A long short-term memory recurrent neural network for generating astrophysics-specific language and assessment using tf-idf, cosine similarity, and a vector space model

Capstone Proposal: CUNY School of Professional Studies, MS in Data Analytics

Daina Bouquin

September 16, 2017

Introduction

In 1987, a very simple definition of an "artificial neural network" (ANN) was presented by Maureen Caudill, an expert on artificial intellegence. Caudill wrote that an artifical neural network is "…a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs" [1]. With this definition in mind, what I propose herein is a small study on the application of a particular type of recurant ANN, long short-term memory (LSTM), to generate domain-specific text in various subdisciplines of astronomy and astrophysics. Subsequent to the training and sampling of astronomy and astrophysics related text, I will apply measures including term frequency-inverse document frequency (tf-idf) and cosine similarity, along with a vector space model to assess the degree to which the text generated by the ANN accurately reflects the text used to train the system. In conducting this study I will be applying this approach to a dataset that has not yet been used to test the capacity for this type of system to "learn": metadata from the SAO/NASA Astrophysics Data System. I will also be applying an LSTM ANN to in a jargon-heavy scientific domain, which is an area of limited research.

Data Source

The SAO/NASA Astrophysics Data System (ADS) [2], is an online database of over eight million astronomy and physics papers from both peer reviewed and non-peer reviewed sources. The ADS is a highly used resource in the astronomy and astrophysics communities and has many levels of indexing. The ADS API makes it possible to query this valuable resource to better understand authorship and publishing behavior in these fields among many other applications. The ADS allows for full-text searching, and in the near future, will be fully incorporating Unified Astronomy Thesaurus keywords into their indexing schema. The ADS is managed by the Smithsonian Astrophysical Observatory at the Harvard–Smithsonian Center for Astrophysics with funding from NASA.

Background

Artificial neural networks can be understood as being organized into "layers" with layers being made up of a number of interconnected "nodes" which each contain an "activation function". Data are passed into the "input layer" of the system, which communicates to "hidden layers" where the actual processing is done via a system of weighted "connections". Subsequently, the hidden layers connect with an "output layer" where the answer is output [3]. Since the creation of ANNs, these layered connections of nodes have been implementated in many different ways. Each implementation has been designed to meet different needs and experiments. Perhaps the most recognizable application of ANNs in popular culture is Google's "Deep Dream", a computer vision program that uses a convolutional neural network to find and enhance patterns in images via algorithmic pareidolia [4].

In text-based applications of ANNs though, recurant neural networks have become favored over convolutional systems. For instance, Twitter bots like "Deep Drumpf" [5] have been used to demonstrate the powerful effectiveness of recurrant ANNs by emulating the linguistical style and syntax of individual people. Similarly, people have used recurrant ANNs to generate text based on the plays of Shakespeare [6], the essays of Paul Graham [7] and even to generate baby names [8]. Andrej Karpathy, Tesla Motors' leader on artifical intelligence reserach, has stated that some of the reasoning behind the "unreasonable effectiveness of recurrant neural networks" is that with convolutional systems, "their API is too constrained" in that they accept only a fixed-sized vector as input and subsequently produce a fixed-sized vector as output using a fixed amount of computational steps. Unlike these convolutional ANNs, recurant ANNs allow us to operate over sequences of vectors: sequences in the input, output, or even both [9].

To further define and justify the approach to be used for the herein described experiment, it is necessary to define the specific recurant ANN I will be employing: long short-term memory (LTSM). LTSM ANNs are "capable of learning long-term dependencies" [10] and do not start from scratch whenever they are presented with new data. LTSMs were introduced by Hochreiter and Schmidhuber in 1997 [11] and were refined iteratively following their work. The LTSM approach to recurant ANNs is ideal for the type of text-based analysis I will be conducting because by using LTSM you can train the system on a large corpus of text and it will "learn" to generate text like it one character at a time. This, along with LTSM's relative insensitivity to "gap length" gives an advantage to LSTM over alternative recurant neural networks [12]. Note, I have not found any evidence so far showing approach being applied to text from astrophysics articles or using language from such a scientific-jargon heavy field.

Test Cases

In order to test out the capability of a LSTM recurant ANN to work effectively with astronomy-related text, I will subset increasingly large datasets from the metadata available through the Astrophysics Data System. I will begin with article titles and increase the size of the datasets from there (testing both the processing power I have available and the difference larger datasets make on the system). I will use the titles to train the described ANN and to generate test titles. I will do this for different subdisciplines within the field (e.g. particle physics, planetary physics, high enery astrophysics, etc). If I am able to execute this process using article titles, I will move on to generating article abstracts using the same process.

I am still investigating the size of the datasets needed to conduct these test cases.

Hypothesis

My hypthothesis is that some fields will be more acurately generated by the ANN than others. This, I think, will depend on the diversity of terms used within the corpus of text representing the subdomain. This is to say that a subdomain with a larger variety of words (combinations of letters) will be more challenging to reproduce than a training corpus with fewer unique words. I hypothesize that more data will be needed to achieve a high degree of similarity between generated text and training text in some fields than others and that this will correspond to unique word count. I am also interested in seeing which real title corresponds most closely with the ANN generated title and to look at the citation count of the most similar title. I will then be able to see if titles with higher citation counts are easier or more difficult to emulate than those with lower citation counts.

Assessment

In order to assess the effectiveness of the proposed system, I will take advantage of tools that have become popular and impactful in the field of digital libraries. I will first make use of the measure "term frequency-inverse document frequency" (tf-idf), which is a numerical statistic that is intended to reflect how important a word is to a document in a corpus of text [13]. This value will allow me to take advantage of an both cosine similarity and a vector space model to determine how similar two titles actually are to each other. Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them [14] and is commonly applied in setting where tf-idf vectors are being compared. By applying these techniques in a vector space model, I will be able to identify which real title a generated title is most similar to, and the degree of similarity between them.

I have created a proof-of-concept Jupyter notebook to demonstrate how this assessment could be done and shown it to be effective with a small dataset made up of 1000 titles extracted from the Astrophysics Data System [15].

Tools

There are many computational tools to help analysts implement LSTM recurant ANNs. I am currently exploring the following options to implement my proposed study: Theano [16], Torch [17], TensorFlow [18], and a straight Python/numpy approach [19]. Each of these potential tools has pros and cons associated with their use. I am limited in the amount of processing power I have available to me and this may in the end be the deciding factor in chosing a tool and scale for the implementation of the ANN. I will use Python and the following modules for the assessment portion of the project: Pandas, Numpy, Sklearn.feature_extraction.text, Skylearn.metricspairwise, and math. I will use the ADS API to obtain the datasets that will be used for training, sampling, and testing the system.

References

[1] https://dl.acm.org/citation.cfm?id=38295
[2] https://ui.adsabs.harvard.edu/
[3] http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html
[4] https://arxiv.org/abs/1409.4842
[5] http://www.deepdrumpf2016.com/about.html
[6] https://github.com/RuthAngus/Fakespeare
[7] https://suriyadeepan.github.io/2017-02-13-unfolding-rnn-2/
[8] https://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/nlp/corpora/names/
[9] https://karpathy.github.io/2015/05/21/rnn-effectiveness/
[10] https://en.wikipedia.org/wiki/Long_short-term_memory
[11] http://www.bioinf.jku.at/publications/older/2604.pdf
[12] https://colah.github.io/posts/2015-08-Understanding-LSTMs/
[13] http://infolab.stanford.edu/~ullman/mmds/book.pdf
[14] https://en.wikipedia.org/wiki/Cosine_similarity
[15] https://github.com/dbouquin/DATA_698/blob/master/ADS_data_exploration_698.ipynb
[16] http://www.deeplearning.net/software/theano/
[17] http://torch.ch/
[18] https://www.tensorflow.org/
[19] https://gist.github.com/karpathy/d4dee566867f8291f086