ISSN 2186-5647

P-13

Neural Turing Machine with GRU Controller

日大生産工（学部）　○石橋　駿　日大生産工　山内 ゆかり

1 まえがき

Recurrent Neural Network(RNN) [1]は隠れ層に再帰的な構造を持っていて、他のNeural Networkとは違い時系列データを扱うことができるモデルである。そのため自然言語処理や音声認識などのタスクで用いられている。

シンプルなRNNは時系列データが長くなってくると学習が難しくなる。そのためRNNを拡張したLong Short-Term Memory（LSTM）[2]やGated Recurrent Unit(GRU) [3]といったモデルがある。

LSTM は RNN の隠れ層のユニットを LSTM block と呼ばれるメモリと 3 つのゲートを持つブロックに置き換えたものである。3 つのゲートは入力ゲート、出力ゲート、忘却ゲートと呼ばれる。入力ゲート及び出力ゲートは入力重み衝突・出力重み衝突を解決し必要な誤差信号だけを適切に伝搬するためのもの。忘却ゲートはメモリに記憶した内容を忘れることを学習する。

GRU は LSTM が 3 つのゲートで構成されていたのに対して、更新ゲートとリセットゲートの 2 つのゲートによって構成される。更新ゲートは最後の状態からどれだけの情報を保持し、前の層からどれだけの情報を取り込むかを決定し、リセットゲートは LSTM の忘却ゲートのように機能する。基本的には LSTM のように機能するが GRU の方が学習の収束が速い。

一般的にコンピュータプログラムは演算、論理的な制御、外部メモリの 3 つの基本的なメカニズムを有する。しかし Neural Network をはじめとした機械学習は論理的な制御と外部メモリの使用をほとんど無視してきた。

そこで DeepMind は RNN と相互作用できる外部メモリを結合した Neural Turing Machine(NTM) [4]を考案した。

NTM はチューリングマシンやノイマン型コンピュータと似た構造をしているが微分可能なエンドツーエンドであり、勾配降下法によって訓練することが可能である。

従来研究ではNeural NetworkにLSTMを用いているが、本研究ではGRUを用いることで学習の収束が速くなることを期待する。

2 従来研究

2.1 NTM にはニューラルネットワークコントローラと外部メモリの二つの基本コンポーネントが含まれている。図 1 は NTM の高レベルの図を示したものである。ほとんどのニューラルネットワークと同様にコントローラは入力ベクトルと出力ベクトルを介して外部世界と相互作用する。また標準的なニューラルネットワークとは異なり、選択的な読み出し及び書き込み動作を使用して外部メモリと相互作用する。これらの操作はチューリングマシンと同様にヘッドによって実現する。

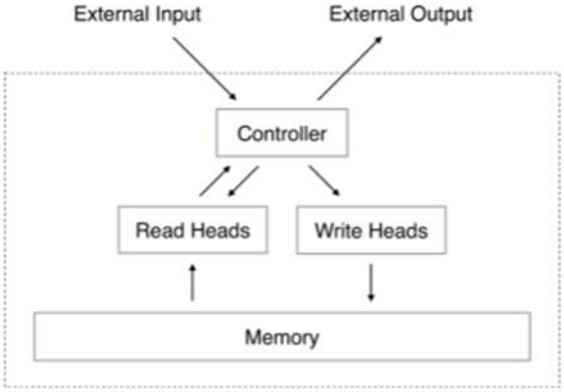

図 1 NTM

2.2　読み取り・書き込み

外部メモリのサイズは N*M である。N はメモリ位置を表し M は各位置におけるベクトルサイズである。また外部メモリへアクセスはメモリのどこに注目すべきかを表す分布である attention との加重合計によって求められる。

読み取り時のベクトル r_t は、attention w_t

Neural Turing Machine with GRU Controller
Shun ISHIBASHI and Yukari YAMAUCHI

とメモリ内の行ベクトル M_t を使って式(1)で求められる。

$$r_t \leftarrow \sum_i w_t(i) M_t(i) \tag{1}$$

書き込みは消去と追加の二つに分解することで可能としている。

消去は消去ベクトルe_t、attention w_t、前の時間ステップでの行ベクトルM_{t-1}を使って式(2)によって求められる。

$$M_t \leftarrow M_{t-1}(i)[1 - w_t(i) e_t] \tag{2}$$

追加は追加ベクトルa_t、attention w_t、消去後の外部メモリの行ベクトルM_tを使って式(3)で求められる。

$$M_t(i) \leftarrow M_t(i) + w_t(i) a_t \tag{3}$$

2.3. attention 調整

attention はコンテンツベースのアドレッシングと位置ベースのアドレッシングの 2 つのアドレッシングを組み合わせることによって求められる。またパラメータはコントローラの出力する 5 つのパラメータと直前のメモリ、ヘッド重みを用いる。図 2 は attention 調整のステップを表す。

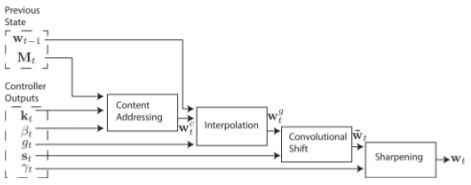

図 2 attention 調整

最初にコンテンツベースのアドレッシングを行う。ここでは式(4)のようにコントローラから与えられたベクトル k_t とメモリ内の各ベクトルの類似度を計算し、その類似度をもとにβ_tで正規化してヘッドの attention を初期化する。

$$w_t^c(i) \leftarrow \frac{\exp(\beta_t K[k_t, M_t(i)])}{\sum_j \exp(\beta_t K[k_t, M_t(j)])} \tag{4}$$

次に行うのが位置ベースでのアドレッシングである。ここでは式(5)のようにコンテンツベースのアドレッシングで出力された attention と直前の attention に対してパラメータ g_t を掛け合わせ attention を更新する。g_t が 0 の場合、コンテンツベースの attention が無視され、1 の場合は直前の attention が無視される。これにより直前の attention をどれだけ引き継ぐか新しい attention をどれだけ採用するかを調整する。

$$w_t^g \leftarrow g_t w_t^c + (1 - g_t) w_{t-1} \tag{5}$$

Convolution Shift では位置ベースのアドレッシングで出力された重みと方向パラメータ s_t を畳み込むために式(6)を使って attention を更新する。これによりコントローラが集中うする位置を変更することができる。

$$\tilde{w}_t(i) \leftarrow \sum_{j=0}^{N-1} w_t^g(j) s_t(i-j) \tag{6}$$

Sharping では Convolution Shift で出力された重みに対して、集中度パラータ γ_t を用いて式(7)により鋭くして最終的な attention を生成する。

$$w_t(i) \leftarrow \frac{\tilde{w}_t(i)^{\gamma_t}}{\sum_j \tilde{w}_t(j)^{\gamma_t}} \tag{7}$$

2.4. コントローラ

NTM における最も重要なアーキテクチャの選択はコントローラに使用されるニューラルネットワークのタイプである。LSTM などのリカレントコントローラには外部メモリより大きなメモリを補うことができる独自のメカニズムがあり、コントローラをコンピュータの中央処理装置、外部メモリを RAM とするとリカレントネットワークの隠れ層はレジスタに似ている。これによってコントローラは複数の時間ステップで情報を混合することができる。またフイードフォワードコントローラは同時読み書きヘッドの数が NTM が実行できる計算のタイプにボトルネックを課してしまうが、リカレントコントローラは、以前のタイムステップからの読み出しベクトル

を内部的に格納することができるのでその制限を受けない。

3　提案手法

本研究では、コントローラに使用するNeural Network を GRU としたモデルを提案する。

GRU は LSTM に比べて、パラメータが少なく計算量も少ない、そのため学習の収束が速いモデルである。

そのため LSTM Controller よりも学習の収束が速くなることを期待するとともに同等のパフォーマンスを期待する。

4　実験環境

本実験では、8ビットのランダムベクトルのシーケンスに対するコピータスクを行う。シーケンスの長さは1から5または1から10の間で決める。

実験対象とするモデルは従来研究のLSTMをコントローラとしたものと提案手法のGRUをコントローラとしたものとする。

学習はバッチサイズを10とし、100回学習するごとに誤差と正解率を求めた。また5000回学習するごとにNTMの重みを保存する。全体的な学習回数としては20000回とする。

実験結果にはLSTM Controller と LSTM Controller に対して誤差(loss)と正解率(accuracy)をそれぞれシーケンス長が5の場合と10の場合に分けてグラフにする。またグラフは5ステップずつ平均化する。

5.　実験結果

5.1.　シーケンス長が5の場合(loss)

図3のLSTM ControllerとGRU Controllerの誤差の学習曲線を比較すると学習の初期はGRUの方が収束が速かった。

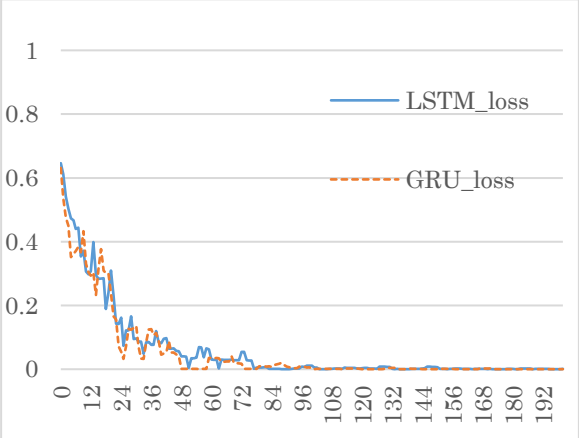

図 3 シーケンス長が 5 の loss

5.2.　シーケンス長が5の場合(accuracy)

図4のLSTM ControllerとGRU Controllerの正解率の学習曲線を比較すると学習の初期はGRUの方が収束が速かった。

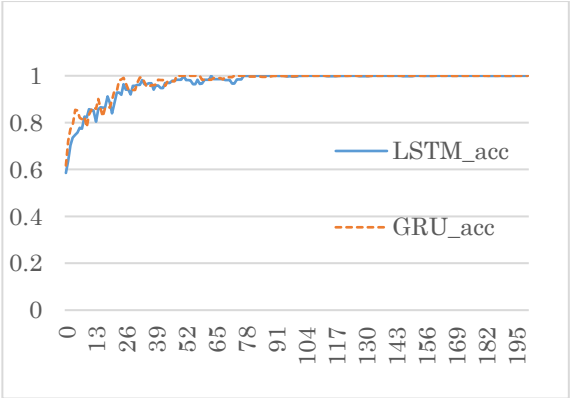

図 4 シーケンス長が 5 の accuracy

5.3.　シーケンス長が10の場合(loss)

図5のLSTM ControllerとGRU Controllerの誤差の学習曲線を比較するとGRUの方がばらつきが大きく、70当たりを境にLSTMのlossが下がって安定していくのに対してGRU はlossが下がっていかなくなった。

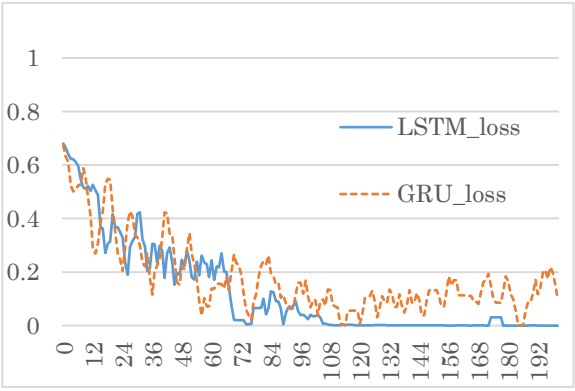

図 5 シーケンス長が 10 の loss

5.4.　シーケンス長が10の場合(accuracy)

図6のLSTM ControllerとGRU Controllerの正解率の学習曲線を比較するとGRUの方がばらつきが大きく、70当たりを境にLSTMのaccuracyが上がって安定していくのに対してGRU はaccuracyが上がっていかなくなった。

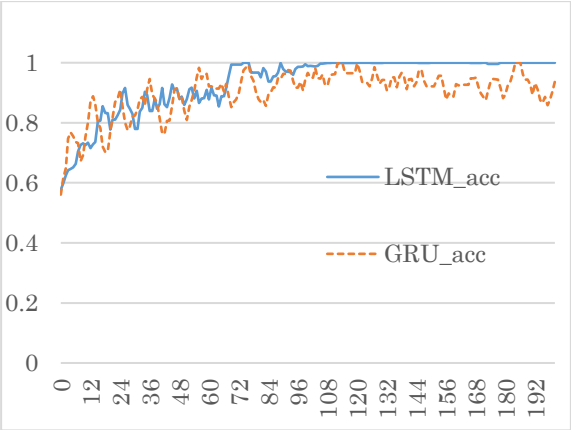

図 6 シーケンス長が 10 の accuracy

6. まとめ

本研究ではNTMのコントローラにLSTMではなくGRUを用いることで学習の収束を速くすることを目的とした。

実験結果を比較した結果、誤差に関してはシーケンス長が5の場合は学習の初期はGRUの方が収束は速かったが、シーケンス長を10とした場合はGRU Controllerはバラツキが大きく誤差が下がりにくくなった。

同様に正解率に関してもシーケンス長が5の場合はあまり差が見られなかったがシーケンス長が5の場合は学習の初期はGRUの方が収束は速かったが、シーケンス長を10とした場合はGRU Controllerはバラツキが大きく正解率が上がりにくくなった。

よってGRUをControllerとするのはLSTMをControllerとするのに比べて長い時系列データに対するタスクを学習するのは難しいものと考えられる。

提案手法の今後の課題としてはなぜGRU Controllerの方が長期間の記憶が難しくなるのかを考察したいと思う。また今回はまだ学習のみなので、テストを行いたいと思う。テストでは学習に用いたシーケンス長より長いシーケンスに対してテストし比較を行いたいと考えている。

参考文献」

1) Siegelmann, H. T. and Sontag, E. D. (1995). On the computational power of neural nets. Journal of computer and system sciences, 50(1):132–150.
2) S. Hochreuter, and J. Schmidhuber. Long short-term memory. Neural Computation, 9(8), pp.1735-1780, 1997.
3) K. Cho, B. Merrienhoer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. Proceedings of the Empiricial Methods Language Processing(EMNLP 2014), 2014.
4)