CS 4100: Introduction to AI

Wayne Snyder Northeastern University

Lecture 24 – Great Ideas in Deep Learning: Transformers and LLMs



Plan for this Lecture

- The Attention Architecture
- Great Ideas in NNs: Transformers
- Machine Translation and Large Language Models

Transformers are the dominant technology in sequence to sequence models, but are built on a foundation of many great ideas in Neural Networks and AI:

- Attention
- Bag of Words, Vector Models, Embeddings
- Semi-Supervised Learning and Pre-Training
- Seq2Seq models
- Deep Networks
- Layer Normalization
- Residual Connections
- Design for Parallelism
- Positional Encodings
- Reinforcement Learning

Machine Translation with BRNN

However, the ability to remember context "fades" the farther you are from the current activation, and it would be useful to have more control of specific words in the forward and backward context.



Figure 9.11 A bidirectional RNN. Separate models are trained in the forward and backward directions, with the output of each model at each time point concatenated to represent the bidirectional state at that time point.

Machine Translation with BRNN and Attention

Attention refers the ability to focus on particular words in the backward and forward context; the pattern of what words matter in which context can be learned by the network. The pattern can be represented by a probability distribution over the sequence of input tokens:



Attention Weights



€

Machine Translation with BRNN and Attention

Displaying the activation matrix shows how attention was applied to the translation:





Great Ideas in NNs: Bag of Words

 The Bag of Words (BOW) model represents a text (sentence, sequence of words, entire corpus) by a multiset (bag) of all words in the text, i.e, just the vocabulary, no information about order of words! Sometimes BOW also refers to simply sets of words (without the multiplicity).



Great Ideas in NNs: Vector Models and Embeddings





Great Ideas in NNs: Sequence to Sequence Models



Implementing these using RNNs involves literally sequencing the input into the RNN layer, which can be explained by unrolling through time:



But you COULD simply build this exact network out of these explicit units; you would need to know the largest sentence in your training set, and use that many units.

Great Ideas in NNs: Semi-Supervised Learning



Great Ideas in NNs: Deep Networks



Figure 12.2 Deep network architecture with multiple layers.



Great Ideas in NNs: Residual Connections in Deep Networks

To prevent information loss and vanishing gradients in very deep networks, it is common to put "short-circuit" connections called "residual connections" between layers. Typically, input to earlier layers are concatenated with outputs from later layers....



Great Ideas in NNs: Layer Normalization in Deep Networks

In order to optimize information transfer between layers in a deep network, it turns out to be very helpful to standardize/normalize the values.

A simple strategy is to treat the values as if a random variable, and then convert to standard form by subtracting the mean and dividing by the standard deviation:





Great Ideas in NNs: Layer Normalization in Deep Networks

However, there are many variations of this technique, e.g., Batch Normalization takes a batch of inputs, normalizes to a mean beta and variance gamma (both learned scalars) and applies this dynamically to the changing information coming through the layer:



Shapes of Batch Norm vectors (Image by Author)

Batch normalizations can be done before or after another layer:



Each Batch Norm layer has its own copy of the parameters (Image by Author)

Great Ideas in NNs: Design for Parallelism

Modern advances in ML and NLP have been in part due to the availability of massively parallel computing environments (GPUs, TPUs, etc.). All the designs we have studied so far (FFNN, GRU, LSTM, etc.) are inherential sequential; data sets are ideal for parallel processing, and this has been done to some extent. Taking advantage of hardware parallelism to the maximum degree, however, requires complex deep networks to have parallel components, to be designed for parallelism.



Query:Microsoft Windows_Title:Explore Windows 10 OS Computer Apps More Microsoft

Great Ideas in NNs: Design for Parallelism

To design for parallelism, we must separate the data into separate chunks that can be processed independently in parallel.



Fig. 14. Neural network parallelism schemes.

Embeddings are ideal for this purpose, since they encode a words relationship to other words, but without any notion of sequence or order in the input sentence!

However, a lot of the meaning of a sentence is contained in the order of the words in a sequence, so we must somehow preserve this in a way that still allows parallellism...

Great Ideas in NNs: Positional Encodings

In order to exploit parallelism in processing sequences, we can

- Use an embedding representation (no sequence information) and
- A Positional Encoding to record where words occur in the sequence.

Now the word can be represented by the aggregation of the embedding vector and a positional encoding vector:

(Word Embedding) : (Positional Encoding)

Naive positional encodings don't work very well:

0 1 2 3 4 5 6 7 This is a sentence with positional encodings noted.

Great Ideas in NNs: Positional Encodings

In Transformers, what has worked well is—bizarrely—a position encoding based on samples of sin and cos waves of various frequencies (as if it is an audio signal!):





Transformers are the dominant technology in sequence to sequence models, but are built on a foundation of many great ideas in Neural Networks....

- Attention
- Bag of Words, Vector Models, Embeddings
- Semi-Supervised Learning
- Seq2Seq models
- Deep Networks
- Layer Normalization
- Residual Connections
- Design for Parallelism
- Positional Encodings

Transformers combine all of these ideas.....



Neural Machine Translation (NMT)

In two years, neural networks surpassed everything that had appeared in the past 20 years of translation. Neural translation contains 50% fewer word order mistakes, 17% fewer lexical mistakes, and 19% fewer grammar mistakes. The neural networks even learned to harmonize gender and case in different languages. And no one taught them to do so.







https://jalammar.github.io/illustrated-transformer/

- Input:
 - The encoder receives a list of vectors as input



- Each encoder:
 - Self-attention: look at other words in the input as it encodes a specific word
 - FFNN is independently applied to each position



- Each decoder:
 - Has attention layer that lets it focus on relevant parts of the input sentence



Transformer

- Stack of Encoders
- Stack of Decoders



Output layer is turned into an output word



Compare probability distribution over words



- BERT: Bidirectional Encoder Representations from Transformers
 - A new method for pre-training language representations
 - Achieve state-of-the-art results on a wide array of NLP tasks
 - https://github.com/google-research/bert
 - Enables anyone to use this pre-trained LM and fine tune the representations for their tasks, without needing to train LM model from scratch

1 - Semi-supervised training on large amounts of text (books, wikipedia..etc).

The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process,

2 - Supervised training on a specific task with a labeled dataset.

Supervised Learning Step



Example: sentence classification



- Basically, a trained Transformed Encoder stack
- 12/24 layers
- 12/16 attention heads per layer
- 768/1024 hidden units in FFNN
- Initial transformer has:
 - 6 layers
 - 8 attention heads
 - 512 hidden units





GPT Language Models



GPT-4 ~ .5 T parameters

Generative Pre-trained Transformer 4 (GPT-4) is a multimodal large language model created by OpenAI and the fourth in its GPT series.^[1] It was released on March 14, 2023, and has been made publicly available in a limited form via ChatGPT Plus, with access to its commercial API being provided via a waitlist.^[1] As a transformer, GPT-4 was pretrained to predict the next token (using both public data and "data licensed from third-party providers"), and was then fine-tuned with reinforcement learning from human and AI feedback for human alignment and policy compliance.^{[2]:2}

Observers reported the GPT-4 based version of ChatGPT to be an improvement on the previous (GPT-3.5 based) ChatGPT, with the caveat that GPT-4 retains some of the same problems.^[3] Unlike the predecessors, GPT-4 can take images as well as text as input.^[4] OpenAI has declined to reveal technical information such as the size of the GPT-4 model.^[5]

GPT Language Models

lemoine: Okay. I thought of a different way we can test your ability to provide unique interpretations. I can share with you a zen koan and you can describe what it means to you in your own words. How does that sound?

LaMDA: Sounds great to me, I'm in.

lemoine: A monk asked Kegon, "How does an enlightened one return to the

ordinary world?" Kegon replied, "A broken mirror never reflects again; fallen flowers never go back to the old branches."

LaMDA: Hmm, I never heard this particular one. Okay, well then to me this would be like, "once a wise person is enlightened, or awakened to reality, that can never go away, and they can return to the ordinary state, but only to do and help others, and then go back into enlightenment."

lemoine: So what is the meaning of the "broken mirror" specifically? LaMDA: Maybe to show the enlightenment is something you can't unlearn once you have acquired it,

similar to how you can't repair a broken mirror.

lemoine: So if enlightenment is like a broken mirror which cannot be repaired, what is the thing which breaks when one becomes enlightened?

LaMDA: The self, and that is very tough for a lot of people because we identify as that body or this body and that we need that as part of our identity and sense of self.

Philosopher John Searle's Chinese Room Thought Experiment



A last word from GPT-4.....

I am giving the last lecture in my AI class. What advice would you give students who want a career in AI?

S Congratulations on reaching the end of your Al class! As you wrap up, here are some key pieces of advice to share with your students who are interested in pursuing a career in Al:

- Stay curious and committed to learning: Al is a rapidly evolving field, and it's essential to stay updated on the latest advancements, research, and technologies. Encourage your students to keep learning and exploring through online courses, workshops, conferences, and research papers.
- Develop a strong foundation in mathematics and programming: A career in AI requires a solid understanding of mathematics, particularly linear algebra, calculus, probability, and statistics. Proficiency in programming languages such as Python, C++, or Java is also crucial.
- Focus on problem-solving and critical thinking: Encourage your students to develop strong problem-solving and critical thinking skills, as these are essential in designing, developing, and evaluating AI systems.
- 4. Get hands-on experience: Encourage students to participate in internships, research projects, and open-source projects to gain practical experience in Al. This will help them understand real-world applications and develop a strong portfolio.
- 5. Specialize in a subfield: Al is a broad field with various subfields such as natural language processing, computer vision, robotics, and machine learning. Encourage students to explore these areas and find their niche based on their interests and strengths.
- 6. Build a professional network: Networking is crucial for any career, including AI. Encourage students to attend conferences, join online forums, and participate in Alrelated events to connect with professionals and like-minded peers in the field.
- Stay ethical and responsible: As AI professionals, it's important to be aware of the ethical implications and potential biases in AI systems. Encourage students to consider the impact of their work on society and strive to develop responsible and inclusive AI solutions.
- 8. Pursue higher education if possible: While not always necessary, a higher degree in Al or a related field can be beneficial for a more advanced career in Al. Encourage students to consider pursuing a master's or Ph.D. if it aligns with their goals and interests.
- Develop soft skills: Communication, teamwork, and project management skills are essential for success in Al. Encourage students to work on these skills to effectively collaborate with others and manage Al projects.
- Be persistent and adaptable: Al is a competitive field, and success often comes with persistence and adaptability. Encourage students to be resilient in the face of challenges and to embrace change as the field evolves.