Classifying Learner Behavior from High Frequency Touchscreen Data Using Recurrent Neural Networks

Zachary A. Pardos
UC Berkeley
Berkeley, CA, USA
zp@berkeley.edu

Changran Hu
Tsinghua University
Haidian Qu, Beijing Shi, China
huchangran@gmail.com

Pengqiu Meng
Wuhan University
Wuhan, Hubei, China
mengpengqiu@whu.edu.cn

Michael Neff
UC Davis
Davis, CA, USA
mpneff@ucdavis.edu

Dor Abrahamson
UC Berkeley
Berkeley, CA, USA
dor@berkeley.edu

ABSTRACT

Sensor stream data, particularly those collected at the millisecond of granularity, have been notoriously difficult to leverage classifiable signal out of. Adding to the challenge is the limited domain knowledge that exists at these biological sensor levels of interaction that prohibits a comprehensive manual feature engineering approach to classification of those streams. In this paper, we attempt to enhance the assessment capability of a touchscreen based ratio tutoring system by using Recurrent Neural Networks (RNNs) to predict the strategy being demonstrated by students from their 60hz data streams. We hypothesize that the ability of neural networks to learn representations automatically, instead of relying on human feature engineering, may benefit this classification task. Our RNN and baseline models were trained and cross-validated at several levels on historical data which had been human coded with the task strategy believed to be exhibited by the learner. Our RNN approach to this historically difficult high frequency data classification task moderately advances performance above baselines and we discuss what implication this level of assessment performance has on enabling greater adaptive supports in the tutoring system.

KEYWORDS

Recurrent neural networks, high frequency data, touchscreen, sensors, tutoring systems, embodied cognition, assessment

ACM Reference Format:
Zachary A. Pardos, Changran Hu, Pengqiu Meng, Michael Neff, and Dor Abrahamson. 2018. Classifying Learner Behavior from High Frequency Touchscreen Data Using Recurrent Neural Networks. In *UMAP'18 Adjunct: 26th Conference on User Modeling, Adaptation and Personalization Adjunct, July 8–11, 2018, Singapore, Singapore.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3213586.3225244

UMAP'18 Adjunct, July 8–11, 2018, Singapore, Singapore

ACM ISBN 978-1-4503-5784-5/18/07…$15.00
https://doi.org/10.1145/3213586.3225244

1 INTRODUCTION

We explore how patterns in students' high frequency touchscreen data can be detected in order to classify strategic behaviors exhibited while learning in an embodied mathematics tutor. The particular application is the Mathematics Imagery Trainer for Proportionality (MITp), a teaching tool in which students learn the concept of proportion, in advanced of it being introduced in the standard school curriculum, by moving two touch markers on the screen over time to achieve the desired ratio between the distance from the left hand marker and the bottom of the screen and the distance from the right hand marker and the bottom of the screen. The screen the learner is interacting with turns green as the correct ratio is arrived at. The telemetry data consists of these two input points, sampled at 60Hz, with the students' activity labeled post-hoc by experts at each time step with the strategy being exhibited. The techniques we bring to bear for classifying these strategies include Long Short-Term Memory models, meant to learn representations of these strategies from raw times-series data, and multinomial logistic regression, meant to serve as a baseline for classification. The goals of this work are to explore the classification of high frequency data using deep learning methods and to extend previous research [1] in order to enable an animated virtual pedagogical agent in the tutor to more effectively guide students through the MITp learning process.

2 BACKGROUND

The use of high frequency sensor data to adapt instruction or contribute to the understanding of the learning process has been used in a variety of contexts. To name only a few, these included using galvanic skin response and other sensors to measure affective states [3], using eye-tracking to study how learners construct knowledge using various graphical representations of concepts [21], and using eye-tracking to diagnose cognitive traits [20]. In these works, the sequence data were pre-processed to a coarser level of frequency before use. The use of these data has been referred to as multimodal learning analytics [4], often to describe data collected from learning contexts in which the learning process is not satisfactorily characterizable from clickstream events or response logs, but rather from other modes of interaction.

The era of deep-learning has catalyzed the use of recurrent neural networks for a variety of time-series tasks outside of education. This

technique has seen nascent adoption by the User Modeling community in predicting within-session clickstream in an e-commerce setting [15] and social [16] recommendation among other tasks in recommendation [9, 23]. The uptake of this deep neural network modeling in education contexts has begun with its application to clickstream, as seen in affect detection from tutor response logs [5], attention levels expected from educational videos [13], tracing skills in a tutor [19], and predicting clickstream events in MOOCs [24]. While application of deep-nets to clickstream data has been largely successful, their application to high-frequency sensor data has struggled to produce above-baseline results [22].

2.1 The Proportionality Tutor App

Our data were obtained from an application called the Mathematics Imagery Trainer for Proportionality (MITp) [12], collected as part of an effort to enable adaptive virtual agent tutoring to children as they use MITp. MITp is an activity design architecture developed to support students in learning the contents of ratio and proportion, an important yet difficult topic for many students. Understanding proportionality involves appreciating multiplicative relations between extensive quantities; a change in one quantity is always accompanied by a change in the other, and these changes are related by a constant multiplier [14, 26]. Our MITp approach to support students in developing multiplicative understanding of proportions draws on embodiment theory, which views the mind as extending dynamically through the body into the natural-cultural ecology [2]. Thus human reasoning emerges, and is expressed through situated sensorimotor interactions [1]. The MITp system (Figure 1) poses the physical coordinative challenge of moving two hands on a touch screen to make it green, a result which occurs when the ratio of hand heights matches the pre-programmed ratio of 1:2. Through engaging in this embodied-interaction activity and building particular movement schemes related to proportions process, students can develop pre-symbolic quantitative understanding of the mathematical notion. By then introducing specific tools into the environment, here a grid and numbers, students are given progressively more mathematical tools with which to express those strategies.

MITp has traditionally been used with a human technician who explains how the system works, sitting beside the child and making suggestions as they interact. The technician also decides when to introduce scaffolding artifacts, such as grid lines or numbers. Interaction sessions can last an hour and require a trained technician to guide every student through the process. This limits the ability to scale up MITp to a large audience. Previous work introduced a virtual pedagogical agent with a limited ability to guide students through the process [1]. It remains an open challenge to enable automatic assessment at the strategy level and to provide subsequent appropriate instruction, valorization, providing of correction hints, asking of questions that provoke reflection, and introduction of new artifacts. The nature of strategies exhibited during interaction with MITp have been deduced through manual analysis of data streams collected during tutoring in previous work and were determined to be helpful states to transition through while learning [7]. Three distinct strategies were identified in past work as most important and are described below.

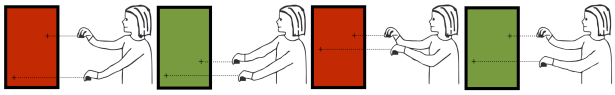


Figure 1: the Mathematics Imagery Trainer (MITp)

In trying to achieve a green screen background (and thus the 1:2 ratio of cursor positions), they can discover "the higher, the bigger": that the gap between their hands is bigger as they make the same proportion higher on the screen. Moving one hand and then adjusting the other leads to the insight "A-per-B": that when they move their left hand one unit, they must move their right two (for 1:2). When they create continuous green, they learn that the "speed" of the right hand is twice that of the left. If we can determine which strategy the child is deploying, this gives great insight into their learning process and is therefore powerful information in determining the most effective response of a pedagogical agent. For example, it can indicate the need to provide remedial instruction. Successful performance of a strategy can indicate that it is time to encourage the child to explore other strategies. Successful performance of the set of strategies can indicate that it is time to introduce new artifacts and advance to the next stage in the tutorial process.

3 DATASET

We used chronological time-stamped sequences of interactions of students with the tutor that record the student touch locations as they progress through instructional prompts with the tutor. The dataset contains 49 students' csv files, of which only 5 were labeled, seen in Table 1. The length of each data file ranged from 39,736 to 176,283 time slices, lasting from 11 minutes to 48 minutes and sampled at 60 hz. For the training of our models, we used only the 5 labeled students, whose descriptive stats can bee seen in Table 4. Two students did not make it to the last of three phases of the tutor session, and thus never demonstrated the **SP** (speed) strategy. While we could have filtered this out, low subject count and missing values are common place real-world classification tasks and thus, we chose to include the label to keep the task authentic and because it was an important label to predict in the pedagogical scheme of the tutor. While filtering out students with such missing data may be common place when more subjects are on hand, further reducing our subject pool would not have been tenable. The data shown in Table 1 included:

(1) Left touch and Right touch indicate the locations of the left most finger and the right most finger on the 2-D coordinate plane of the touchscreen. In practice, only the Y values are used, as it is the ratio of the two heights that affect satisfaction of the task. NA values represent no touch during that 60hz reading.
(2) Color indicates how close the screen is to the goal color of green. When the color value is 1, this means that the student has achieved the target ratio of 1:2 between the heights of her two fingers.

Table 1: Data Examples

(A) Raw Data from one student. This is example showing one reading per second for demonstrating purposes. The actual data we use is sampled at 60 readings per second.

Left Touch	Right Touch	Color	Time
(-3.0 2.9)	(3.4 4.2)	0.561	08:13.000
(-3.0 3.1)	(3.4 8.7)	0.772	08:14.017
(-3.0 3.1)	(3.4 9.3)	1.000	08:15.033

(B) Data format after processing

Left-y	Right-y	Color	ID	Prompt	Label
2.9	4.2	0.561	EL	1003	D
3.1	8.7	0.772	EL	1003	D
3.1	9.3	0.940	EL	1003	D

Table 2: Instructional Prompts

Prompt ID	Text displayed to students
1003	"Your goal is to make the screen green"
1004	"You make the screen green by moving the cursors"
1005	"You can move the cursor up and down like this"

Table 3: Label Definitions. The labels marked as strategy labels are various ways in which students can find or maintain the proper ratio between two fingers and thus keep the screen green.

		Label	Name	Description
extended labels	strategy	**D**	Drag one hand	Keeping one hand still, exploring with the other.
		AB	A per B	Step-wise movement
		SP	Speed	Simultaneous movement.
	non-strategy	NT	No Touch	
		IT, RT	Initial Touch, Release Touch	
		TL, DBH	The Ladder, Drag Both Hands	
		T, NV	Tuning, Not Visible	
		H, O	Horizontal movement, Other	

(3) Time shows time-stamp taken from the data stream file. The data were collected at the 60hz level and logged in chronological order.
(4) Prompt is the task given to the student by the tutor, with examples shown in Table 2.
(5) Label is the human coded strategy exhibited by the student in attempting to complete the task of turning the screen green at various segments throughout their interaction with the tutor. The collection of labels can be seen in Table 3.

Table 4: Descriptive stats of the 5-labeled students showing the number of time slices coded with each of the three strategy labels

	AB	D	SP	Total
EL	1740	15780	5940	23460
ER	5100	3960	0	9060
KN	6180	2700	8040	16920
MS	4680	6120	4140	14940
ND	14640	13500	0	28140

4 METHODS

In this section we describe a multinomial logistic baseline for classification, the LSTM classification model, and a variety of cross-validations used to evaluate the models which correspond to different ways in which the models could be applied in the tutor.

4.1 Multinomial Logistic Regression

We used a standard multinomial logistic regression model to serve as the baseline classifier. We used six very simple features as the input features to the model at every time slice: (1) Left hand-y coordinate (2) Right hand-y coordinate (3) Student ID represented as a one-hot dummy variable (4) Whether or not at least on finger was touching the screen (5) Instructional prompt represented as a one-hot dummy variable (6) Boolean representing if the student had reached the goal state of exhibiting the 1:2 ratio (within the tutor's specified margin of error). At every time slice, these features were used to classify the distinct label associated by an expert with the behavior exhibited at that time.

This is a baseline approach as it represents the degree to which a class can be predicted by the student, prompt, and instantaneous value of the left and right finger. The RNN-based method, described in the next subsection, is hypothetically better equipped to identify a pattern characterized from a temporal series of finger movements.

4.2 Long Short-Term Memory Classification

Long Short-Term Memory is an augmentation to the classic Recurrent Neural Network (RNN) model [17] and was first proposed by Sepp Hochreiter and Jurgen Schmidhuber [11]. It maintains a hidden state and a "longer-term" cell state. Through its architecture [10], it has been demonstrated as being able to classify patterns based on longer sequences than an RNN; diminishing the phenomenon known as the vanishing gradient. In our classification task, we choose an LSTM as the model of choice with a times series of left and right hand coordinates as the inputs and the hand coded labels as the categorical outputs.

We used the 60Hz sampled x, y position of both hands (4 features in total) as input, and used the labels (in one-hot representation) as ground truth. The model will predict the current label with current input every time step or every entire sequence, depending on the specific cross-validation setting discussed in the evaluation section.

Because labels only exist for part of the whole time series, we use a default value -1 to pad all the time when no label exists. A custom loss function is defined to only calculate the loss when the

label is not the default value. A small search of the LSTM's hyperparameters [10] was conducted where the dimension of the hidden state (64, 128, 256) and optimizer (RMSprop, Adamax, Adagrad) was varied. The programmatic framework used to define the models and run the experiments was python's Keras [6] using Theano [25] as backend.

5 EVALUATION

We constructed a set of experiments which evaluated different generalizing properties of the models. All experiments were tested on both the expanded label set and the set only containing the *strategy* labels. In addition to training the models on the entire student sequence, we also tried partitioning the sequences into different segments (or chops) in order to reduce the length of the sequences being trained on and to explore if the model would better generalize when training on larger batches of sequences instead of fewer longer sequences. We also varied the frequency of the predictions being made between (1) predicting the label at every time slice (60hz) and (2) only predicting the presence of the label in the sequence at the very end of the sequence, as defined by the segmenting. When predicting at the sequence level, a binary prediction was made for every label for every sequence. Accuracy was calculated based on the aggregate performance of all binary predictions. The topology for this sequence prediction model was the same as the time slice LSTM except that instead of a softmax over all the labels at every time slice, there were instead independent sigmoid outputs for each label occurring at the very last time slice of the sequence. Five-fold cross-validation is conducted at both the student level and sequence chop level for all experiments. The data were separated into 5 folds and each fold served as the test set once, with the rest serving as the training set. The results from the 5 phases were averaged to produce a single accuracy metric.

5.1 Sequence segmentation

Besides using the original whole sequence, we chop the sequence into segments in two ways (1) Chop by prompts: during the tutor sessions, students were given prompts to direct them to adjust their movements. Accordingly, there is a column named 'prompt' illustrating some specific instructions given at that time period. A chop segmented by prompt runs from the start of a prompt to the start of the next prompt, normally including some time after the prompt during which the student is interacting with the system and the tutor is silent. This sequence chop approach produced 162 total segments of various lengths. (2) Chop by labels: domain experts labeled the behavior from recordings of sessions based on the movement pattern exhibited by the student. Chop by label segmented the sequences using contiguous labels and produced 1239 chops of various lengths. We note that this level of segmentation would not practically be available in a real-world scenario as it requires knowledge of the label beforehand. Nevertheless, this can serve as a test of smaller sequence length segmentation for classification.

5.2 Hypotheses

We enumerate the following expectations for the classification results:

- **H1:** Compared to logistic regression, the LSTM has the ability to learn the chronological information in the sequence. Therefore, we expect that the LSTM models will perform better on average than logistic regression.
- **H2:** Different students may exhibit strategies in different ways. Therefore, we anticipate that models that have trained on some portion of the student they are predicting (sequence level cross-validation) will perform better than their respective student level cross-validation experiments.
- **H3:** We assume predicting the label at the frequency of every sequence will be more accurate than predicting every timestep due to the former being the easier scenario since the model need only predict if a label occurred and not the temporal sequencing in which they occurred within the sequence.
- **H4:** After restricting to the strategy label set, the classifications should be easier to learn compared to the expanded label set since there will be less opportunity for similar labels to be confused for one another.

6 RESULTS

Results in terms of accuracy of the majority class baseline (B), multinomial logistic regression (LR), and Long short-term memory model[1] (LSTM) are shown for predicting the strategy label set (Table 5) and the extended label set (Table 6).

In all but two of the experiments, the LSTM outperformed the baseline models, mostly confirming **H1**. Focusing on the results of the strategy label set which were cross-validated at the student level and made predictions of labels at the moment-by-moment (time step) frequency, we see that the majority class and logistic models predicted the same across all sequence chop experiments. This was because the training of neither baseline is affected by the chop level - the logistic was trained using instantaneous independent readings from each time slice, whereas the LSTM's hidden state, and thus predictions, are affected by inputs from previous time slices within the chop. The LSTM most benefited from training and predicting using the entire student's sequence, not training on partitions (chops) of the sequence. This indicates that the LSTM was able to leverage signal from previous prompts in making predictions of labels in the current prompt. The LSTM scored 20% above the logistic and 51% above majority class in this evaluation category. When dealing with many more classes in the extended label set, an improvement over logistic was only seen when training and predicting on more granular label or prompt segmented sequences.

Contrary to **H2**, the sequence level cross-validation outperformed its student level counterpart in only a few experiments (strategy labels/sequence cv/sequence freq./prompt chop and extended labels/sequence cv/time step freq./no label chop). This perhaps suggests that there is less importance to pickup on a "signature" of a student being predicted by observing some of her behaviors in the training set.

As expected, classification at the label frequency of sequence (**H3**) was by in large the easier classification task, only performing worse than its by-time-step counterpart in one out of the 10

[1]The LSTM model trained with a 256 node hidden layer and RMSprop optimizer consistently performed best in the hyperparameter search.

Table 5: Accuracy of Strategy Labels

CV_by	label frequency	sequence chop	B	LR	LSTM
student	by time step	no chop	31.2	39.29	47.1
		label chop		39.29	45.3
		prompt chop		39.29	42.4
	by seq.	no chop	66.6	NA	86.7
		label chop	47.1	NA	46.6
		prompt chop	71.7	NA	79.5
seq.	by time step	no chop	31.2	40.6	NA
		label chop	15.9	24.88	39.0
		prompt chop	43.4	43.5	37.1
	by seq.	no chop	NA	NA	NA
		label chop	47.5	NA	33.0
		prompt chop	73.7	NA	89.8

Table 6: Accuracy of Extended Labels

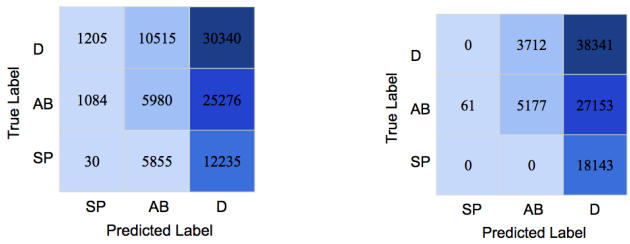
CV_by	label frequency	sequence chop	B	LR	LSTM
student	by time step	no chop	13.9	17.9	14.8
		label chop		17.9	18.3
		prompt chop		17.9	21.0
	by seq.	no chop	82.9	x	88.6
		label chop	15.2	NA	26.9
		prompt chop	75.5	NA	90.5
seq.	by time step	no chop	NA	NA	NA
		label chop	22.2	23.8	24.3
		prompt chop	18.7	22.7	18.5
	by seq.	no chop	NA	NA	NA
		label chop	21.2	NA	26.7
		prompt chop	77.0	NA	87.0

experiments across label sets (strategy labels/sequence freq./label chop).

Finally, it can also be observed that in comparing the 10 LSTM results of the strategy label set to the expanded label set, in only one experiment does the expanded label set perform better (student cv/sequence freq./prompt chop), mostly confirming **H4**.

6.0.1 Notable Null Results. Additional methods were attempted to improve the classification but did not enhance the results. They were abandoned after the early or middle stages of evaluation and are reported here for posterity. We tried incorporating additional input features into the LSTM such as the color of the screen, the current prompt, and student identifier but no improved accuracy was found. We also tried different preprocessing procedures, such as downsampling the raw data to one reading per second (instead of 60) so that the entire sequence length could be reduced and so patterns could be better identified at a less granular level. We also tested different model structures, including multiple LSTM hidden layers, Phased LSTM [18], different activation functions, GRUs, and simple RNNs, but no improvements were seen.

6.0.2 Discussing the Suitability of these Results for LSTM Integration into the MITp App. It can be seen in the confusion matrices (Fig. 2) that both models do a reasonable job of correctly predicting



(a) logistic regression **(b) LSTM classification**

Figure 2: Confusion matrices for the logistic and LSTM models based on the strategy label set and experiment with student level CV, with time step label frequency, and trained on the full student sequence (no chop).

the D label when D was actually employed, but the LSTM classification is much better (91% LSTM vs. 72% logistic). Unfortunately, both models have a low true positive rate for the other labels; 18% (logistic) or 16% (LSTM) for AB and almost 0% for SP, a label which did not appear in two out of the five students' sequences. This is because of the high false positive rate for D (55% for logistic and 54% for LSTM). Essentially, both models report D much more frequently than the ground truth data. D is one of the earliest strategies students use when finding proportions in MITp, so the false positive rate is not overly problematic for integration into the tutor. It tells the system that the child may be doing something more simple than they are actually doing, but this will lead to simply repeating instructions to try to create a higher level behavior, which is less pedagogically problematic than skipping ahead prematurely, and repetition may even offer learning benefits.

A particularly interesting finding is the false positive rate for AB. This is 73% for logistic regression, but only 42% for LSTM, so when the LSTM predicts AB, it is much more likely to be correct. Detecting correct performance of AB is particularly useful for an autonomous tutor design as AB is the second strategy taught to a child while using the system and successful performance of AB is a pre-cursor for moving on to teaching SP. While an approximately 60% correct rate for AB labels is not sufficient on a single instance to decide the child is ready to move on, when this occurs multiple times and is combined with knowledge of where the system is in the overall tutoring process, it provides a useful signal.

7 CONCLUSIONS

Methodologically, the application of LSTMs to our dataset of 60hz touchscreen sensor data was successful in realizing a moderate gain in classification performance. It achieved a 47.1% accuracy in predicting the moment-by-moment strategy being employed by the student, compared to 39.29% accuracy when using logistic regression with simple features and 31.2% when classifying based on the majority class of the training set. If, instead of predicting the strategy at every moment, the classification predicts if the strategy was ever employed by the student, the accuracy climbs to 86.7% with an RNN, compared to 66.6% using the majority observation (or non-observation) of each class across students in the training set.

8 DISCUSSION AND FUTURE WORK

Students devise a broad variety of sensorimotor schemes for enacting a target movement [2, 7, 8, 12]. Knowing what schemes students are employing is critical for supporting their learning, and yet determining these schemes has been a challenging engineering task. The difficulty that we encountered in modeling students' schemes thus corroborates expectations coming from constructivist and enactivist theories, viz. that the mind and thus learning is highly individualistic and thus difficult to model in terms of generalizable qualities. Nonetheless, progress was made in establishing the improved performance of LSTMs which have now produced predictions that would be actionable in the MITp application. Most notably, the accuracy of AB labels provides an a signal the tutor can use in determining if the child is ready to advance to the more complicated *Speed* strategy.

There are several future directions that may prove profitable. The lessons in the MITp tutor go through several phases: guided exploration, using the A-per-B strategy, and then using the Speed strategy. Including information about the current phase of the tutoring process in the analysis may improve labeling prediction as different labels are more likely in different phases. The importance of accuracy for different labels also varies depending on the phase, so results could be more effectively interpreted if this information is included. A different approach would be to include additional multi-modal input data. Eye-tracking analyses may offer a promising direction [8], if applied in real-time and in concert with the touchscreen stream. Incorporating eye-tracking could also take us beyond the hand coded 'strategy' labels, which represent *what* a student is performing, objectively, as opposed to the interpretation of *how* a student is orienting toward the enactment of a movement. It is expected that with a larger set of labeled students and the ability to limit the imbalance of the *Speed* class, classification accuracy would improve to levels which would justify deeper integration of these models in the tutor. Finally, data from the unlabeled set of students could be brought to bear in order to explore, instead of classify, the common patterns of learning behavior exhibited during each phase of tutoring. These patterns could then be reconciled with subject matter expert tagged strategies to deepen our understanding of the mechanics of learning in this context.

ACKNOWLEDGMENTS

Support for this work was partially provided by the National Science Foundation through grant awards 1320029, 1321042, and 1547055.

REFERENCES

[1] Ahsan Abdullah, Mohammad Adil, Leah Rosenbaum, Miranda Clemmons, Mansi Shah, Dor Abrahamson, and Michael Neff. 2017. Pedagogical Agents to Support Embodied, Discovery-Based Learning. In *Intelligent Virtual Agents*, Jonas Beskow, Christopher Peters, Ginevra Castellano, Carol O'Sullivan, Iolanda Leite, and Stefan Kopp (Eds.). Springer International Publishing, Cham, 1–14.

[2] Dor Abrahamson and Arthur Bakker. 2016. Making sense of movement in embodied design for mathematics learning. *Cognitive Research: Principles and Implications* 1, 1 (12 2016), 1–13. https://doi.org/10.1186/s41235-016-0034-3

[3] Ivon Arroyo, David G Cooper, Winslow Burleson, Beverly Park Woolf, Kasia Muldner, and Robert Christopherson. 2009. Emotion Sensors Go To School.. In *AIED*, Vol. 200. 17–24.

[4] Paulo Blikstein. 2013. Multimodal learning analytics. In *Proceedings of the third international conference on learning analytics and knowledge*. ACM, 102–106.

[5] Anthony F Botelho, Ryan S Baker, and Neil T Heffernan. 2017. Improving Sensor-Free Affect Detection Using Deep Learning. In *International Conference on Artificial Intelligence in Education*. Springer, 40–51.

[6] François Chollet. 2015. keras. https://github.com/fchollet/keras.

[7] Abrahamson D., Lee R. G., Negrete A. G., and Gutiérrez J. F. 2014. Coordinating visualizations of polysemous action: Values added for grounding proportion. In *ZDM Mathematics Education, Visualization as an epistemological learning tool [Special issue]*, F. Rivera, H. Steinbring, and A. Arcavi (Eds.). Vol. 46. 79–93. https://doi.org/10.1007/s11858-013-0521-7

[8] Abrahamson D., Shayan S., Bakker A., and Van der Schaaf M. F. 2016. Eye-tracking Piaget: Capturing the emergence of attentional anchors in the coordination of proportional motor action. *Human Development* 58, 4-5 (2016), 218–244.

[9] Robin Devooght and Hugues Bersini. 2017. Long and Short-Term Recommendations with Recurrent Neural Networks. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization (UMAP '17)*. ACM, New York, NY, USA, 13–21. https://doi.org/10.1145/3079628.3079670

[10] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. 2017. LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems* 28, 10 (2017), 2222–2232.

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

[12] Mark Howison, Dragan Trninic, Daniel Reinholz, and Dor Abrahamson. 2011. The Mathematical Imagery Trainer: From Embodied Interaction to Conceptual Learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 1989–1998. https://doi.org/10.1145/1978942.1979230

[13] Stephen Hutt, Caitlin Mills, Nigel Bosch, Kristina Krasich, James Brockmole, and Sidney D'Mello. 2017. "Out of the Fr-Eye-ing Pan": Towards Gaze-Based Models of Attention during Learning with Technology in the Classroom. (07 2017), 94–103.

[14] Lamon S. J. 2007. Rational numbers and proportional reasoning: Toward a theoretical framework. In *Second handbook of research on mathematics teaching and learning*, F. Lester (Ed.). Charlotte, NC: Information Age Publishing, 629–668.

[15] Dietmar Jannach, Malte Ludewig, and Lukas Lerche. 2017. Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. *User Modeling and User-Adapted Interaction* (2017), 1–42.

[16] Antoine Lefebvre-Brossard, Alexandre Spaeth, and Michel C. Desmarais. 2017. Encoding User As More Than the Sum of Their Parts: Recurrent Neural Networks and Word Embedding for People-to-people Recommendation. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization (UMAP '17)*. ACM, New York, NY, USA, 298–302. https://doi.org/10.1145/3079628.3079700

[17] LR Medsker and LC Jain. 2001. Recurrent neural networks. *Design and Applications* 5 (2001).

[18] Daniel Neil, Michael Pfeiffer, and Shih-Chi Liu. 2016. Phased LSTM: Accelerating Recurrent Network Training for Long or Event-based Sequences. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc, 3882–3890.

[19] 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.

[20] George E. Raptis, Christina Katsini, Marios Belk, Christos Fidas, George Samaras, and Nikolaos Avouris. 2017. Using Eye Gaze Data and Visual Activities to Infer Human Cognitive Styles: Method and Feasibility Studies. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization (UMAP '17)*. ACM, New York, NY, USA, 164–173. https://doi.org/10.1145/3079628.3079690

[21] MA Rau, HE Bowman, and JW Moore. 2016. Intelligent technology-support for collaborative connection-making among multiple visual representations in chemistry. *Structure-Function Relationships in the Gas-Sensing Heme-Dependent Transcription Factors RcoM and DNR* 1001 (2016), 178.

[22] Martina A Rau and Zachary A Pardos. 2016. Adding eye-tracking AOI data to models of representation skills does not improve prediction accuracy.. In *EDM*. 622–623.

[23] Alessandro Suglia, Claudio Greco, Cataldo Musto, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. 2017. A Deep Architecture for Content-based Recommendations Exploiting Recurrent Neural Networks. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization (UMAP '17)*. ACM, New York, NY, USA, 202–211. https://doi.org/10.1145/3079628.3079684

[24] Steven Tang, Joshua C Peterson, and Zachary A Pardos. 2017. Modelling Student Behavior using Granular Large Scale Action Data from a MOOC. *The Handbook of Learning Analytics* (2017), 223–233.

[25] Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints* abs/1605.02688 (May 2016). http://arxiv.org/abs/1605.02688

[26] Boyer T.W. and Levine S.C. 2015. Prompting children to reason proportionally: Processing discrete units as continuous amounts. *Developmental psychology* 51, 5 (2015), 615–620.