Parts of Speech Tagging using GRU

Durgesh Kumar Singh

Dept. of Mathematics
Indian Institute of Space Science and Technology

May 8, 2019

Section 1: Introduction

POS Tagging Problem

- Process of marking up a word in a sentence to a corresponding part of a speech tag, based on its context and definition
- Tags are useful for -
  - Building parse trees
  - Extracting relations between words
  - Building lemmatizers

Techniques for POS Tagging

Section 2: Techniques of POS Tagging

Techniques for POS Tagging

- Lexical Based Methods - most frequently occurring with a word
- Rule-Based Methods - based on rules
- Probabilistic Methods - based on the probability of a particular tag ex. Hidden Markov Models (HMMs)
- Deep Learning Methods - Recurrent Neural Networks (GRU,LSTM) can also be used for POS tagging

Section 3:RNN and GRU cell

Recurrent Neural Network

- Recurrent neural networks persist context information and used for sequence learning problem
- Used in Time series analysis-next word prediction, music composition, image captioning, speech recognition, time series anomaly detection, stock market prediction
- Suffer from short-term memory due to vanishing/exploding gradient problem

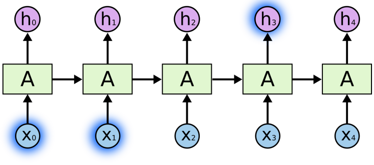

Figure: RNN cell

Gated Recurrent Unit(GRU)

- Repeating modules like RNN but the repeating modules have a different structure
  - Long short-term memory and allows retain any information without much loss
  - Variant of LSTM
- GRU cell has Update and Reset Gates
  - help to regulate the flow of information to the cell state

Structure of GRU Cell

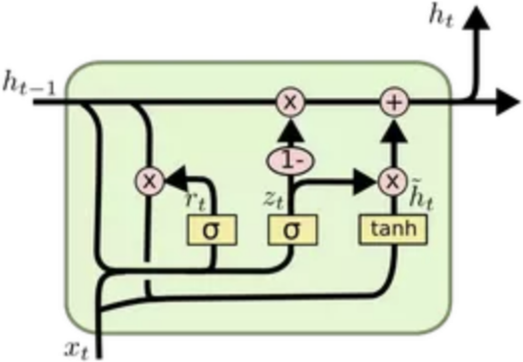

Figure: GRU cell

Structure of GRU Cell

$$\tilde{h}_t = tanh(W \cdot [r_t * h_{t-1}, x_t]$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

Implementation details

Section 4:Implementation details

Standard cell and Mutations

- Analyzed POS Tagging using standard GRU cell and three mutations
- Mutation M1-
  $z = \sigma(W_z \cdot x_t + b_z)$
  $r = \sigma(W_r \cdot x_t + W_r \cdot h_t + b_r)$
  $h_{t+1} = tanh(W_h[r \odot h_t] + tanh(x_t) + b_h) \cdot z + h_t \odot (1 - z)$
- Mutation M2-
  $z = \sigma(W_z x_t + W_z h_t + b_z)$
  $r = \sigma(x_t + W_r h_t + b_r)$
  $h_{t+1} = tanh(W_h[r \odot h_t] + W_x \cdot x_t + b_h) \odot z + h_t \odot (1 - z)$
- Mutation M3-
  $z = \sigma(W_z x_t + W_z \cdot tanh(h_t) + b_z)$
  $r = \sigma(W_r x_t + W_r h_t + b_r)$
  $h_{t+1} = tanh(W_h[r \odot h_t] + W_h x_t + b_h) \odot z + h_t \odot (1 - z)$

Other details

- Default word2vec model provided by torch is used to obtain word embedding of input data
  - Embedding dimension used 300
- Negative Log Likelihood loss(NLL) function is used

$$L = -\frac{1}{n}\sum log(\hat{y}^{(i)})$$

- SGD Optimizer is used to minimize the error function
  - Learning rate 0.1

Implentation Steps

1. Read the dataset in <word>,<tag> form
2. Preprocessing of the dataset
3. Obtained WordEmbedding using Word2Vec model
4. Specify gate equation of the GRU cell(Standard,M1,M2,M3) and created the computation graph using torch package
5. Divided the data set into training and validation sets and performed batching
6. Trained the POS Tagger using GRU cell defined in step 4(approximately 10 epochs)
7. Saved the network paramters for future prediction and obtained results

Result

Section 5:Results

Result

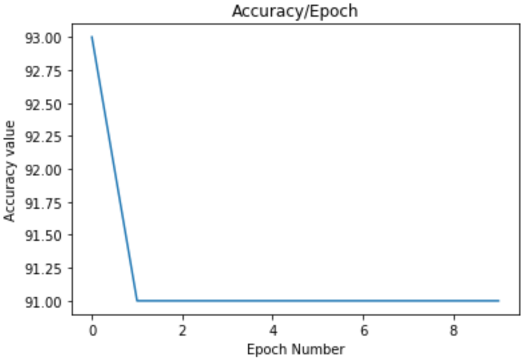

Figure: Accuracy/Epoch Graph for standard GRU cell

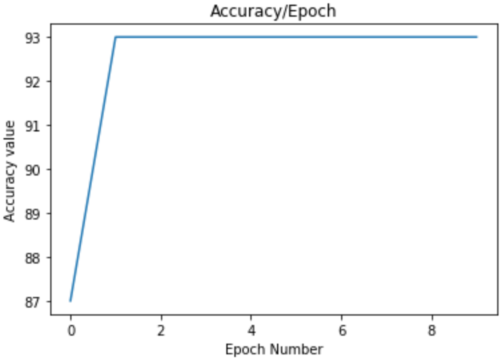

Figure: Accuracy/Epoch Graph for Mutation 1

Result

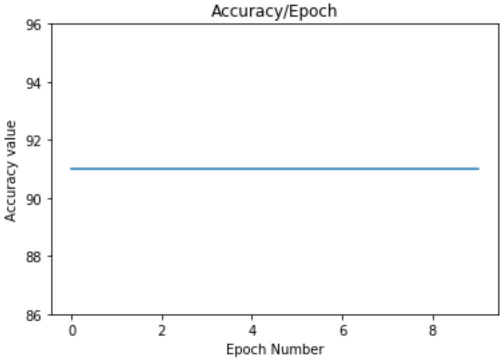

Figure: Accuracy/Epoch Graph for Mutation 2

Result

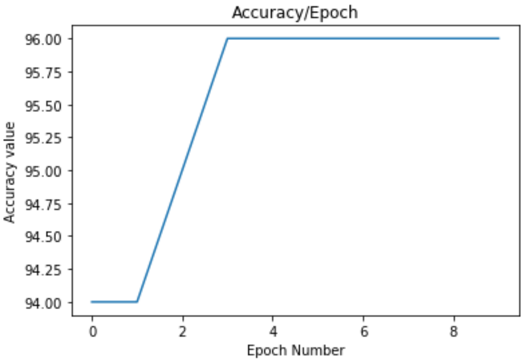

Figure: Accuracy/Epoch Graph for Mutation 3

Conclusion

- Mutation 3 is performing best having 96 percent accuracy in POS Tagging Task

References

1. An Empirical Exploration of Recurrent Network Architectures : Rafal Jozefowicz, Wojciech Zaremba, Ilya Sutskever ; Proceedings of the 32nd International Conference on Machine Learning, PMLR 37:2342-2350, 2015.
2. CS7015: Deep Learning Lecture by Mitesh M Khapra (IIT M) https://www.cse.iitm.ac.in/~miteshk/CS7015.html

THANK YOU

Thank You