

進捗報告

情報科学類 3 年 江畑 拓哉 (201611350)

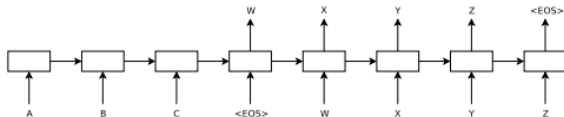
Outline

- 1 要約
- 2 Seq2Seq model について
- 3 LSTM について
- 4 使用するライブラリを変更した
- 5 新たに調べたモデル
- 6 実験したいモデル
- 7 今後の予定

要約 I

- Seq2Seq model について学習した。
- 使用するライブラリを変更した。(それに伴い使用言語も変更)
- 幾つかのモデルについて調べた。
- 実験したいモデル
- 今後の予定

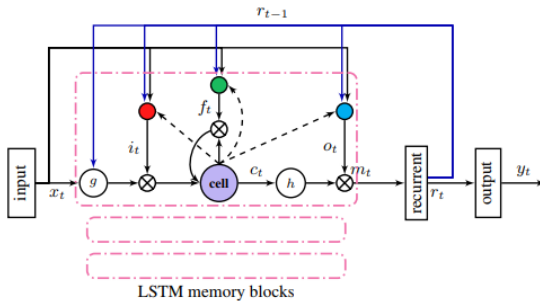
- 翻訳に多く使われている。
- Q & A 形式の一文が短い会話についても適用できる。
- モデル図



- 参考文献
CS 20SI: Tensorflow for Deep Learning Research
Deep_Learning_Based_Chatbot_Models

lstm I

- Long Short-Term Memory
- 概図



それぞれの要素に対する式 I

$$\mathbf{i}_t = \sigma(\mathbf{W}_{ix}\mathbf{x}_t + \mathbf{W}_{im}m_{t-1} + \mathbf{W}_{ic}c_{t-1} + \mathbf{b}_i) \quad (1)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{fx}\mathbf{x}_t + \mathbf{W}_{fm}m_{t-1} + \mathbf{W}_{fc}c_{t-1} + \mathbf{b}_f) \quad (2)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot g(\mathbf{W}_{cx}\mathbf{x}_t + \mathbf{W}_{cm}m_{t-1} + \mathbf{b}_c) \quad (3)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{ox}\mathbf{x}_t + \mathbf{W}_{om}m_{t-1} + \mathbf{W}_{oc}c_t + \mathbf{b}_o) \quad (4)$$

$$\mathbf{m}_t = \mathbf{o}_t \odot h(\mathbf{c}_t) \quad (5)$$

$$\mathbf{y}_t = \phi(\mathbf{W}_{ym}\mathbf{m}_t + \mathbf{b}_y) \quad (6)$$

それぞれの要素に対する式 II

where \mathbf{W} is weight (7)

\mathbf{b} is bias (8)

\odot is the element – wise product for vectors (9)

σ is the sigmoid function (10)

ϕ is network output activation (e.g. softmax) (11)

g, h is activation function (e.g. tanh) (12)

なお、i は input、f は forget、o は output gate を表しており、c は cell activation を示している。

※すべてのベクトルのサイズは等しい。

■ 参考文献

Long Short-Term Memory Recurrent Neural Network Architectures
for Large Scale Acoustic Modeling

- 以前は common lisp mgl を用いていたが、事情により撤回を余儀無くされた。
 - 事情 Optimizer を絡ませずに lstm セルを回すことができない。
- 現在は TensorFlow を Hylang から利用している。
Python と TensorFlow の実行環境が入っていれば動く
(common lisp mgl の内部実装を熟読したが、頑張れば目的に対応する LSTM を実装できると思われる。→ それは卒研の目的とは外れてしまう。)
- 現在ベースとなる Seq2Seq モデルそのものは完成 (ほとんどドキュメントの翻訳に近い) しているが、実行にかなり時間がかかっており満足な結果を手に入れられていない。
リポジトリ
github.com/MokkeMeguru/hy-seq2seq-with-tensorflow

■ Doc2Vec

任意の長さの文章を分類するためのモデル。ある単語それぞれについてベクトル化を行い比較をする Word2Vec の文章版。類似の手法として Bag of Words という手法があるが、Doc2Vec は語順を考慮するという点や、単語の意味を表現するうことが出来るようになるらしいという点で有利。

文書ベクトルを並べた重み行列を用いて、入力となるある文章に対して最もふさわしい文書ベクトルを返す。

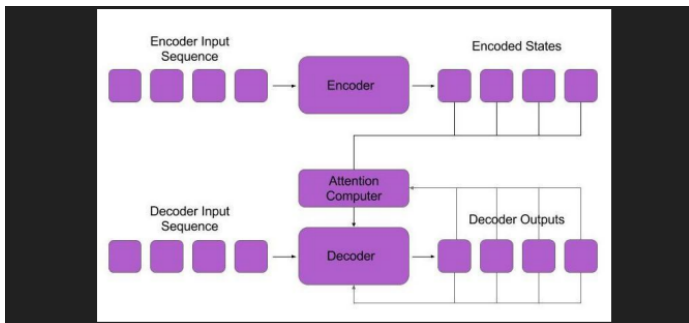
内部では PV-DM (文と文中の単語に固有のベクトルを与えて文に含まれる単語の予測を行う (語順を考慮している)) PV-DBOW (ランダムに抽出した文中の単語から、その文に含まれるだろう単語を予測する) が行われている。

既に文章からの感情分析において成果を出している。

■ Seq2Seq with Attention

Seq2Seq model を長文にも耐えられるように改良したもの。

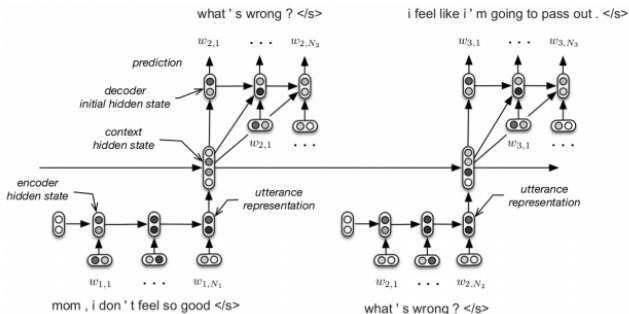
Decoder で最終的に取り出される hidden state を Encoder で毎回考慮させるようにしている。



■ HRED

Seq2Seq model に文脈についての情報を与えるようにしたもの。長い会話を行えるようになる。

このため、RNN は Encoder Decoder に加え Context が必要になる。



■ 参考文献

Distributed Representations of Sentences and Documents
Building End-To-End Dialogue Systems Using Generative Hierarchical
Neural Network Models

- Doc2Vec でトピックごとの小さい問題に分割してそれぞれのトピックごとに Seq2Seq 等を与えてみたい
=> 計算量が多く結果が満足に得られない/Doc2Vec で話者の気分を読み取ることができればそれに応じた返答が出来るのではないかな？
- Seq2Seq の部分を Seq2Seq with Attention や HRED に変えてみたい。
=> どの程度の差が生まれるのか比較を行いたい

- TensorFlow に慣れる。
- 実験を行い、結果データを得る。
- 文体変換に関する論文を読む。(日本語の既存研究は見てきたが、あまりめぼしい物はなかった) 特徴として、すでに文章からの感情分析において成果を出している。
具体的には seq2seq を用いたものと GAN を用いたもの、Zero-Shot 変換を用いたものが論文としてあるので、この3つについて学習していきたい。
- VHRED の論文を読む (HRED に確率成分を追加したものらしい)
- 参考文献
 - A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues
 - Deep Text Style Transfer
 - Zero-Shot Style Transfer in Text Using Recurrent Neural Networks
 - Ridge regression, hubness, and zero-shot learning