Multilingual

Surabhi Kumari

16 April 2017

1 Abstract of the project

English is a fortunate language in terms of number of NLP tools available. But Hindi language lack this facilities. In this project I'm using the deep learning to get corresponding Hindi word for given English word. Here two languages L1 and L2 are represented as two view i.e word vector. I'm using concept of Auto encoder neural network with explicitly maximizing correlation between two languages word vector in common subspace. Once the neural network is train it is able to regenerate the given word vector of L1 or L2 and also for given L1 or L2 word vector corresponding L2 or L1 word vector. Here L1 is for English language and L2 is for Hindi language.

2 Introduction

2.1 Literature survey

Relevant literature regarding this project from where I have taken idea.

1. Multilingual Deep Learning by Sarath Chandar Department of Computer Science Indian Institute of Technology Madras, India.
   - Link: http://www.sarathchandar.in/paper/MultilingualDeepLearning.pdf
2. Correlational Neural Networks by Sarath Chandar University of Montreal.
   - Link: https://arxiv.org/abs/1504.07225
3. Learning a Dual-Language Vector Space for Domain-Specic Cross-Lingual Question Retrieval by Chunyang Chen Nanyang Technological University, Singapore
   - http://ccywch.github.io/chenchunyang.github.io/publication
4. HindEnCorp Hindi English and Hindi only Corpus for Machine Translation by Ondrej Bojar as using hindmonocorp05

In Multilingual Deep learning Predictive auto encoder(PAE) is used which learn the shared representation for two different language. They are taking L1 as English language and L2 as French language. There are four phase

1. Language specific representation:- In this phase features of language L1 and L2 is is extracted by using k-layered stack auto encoder.
2. Shared Representation Learning(SRL):- In this phase parallel entities in L1 and L2 is obtained. This parallel entities is passed through PAE to learn shared representation.
3. Source Language Training:- Here model is trained using the data available in L1 and apply this model to data from L2.
4. Target Language Testing:- Above trained model is applied to test data from L2 by first projecting it to the common space.

For above PAE model Precision is 0.79, Recall is 0.844 and F1 measure is 0.815.

In Correlation Neural Network left view and right view of MNIST data i.e hand written number is divide into two parts left view xi and right view yi. This model is using Multimodel auto encoder neural network that explicitly maximizes correlation among the views when projected to the common subspace. At training time main objective is

- Minimize the self-reconstruction error, i.e., minimize the error in reconstructing xi from xi and yi from yi.
- Minimize the cross-reconstruction error, i.e., minimize the error in reconstructing xi from yi and yi from xi.
- Maximize the correlation between the hidden representations of both views.

It is also extended to deep correlation neural network. By stacking the the correlation neural network and adding new hidden layer as auto encoder. Using CorrNet modal MSE of self reconstruction is 3.6. MSE of cross reconstruction is 4.3. Accuracy of this model for MNIST example is from View1 to View2 is 77.05 and from View2 to View1 is 78.81.

3 Resources

- Data resources:
  For Hindi mono corpus I have use hindmonocorp05.plaintext(Link given above). It is based primarily on web crawls performed using various tools and at various times. It consist 44 million sentences. I have converted this corpora to word vector of dimension 300.
  For English word vector I have use Pre-trained word on part of Google News data set. It consist of word vector of dimension 300 for 3 million word and phrases.
  For constructing word vector of English and Hindi corresponding to each other i.e. parallel word vector of Hindi and English I have used dictionary Murli Glossary Hindi English GCH London 2002 and Hindi English Glossary . This has devanagari script corresponding to English. In total around 32,000 Hindi words vectors correspond to English word vector. This parallel word vector are used to train the neural network.
- Tools: Theano, numpy, scikit-learn etc.

3.1 Work done

This may include the following.

- Description of the data:In total there is 32000 word vector corresponding to English word vector. These word vector have dimension of 300. From 32,000 word vector 25,000 are training set, 5000 are testing set and 2000 is validation set. These data are store in NumPy array.
- Observation on different number of Hidden Layer: By changing number of hidden layer accuracy effect little bit.

Number of Hidden Layer	Accuracy for L1 to L2	Accuracy for L2 to L1
50	76.19	78.46
100	79.19	80.14
150	78.74	80.83
200	79.19	80.14

- Final architecture: The neural network architecture is very much influenced by CorrNet(Correlation Neural Network). Here two views are corresponds to Hindi word vector and English word vector. Model has single input layer and output layer. Initially there is only single hidden layer. Consider the word vector x and y of English word and Hindi. Given input $z = (x, y)$, the hidden layer computes an encoded representation as follows:

$$h(z) = f(Wx + Vy + b) \tag{1}$$

where **W** and **V** are weight matrix and **b** is bias input. The output layer then tries to reconstruct z from this hidden representation by computing

$$z' = g([W'h(z), V'h(z)] + b') \tag{2}$$

Vector z' is the reconstruction of z. Function f and g are a non-linear activation function, here I'm using sigmoid function. Once the W, V, b, W',V' and b' parameter are set, I decoupled the hidden layer and add new common hidden layer as shown in figure. Repeat this step for multiple hidden layers.
For this problem I have taken 150 hidden layers. Rmsprop optimizer technique is used. For loss function I have use mean square error of self-reconstruction error i.e from L1 to L1 and L2 to L2 and cross-reconstruction error i.e L1 to L2and L2 to L1.

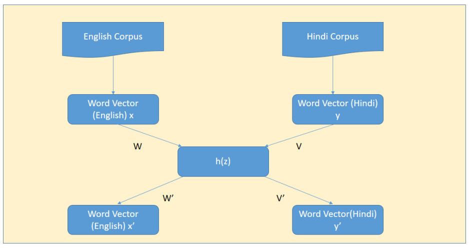


Figure 1: The architecture of neural network with single hidden layer.

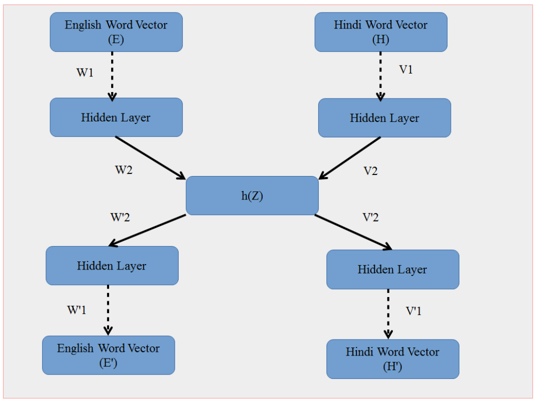


Figure 2: The architecture of neural network with Multiple hidden layer.

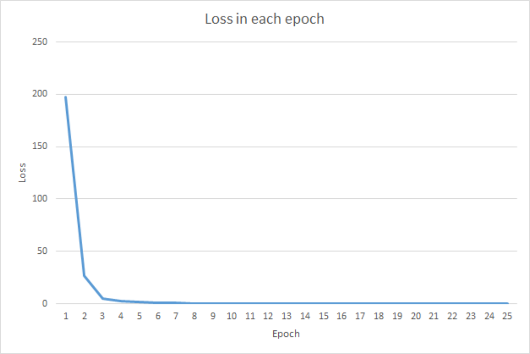
- Results from different optimization techniques: These results are corresponds to 150 Hidden Layers.

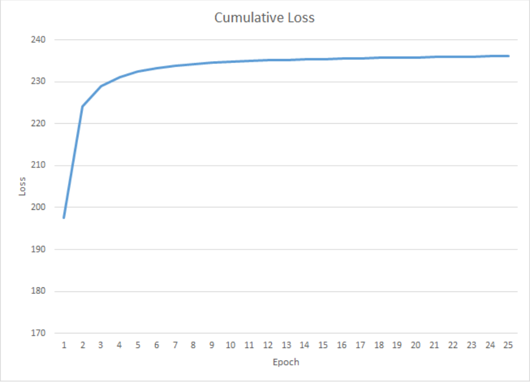
Optimization Techniques	Accuracy for L1 to L2	Accuracy for L2 to L1
Rmsprop	78.74	80.83
Sgd	75.83	76.13
CM	77.68	79.30
Nag	78.44	79.31
Adagrad	74.6	77.62
Adadelta	75.89	77.79

  - Sgd- Simple Gradient Descent

- CM- Classical Momentum
- Nag- Nesterov Accelerated Gradient

- Loss vs Epoch Plot:





- github location for code: https://github.com/Surabhi-Kumari/Multi-lingual

3.2 Future work

- Further I try to improve more accuracy by increasing training data set.
- This model can be used for finding corresponding word of different sets of languages other than Hindi and English, e.g. Hindi, Bengali or English, Bengali or Marathi, Hindi.