Conditional Text Generation Based on Conditional Layer Normalization

Su Jianlin

December 14, 2019

From the article "From Language Models to Seq2Seq: Transformer is all about Masking", we know that as long as it is paired with an appropriate Attention Mask, BERT (or other Transformer models) can be used for unconditional generation (Language Model) and sequence translation (Seq2Seq) tasks.

But what about conditional generation? For example, controlling the category of the text to generate text randomly according to a category (Conditional Language Model); or passing in an image to generate a related text description (Image Captioning).

1 Related Work

The paper "Encoder-Agnostic Adaptation for Conditional Language Generation" published in August systematically analyzed several schemes for using pre-trained models for conditional generation. In September, a paper "CTRL: A Conditional Transformer Language Model for Controllable Generation" provided a pre-trained model based on conditional generation, though this is essentially a language model like GPT that can only take text input as a condition. More recently, the paper "Plug and Play Language Models: a Simple Approach to Controlled Text Generation" explored conditional generation based on pre-trained models by converting $p(x|y)$ into $p(x)p(y|x)$.

However, these classic works are not what this article aims to introduce. This article focuses on the scenario of text generation using a fixed-length vector as a condition, and the method is **Conditional Layer Normalization**—incorporating the conditions into the β and γ of Layer Normalization.

2 Method Details

The idea of Conditional Layer Normalization stems from the popular conditional GAN approach in image processing—Conditional Batch Normalization (Conditional BN); related content can be found in "An Overview of GAN Architecture Development: From DCGAN to SELF-MOD". There is also a variant of Conditional BN called AdaIN (Adaptive Instance Normalization). Both Conditional BN and AdaIN turn the β and γ in existing Normalization methods into functions of the input condition, thereby allowing the condition to control the generation behavior.

In Transformer models like BERT, the primary Normalization method is Layer Normalization. Therefore, it is natural to think of turning the corresponding β and γ into functions of the input condition to control the generation behavior of the Transformer model. This is the core idea of Conditional Layer Normalization. (However, I have not yet seen other work using this same approach, so this can be considered a fresh idea developed behind closed doors.)

For a model that has already been pre-trained, there are already existing, unconditional β and γ, which are fixed-length vectors. We can use two different transformation matrices to transform the input condition into the same dimension as β and γ, and then add the two

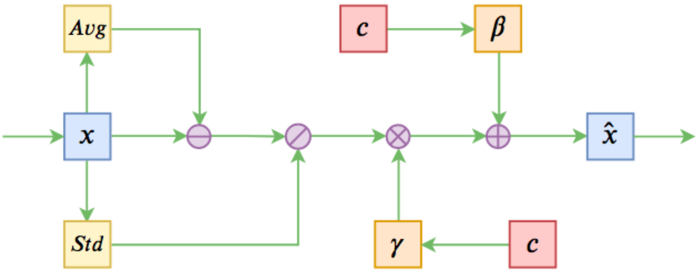

Figure 1: Schematic diagram of Conditional Normalization

transformation results to β and γ respectively. To prevent disturbing the original pre-trained weights, the two transformation matrices can be initialized to all zeros (a single-layer neural network can use zero initialization; only continuous multi-layer neural networks should not). Thus, in the initial state, the model remains consistent with the original pre-trained model.

3 Code Implementation

Intuitively, this type of fine-tuning for text generation should require auto-regressive pre-trained models like GPT to improve effectiveness. However, as shown in the previous article "From Language Models to Seq2Seq: Transformer is all about Masking", even if you load BERT's pre-trained weights for generation tasks, the performance remains good. Therefore, regardless of which Transformer-based pre-trained model is used, it can be considered for fine-tuning as a text generation model. This article uses pre-trained BERT as the base model for experiments.

As for the code, the Conditional Layer Normalization technique described in this article has been integrated into the bert4keras library developed by the author. The base function `build_transformer_model` now includes the following parameters:

1. `layer_norm_cond`: If this parameter is not `None`, it means it is a tensor with `shape=[batch_size, cond_size]`, used as the condition for Layer Normalization.

2. `layer_norm_cond_size`: If this parameter is not `None` and `layer_norm_cond` is `None`, it means it is an integer; an input layer with `shape=[batch_size, layer_norm_cond_size]` will be constructed automatically as the condition for Layer Normalization.

3. `layer_norm_cond_hidden_size`: If this parameter is not `None`, it is an integer used to first project the input condition into a lower-dimensional space. This is because the input condition might have high dimensionality; projecting directly to `hidden_size` (e.g., 768) might involve too many parameters, so it can be projected to a lower dimension first and then up-sampled.

4. `layer_norm_cond_hidden_act`: The activation function used when projecting to the lower-dimensional space. If `None`, no activation function is added (linear activation).

5. `additional_input_layers`: Additional input layers. If a tensor is passed externally as a condition, all input layers that the condition tensor depends on must be added as input layers to build the final model.

4 Experimental Results

No matter how much is introduced, it is better to look at actual examples. I conducted two experiments to verify the effect of Conditional Layer Normalization. One is controlling text generation through sentiment polarity (the inverse problem of sentiment classification), where the class Embedding is used directly as the condition for Layer Normalization. The other is Image Captioning, where a pre-trained ImageNet model encodes the image into a fixed-length vector as the condition for Layer Normalization.

The code for these two experiments is located in task_conditional_language_model.py and task_image_caption.py.

4.1 Sentiment Text Generation

The sentiment text generation uses a sentiment classification corpus I previously collected, using the input text and labels in reverse. Finally, random sampling based on probability is used during generation to produce diverse texts.

Partial output:

Positive Samples:

- Stylish appearance, beautiful, high cost-performance ratio.
- Beautiful appearance, balanced configuration, quite satisfied, high cost-performance ratio, beautiful appearance, high performance.
- I saw this book when I was in university, so I have been buying it. The author, Lin Jinglei, uses her own tone to write about the mental journey of a child's growth, letting me see the differences in their growth and the different realms of their growth process. I really appreciate it!
- I think this is a book that can tell readers what is bad, rather than teaching you how to speak or telling me what is wrong. Here I recommend "I Want to Tell a Story". This is a book I like very much. I think there are many reasons for it, but I believe in myself. If you get some improvement from it, or you have already made a wise decision.
- Our family of five stayed in a standard room, a king room. The king bed was very comfortable; we booked two king rooms on Ctrip. The price of this hotel is quite reasonable; however, the soundproofing of the room is not ideal, there is some noise; the subway at the hotel entrance is under construction, which is inconvenient; but I don't know which car the taxi at the hotel entrance belongs to, taking a taxi is not very convenient; the parking outside the hotel...

Negative Samples:

- I don't know if it's because the battery is not very good, it's not that I don't like it.
- I bought it only after reading the reviews. As a result, I found it wasn't that cheap, and the price wasn't cheap either.
- 1. The shell is not easy to get fingerprints on, but not easy to wash. 2. The screen is a bit old and cannot download ringtones.
- I ordered "Du Lala's Promotion" on July 6th and paid via bank. Why has the order not arrived after more than two weeks? Is the delivery time too fast, maybe it just passed like that?
- I read this book online first, and then I read it again. I feel the author's writing is really bad, especially when writing his blog, it's very awkward and unprofessional. Especially when he writes about the little boy who regulates emotions when trading stocks, it's just a self-righteous look, a manifestation of being self-righteous!

4.2 Image Captioning

For Image Captioning, the COCO dataset is used as an example, as the image scenes in this dataset are quite rich. Additionally, the 2017 challenger.ai held an Image Chinese Description Generation Competition, which also included a good dataset (readers can find ways to collect it themselves), though the image scenes are relatively more monotonous.

Partial output:

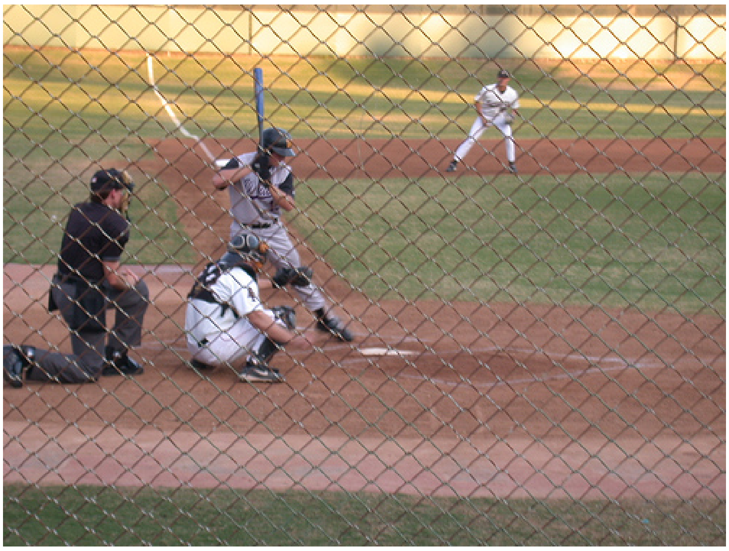

Figure 2: Model prediction: a baseball game in progress with the batter up to plate.

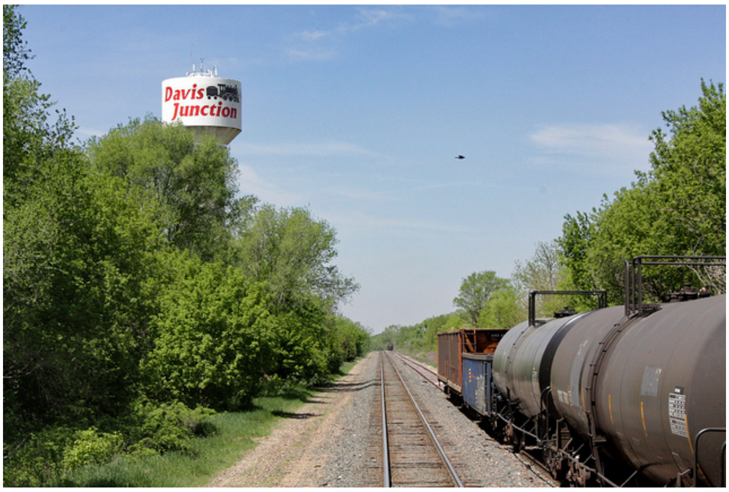

Figure 3: Model prediction: a train that is sitting on the tracks.

image_id: COCO_val2014_000000524611.jpg
url: http://images.cocodataset.org/val2014/COCO_val2014_000000524611.jpg
predict: a train that is sitting on the tracks.
references: [u'A train carrying chemical tanks traveling past a water tower.', u'Dual train tracks with a train on one of them and a water tower in the background.', u'a train some trees and a water tower ', u'Train on tracks with water tower for Davis Junction in the rear.', u'A train on a train track going through a bunch of trees.']

image_id: COCO_val2014_000000202923.jpg

url: http://images.cocodataset.org/val2014/COCO_val2014_000000202923.jpg
predict: a baseball game in progress with the batter up to plate.
references: [u'Batter, catcher, and umpire anticipating the next pitch.', u'A baseball player holding a baseball bat in the game.', u'A baseball player stands ready at the plate.', u'Baseball players on the field ready for the pitch.', u'A view from behind a mesh fence of a baseball game.']

5 Summary

This article proposes the idea of using Conditional Layer Normalization to integrate external conditions into pre-trained models. Its direct application is conditional text generation, but it is not limited to generative models; it can also be used in scenarios like classification models (where external conditions might be information from other modalities to assist classification). Finally, a code implementation and two examples based on `bert4keras` are provided.