夏季進捗レポート

情報科学類 3 年知能情報メディア主専攻 江畑 拓哉 (201611350)

1 概要

1. deeplearning4j の理解
2. LSTM 式 qna システムの作成とその評価
3. データセットの作成
4. CopyNet, HRED の理解
5. seq2seq attention の理解
6. word2vec の理解と実装
7. doc2vec の利用方針
8. 文体変換について
9. 全体的なシステムの想定図

2 deeplearning4j の理解

Java で書ける機械学習フレームワーク [2] 。Tensorflow が速度と記法、記述量として不満があったのでこちらに移行した。(5 分の 1 程度の記述量になる。またオブジェクト指向中心のためコードの再利用が楽。)
また同時に比較として PyTorch (ドキュメント次第では Tensorflow) の実行結果を用いていくつもりである。(自然言語処理などではこちらの方が記述しやすい。)

3 LSTM 式 qna システムの作成とその評価

このシステムに注目したのは ユーモアを理解できるチャットボットについての論文 [1] を読んだため、そして deeplearning4j を使った自然言語処理の例として CNN 式の qna システムがあったためです。

前者では、ユーモアを理解しそれにより感情パラメータを変化させるという実験を行っています。そしてこのシステムではユーモアの理解のために AIML というマークアップ言語を用いて文章の完全ないし一部の一致検索をしている。そして文章の極性 (negative or positive) を判定し、それに基づいた反応 (文字、アバターの表情) をさせている。

後者では以下に説明する qna タスクを VAE (Variational AutoEncoder) を用いて解決しようとしている。しかし同様のコードを作成し実行した所、芳しい結果を得られなかった 1 。そのため、Seq2Seq の考え方を利用して Seq2Label という手法を提案し 3 実験 [1] 、その比較を行った。

結果としてかなりの精度を得ることが出来た。

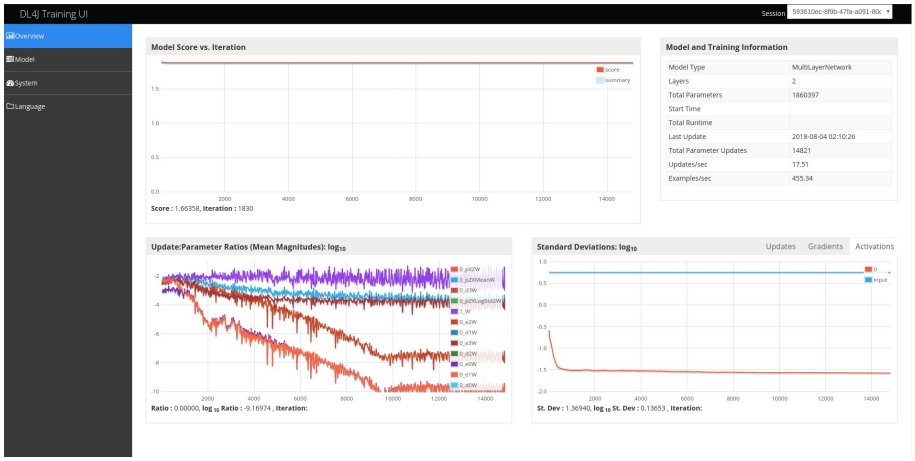

Figure 1: 上手く行かなかった例

しかしいくつかの実験を行った結果、このモデルの欠陥をいくつか発見した。また日本語特有の問題もあるが、これについては 4 で説明する。

- モデルが収束しすぎる。
  余りにも質問文の内容が乖離しすぎていると深層学習する必要がなくなってしまう。また誤差逆伝搬法で問題が起きる (NaN 値を発生してしまう。)

- 単語数が増えると空間計算量が増える。
  空間計算量を減らしすぎると今度はモデルが上手くフィットしない。

[1] https://github.com/MokkeMeguru/self_introduction

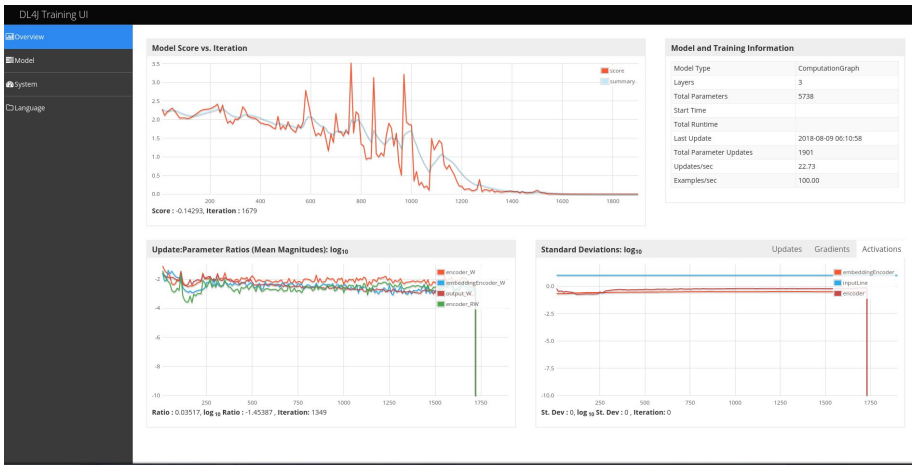

Figure 2: Seq2Label を使った例

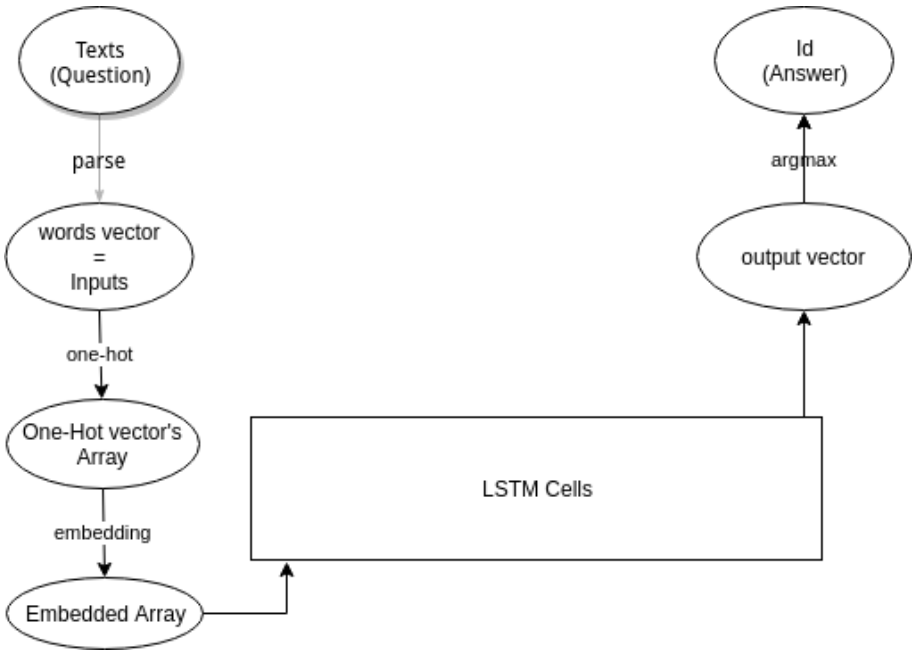

Figure 3: Seq2Label の概要

4 データセットの作成

日本語のデータセットを収集する必要があったが良質と思われる [2] データセットが見つからず、実際にデータセットを作成することにした。また時間があればこのデータセットと他のデータセットを比較してみようと考えている。[3]

作成しているデータセットは MIT ライセンスで公開する [4]ことで他の研究者からの意見を聞きたいと考えている。

また、最終的には Seq2Seq で使われる既存のデータセットの中で最小となっていた 3k を目指している。

文脈不明・理解不能な会話文が一切ないため、低品質なデータセットに比べてもそれなりの成果が得られると予想している。

5 Seq2Seq attention の理解

deeplearning4j では Seq2Seq が実装できず、 Seq2Seq attention [5] が実装でき、公式のサンプルがその実装例を紹介している。そのため、このソースコードの理解を以てこのモデルを理解したことになると考えている。[6]

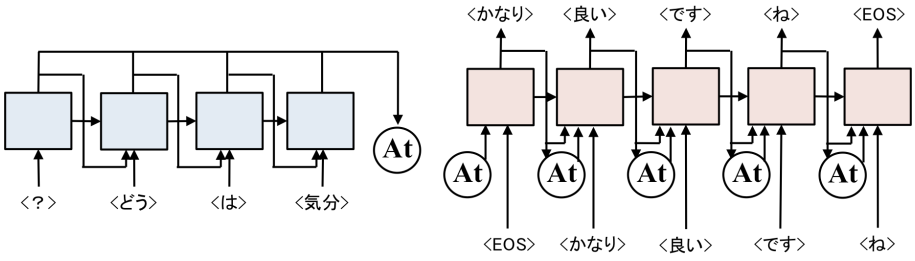

Figure 4: Seq2Seq Attention のモデル図

6 word2vec の理解と実装

こちらに関しては deeplearning4j が完全にサポートしており、文字通り一行でモデルを作ることが出来る。これを用いて、qna システム内 の Encoder 部分に適用した際の適

[2]例えば後述する CopyNet や HRED などでは Q&A ではなく、 **意味の通る** 長い会話データが必要となる。

[3]しかしデータの前処理によってもデータが大きく変化してしまうので実験は難しいと考えられる。

[4]https://github.com/MokkeMeguru/japanese-corpus

[5]https://qiita.com/kenchin110100/items/eb70d69d1d65fb451b67

[6]https://github.com/MokkeMeguru/seq2seq-memo

合率について比較を行いたいと考えている。(seq2seq ではいくつかの特殊な Token が存在するため用いることが出来ない。)[7]

7 doc2vec の利用方針

IBM のチャットボット Watson では質問文を細かくカテゴライズするという手法を取っている。[8]

データの面でのカテゴライズはデータセット生成の時点で済ませてあるものとしてみなすことが出来るため、問題となるのは入力後のデータ処理についてである。この、入力された後のデータ分類を doc2vec を用いて行いたいと考えている。この実装はデータセットが目標数に達成したところで、 deeplearning4j のドキュメントを参考に実装していく予定である。

8 文体変換について

文体変換についてはいくつかの手法を検討している。

例えば日本語で研究が行われた先行例として、転移学習を用いたもの [3] がある。

本研究ではこれに加えて、転移学習についていくつかの手法を提案する他、VAE を用いたパターン、Zero-shot 変換を用いたパターン、Seq2Seq attention (応答-¿応答) 、などを実験する予定である。

9 全体的なシステムの想定図

別紙を参照

参考文献

[1] Agnese Augello et al. *Humorist Bot: Bringing Computational Humour in a Chat-Bot System*. Apr. 2008.

[2] *deelearning4j*. Skymind. DL4J. 2018.

[3] 赤間怜奈. “対話生成における応答のスタイル制御に関する研究”. In: (Mar. 2017).

[7] `https://github.com/MokkeMeguru/word2vec_memo`

[8] `https://www.ibm.com/blogs/solutions/jp-ja/watson-machinelearning-2/` この実装と本研究の共通点は、どちらも「手で」データを選別 (生成) しているということである。