EECS 182 Deep Neural Networks
Fall 2022 Anant Sahai

Discussion 10

1. Finetuning Pretrained NLP Models

In this problem, we will compare finetuning strategies for three popular architectures for NLP.

(a) **BERT** - encoder-only model
(b) **T5** - encoder-decoder model
(c) **GPT** - decoder-only model

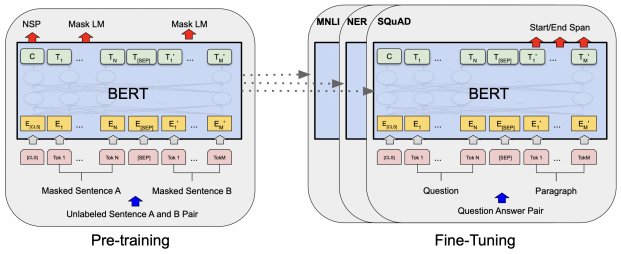

Figure 1: Overall pre-training and fine-tuning procedures for BERT.

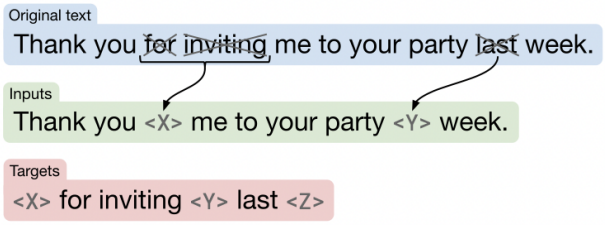

Figure 2: T5 Training procedure

(a) For each of the three models, state the objective used for pretraining.

Solution: BERT uses two training objectives: a masked token prediction objective (i.e. a cross-entropy loss for predicting input tokens which were masked out), and a next sentence prediction objective which predicts whether the two sentences passed in are sequential or not. This is shown in Figure 1 (Followup works such as RoBERTa only use the masked prediction objective.)

T5 uses a slightly different masking objective. Spans of multiple words are masked out from the encoder. The decoder must predict the tokens for each span, as shown in Figure 2.

GPT is trained using using a next-token classification loss.

(b) Consider the MNLI (Multi-Genre Natural Language Inference Corpus) task. It provides a passage and a hypothesis, and you must state whether the hypothesis is an entailment, contradition, or neutral.

> **EXAMPLE:**
> **Passage**: At the other end of Pennsylvania Avenue, people began to line up for a White House tour.
> **Hypothesis**: People formed a line at the end of Pennsylvania Avenue.
> **Classification**: entailment

(i) With each of the 3 models, state whether it is possible to use the model for this task with no finetuning or additional parameters. If so, state how.

Solution: With GPT and T5, you could accomplish this by few-shot prompting. Input a set of examples (passage, hypothesis, and answer), followed by the passage and hypothesis for the current question. The decoder would then predict the answer.
It is technically possible to use BERT for this task by inputting a prompt, masking a token for the answer, and predicting it, but BERT is not good at learning through prompting.[1].

(ii) With each of the 3 models, state how you would use the model for this task if you were able to add additional parameters and/or finetune existing parameters.

Solution: With GPT and T5, you could input the passage and hypothesis, then finetune the model to predict the answer as the next token. Note that with large models it is often common to only finetune a subset of the model parameters.[2]
With BERT, you input a sequence to the model: [<CLS>, passage, <SEP>, hypothesis]. Remove the output layer of the network and replace it with a classification head on the CLS output token. (Some approaches also make use of the other tokens, such as by taking a mean across the other output tokens and concatenating them with the CLS token.) You can either finetune all parameters or only the parameters of the classification head.
(Students may suggest other answers which are valid too - e.g. you could extract features from GPT by concatenating token features from the last couple of layers and training a classification head on top of them, but these may not perform as well).

(c) Next, consider the SQuAD question-answering task. It provides a passage and asks a question about it. The answer is a span within the task.

[1] For very new work modifying BERT to benefit from prompting, see https://arxiv.org/pdf/2201.04337.pdf
[2] For a comparison between of several parameter-efficient finetuning methods, check out the experiments in this paper: https://arxiv.org/pdf/2205.05638.pdf

EXAMPLE: Solution: [a]

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?
gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

Where do water droplets collide with ice crystals to form precipitation?
within a cloud

[a] https://developer.nvidia.com/blog/developing-a-question-answering-application-quickly-usir

Discuss how you could use each of the three models for this task. (You may consider frozen or finetuned methods.)

Solution: You could use prompting methods for GPT and T5, like in the previous example, and have the model output the answer as a sequence. If you have a large dataset available you can improve performance by finetuning the model on the task.

For BERT, first remove the output layer of the network. Pass in [<CLS>, passage, <SEP>, question]. For each item in the output sequence, classify whether it is the start of the span, end of the span, or neither.

(Other answers are possible.)

(d) Compare and contrast the ways we use pretrained representations in BERT to the way we use pretrained autoencoder representations. **Solution:** When you use a single layer of BERT as your representation, the bottom of the network serves the role of the AE encoder (the piece you use to encode inputs for the downstream task), and the final layer(s) serve the role of the AE decoder (the piece which is used in training then thrown away). However, NLP applications sometimes use features from multiple layers in the model, whereas AEs typically use a single layer representation.

2. Vision Transformer

Vision transformers (ViTs) apply transformers to image data by following the following procedure:

(a) **Split image into patches** - The original ViT paper split images into a 16x16 grid of patches.

(b) **Convert each patch into a single vector** - In the original paper, they flattened the patch and applied a linear projection.

(c) **Stack the patches into a sequence, concatenate a CLS token, and add in positional embeddings.** Absolute learned positional embeddings are most common here.

(d) **Pass the sequence through a transformer as usual.**

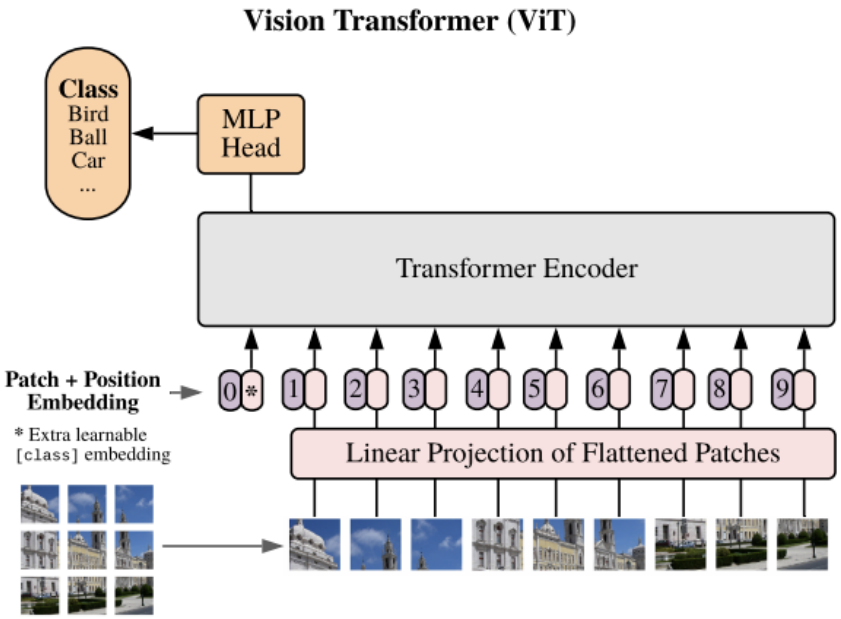

Figure 3: Vision Transformer https://ai.googleblog.com/2020/12/transformers-for-image-recognition-at.html

(a) Does it matter which order you flatten the sequence of patches?

Solution: No, the positional encoding will tell the model where the patch came from.

(b) What is the complexity of the vision transformer attention operation? Assume you have an image of size HxW and patches of size PxP. Only consider the time of the attention operation, not the time to produce queries, keys, and values. Queries, keys, and values are each size D.

Solution: We have (H/P)*(W/P) patches, so the complexity is $O(seq^2 D) = O((HW/P^2)^2 D$.

(c) What is the receptive field of one sequence item after the first layer of the transformer? How does this compare to a conv net, and what are the pros and cons of this? **Solution:** After a single transformer layer, the receptive field is the entire image. In contrast, a conv net's receptive field grows slowly through the layers. This change allows transformers to pick up on spatially distant patterns more easily. However, the conv net's inductive bias toward spatially local patterns can be helpful for learning, especially when training from scratch on a small dataset. Some transformers have also explored only attending to spatially-local patches in early layers of the transformer to recover this inductive bias.

(d) If we forgot to include positional encodings, could the model learn anything at all? State one task where a model could perform well without positional encoding, and one task where it would do poorly. **Solution:** The model would see the image as a bag of patches (similar to a bag of words in NLP). It could still perform well on tasks where patches are sufficient to understand the content of the image (e.g. classifying trafic lights as red, yellow, or green), but it would perform poorly on patches that rely on the order of patches (e.g. reading text from a screen).

(e) If you wanted to add a few conv layers into this achitecture, how would you incorporate them? **Solution:** You first apply a few conv layers to the input image. Then, split the resulting feature map into patches and pass it into the ViT.[3]

(f) How would you use this architecture to do GPT-style autoregressive generation of images? **Solution:** You would need to pass the transformer's output into a decoder which can generate image patches. Like with language, you could then feed the generated patch back into the model to generate the next patch. Train with MSE loss against the ground-truth patches. (Or cross-entropy loss if you're using discrete tokens, e.g. Parti https://parti.research.google/. I don't think discrete tokenization e.g. VQVAE has been covered yet.)

Contributors:

- Olivia Watkins.
- Anant Sahai.
- CS 182/282A Staff from previous semesters.

[3]Example: https://arxiv.org/abs/2106.14881

