

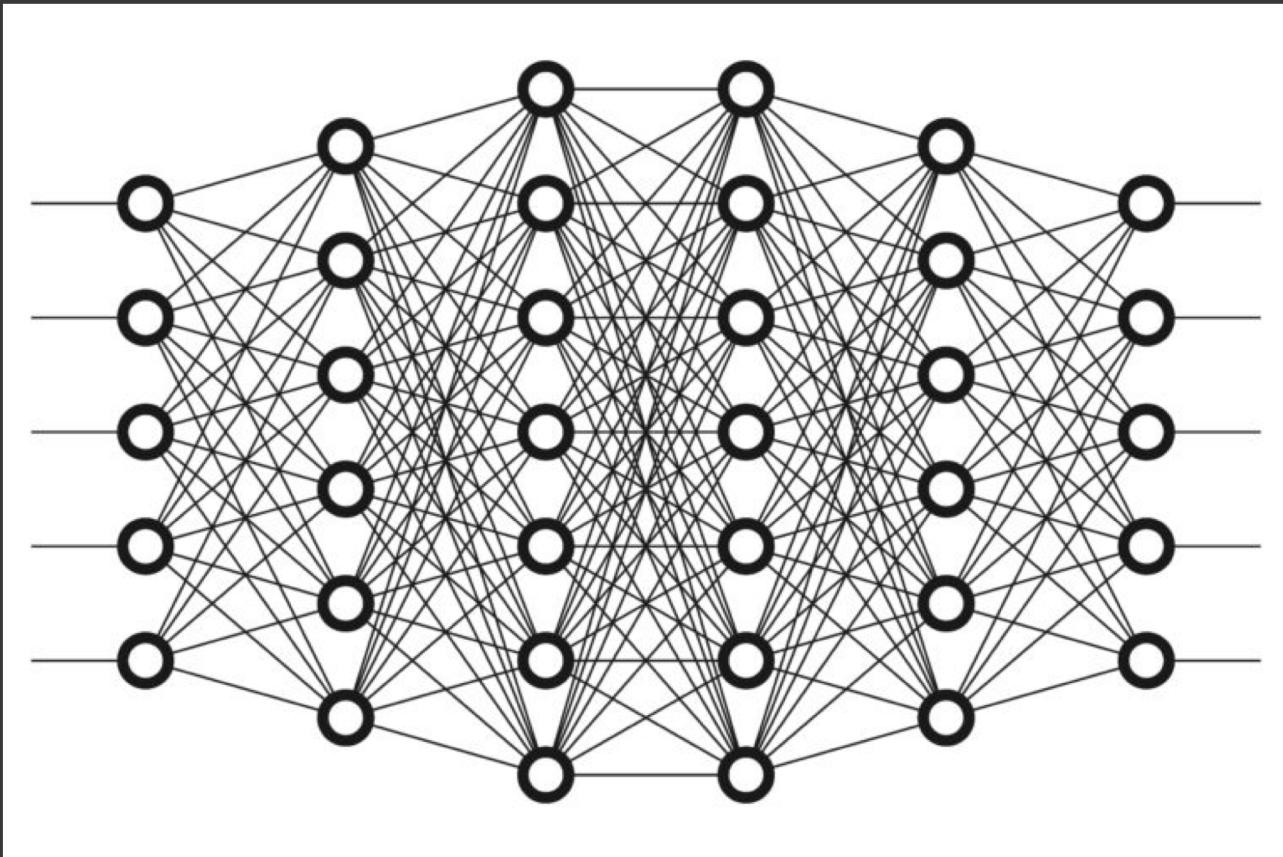


Slides for the tutorial on the **Convolutional Tsetlin Machine**

<https://github.com/cair/convolutional-tsetlin-machine>

Ole-Christoffer Granmo
June 5, 2019

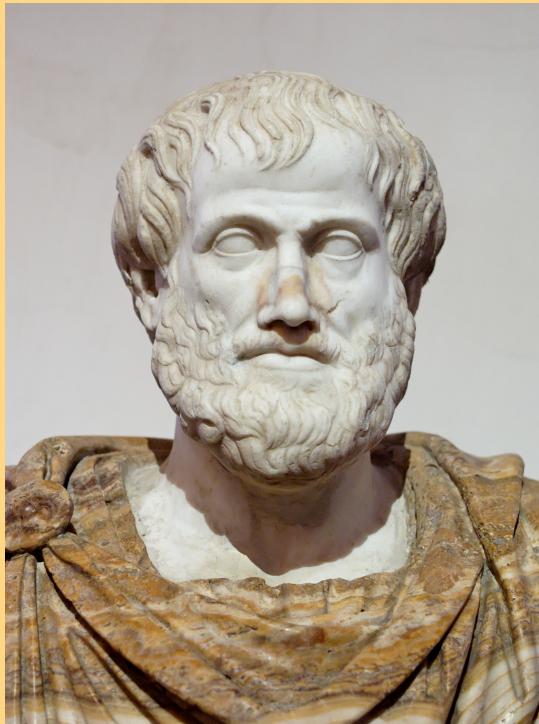
The complexity of neural networks



011011001110000111

Why Tsetlin Machines?

Interpretable



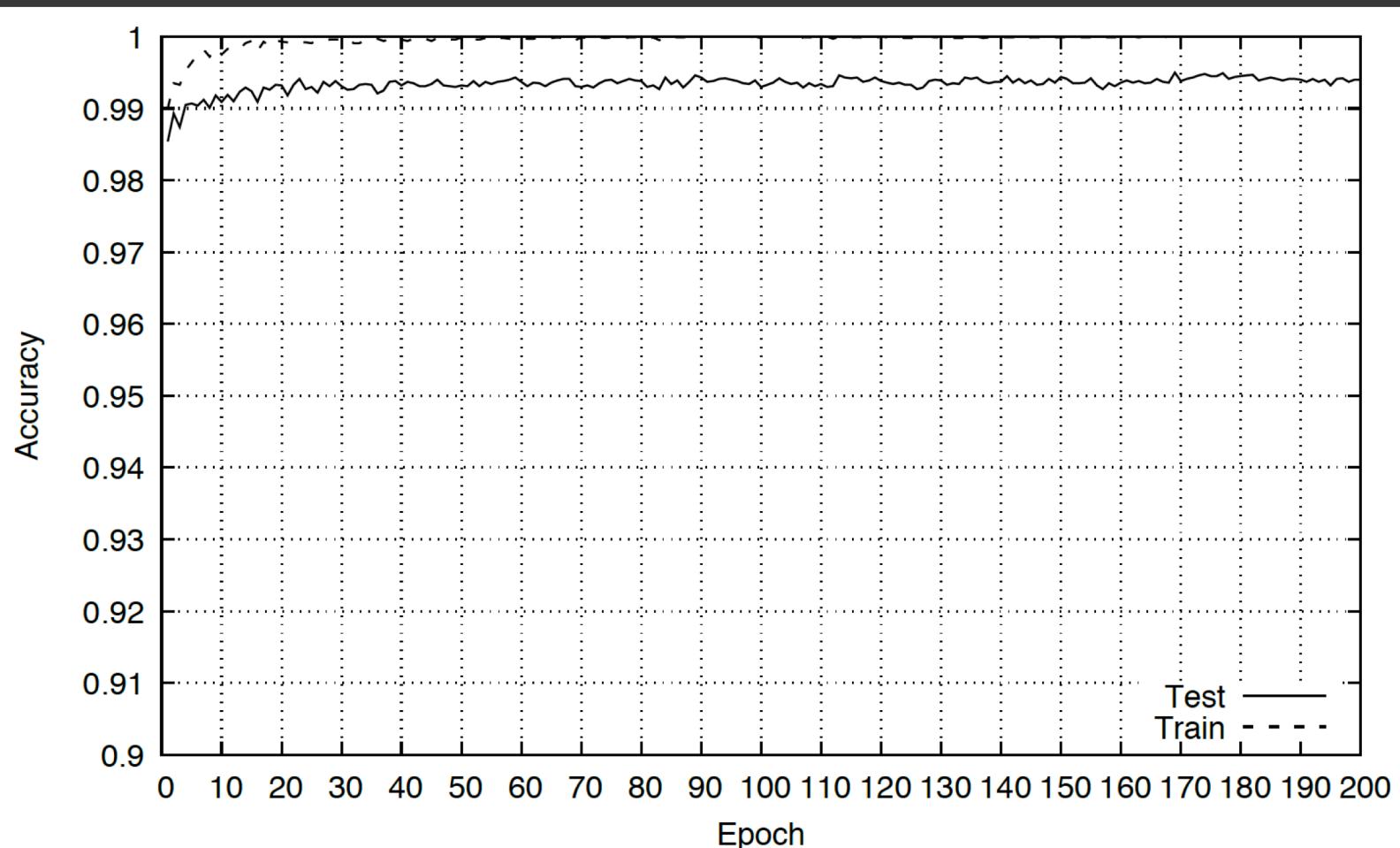
Propositional logic is the branch of logic that studies ways of joining and/or modifying entire propositions, statements or sentences to form more complicated propositions, statements or sentences, as well as the logical relationships and properties that are derived from these methods of combining or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

The Internet Encyclopedia of Philosophy

Interpretable

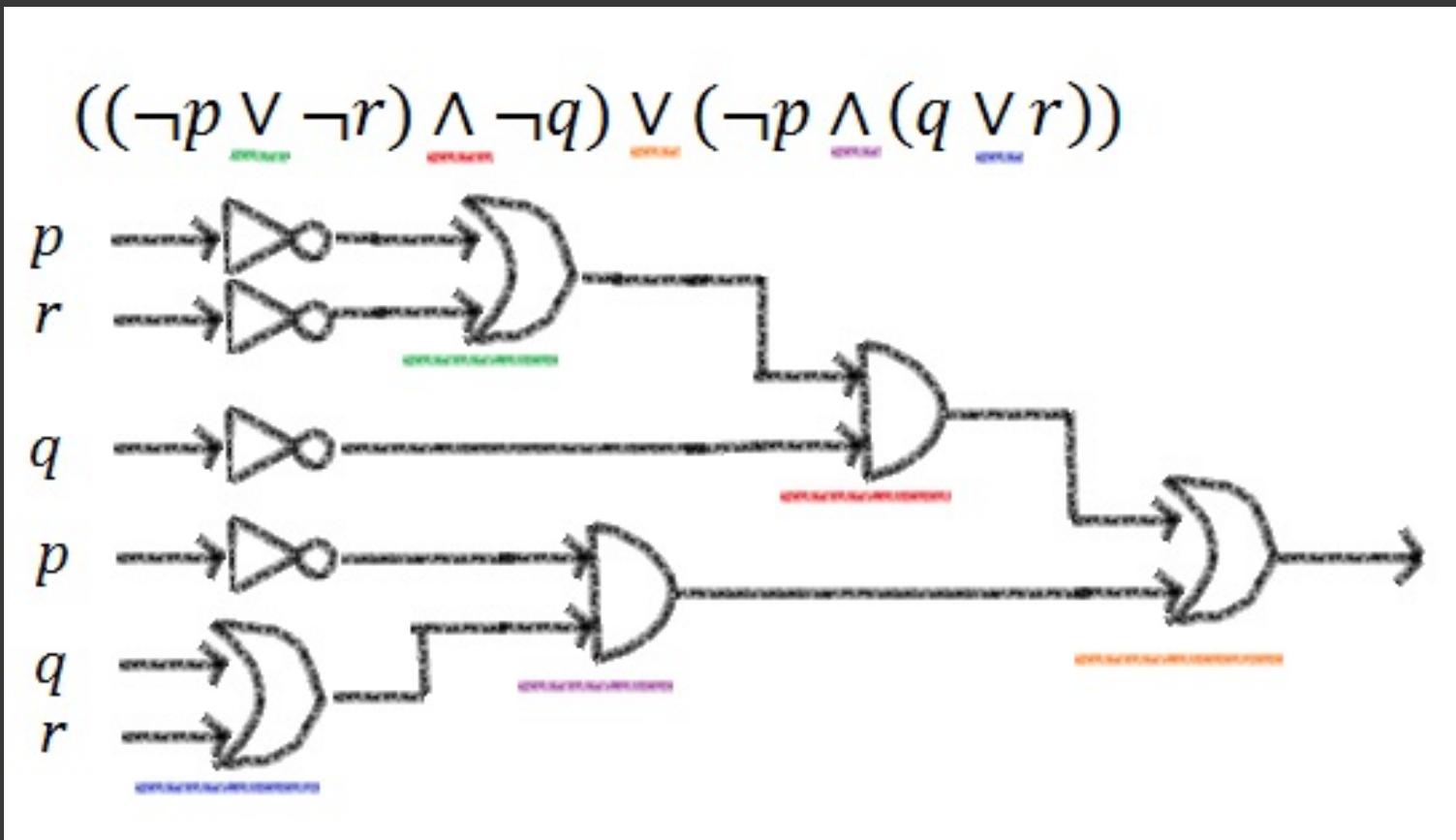
**IF "grass" AND "asthmatic" AND "short
of breath" AND "itchy skin" AND
"symptom picture" AND "Cetirizin" AND
"hyperventilating" AND "causes" AND
"episodes" THEN Allergy**

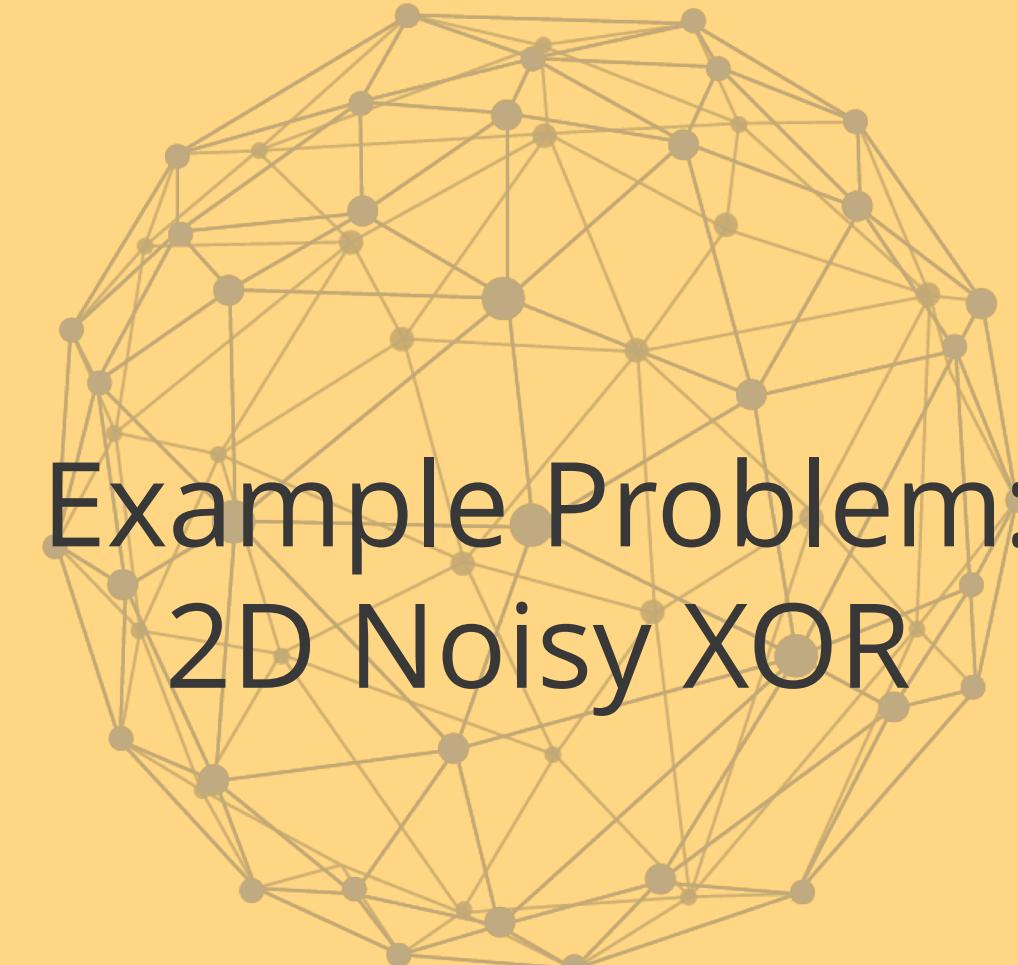
Competitive accuracy



Training and test accuracy for the Convolutional Tsetlin Machine on MNIST, epoch-by-epoch, in a single run

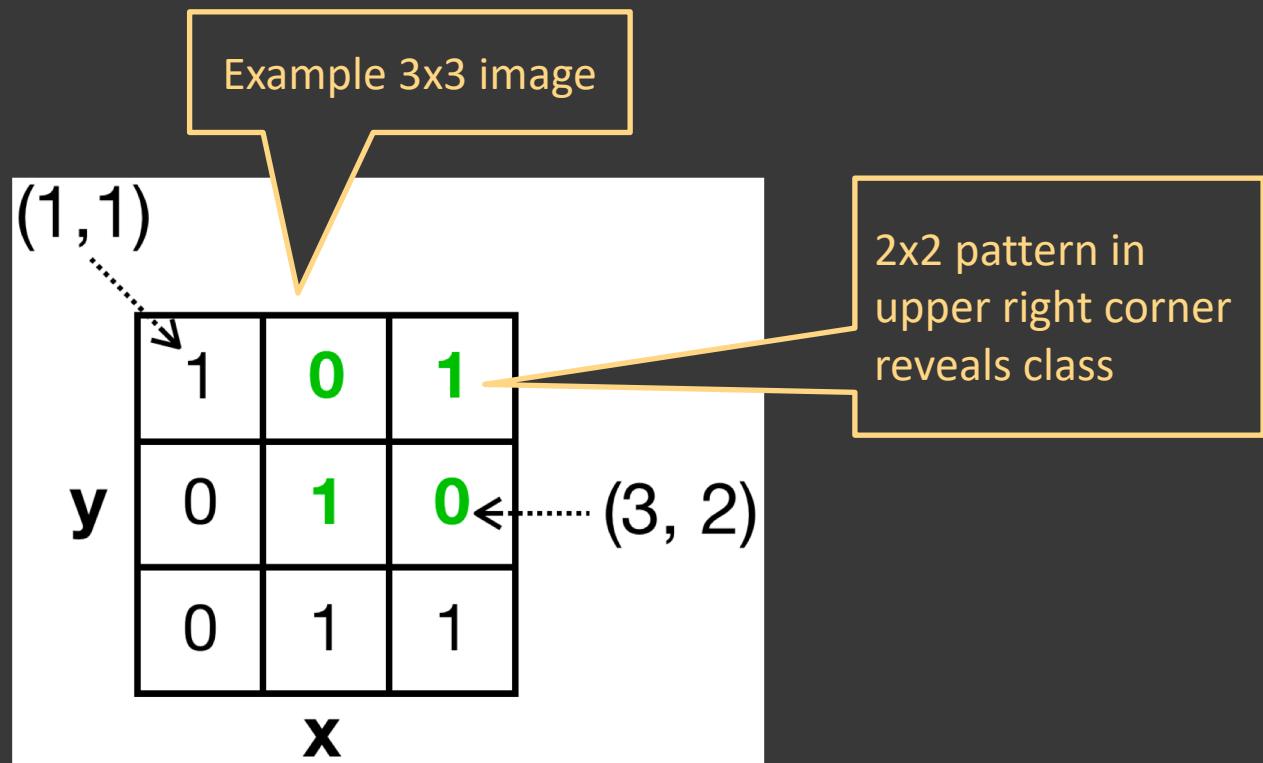
Hardware-near





Example Problem: 2D Noisy XOR

Example image



Convolution with 2x2 filters

2x2 image patch represented as four propositional variables

| | |
|-----------|-----------|
| $x_{1,1}$ | $x_{2,1}$ |
| $x_{1,2}$ | $x_{2,2}$ |

Target patterns

Class $y=0$

| | |
|---|---|
| 1 | 1 |
| 0 | 0 |

Class $y=1$

| | |
|---|---|
| 1 | 0 |
| 0 | 1 |

| | |
|---|---|
| 0 | 0 |
| 1 | 1 |

| | |
|---|---|
| 0 | 1 |
| 1 | 0 |

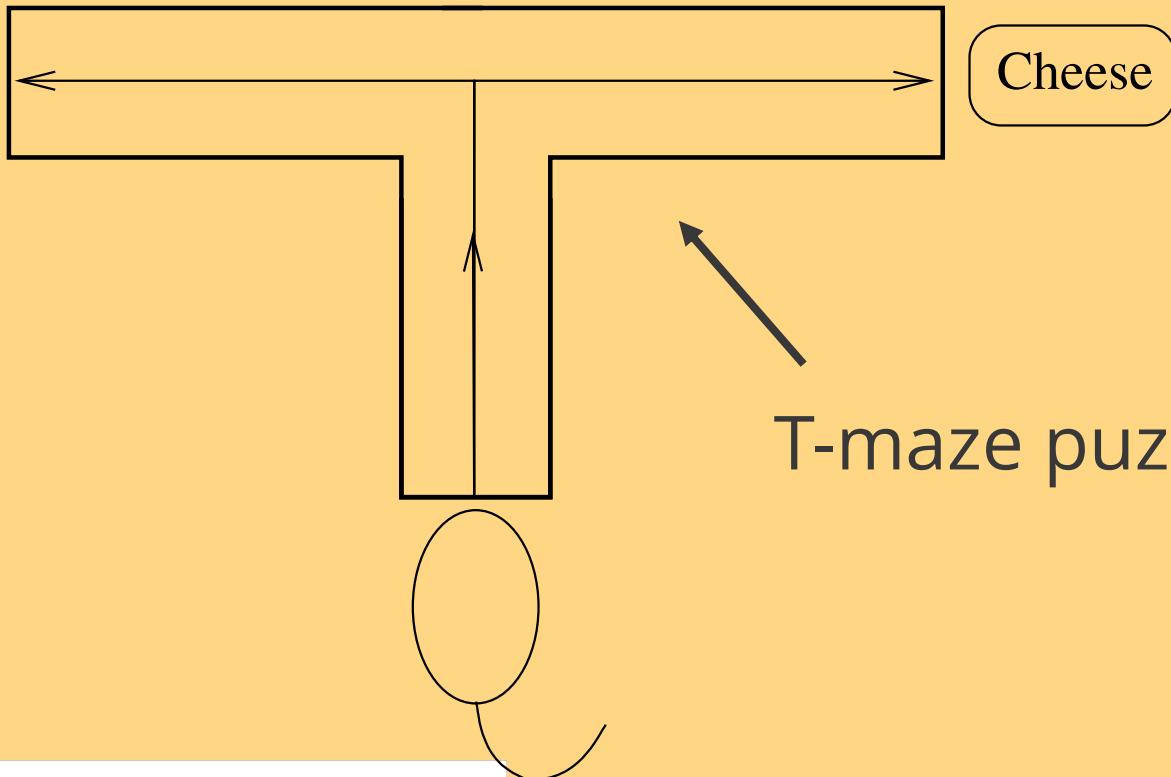
The Tsetlin Automaton

Michael Lvovitch Tsetlin

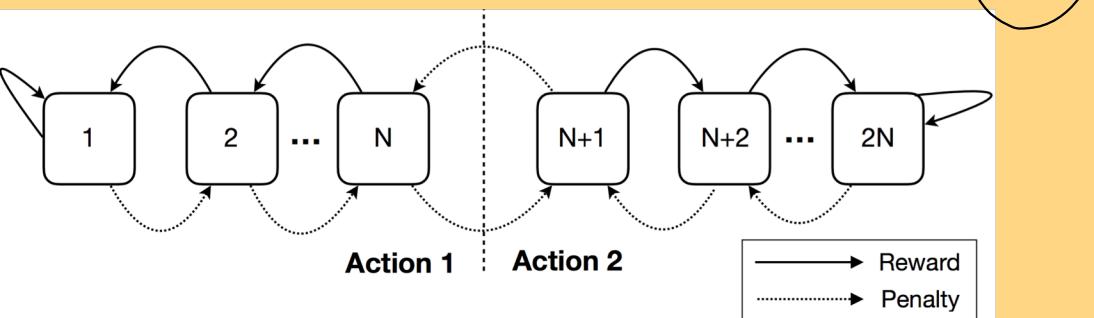


Michael Lvovitch Tsetlin (the surname is also written Cetlin, Tzetlin, Zeitlin, Zetlin; cyrillic: Михаил Львович Цетлин) (22 September 1924 – 30 May 1966) was a Russian mathematician and physicist who worked on cybernetics.

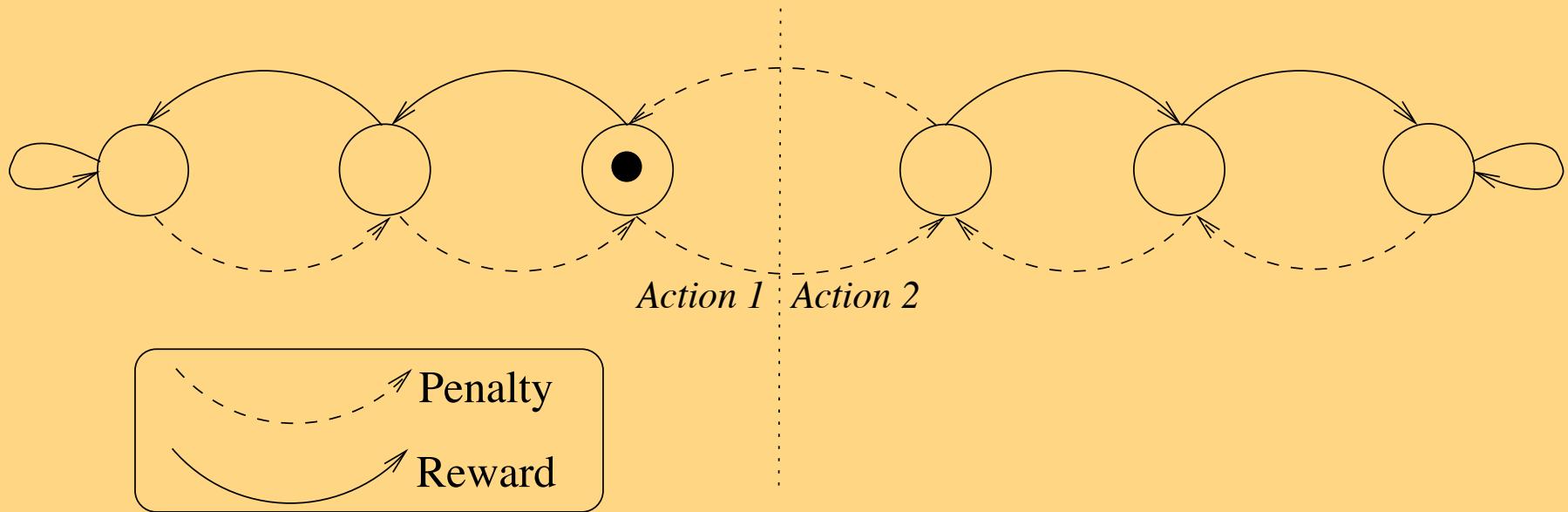
The Tsetlin Automaton



T-maze puzzle

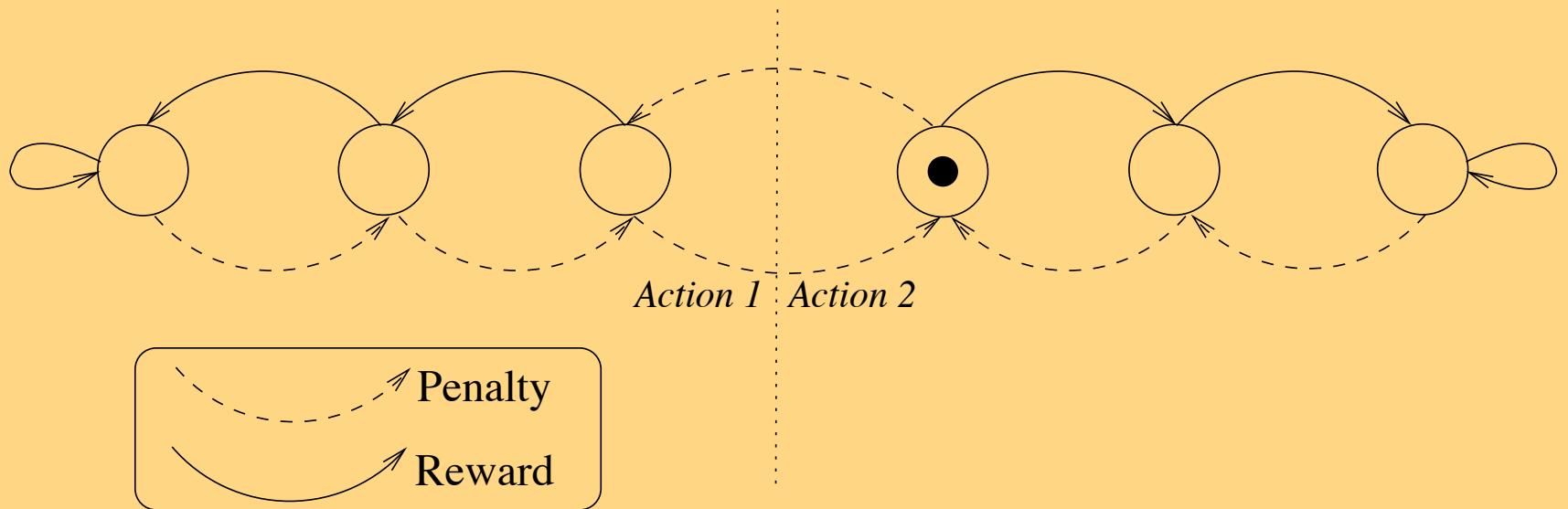


The Tsetlin automaton



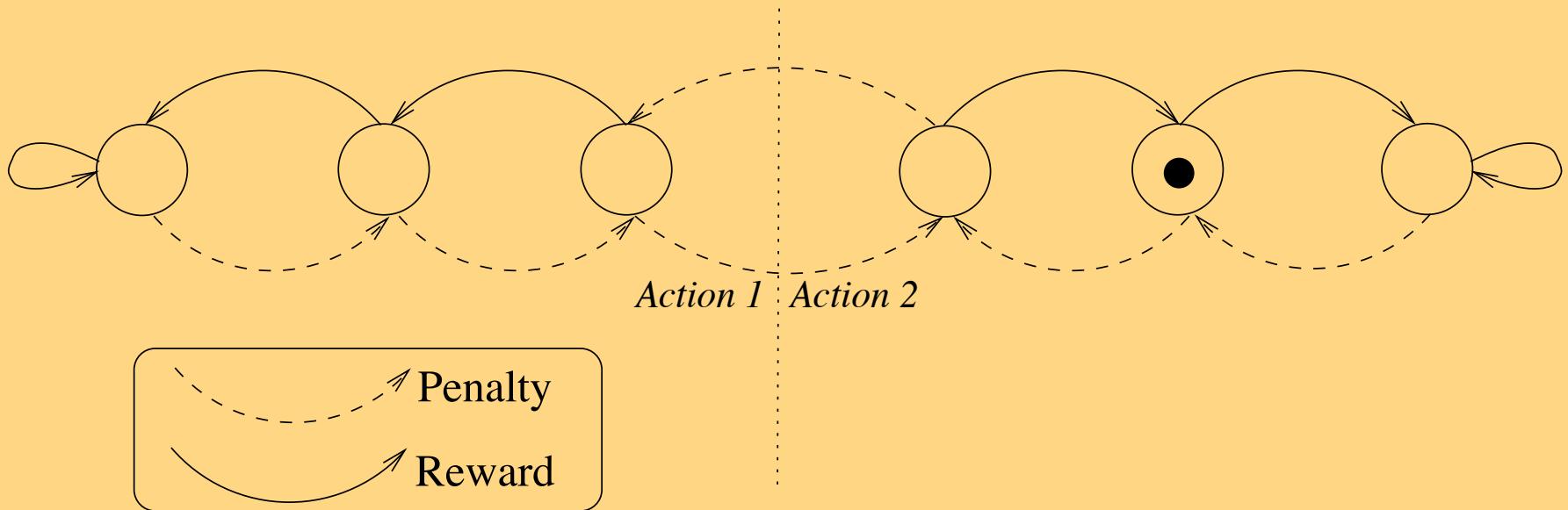
Selected Action: Action 1
Response from Environment: Penalty

The Tsetlin automaton



Selected Action: Action 2
Response from Environment: Reward

The Tsetlin automaton



Selected Action: Action 2
Response from Environment: Reward



The architecture of the Convolutional Tsetlin Machine

Bit vector input

| | |
|-----------|-----------|
| $x_{1,1}$ | $x_{2,1}$ |
| $x_{1,2}$ | $x_{2,2}$ |

$[x_{1,1}, x_{2,1}, x_{1,2}, x_{2,2}, \neg x_{1,1}, \neg x_{2,1}, \neg x_{1,2}, \neg x_{2,2}]$



Representing sub-patterns with conjunctive clauses

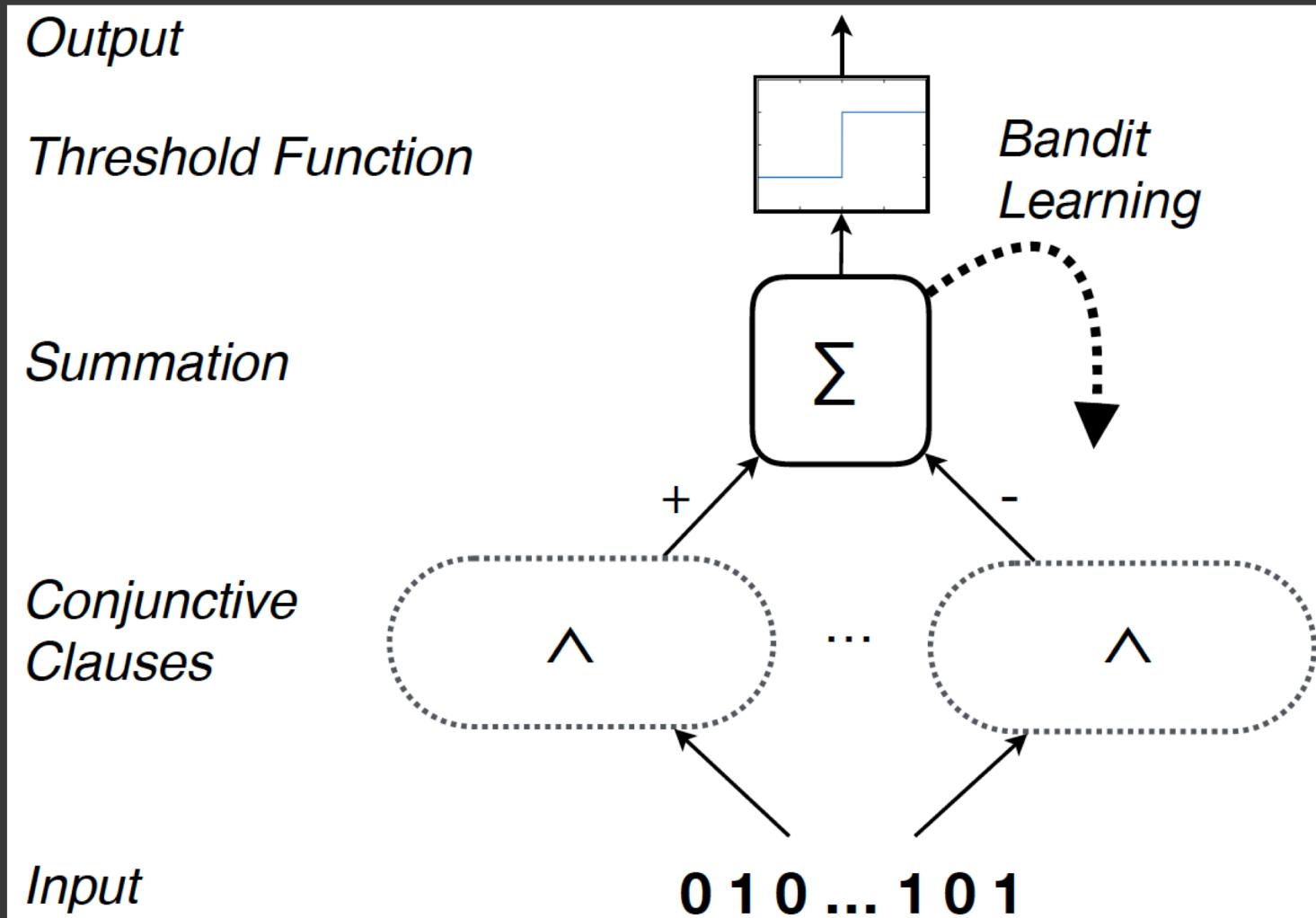
$$C = x_{1,1} \wedge x_{2,2} \wedge \neg x_{2,1} \wedge \neg x_{1,2}$$

Captures a 2x2 sub-pattern

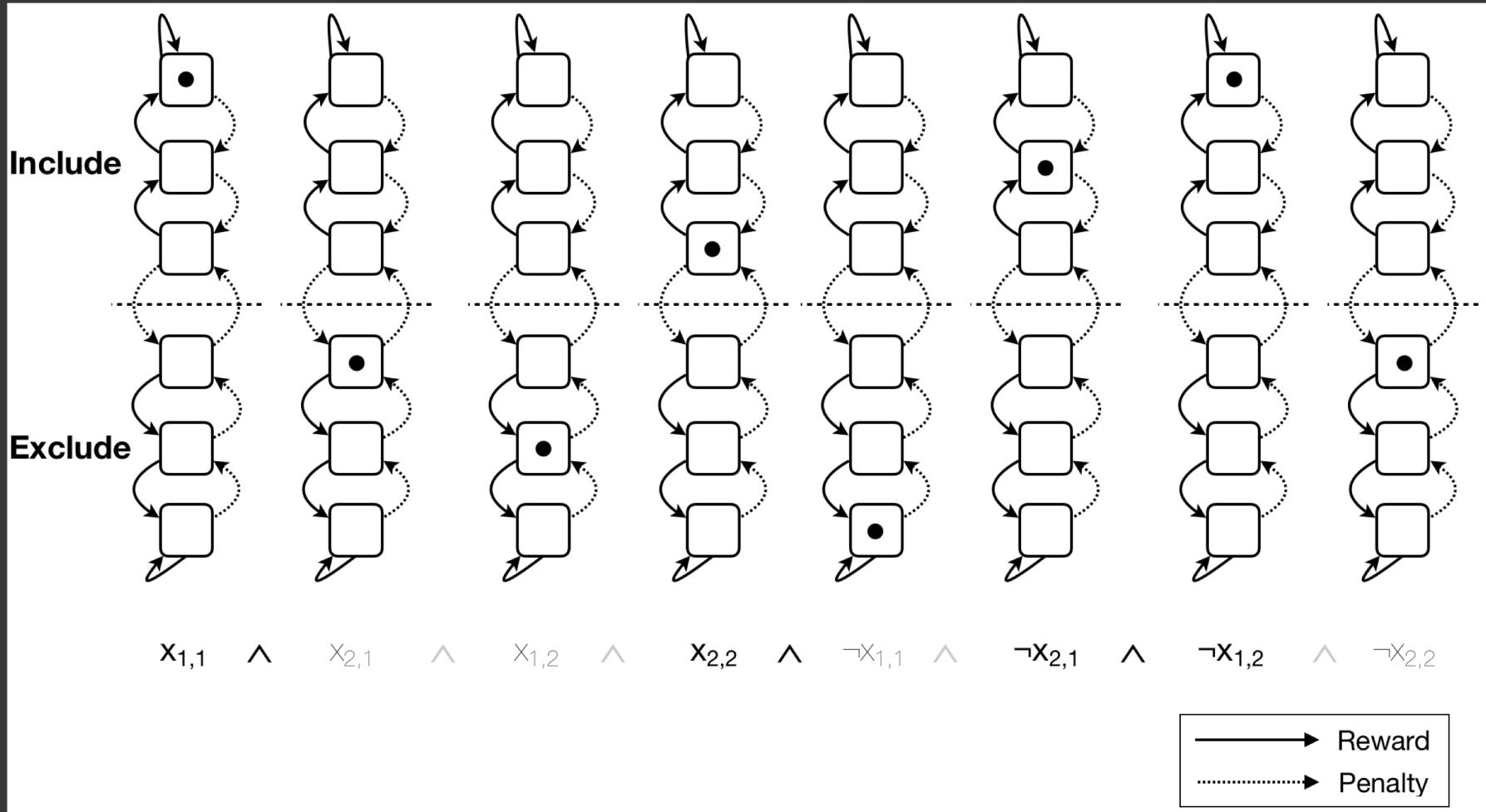
| | |
|---|---|
| 1 | 0 |
| 0 | 1 |

Evaluates to 1 on 2x2 image patch

The Tsetlin Machine



The Tsetlin Automata team for composing clauses

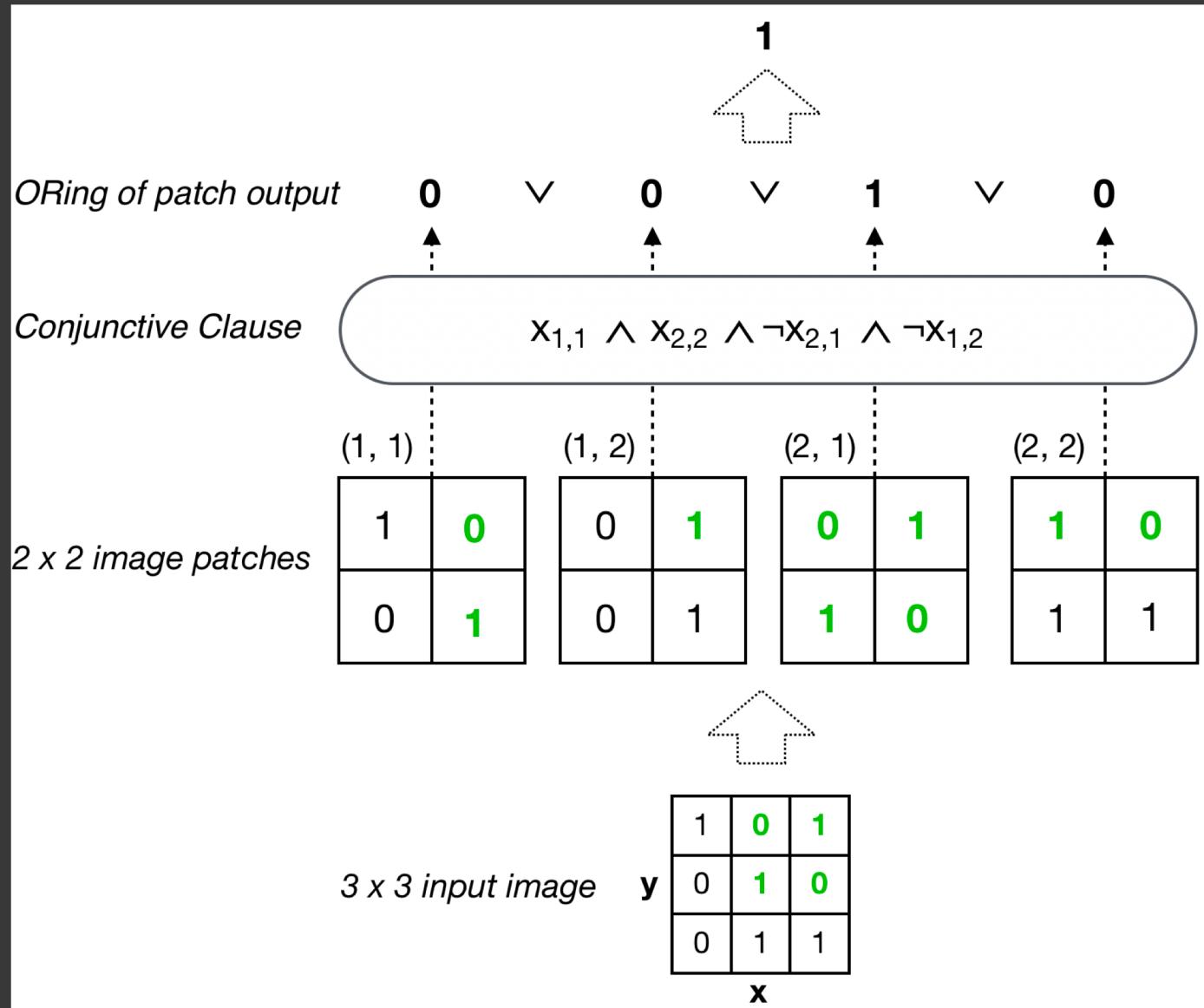


$$C = x_{1,1} \wedge x_{2,2} \wedge \neg x_{2,1} \wedge \neg x_{1,2}$$

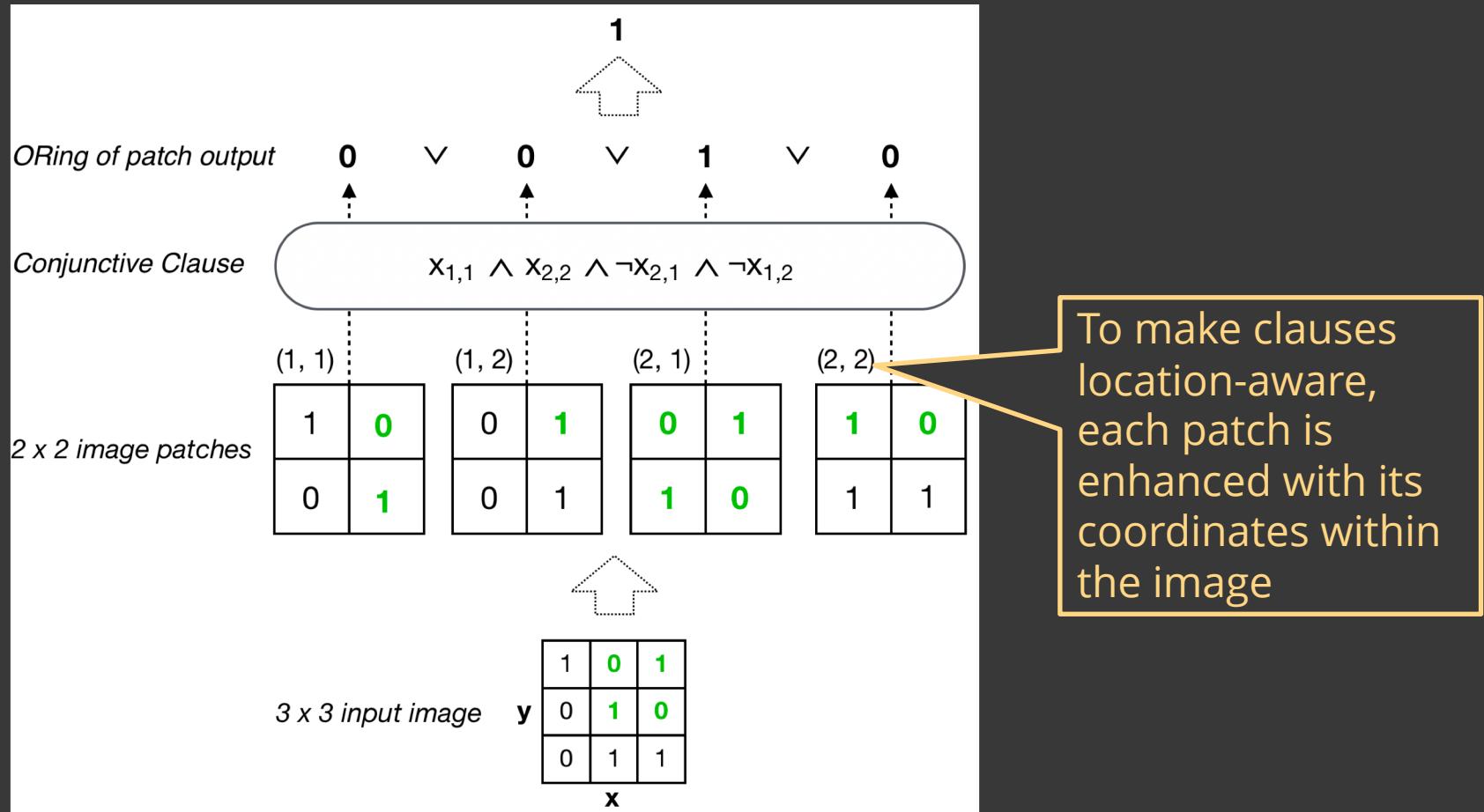


Recognition with the Convolutional Tsetlin Machine

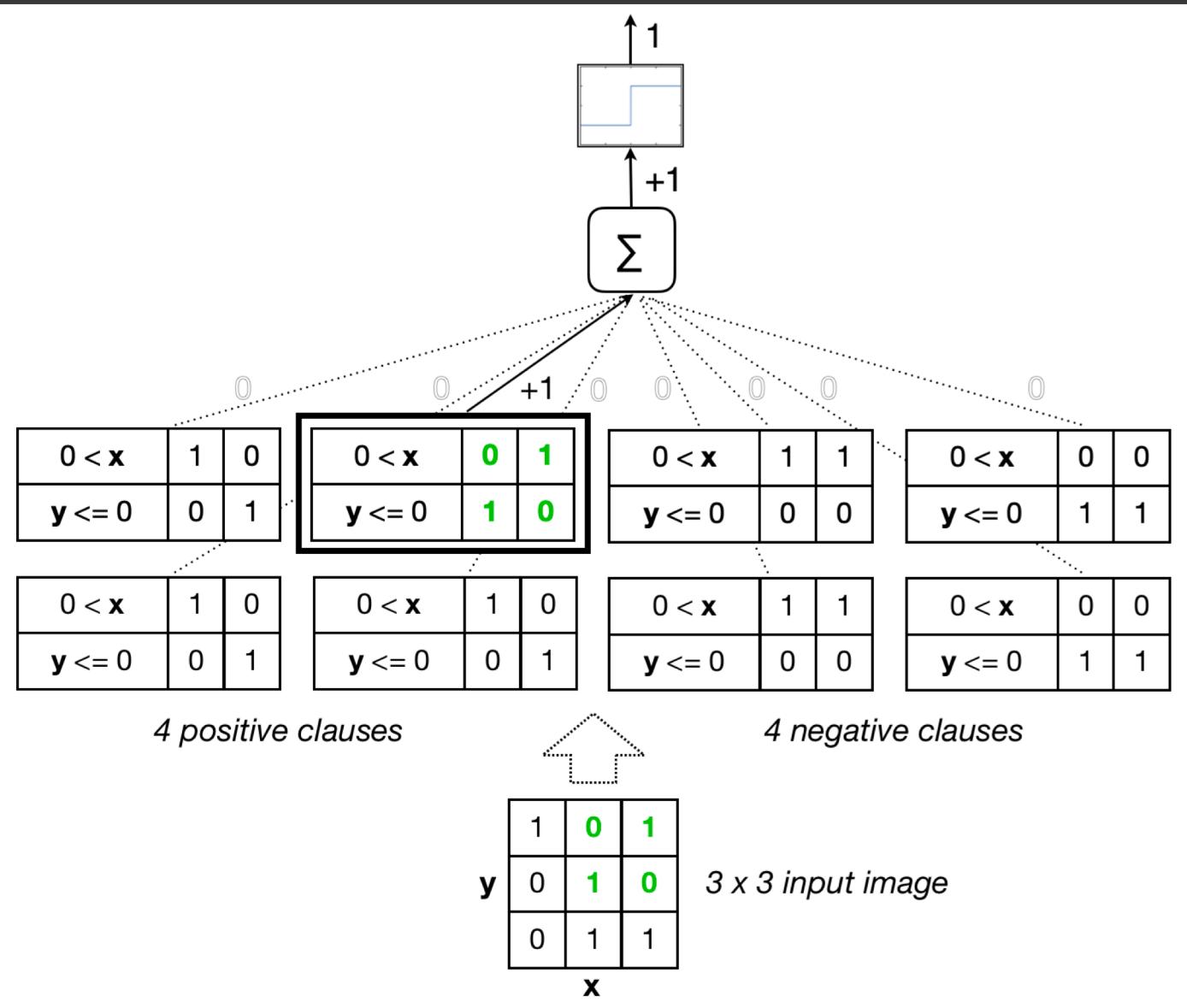
Clause convolution step



Clause convolution step



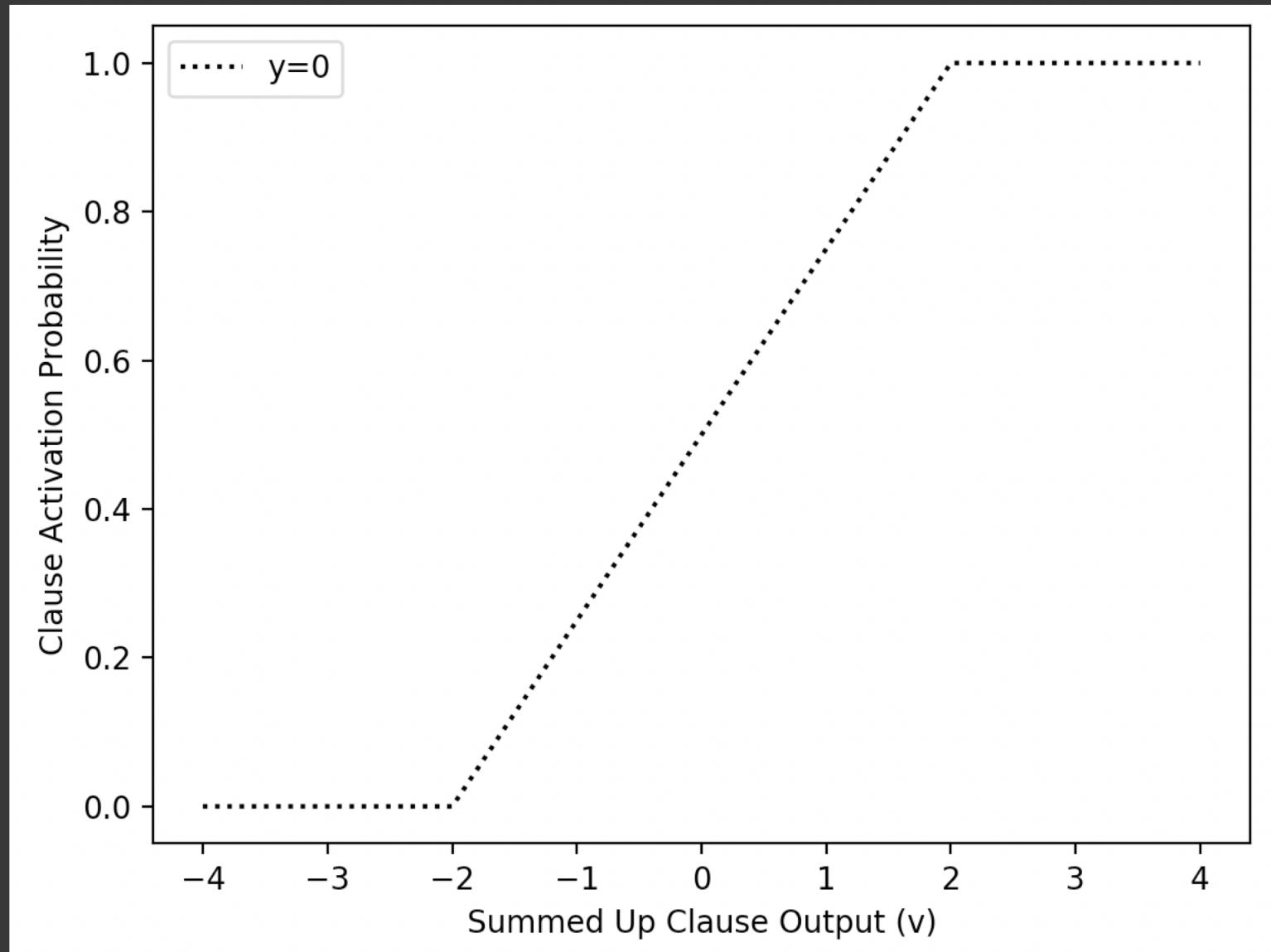
Summation and thresholding step



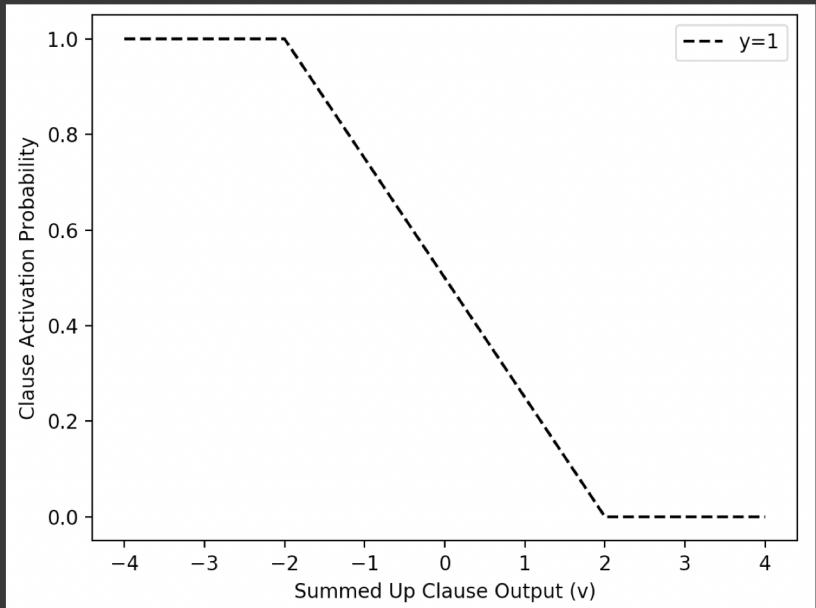
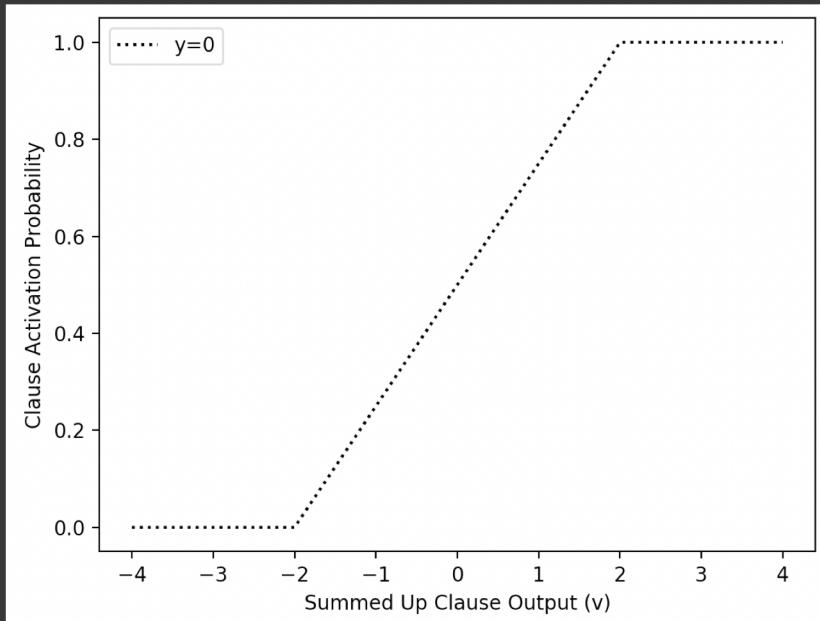


Learning with the Convolutional Tsetlin Machine

Allocation of pattern representation resources



Allocation of pattern representation resources

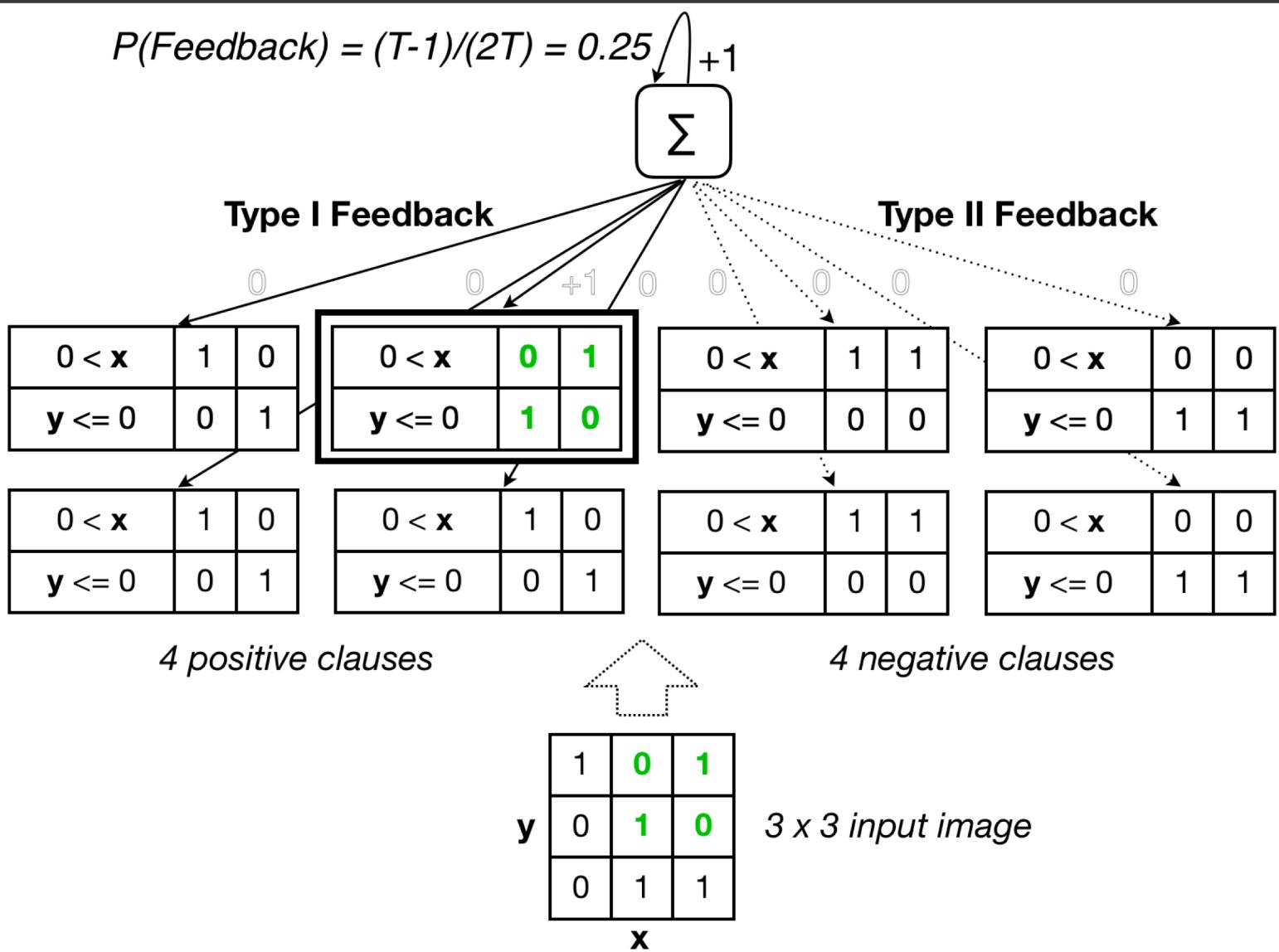


Remark 1. Observe that the future returns of a clause are diminishing with the number of other clauses that capture the same sub-pattern. This is crucial for stimulating the Tsetlin Automata teams to distribute themselves across the critical sub-patterns.

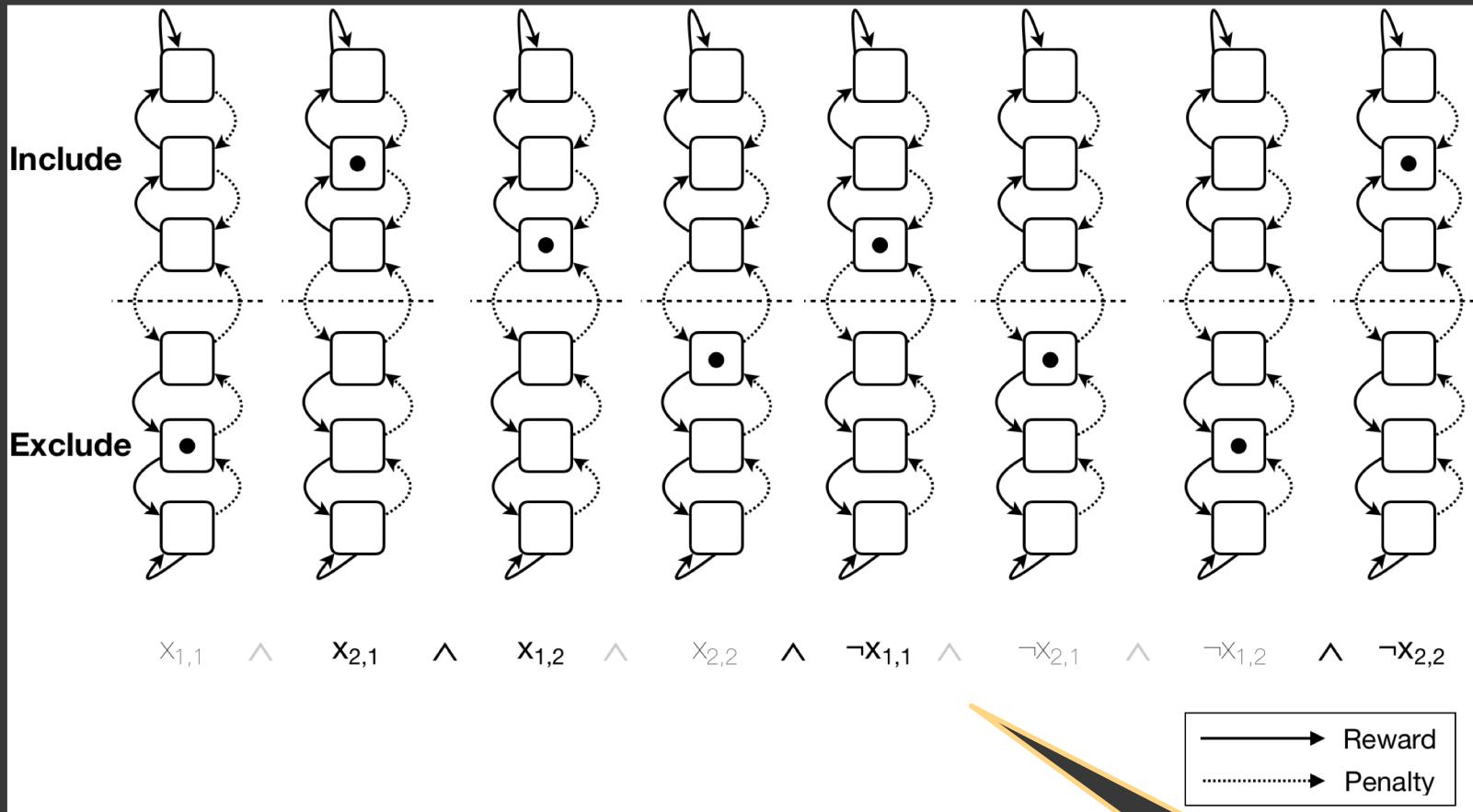
Remark 2. A larger T (with a corresponding increase in the number of clauses) makes the learning more robust. This is because more clauses are involved in learning each specific sub-pattern, introducing an ensemble effect.

Allocation of pattern representation resources

$$P(\text{Feedback}) = (T-1)/(2T) = 0.25$$



Type Ib Feedback

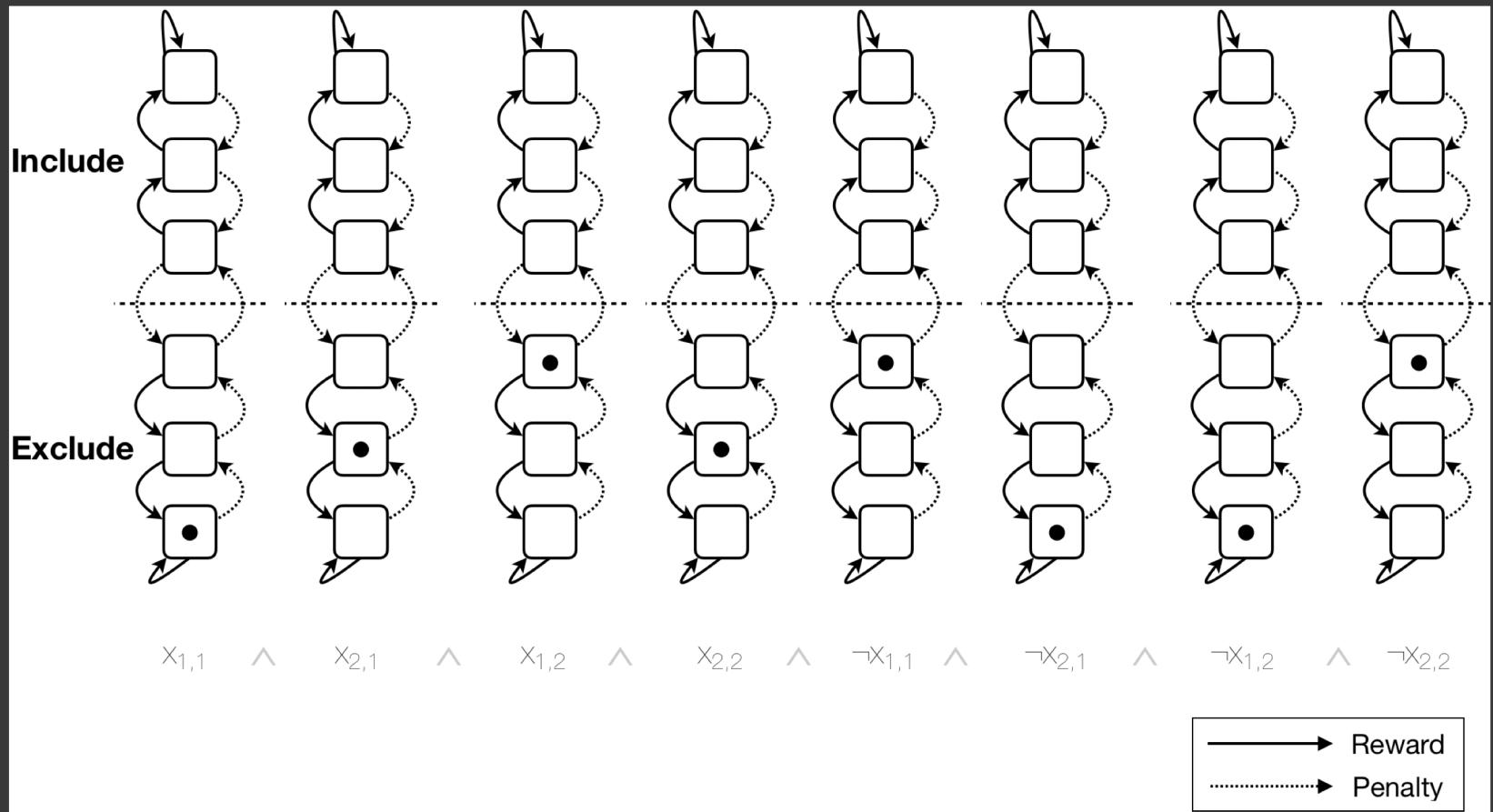


| | |
|---|---|
| 1 | 0 |
| 0 | 1 |

Underrepresented sub-pattern,
incompatible with 3 positive clauses

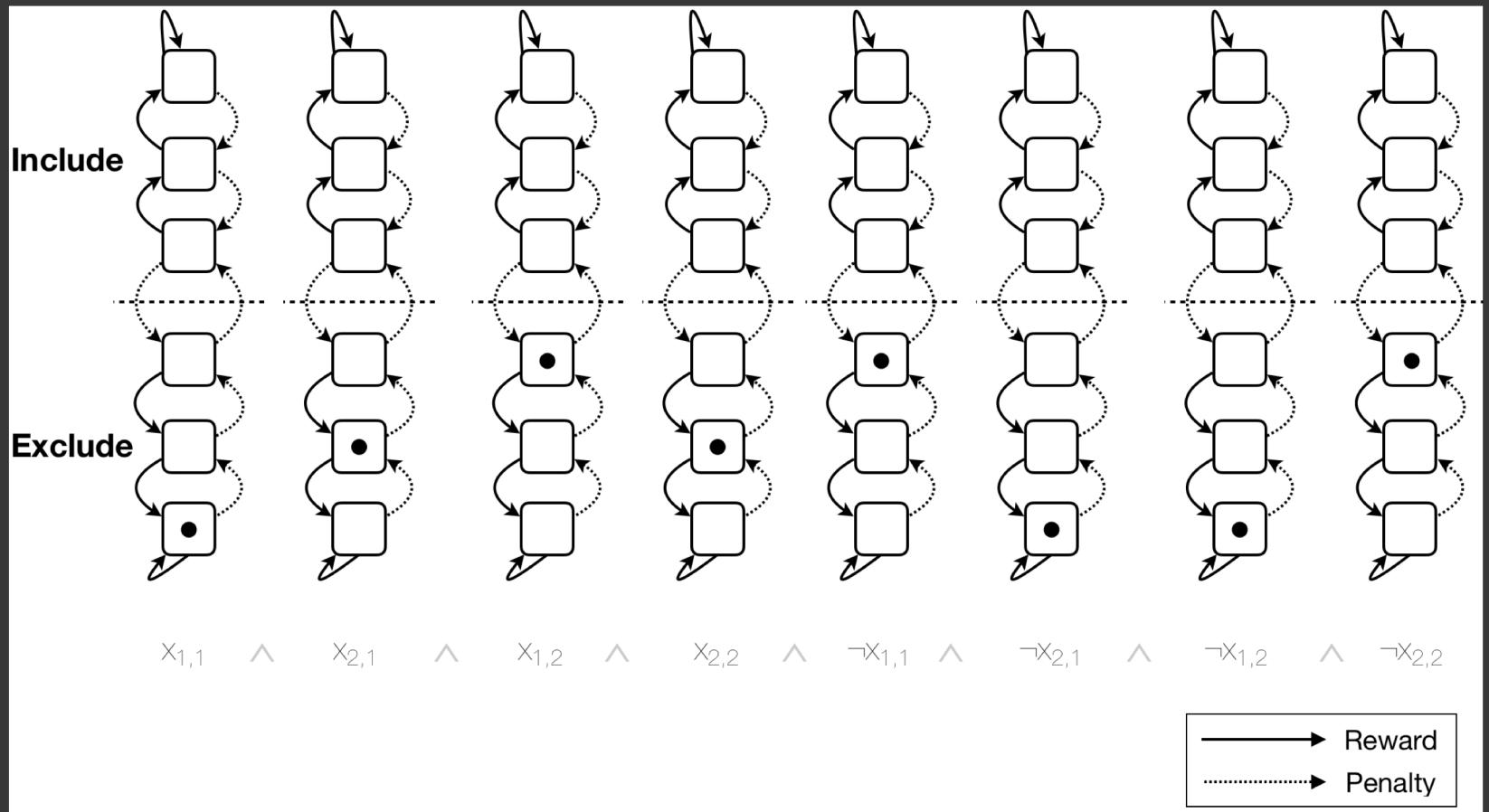
Incompatible clause

Type Ib Feedback



Combats overfitting and erases patterns to make room for new, by reinforcing Exclude actions

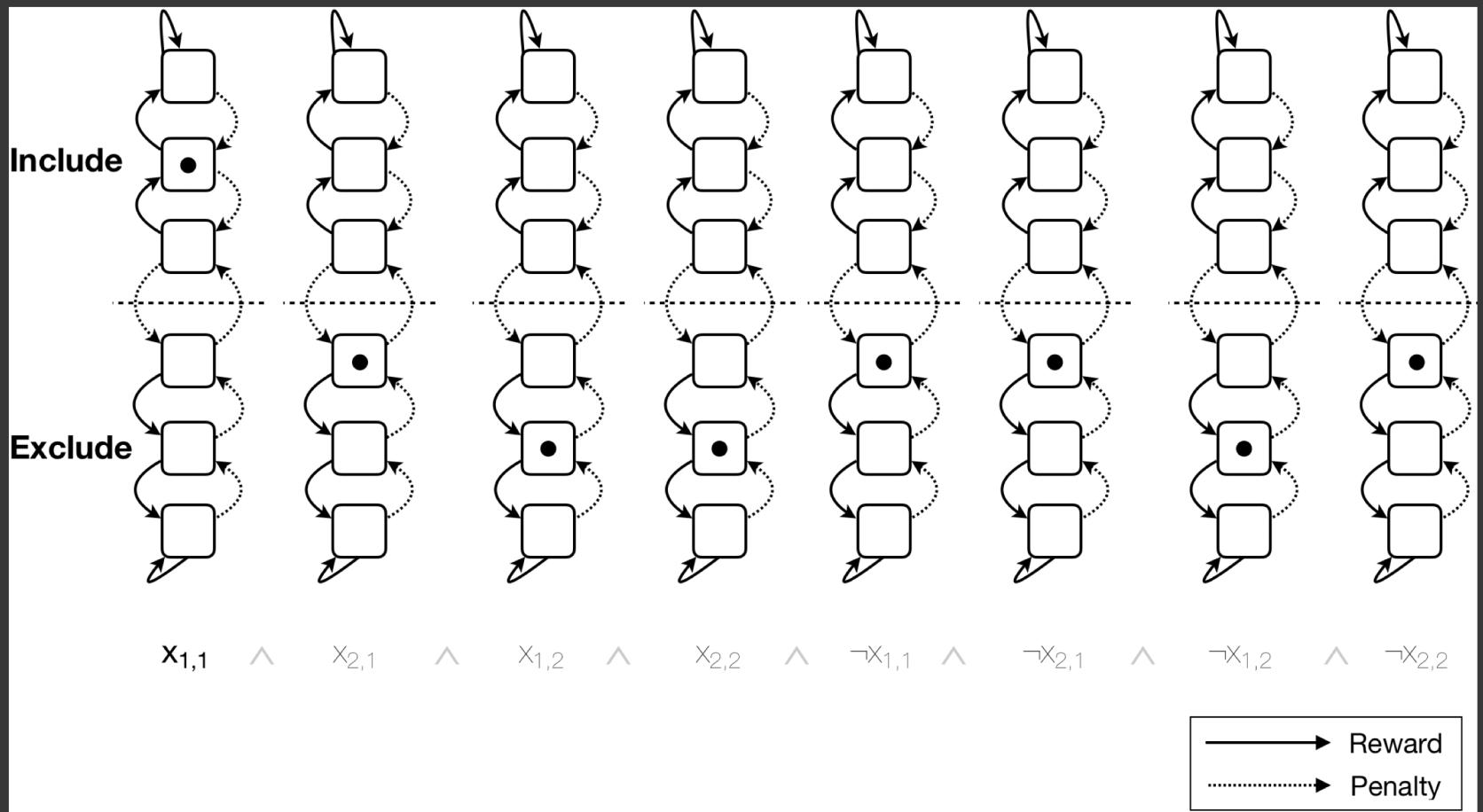
Type Ia Feedback



Increases precision of
recognized pattern

| | |
|---|---|
| 1 | 0 |
| 0 | 1 |

Type Ia Feedback

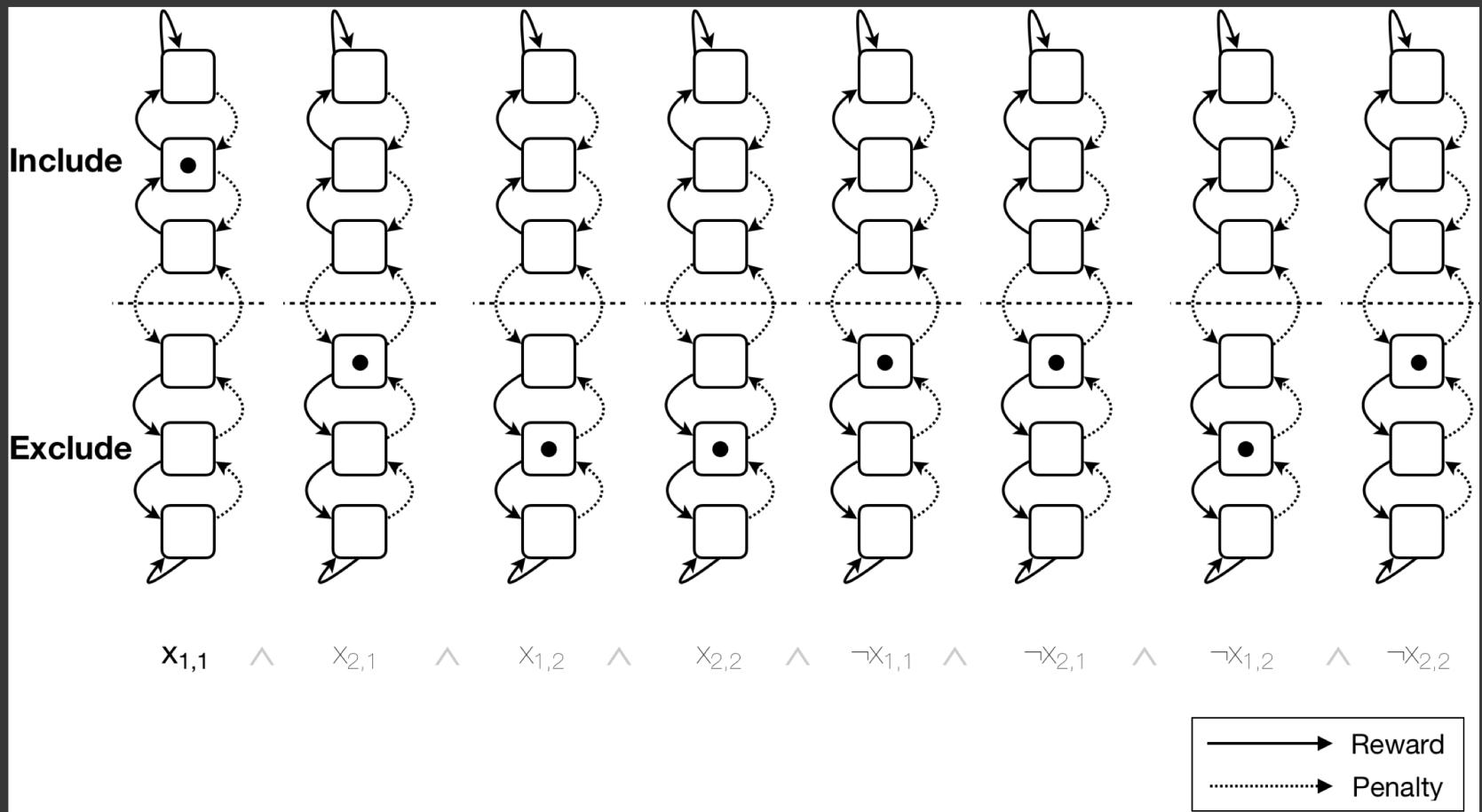


Reinforces Include action
for literals of value 1

| | |
|---|---|
| 1 | 0 |
| 0 | 1 |

Reinforces Exclude action
for literals of value 0

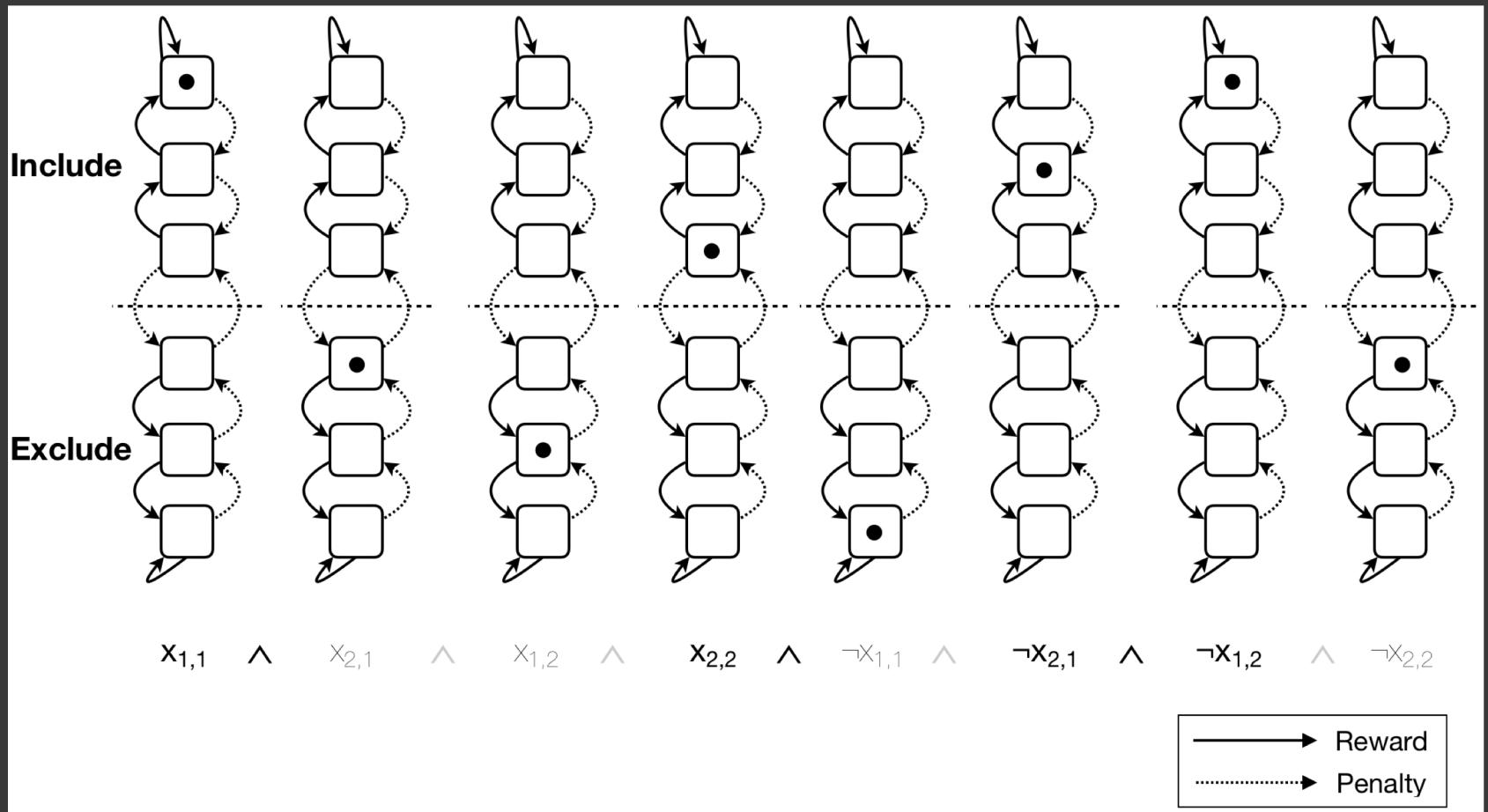
Type II Feedback



Class $y=1$ sub-pattern too loosely specified, also matching sub-patterns for class $y=0$

| | |
|---|---|
| 1 | 1 |
| 0 | 0 |

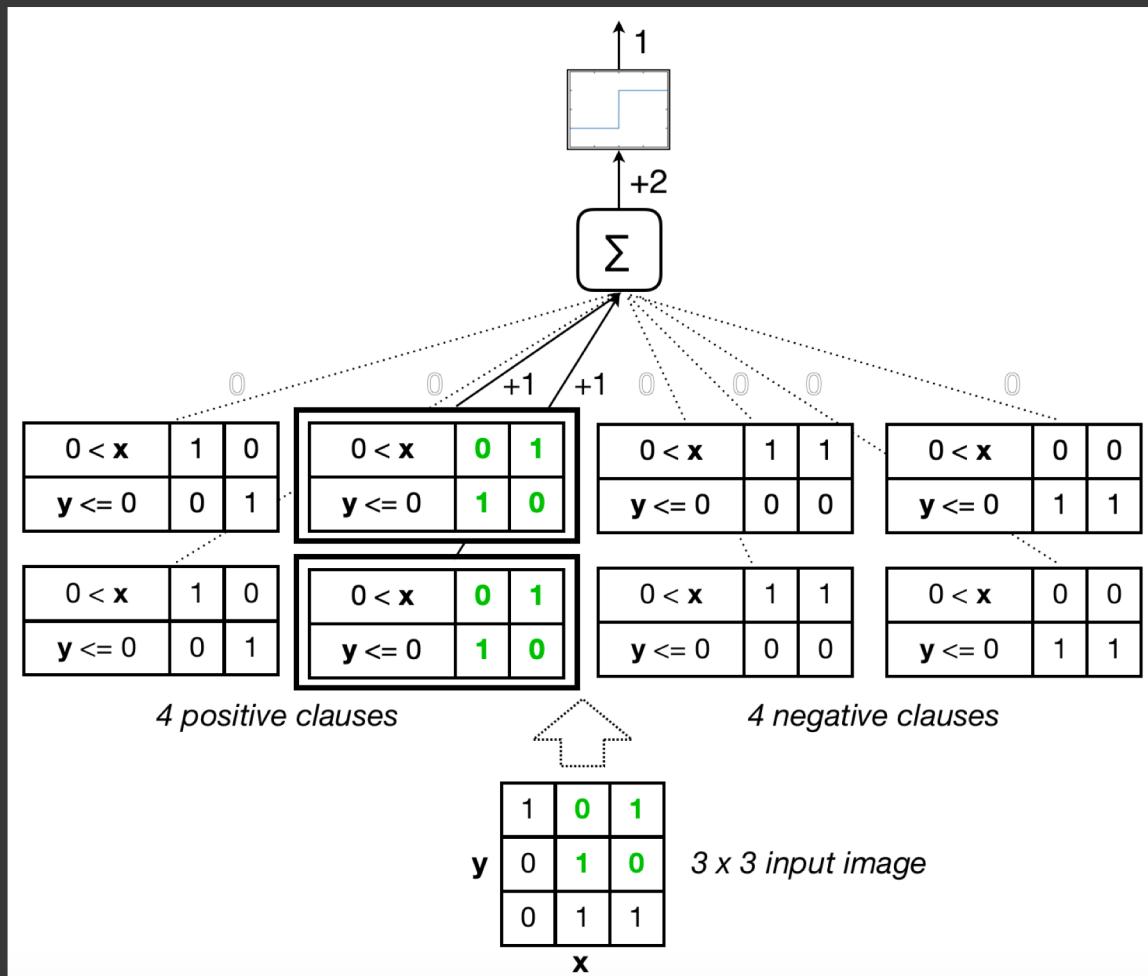
Type II Feedback



| | |
|---|---|
| 1 | 1 |
| 0 | 0 |

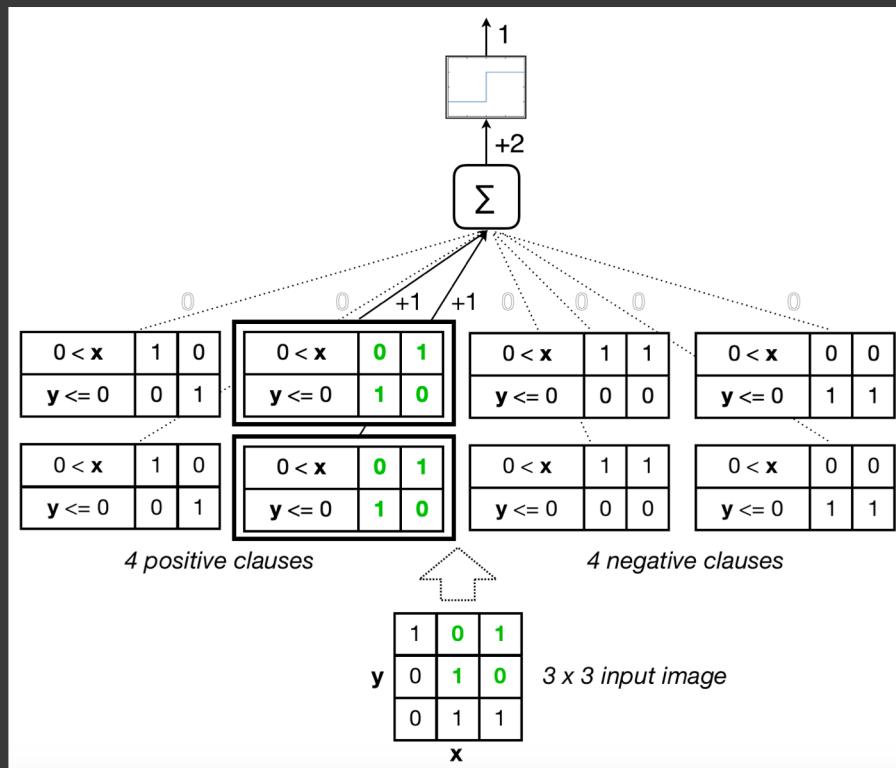
Combats false positives by reinforcing the Include action for excluded discriminating features

Goal state and Nash equilibrium



Two clauses have been allocated to represent each sub-pattern of each class

Goal state and Nash equilibrium



The Nash equilibria of the Convolutional Tsetlin Machine balance false negative against false positive classifications, while combating overfitting. This happens when Type Ia, Type Ib and Type II feedback are in balance.

Demo

Coming soon