Embeddings of Physics Equations

Amir Raza, Marley Xiong

1 Abstract

We present embedding representations of physics equations from scientific literature. We collect a novel dataset of ˜29k physics equations paired with word context from research articles and present a CBOW and Bernoulli embedding model. Through qualitative inspection, we found our embeddings to contain coherent information and can provide the ability to discover semantically similar equations and words.

2 Introduction

Many existing machine learning methods ignore information from equations when analyzing scientific articles. Documents that contain many equations—notably, literature in math and physics—stand to lose a lot of information once mathematical symbols are removed. Finding semantic representations of equations would help with downstream tasks and improve the navigation and retrieval of scientific articles.

Current frameworks of machine learning do not work on symbols directly. Symbol data is converted to feature vector representations. Such representations can be developed in various ways, many of which have their associated shortcomings. For instance, one-hot encodings cannot capture the semantic meaning of a word. Furthermore, for large vocabulary sizes, this approach suffers from the so called curse of dimensionality. A model which learns the distribution of these one-hot encoded 'words' , in a sequence of inputs, could still fail if it has not seen that particular sequence of words before.

The motivation of this work is twofold: (1) to create a semantic representation of physics equations using existing techniques, and (2) extend the work of Krstovski and Blei [2018] with exponential word embeddings. Thus far, only papers in machine learning have been investigated for equation embeddings. We hypothesize that similar levels of performance can be achieved on equations in physics research, and that exponential embeddings can improve upon Bernoulli embeddings for the purpose of equation representation.

To do so, we collect a novel dataset of physics equations and word context from scientific articles. We evaluate the viability of our representations with a qualitative analysis, assessing our ability to discover related equations and related words provided a query equation. We implemented from scratch an exponential embedding model, for which no code was previously available in PyTorch.

We hope that this would be a good contribution towards the NLP community and aid downstream tasks such as document recommendation system for educational purposes.

3 Related Work

Word embeddings were initially proposed by Bengio et al. [2003] and have revolutionized the field of NLP through providing a mechanism of finding representations of natural language in an unsupervised setting. Over the years, new models have emerged such as Word2Vec (Xin Rong, 2014), GloVe (2014), and FastText Joulin et al. [2016]. However, none of these investigations have involved equations or attention to the relationship between equations and vocabulary in a specific academic

field.
Most recently, Krstovski and Blei [2018] was able to model equations from machine learning disciplines with source equations from arXiv papers. They provide a qualitative analysis of equation examples and related words, including the commonly used equations for cosine similarity and LDA. We hope to extend their results to physics, where equations may be more heterogenous and less comprehensible to the reading public.

Fast-Text ,Joulin et al. [2016], has the ability to compute word representations for words that did not appear in the training data. It is based on the assumption, that there are 'sub-words' or 'roots + prefixes' for all the words. We did not approach our equation embedding problem with the fast text approach, as we don't believe equations fit this assumption of having subwords. And it is better to treat equations as unique singletons.

4 Method

4.1 Physics Equation Dataset

We construct a novel dataset of words and equations from scientific articles published on arXiv under the primary category of High Energy Physics - Theory (hep-th). We downloaded the LaTeX sources of the publications from 2002 - 2003 from the dataset posted for the KDD cup, a reputable and often-used repository of physics papers.

Collection	# Docs	# Words	# Eqs.
hep-th	1020	14 329	29 344
AI	1054	11 171	18 659

Above: Statistics for our physics dataset (hep-th) compared to previous work (AI).

4.2 Preprocessing

As per Krvstovski et al. (2019), we investigate display equations as opposed to inline equations, which tend to represent variables with general meaning. Equations are detected with a custom Python parser which interprets macros and inspects the LaTeX source code for the beginning and end of equations.
Each equation is treated as a "singleton word" in the context of surrounding text. We tokenize the text in a neighborhood of 10 words before and after each equation, including words with character length greater than or equal to 4. The vocabulary is constructed after lower-casing and removing punctuation and LaTeX symbols, yielding a dataset with similar statistics as the AI subset created by Krvstovski et al.

4.3 Equation embedding models

We replicate the CBOW model described in Krstovski and Blei [2018] with the same parameters. Our implementation was in gensim using 100 latent dimensions and a context window of 4 (4 words before, 4 words after) each word and equation in an example.

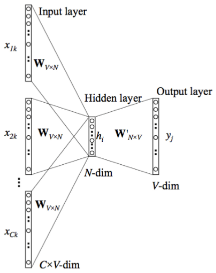

Figure 1: Continous Bag of Words network

4.4 Exponential embeddings for equations

Standard embedding learning approaches are designed with text data in mind. Exponential family is a class of probability distributions including the Gaussian distribution, Bernoulli, and Poisson. They can be written in the form:

$$P(X = x; \theta) = h(x) \exp(\eta(\theta)^T T(x) - A(\theta))$$

Model assumption
Bernoulli-embeddings parameterize the conditional probability of the 'target' word given its 'context' words via a linear combination of the embedding vector and the context vectors . That is:

$$P(x_{iv} \mid x_{c_i}) = Bernoulli(\sigma({\rho_v}^T \sum_i))$$

Using a CBOW approach, x_{iv} is the target word and x_{c_i} is the set of context words surrounding it.

The natural parameter,η here, is a function of the context and the embedding (ρ) and context vectors (α). The observations in the context and the context vectors are combined into a weighted average called the context representation. The context representation captures the latent attributes of the context. For textual data, there is sharing of parameters ρ and α across the whole dataset. The original paper (Rudolph et al. [2016]) claimed that Bernoulli embeddings relax the 'one-hot constraint' of x_i , and work well in practice. Though for our case, we had to resort to a one-hot encoded representation of the vocabulary to make the model work.

Changing unsupervised learning to supervised learning framework

This idea was introduced by Mikolov et al. [2013b]. The original problem of learning embeddings is an unsupervised learning problem but it is converted to a supervised learning problem by defining a 'positive' loss and 'negative' loss.

$$\sum_{i=1}^{N} \Big(loss_{pos} + loss_{neg} \Big)$$

where

$$loss_{pos} = \sum_{V:x_{iv}=1} \log \sigma({\rho_v}^T \sum_{ipos)}$$

and

$$loss_{neg} = \sum_{V:x_{iv}=0} \log \sigma(-{\rho_v}^T \sum_{ineg})$$

σ is the logistic sigmoid.

Negative sampling

Here $\sum_{ipos}$ is a term which captures information from the context , C_i of a target word x_{iv} in the vocabulary.

$$\sum_{ipos} = \sum_{(j,w)\in C_i} \alpha_w x_{jw}$$

For negative loss, we randomly sample target words, believing that since we are negatively sampling, we would pick up random target words for a given context.

$$\sum_{ineg} = \sum_{(j,w)\in C_i} \alpha_w sample(x_{jw})$$

where $sample(x_{jw}) \sim multinoulli(corpusdistribution)$ Doing the summation over all the negative words would be very computationally expensive. Rather than summing over all words which are not at position i, we sum over a subset of n negative samples drawn at random. D Mikolov et al. [2013b] recommend sampling from p, the unigram distribution raised to the power of $\frac{3}{4}$'.

Negative sampling breaks the constraint that the sum of the probability of each word must equal one, and instead models the probabilities of the individual entries of the one-hot vectors representing the words.

4.5 Equation Embeddings

Krstovski and Blei [2018] showed in their paper that extending Bernoulli embedding with two more parameters for equations , could achieve good performance on learning equation embeddings.

$$Loss = \sum_{i=1}^{N} \log p(w_i \mid w_{ci}) + \sum_{m=1}^{E} \log p(e_m \mid e_{cm})$$

whereas before in the exponential Bernoulli embeddings,for both words and equation tokens,

$$p(x_i \mid x_{ci}) = \sigma({\rho_v}^T \sum_j \alpha x_j)$$

This is a sum of two sets of conditional distributions, the first over observed words (w_i) and the second over observed equations (e_m).

We could not implement the equation embedding loss as we originally set out to do. But looking at the loss term, we decided to simplify our goals, and to implement loss for Bernoulli embeddings only. In future we hope that we would be able to extend our implementation for equation embeddings.

4.6 Bernoulli implementation

Code: `https://colab.research.google.com/drive/1vXEOizJwO4RsxnPYPbvTzjX-Hba-uRKV#scrollTo=4IWkxZFNi1qc`

We took a CBOW approach to train our neural network Mikolov et al. [2013a]. We used Xavier Initialization to initialize for the embedding layers. Default initialization did not work for us, as it was leading to loss values increasing in the iterations.

The number of negative samples we used for negative loss calculation was 5. The empirical probability of each of the tokens was calculated, and we used a multinoulli sampling approach to pick up the negative samples. The original Word2Vec implementation had suggested using empirical frequencies raise to power $\frac{3}{4}$, ie $P(w)^{\frac{3}{4}}$

We used a latent dimension of 25 and context window of 3. We added an additional ReLu layer on the probability values before passing to the log function.

We used Adam Optimization with learning rate set to 10^{-5}. Increasing the learning rates was leading to model loss not decreasing at all. After tweaking with all the hyper-parameters, the loss did not converge, although it did decrease until a low value then remained unchanged. We speculate that our model loss was stuck in a local minima regime.

5 Results

Following the method of qualitative analysis described in Krstovski and Blei [2018], we queried two physics equations and retrieved the top related equations and words. The equations and words were generated by computing the cosine distance across all the equation and word embeddings and taking the 5 nearest neighbors.

Query:

$$\mathcal{H}\Psi = E\Psi$$

	Top Equations	Top Words
1	$\mathrm{T}_s = \frac{1}{2\pi\alpha'}$	energy
2	$\langle \underline{O} \vert \underline{I}_{p'} \underline{I}_p \vert \underline{O} \rangle = Z \frac{1}{N} \delta_{\Gamma_m}(, ')\chi() + Z^2 \Big(1 - \frac{1}{N}\Big) \chi()\chi(').$	unregularized
3	e_j	d'Alembertian
4	$(\mathrm{T}^i) = 16 C_{ijk} T^i T^j T^k \overset{!}{=} 1 \ ,$	2-dimension
5	$\Big\{ V_k^{(\alpha\beta)}, V_l^{(\alpha\beta)} \Big\}_P = -i\Lambda^{\alpha\alpha}(\beta - \alpha)\,(k - l) V_{k+l}^{(\alpha\beta)} \ , \qquad V_k^{(\alpha\beta)}{}^{\delta_\alpha}_{k u_{\alpha\beta}}$	corrections

Query:

$$i[Q_{ja}, z] = \chi_{ja} \qquad [\bar{Q}_{j\bar{a}}, z] = 0.$$

	Top Equations
1	$\text{g(r)=}\left(-M +^2 + \frac{J^2}{4r^2}\right)$
2	$^{(\gamma)}\sqrt{\gamma}d^2y + \sum_n \delta_n = 4\pi.$
3	$_1 = (\sqrt{2}\,, 1\sqrt{2})\,, \qquad _2 = (-\sqrt{2}\,, 1\sqrt{2})\,.$
4	$\text{=-4g}^2 \sum_{i<j} X_i\, X_j\ .$
5	$\sigma^+_{f_1}$

	Top Words
1	fields
2	group
3	semi-direct
4	landau
5	spontaneously

For a given equation, we are able to retrieve relevant words to make sense of the equation.

For the first query ($[\mathcal{H}\Psi = E\Psi]$):

The top equation $[T_s = \frac{1}{2\pi\alpha'}$ is a frequency relation (which occurs with energy in physics). The top words retrieved do make sense, as they are energy, unregularized and d'Alembertian, a single symbol operator in electromagnetism.

For the second query equation ($[i[Q_{ja}, z] = \chi_{ja} \qquad [\bar{Q}_{j\bar{a}}, z] = 0.$)

The top word retrieved, fields, is related to gravitational field, and the top equation is similar to a gravitational equation $g(r) = \frac{GM}{r^2}$.

Landau equation is $\frac{\partial M}{\partial t} = -\gamma MxH_{eff} - \lambda Mx(MxH_{eff})$, which occurs in theory of magnetization, while the group equation is $\sigma^{-1}(\frac{\mu}{M}^d G(g(M)))$

Collection	# Score
hep-th	-71.5
AI	-11.52

The log probability score of our CBOW model compared to the score obtained in the paper indicated a signifcantly worse performance. We hypothesize this is due to the heterogeneity of our data and the number of singletons.

For Bernoulli embeddings, loss values started at 82 and ended at around ~ 1.3. We postulate the model was capable of learning information but did not converge due to the number of singleton examples, far higher in our dataset than normal text documents.

6 Discussion and Conclusion

We validated the hypothesis that relevant semantic information can be extracted from equations using embeddings. However, the results were not as convincing as those presented in Krstovski and Blei [2018]. This may be due to the difference in datasets: the physics dataset had nearly twice as many equations for the same number of papers and more singleton examples.

We were also able to implement Bernoulli embeddings from scratch for a new use case. We found that theoretically Bernoulli embeddings were the generalized version of CBOW and Word2Vec, and we achieved a better understanding of why negative sampling is required, and why it doesn't disturb the original loss objective.

Given our observation, that even CBOW is able to retrieve some relevant equations (i.e. the example of gravitation field energy), we can postulate that if Bernoullli embeddings are a more general version of CBOW, it should have better learning capacity, and hence better performance.

7 Statement of Contribution

Amir: Worked on Bernoulli embedding implementation and theoretical underpinnings of equation embeddings

Marley: Created dataset and CBOW model, wrote report

$$
\begin{gathered}
-\log p(D_t \mid W) + \sum_{l=1}^{L} \left[\frac{1}{2} ||\Lambda^l \circ ({\mu_t}^l - {\mu_{t-1}}^l)||_2^2 + \right. \\
({\sigma_{init}}^l)^2 ||(\frac{{\mu_{t-1}}^l}{{\sigma_{t-1}}^l})^2 \circ ({\mu_t}^l - {\mu_{t-1}}^l)||_1 + \\
\left. \frac{\beta}{2}.1^T [(\frac{{\sigma_t}^l}{{\sigma_{t-1}}^l})^2 - \log(\frac{{\sigma_t}^l}{{\sigma_{t-1}}^l})^2 + ({\sigma_t}^l)^2 - \log({\sigma_t}^l)^2] \right]
\end{gathered}
$$

References

Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.

K. Krstovski and D. M. Blei. Equation embeddings. *arXiv preprint arXiv:1803.09123*, 2018.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013b.

M. Rudolph, F. Ruiz, S. Mandt, and D. Blei. Exponential family embeddings. In *Advances in Neural Information Processing Systems*, pages 478–486, 2016.