情報工学英語演習 「AttentionIsAllYouNeed」の和訳

Abstruct

今までの主要な sequence transduction model は, エンコーダーとデコーダーを含む複雑な RNN や CNN に基づいていた. 最も良いパフォーマンスであったモデルも, エンコーダーとデコーダーをアテンションの仕組みを用いて接続しているモデルであった.

私達は,RNN や CNN を用いず, アテンションのみからなる Transformer と呼ばれる新しい簡潔なモデルを提案する.2 回の翻訳実験の結果,Transformer は今までのモデルより優れており, 更には, より並列化が可能で学習の時間も少ないことが分かった. Transformer は WMT2014 英独翻訳タスクにおいて,「プロの翻訳者の訳と近ければ近いほどその機械翻訳の精度は高い」という考え方に基づく機械翻訳の評価方法である BLEU スコアで 28.4BLEU を記録した. これは, 複数のモデルを融合させて 1 つの学習モデルを生成するアンサンブル学習を含めたこれまでの最高記録を 2BLEU 上回る結果であった. また、WMT2014 英仏翻訳タスクにおいては、8 個の GPU を用いた 3.5 日の学習というこれまでの最先端のモデルの学習よりも遥かに少ないコストで,41.0BLEU という単一モデルの最高記録を打ち立てた.

1 Introduction

RNN, 特に RNN において文章の長期的な依存関係を学習できるようにした LSTM や gated RNN は, 言語モデルや機械翻訳などの Sequence 問題への最適な手法として確固たる地位を築いていた. それ以来,Recurrent 言語モデルとエンコーダー-デコーダー構造の限界を押し上げる数々の努力がなされてきた.

リカレントモデルでは, 通常, 入力と出力の時系列データの時間的な位置に沿って計算を行う. よって計算は逐次的に行われ, 時刻 t における隠れ状態 h_{t} は, 時刻 $t-1$ の隠れ状態の $h_{\mathrm{t}-1}$ と時刻 t における入力から導かれる. このように本質的に逐次的な性質を孕んでいるため, 学習の並列処理が困難である. そのため, メモリの制約上, 長い時系列データなどの学習には致命的であった. 直近の研究では factorization tricks や conditional computation といった方法で計算効率がかなり改善され, 後者ではモデルの性能まで向上させることができたが, 逐次的な計算の問題は残ったままだった.

Attention は入力と出力の時系列データにおける距離を気にせず依存関係をモデル化することができ, 様々なタスクにおいて有効な sequence model と transduction model の必要不可欠な部分となっている. しかし, 一部の場合には Attention は RNN と合わせて用いられる.

本研究で私達が提案する Transformer は,RNN を用いず,Attention のみで入力と出力の完全な依存関係を取り出すモデルのアーキテクチャである. Transformer は学習の並列処理が可能であり,8 個の P100 GPU で 12 時間という小規模な学習後に, 最高の機械翻訳性能に達することができた.

2 BackGround

逐次的な計算を減らすという目標は,Extended Neural GPU,ByteNet,ConvS2S といったモデルの基礎にもなっている. これらはどれも CNN を基本構成要素として, 入力と出力のすべての位置で隠れ状態の値を計算する. またこれらのモデルにおいて, 任意の入力の位置と出力の位置の信号を関連付けるために必要な計算時間は,ConvS2Sd では線形的,ByteNet では指数的となる. そのため離れた位置の依存関係を学習することはより困難になる. Transformer では, この計算時間を定数時間に減らすことができる.Attention で重み付けした位置を平均化することで有効な解像度が下がってしまうが,3.2 説で述べる Multi-Head Attention により相殺できる.

intra-attention とも呼ばれる Self-Attention は, 単一の文章の異なる位置を関連付ける Attention である.Self-Attention は文章読解, 要約, テキスト含意, 独立した文の表現の学習などのタスクで用いられ成功している.

End-to-End memory Networks は RNN の代わりに再帰的な Attention を元にしており, 単純な言語の質疑応答, 言語モデリングといったタスクにおいて優れた結果を示している.

しかし私達が知る限り,Transformer は RNN や

CNN を用いずに入出力の表現を計算するために,Self-Attention のみに依存した最初の transduction model である. 次節以降では,Transformer,self-attention について説明しこれまでのモデルと比較した利点を議論する.

3 Model Architecture

最も優位性のある sequence transduction models はエンコーダー-デコーダー構造を有している. エンコーダーは配列で表現される入力 $(x_1,...,x_n)$ を配列 $\boldsymbol{z}=(z_1,...,z_n)$ に変換する. デコーダーは $\boldsymbol{z}$ から出力として配列 $(y_1,...,y_n)$ を 1 要素ずつ出力する. このステップで, モデルが生成する要素はこれまでに生成した要素のみに依存する自己回帰モデルであり, 直前に生成された要素を新しく入力として次の要素を生成する.

Transformer は図 1 の左半分と右半分に示すように, 全体としてはエンコーダー-デコーダー構造を踏襲しつつ,self-attention 層と point-wise 全結合層を積み重ねた層を使用している.

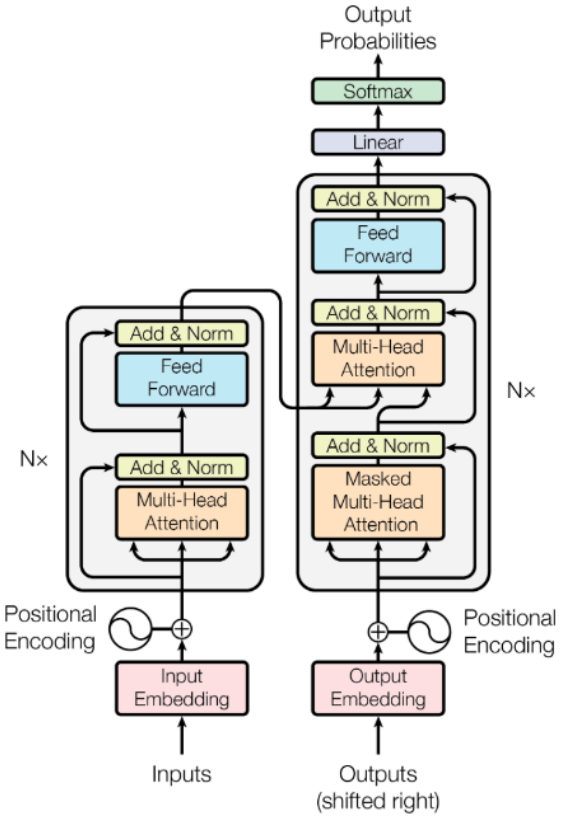

図 1: Transformer のモデルアーキテクチャ

3.1 Encoder and Decoder Stacks

エンコーダー: 6 層からなり, 各層は全く同じ構造である. それぞれの層は 2 つの下位層を持ち, 下位層の後には残差接続や標準化が行われている. よって, 下位層自身の出力を Sublayer(x) として, 下位層全体としての出力は LayerNorm$(x$+Sublayer$(x))$ となる. 残差接続を容易にするために, すべての下位層,embedding layers も出力の次元を d_{model}=512 としている.

デコーダー: デコーダーも同一の 6 層からなる. エンコーダーの 2 つの下位層に加えて, エンコーダーの出力を入力として受ける 3 つ目の下位層を加えている. エンコーダーと同じく, 下位層の後には残差接続や標準化が行われている. ただ,self-attention 下位層は改良しており, 後続の要素が影響しないようにしている. このマスキングと, 出力が 1 要素ごと補われることを組み合わせることで, マスキングにより i での予測が i 未満での既知の出力のみに依存することが保証される