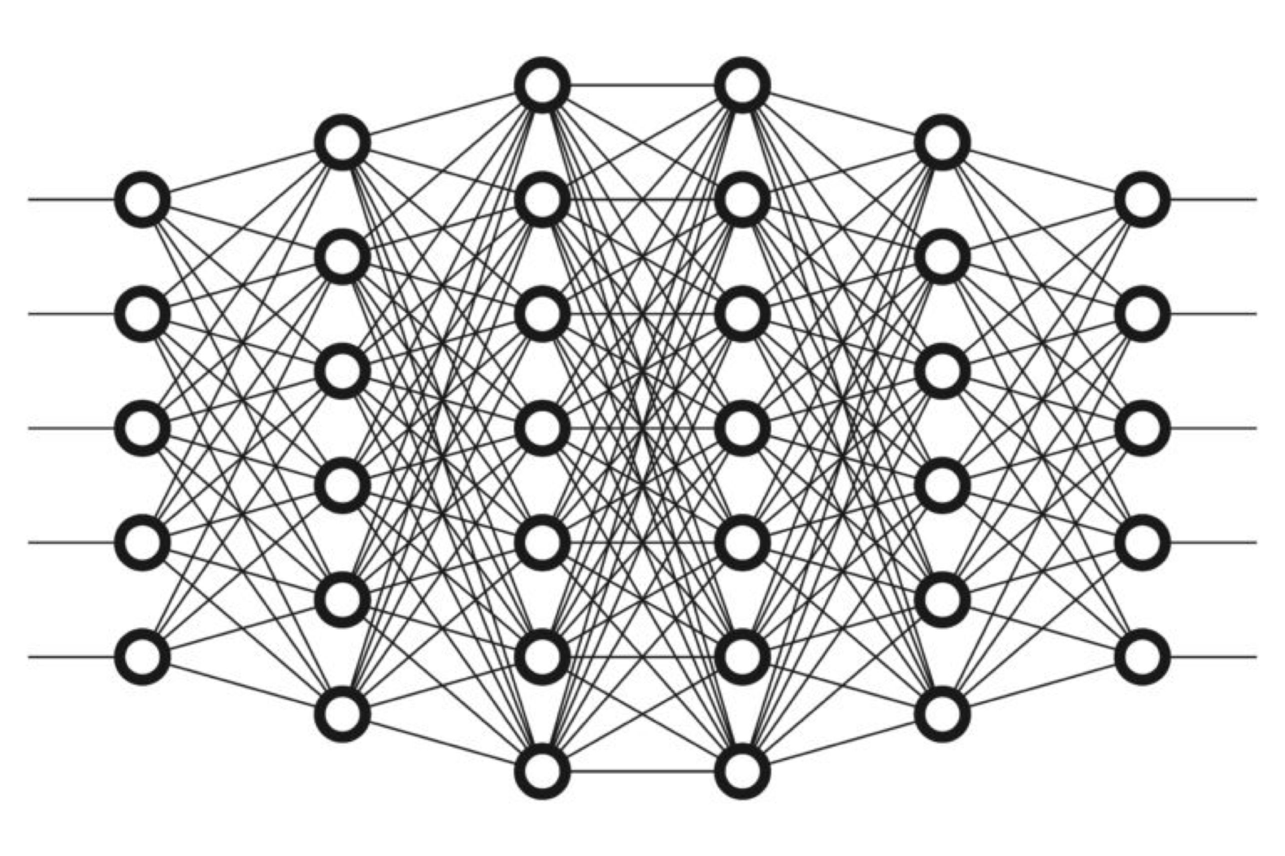
Introduction to the Tsetlin Machine

Ole-Christoffer Granmo

April 1, 2019

The complexity of neural networks

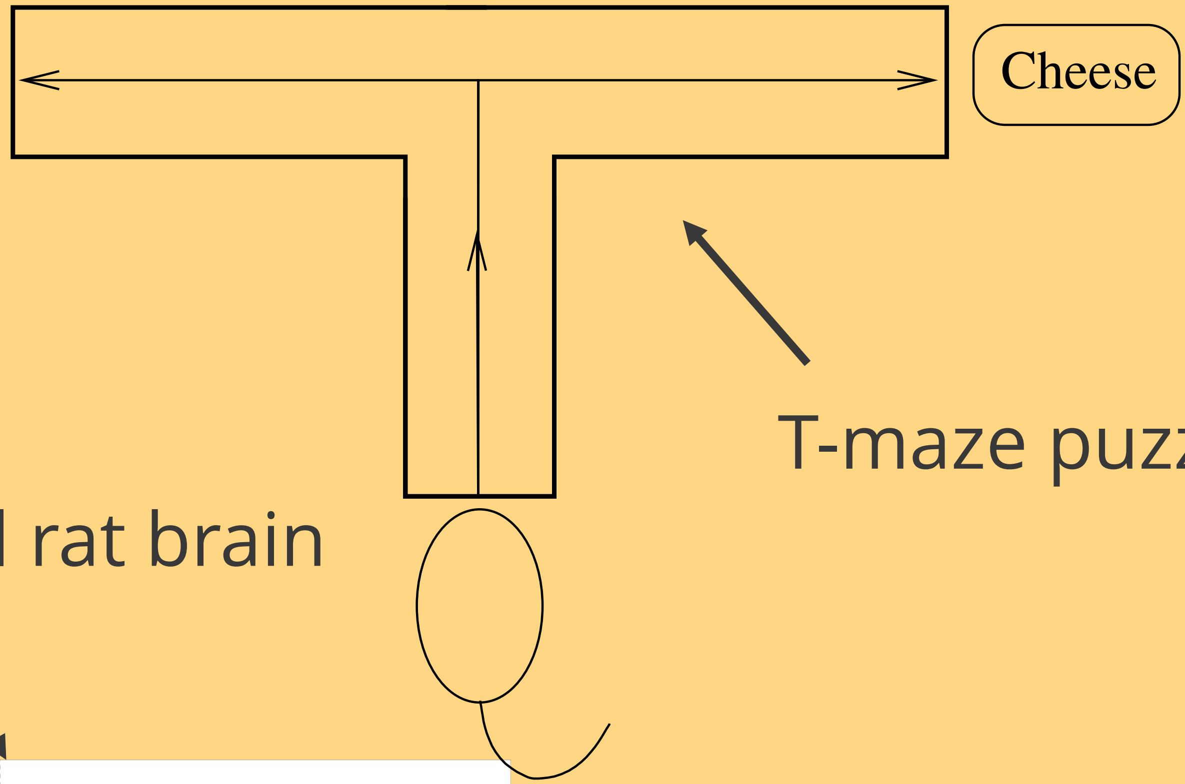


011011001110000111

Michael Lvovitch Tsetlin

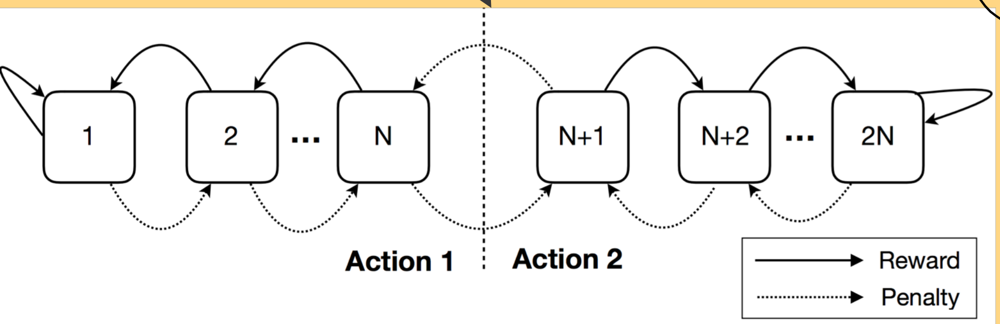
Michael Lvovitch Tsetlin (the surname is also written Cetlin, Tzetlin, Zeitlin, Zetlin; cyrillic: Михаил Львович Цетлин) (22 September 1924 – 30 May 1966) was a Russian mathematician and physicist who worked on cybernetics.

The Tsetlin automaton:
A mathematical rat brain

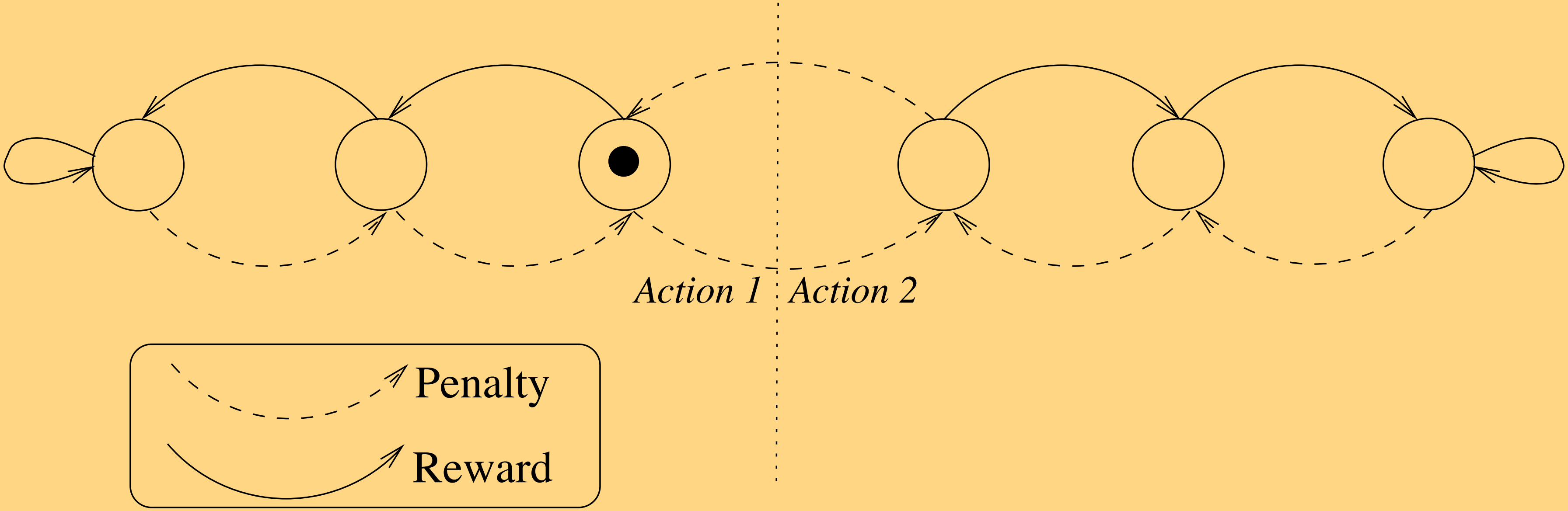


T-maze puzzle

Mathematical rat brain



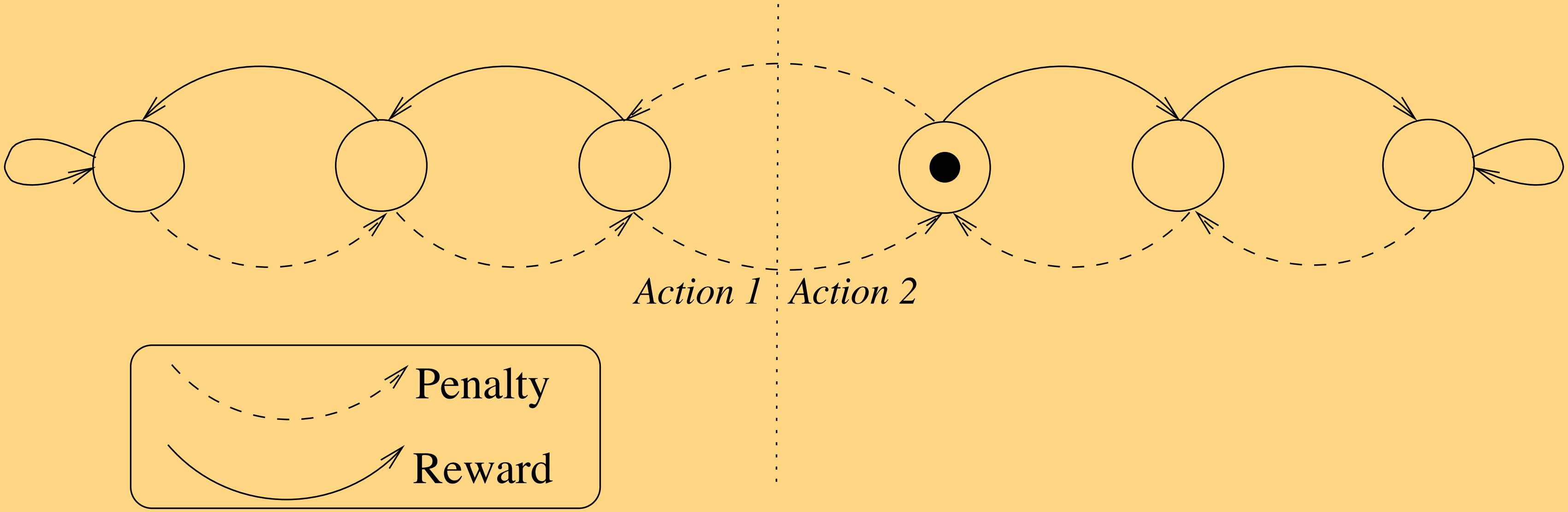
The Tsetlin automaton



Selected Action: Action 1

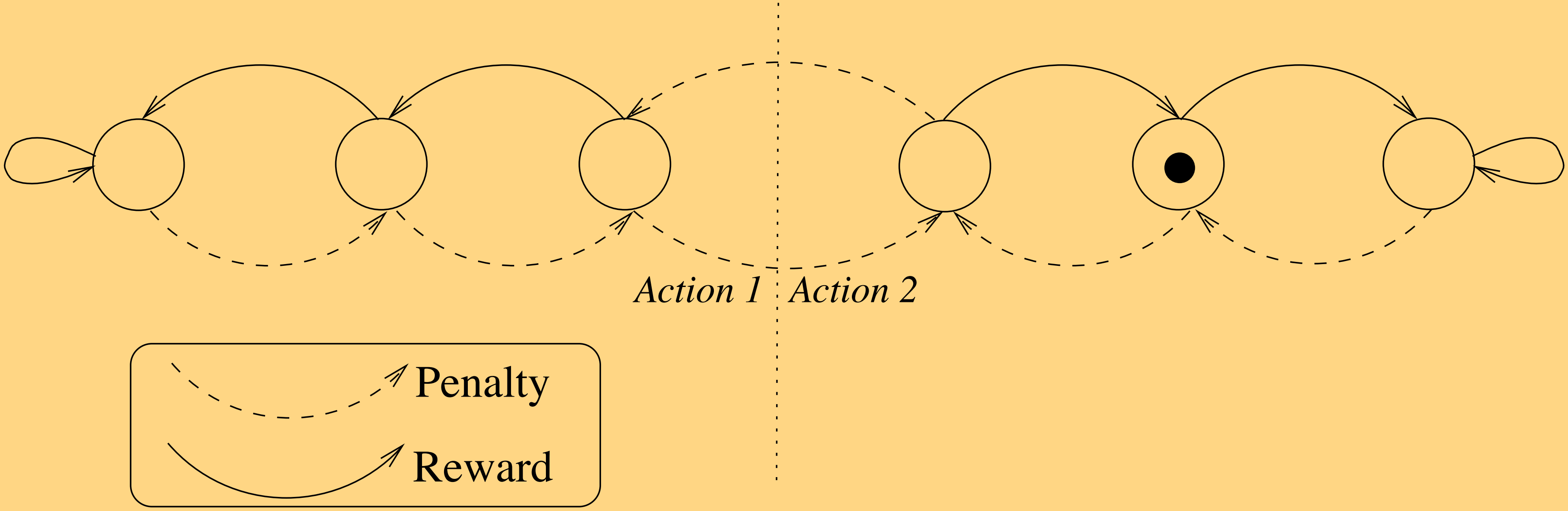
Response from Environment: Penalty

The Tsetlin automaton



Selected Action: Action 2
Response from Environment: Reward

The Tsetlin automaton



Selected Action: Action 2

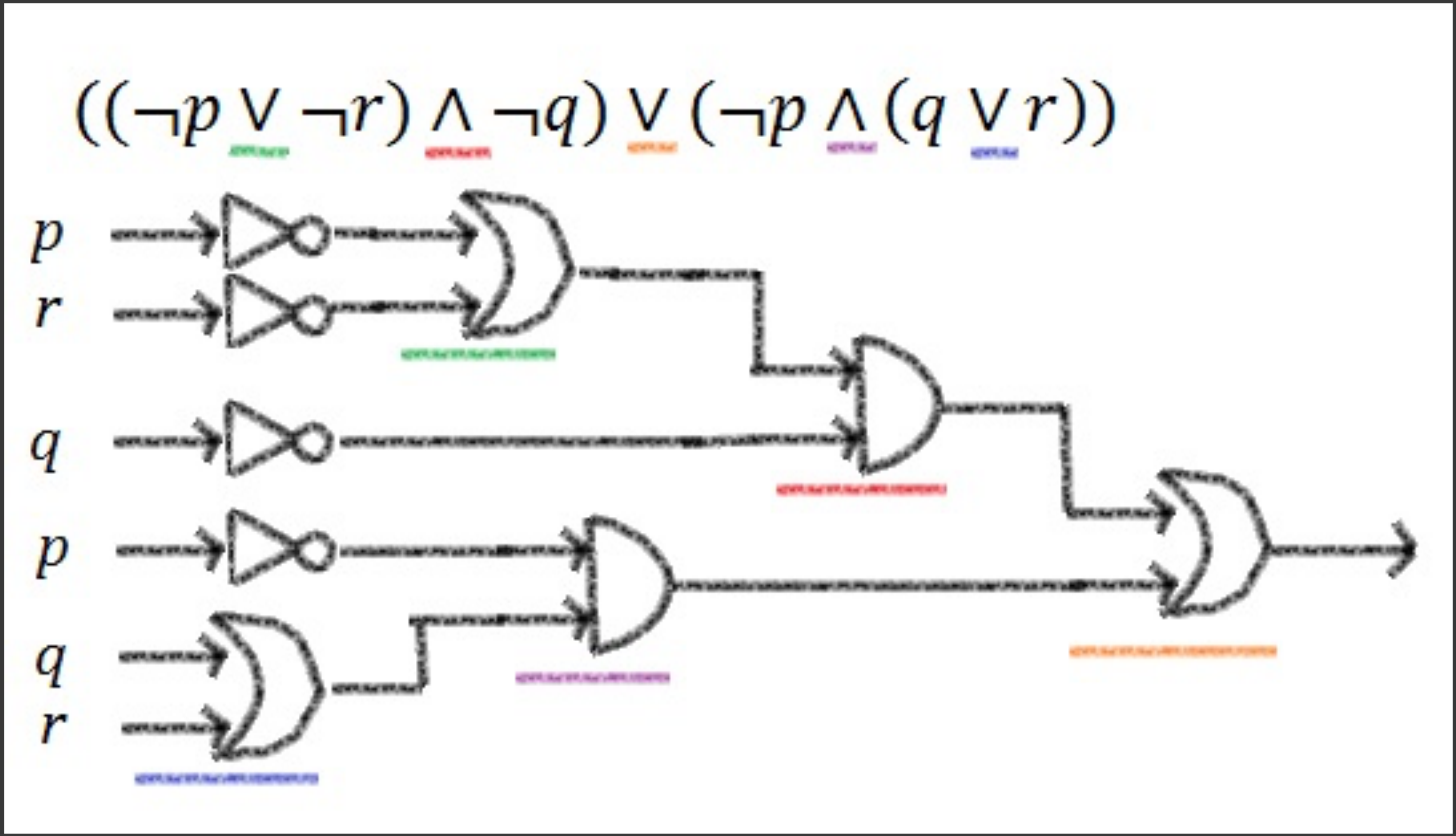
Response from Environment: Reward

Propositional logic

Propositional logic is the branch of logic that studies ways of joining and/or modifying entire propositions, statements or sentences to form more complicated propositions, statements or sentences, as well as the logical relationships and properties that are derived from these methods of combining or altering statements.

The Internet Encyclopedia of Philosophy

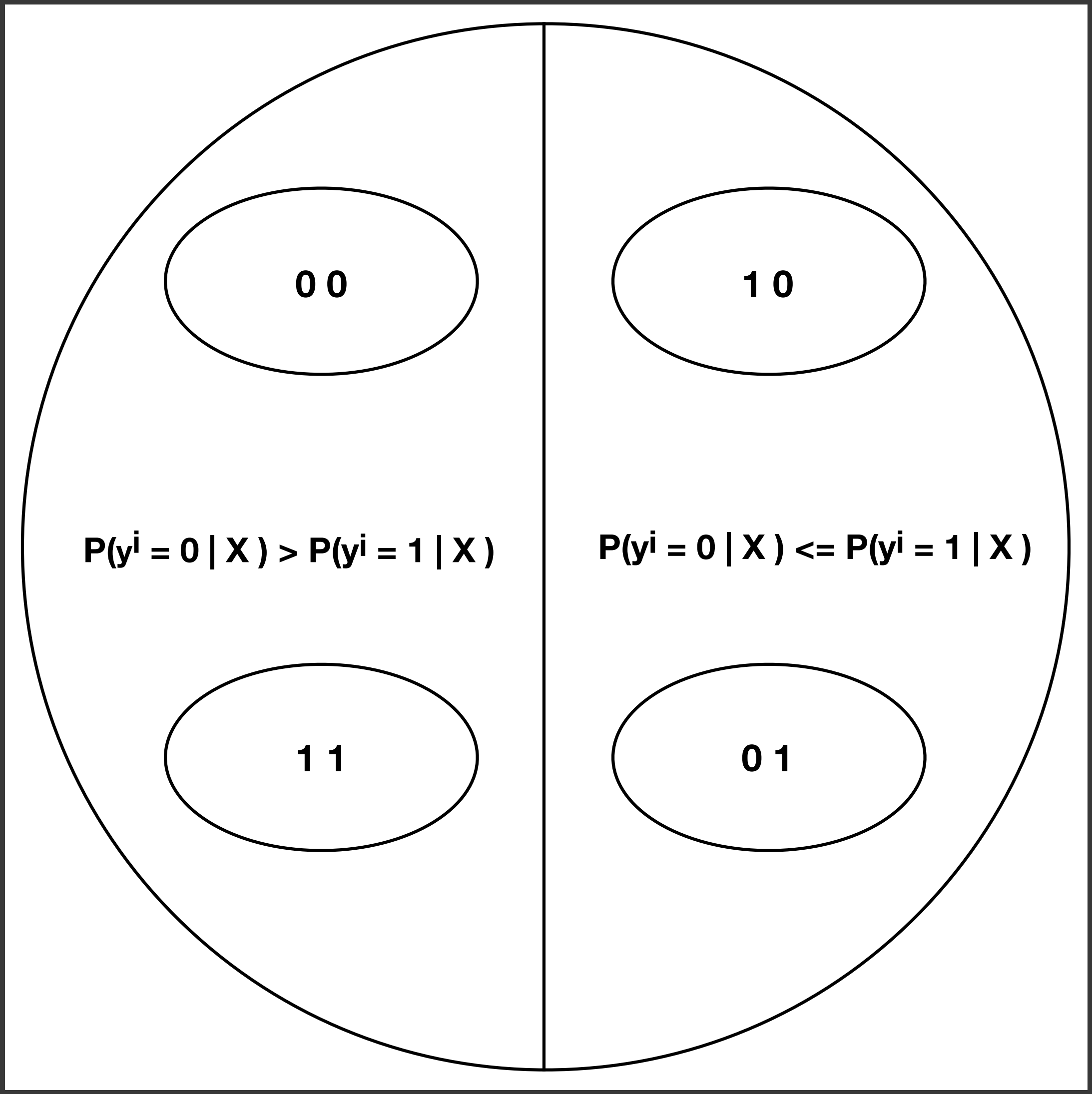
Propositional logic



http://mathrescue.blogspot.no/2013/04/

Tsetlin Machine =
thousands of rat brains +
propositional logic

The pattern recognition problem

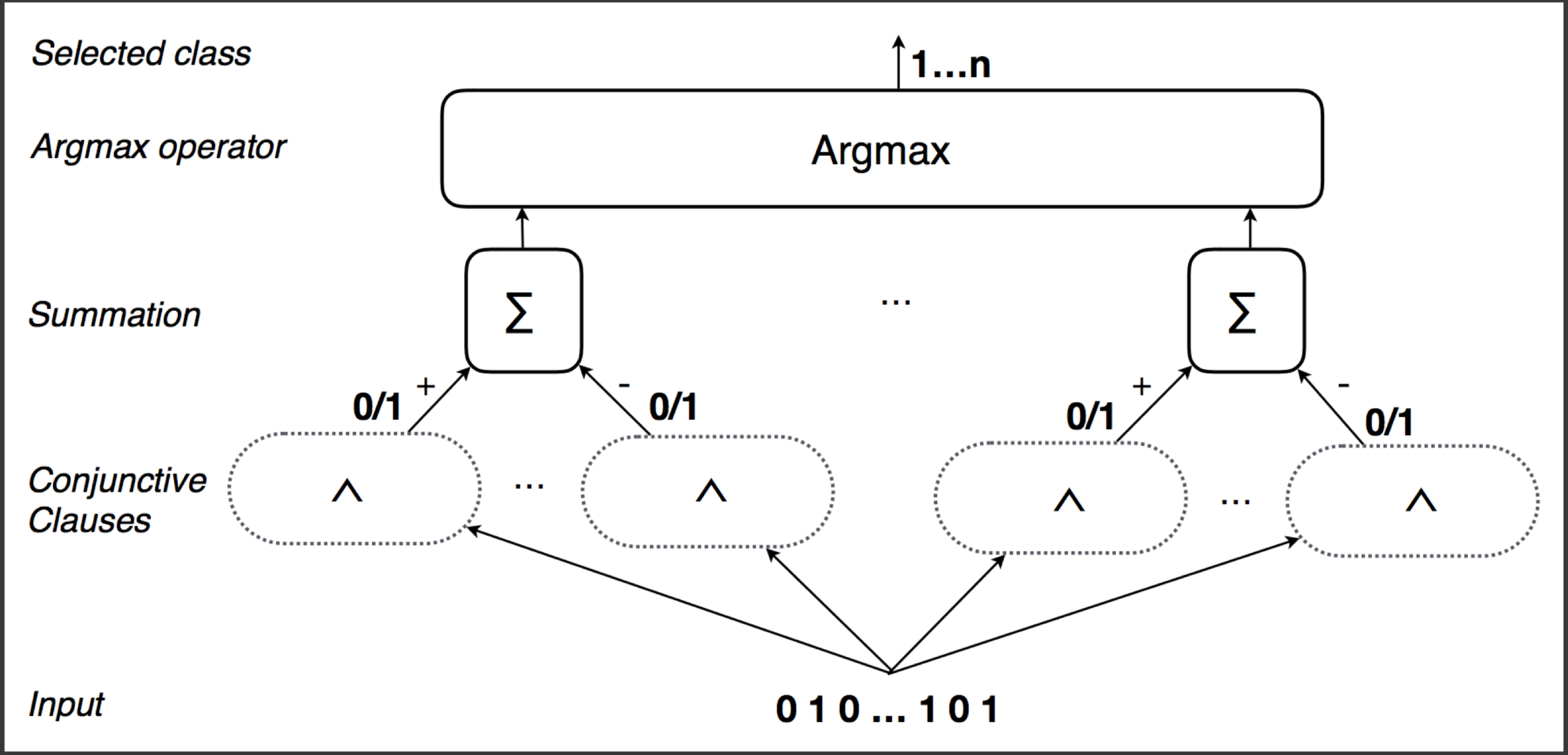


$X = [x_1, x_2]$ f(X) $= (x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2)$

The pattern recognition problem

Input	Target
0 1 0 1 1 0 1 1 0 1 1 0	1
1 1 1 0 1 0 1 1 0 0 1 1	0
0 0 1 1 0 1 1 1 1 0 1 0	0
1 0 1 0 1 0 1 1 1 0 1 1	0
1 1 1 0 1 1 1 0 1 1 0 0	1
1 0 0 0 0 0 0 1 0 1 1 0	1
1 1 1 0 1 1 1 0 0 0 1 0	0
1 0 1 0 1 0 1 1 1 0 1 1	1
0 0 0 0 1 1 0 1 1 1 0 0	0
1 0 0 0 1 0 1 1 0 0 1 0	1

The Tsetlin Machine



https://github.com/cair/TsetlinMachine

Tsetlin Machine **learning procedure**

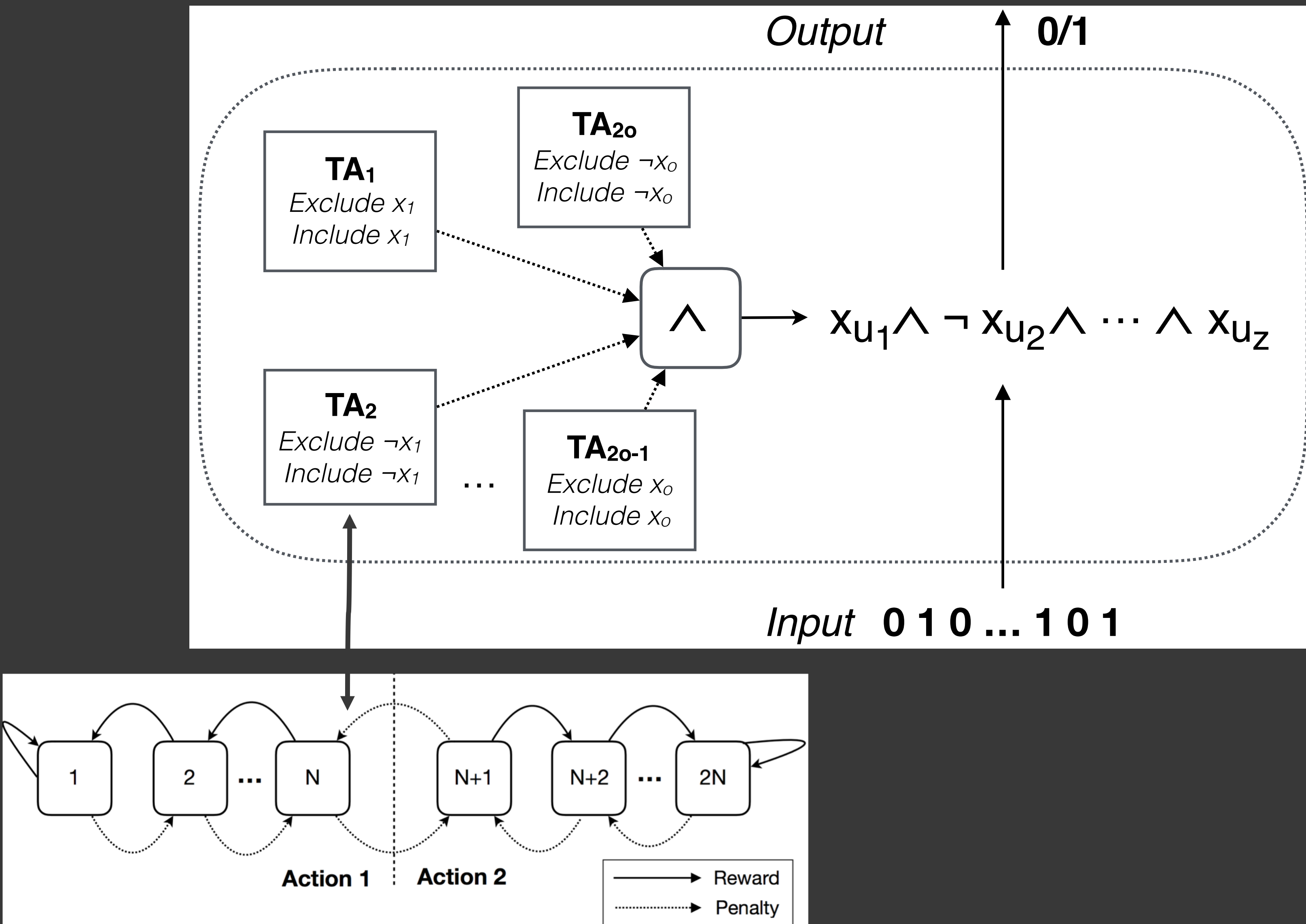
It's astounding that those things work at all... it's like "pour Scrabble letters into box... shake vigorously... open and read the answer!"

claytonkb, Reddit, February, 2019

The Tsetlin Machine game

1. The arrival of a labelled object $(\widehat{\mathrm{X}}, \hat{\mathrm{y}})$ signals the start of a new game round.
2. Each Tsetlin Automaton decides whether to *include* or *exclude* its designated literal, leading to a new configuration of clauses **<u>C</u>**
3. Each clause, $\mathrm{C_j} \in$ **<u>C</u>**, is then evaluated with $\widehat{\mathrm{X}}$ as input.
4. The final output, y, of the Tsetlin Machine is decided and compared with the target output $\hat{\mathrm{y}}$.
5. Each Tsetlin Automaton is independently and randomly given either Reward, Inaction, Penalty feedback, based on a novel game matrix

The Tsetlin Machine: Clause formation as a **game**



CAIR
CENTRE FOR ARTIFICIAL INTELLIGENCE RESEARCH

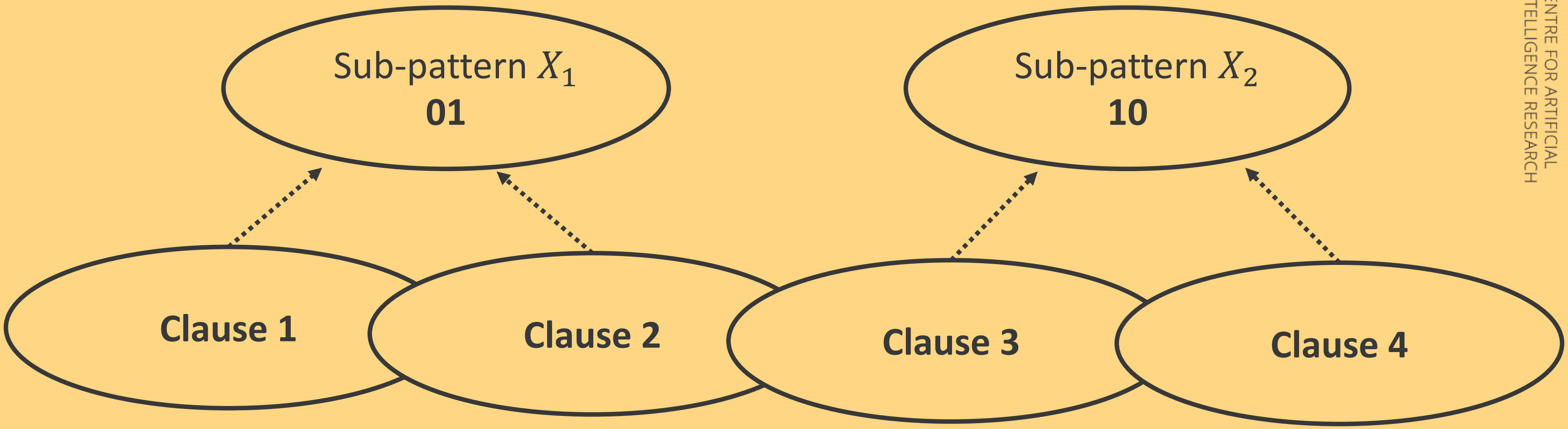
Type I feedback (true positive & false negative)

Document→		Target Clause evaluates to / Target Literal evaluates to	1		0	
			1	0	1	0
Include literal		P (Reward)	$\frac{s-1}{s}$	NA	0	0
		P (Inaction)	$\frac{1}{s}$	NA	$\frac{s-1}{s}$	$\frac{s-1}{s}$
		P (Penalty)	0	NA	$\frac{1}{s}$	$\frac{1}{s}$
Exclude literal		P (Reward)	0	$\frac{1}{s}$	$\frac{1}{s}$	$\frac{1}{s}$
		P (Inaction)	$\frac{1}{s}$	$\frac{s-1}{s}$	$\frac{s-1}{s}$	$\frac{s-1}{s}$
		P (Penalty)	$\frac{s-1}{s}$	0	0	0

Type II feedback (false positive & true negative)

Document→		Target Clause evaluates to / Target Literal evaluates to	1		0	
			1	0	1	0
Include literal		P (Reward)	0	NA	0	0
		P (Inaction)	1.0	NA	1.0	1.0
		P (Penalty)	0	NA	0	0
Exclude literal		P (Reward)	0	0	0	0
		P (Inaction)	1.0	0	1.0	1.0
		P (Penalty)	0	1.0	0	0

Allocation of sparse pattern representation resources



To effectively utilize sparse pattern representation capacity (a constrained number of clauses), we use a threshold value *T* as target for the summation $f_{\Sigma}(X)$. That is, the probability of activating *Type I Feedback* is:

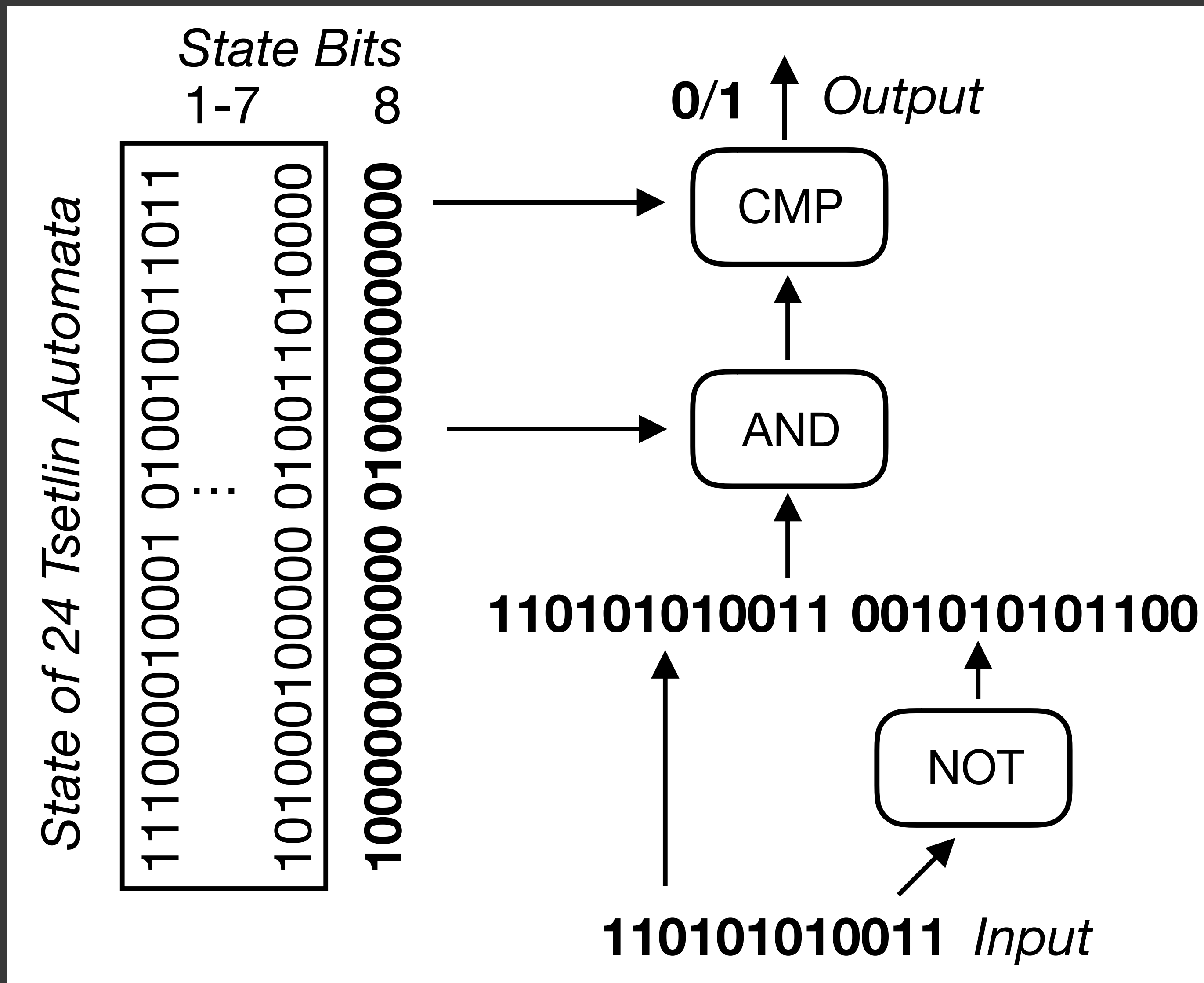
$$(T - \max(-T, \min(T, f_{\Sigma}(X)))/2T$$

while *Type II Feedback* is activated with probability:

$$(T + \max(-T, \min(T, f_{\Sigma}(X)))/2T$$

If the votes accumulate to a total of +/- T or more, neither rewards or penalties are handed out to the involved Tsetlin Automata.

Bit representation of clause



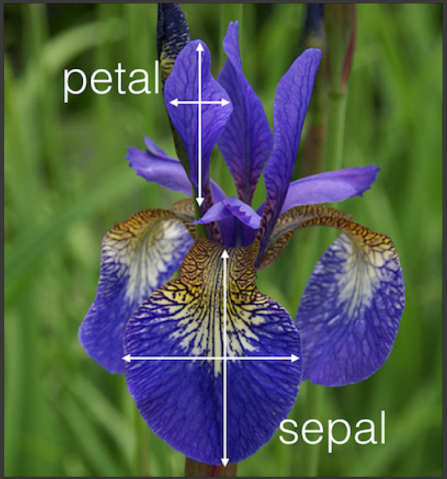
Bit representation of clause

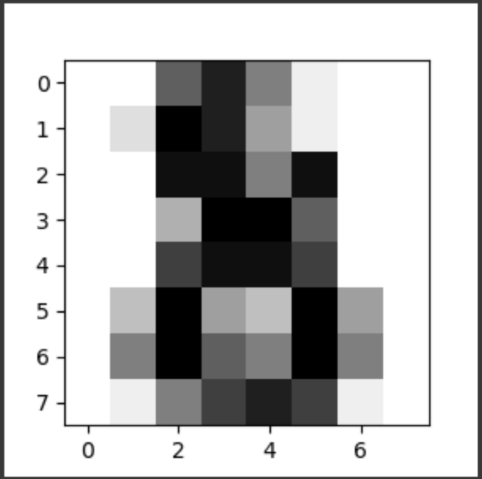
```
((*tm).ta_state[j][k][STATE_BITS-1] & Xi[k]) == (*tm).ta_state[j][k][STATE_BITS-1];
```

https://github.com/cair/fast-tsetlin-machine-with-mnist-demo

Applications and empirical results

Does it work?



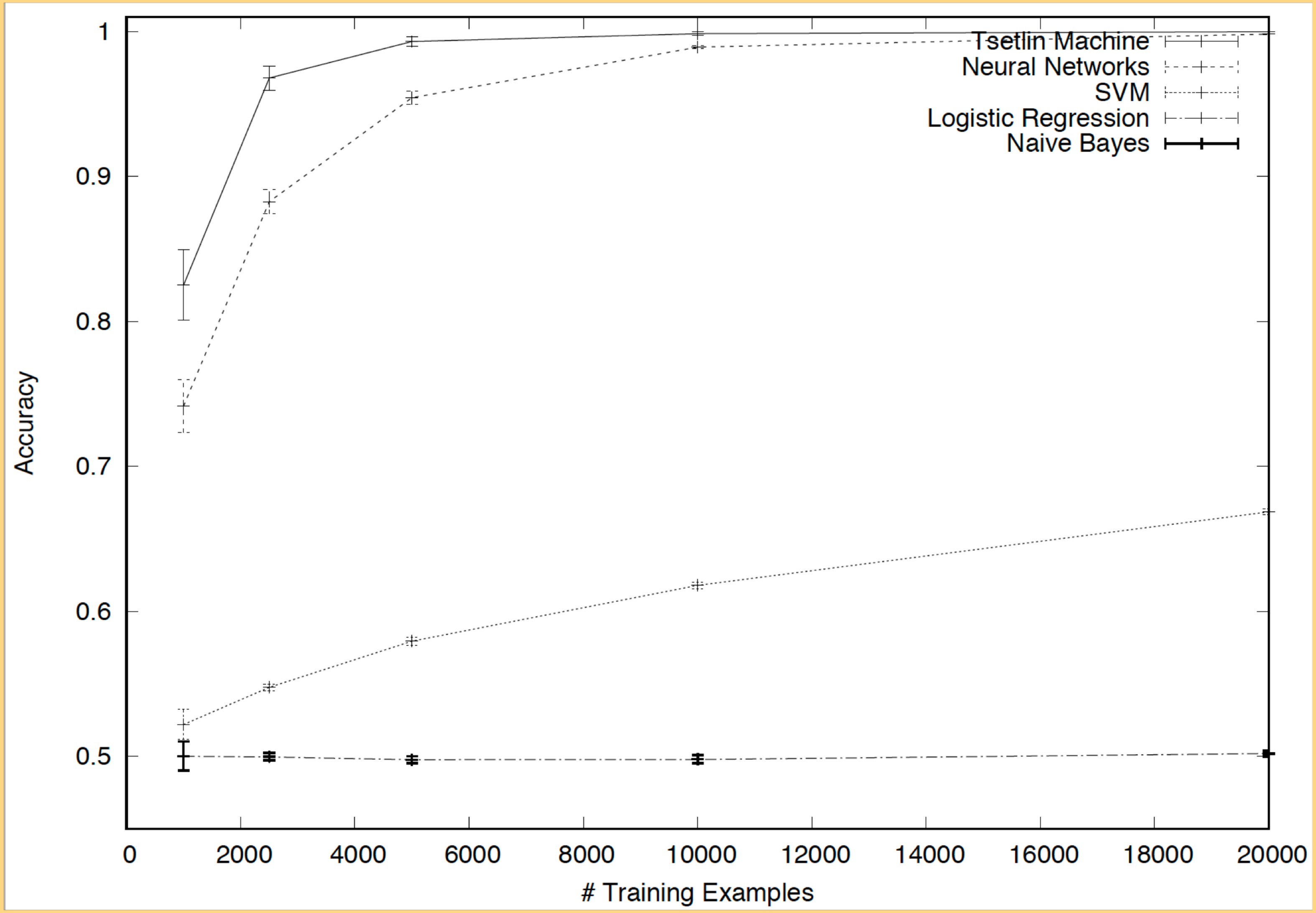


Input	Target
010110110110	1
111010110011	0
001101111010	0
101010111011	0
111011101100	1
100000010110	1
111011100010	0
101010111011	1
000011011100	0
100010110010	1

Noisy XOR dataset

Input	Target
0 1 0 1 1 0 1 1 0 1 1 0	1
1 1 1 0 1 0 1 1 0 0 1 1	0
0 0 1 1 0 1 1 1 1 0 1 0	0
1 0 1 0 1 0 1 1 1 0 1 1	0
1 1 1 0 1 1 1 0 1 1 0 0	1
1 0 0 0 0 0 0 1 0 1 1 0	1
1 1 1 0 1 1 1 0 0 0 1 0	0
1 0 1 0 1 0 1 1 1 0 1 1	1
0 0 0 0 1 1 0 1 1 1 0 0	0
1 0 0 0 1 0 1 1 0 0 1 0	1

Noisy XOR dataset

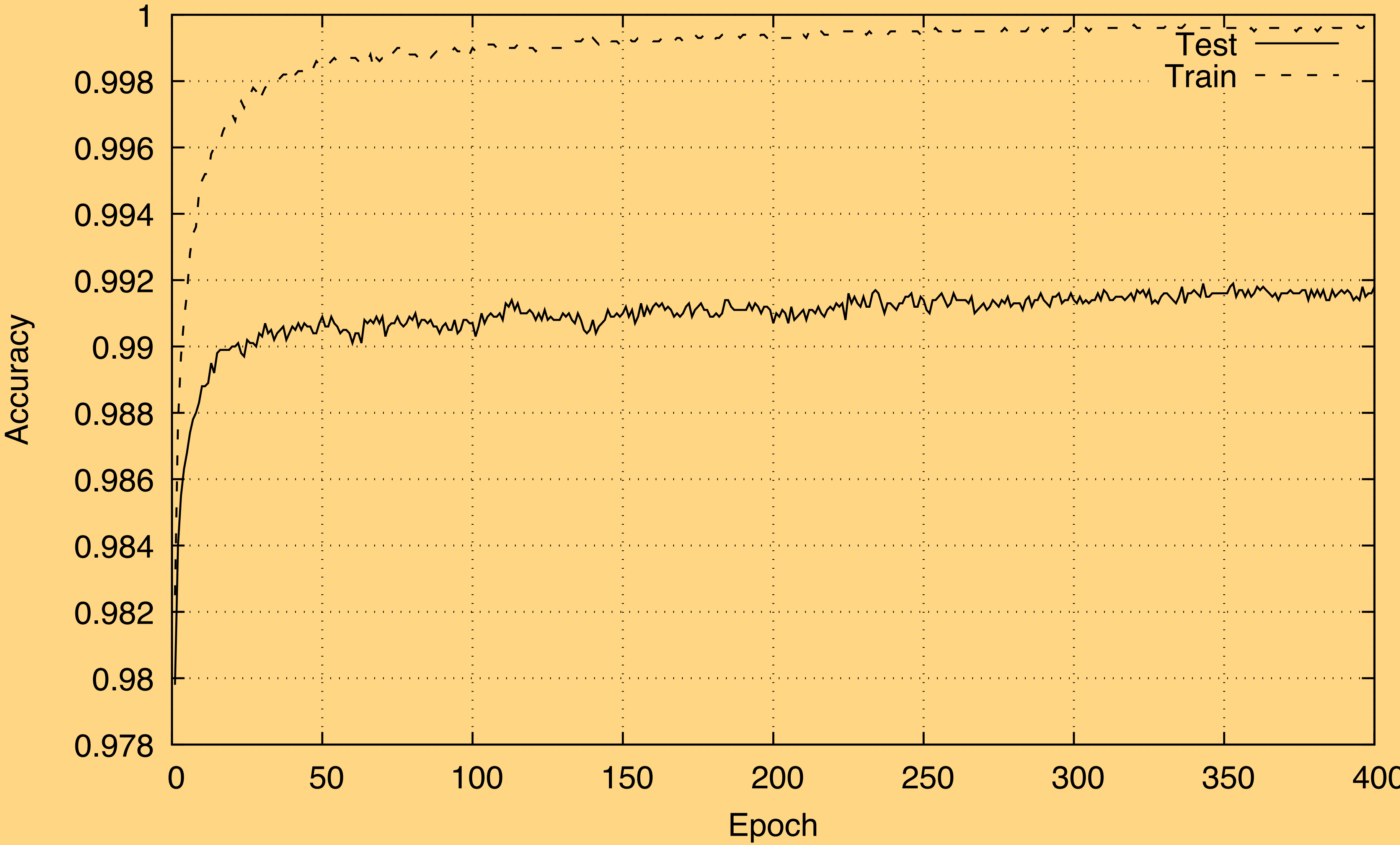


CAIR
CENTRE FOR ARTIFICIAL INTELLIGENCE RESEARCH

Using the Tsetlin Machine to classify handwritten digits

```
............................
............................
............................
............................
............................
............................
............@@..............
...........@@.....@@@.......
...........@@.....@@........
..........@@......@@........
.........@@@......@@........
.........@@.......@@........
.........@@......@@@........
........@@@.@@@..@@@........
........@@@@@@@@@@@@........
.......@@@@@@@@@@@@.........
.......@@........@@.........
.................@@.........
................@@@.........
................@@@.........
................@@@.........
................@@@.........
................@@@.........
................@@@.........
................@@@.........
................@@@.........
............................
............................
```

Learning behaviour



CAIR CENTRE FOR ARTIFICIAL INTELLIGENCE RESEARCH

Results on raw, unenhanced, and unextended MNIST data

Technique	Accuracy (%)
Convolutional Tsetlin Machine	99.2 ± 0.0
2-layer NN, 800 HU, Cross-Entropy Loss	*98.6*
Tsetlin Machine	98.2 ± 0.0
K-nearest-neighbors, L3	*97.2*
3-layer NN, 500+150 hidden units	*97.1*
40 PCA + quadratic classifier	*96.7*
1000 RBF + linear classifier	*96.4*
Logistic regression	91.5
Linear classifier (1-layer NN)	*88.0*
Decision tree	87.8
Multinomial Naive Bayes	83.2

CAIR CENTRE FOR ARTIFICIAL INTELLIGENCE RESEARCH

Tsetlin Machine based text classification

CAIR
CENTRE FOR ARTIFICIAL INTELLIGENCE RESEARCH

Documents

1. He is allergic to Penicillin..

2. She reacts to Penicillin..

Terms	Document 1	Document 2
He	1	0
is	1	0
allergic	1	0
to	1	1
Penicillin	1	1
She	0	1
Reacts	0	1
..	..	..

Bit vectors

Document 1: 1111100..

Document 2: 0001111..

If not “don’t” **and** “react” **and** “Voltaren” **then** Allergy

https://github.com/cair/TextUnderstandingTsetlinMachine

Text understanding with human interpretable rules

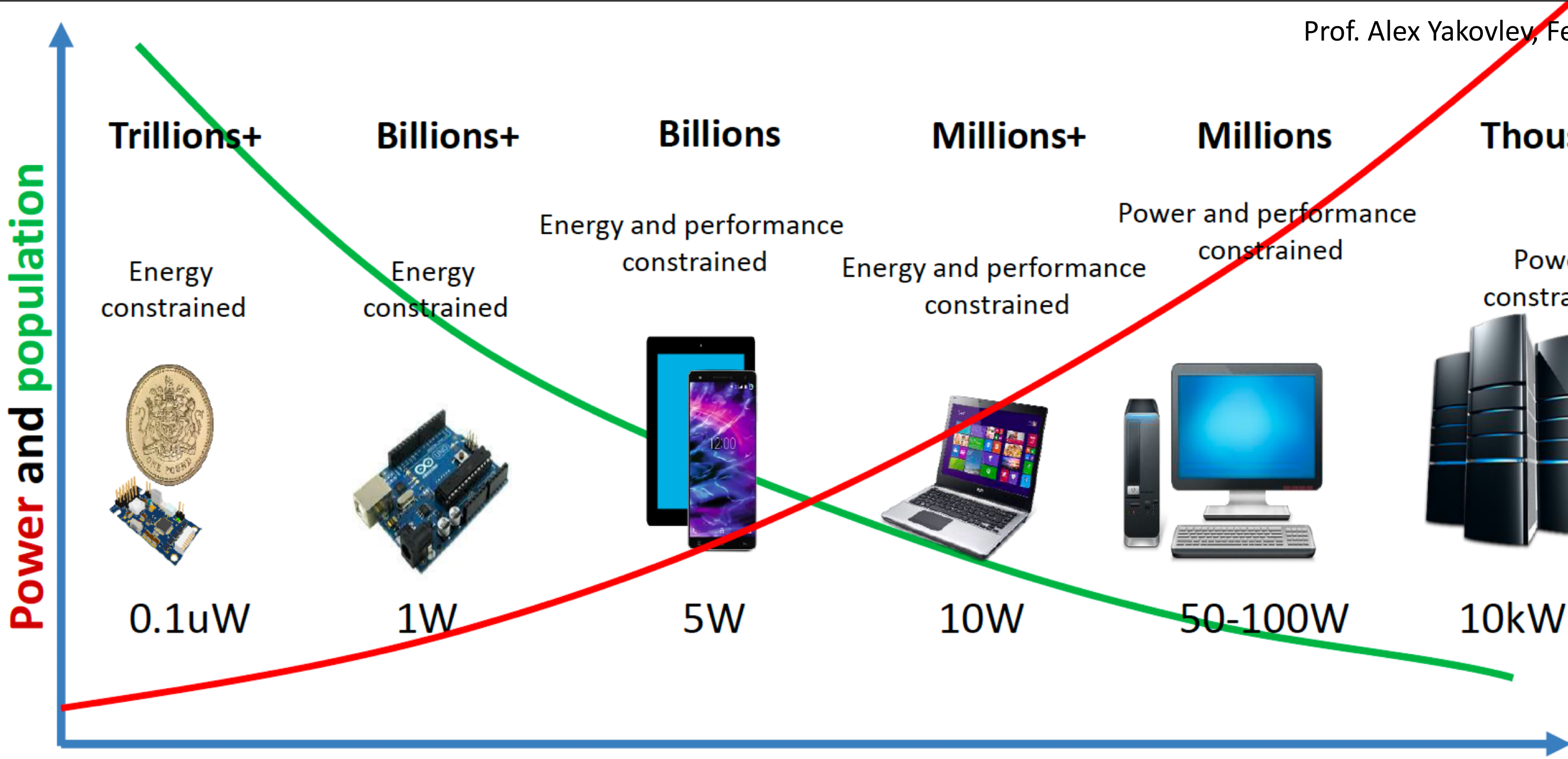
Method	Precision	Recall	F-measure
Multinomial Naïve Bayes	85.9±0.0	86.1±0.0	86.0±0.0
Logistic regression	86.5±0.0	87.6±0.0	87.1±0.0
Decision tree	71.1±0.0	68.4±0.0	69.7±0.0
Random forest	78.9±0.1	78.1±0.1	78.5±0.1
kNN	58.4±0.0	63.5±0.0	60.8±0.0
Linear SVM	88.0±0.0	89.1±0.0	88.5±0.0
MLP	82.6±0.1	82.6±0.1	82.6±0.1
LSTM	87.2±0.7	84.3±0.9	85.6±0.6
LSTM CNN	89.5±0.2	86.8±0.4	88.1±0.1
Bi-LSTM	87.6±0.8	83.9±1.1	85.5±0.6
Bi-LSTM CNN	88.3±0.2	87.5±0.5	87.9±0.2
Tsetlin Machine	89.7±0.0	89.7±0.0	89.7±0.0

if “rash” **and** “reaction” **and** “penicillin” **then** Allergy

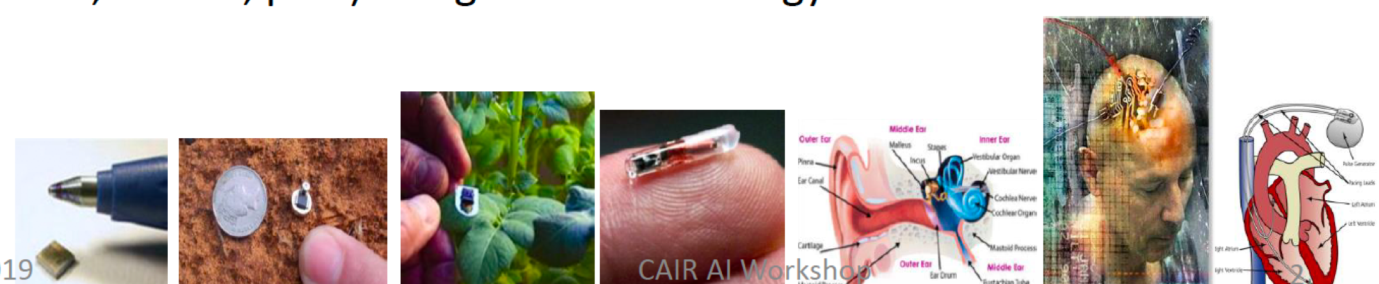
https://github.com/cair/TextUnderstandingTsetlinMachine

CAIR CENTRE FOR ARTIFICIAL INTELLIGENCE RESEARCH

Tsetlin Machines in hardware



Trillions of ubiquitous systems (sensors, probes, monitors, actuators, controllers) are being deployed to operate in myriad of places (organisation, human, body part, household, offices, pets) using harvested energy or micro-batteries



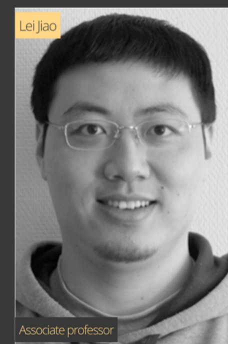
Tsetlin Machine – current activities

CAIR

Recurrent Tsetlin Machine

Convolutional Tsetlin Machine

Hydropower optimization



Theoretical analysis

Natural language understanding

EGG analysis

Medical applications

Cause and effect

House of CAIR

Adaptive and scalable hardware, Newcastle University

Reinforcement learning, University of Groningen

Hardware, Temporal Computing, UK