Data 188 Introduction to Deep Learning
Spring 2026 Eric Kim

Discussion 10

This week we will cover Visual Transformers.

1. Decoder "Shift Right"

In this question, we will learn why it is important for encoder-decoder models to use *right-shifted* decoder inputs, i.e. why we need to prepend a `<BOS>` token to the decoder input sequence like below:

```
X_src = ["I", "like", "to", "sing"]
X_tgt = [<BOS>, "J'aime", "chanter"]
Y_tgt = ["J'aime", "chanter", <EOS>]
```

where `X_src` is the encoder input, `X_tgt` is the decoder input, `Y_tgt` are the prediction targets, and `<BOS>` and `<EOS>` are the "beginning of sequence" and "end-of-sequence" tokens, respectively.

For simplicity, assume that the above `X_src`, `X_tgt`, `Y_tgt` are already tokenized, i.e. the above are string representations of integer token IDs, something like: `{"I": 829, "like": 9001, "to": 642, "sing": 543, "J'aime": 10282, "chanter": 12343, <BOS>: 1, <EOS>: 2, <PAD>: 0}`. In your answers for this question, you can just refer to tokens as their string representation.

Recall that in order to perform text generation, the decoder is trained to autoregressively predict the next token in the target sequence. In other words, the decoder repeatedly predicts one token at a time, taking the output of the previous task as the input for the next task.

(a) Fill in the blanks in the table below **without** using right-shifted decoder inputs (i.e. if we set `X_tgt = Y_tgt = ["J'aime", "chanter", <EOS>]`).

Task	Decoder Input Tokens	Target Prediction Token
1		
2		
3		

(b) Fill in the blanks in the table below **with** right-shifted decoder inputs.

Task	Decoder Input Tokens	Target Prediction Token
1		
2		
3		

(c) Observe the differences between the two tables above. Why is it important that we prepend `<BOS>` to `X_tgt`, and not to `Y_tgt`?

2. Cross-Token Interactions

(a) Suppose I removed the multi-head self attention (MHA) blocks from a transformer encoder, and replaced it with a no-op (say, an Identity layer). What remains are blocks like: position encoding, Linear layers, LayerNorms, residual connections, and MLP/FFNs. Will this "hollowed out" transformer encoder be able to learn cross-token interactions? Why or why not?

(b) Aside from MHA, what is another way of calculating cross-token interactions? **Hint:** Use an MLP, but on a modified version of the input token embeddings `X_src` (with original shape `[n_src, d]`).

3. Attention Cost

Recall that the cost for a naive transformer implementation scales quadratically with the input sequence length. Which transformer component primarily contributes to this quadratic cost?

4. Visual Transformer

(a) How do we pass an input image into a visual transformer encoder?

(b) Suppose I wanted to pass higher-resolution images to my visual transformer model. I double the input image resolution (ex: $224 \times 224 \rightarrow 448 \times 448$), but keep the same patchify step. This results in quadrupling the input sequence length. Will I be able to run inference with the larger 448×448 images on a model trained on the smaller 224×224 images?

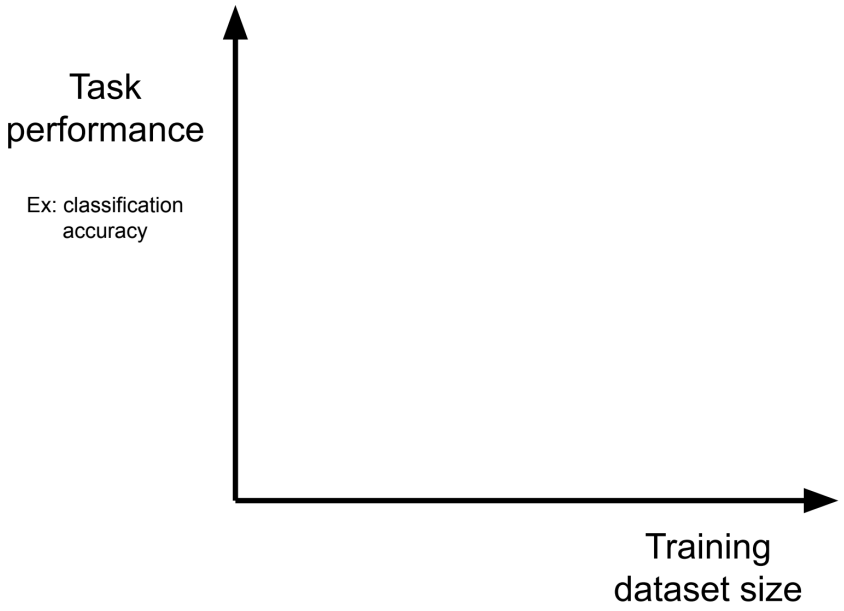
(c) Recall that the *inductive bias* of a model architecture is the space of possible hypothesis functions that a particular model architecture restricts itself to. The **Visual Transformer** model is said to have fewer inductive biases than ConvNets. Describe why visual transformer models can learn both spatially-global and local features, while ConvNets learn only spatially-local features.

5. Large-Scale Pretraining

(a) Recall that the Visual Transformer paper pretrained on a separate larger image classification dataset (e.g. JFT-300M) before finetuning on ImageNet-1k image classification.

In this case, the pretraining task and finetuning task are the same: image classification. What is an example of an effective pretraining strategy where the pretraining task is **different** from the finetuning task?

(b) Consider the following chart. Note that a common (yet imperfect) proxy for model capacity is the number of model parameters.



Based on learnings from the field (and this class), draw the expected behavior of (1) high-capacity models like vision transformer models and (2) lower-capacity models like ConvNets in the chart. Your answer should have two curves, one for each type of model. Describe what's going on and why. How does inductive bias fit into this?

Contributors:

- Eric Kim.
- Rebecca Dang.

