Analog SAT solving

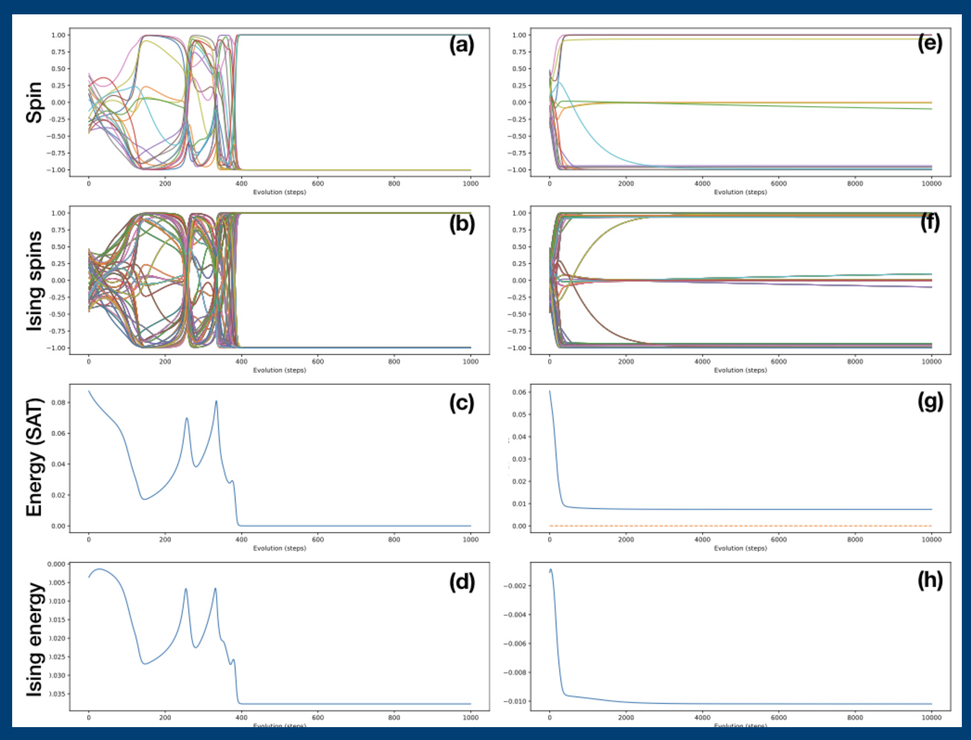
Sudoku with Neural Networks

Shahnawaz Ahmed[1,2], Clemens Gneiting[2], Anton Frisk Kockum[1], Franco Nori[2,3] and Göran Johansson[1]

Neural Networks can recognize Sudoku rules and predict correct solutions via training from examples only - one of many analog approaches where a dynamical system solves NP-hard combinatorial optimization.

The local minima problem is ubiquitous in all analog approaches. Adaptive interactions leading to transient chaos in smartly designed dynamical systems could be used to overcome it.

Can Neural Networks help?



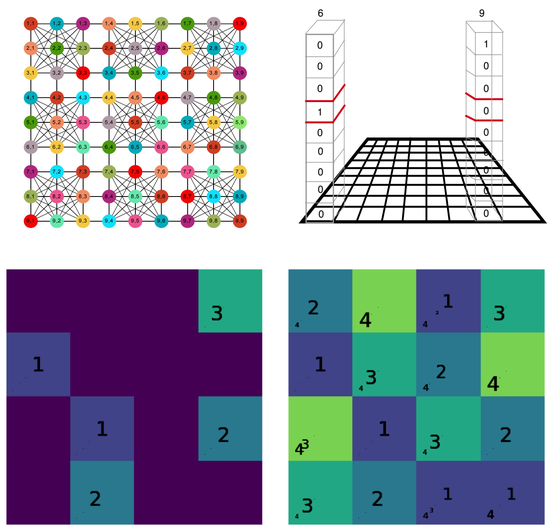
An analog approach (CTDS) finds the solution to a benchmark 3-SAT problem in 400 steps using auxiliary variables (left) but without them the even after 10,000 steps (right) a solution is not found. Energies (both 3-SAT with muti-spin interactions and Ising interactions)

A time-dependent interaction matrix which can lead to complicated dynamics could potentially be useful in avoiding local minimas. But these auxiliary variables might also grow exponentially making it hard to implement such a dynamical system on hardware. A Neural Network could be used to *learn* the best interaction via training.

[1]Wallenberg Centre for Quantum Technology, MC2, Chalmers University of Technology, 412 96 Gothenburg, Sweden
[2]Theoretical Quantum Physics Laboratory, RIKEN Cluster for Pioneering Research, Wako-shi, Saitama 351-0198, Japan
[3]Department of Physics, University of Michigan, Ann Arbor, Michigan 48109-1040, USA

Sudoku, SAT and NP-hardness

Learning Sudoku amounts to learning how to solve a hard combinatorial optimization problem such as SAT or graph coloring.



3-SAT

- N boolean variables, $x_i \in \{0, 1\}$ and m constraints as OR operations.

$$C_1 : x_2 \vee x_3 \vee \neg x_9; C_2 : x_6 \vee \neg x_2 \vee x_4; C_m$$

- Conjunctive Normal Form (CNF) of constraints: $C_1 \wedge C_2 \wedge ...C_m$

Spin formulation

- N spins, $s_i \in \{-1, 1\}$ and an energy cost for each constraint E_m

$$E_m(\vec{s}) = \prod_{i=1}^{N}(1 - c_{mi}s_i) \quad (1)$$

$c_{mi} \in \{-1, 0, 1\}$ denotes if variable s_i exists in the constraint C_m.

- Total energy, $H_{k-SAT}(\vec{s}) = \sum_{m=1}^{M} E_m^2$

Assign +/- (TRUE/FALSE) to each spin σ_i (x_i) such that each energy term (constraint) evaluates to zero (TRUE).

By design, $H_{k-SAT} = 0$ if and only if all the individual energy terms vanish. This is a complicated energy function with multi-spin interactions, e.g., $s_i s_j^2 s_k$.

Ising formulation

A mapping to the Maximum Independent Set problem converts the problem to finding the ground state of an Ising hamiltonian,

$$H_{Ising}(\vec{s}) = -\sum_{i,j} J_{ij}s_i s_j - \sum_i h_i s_i \quad (2)$$

J_{ij} encode the constraints of the problem in the spin-spin interactions and the ground state gives us the solution via some algorithm, e.g., simulated annealing. But often we face the local minima issue for difficult problems and simple dynamical methods fail.

Analog system approaches

Let spins be continuous $s_i \in [-1, 1]$ to make the energy landscape smooth.

Simple gradient descent

$$\dot{s}_i = -\nabla_{s_i} H_{Ising} = \sum_j J_{ij}s_j + h_i \quad (3)$$

Hopfield Neural Network (HNN)

$$\dot{s}_i = -r_i s_i + \sum_j J_{ij}s_j + h_i \quad (4)$$

Non-autonomous HNN (Ding et. al., 2002)

$$\dot{s}_i = -r_i s_i + \sum_j (J_{ij} + J_{ij}(t))s_j + (h_i + h_i(t)) \quad (5)$$

Also, Transiently Chaotic Neural Networks (Chen, Aihara, 1995) which proposed time-dependent interactions.

Download poster

Adaptive interactions

It is also possible to minimize the more complicated energy

$$H_{k-SAT} = \dot{s}_i = -\nabla_{s_i} H_{k-SAT} = \sum_m 2c_{mi}\frac{E_m^2}{(1 - c_{mi})} \quad (6)$$

Lagrange multiplier methods

Nagamatu and Yanaru (1996) introduced auxiliary variables for each constraint, a_m which add a weight to the energy terms as $H_{k-SAT}(\vec{s})' = \sum_{m=1}^{M} a_m E_m^2$ and give the dynamical system,

$$\dot{s}_i = \sum_m 2a_m c_{mi}\frac{E_m^2}{(1 - c_{mi})}$$
$$\dot{a}_m = E_m \quad (7)$$

Continuous time deterministic systems (CTDS)

Ravasz, M. and Toroczkai, (2011) modified this for a better dynamical system,

$$\dot{s}_i = \sum_m 2a_m c_{mi}\frac{E_m^2}{(1 - c_{mi})}$$
$$\dot{a}_m = a_m E_m \quad (8)$$

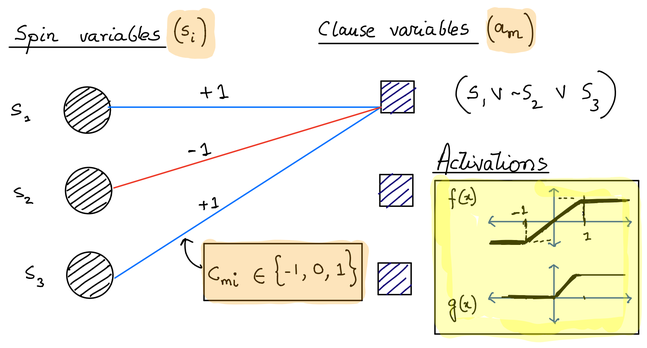
Coherent Ising Machine

A physical device implementing the following dynamics,

$$\dot{s}_i = (-1 + p - s_i^2)s_i + \sum_j J_{ij}s_j \quad (9)$$

p is the pumping rate of parameteric oscillators. This approach fails due to local minimas and needs to be augmented with an auxiliary "error" signal which makes the interaction matrix adaptive and time-dependent (Leleu et. al., 2019) by modifying the interaction term $e_i(t)\sum_j J_{ij}s_j$.

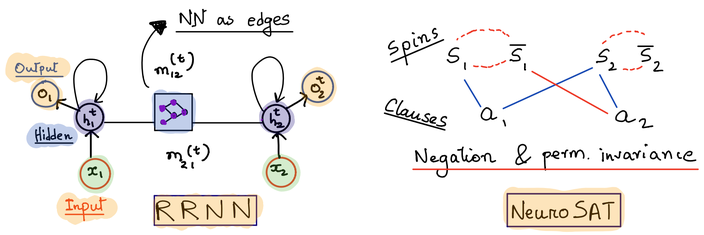
Assymmetric continuous time NN: Molnár and Ercsey-Ravasz (2013) considered a system with two types of activation functions resembling tanh and sigmoid (f and g),



$$\dot{s}_i = -s_i + Af(s_i) + \sum_m c_{mi}g(a_m)$$
$$\dot{a}_m = -a_m + Bg(a_m) - \sum_i c_{mi}f(s_i)$$
$$+ 1 - k \quad (10)$$

Neural Networks for SAT

Graph Neural Networks (GNN):



- NNs such as **Recurrent Relational Neural Network (RRNN)** (Palm et. al., 2018) or **NeuroSAT** (Selsam et. al., 2018) could be trained to solve SAT problems. RRNN was used to solve difficult Sudoku instances. NeuroSAT could generalize to new domains and predict solutions after only being trained to classify satisfiability.
- A system of evolving units (spins and constraint variables) which interact with each other dynamically to find the solution to a given problem.