EECS 182 Deep Neural Networks
Spring 2023 Anant Sahai

Homework 9

This homework is due on Friday, April 14, 2022, at 10:59PM.

1. Read a Research Paper: FaceNet

In class, you have learnt how self-supervised learning can be used to learn useful representations from large datasets without labels (e.g., learning features from ImageNet). While these features may inherently pick out some notion of "similarity" between different images in the dataset, they are not incentivized to *cluster* different data points based on any interesting similarity measure.

The paper "FaceNet: A Unified Embedding for Face Recognition and Clustering" explores how we can view task of face recognition through the lens of self-supervised (or to be more accurate, *slightly* supervised) learning.

Read the paper and answer the questions below.

(a) **What are the two neural network architectures considered by the authors?**

Solution: The first architecture is based on the Zeiler & Fergus model which consists of multiple interleaved layers of convolutions, non-linear activations, local response normalizations, and max pooling layers. The second architecture is based on the Inception model of Szegedy et al.

(b) **Briefly describe the *triplet loss* and how it differs from a typical supervised learning objective.**

Solution: Open-ended question, there's no "right" answer. The most important distinction is the fact that triplet loss doesn't require labels and can learn by simply specifying "positives" and "negatives" with respect to the "anchor".

(c) **What is the challenge with generating all possible triplets? Briefly describe how the authors address this challenge.**

Solution: Generating all possible triplets would simultaneously be computationally expensive (and likely infeasible for large datasets) and wasteful (since many triplets would trivially satisfy the condition set by equation 1). The authors use a variety of methods outlined in Section 3.3, including "hard" triplet mining for selecting the most informative triplets.

(d) **How many parameters and floating point operations (FLOPS) do the authors use for their neural network? How does this compare to a ResNet-50?**

Solution: Table 1 suggests that the authors use a 140M (NN1) / 7.5M (NN2) parameter model for with 1.6B FLOPS. ResNet-50 uses approximately 23M parameters and 3.8B FLOPS.

(e) Briefly explain what the authors mean by *semi-hard* negatives. What are harmonic embeddings?

Solution: Since the "hardest" negatives may lead to bad local minima (in practice), the authors use *semi-hard* negatives to balance the trade-off between having informative negative examples and avoiding trivial examples that violate the triplet constraint. Semi-hard negatives refer to negative examples in a triplet that are closer to the anchor image than the positive example, but are still far enough away from the anchor image to provide a meaningful training signal.

Harmonic embeddings refer to a way of embedding data points across different neural network models such that they satisfy a special property: embeddings generated by different models v1 and v2 are compatible in the sense that they can be compared to each other. This compatibility greatly simplifies the path to using multiple models and comparing their results.

(f) How does the performance vary with embedding dimensionality?

Solution: The authors find that the performs improves slightly by increasing embedding dimensionality from 64 to 128, then drops on increasing it further. While it is natural to expect the performance to improve with higher dimensionality, the authors speculate that the performance drop may be due to larger embedding spaces requiring more training time to achieve similar (or better) performance. It is also likely that the difference in performance in Table 5 is statistically insignificant.

(g) How does the performance vary with increasing amounts of training data?

Solution: Expectedly, the validation performance increases with increasing amounts of training data, although Table 6 suggests plateauing of the performance. $10\times$ increase from 2.6M to 26M training images improves the performance by $\approx 9\%$, while a further $10\times$ increase only improves performance by 1%.

(h) Briefly share your favorite *emergent* property/result of the learned behavior with a triplet loss from the paper.

Solution: Open ended question, there's no "right" answer.

(i) **Which approach taken by the authors interested you the most? Why?** ($\approx$ 100 words)

Solution: Open ended question, there's no "right" answer.

2. Masked Auto-Encoding

Please follow the instructions in this notebook. You will implement Vision Transformer and Masked Autoencoder in PyTorch. Once you finished with the notebook, download `hw9_submission.zip` and submit it to "Homework 9 (Code) (MAE)" in Gradescope.

3. Coding Question: Summarization (Part I)

Please follow the instructions in this notebook. You will implement a Transformer using fundamental building blocks in PyTorch. You'll apply the Transformer encoder-decoder model to a sequence-to-sequence NLP task: document summarization. Refer to the Attention is All You Need paper for details on the model architecture. Once you finished with the notebook,

- Download `submission_log.json` and submit it to "Homework 9 (Code) (Summarization)" in Gradescope.
- Answer the following questions in your submission of the written assignment:

(a) **Please submit the screenshots of the training loss and the validation loss displayed on Tensorboard.**

Solution: Please refer to the solution notebook.

4. Coding Question: Visualizing Attention

Please run the cells in the Visualizing_BERT.ipynb notebook, then answer the questions below.

(a) Attention in GPT: Run part a of the notebook and generate the corresponding visualizations

i. **What similarities and differences do you notice in the visualizations between the examples in this part?** Explore the queries, keys, and values to identify any interesting patterns associated with the attention mechanism.

Solution: Notice that GPT is paying attention to previous words in the sequence. Specifically, in the first example, the word 'ran' attends to the token 'dog' because it is important for the model to know who or what is doing the running. In the second example, notice that GPT is paying more attention to the subject of the sentence 'dog', rather than the object of the prepositional phrase 'car'. In the example where GPT is tasked with keeping track of past information, the model is indeed appropriately making a connection between the name mentioned previously and the vague pronoun reference in the second sentence. The queries, keys, and values further support this fact that GPT is successful in identifying linguistic patterns.

ii. **How does attention differ between the different layers of the GPT model? Do you notice that the tokens are attending to different tokens as we go through the layers of the network?**

Solution: Students may notice any number of patterns. Here's an example. As you look through the layers of the network, you may notice that GPT tends to place a lot of attention on the first word in the sentence. This is because when the model doesn't know what linguistic patterns are interesting or relevant to the current setting, it looks to the start of the sentence for answers. If you are interested in seeing some more interesting patterns, feel free to play around with this tool some more, or read this interesting blog post.

(b) BERT pays attention: Run part b of the notebook and generate the corresponding visualizations.

i. Look at different layers of the BERT model in the visualizations of part (b) and identify different patterns associated with the attention mechanism. Explore the queries, keys, and values to further inform your answer. **For instance, do you notice that any particular type of tokens are attended to at a given timestep?**

Solution: At some layers, words attend to the immediate next or previous token. At others, there are a lot of attentions to the CLS token. Another occurence is that that words will attend to other instances of the word in the sentence.

ii. **Do you spot any differences between how attention works in GPT vs. BERT? Think about how the model architectures are different.**

Solution: BERT is a bidirectional model, whereas GPT only attends to tokens one at a time. When looking at the words that the attention heads are paying atteniont to this difference becomes obvious.

iii. For the example with syntactically similar but definitionally different sentences, look through the different layers of the two BERT networks associated with sentence a and sentence b, and take a look at the queries, keys, and values associated with the different tokens. **Do you notice any differences in the embeddings learned for the two sentences that are essentially identical in structure but different in meaning?**

Solution: The word "play" in this context has significantly different meanings based on the context. While the overall attention patterns look very similar, in some layers you can see that keys corresponding to “play” look different between the two examples.

iv. **For the pre-training related examples, do you notice BERT's bi-directionality in play? Do you think pre-training the BERT helped it learn better representations?**

Solution: BERT uses bidirectionality in order to attend backwards. For example, SEP tokens look for CLS tokens even though the attention head at layer 2, head 0 mostly consists of words that look at the next work. And yes! BERT is very powerful because the training is self-supervised, and so we do not need to manually label data–pretraining BERT gives it access to larges amounts of data that it can generalize from to form useful representations. If you are interested in seeing some more interesting patterns, feel free to play around with this tool some more, or read this interesting blog post.

(c) BERT has multiple heads!: Run part c of the notebook and generate the corresponding visualizations.

i. **Do you notice different features being learned throughout the different attention heads of BERT? Why do you think this might be?**

Solution: Students should notice different patterns being learned by the different attention heads in the rows of the visualization. Some interesting patterns include the following: an attention head focuses on the special SEP and CLS tokens in the text; an attention head focuses on all words in the text equally; an attention head focuses on the word that was directly before it; etc. The attention heads are able to focus on different patterns as their queries, keys, and values are calculated independently and then combined together to produce a final attention score that is eventually used downstream.

ii. **Can you identify any of the different features that the different attention heads are focusing on?**

Solution: Students should be able to see the different patterns in the visualization. Some are listed in the solution for the part above.

(d) Visualizing untrained attention weights

i. **What differences do you notice in the attention patterns between the randomly initialized and trained BERT models?**

Solution: All of the words pay equal attention to all of the other words. The untrained BERT has no way of distinguishing which words might be more important, and which words are less so.

ii. **What are some words or tokens that you would expect strong attention between? What might you guess about the gradients of this attention head for those words?**

Solution: As seen from the previous parts of the notebook, BERT pays particular attention to word before, the word after, separators, and words that have similar meanings. We expect gradients at these steps to be higher, since these are the features that the model eventually learns.

(e) **Were you able to identify interesting patterns in the visualizations?** If yes, please share some examples (describe in text or paste a screenshot). If not, feel free to use this space for your frustrations.

Solution: We understand that making interpretable observations from large, parametrized models can be challenging, and hope you appreciate this fact too.

5. Homework Process and Study Group

Citing sources and collaborators are an important part of life, including being a student!
We also want to understand what resources you find helpful and how much time homework is taking, so we can change things in the future if possible.

(a) **What sources (if any) did you use as you worked through the homework?**

(b) **If you worked with someone on this homework, who did you work with?**
List names and student ID's. (In case of homework party, you can also just describe the group.)

(c) **Roughly how many total hours did you work on this homework? Write it down here where you'll need to remember it for the self-grade form.**

Contributors:

- Dhruv Shah.
- CS182 Staff from past semesters.
- Jake Austin.
- Olivia Watkins.
- Linyuan Gong.
- Sheng Shen.
- Hao Liu.
- Allie Gu.
- Anant Sahai.
- Shivam Singhal.
- Kevin Li.
- Bryan Wu.