W_v$

$q = W_q x \quad k = W_q x \quad v = W_q x$

Query key value

Attention heads:

$(q_1, k_1, v_1) \quad (q_6, k_6, v_6) \quad (q_5, k_5, v_5)$

$k_1^T q_5 \quad \ldots \quad k_i^T q_5 \quad \ldots \quad k_6^T q_5$
Self-attention

I went to the climbing gym

Word-2-vec + positional

Attention head: three matrices:

\[ W_q, \ W_k, \ W_v \]

\[ q = W_q x \quad k = W_q x \quad v = W_q x \]

Query     key     value

Query \( k_i^T q \) to value \( v \) via

\[ \text{Softmax: } \]

\[ p_i = \frac{\exp(k_i^T q)}{\sum_{j=1}^{6} \exp(k_j^T q)} \]

\[ \sum_{i=1}^{6} p_i = 1 \]
Self-attention

I went to the climbing gym

Word-2-vec + positional

Attention head:
three matrices:

$W_q, W_k, W_v$

$q = W_qx \quad k = W_qx \quad v = W_qx$

Query key value

$\text{Softmax: } p_{i,5} = \exp(k_i^T q_5) / \sum_{j=1}^{6} \exp(k_j^T q_5)$

$k_i^T q_5 \rightarrow \infty$

$p_{i,j} \rightarrow 1$
Self-attention

I went to the climbing gym

Word-2-vec + positional

Attention head: three matrices:

\[ W_q, W_k, W_v \]

Query \quad key \quad value

\[ q = W_q x \quad k = W_q x \quad v = W_q x \]

Softmax:

\[ p_{i,5} = \exp(k_i^T q_5) / \sum_{j=1}^{6} \exp(k_j^T q_5) \]

\[ p_{1,5}, p_{2,5}, \ldots, p_{6,5} \in \mathbb{R}^6 \]
Self-attention

I went to the climbing gym

Word-2-vec + positional

Attention head:
three matrices:

\[ W_q, W_k, W_v \]

Query key value

Query: \( q = W_q x \)
Key: \( k = W_q x \)
Value: \( v = W_q x \)

Softmax:

\[ p_{i,5} = \exp(k_i^T q_5) / \sum_{j=1}^{6} \exp(k_j^T q_5) \]

\[ p_{1,5}, p_{2,5}, \ldots, p_{6,5} \]

\[ x'_5 = p_{1,5}v_1 + p_{2,5}v_2 + \ldots + p_{6,5}v_6 \]
Self-attention

I went to the climbing gym

Word-2-vec + positional

Attention head:
three matrices:

\[ W_q, W_k, W_v \]

\[ q = W_q x \quad k = W_q x \quad v = W_q x \]

Query key value

\[ (q_1, k_1, v_1) \quad (q_5, k_5, v_5) \quad (q_6, k_6, v_6) \]

\[ k_i^T q_5 \quad \ldots \quad k_i^T q_5 \quad \ldots \quad k_6^T q_5 \]

Softmax:

\[ p_i,5 = \exp(k_i^T q_5) / \sum_{j=1}^{6} \exp(k_j^T q_5) \]

Self-attention layer

\[ p_{1,5}, p_{2,5}, \ldots, p_{6,5} \]

\[ x'_5 = p_{1,5} v_1 + p_{2,5} v_2 + \ldots + p_{6,5} v_6 \]
Multi-head self-attention

I went to the climbing gym

Word-2-vec + positional

$x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4 \rightarrow x_5 \rightarrow x_6$

Self-attention layer $(W_q, W_k, W_v)$

$x_1' \rightarrow x_2' \rightarrow x_3' \rightarrow x_4' \rightarrow x_5' \rightarrow x_6'$
Multi-head self-attention

I went to the climbing gym

Word-2-vec + positional

Self-attention layer \((W'_q, W'_k, W'_v)\)

Self-attention layer \((W_q, W_k, W_v)\)
Multi-head self-attention

I went to the climbing gym

Word-2-vec + positional

Self-attention layer \((W_q', W_k', W_v')\)

Self-attention layer \((W_q, W_k, W_v)\)

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  x_3 \\
  x_4 \\
  x_5 \\
  x_6
\end{bmatrix}
\]
I went to the climbing gym

Word-2-vec + positional

Multi-head self-attention

Self-attention layer \((W'_q, W'_k, W'_v)\)

Self-attention layer \((W_q, W_k, W_v)\)
Summary so far

I went to the climbing gym

Word-2-vec + positional

Multi-head Self-attention layer \( (W^i_q, W^i_k, W^i_v)_{i=1}^3 \)

Contextual features: e.g., \( x'_4 \) encodes information from all words
Outline today

1. Word-2-Vec embedding and positional embedding

2. Attention model

3. Putting things together: the Transformer model
The Transformer model: encoder

I went to the climbing gym

Word-2-vec + positional

Mutt-head Self-attention layer \((W_q^i, W_k^i, W_v^i)_{i=1}^3\)
The Transformer model: encoder

I went to the climbing gym

Word-2-vec + positional

Mutt-head Self-attention layer \((W_q^i, W_k^i, W_v^i)_{i=1}^{3}\)
The Transformer model: encoder

I went to the climbing gym

Word-2-vec + positional

Mutt-head Self-attention layer \((W^q_i, W^k_i, W^v_i)_{i=1}^3\)
The Transformer model: encoder

I went to the climbing gym

Word-2-vec + positional

Mutt-head Self-attention layer \((W^j, W^k, W^v)^i_{i=1}\)

\[ x_1' + x_1 \quad x_2' + x_2 \quad x_3' + x_3 \quad x_4' + x_4 \quad x_5' + x_5 \quad x_6' + x_6 \]

FC FC FC FC FC FC

\[ \text{relu Two layer} \quad \text{Fully-connected} \]
The Transformer model: encoder

I went to the climbing gym

Word-2-vec + positional

Mutt-head Self-attention layer \((W_q^i, W_k^i, W_v^i)_{i=1}^3\)

\[
x_1' \rightarrow x_1 \quad x_2' \rightarrow x_2 \quad x_3' \rightarrow x_3 \quad x_4' \rightarrow x_4 \quad x_5' \rightarrow x_5 \quad x_6' \rightarrow x_6
\]

\[
x_1 + x_1' \quad x_2 + x_2' \quad x_3 + x_3' \quad x_4 + x_4' \quad x_5 + x_5' \quad x_6 + x_6'
\]

FC FC FC FC FC FC

x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6
The Transformer model: encoder

I went to the climbing gym

Word-2-vec + positional

Mutt-head Self-attention layer \((W^{ij}, W^{ik}, W^{iv})_{i=1}^{3}\)

Layer 1
The Transformer model: encoder

I went to the climbing gym

Word-2-vec + positional

Mutt-head Self-attention layer \((W^i_q, W^i_k, W^i_v)_{i=1}^3\)

\[x_1' + x_1, x_2' + x_2, x_3' + x_3, x_4' + x_4, x_5' + x_5, x_6' + x_6\]

Layer 1

\[x_1, x_2, x_3, x_4, x_5, x_6\]
The Transformer model: decoder

I went to the climbing gym

Transformer Encoder

$x_1$  $x_2$  $x_3$  $x_4$  $x_5$  $x_6$
I went to the climbing gym
and I trained
The Transformer model: decoder

I went to the climbing gym

Transformer Encoder

and I trained

Word2vec + positional

$u_1, u_2, u_3 \in \mathbb{R}^{d_w}$
The Transformer model: decoder

I went to the climbing gym

Transformer Encoder

and I trained

Word2vec + positional

\[ u_1 \quad u_2 \quad u_3 \]

cross-attention \((W_q, W_k, W_v)\)
The Transformer model: decoder

I went to the climbing gym

and I trained

Word2vec + positional

Transformer Encoder

\[ k_1, v_1 k_2, v_2 k_3, v_3 k_4, v_4 k_5, v_5 k_6, v_6 \]

\[ K = W_k \cdot x, V = W_v \cdot x \]

cross-attention \( (W_q, W_k, W_v) \)
I went to the climbing gym

and I trained

Word2vec + positional

$q_1 = W_q u_1$

cross-attention ($W_q, W_k, W_v$)
The Transformer model: decoder

I went to the climbing gym

Transformer Encoder

and I trained

Word2vec + positional

\[ q_1 = W_q u_1 \]

\[ x_7 = \sum_{i=1}^{6} p_i v_i \]

cross-attention \((W_q, W_k, W_v)\)
The Transformer model: decoder

I went to the climbing gym

and I trained

Word2vec + positional

Transformer Encoder

$u_1$, $u_2$, $u_3$

cross-attention ($W_q$, $W_k$, $W_v$)
The Transformer model: decoder

I went to the climbing gym

and I trained

Transformer Encoder

$u_1$  $u_2$  $u_3$

$k_7, v_7$

$k_7 = W_k u_1$

$v_7 = W_v u_1$

cross-attention $(W_q, W_k, W_v)$
The Transformer model: decoder

I went to the climbing gym

Transformer Encoder

and I trained

Word2vec + positional

cross-attention \((W_q, W_k, W_v)\)
The Transformer model: decoder

I went to the climbing gym

and I trained

Word2vec + positional

\begin{align*}
q_3 &= W_q u_3 \\
x_8 &= \sum_{i=1}^{7} p_i v_i
\end{align*}

cross-attention \((W_q, W_k, W_v)\)
The Transformer model: decoder

I went to the climbing gym

Transformer Encoder

and I trained

Word2vec + positional

\[ q_3 = W_q u_3 \]
\[ x_8 = \sum_{i=1}^{7} p_i v_i \]

Note: we do not pay attention to future words
The Transformer model: decoder

I went to the climbing gym

and I trained

Word2vec + positional
I went to the climbing gym

Transformer Encoder

and I trained

Word2vec + positional

Multi-head cross-attention

\[(W_q^i, W_k^i, W_v^i)_{i=1}^3\]

+residual connection and FC
The Transformer model: decoder

I went to the climbing gym

Transformer Encoder

Word2vec + positional

Multi-head cross-attention

$$(W_q^i, W_k^i, W_v^i)_{i=1}^3$$

+residual connection and FC

and I trained

$\times N$

$N=6$
The Transformer model: decoder

I went to the climbing gym

Transformer Encoder

and I trained

Word2vec + positional

Multi-head cross-attention

\((W^i_q, W^i_k, W^i_v)_{i=1}^3\)

+residual connection and FC

\(\times N\)

\(x_1, x_2, x_3, x_4, x_5, x_6\)

\(x_7, x_8, x_9\)
The Transformer model: decoder

I went to the climbing gym

and I trained

Word2vec + positional

Multi-head cross-attention

$(W^i_q, W^i_k, W^i_v)_{i=1}^3$ + residual connection and FC

$\times N$

$x_7, x_8, x_9$

Linear classifier w/ 100k labels
The Transformer model: decoder

I went to the climbing gym

and I trained

Word2vec + positional

Multi-head cross-attention

$\left( W_q^i, W_k^i, W_v^i \right)_{i=1}^{3}$

+ residual connection and FC

$x_7, x_8, x_9$

Linear classifier w/ 100k labels

$\{ p_1, p_2, \ldots, p_{100k} \}$
I went to the climbing gym

Transformer Encoder

and I trained

Word2vec + positional

Multi-head cross-attention

$(W_q^i, W_k^i, W_v^i)_{i=1}^3$

+residual connection and FC

$x_7, x_8, x_9$

Linear classifier w/ 100k labels

$\{p_1, p_2, \ldots, p_{100k}\} \sim$ really
The Transformer model: decoder

I went to the climbing gym

and I trained really hard

Transformer Encoder

Word2vec + positional

Multi-head cross-attention

$W^i_q, W^i_k, W^i_v$ $i=1$ +residual connection and FC

$x_7, x_8, x_9$

Linear classifier w/ 100k labels

$\{p_1, p_2, \ldots, p_{100k}\} \sim$ really

$x_1, x_2, x_3, x_4, x_5, x_6$
Take home task:

Check out the the original paper (not too hard to read!)