Transformers from the Ground Up

Sebastian Raschka
https://sebastianraschka.com
Recent Advances in Google Translate

Monday, June 8, 2020

Posted by Isaac Caswell and Bowen Liang, Software Engineers, Google Research

https://ai.googleblog.com/2020/06/recent-advances-in-google-translate.html
Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, D…

https://www.pnas.org/content/118/15/e2016239118.short
https://github.blog/2021-06-29-introducing-github-copilot-ai-pair-programmer/
Topics

1. Augmenting RNNs with attention
2. Self-attention
3. The original transformer architecture
4. Large-scale language models
5. Fine-tuning a pre-trained BERT model in PyTorch
6. Quo vadis, transformers?
Think of this talk as a conceptual overview

That may help to navigate the transformer jungle if you are interested

Please don't worry so much about the mathematical or conceptual details in this talk. These details would take many, many hours to talk about and digest.
Topics

1. Augmenting RNNs with attention
2. Self-attention
3. The original transformer architecture
4. Large-scale language models
5. Fine-tuning a pre-trained BERT model in PyTorch
6. Quo vadis, transformers?
Regular encoder-decoder RNN for seq2seq tasks

RNN = Recurrent neural network

Seq2seq = sequence-to-sequence (e.g., translation, summarization, ...)

Hidden state containing information about entire input sequence and previous hidden states
Why parsing all the input before attempting translation?

Because we can't just translate word by word

<table>
<thead>
<tr>
<th>German input sentence:</th>
<th>Kannst du mir helfen diesen Satz zu uebersetzen?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word-by-word translation (incorrect):</td>
<td>Can you me help this sentence to translate?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>German input sentence:</th>
<th>Kannst du mir helfen diesen Satz zu uebersetzen?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct English translation:</td>
<td>Can you help me to translate this sentence?</td>
</tr>
</tbody>
</table>
Problem: RNN has to remember a lot in 1 hidden state

How can we deal with long-range dependencies better and improve language translation?

Idea: attention mechanism that lets each decoder step access relevant inputs

An RNN with attention mechanism

Output sequence:

Hidden states (with context information):

Context vector:

Attention weights:

Concatenated hidden states:

Hidden states from:
- reverse direction:
- forward direction:

Input sequence:
Topics

1. Augmenting RNNs with attention

2. **Self-attention**

3. The original transformer architecture

4. Large-scale language models

5. Fine-tuning a pre-trained BERT model in PyTorch

6. Quo vadis, transformers?
A simple form of self-attention*

*Minor detail: "self"-attention, because in the attention-based RNN, attention weights are derived from the connection between input & output elements, while self-attention mechanism only focuses on the inputs*
Self-attention with learnable weights

"scaled dot product attention"

Where we have three weight matrices $U_q, U_k, U_v$

- Query: $q^{(i)} = U_q x^{(i)}$
- Key: $k^{(i)} = U_k x^{(i)}$
- Value: $v^{(i)} = U_v x^{(i)}$

where $\alpha_{2,i} = \text{softmax} \left( \frac{\omega_{2,i}}{\sqrt{m}} \right) = \frac{\exp(\omega_{2,i}^T)}{\sum_j \exp(\omega_{2,j}^T)}$

where $z^{(2)} = \sum_{j=1}^{T} \alpha_{2,j} v^{(j)}$
Topics

1. Augmenting RNNs with attention
2. Self-attention
3. The original transformer architecture
4. Large-scale language models
5. Fine-tuning a pre-trained BERT model in PyTorch
6. Quo vadis, transformers?
Attention Is All You Need


• A deep learning architecture for language translation centered around self-attention

• Without any RNN parts
The original transformer architecture
What is multi-head attention?

Multiple matrices $U$ to stack the following multiple time (like kernels/ channels in a CNN):

- Query: $q^{(i)} = U_q x^{(i)}$
- Key: $k^{(i)} = U_k x^{(i)}$
- Value: $v^{(i)} = U_v x^{(i)}$

Image credit: Jitian Zhao
What is "masked" multi-head attention?

It masks out words after the current position (those words that the decoder still has to generate)

Can be considered as a form of unidirectional (as opposed to bidirectional) parsing
Topics

1. Augmenting RNNs with attention
2. Self-attention
3. The original transformer architecture
4. Large-scale language models
5. Fine-tuning a pre-trained BERT model in PyTorch
6. Quo vadis, transformers?
Focus on 2 popular models/approaches:

- **GPT (unidirectional)** -> good for generating text
- **BERT (bidirectional)** -> good for prediction (classification)

Common theme among most transformers:

1. Pre-training on very large, unlabeled datasets
2. Then fine-tuning on labeled dataset for respective target tasks
Training transformers: a 2-step approach

1. Unsupervised pre-training (self-supervised learning)
   - Unlabeled text corpus
   - Transformer

2. Transfer to specific task

---

**Feature-based approach**
- Labeled training set
- Pre-trained Transformer
- Embedding: output of last layer(s)
- Classifier
- Train classifier (e.g., logistic regression, multi-layer perceptron, etc.) on embeddings
- Only update classifier parameters

**Fine-tuning approach**
- Labeled training set
- Pre-trained Transformer
- Added output layer & softmax activation for classification
- Update all model parameters
## GPT: Generative Pretrained Transformer

<table>
<thead>
<tr>
<th>Model</th>
<th>Release year</th>
<th>Number of parameters</th>
<th>Title</th>
<th>Manuscript link</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPT-2</td>
<td>2019</td>
<td>1.5 billion</td>
<td>Language Models are Unsupervised Multitask Learners</td>
<td><a href="https://www.semanticscholar.org/paper/Language-Models-are-Unsupervised-Multitask-Learners-Radford-Wu/9405cc0d616988371b2755e573cc28650d14dfe">https://www.semanticscholar.org/paper/Language-Models-are-Unsupervised-Multitask-Learners-Radford-Wu/9405cc0d616988371b2755e573cc28650d14dfe</a></td>
</tr>
</tbody>
</table>
GPT-1

Decoder

Additional layer(s) for fine-tuning

Unidirectional language model: next word prediction
GPT-2: Zero-shot learning

Examples are provided via context (no fine-tuning), e.g.,

“translate to french, english text, french text”
GPT-3: Zero- and few-shot learning

The three settings we explore for in-context learning

1) Zero-shot
The model is only given a natural language description of the task. No weight updates are performed.

- Translate English to French:
  - Task description
  - Prompt
  - cheese =>

2) One-shot
In addition to the task description, the model is also given a simple example of a task. No weight updates are performed.

- Translate English to French:
  - Task description
  - Example
  - Prompt
  - sea otter => loutre de mer
  - cheese =>

3) Few-shot
In addition to the task description, the model is also given a simple example of a task. No gradient updates are performed.

- Translate English to French:
  - Task description
  - Examples
  - Prompt
  - sea otter => loutre de mer
  - peppermint => menthe poivrée
  - plush giraffe => giraffe peluche
  - cheese =>

Traditional fine-tuning (not for GPT-3)

- Fine-tuning
  - The model is trained using a large corpus of example tasks.
  - sea otter => loutre de mer → Example #1
  - Gradient update
  - peppermint => menthe poivrée → Example #2
  - Gradient update
  - ... → Prompt
  - plush giraffe => giraffe peluche → Example #N
  - Gradient update
  - cheese =>
Bidirectional Encoder Representations from Transformers: BERT


featuring a bidirectional ("nondirectional") training as it reads all elements at once (as opposed to word by word, left to right)
BERT's Pre-training task 1/2: Masked Language Model

15% of the words are masked (or "marked") and are treated as follows ...

Input sentence: A quick brown fox jumps over a lazy dog.

Output sentence:

- 80% Mask token: replace fox with [MASK]
- 10% Random token: replace fox with coffee
- 10% Unchanged: keep fox
BERT's Pre-training task 2/2: Next sentence prediction


Sentence separator token

Classification token: IsNext, NotNext

Sentence A

Sentence B
BERT Fine-tuning tasks

(a) Sentence pair classification tasks

(b) Single sentence classification tasks

(c) Question answering tasks

(d) Single sentence tagging tasks
Topics

1. Augmenting RNNs with attention
2. Self-attention
3. The original transformer architecture
4. Large-scale language models
5. Fine-tuning a pre-trained BERT model in PyTorch
6. Quo vadis, transformers?
Fine-tuning BERT

(a) Sentence pair classification tasks

(b) Single sentence classification tasks

Multiple sentences can be concatenated (512 token limit)

(c) Question answering tasks

(d) Single sentence tagging tasks
Large Movie Review Dataset

This is a dataset for binary sentiment classification containing substantially more data than previous benchmark datasets. We provide a set of 25,000 highly polar movie reviews for training, and 25,000 for testing. There is additional unlabeled data for use as well. Raw text and already processed bag of words formats are provided. See the README file contained in the release for more details.

Large Movie Review Dataset v1.0

When using this dataset, please cite our ACL 2011 paper [bib].

Contact

For comments or questions on the dataset please contact Andrew Maas. As you publish papers using the dataset please notify us so we can post a link on this page.

Publications Using the Dataset

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis, The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).

https://ai.stanford.edu/~amaas/data/sentiment/
Code notebook: https://github.com/rasbt/2021-pydata-jeddah
Topics

1. Augmenting RNNs with attention
2. Self-attention
3. The original transformer architecture
4. Large-scale language models
5. Fine-tuning a pre-trained BERT model in PyTorch
6. Quo vadis, transformers?
GPT-3
(175 billion)

Image Source: https://medium.com/huggingface/distilbert-8cf380435b5
THE COST OF TRAINING NLP MODELS
A CONCISE OVERVIEW

Or Sharir
A121 Labs
ors@ai21.com

Barak Peleg
A121 Labs
barakp@ai21.com

Yoav Shoham
A121 Labs
yoavs@ai21.com

April 2020


- $2.5k - $50k (110 million parameter model)
- $10k - $200k (340 million parameter model)
- $80k - $1.6m (1.5 billion parameter model)
OpenAI’s text-generating system GPT-3 is now spewing out 4.5 billion words a day

_Robot-generated writing looks set to be the next big thing_  
By James Vincent | Mar 29, 2021, 8:24am EDT

https://www.theverge.com/2021/3/29/22356180/openai-gpt-3-text-generation-words-day
Figure credits:

Python Machine Learning book chapter on Transformers
by Jitian Zhao and Sebastian Raschka
Python Machine Learning

Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2

Sebastian Raschka & Vahid Mirjalili

https://sebastianraschka.com/books/

@rasbt

https://www.youtube.com/playlist?list=PLTKMiZHvD_2KjtXOW0zFhFbAjjIH51

https://sebastianraschka.com/blog/2021/dl-course.html