Deep Neural Networks and How They Apply to Sequential Education Data

Conference Paper · April 2016

DOI: 10.1145/2876034.2893444

CITATIONS
11

READS
369

3 authors, including:

Joshua C. Peterson
Princeton University
37 PUBLICATIONS **148** CITATIONS

SEE PROFILE

Zachary A Pardos
University of California, Berkeley
81 PUBLICATIONS **1,040** CITATIONS

SEE PROFILE

Deep Neural Networks and How They Apply to Sequential Education Data

Steven Tang
UC Berkeley
Tolman Hall
Berkeley, CA 94720
steventang@berkeley.edu

Joshua C. Peterson
UC Berkeley
Tolman Hall
Berkeley, CA 94720
jpeterson@berkeley.edu

Zachary A. Pardos
UC Berkeley
4641 Tolman Hall
Berkeley, CA 94720
zp@berkeley.edu

L@S 2016, April 25-26, 2016, Edinburgh, Scotland Uk
ACM 978-1-4503-3726-7/16/04.
http://dx.doi.org/10.1145/2876034.2893444

Abstract
Modern deep neural networks have achieved impressive results in a variety of automated tasks, such as text generation, grammar learning, and speech recognition. This paper discusses how education research might leverage recurrent neural network architectures in two small case studies. Specifically, we train a two-layer Long Short-Term Memory (LSTM) network on two distinct forms of education data: (1) essays written by students in a summative environment, and (2) MOOC clickstream data. Without any features specified beforehand, the network attempts to learn the underlying structure of the input sequences. After training, the model can be used generatively to produce new sequences with the same underlying patterns exhibited by the input distribution. These early explorations demonstrate the potential for applying deep learning techniques to large education data sets.

Author Keywords
Educational Data Mining; Deep Learning; Long Short-Term Memory.

ACM Classification Keywords
K.3.2 [Computers and Education]: Computer and Information Science Education.

Deep Learning & Education

In recent years, deep learning techniques have dramatically improved on previous state-of-the-art results in many fields. As a result, some educational researchers have recently begun to apply deep learning techniques to improve student modeling. As one example, [4] represented the Knowledge Tracing problem using Recursive Neural Networks (RNNs) to model student learning and achieve improved predictive capability over Bayesian Knowledge Tracing models [1]. The authors noted that a key advantage of using deep learning models was that they do not require human expert annotations and can take advantage of any student input that can be vectorized.

In this work-in-progress paper, we leverage a standard Long Short-Term Memory neural network model to explore two different contexts within education: writing samples from students in summative environment and clickstream activity within a Massive Open Online Course (MOOC). The use of a single model and architecture highlights the flexibility and broad applicability of deep recurrent neural networks to large, sequential student data.

Recurrent Neural Networks & LSTMs

Recurrent neural networks (RNNs) are a family of networks that can connect neurons over time, allowing for input sequences of arbitrary length. Crucially, RNNs incorporate a high dimensional, continuous latent state. This representation allows RNNs to use information from the past to impact a prediction at a later point in time. A popular variant of the RNN is the Long Short-Term Memory [3] architecture, which is thought to help recurrent networks train more effectively through the addition of gates that learn when to retain information in the latent state and when to clear or "forget" that information, allowing for meaningful long-term interactions to persist. Figure 1 shows a diagram of a simple recurrent neural network.

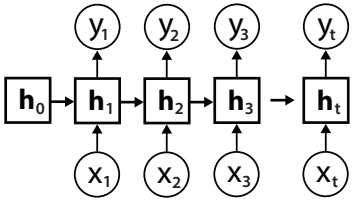

Figure 1: Simple recurrent neural network

$$\boldsymbol{h}_t = tanh(\boldsymbol{W}_x \boldsymbol{x}_t + \boldsymbol{W}_h \boldsymbol{h}_{t-1} + \boldsymbol{b}_h) \quad (1)$$

$$\boldsymbol{y}_t = \sigma(\boldsymbol{W}_y \boldsymbol{h}_t + \boldsymbol{b}_y) \quad (2)$$

The RNN model is parameterized by an input weight matrix $\boldsymbol{W}_x$, recurrent weight matrix $\boldsymbol{W}_h$, initial state $\boldsymbol{h}_0$, and output matrix $\boldsymbol{W}_y$. $\boldsymbol{b}_h$ and $\boldsymbol{b}_y$ are biases for latent and output units, respectively.

LSTMs add additional gating parameters that are explicitly learned in order to determine when to clear and when to augment the latent state with useful information. Each hidden state h_i is instead replaced by an LSTM cell unit, which contains additional gating parameters. As a result of these gates, LSTMs have been found to train more effectively than simple RNNs. The update equations for an LSTM are:

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_{fx} \boldsymbol{x}_t + \boldsymbol{W}_{fh} \boldsymbol{h}_{t-1} + \boldsymbol{b}_f) \quad (3)$$

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_{ix} \boldsymbol{x}_t + \boldsymbol{W}_{ih} \boldsymbol{h}_{t-1} + \boldsymbol{b}_i) \quad (4)$$

$$\tilde{\boldsymbol{C}}_t = tanh(\boldsymbol{W}_{Cx} \boldsymbol{x}_t + \boldsymbol{W}_{Ch} \boldsymbol{h}_{t-1} + \boldsymbol{b}_C) \quad (5)$$

$$\boldsymbol{C}_t = \boldsymbol{f}_t * \boldsymbol{C}_{t-1} + \boldsymbol{i}_t * \tilde{\boldsymbol{C}}_t \quad (6)$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_{ox} \boldsymbol{x}_t + \boldsymbol{W}_{oh} \boldsymbol{h}_{t-1} + \boldsymbol{b}_o) \quad (7)$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t * tanh(\boldsymbol{C}_t) \quad (8)$$

Figure 2 shows the anatomy of a cell, where the numbers in the figure correspond to the previously mentioned update equations for the LSTM. $\boldsymbol{f}_t$, $\boldsymbol{i}_t$, and $\boldsymbol{o}_t$ represent the gating mechanisms used by the LSTM to determine "forgetting" data from the previous cell state, what to "input" into the new cell state, and what to output from the cell state. $\boldsymbol{C}_t$ represents the latent cell state for which information is removed from and added to as new inputs are fed into the LSTM. $\tilde{\boldsymbol{C}}_t$ represents an intermediary new candidate cell state that is gated to update the next cell state. The model

used in this paper consisted of two LSTM layers with 128 hidden nodes each.

LSTM on Student Essays

LSTMs have been used in many language modeling applications in recent years. Such models applied to student essay corpora can potentially be used to tutor students by providing automated word suggestions based on what has already been written in the context of a given writing prompt. The ideal tutoring system would make suggestions that are both grade-appropriate and prompt-specific. The data for the student essays [5] came from a publicly held competition initiated on Kaggle, a platform for machine learning competitions. We were interested in generating "good" student writing at this grade level, only considering essays which were scored highly (top third of graded essays) from a particular prompt. This "good quality" data set included 472 essays, with an average of 541 words per essay. Twenty-three essays were used as a hold-out set for validation, while the rest were used for training. There was a total of 7485 unique words in the entire set.

Essay Results

In the essay case, the LSTM was successful in generating samples of text that emulate the underlying structure of the data. To ask the model to generate text, one simply needs to start with a “seed” context, and the LSTM will then produce predictions for the next word in the sequence. For the sake of example, we randomly sampled a seed of 10 words (not necessarily at the start of a sentence or essay) from an actual student essay: “hit the ball. Correctly. Tennis also increase your” and asked the model to generate 20 new predictions. The model produced: “way to get out from the computer. The computer is a great way to communicate. If you ...” Although this English sounds choppy at first glance, the LSTM has picked up elements of structure in English relative to the data it has been trained on, which consists of middle school level writing and a particular prompt. It is also important to remember that the LSTM starts with absolutely no knowledge of English, and simply learns both long- and short-term relationships between words from the training data. These results were expected given the success of these methods in other research contexts surrounding languages [2].

Like any neural network machine learning architecture, these methods greatly run the risk of overfitting to the training data, especially in the present case since our training set was quite small. One metric that can evaluate the generalizability of the model is accuracy on a held out validation set. In this case, “accuracy” refers to predicting the next word given some context. At the start of training, accuracy on a validation set of essays starts at .05, which means the model is simply guessing the most common word, which in this case is a period. After 150 iterations of training, the model has learned to predict words other than periods based on the previous context. During training, prediction accuracy on the held out set peaked at .21. Thus, out of 7485 possible words to choose from, the model is predicting the next word a student will write with 21% confidence.

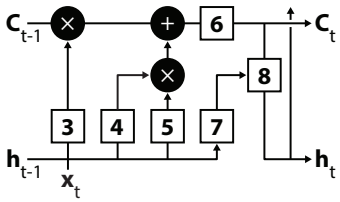

Figure 2: An LSTM cell, where the numbers correspond to previously mentioned update equations

LSTM on MOOC streams

Constructing a model to predict student actions has the potential to augment automated recommendation systems, so that the MOOC system might be able to automatically suggest a resource to go to when a learner is struggling. Indeed, previous work has shown that relatively simple topical N-gram models based on clickstream logs can capture student activity patterns associated with success in a course [6]. We applied the same LSTM architecture from the previous section to student clickstream data from a BerkeleyX MOOC from Spring 2013. There were 27 dis-

Event	Count
Problem view	4877414
Page view	3114154
Video play	2488490
Page close	1878973
Video pause	1531153
Problem check	1370943
Forum view inline	703938
Seq goto	415883
Problem save	171160
Wiki view	136020
Problem show answer	114745
Seq next	78672
Forum view inline	76836
Seq prev	20428
Forum search	19580
Forum upvote	13512
Forum reply	11350
Forum view user profile	5639
Forum create	4414
Forum view followed threads	4180
Forum follow	1583
Forum update	1314
Forum unvote	1234
Forum unfollow	442
Forum delete	355

Figure 3: Logged student action types from all students

tinct student actions logged, which are shown in figure 3. Note that this figure includes log data from all students enrolled in the MOOC, not just those who ended up certified. The MOOC was a 5-week course with a midterm and a final. We chose to train on student action data from students who were considered "certified" by the end of the course, which primarily reflects passing grades on the homework assignments, midterm, and final. There were a total of 8094 distinct learners in the data set, with an average of 1595 actions per learner.

MOOC Results

In the MOOC case, the LSTM was unable to learn beyond predicting better than the majority class for samples of student actions. This result shows that the granularity at which the student action data was fed into the model did not have an underlying pattern governing the actions that was consistently predictive across different students. The next steps in applying our LSTM model to student clickstream data will aim to improve the granularity of the student action sequence data by including more specific information about each action, such as an indication of which problem or unit the learner is accessing, with the goal of uncovering an underlying pattern that generalizes across students to predict which specific learning objects a student will interact with next. Future work applying this model is currently underway and has improved prediction beyond baseline.

Conclusion

This work-in-progress paper detailed preliminary results in applying deep recurrent neural network models to two sets of education data. Future work will attempt to extend and augment these models with the hope of creating tutoring systems that can generate helpful recommendations to students based on their previous actions and choices. Recurrent neural networks and other deep learning techniques are impressively finding insights in many fields when it comes to large sets of often uncategorized data, and there are clear applications where education and learning can be improved, specifically where large amounts of student actions have already been collected.

Acknowledgement

This work was supported by a grant from the National Science Foundation (IIS: BIGDATA 1547055).

References

[1] Albert T Corbett and John R Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction* 4, 4 (1994), 253–278.

[2] Alex Graves. 2013. Generating Sequences With Recurrent Neural Networks. *CoRR* abs/1308.0850 (2013). http://arxiv.org/abs/1308.0850

[3] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[4] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep Knowledge Tracing. In *Advances in Neural Information Processing Systems*. 505–513.

[5] Mark D. Shermis. 2014. State-of-the-art automated essay scoring: Competition, results, and future directions from a United States demonstration. *Assessing Writing* 20 (2014), 53 – 76.

[6] Miaomiao Wen and Carolyn Penstein Rosé. 2014. Identifying latent study habits by mining learner behavior patterns in massive open online courses. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*. ACM, 1983–1986.