Data C182 Designing, Visualizing & Understanding DNN
Fall 2024 Eric Kim, Naveen Ashish

Discussion 09

This discussion covers Vision Transformer (ViT) and Masked Autoencoder (MAE).

1. 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.**

Below is a diagram of ViT for supervised image classification.

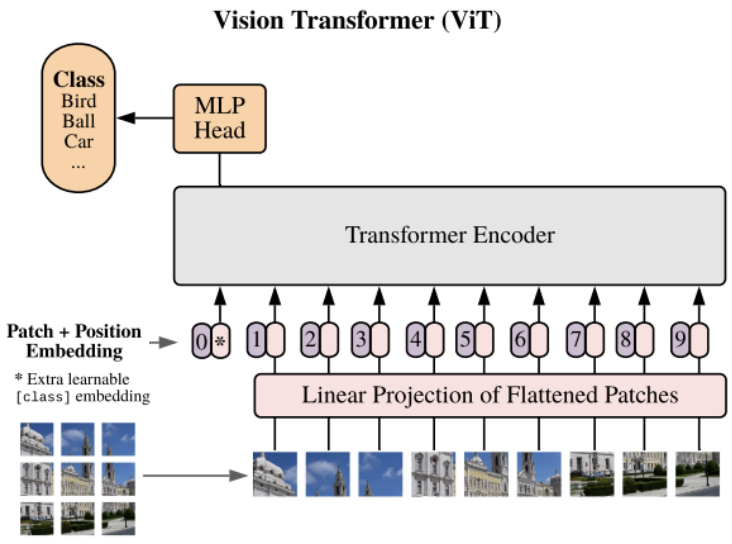

Figure 1: Vision Transformer (Link to Google blog)

(a) We are not going to cover this in detail, but for Homework 4, you will need to use the `einops` Python library. It is a better way to reshape Python tensors, and frequently used to simplify the image-patchify implementation for ViT.

We point you to the good official tutorials:

- https://einops.rocks/pytorch-examples.html
- https://github.com/arogozhnikov/einops/tree/main/docs
- https://githubtocolab.com/arogozhnikov/einops/blob/main/docs/1-einops-basics.ipynb

And if you are adventurous, try lucidrain's very clean implementation of ViT.

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

(c) What is the complexity of the vision transformer attention operation? Assume you have an image of size $H \times W$ and patches of size $P \times P$. 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.

(d) The receptive field of a neuron in a CNN is the subset of all input entries that can affect what the neuron's activation is. That is, if you change any pixel outside the receptive field, then the neuron's activation would stay the same, but if you change any pixel inside the receptive field, then the neuron's activation can change. This is shown in Figure 2.

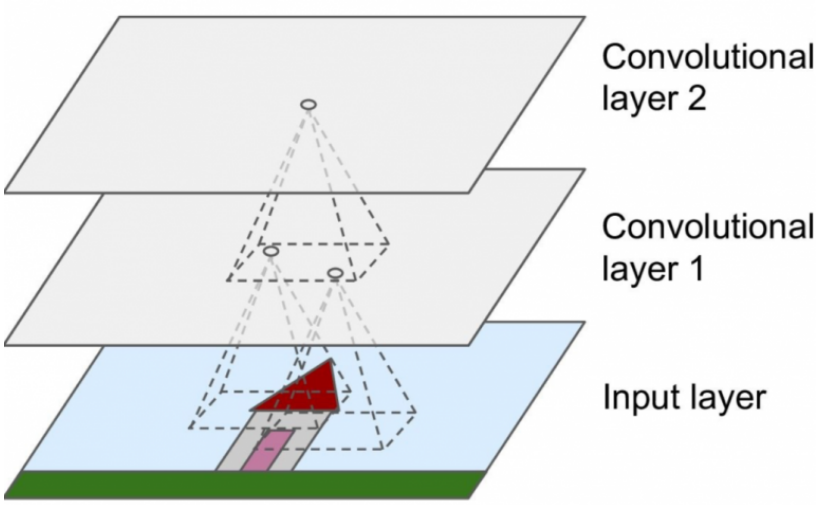

Figure 2: Receptive field of a neuron in a CNN.

What is the receptive field of one sequence item after the first layer of the transformer? How does this compare to a CNN, and what are the pros and cons of this?

(e) 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.

(f) (Pooling) The original ViT described above is almost exactly the same as BERT. In particular, each image is divided into 16×16 "words" [1], and a single special token `<CLS>` is appended to the start. This then allows us to use a small layernorm-linear-softmax module on the output vector of `<CLS>` to predict the probability of the given image being in the categories. Equivalently, we can remove the softmax to predict the logits.

More recently, it has been shown that the special token is unnecessary, and better performance can be achieved by simply pooling the output vectors of the tokens. Let $x_1, \ldots, x_{256}$ be the 256 input image patches, and let $y_1, \ldots, y_{256}$ be the 256 output vectors. The task of pooling is to compute a single y from the 256 output vectors.

[1] Thus the title "An Image is Worth 16x16 Words"

- Global Average Pooling (GAP) is essentially the same as average pooling in convolutional networks. What is y if we use GAP?
- Multiheaded Attention Pooling (MAP) is essentially using a query vector to "query" the output vectors in an attention mechanism. Let the query vector be q. What is y? What are the learned parameters? What should be the type of attention?

(g) So far the ViT training procedure has been fully-supervised. Yet we know that most of the available data are unlabelled. How can we do BERT-style self-supervised **representation** learning with vision transformers?

- *Hint 1*: Think about how BERT is trained. How should the input image be modified? What should be the target?
- *Hint 2*: ViT in this question only has an encoder. For BERT-style training, you will need a decoder. Why is this the case?

(h) (Bonus) If you wanted to add a few convolutional layers into ViT, how would you incorporate them?

(i) (Bonus) How would you use ViT to do GPT-style autoregressive generation of images?

