Homework 4 Paper Questions: ResNets and Transformers

Due: Friday, November 21st at 11:59 pm

Deliverables. Submit a PDF of your write-up to Gradescope *HW4 Paper Questions*
Note: we **highly discourage** very long answers, most or all of your free response answers should be 1-3 sentences.

Overview

Machine learning architectures have evolved over time. Starting from the humble multi-layer perceptron (MLPs) to highly complex neural networks with hundreds of layers! In this homework, we'll explore 2 monumental architectural innovations: **ResNets** and **Transformers**.

You are encouraged to skim related works, but focus on the problems, current solutions, contributions, methods, and limitations.

This assignment is not about memorizing details, but about developing the ability to *read research papers methodically.* Most research papers follow a common structure: they first motivate a **problem**, then describe **current solutions** and their limitations, followed by the **proposed solution and key insights**, the **methods** used, and finally a discussion of **limitations**.

This homework is designed to help you practice and internalize this process of critical reading as you answer the questions. We have also provided pointers to relevant section that you might want to pay more attention to for each question.

- ResNets: Deep Residual Learning for Image Recognition (arXiv:1512.03385)
- Transformers: Attention Is All You Need (arXiv:1706.03762)

Some helpful resources compiled by course staff for understanding the Attention Is All You Need Paper:

- YouTube: Transformers, the tech behind LLMs — Deep Learning Chapter 5 - 3Blue1Brown
- YouTube: Attention in transformers, step-by-step — Deep Learning Chapter 6 - 3Blue1Brown
- YouTube: How might LLMs store facts — Deep Learning Chapter 7 - 3Blue1Brown
- YouTube: Transformer Neural Networks, ChatGPT's foundation, Clearly Explained!!! - StatQuest
- YouTube: Deep Dive into LLMs like ChatGPT
- The Transformer Architecture - Dive Into Deep Learning

1 ResNets

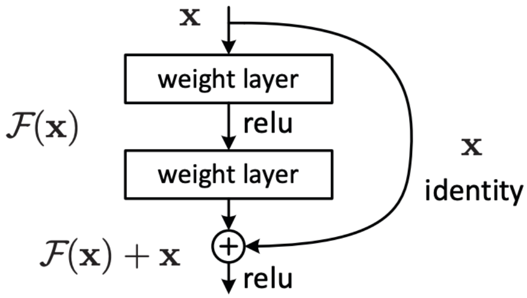

Figure 2 from the Deep Residual Learning for Image Recognition: "Residual learning: a building block."

Problem

Q1. (`1 Pt.`) In less than 5 words, describe the kind of tasks was ResNet designed for. *[Sections: 1 - Introduction]*

Q2. (`1 Pt.`) What problem did the authors of the ResNet paper aim to solve? *[Sections: Abstract, 1 - Introduction]*

Hint: What kind of problems did researchers/engineers notice happening with deeper neural networks? What differentiates **degradation** from **overfitting**?

Current works

Q3. (`2 Pts.`) How might you construct a deeper counterpart of a neural network that produces the exact same outputs as a shallower network? Why doesn't this work in practice, i.e. why does **degradation** occur in deeper neural networks? *[Sections: 1 - Introduction, 2 - Related Work]*

Proposed Solution

Q4. (`1 Pt.`) The ResNet authors proposed building layers that can learn the residual function $F(x)+x$ instead of directly learning $H(x)$. Why might we want to learn the function $F(x)+x$ if it is equivalent to $H(x)$? How does learning $F(x)$ help solve the degradation problem? *[Section: 3.1 - Residual Learning]*

Method Details

Q5. (`2 Pts.`) What are two workarounds if your original input x and the output of your sublayer $F(x)$ don't have equal dimensions? *[Section: 3.3 - Network Architectures - Residual Network]*

Q6. (`2 Pts.`) Explain the 3 options for how residual/shortcut connections might be implemented. and compare their parameter counts (relative comparisons—which has more vs. less—are okay). Based on the parameter count comparisons, briefly discuss how the 3 different residual connection implementations might affect the model's memory requirements or training time. *[Section: 4 - Experiments - Identity vs. Projection Shortcuts]*

Key Insight & Contributions

Q7. (`1 Pt.`) What benefits did the authors notice with residual networks? What empirical proof did the authors provide to show that ResNet architectures didn't suffer from the degradation problem? *[Sections: Abstract, 4 - Experiments, Figure 4, Table 2]*

2 Transformers - Attention Is All You Need

A helpful preface:

1. Transduction (AKA seq2seq) refers to the task of mapping input sequences to output sequences. For example, machine translation refers to a model translating a *source sequence* in one language into a *target sequence* in another language.

2. During generation, the decoder predicts the token at position i. The input embedding the decoder receives comes from the first $i-1$ output tokens. This is called **teacher forcing** during training (feeding in ground-truth tokens shifted by one) and **auto-regressive generation** at inference. Offsetting by one during training ensures the model's prediction for position i depends on previous outputs, never on itself or future targets.

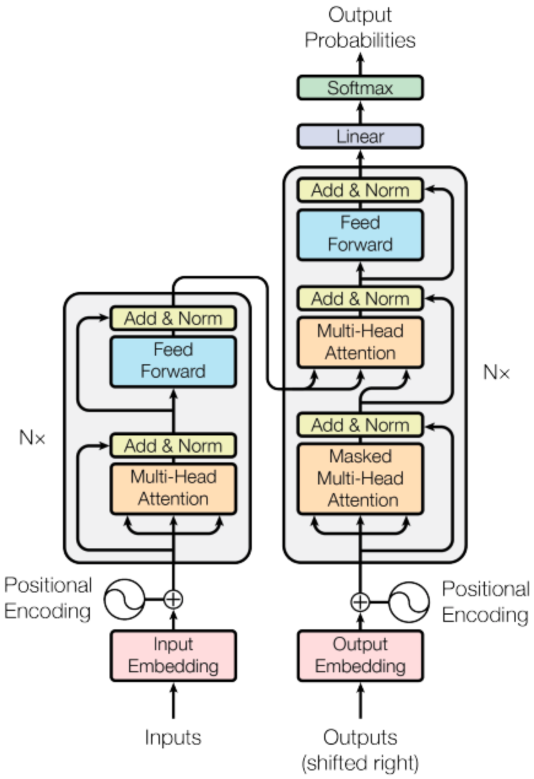

Figure 1 from Attention Is All You Need: ”The Transformer - model architecture.”

Problem

Q8. (`1 Pt.`) In a short sentence, What task is the paper ”Attention Is All You Need” trying to solve? *[Section: 1- Introduction]*

Metrics

Q9. (`2 Pts.`) Two important metrics that the authors highlight in the paper are BLEU scores and perplexity. Describe what BLEU score and perplexity measure, and what their limitations are. Feel free to use some outside research! *[Sections: Abstract, Table 3]*

Current Works

Q10. (`1 Pt.`) What are some challenges that existing architectures face? Describe the shortcomings of both recurrent neural networks (RNNs) and sequential convolutional neural networks. *[Sections: 1 - Introduction, 2 - Background]*

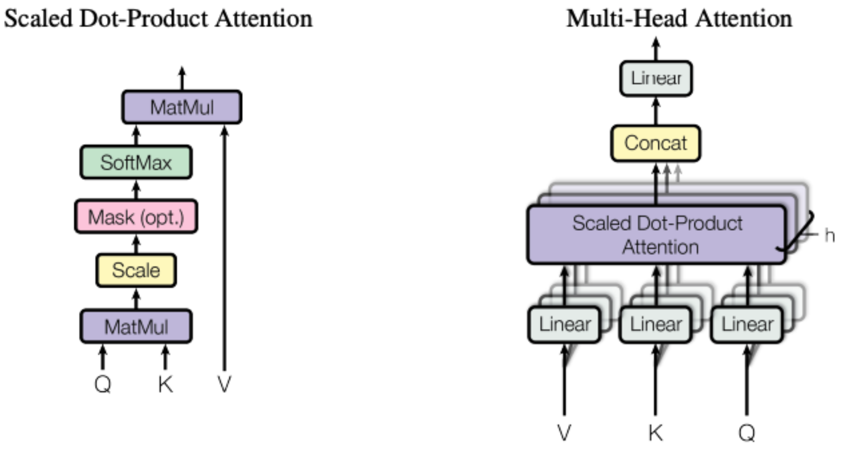

Figure 2 from Attention Is All You Need: ”(left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.”

Proposed solution

Q11. (`1 Pt.`) In a short sentence, describe what ”attention” is in your own words. This doesn't have to be within the context of machine learning, but feel free to start also thinking about how the idea of attention might transfer over to what the transformer architecture tries to do. *[Section: your mind]*

Q12. (`1 Pt.`) What are the jobs of the encoder and decoder? *[Section: 3 - Model Architecture]*

2.1 Method Details

Q13. (`1 Pt.`) What is the purpose of masking during decoder self-attention? *[Sections: 3.1 - Encoder and Decoder Stacks, 3.2.3 - Applications of Attention in our Model]*

Q14. (2 Pts.) Why does the Transformer paper scale QK^T by $\frac{1}{\sqrt{d_k}}$? *[Section: 3.2.1 - Scaled Dot-Product Attention]*

Hint: Imagine we sum up I.I.D. random variables where $x_i \sim \mathcal{N}(0, 1)$. What is the variance of this sum?

Q15. (1 Pt.) What are 2 benefits of multi-headed attention? *[Section: 3.2.2 - Multi-Head Attention]*

Q16. (2 Pts.) Where do the query, key, and value vectors come from in the following parts of a transformer, and why? *[Section: 3.2.3 - Applications of Attention in our Model]*

(a) Encoder self-attention

(b) Decoder self-attention

(c) Decoder cross-attention (the paper refers to this as ”encoder-decoder attention”)

Q17. (2 Pts.) Explain the differences between what the transformer decoder receives as input and what it outputs during training vs. inference. *[Sections: 3 - Model Architecture, 5.1 - Training Data and Batching]*

Q18. (2 Pts.) Why does the standard input to a transformer model (before adding positional encodings) not contain any information about the order of tokens in a sequence? How do positional encodings help the model understand token order? *[Section: 3.5 - Positional Encoding]*

Key Insight / contributions

Q19. (2 Pts.) What advantages does the self-attention mechanism have over recurrent layers and convolutional layers in the context of sequence modeling? *[Section: 4 - Why Self-Attention]*

Future Work

Q20. (2 Pts.) As mentioned earlier in the assignment, the transformer architecture has rocked the machine learning world in countless ways! One application has been in computer vision, where Vision Transformers have matched and even beat state-of-the-art convolutional neural networks in image-processing tasks. How might you adapt transformers to process images? Think about how you might create ”tokens” from images. If you don’t know where to start, feel free to skim follow-up works to this paper which look into this.

Q21. (2 Pts.) Suppose you have a ResNet and a Vision Transformer, both trained on ImageNet with standard preprocessing (e.g. image resizing, pixel standardization). Now, you are given a new image of size 1024×512. How would you process this image for each model? What are the key differences in how ResNets and Vision Transformers handle input image sizes?