TEACHING MACHINE LEARNING IN SCHOOL: SOME EMERGING RESEARCH TRAJECTORIES

Matti Tedre & Henriikka Vartiainen
University of Eastern Finland

October 27, 2021: Paderborn Colloquium on Artificial Intelligence and Data Science Education at School Level

1

TEACHING MACHINE LEARNING IN SCHOOL: SOME EMERGING RESEARCH TRAJECTORIES

Matti Tedre & Henriikka Vartiainen
University of Eastern Finland
Ilkka Jormanainen, Juho Kahila, Teemu Valtonen, Tapani Toivonen, Arnold Pears

October 27, 2021: Paderborn Colloquium on Artificial Intelligence and Data Science Education at School Level

2

THIS TALK IS BASED ON:

Teaching Machine Learning in K–12 Classroom: Pedagogical and Technological Trajectories for Artificial Intelligence Education	IEEE Access 9, 2021
CT 2.0	Koli Calling 2021
What Makes Computational Thinking so Troublesome?	FIE 2021
Machine learning for middle schoolers: Learning through data-driven design	Int. Jnl of Child-Comp. Interaction

3

COMPUTING EDUCATION IN SCHOOL: A PARADIGM SHIFT LOOMING

4

CLASSICAL PROGRAMMING (IN K12)
)

- The driving force of automation since the 1940s
- A mainstay of computing education
- The paradigm of the Computational Thinking movement of the 2000s
- Well suited for the needs of the software industry

5

THE RULE-DRIVEN PARADIGM IN CSE
(think of Java, Scratch, imperative programming)

Deterministic	Well known notional machines
Stepwise	Avoid trial and error
Unambiguous transition rules	Glass-box testing
Strict syntax	Tracking and tracing program states
Discrete	Deductive problem solving
Highly structured	

6

DATA-DRIVEN AUTOMATION

- Machine learning: breakthrough in the 2000s
- ML is the engine of recommender systems, natural language understanding, speech recognition, …
- Drives many apps and services popular with children
- Well suited for media, unstructured data

7

Picture: Where Google uses machine learning

8

ML IN K12?

9

PILOT STUDY MACHINE LEARNING FOR MIDDLE SCHOOLERS

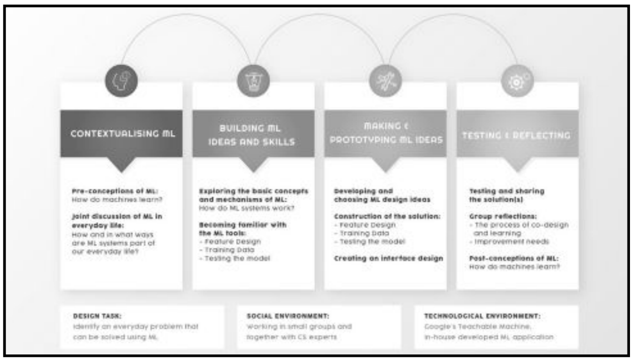
Team: Tapani Toivonen, Ilkka Jormanainen, Juho Kahila, Henriikka Vartiainen, Teemu Valtonen, Matti Tedre

10

RESEARCH DESIGN

- Co-design
- 34 sixth-grade children
- Data collection:
  - Pre/post tests
  - Group discussions, interviews
  - Design ideas and implemented apps

11



12

CONTEXTUALIZING AND EXPLORING ML

13

CHILDREN'S ML IDEAS

HOME AUTOMATION APPLICATIONS
e.g. a gesture-based door-opener, recycling robot

SCHOOL WORK AND HOMEWORK AUTOMATION
e.g. a writing inspector, pupils' attendance detector

SERVICE AUTOMATION
e.g. an automated shopping list, fake product detector

IMPROVING SECURITY AND PRIVACY
e.g. an application that hides other applications, criminal detector for the police

WELL-BEING
e.g. health detector, ambulance caller, mood improver

14

IDEAS SELECTED TO BE FURTHER DEVELOPED AS WEB-BASED ML APPLICATIONS

DESIGN TEAM	GTM'S MODEL TYPE AND DATA	STUDENTS' OWN DESCRIPTIONS OF THEIR SPECIFIC PURPOSE
Group 1 (3 girls and 1 boy)	Image recognition: different colour pictures derived from the Internet and colour paper	"Identification of colors for color-blinds"
Group 2 (3 girls)	Image recognition: Students' own facial expressions and poses	"An app that detects your mood. If you are bored the app will tell you something to do and if you are feeling sad, the app will comfort you. "
Group 3 (5 boys)	Image recognition: pictures from the internet and text books	"When children or adults collect mushrooms and berries, they may not be sure the mushroom or berry is toxic. So it would be good for them to have something that helps them to check it. That's why I thought it would be good to have an application that could check this."
Group 4 (3 boys)	Image recognition: students' hand-written letters	"An application that allows you to take a picture of an essay and it recognize the letters and correct errors automatically."
Group 5 (4 girls)	Image recognition: students' hand-written numbers	"It can check math calculations but also handwriting. So you show the calculations to the camera and if it doesn't understand the handwriting then you need to improve it. Then, when the handwriting is good, it shows whether the calculation is right or wrong."
Group 6 (2 girls and 2 boys)	Image recognition: students' hand-written numbers	"calculator, if you can't count something on your head then you can use it."
Group 7 (4 girls)	Sound recognition: students' own speech	"Vahturi" ("watchman")- When the teacher leaves the classroom, she/he leaves the app to record the speech of the students. The app recognizes who talks and counts how much each student talked."
Group 8 (3 boys and 1 girl)	Sound recognition: students playing their own instruments	"Teachable Machine could be taught to recognize music on different instruments .. and different chords of guitar and other instruments"
Group 9 (3 boys)	Posenet: Students' own poses	"Door opening it with Teachable Machine that recognizes the feelings of people's from their faces, for example, if you are angry, that program also recognizes different positions "

15

DEVELOPMENT OF ML APPLICATION IDEAS

ML design template that asked students to negotiate

- what the app does
- what kind of data are collected and from where (image, sound, poses)
- how many different categories should the model recognize
- under what conditions the teaching data will be given (such as background noise or background setting)

16

CONSTRUCTION OF SOLUTION

Children created training data sets using pictures, poses and 1-second sound clips

17

CREATION OF INTERFACE DESIGN

Interface design of an application to recognize different instruments and chords

18

TESTING & REFLECTING

- Deploying the models
- Presenting the apps to others
- Testing the apps
  - When does it not work? Why?
  - What breaks the model?
  - What makes a model good/bad?
  - How could it be improved?

19

DEVELOPMENT OF CONCEPTUAL UNDERSTANDING:

ML WORKFLOW, TRAINING DATA, CLASSIFICATION, CONFIDENCE, SOFTNESS, BRITTLENESS

Teemu: ***Then it doesn't work***
Interviewer: *Okay. Well*
Timo: ***it took those particular chords that we taught it***
Hanna: ***So, it should have been taught more***
Timo: *Mm*
Interviewer: *Mm. Okay. So, it probably doesn't work in every situation?*
Hanna: *No*
Timo: *No*
Interviewer: *eah. And what do you think is the reason for that or for why it does not work?*
Hanna: ***It doesn't have enough data, for example, about the piano or the guitar, or it has too much information about one and a little less about the others***

Example from the post-test

Except from the interview data

20

DATA AGENCY

- Children noticed that their everyday apps learn when they
  - Listen to music
  - Watch movies
  - Do things online
- Children were noticing and naming data-driven services in their life
- Yet, giving one's personal data was considered as an acceptable trade-off

21

DATA AGENCY

- After the process, students talked about themselves as designers, inventors, collaborators and makers. i.e. positioned themselves as **active subjects** in relation to ML
- They also reflected on the process of design in terms of the change in their experienced agency

22

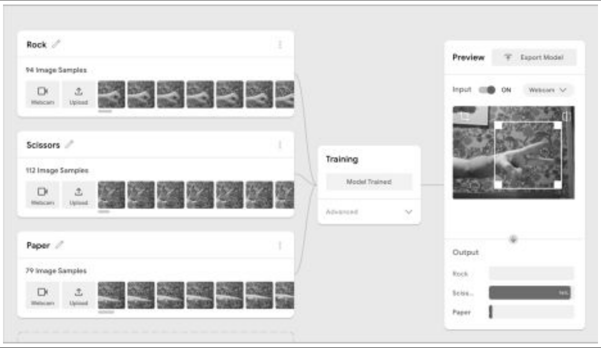
CER ON ML IN K12: OPPORTUNITIES AND NEW HORIZONS

23

1. NEW CLASSES OF MEDIA-HEAVY APPLICATIONS BECOME AVAILABLE

- Anything that allows a lot of data to be collected
  - Pictures
  - Sound
  - Gestures
  - Sensor data
- How would you write a Java/Scratch program that can classify gestures in "rock-paper-scissors" game?
  - Making a ML model for the same is trivial

24



25

2. FROM RULES TO DATA

- Anything that allows a lot of data to be collected can be made into an ML model:
  - Children's drawings
  - Sports activities
  - Gestures, poses
  - Web searches
  - Cartoon pictures
  - Sound clips

26

3. SHIFT IN THE ROLE OF SYNTAX

- Syntax is one of the harder bits in learning programming
- Most common data-driven learning tools at the moment are drag & drop
- But not all:
  - Wolfram Programming Lab
  - eCraft2Learn (Ken Kahn's Snap! tools)

27

Picture: Wolfram Programming Lab

28

4. AGE-APPROPRIATENESS

- ML tools scale well to different age groups
- Our projects have studied different ML/AI tools with
  - 3-year olds (teaching the computer to recognize their moods: angry, sad, happy)
  - Primary schoolers
  - Secondary schoolers
  - High school students (create their own classifier)

29

Pilot study

LEARNING MACHINE LEARNING WITH YOUNG CHILDREN

30

RESEARCH DESIGN

- Participatory learning and design
- Six children (aged 3-9 years-old) and their families
- Data collection:
  - Video recordings
  - Interviews

homes

31

32

5. NATURAL FORMS OF INTERACTION

- Instead of programming language (syntax-driven) interaction, many ML tools take use of
  - Video
  - Pictures
  - Body poses
  - Natural language

33

6. THE ALGORITHMIC STEP

- "From coding to teachable machines" (Druga, 2018)
- In traditional programming one can trace program execution step by step
  - Programs are designed by stepwise rules
- In neural networks "steps" are not key
  - Describing users' intentions is important for getting enough of the right kind of data for the job

34

Interface design of an application to recognize different instruments and chords

35

7. GLASS & BLACK BOXES

- All computing education uses abstraction to hide complexity and focus on what's important
- ML models are extremely **opaque**: individual weights and parameters make no sense to humans

36

8. NEW NOTIONAL MACHINES

- Notional machines: what happens in the runtime environment when a program is executed
- E.g. Java program execution:
  - Named memory locations
  - Control flow
  - Branching and looping
  - etc.

37

8. NEW NOTIONAL MACHINES

- What kinds of notional machines are needed for describing...
  - Passing data through a neural network?
  - Training a network using a training algorithm that adjusts weights to realize a function?
  - Massively parallel systems: thousands of matrix cores?
- The problem is: **We don't know**

38

Picture: Wolfram Programming Lab

39

Picture: playground.tensorflow.org

40

9. TESTING AND DEBUGGING

- "Debugging" ML models ≠ debugging program code
- ML models are **soft**: no discrete results but e.g., the model's confidence in its classification result ("92%")
- ML models are **brittle**: minuscule changes in the environment may render the model useless
- ML models are **opaque**: it's rare that you know exactly *why* the output is what it is
- A *model* isn't even a thing that can be right / wrong

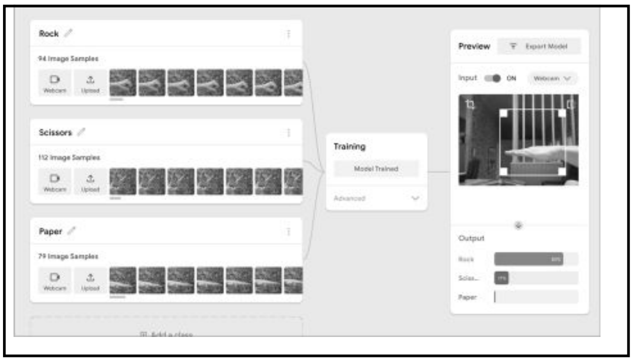
41

9. TESTING AND DEBUGGING

- In traditional programming tinkering and trial-and-error are discouraged
- In ML trial-and-error is typical of searching the optimal hyperparameter and feature space

- Beware AI alchemy!

42



43

44

10. GOODNESS OF SOLUTIONS

- Trust in ML models cannot be based on correctness and verification
- ML solutions are, at best, "probably approximately correct"
  - Their goodness can be statistically determined

45

10. GOODNESS OF SOLUTIONS

- Reductionism is lost
- Emergence dominates

- Complex systems have properties that rise from the interactions of massively many interacting parts
  - Neural networks

46

11. STE(A)M INTEGRATION

- Epistemology of rule-based programming: deductive, positivist
- Epistemology of data-driven computing: inductive, falsificationist
- Empirical research is of the latter type
  - (Of course there are deductive parts!)

47

11. STE(A)M INTEGRATION

- Messing about in science:
  - Data from bicycle sharing in Chicago
  - Language corpora from Dr. Seuss, Taylor Swift
  - ML models of mango sweetness, mango quality, and mango market
- ML-based learning environments offer high degrees of freedom for experiments

48

12. BANISHING MAGIC

- Tenet of technology education: Teach the student how the world around them works
- But how do the following work:
  - TikTok's recommendations
  - Face recognition
  - Speech recognition
  - Translation

49

12. BANISHING MAGIC

- ML isn't magic
- ML systems are not intelligent
- They are cleverly designed technology trained with copious amounts of data

50

13. ETHICAL AND SOCIETAL IMPLICATIONS

- Privacy
- Surveillance
- Tracking
- Job losses
- Misinformation
- Algorithmic bias
- Diversity
- Accountability
- Democracy
- Veracity
- Etc.

51

COMPUTING EDUCATION IN SCHOOL: CONCEPTUAL SHIFTS

52

PROBLEM SOLVING WORKFLOWS

CT 1.0 (RULE-DRIVEN)	CT 2.0 (DATA-DRIVEN)
Formalize the problem	Describe the job and collect data from the intended context
Design an algorithmic solution	Filter and clean the data. Label the data
Implement a solution in a stepwise program	Train a model from the available data
Compile and execute the program	Evaluate and use the model

53

CONCEPTUAL CHANGES IN COMPUTING EDUCATION

CT 1.0	CT 2.0
Correctness can be formally proven	Models may display higher or lower confidence, efficiency
Debugging: Tracking and tracing	Evaluate the model wrt predictions
Deductive problem-solving	Inductive problem-solving
Transparent structure	Black-boxed
Stepwise, deterministic, discrete flow of program through states	Parallel, possibly nondeterministic passing data through a network
Structured data	Unstructured data

54

CONCEPTUAL CHANGES IN COMPUTING EDUCATION

CT 1.0	CT 2.0
Reductionism	Emergence
Formal verification	Statistical measures
Black/glass box testing	Black box testing
No tinkering, toying, trial-and-error	Experimenting with data, parameters, hyperparameters
Prepare for worst-case complexity, optimize for average case	No time/space variance between passes of data through the network
Tedious to ensure portability	Straightforwardly portable

55

CHALLENGES

- Brittleness, softness
- Opaqueness
  - Shallow, superficial learning
- AI alchemy
  - Very advanced mathematics
- School systems already struggling with CT 1.0
- Emerging topic: unrealistic expectations, misconceptions

56

THANK YOU!
QUESTIONS, COMMENTS?

Team involved in this research:
Matti Tedre, Henriikka Vartiainen, Ilkka Jormanainen, Juho Kahila, Teemu Valtonen, Tapani Toivonen, Arnold Pears

October 27, 2021: Paderborn Colloquium on Artificial Intelligence and Data Science Education at School Level

57