第2章 トークナイズ/エンベディング/アテンション/出力の生成

大規模言語モデルTransformerの動作

新谷 俊了

OpenAIが提供しているChatGPTが，高性能な対話が可能な大規模言語モデル(LLM：Large Language Model)であったことを受けて，言語にとどまらずさまざまな生成AIをどのように活用するかが急速に検証されています．ここではChatGPTなどのLLMがどのようにして動いているかを実際のモデルの入出力を追っていくことにより理解していきます．

Transformerの構造

LLMはOpenAIからだけではなく，日本語ではrinna，サイバーエージェントなどの企業からさまざまなモデルが公開されています．近年公開されているLLMの多くはTransformerと呼ばれる機構をベースにしています．

Transformerは，「Attention Is All You Need」という論文で2017年に発表されました[1]．タイトルの通りattention(アテンション)という機構を導入し，効率的に入力文字列の関係を計算できるようになりました．

● エンコーダとデコーダの2つからなる

Transformerはエンコーダとデコーダという2つの構造を持っています(**図1**)．今は分かりやすさのために1層のみとしてますが，本来は何層にも積み上がった形となっています．ここで，デコーダとエンコーダの主な違いは，デコーダはエンコーダによって処理された入力のアテンション情報を加味する点です．

アテンションは3カ所存在し，エンコーダ/デコーダへの入力から計算するセルフ・アテンションと，デコーダの入力にエンコーダの計算結果を反映するクロス・アテンションという2種類が存在します．

● 機構におけるGPT系との違い

Transformerで導入された機構と，現在よく使われているGPTの機構は少し異なります[3]．大きな違いとしてTransformerでは，入力を処理するエンコーダ部分とデコーダ部分から構成されていますが，

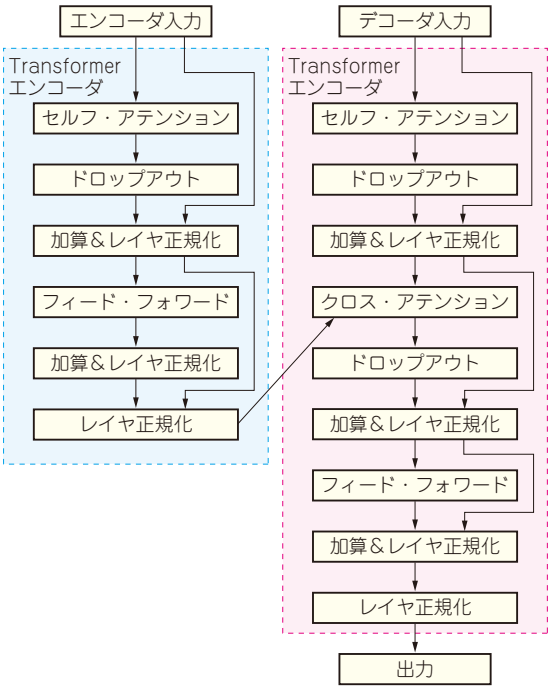

図1　PyTorch実装のTransformersモデルをtorchview[2]によって可視化し加工したネットワーク図

GPTではデコーダ部分のみを使っています．

デコーダのみのGPTでは，クロス・アテンションは省かれており，セルフ・アテンションのみで構成されています．デコーダの特徴としては，計算する際にその時点より前のデータのみを見る点にあります．このことから，GPTなどのデコーダのみのモデルは入力に対して続きの文章を生成することが得意な構造となっています．