



Beyond Black-Boxing: Building Intuitions of Complex Machine Learning Ideas Through Interactives and Levels of Abstraction

Brian Broll
brian.broll@vanderbilt.edu
Vanderbilt University
Nashville, Tennessee, USA

Shuchi Grover
Looking Glass Ventures
Austin, Texas, USA
shuchig@cs.stanford.edu

Derek Babb
The University of Nebraska Omaha
Omaha, Nebraska, USA
dvbabb@unomaha.edu

ABSTRACT

Existing approaches to teaching artificial intelligence and machine learning often focus on the use of pre-trained models or fine-tuning an existing black-box architecture. We believe advanced ML topics, such as optimization and adversarial examples, can be learned by early high school age students given appropriate support. Our approach focuses on enabling students to develop deep intuition about these complex concepts by first making them accessible to novices through interactive tools, pre-programmed games, and carefully designed programming activities. Then, students are able to engage with the concepts via meaningful, hands-on experiences that span the entire ML process from data collection to model optimization and inspection.

CCS CONCEPTS

• **Social and professional topics** → **K-12 education**.

KEYWORDS

high school computer science, artificial intelligence, machine learning, levels of abstraction, scaffolding

ACM Reference Format:

Brian Broll, Shuchi Grover, and Derek Babb. 2022. Beyond Black-Boxing: Building Intuitions of Complex Machine Learning Ideas Through Interactives and Levels of Abstraction. In *Proceedings of the 2022 ACM Conference on International Computing Education Research V.2 (ICER '22)*, August 7–11, 2022, Lugano and Virtual Event, Switzerland. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/3501709.3544273>

1 MOTIVATION

“Any subject can be taught effectively in some intellectually honest form to any child at any stage of development.” (Bruner, 1960)

Efforts to teach Artificial Intelligence & Machine Learning (AI/ML) in schools are gaining momentum propelled by developments that allow AI/ML models to be run in a web browser. However, most of these efforts introduce AI/ML models such as classifiers and neural networks through the use of extant pre-trained models. The goal in our project is to help secondary students build deeper understanding of how AI/ML techniques work and how the machine actually “learns”. Our innovation lies in resolving the

challenge of making concepts that involve complex mathematics accessible to students that have not yet learnt those mathematical ideas. Through making apparent real-world connections and the interplay with mathematics in accessible ways, we contend that students will build deeper and better intuitions of ML techniques. We believe that such deeper understanding also aids a more meaningful interrogation of critical issues such as ethics and bias in AI/ML. We draw inspiration from past work in turtle geometry by pioneers such as Abelson and diSessa that made sophisticated ideas in mathematics and physics accessible to young learners through leveraging multiple representations and programming [1].

2 DESIGN APPROACH

Our curricular activities introduce core ML ideas such as classification, optimization, gradient descent, and adversarial examples through a range of unplugged, non-programming, and programming activities in NetsBlox [3] (an extension of Snap!) to engender material engagement and manipulation for constructivist knowledge construction [2]. These activities evolved following feedback from 7 high school teachers who participated in a preliminary workshop. This poster shares our current approach with the goal of soliciting feedback ahead of further refinements prior to classroom implementation.

Data exploration with Common Online Data Analysis Platform (CODAP). CODAP empowers students to explore data to gain insight about underlying patterns. Using CODAP [5] from within NetsBlox allows students to leverage the complementary strengths of each; NetsBlox is used to preprocess or filter segments of the dataset while CODAP is used to gain insight into the dataset as well as select regions of interest for further exploration (Figure 1). In the context of ML, this is powerful when building rule-based classifiers or decision trees.

Pre-programmed games. Pre-programmed games are used to introduce fundamental ML topics in an interactive, engaging way prior to programming activities (Figure 2).

Programming at multiple levels of abstraction. In order to help learners at all levels of interest and ability succeed in engaging with non-trivial ML algorithms, we employ “levels of abstraction” [4, 5] as a scaffolding tool. We introduce the basic algorithm in pseudocode, then provide subgoal-like design blocks for implementing the algorithm and Parson’s problems for code completion (Figure 3), and at the lowest level, the entire code that implements the algorithm. Regardless of the level(s) at which learners progressively engage, we believe they all leave with an intuition of how the ML technique works. This will be examined empirically in our upcoming research study.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ICER '22, August 7–11, 2022, Lugano and Virtual Event, Switzerland

© 2022 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-9195-5/22/08.

<https://doi.org/10.1145/3501709.3544273>

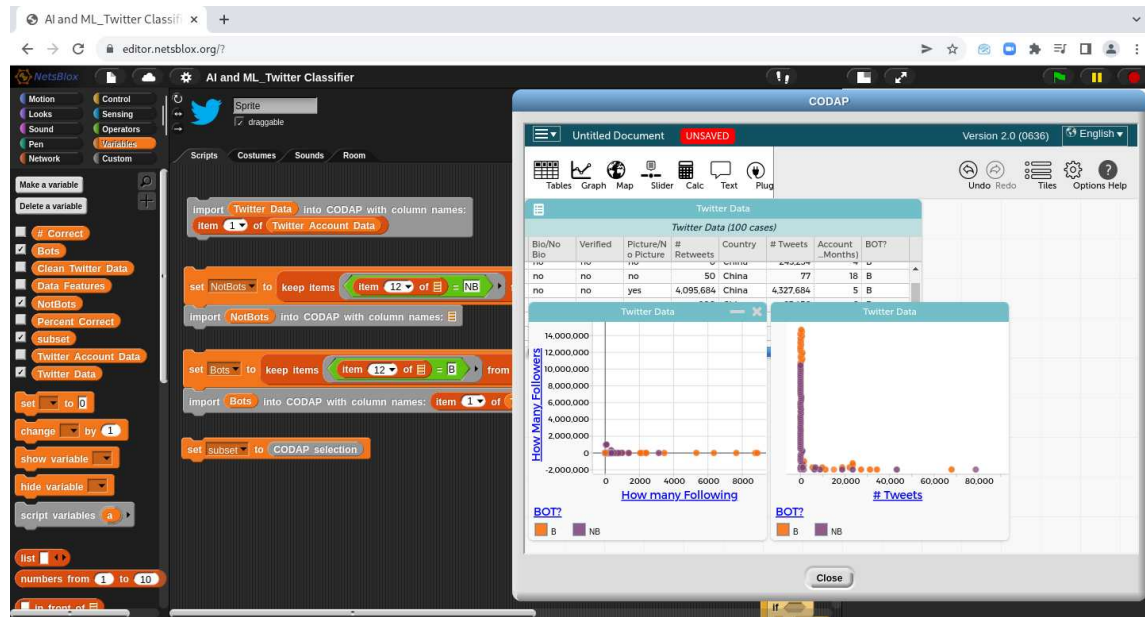


Figure 1: Dynamic data exploration with CODAP within NetsBlox

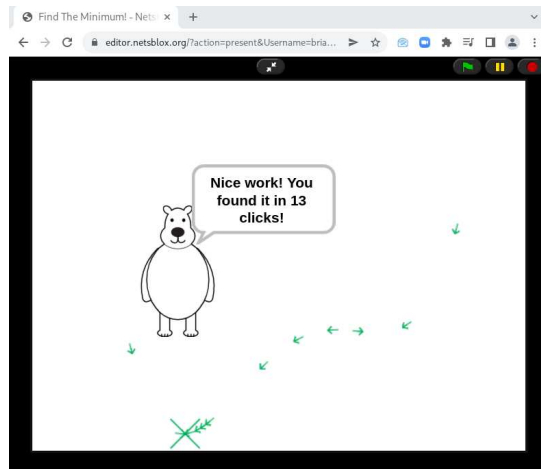


Figure 2: In the “Find the Minimum” game, students embody an optimization algorithm to find the minimum value of an unknown function in order to develop preliminary intuition of how the gradient descent algorithm works prior to coding activities.

ACKNOWLEDGMENTS

This project is supported by a grant from the National Science Foundation (#2113803).

REFERENCES

- [1] Harold Abelson and Andrea DiSessa. 1986. *Turtle geometry: The computer as a medium for exploring mathematics*. MIT press.
- [2] Edith Ackermann. 2001. Piaget’s constructivism, Papert’s constructionism: What’s the difference. *Future of learning group publication* 5, 3 (2001), 438.

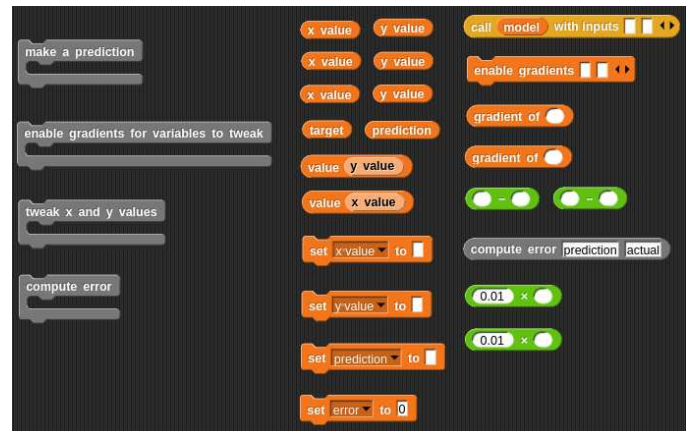


Figure 3: In order to understand a simple algorithm for generating an adversarial example, students can explore the gray blocks on the left that represent subgoals for the algorithm. The gray blocks are also implemented such that students get feedback on a correct or incorrect sequence. Once complete, interested students can use the blocks presented on the right as a Parson’s problem to implement each sub-goal.

- [3] Brian Broll, Akos Lédeczi, Peter Volgyesi, Janos Sallai, Miklos Maroti, Alexia Carrillo, Stephanie L. Weeden-Wright, Chris Vanags, Joshua D Swartz, and Melvin Lu. 2017. A visual programming environment for learning distributed programming. In *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education*. 81–86.
- [4] Andrew Csizmadia, Bernhard Standl, and Jane Waite. 2019. Integrating the constructionist learning theory with computational thinking classroom activities. *Informatics in Education* 18, 1 (2019), 41–67.

- [5] William Finzer. 2016. Common online data analysis platform (CODAP). Emeryville, CA: The Concord Consortium.[Online: concord.org/codap] (2016).