Modular Deep Encoder-Decoders

Group 40: Arnold, Gurunat, Huang, & Qu

March 29, 2016

Abstract

In this short paper we propose an approach to transfer learning using rich vector embeddings. The suggested technique can be applied to any supervised task, and it handles multiple sources and changing sources of data without the need for retraining. To verify our ideas, we apply our ideas to the task of text-classification.

I Introduction

Our goal is to generate rich vector embeddings from articles to classify them into predefined categories. Then, we want to extend our model with an additional source of data: the title of the videos. To handle this new knowledge source without retraining our previous model, we suggest to generate a new embedding that will be used to modify the original one. This combined embedding will then be used for the classification task.

More formally, we want to jointly learn a set of *encoder* functions $\{E_i\}_0^N$ mapping samples $x \sim \chi_i$ from a set of data distributions $\{\chi_i\}_0^N$ to a fixed-sized vector embedding V.

$$\forall i \in [0; N] : E_i(x) : \chi_i \rightarrow V_i \in \Re^M$$

$$V = \sum_i^N V_i$$

The embedding V is then fed into a decoder function D which in the case of classification learns a mapping from the vector space of V to the label space L.

$$D(v) : \Re^M \rightarrow L \in \Re^D$$

Finally, we want to extend the set of encoders by learning a new encoder E_{N+1} which handles samples from a different dataset χ_{N+1}, without retraining our trained decoders and encoders.

$$E_{N+1}(x) : \chi_{N+1} \rightarrow V_{N+1} \in \Re^M$$

Although we could have used any kind of mapping, we chose to use deep learning algorithms, as they easily learn hierarchical representations and have been known to highly outperform other statistical techniques on natural language tasks. Learning embeddings has been previously done by [1], although our proposed contribution is slightly different.

[1] used deep autoencoders to obtain a better initialization of the parameters of their model. Our approach instead is much closer in spirit to the work of Sutskever [3] and Vinyals [4]. They both use encoders on a sequence of data to generate a *thought-vector* which will then be decoded in the desired terget sequence. In some sense, our proposal adds the transfer learning component to their contribution. This approach is also similar to the work of Karpathy & al [5] where they mapped images to their captions with embeddings.

II Method

Materials

We will use the freely avaiable dataset from `https://github.com/ParallelMazen/SaudiNewsNet` and extract the titles and articles content. Since the dataset initial purpose is not the classification, we will have to slightly rework it to fit our purpose.

We will use the neon python library for training deep networks. In the case where the training phase would become too expensive, we will also rely on the mpi4py library.

Procedure

The procedure is quite straight-forward. While part of the team will work on building our tailored dataset, the other half will work on the model definition and training procedure. Coordination at the beginning of the work will be curcial. In particular, we plan on feeding the text data as a sequence of one-hot character encodings.

Once every pre-requisite is available, we will train the first encoder E_1 (implemented as a recurrent neural network) to build the embedding V_1. Since we will only be dealing with a relatively simple classification task, our decoder D_1 will simply be a fully connected multi-layer perceptron. They will be jointly trained end-to-end by propagating the gradients through the embedding from D_1 to E_1.

The second training step will be to train the second encoder E_2. Again, we also perform training end-to-end, but specifically do not propagate the gradients through E_1. We hope to avoid optimizing the parameters of D_1, but this might not be practically doable.

Evaluation

The final evaluation scheme is mostly to be determined depending on our obtained outcomes. Obviously, we want to compare the new model with the the the jointly trained one, as well as a classification baseline.

III Participants & Labor Division

The participants in alphabetical order:

- Seb Arnold - 9013085897 - arnolds@usc.edu

- Prashanth Gurunath Shivakumar - 2251924199 - pgurunat@usc.edu
- Qiangui Huang - 9532576000 - qianguih@usc.edu
- Cheng Qu - 2385279985 - chengqu@usc.edu

Instead of having a fixed division of the labor, we assigned one responsible to each task. The responsible acts as a project lead on the designed task and other team members can contrbibute to the task as required.

- Dataset: Cheng
- Model Training: Quiangui
- Knowledge Transfer: Seb
- Report Writing: Prashanth

IV References

Part of the relevant literature review. More literature was involved for the deep learning part.

1. Supervised Representation Learning: Transfer Learning with Deep Autoencoders, Zhuang & al, http://ijcai.org/Proceedings/15/Papers/578.pdf
2. Transfer Learning via Dimensionality Reduction, Pan & al, https://www.cse.ust.hk/~jamesk/papers/aaai08.pdf
3. Sequence to Sequence Learning with Neural Networks, Sutskever & al, http://arxiv.org/abs/1409.3215
4. Grammar as a Foreign Language, Vinyals & al, http://arxiv.org/abs/1412.7449
5. Deep Fragment Embeddings for Bidirectional Image Sentence Mapping, Karpathy & al, https://cs.stanford.edu/people/karpathy/nips2014.pdf