UPPSALA UNIVERSITET

Memristive Boltzmann Machine: A Hardware Accelerator for Combinatorial Optimization and Deep Learning*

Chris Kjellqvist

*Mahdi Nazm Bojnordi and Engin Ipek. Memristive boltzmann machine: A hardware accelerator for combinatorial optimization and deep learning. In Proceedings on the 22nd IEEE Symposium on High Performance Computer Architecture, Rochester, NY, March 2016.

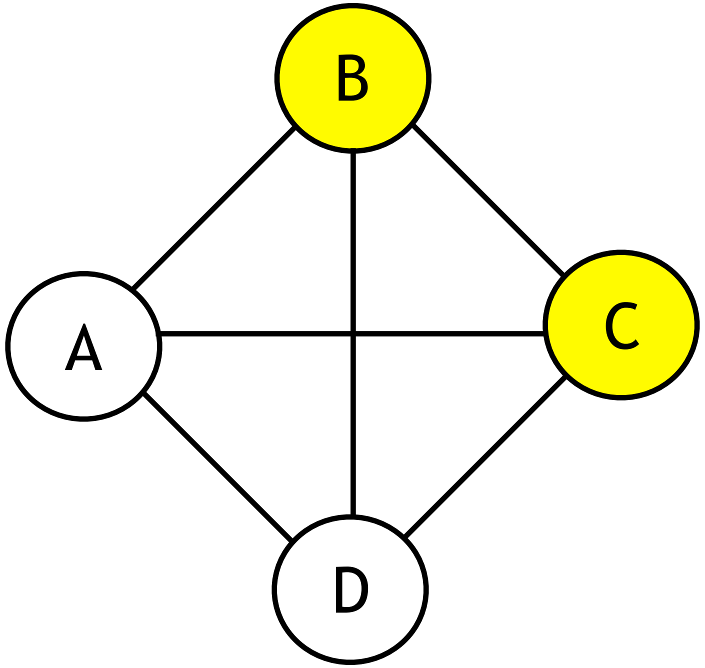
UPPSALA UNIVERSITET

The Boltzmann Machine, Problem Mapping, and Memristors

BACKGROUND

The Boltzmann Machine

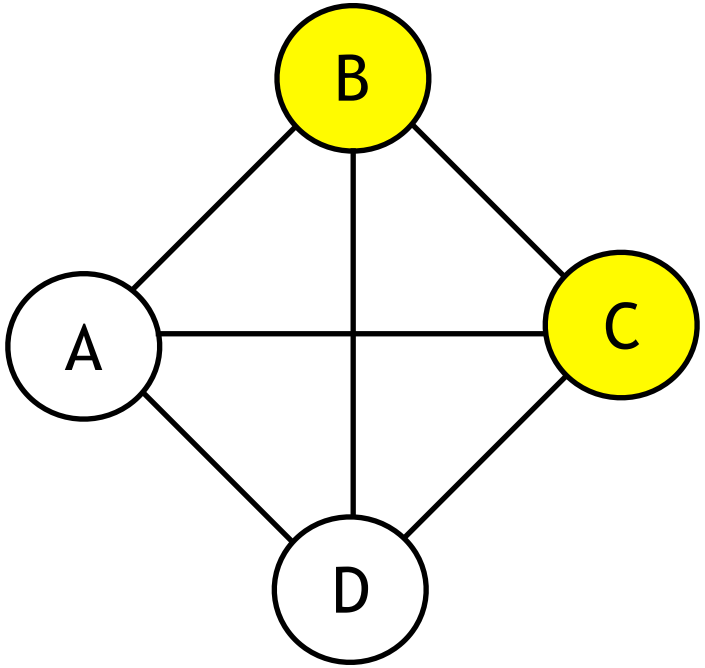
- A Fully Connected Stochastic Recurrent Neural Network capable of solving optimization problems



1) Ackley, David H; Hinton Geoffrey E; Sejnowski, Terrence J (1985), "A learning algorithm for Boltzmann machines", Cognitive Science, 9 (1): 147–169, doi:10.1207/s15516709cog0901_7

The Boltzmann Machine

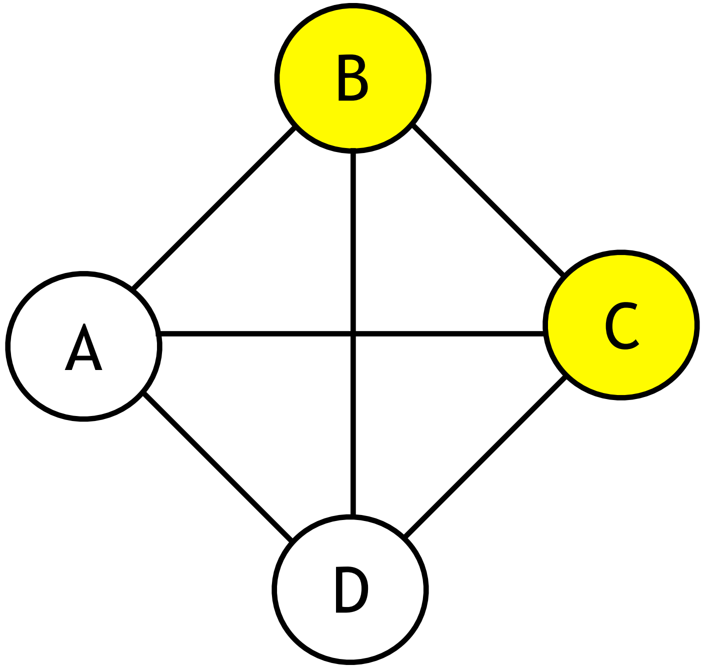
- A Fully Connected Stochastic Recurrent Neural Network capable of solving optimization problems
  - Traveling Salesman



1) Ackley, David H; Hinton Geoffrey E; Sejnowski, Terrence J (1985), "A learning algorithm for Boltzmann machines", Cognitive Science, 9 (1): 147–169, doi:10.1207/s15516709cog0901_7

The Boltzmann Machine

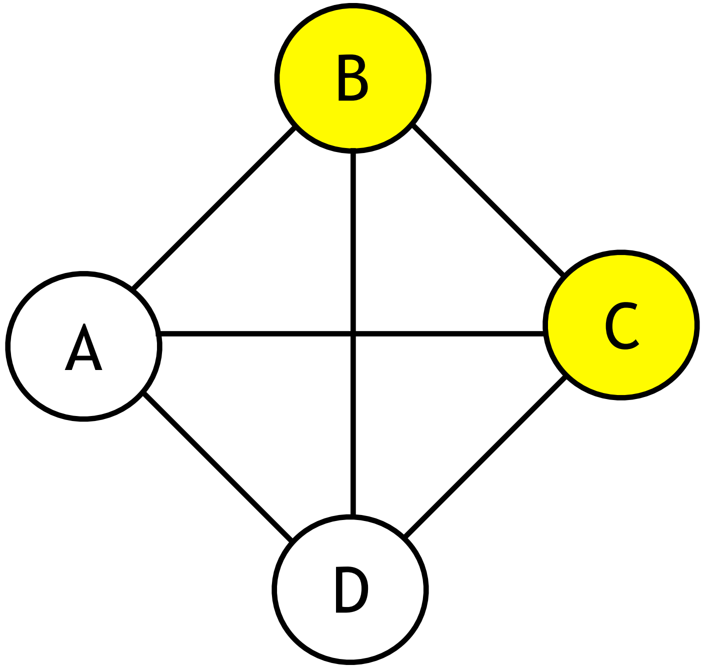
- A Fully Connected Stochastic Recurrent Neural Network capable of solving optimization problems
  - Traveling Salesman
  - Max-Cut



1) Ackley, David H; Hinton Geoffrey E; Sejnowski, Terrence J (1985), "A learning algorithm for Boltzmann machines", Cognitive Science, 9 (1): 147–169, doi:10.1207/s15516709cog0901_7

The Boltzmann Machine

- A Fully Connected Stochastic Recurrent Neural Network capable of solving optimization problems
  - Traveling Salesman
  - Max-Cut
  - Maximum Satisfiability

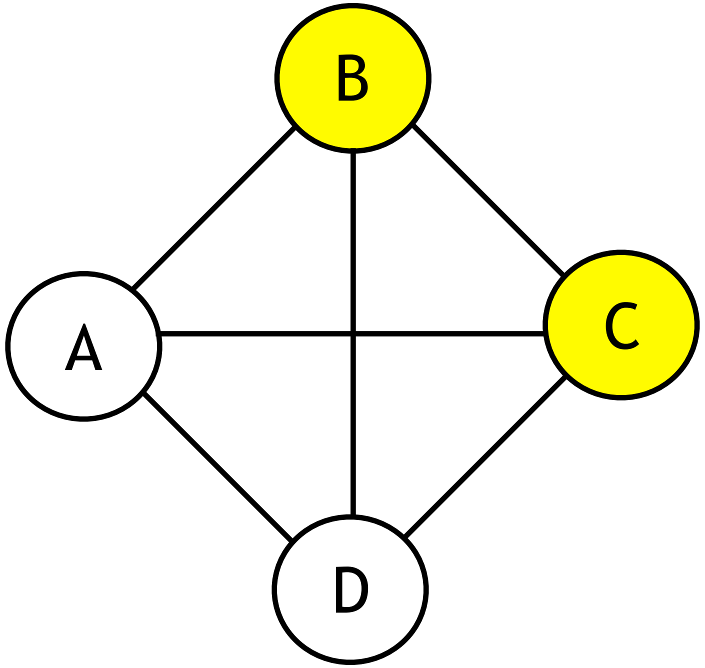


1) Ackley, David H; Hinton Geoffrey E; Sejnowski, Terrence J (1985), "A learning algorithm for Boltzmann machines", Cognitive Science, 9 (1): 147–169, doi:10.1207/s15516709cog0901_7

UPPSALA UNIVERSITET

The Boltzmann Machine

- A Fully Connected Stochastic Recurrent Neural Network capable of solving optimization problems
  - Traveling Salesman
  - Max-Cut
  - Maximum Satisfiability
  - Deep Learning

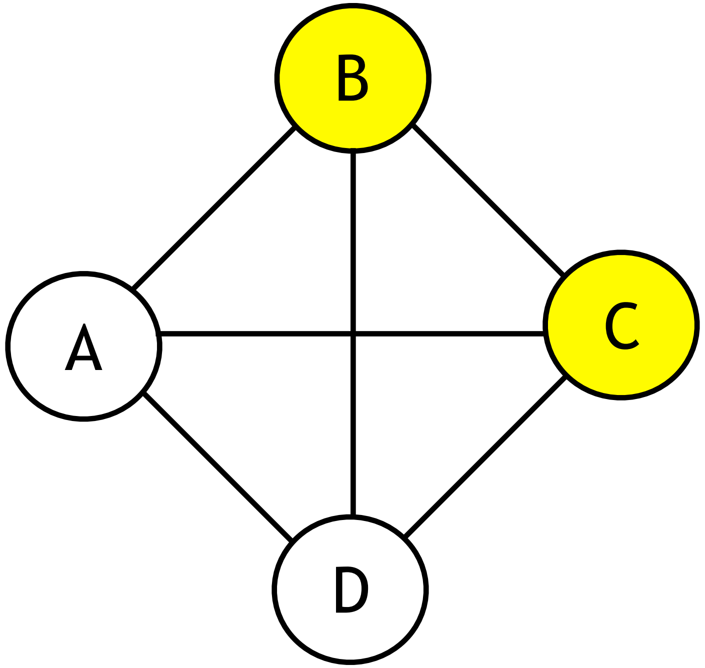


1) Ackley, David H; Hinton Geoffrey E; Sejnowski, Terrence J (1985), "A learning algorithm for Boltzmann machines", Cognitive Science, 9 (1): 147–169, doi:10.1207/s15516709cog0901_7

UPPSALA UNIVERSITET

The Boltzmann Machine

- A Fully Connected Stochastic Recurrent Neural Network capable of solving optimization problems
- Nodes change state (on or off) as a function of their neighbor states and corresponding edge weights
- Goal to maximize "consensus" function



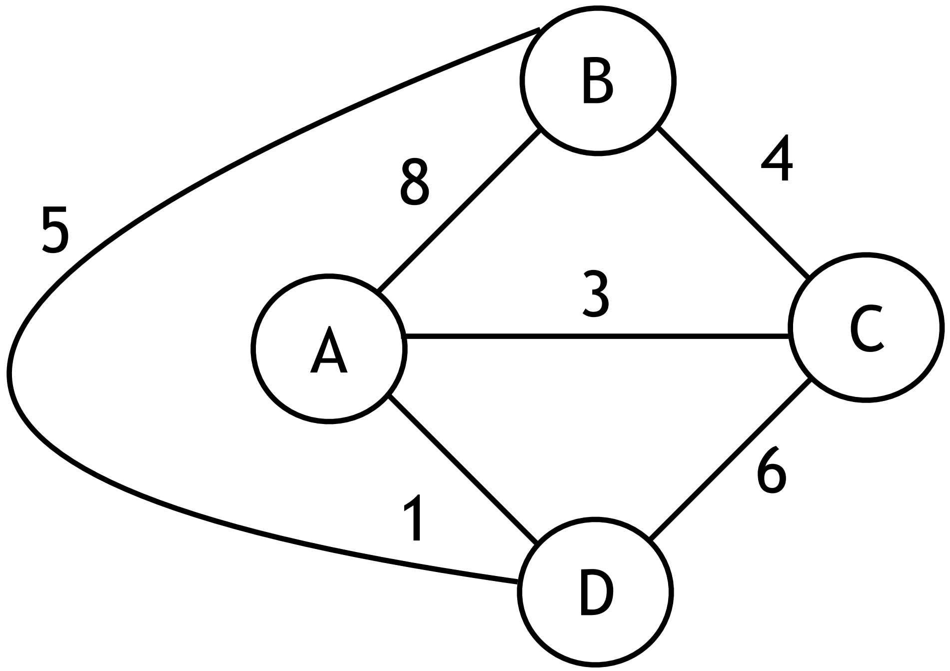
1) Ackley, David H; Hinton Geoffrey E; Sejnowski, Terrence J (1985), "A learning algorithm for Boltzmann machines", Cognitive Science, 9 (1): 147–169, doi:10.1207/s15516709cog0901_7

Example Problem: Max-Cut

- Given graph G , find a subset of graph G, S, such that the sum of the connection weights between the graphs are maximized

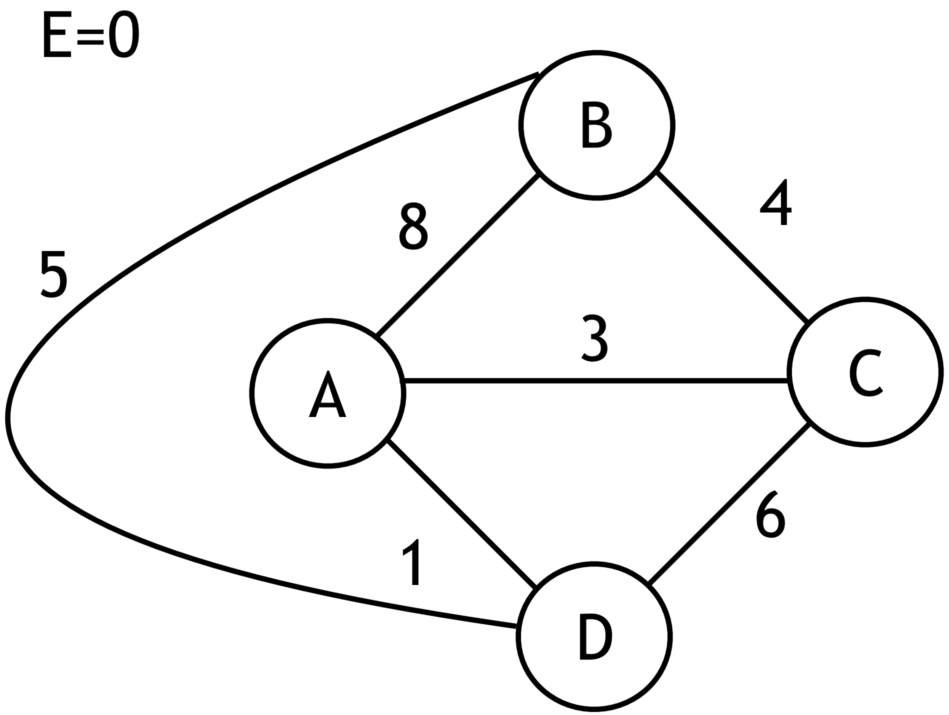
Example Problem: Max-Cut

- Given graph G , find a subset of graph G, S, such that the sum of the connection weights between the graphs are maximized



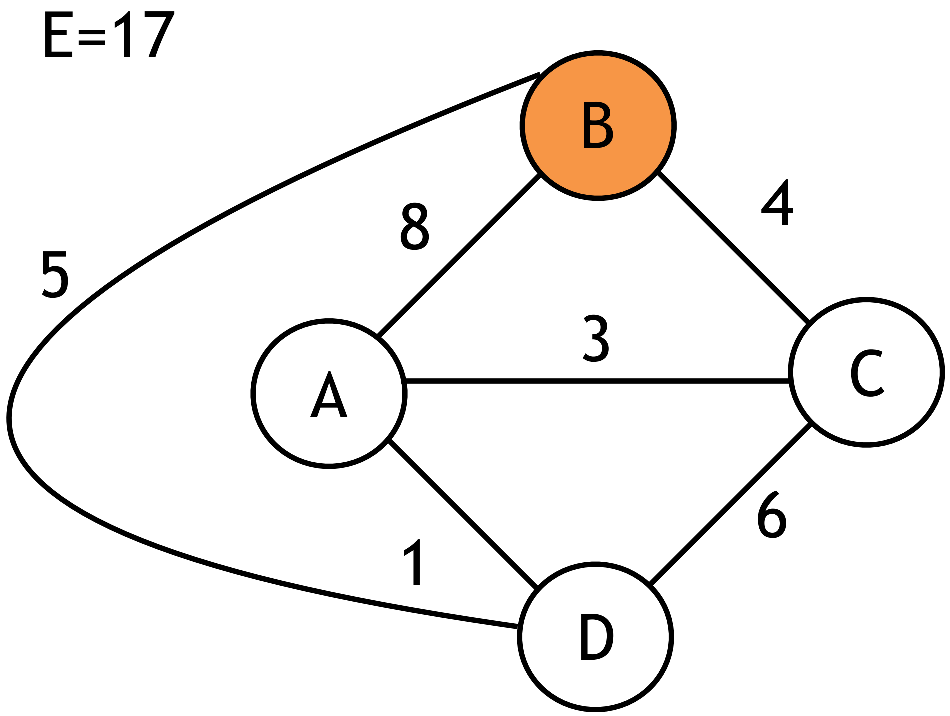
Example Problem: Max-Cut

- Given graph G , find a subset of graph G, S, such that the sum of the connection weights between the graphs are maximized



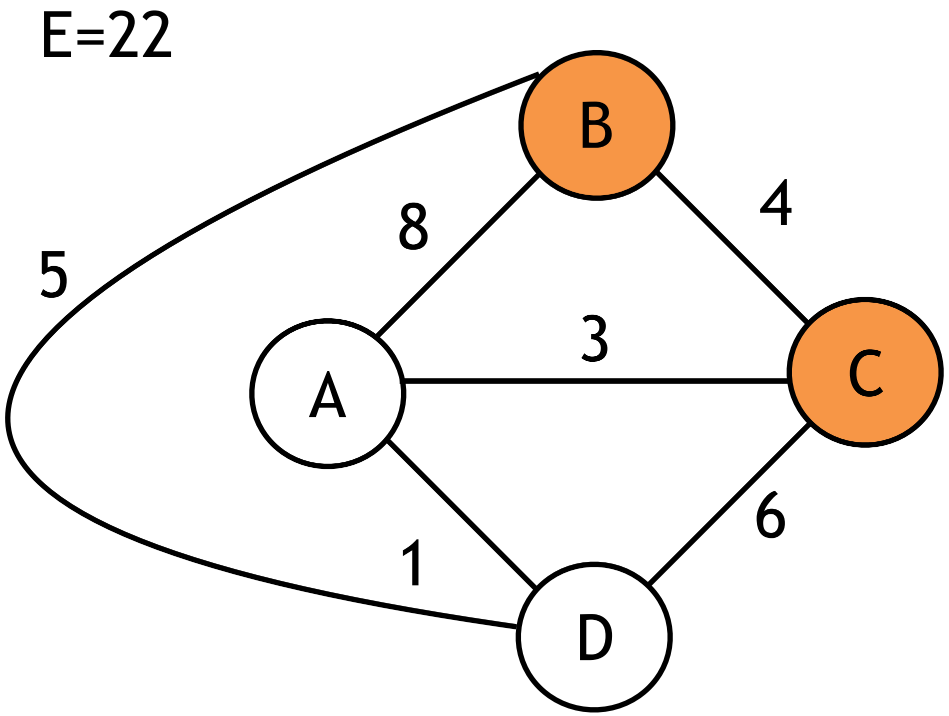
Example Problem: Max-Cut

- Given graph G , find a subset of graph G, S, such that the sum of the connection weights between the graphs are maximized



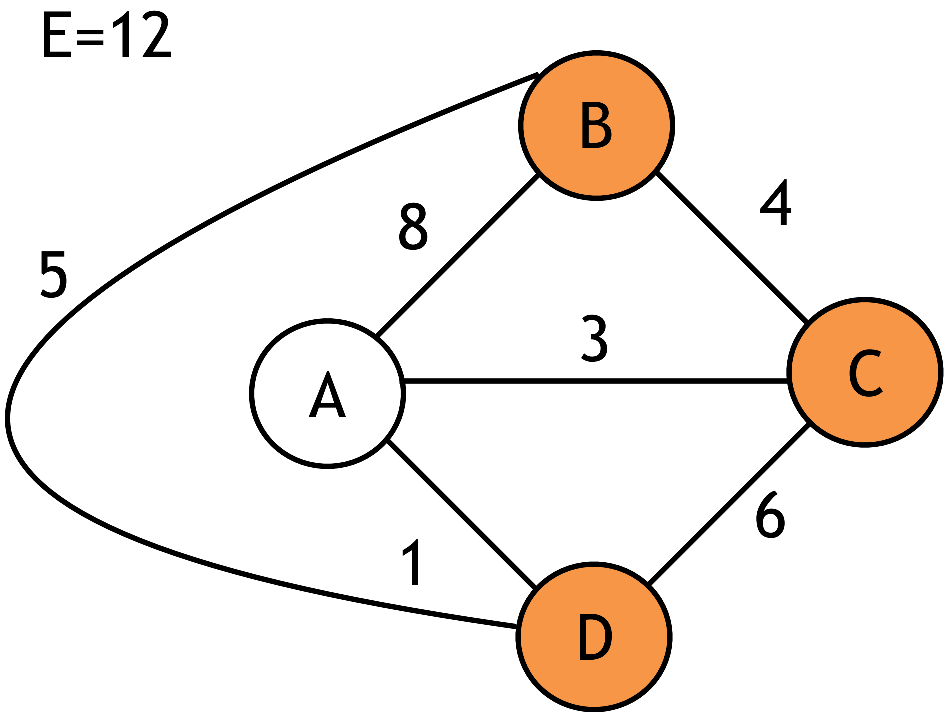
Example Problem: Max-Cut

- Given graph G , find a subset of graph G, S, such that the sum of the connection weights between the graphs are maximized



Example Problem: Max-Cut

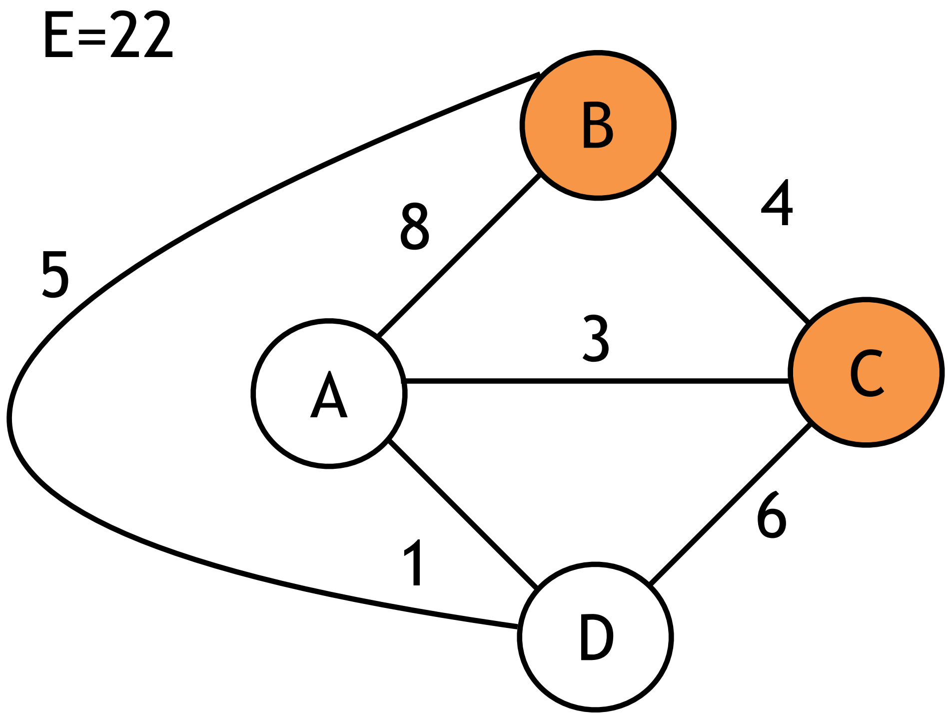
- Given graph G , find a subset of graph G, S, such that the sum of the connection weights between the graphs are maximized



Example Problem: Max-Cut

- Given graph G , find a subset of graph G, S, such that the sum of the connection weights between the graphs are maximized

$$S = \{B, C\}$$

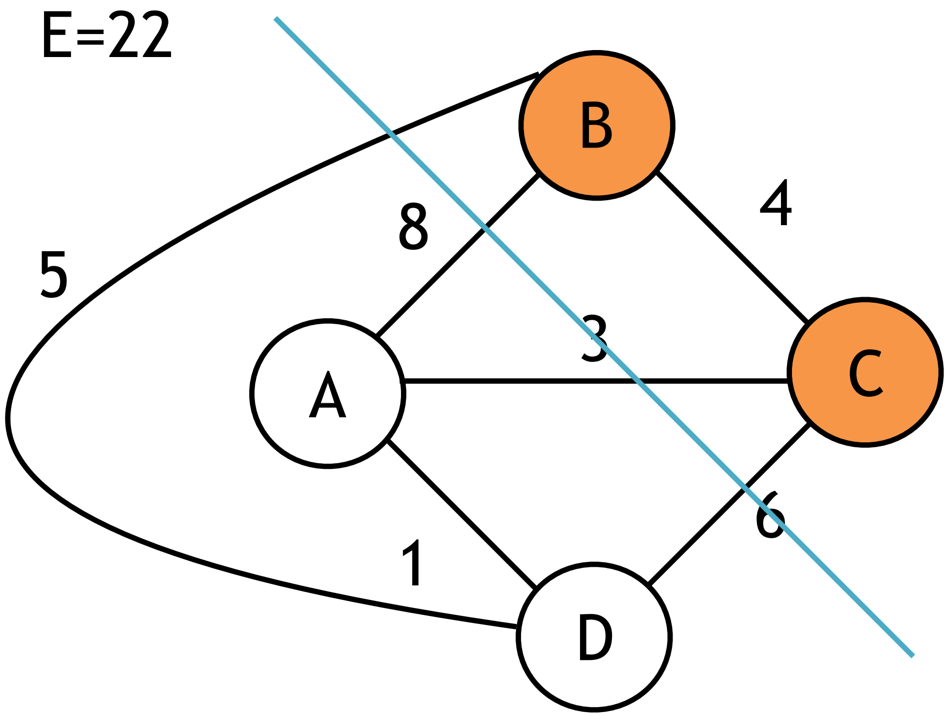


UPPSALA UNIVERSITET

Example Problem: Max-Cut

- Given graph G , find a subset of graph G, S, such that the sum of the connection weights between the graphs are maximized

$$S = \{B, C\}$$



Mapping

- In order to map all graphs to the Boltzmann Machine, we add an incidence matrix ***I***
- Change edge weights to allow proper training

UPPSALA UNIVERSITET

Mapping

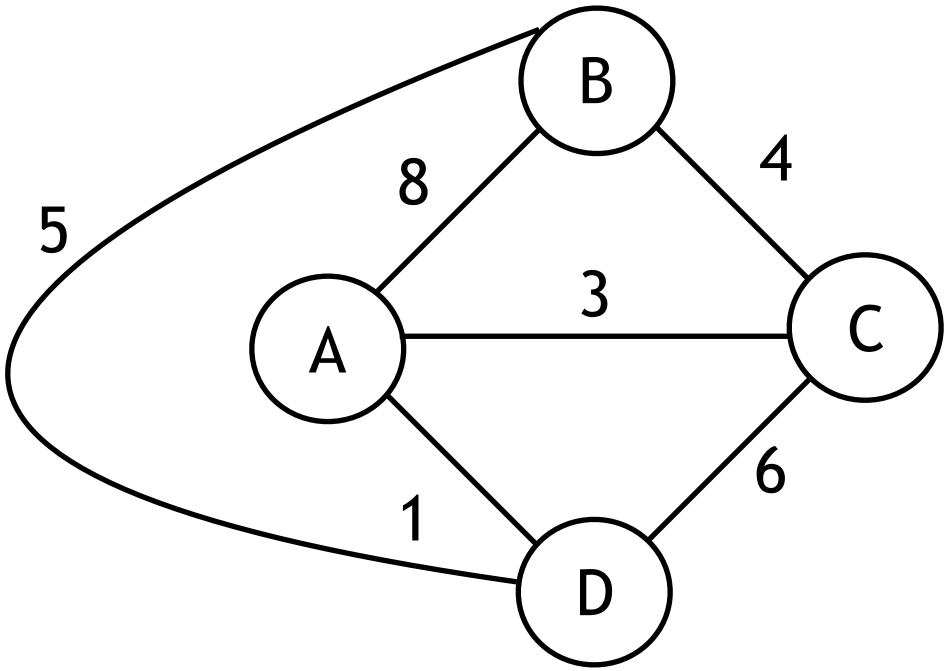
- In order to map all graphs to the Boltzmann Machine, we add an incidence matrix ***I***
- Change edge weights to allow proper training

$$w_{jj} = \sum_i d_{ij} \qquad w_{ji} = -2d_{ji}$$

Mapping

- In order to map all graphs to the Boltzmann Machine, we add an incidence matrix ***I***
- Change edge weights to allow proper training

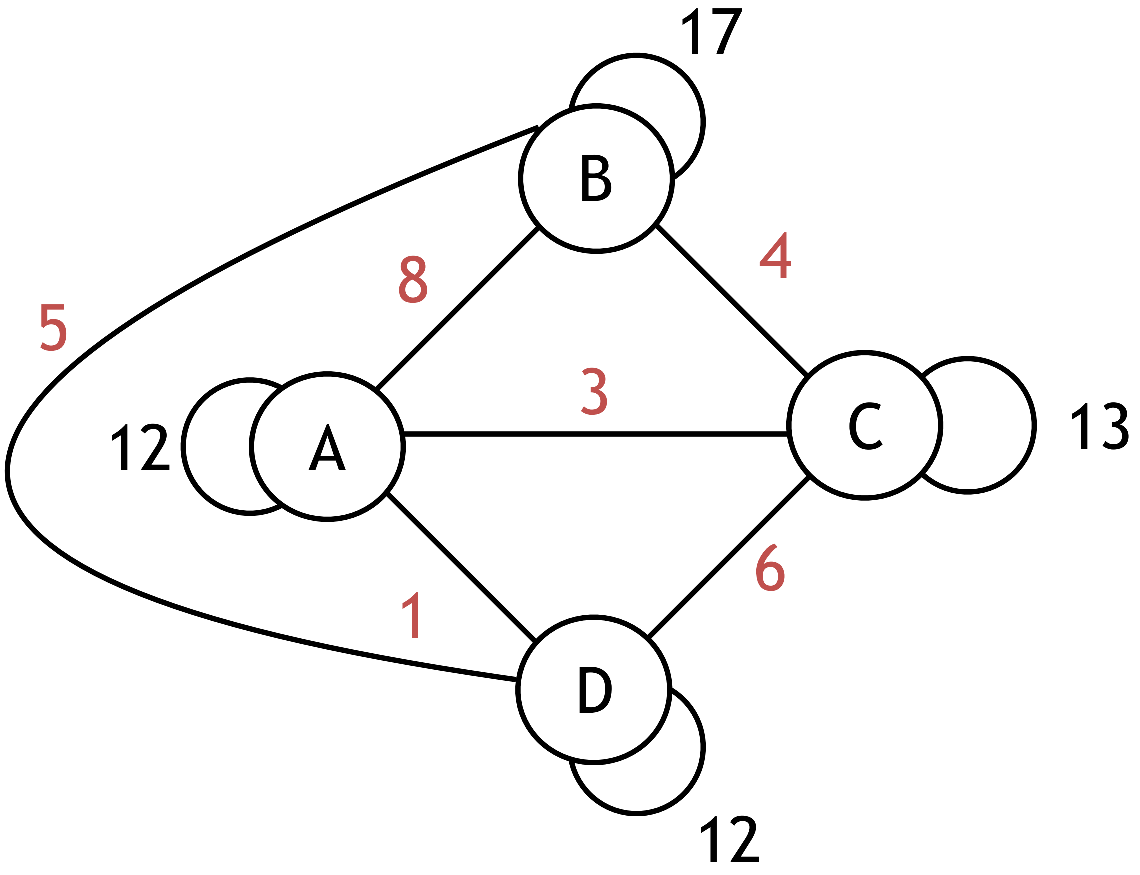
$$w_{jj} = \sum_i d_{ij} \qquad w_{ji} = -2d_{ji}$$



Mapping

- In order to map all graphs to the Boltzmann Machine, we add an incidence matrix ***I***
- Change edge weights to allow proper training

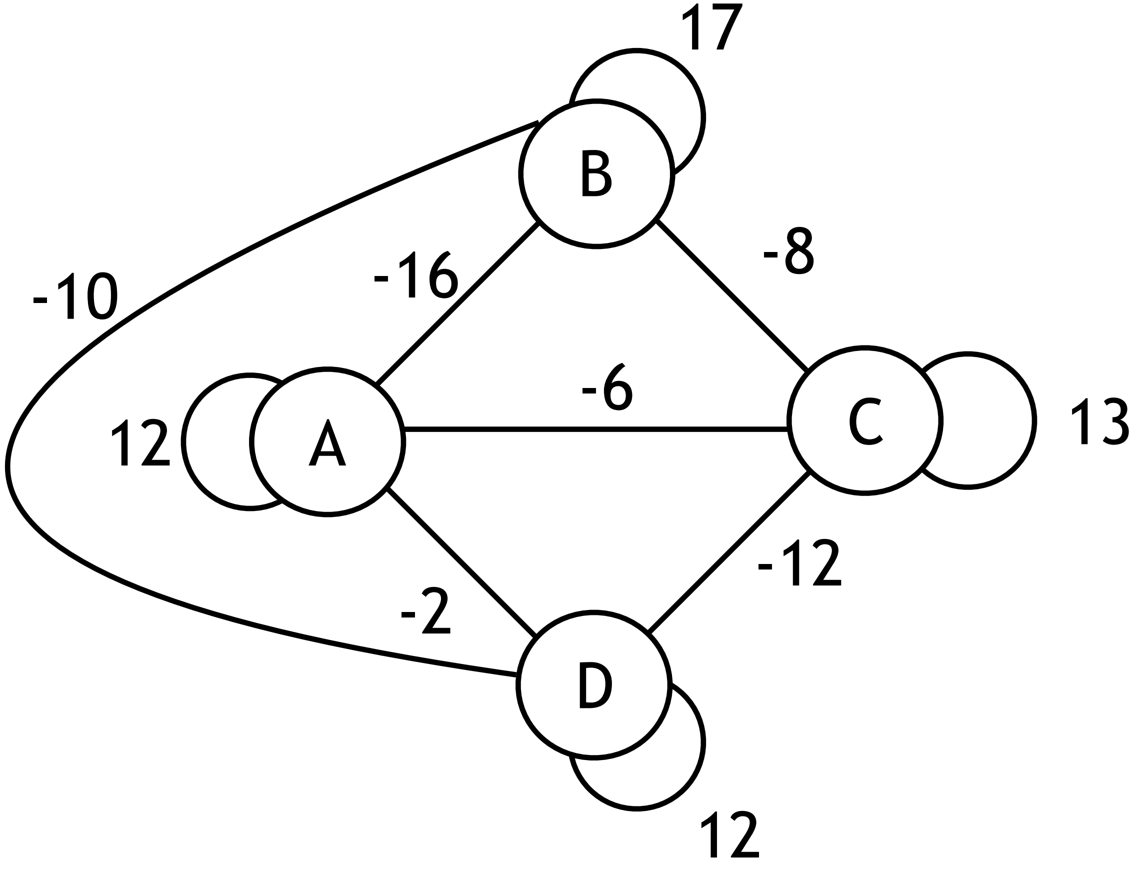
$$w_{jj} = \sum_{i} d_{ij} \qquad w_{ji} = -2d_{ji}$$



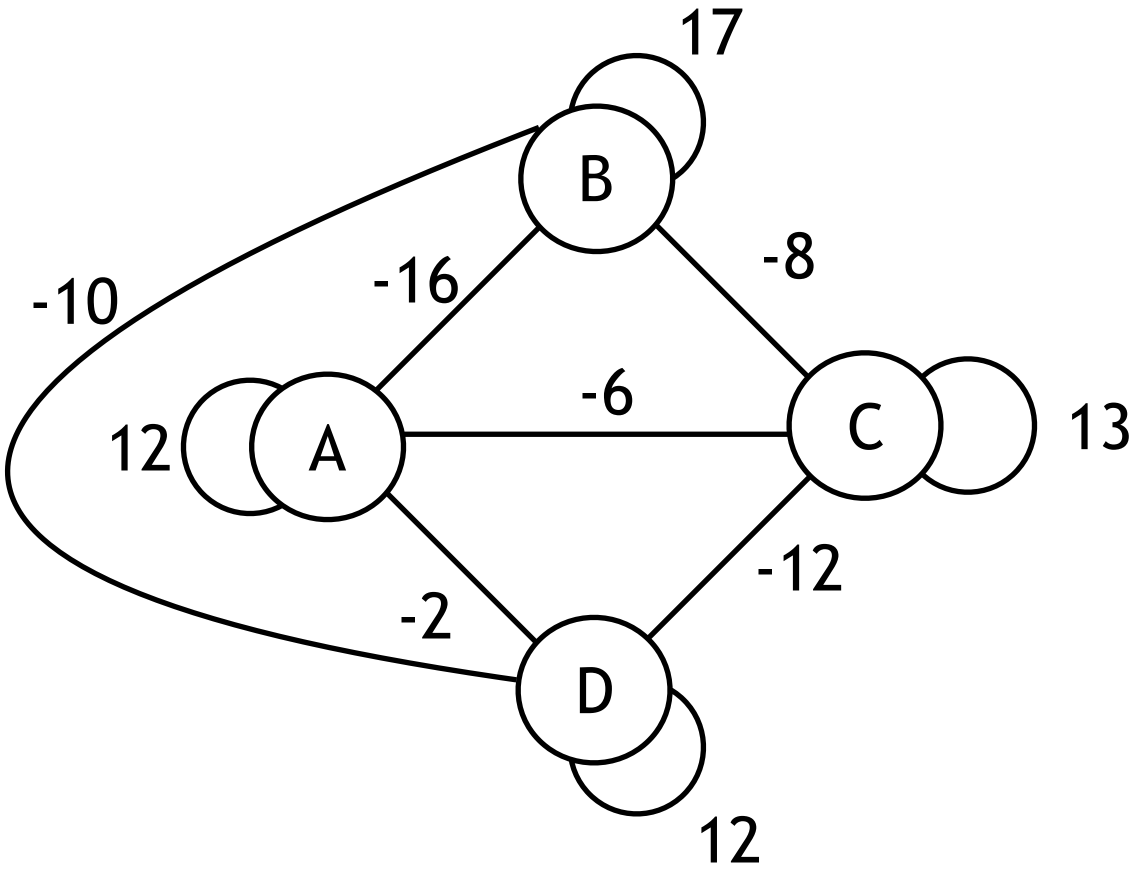
Mapping

- In order to map all graphs to the Boltzmann Machine, we add an incidence matrix ***I***
- Change edge weights to allow proper training

$$w_{jj} = \sum_i d_{ij} \qquad w_{ji} = -2d_{ji}$$

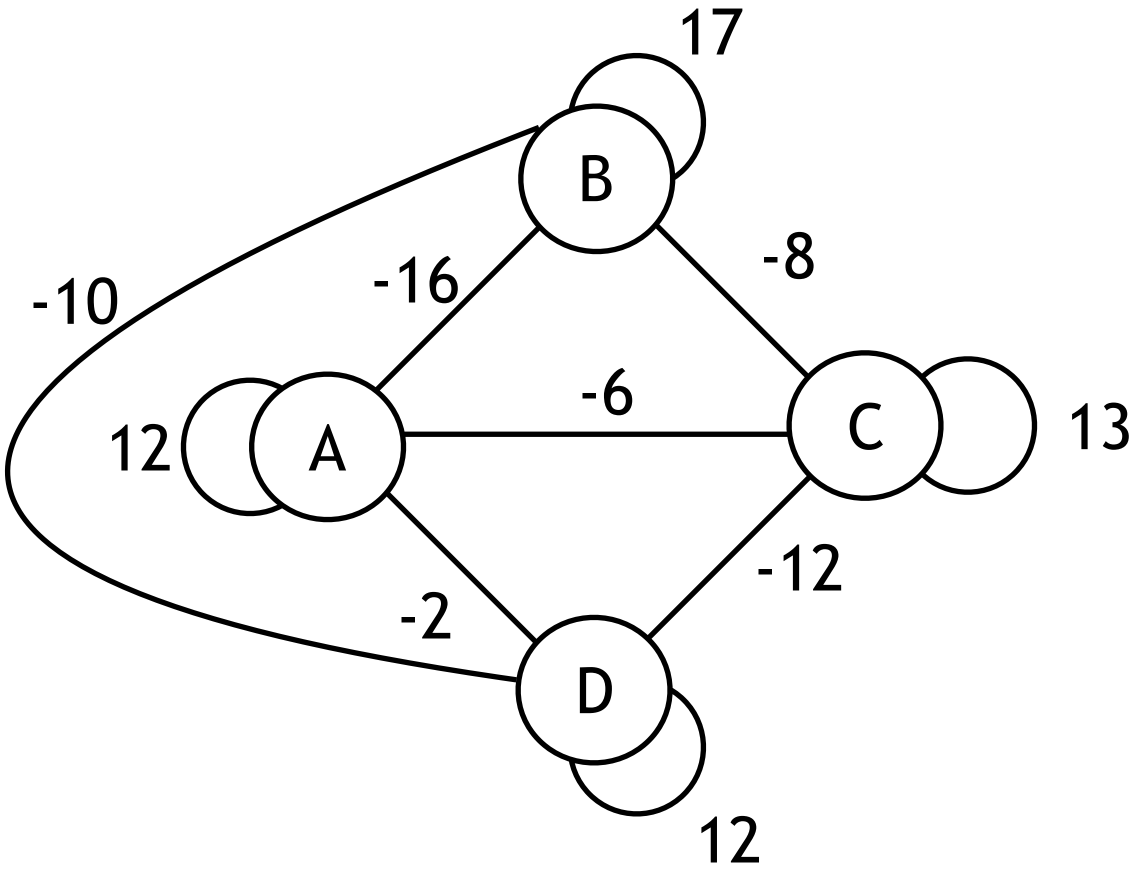


Mapping



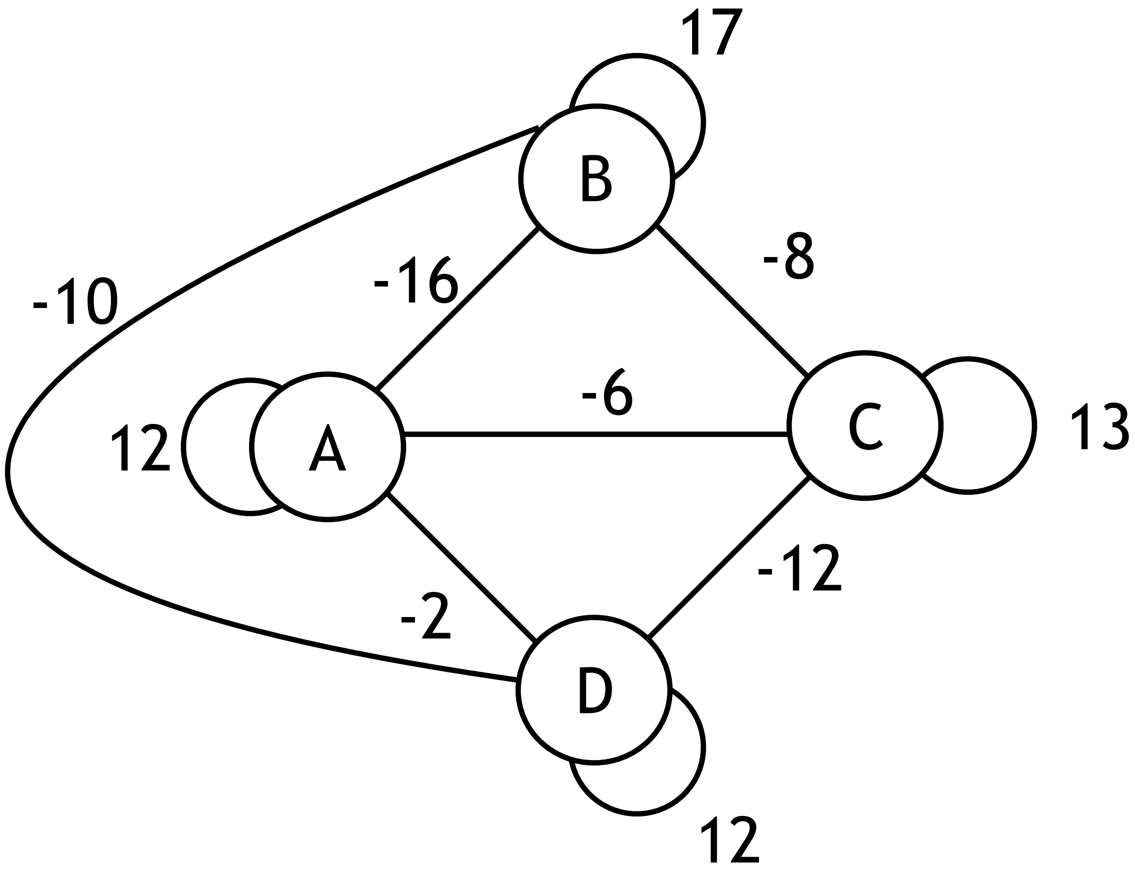
Mapping

- The graph is mapped to a Boltzmann Machine but what's the task now?



Mapping

- The graph is mapped to a Boltzmann Machine but what's the task now?
- Minimize "Energy"

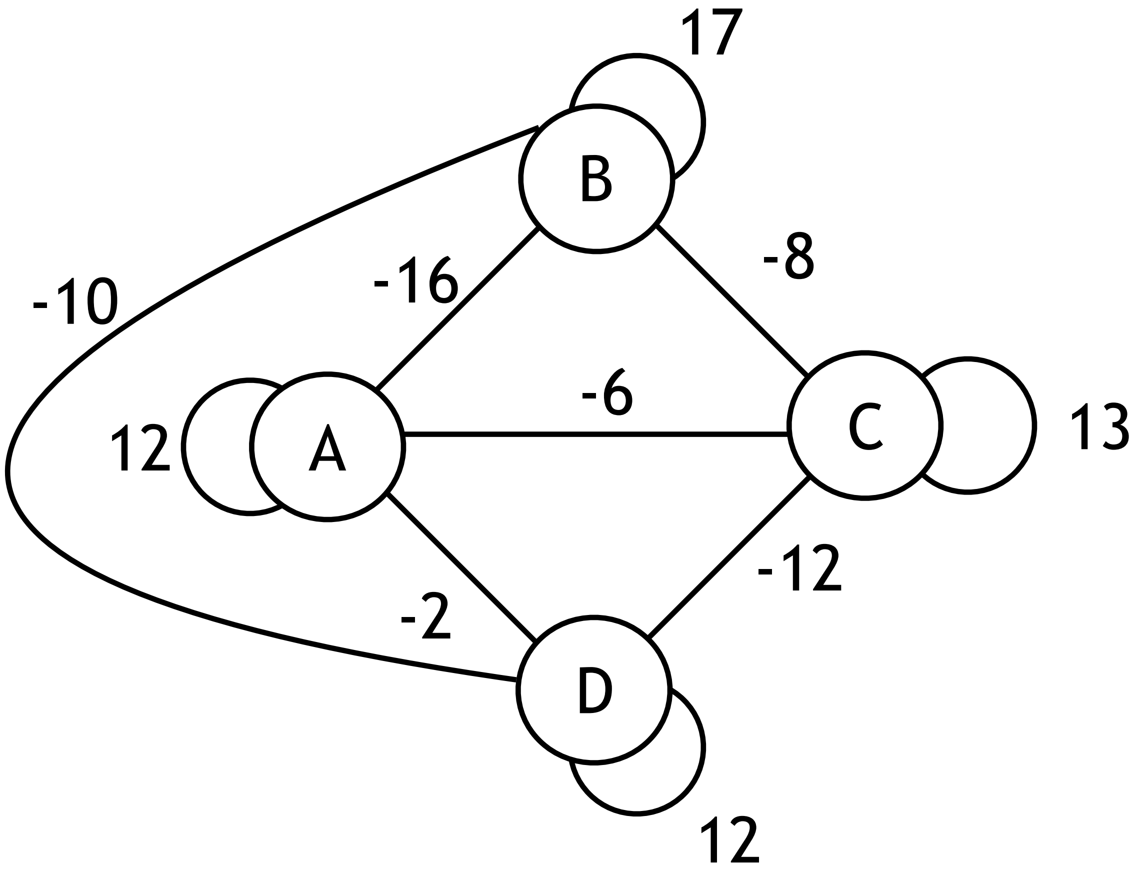


UPPSALA UNIVERSITET

Mapping

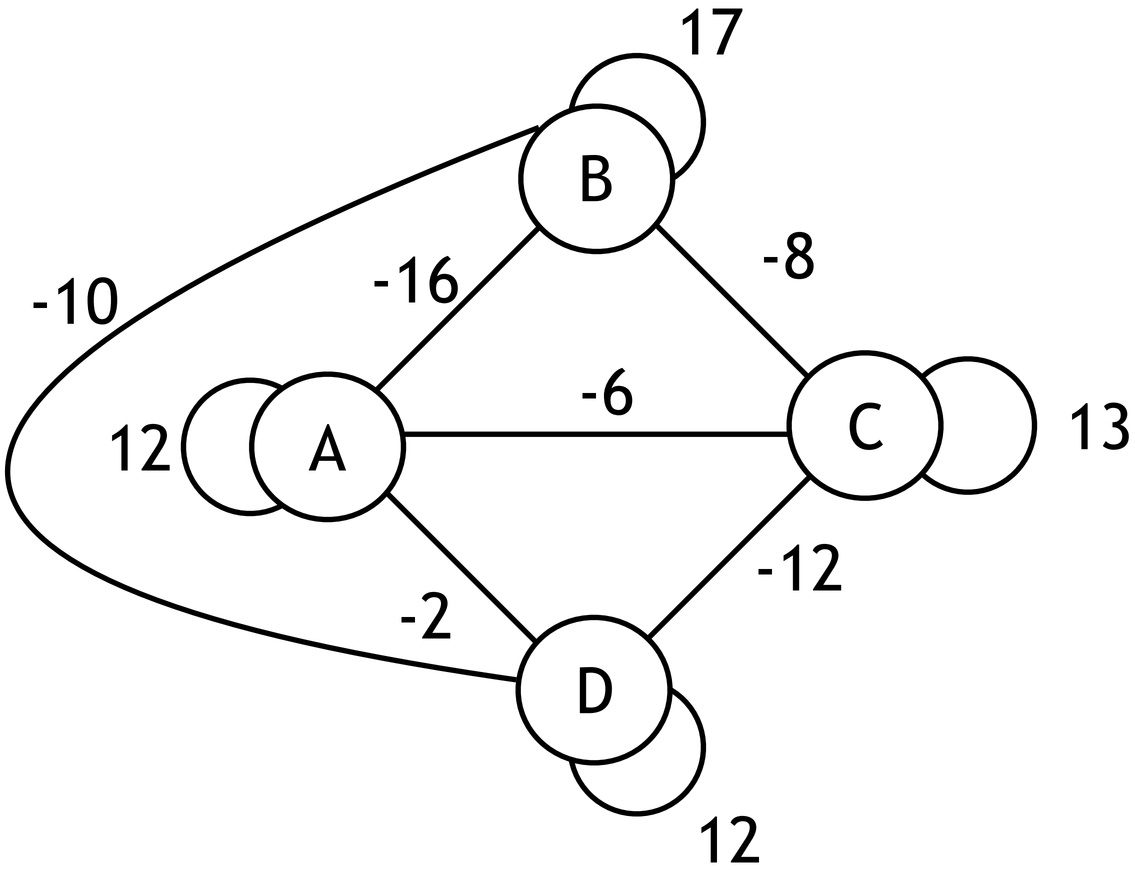
- The graph is mapped to a Boltzmann Machine but what's the task now?
- Minimize "Energy"

$$E(X) = -\frac{1}{2}\sum_{j}\sum_{i\neq j} x_j x_i w_{ij} - \sum_{j} x_j w_{jj}$$



Example

$$E(X) = -\frac{1}{2}\sum_j \sum_{i \neq j} x_j x_i w_{ij} - \sum_j x_j w_{jj}$$

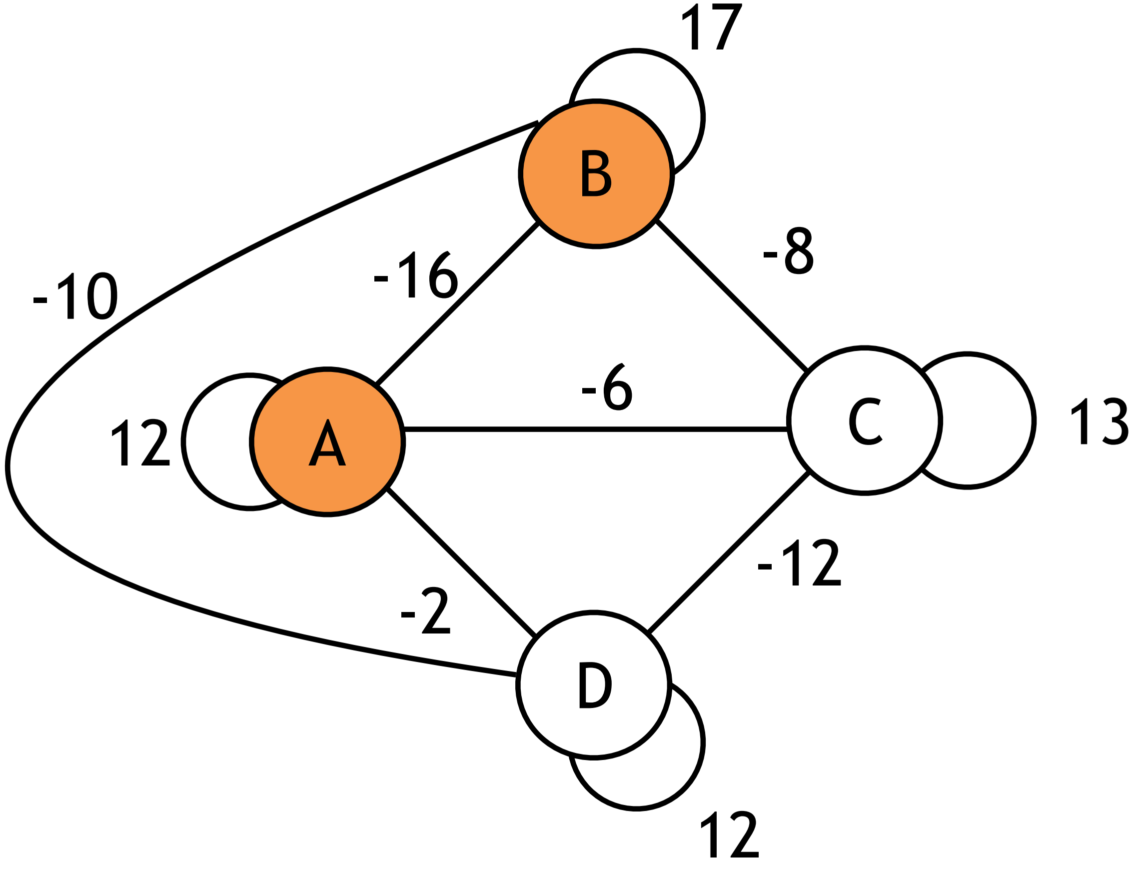


Example

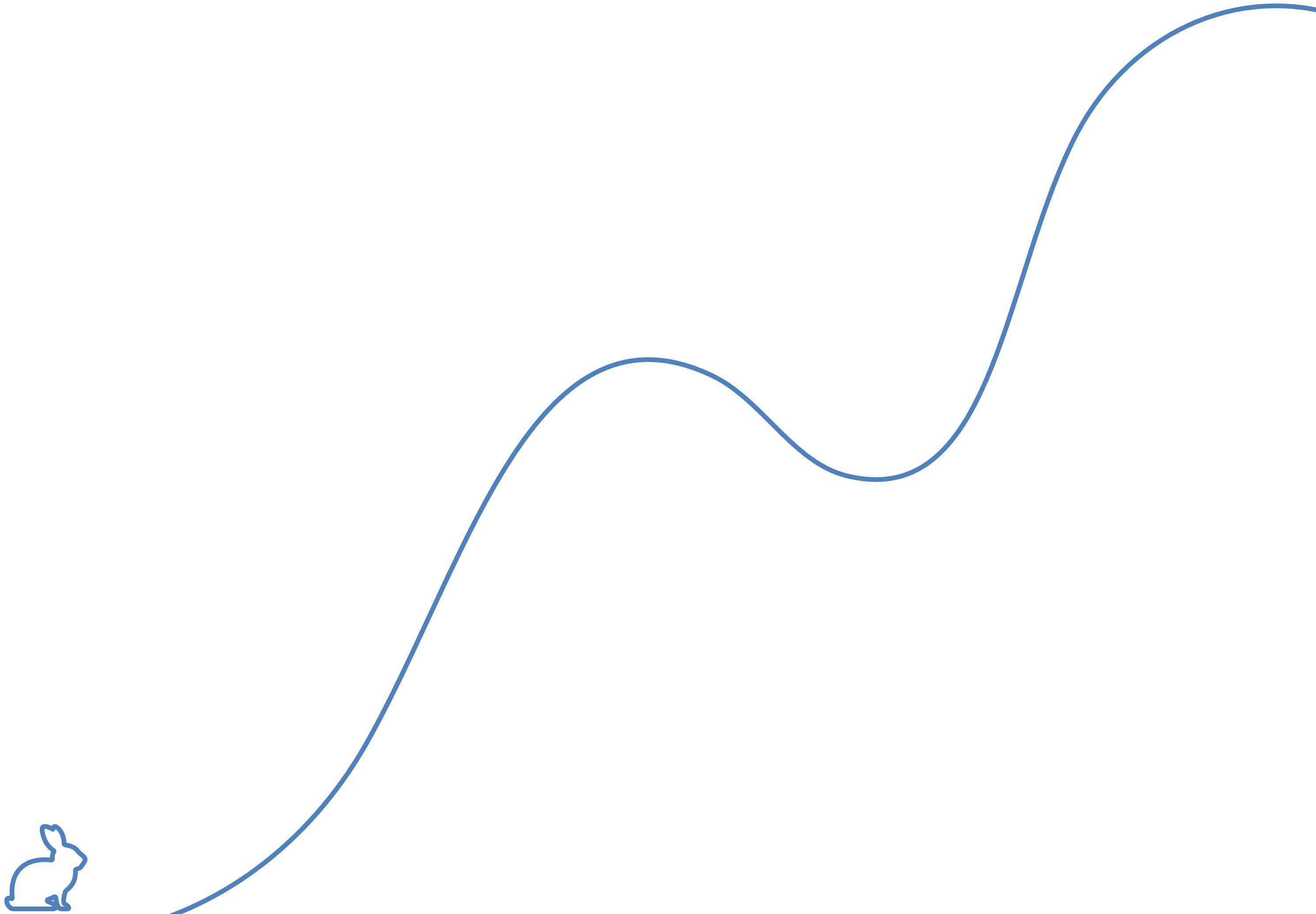
$$E(X) = -\frac{1}{2}\sum_{j}\sum_{i\neq j} x_j x_i w_{ij} - \sum_{j} x_j w_{jj}$$

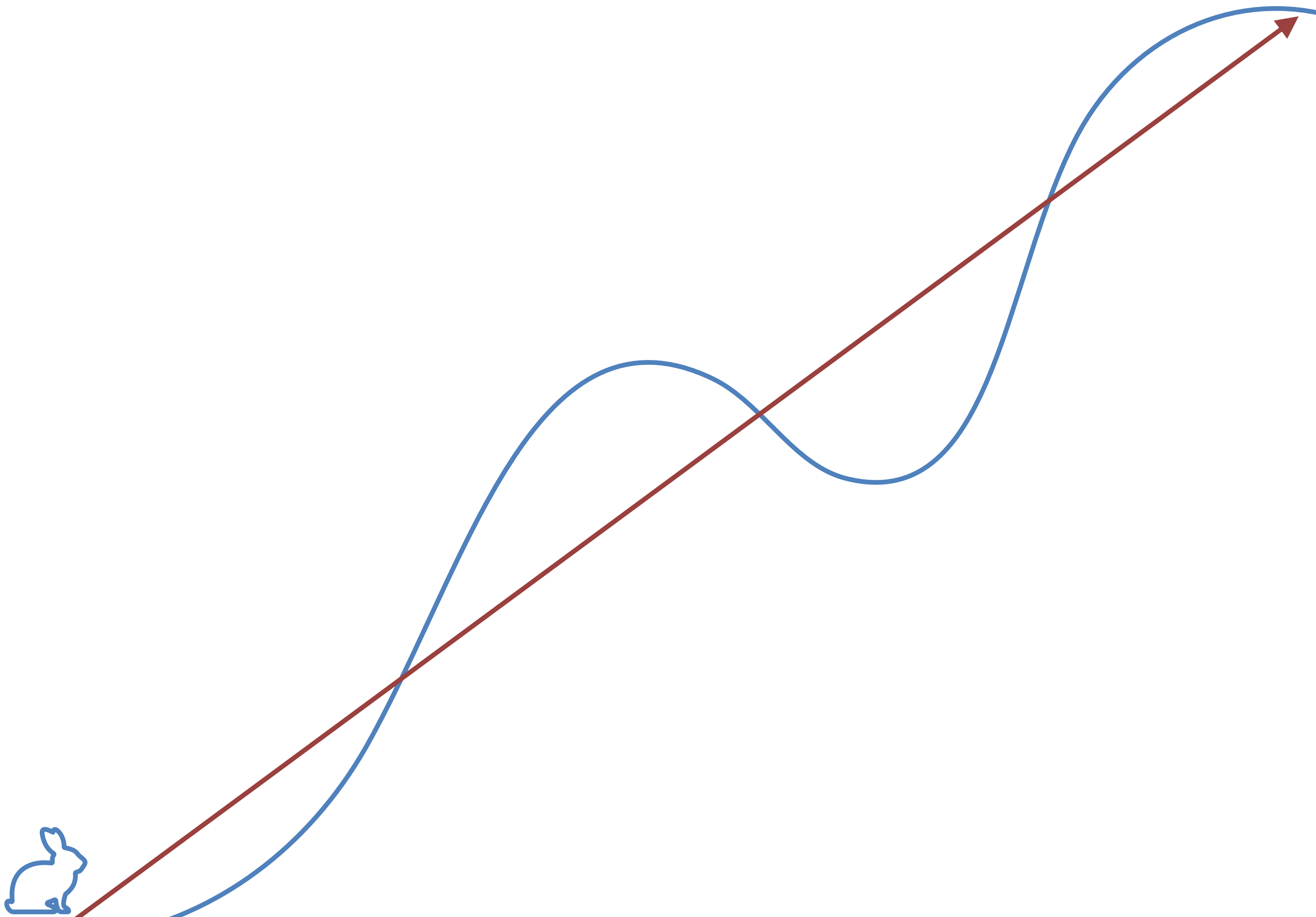
$$E(X) = -\frac{1}{2}\cdot(-16 + -16) - (12 + 17)$$

$$= 16 - 29 = -13$$

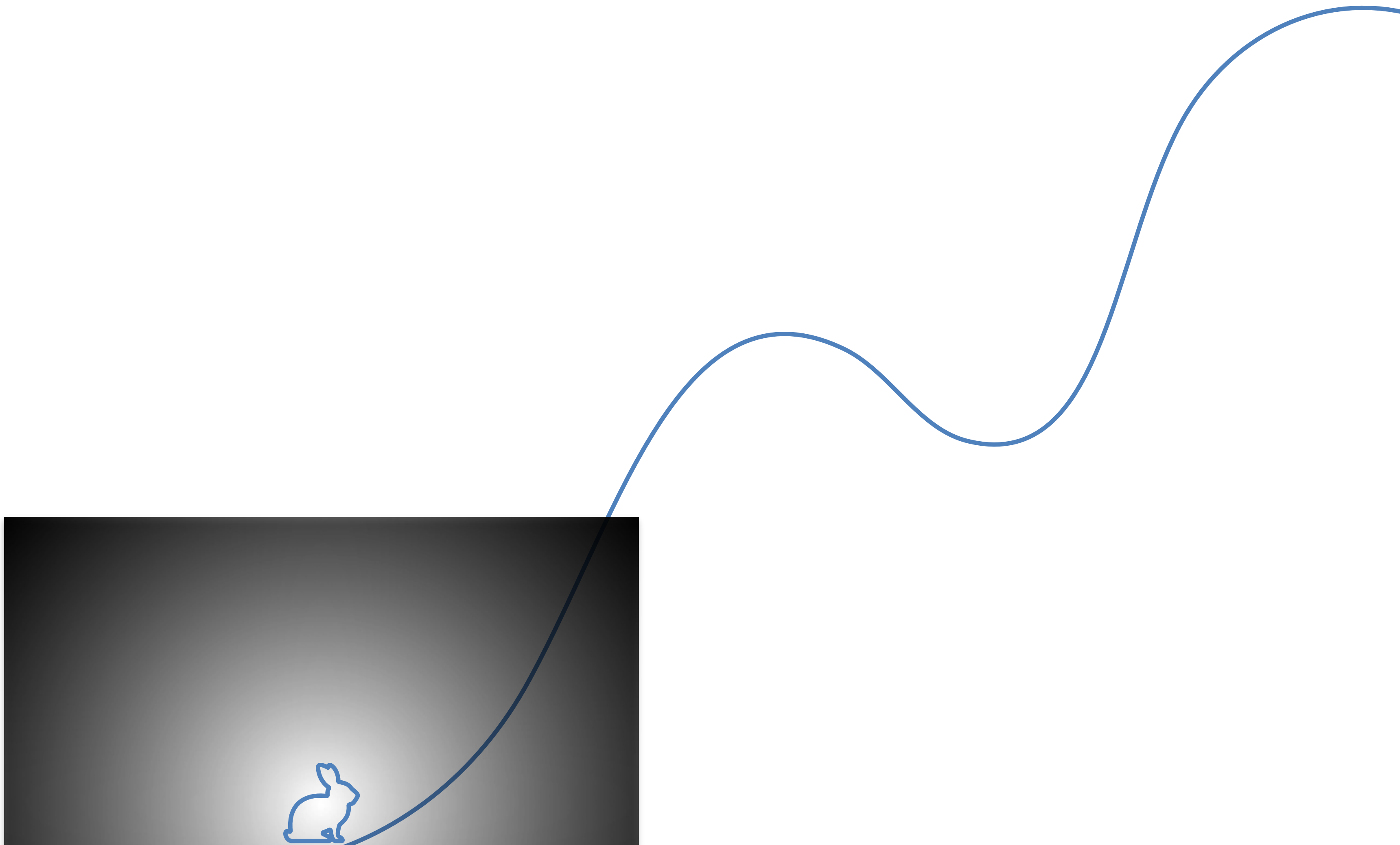


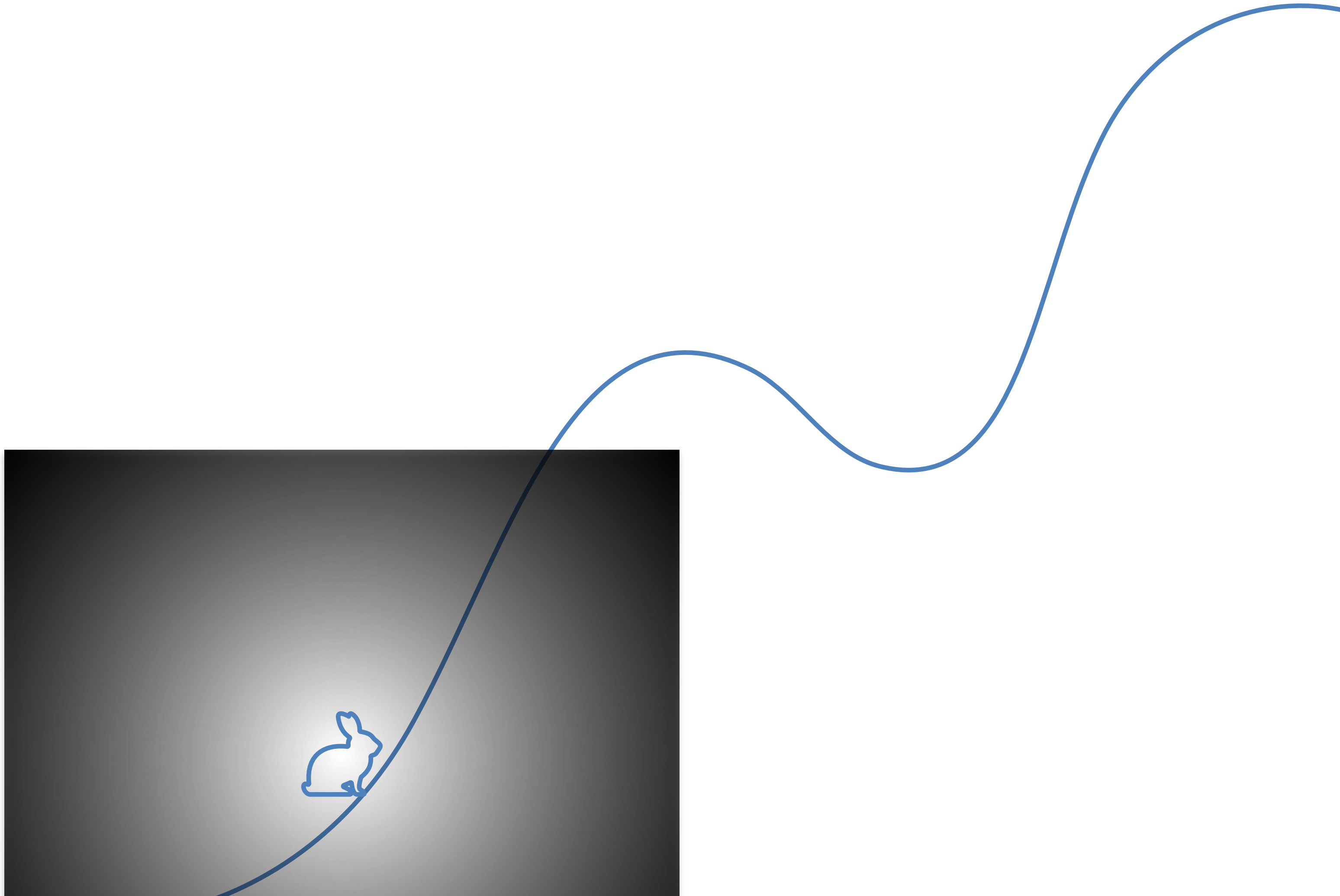
Simulated Annealing



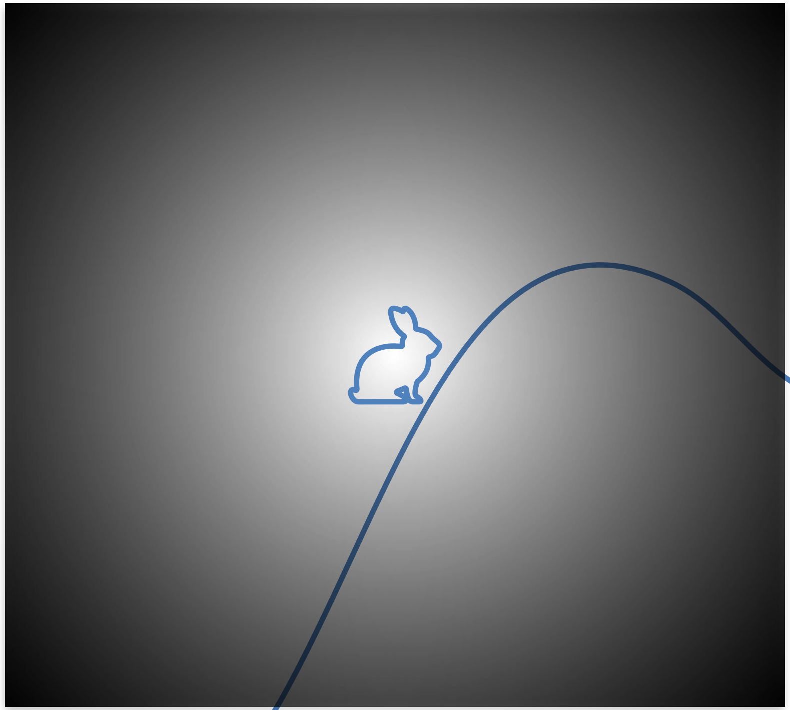


UPPSALA
UNIVERSITET

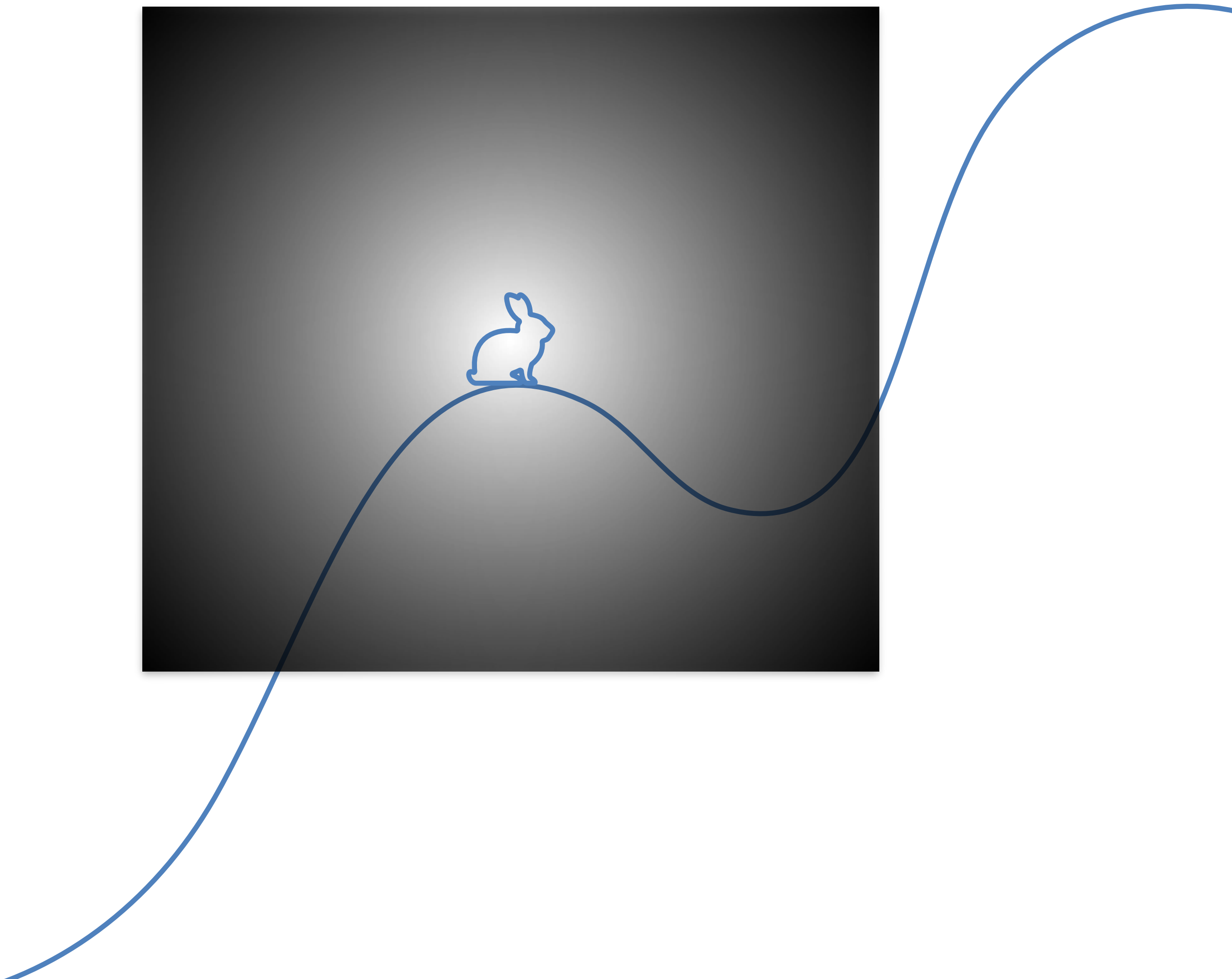


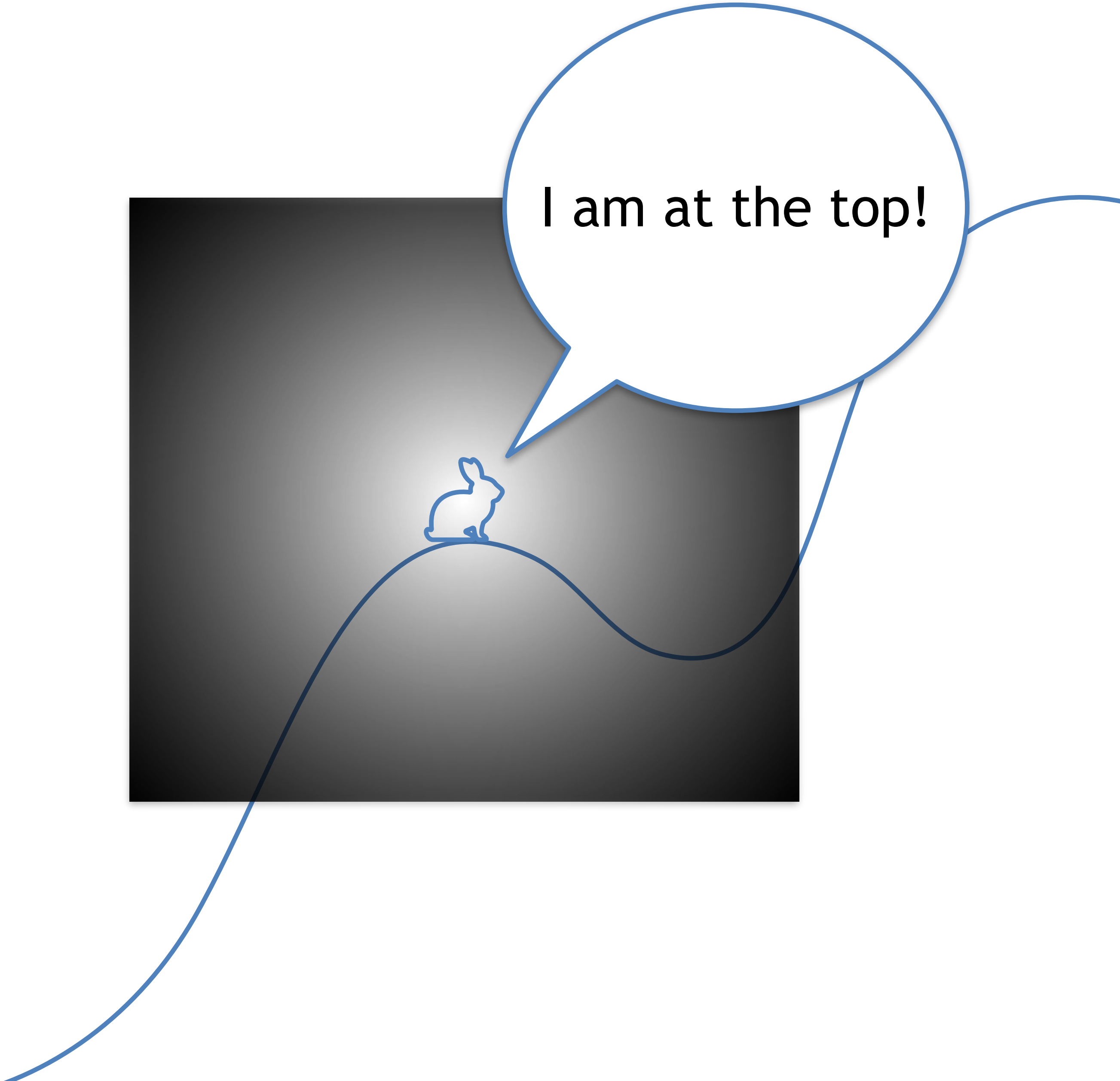


UPPSALA
UNIVERSITET

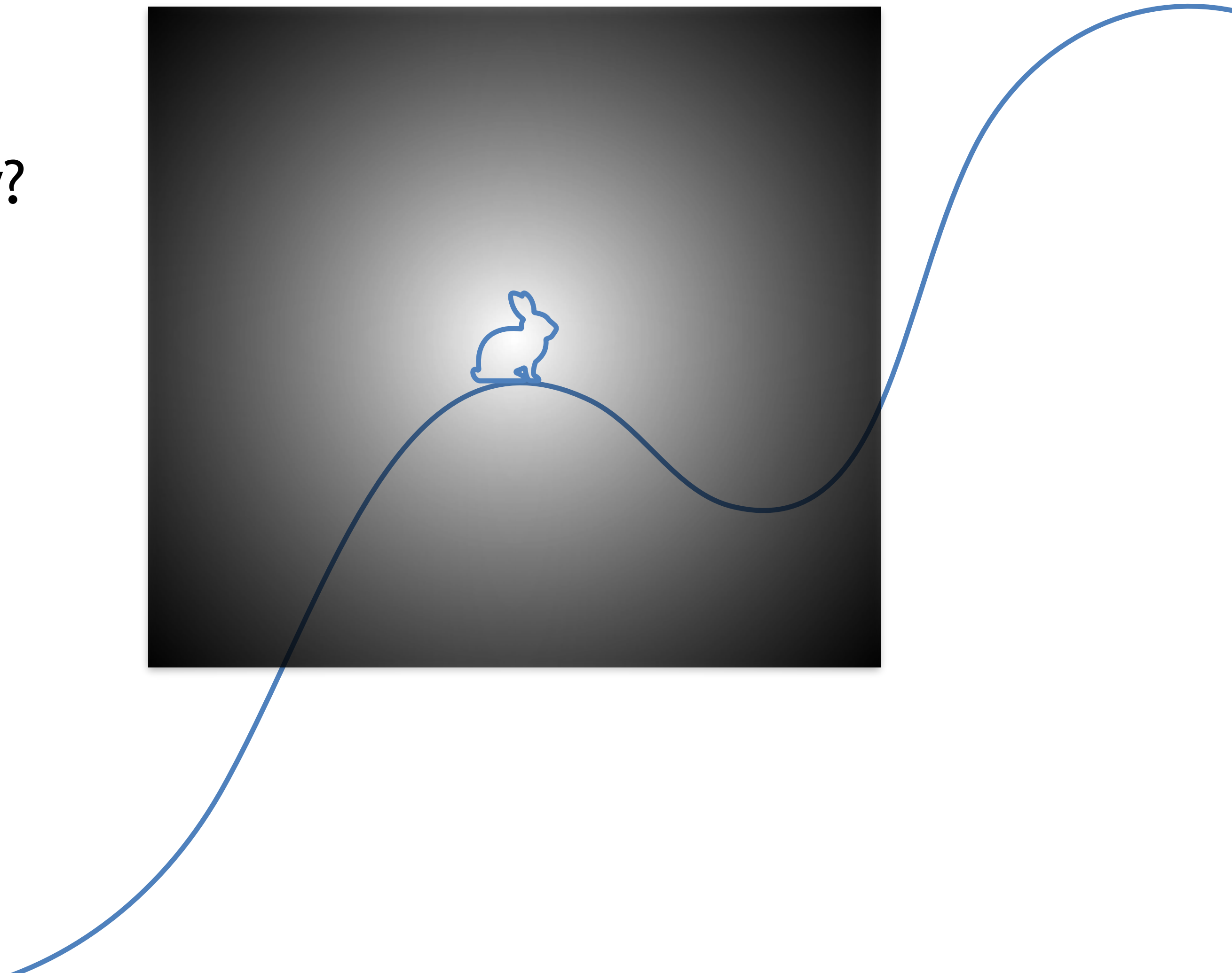


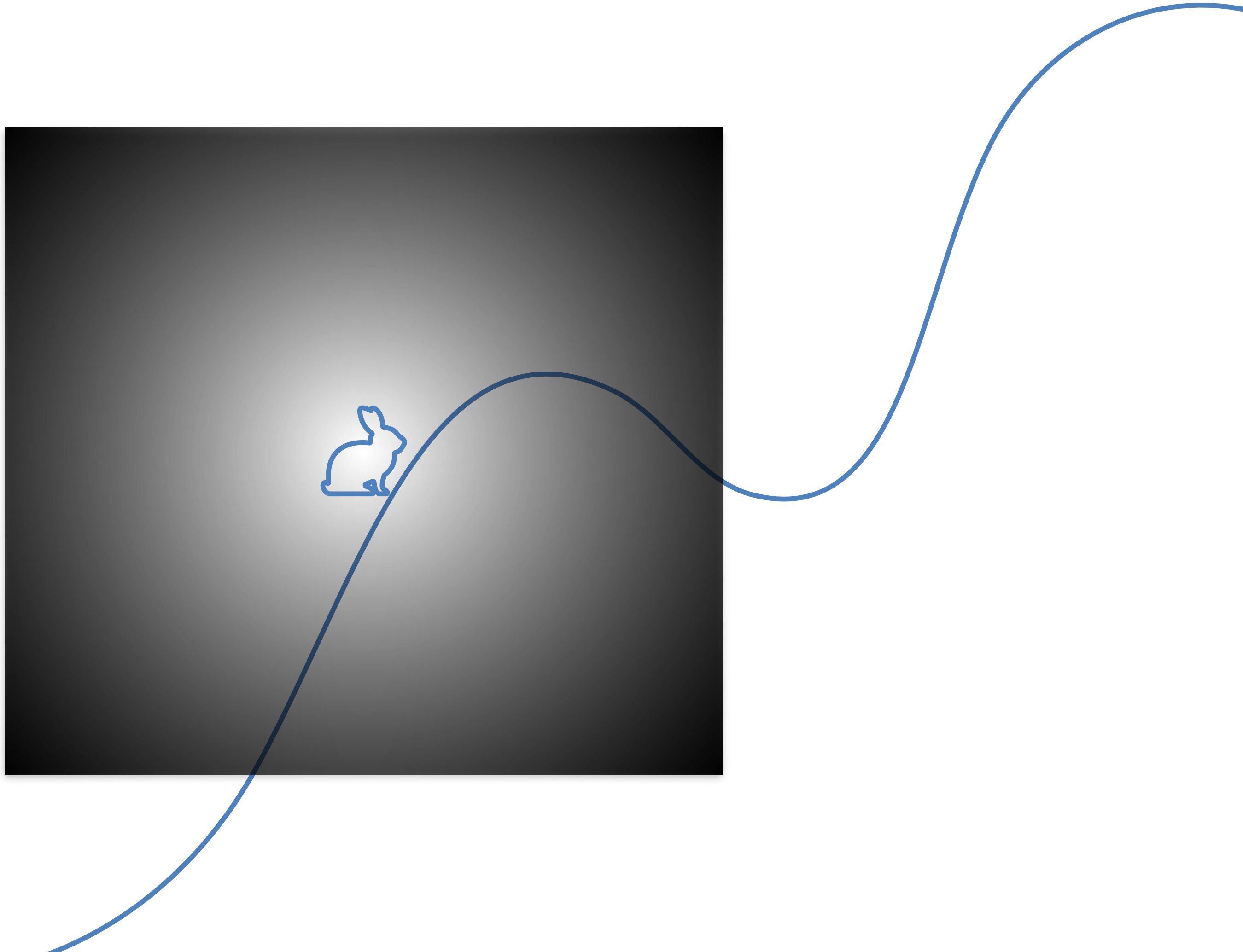
UPPSALA
UNIVERSITET



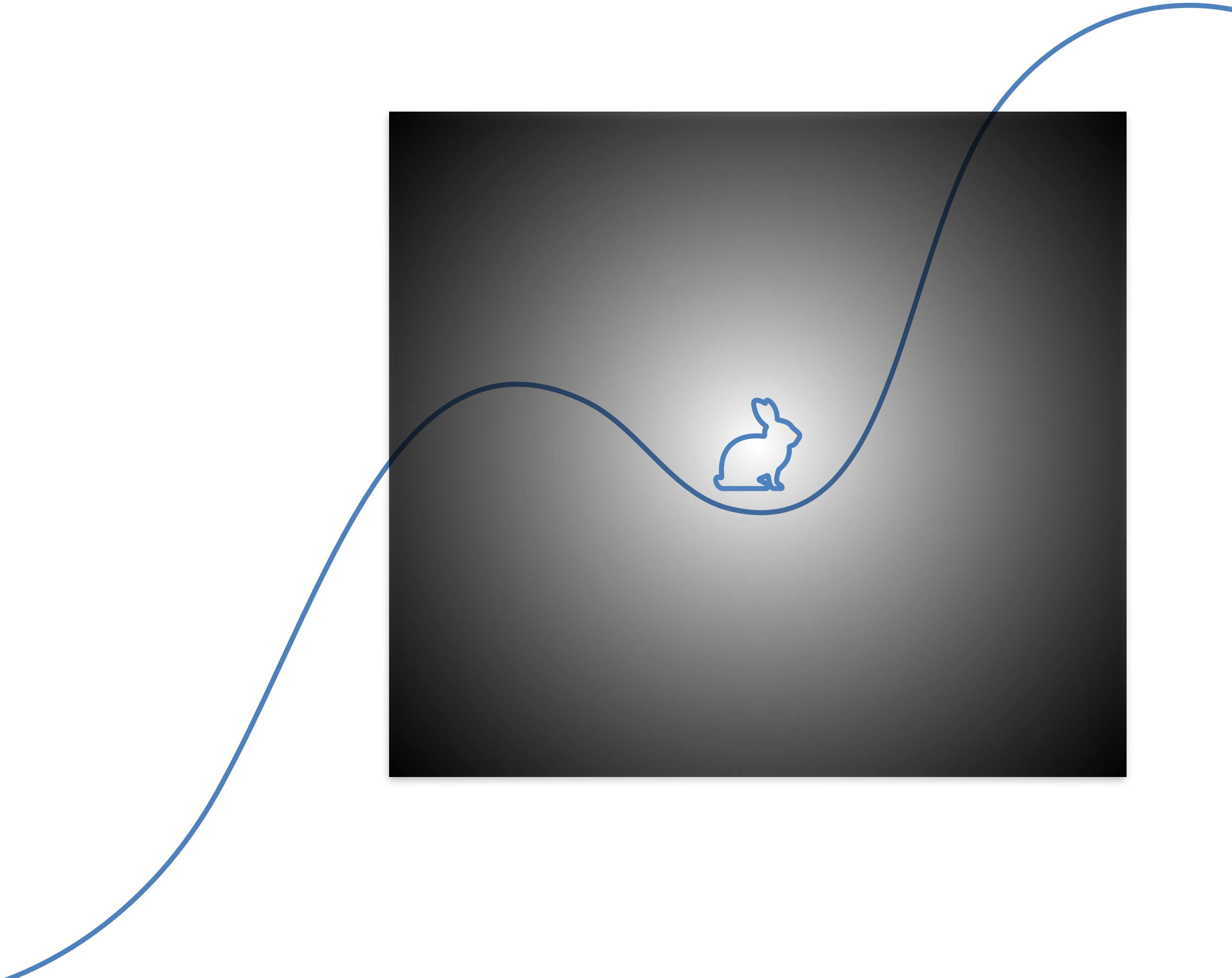

I am at the top!

What if the rabbit is feeling risky?

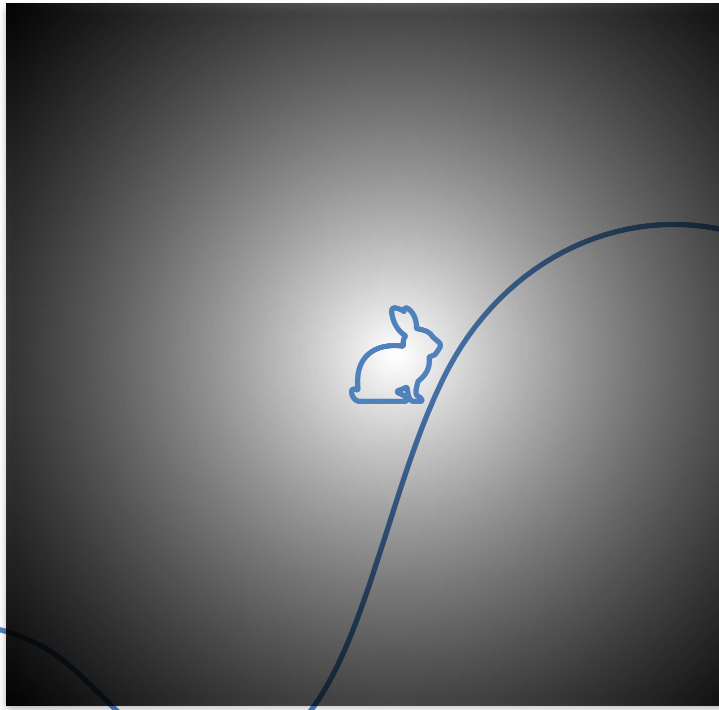




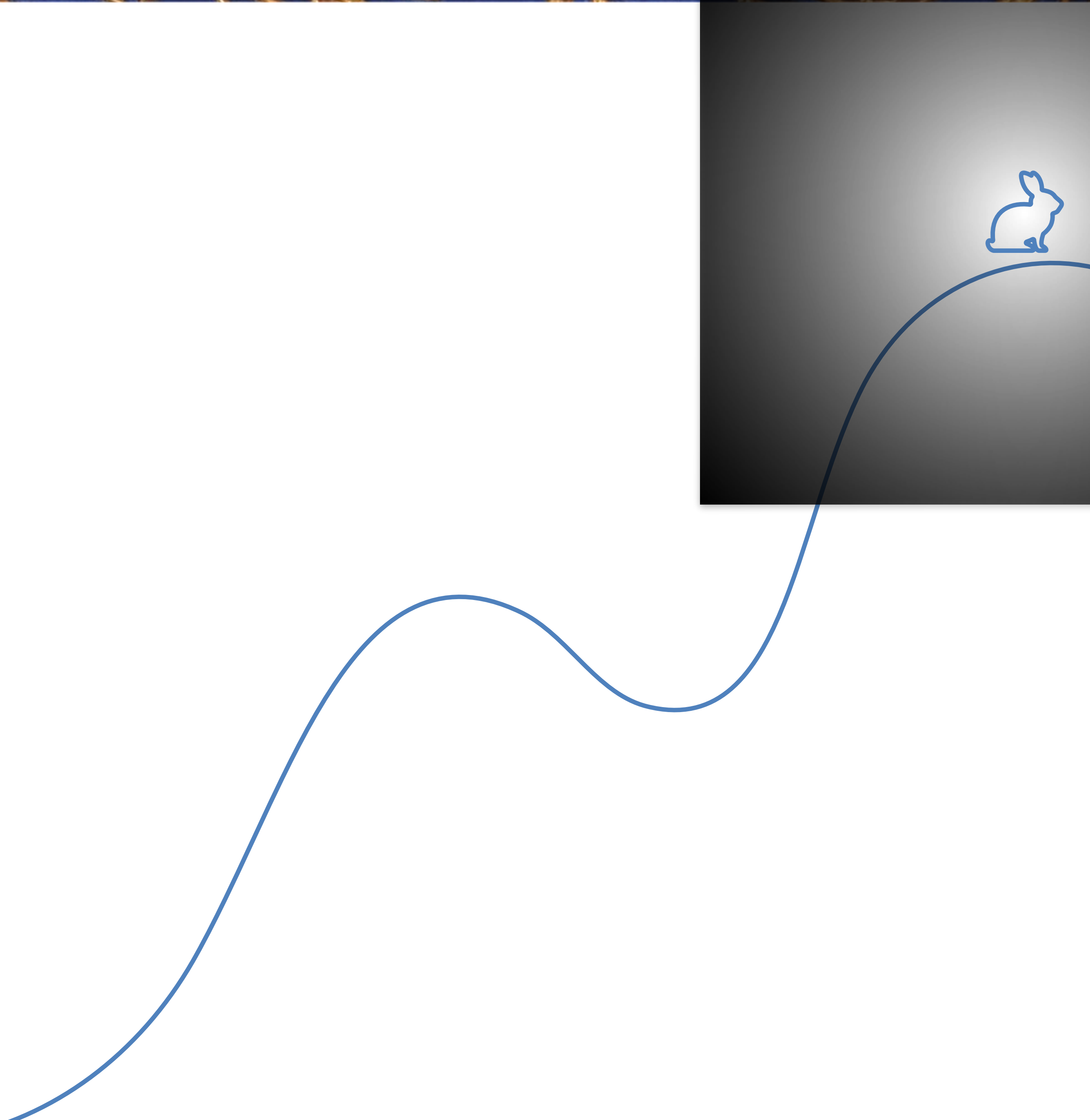
UPPSALA
UNIVERSITET



UPPSALA
UNIVERSITET



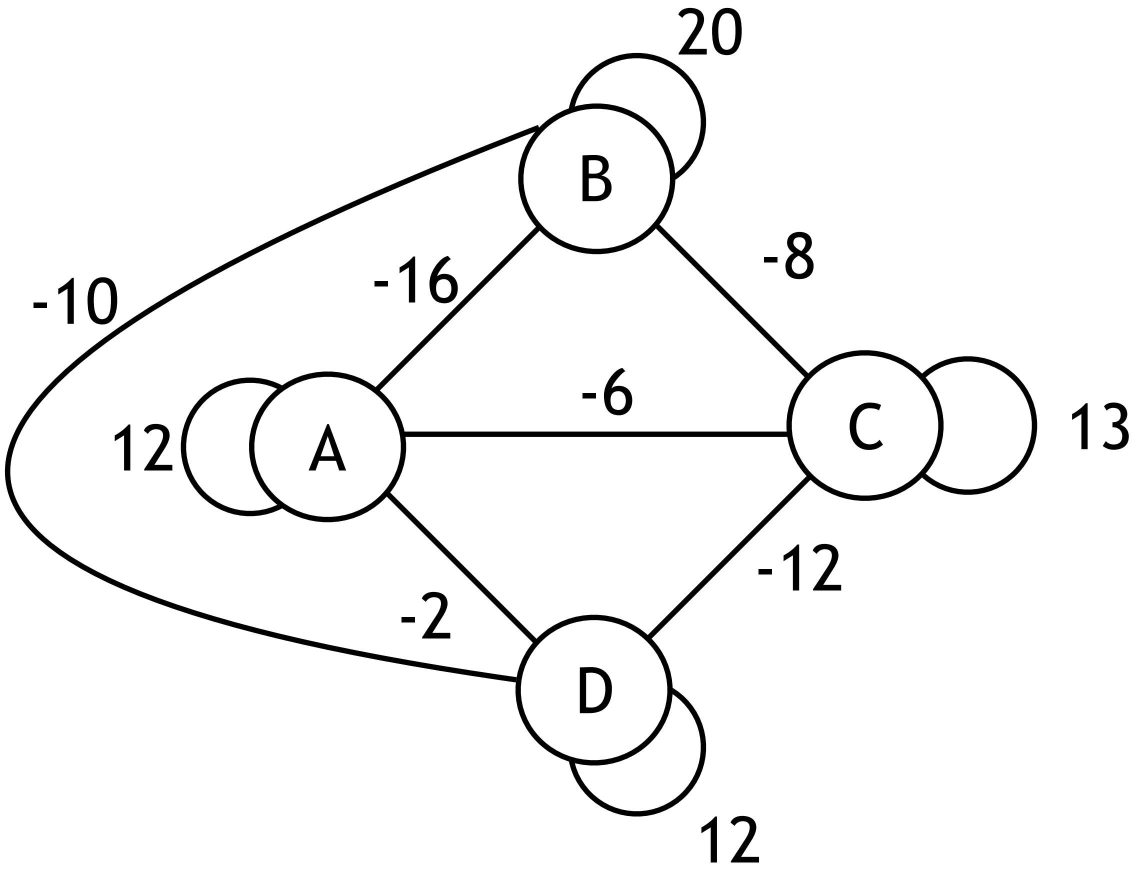
UPPSALA
UNIVERSITET



Learning the Solution

- Stochastically learn solution over time

$$\Delta E = (2x_j - 1)(\sum_{i \neq j} x_i w_{ji} + w_{jj})$$



Learning the Solution

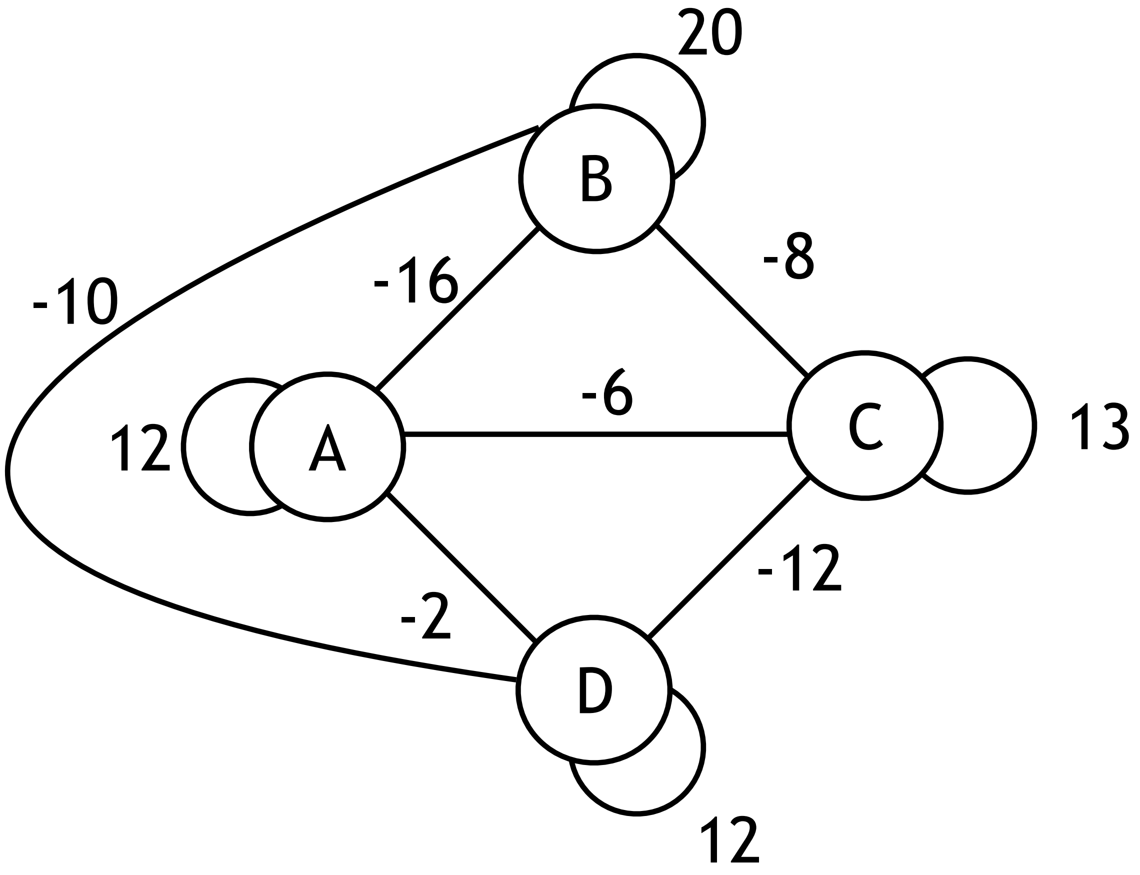
- Stochastically learn solution over time

$$\Delta E = (2x_j - 1)(\sum_{i \neq j} x_i w_{ji} + w_{jj})$$

$$x^j = < x_0, x_1, \ldots, \neg x_j, \ldots, x_n >$$

$$P(x^j | x) = \frac{1}{1 + e^{\frac{\Delta E}{C}}}$$

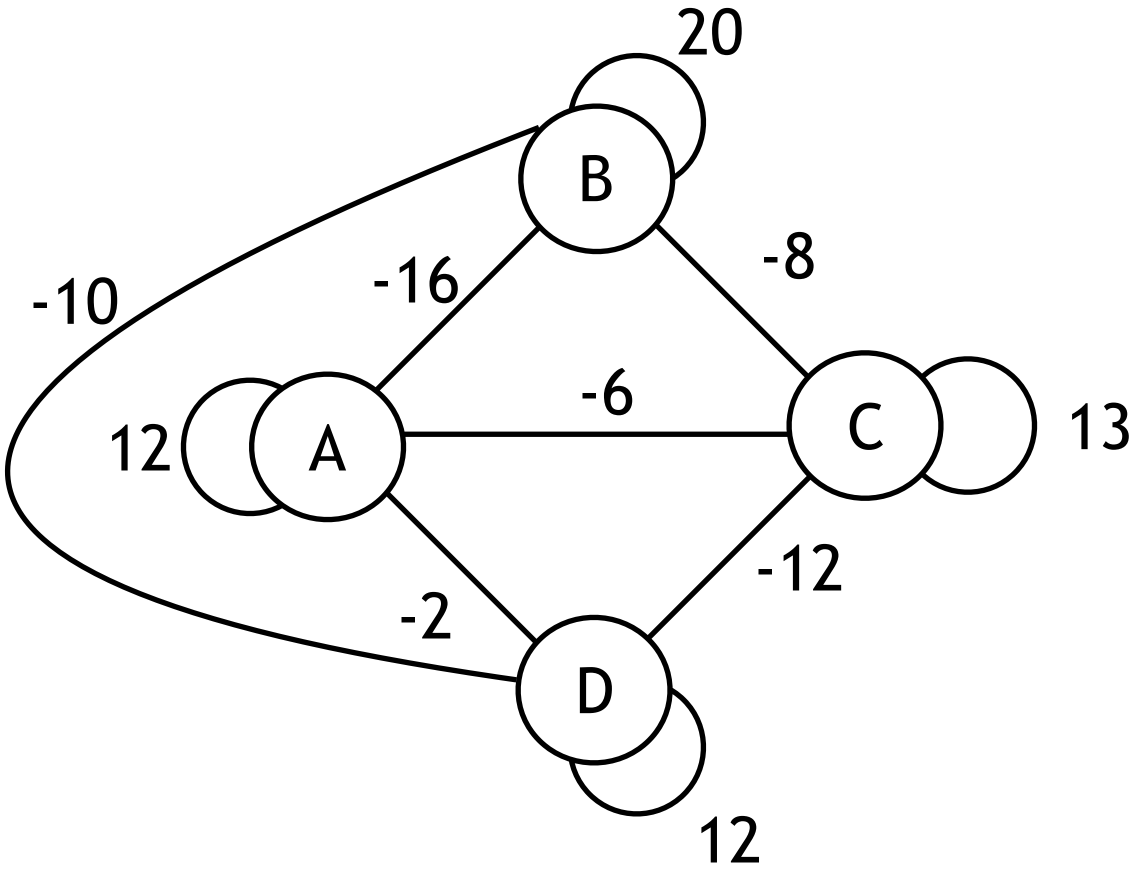
- C allows for probability control of suboptimal state changes



Learning the Solution

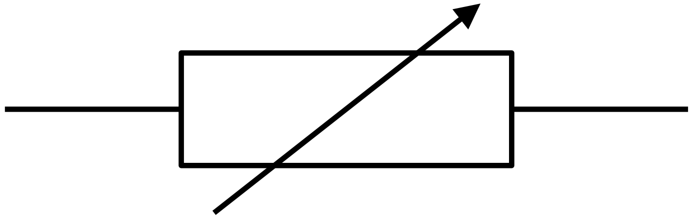
$$P(x^j \mid x) = \frac{1}{1 + e^{\frac{\Delta E}{C}}}$$

- The change in E can be continuously calculated *in situ* for each node
- C allows for simulated annealing, eventually *probably* resulting in a optimized graph



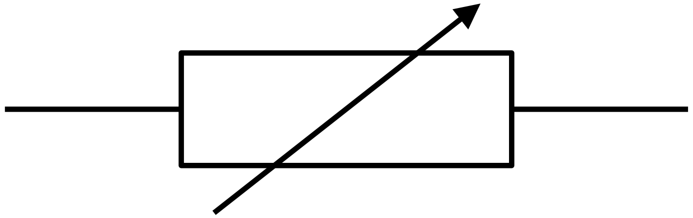
Memristors

- A component with variable conductance



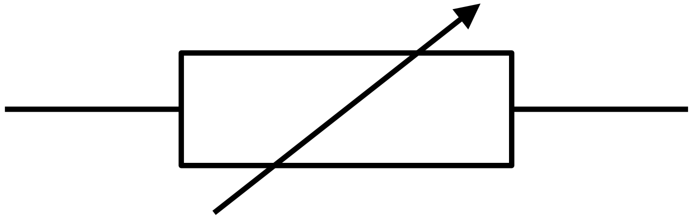
Memristors

- A component with variable conductance
- Recall $V = I \cdot R$



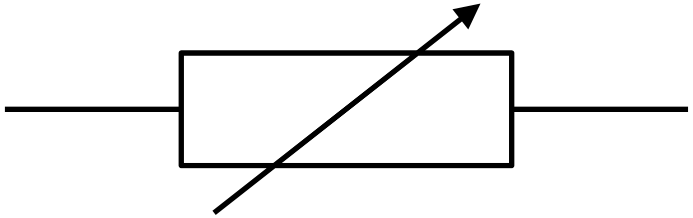
Memristors

- A component with variable conductance
- Recall $V = I \cdot R$
- Used to perform multiplications with the assistance of hardware to detect current



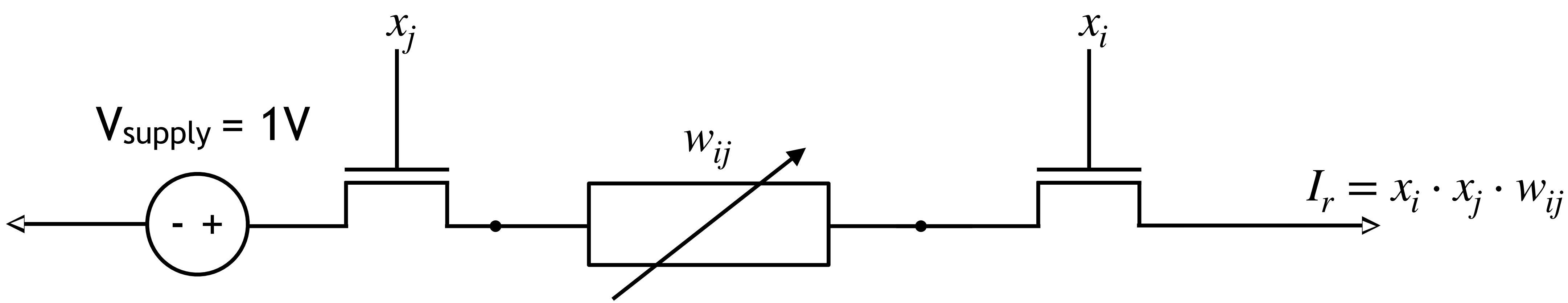
Memristors

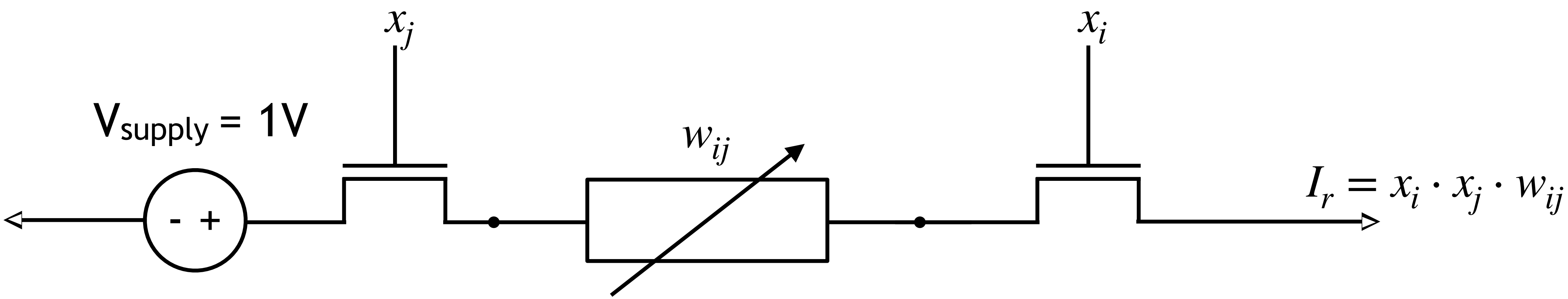
- A component with variable conductance
- Recall $V = I \cdot R$
- Used to perform multiplications with the assistance of hardware to detect current
- Components of RRAM cells - memory with density of FLASH but DRAM speed

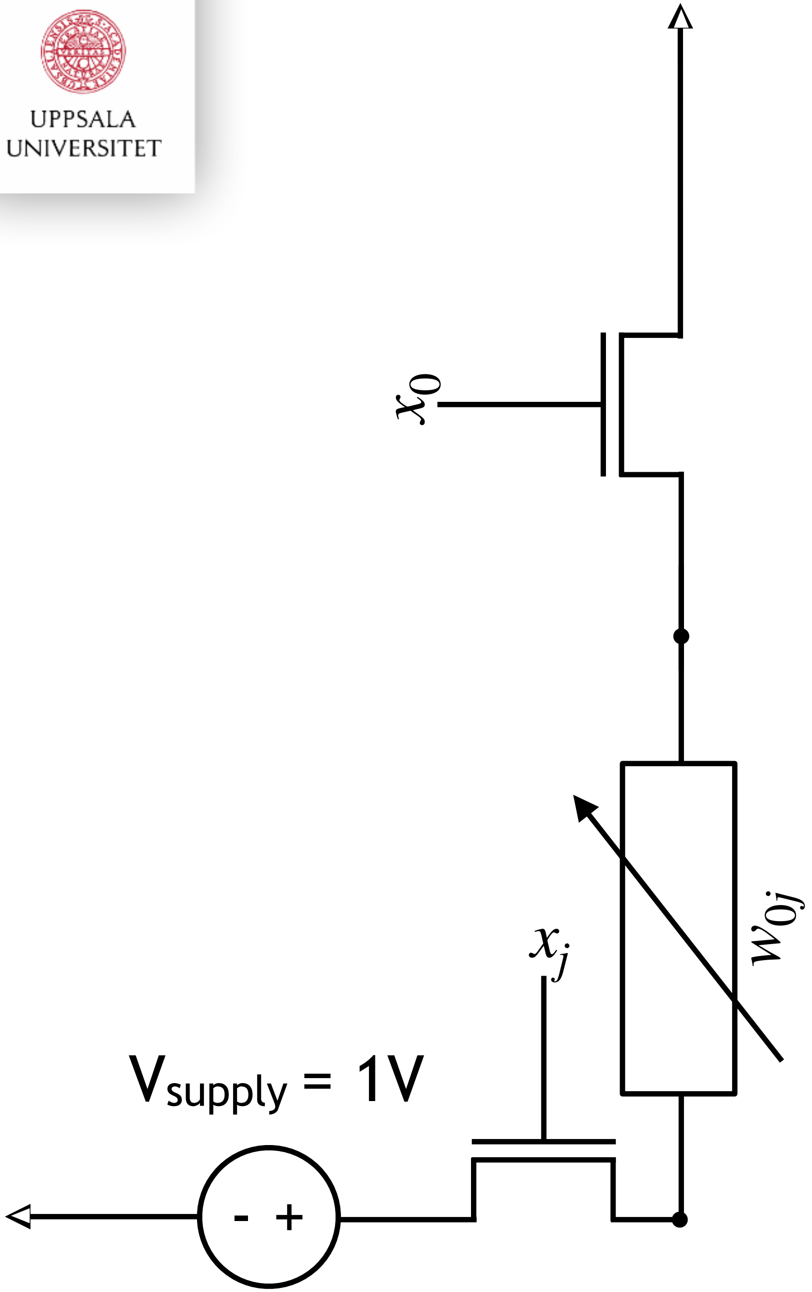


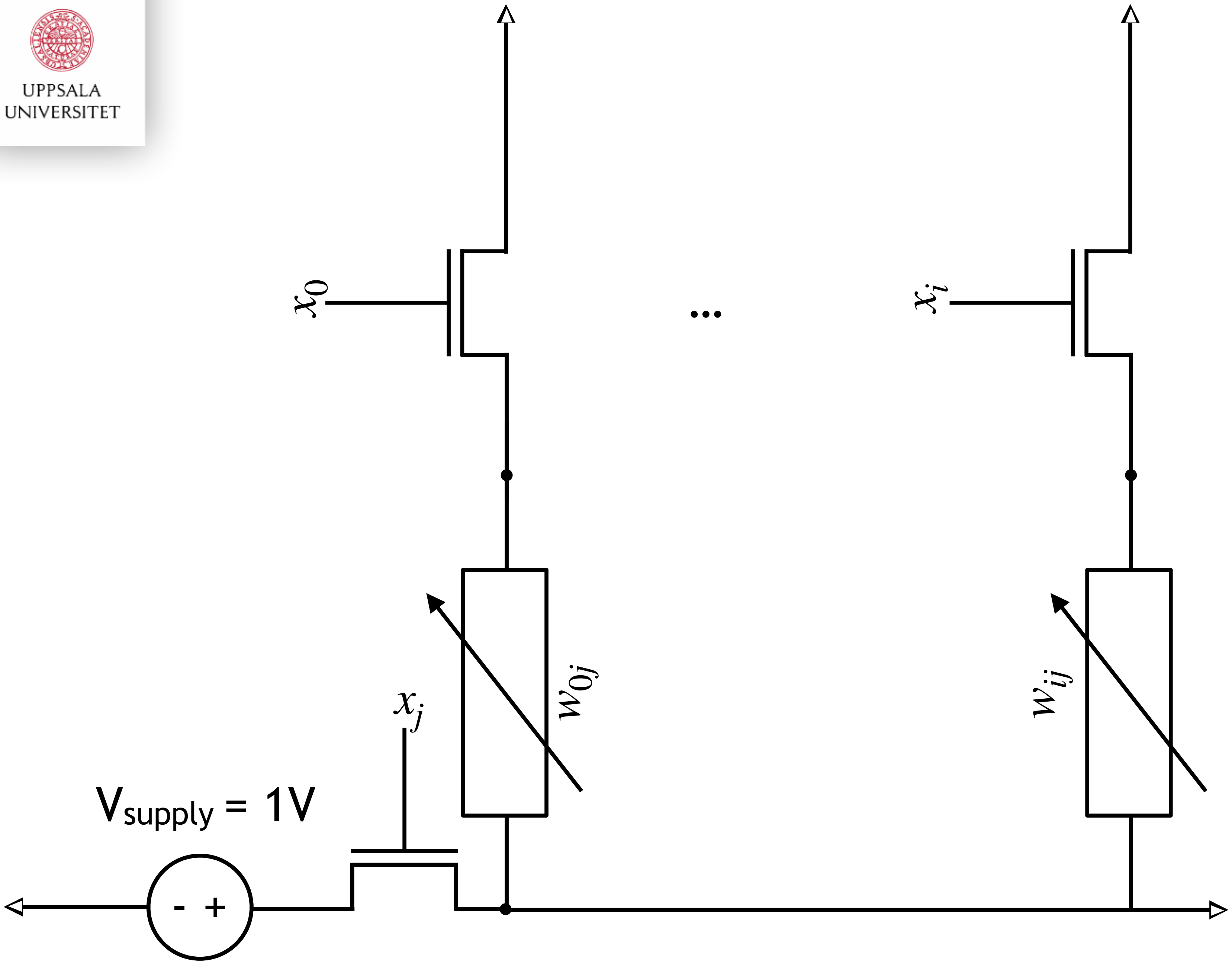
Memristors

$$V = I \cdot R$$

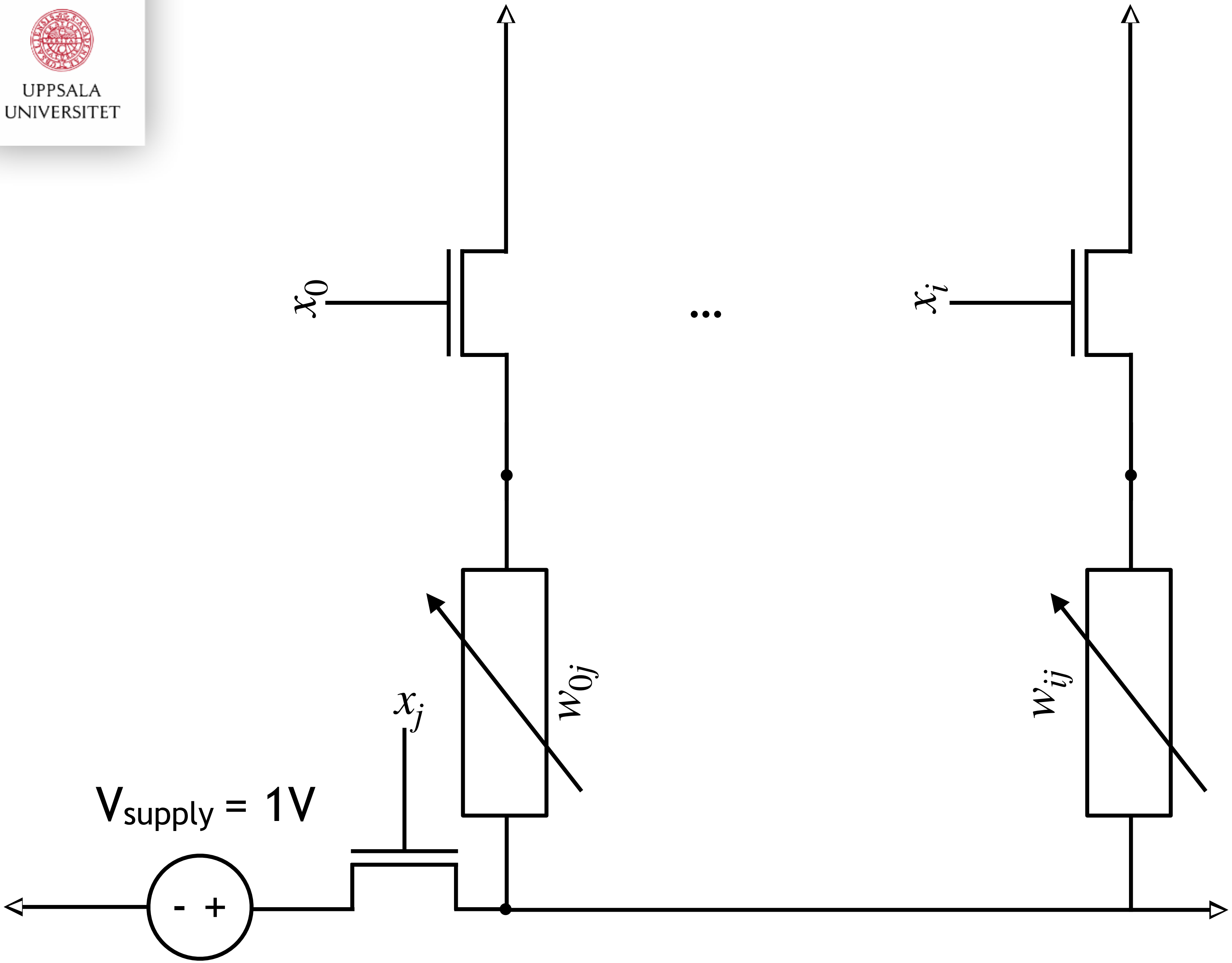



Vsupply = 1V
- +
x_j
w_{ij}
x_i
$I_r = x_i \cdot x_j \cdot w_{ij}$

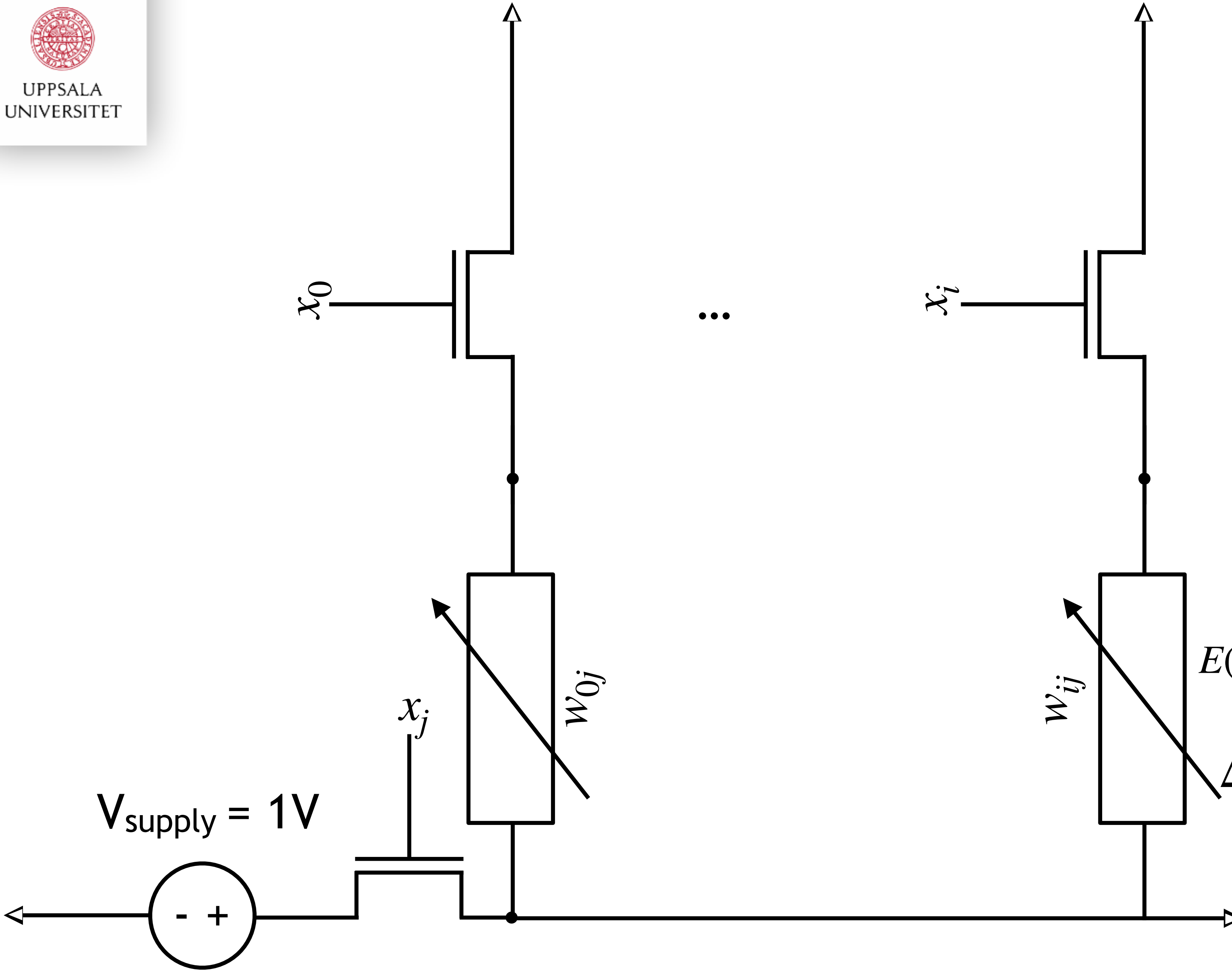

x_0
w_{0j}
x_j
V_{supply} = 1V
- +



x_0
...
x_i
w_{0j}
w_{ij}
x_j
$V_{supply} = 1V$
- +



$$I_j = \sum_{i=0} I_{ji} = \sum_{i=0} x_j x_i w_{ij}$$

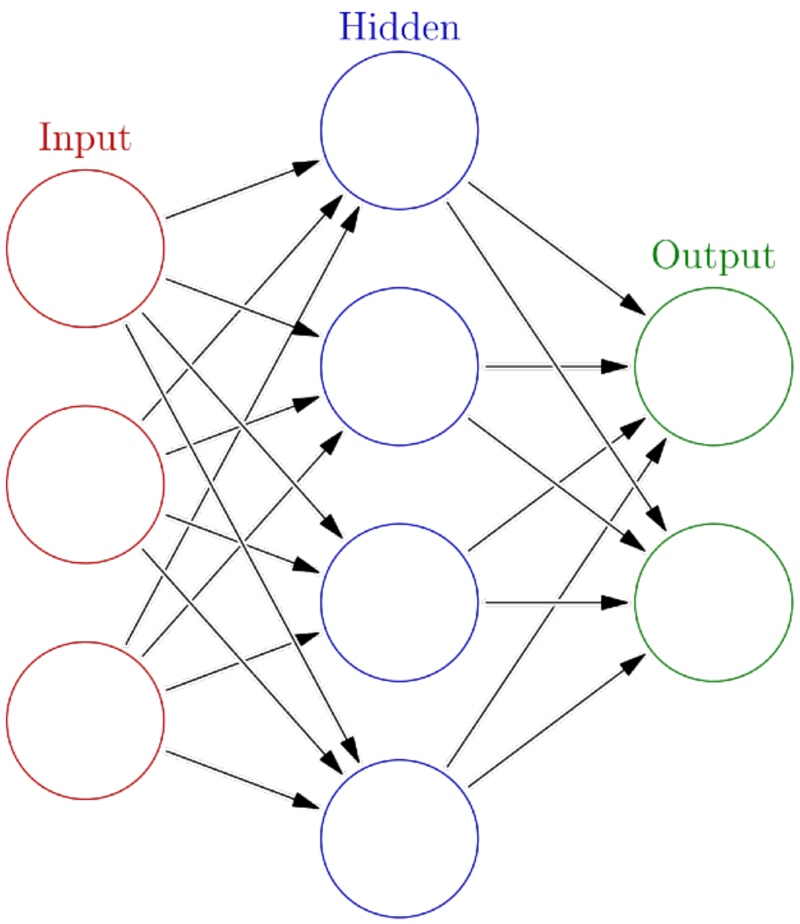


$$E(X) = -\frac{1}{2}\sum_{j}\sum_{i\neq j} x_j x_i w_{ij} - \sum_{j} x_j w_{jj}$$

$$\Delta E = (2x_j - 1)\Big(\sum_{i\neq j} x_i w_{ji} + w_{jj}\Big)$$

$$I_j = \sum_{i=0} I_{ji} = \sum_{i=0} x_j x_i w_{ij}$$

Application to Other Problems



- Different mappings for different problems
- Deep learning uses “Restricted Boltzmann Machines”
  - Combination of layered approach of modern neural nets and Boltzmann Machines

UPPSALA UNIVERSITET

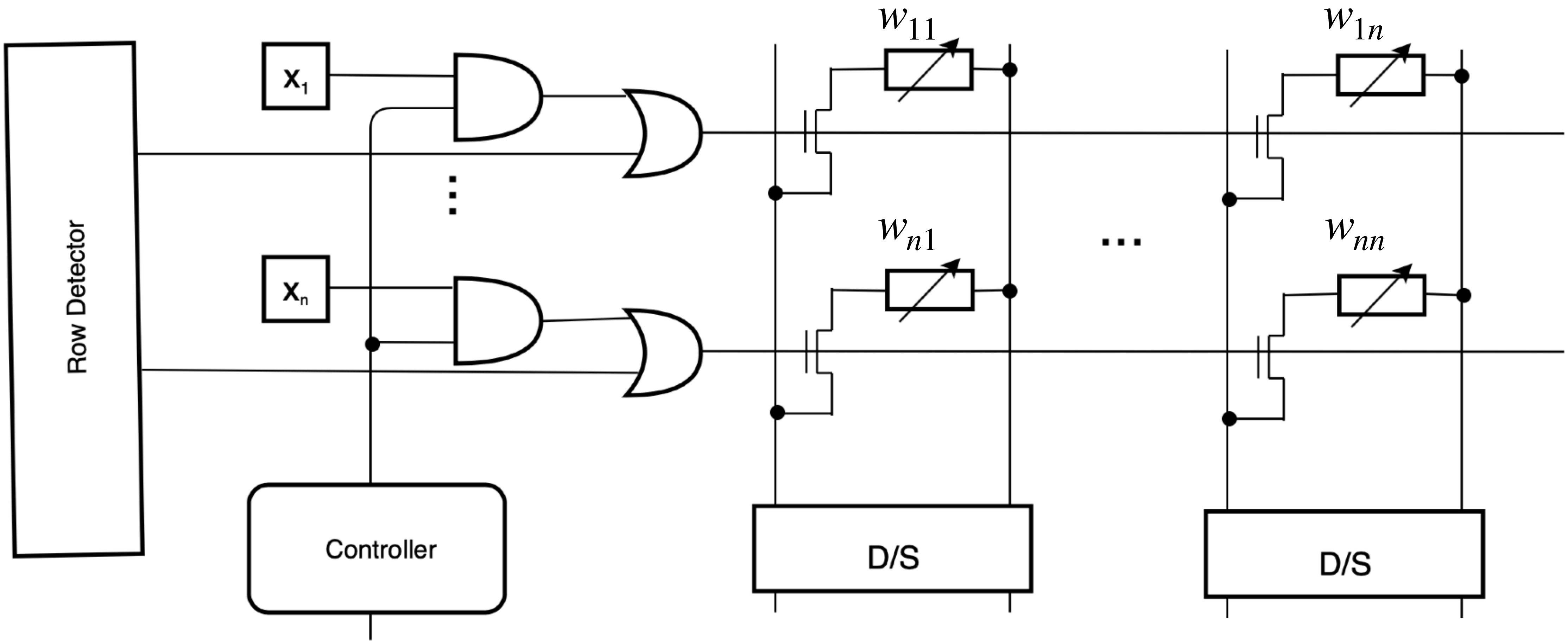
System Organization and Training

ORGANIZATION

Interface

- Interfaces through regular DDRx
- Envisioned as a modular addition to any system with applicable workload

Organization

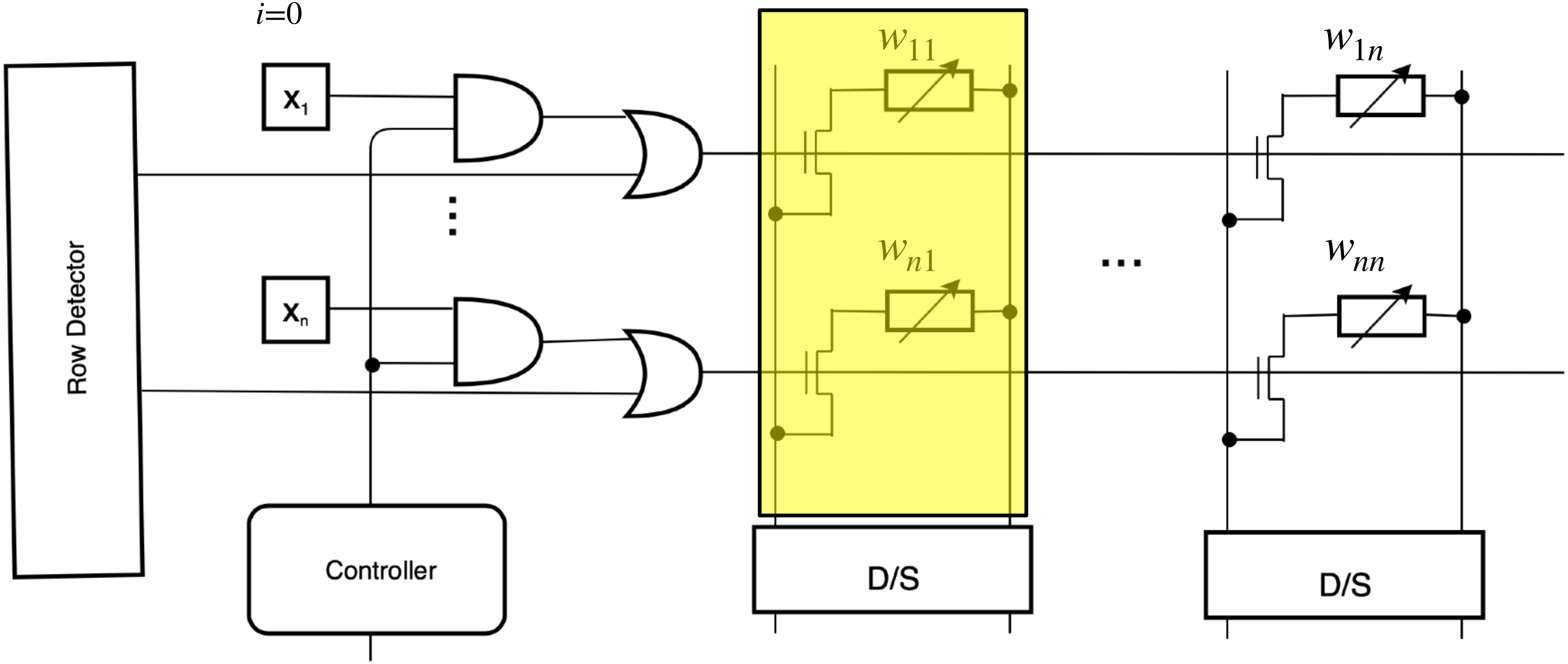


Organization

Current pulled by this column is

$$x_1 \cdot \sum_{i=0}^{n} x_i w_{ij}$$

Sensed by 'D/S' circuit



Consensus

$$P(x^j \mid x) = \frac{1}{1 + e^{\frac{\Delta E}{C}}}$$

Consensus

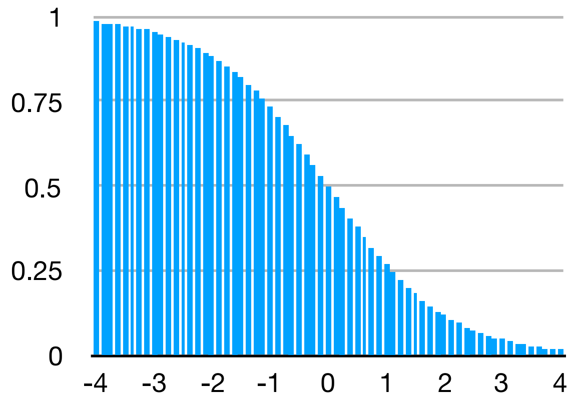
$$P(x^j \mid x) = \frac{1}{1 + e^{\frac{\Delta E}{C}}}$$

- Sigmoid function is hard to implement in hardware

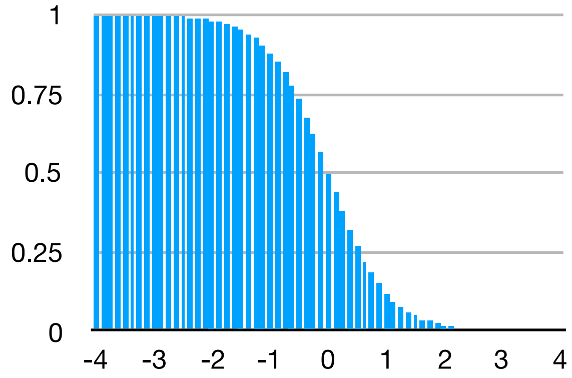
Consensus

$$P(x^j \mid x) = \frac{1}{1 + e^{\frac{\Delta E}{C}}}$$

- Sigmoid function is hard to implement in hardware
- Use 64 evenly sampled points from expected range for values of ΔE [-4, 4]
- Different Cs are valid but paper uses C=1



C=1



C=.5

Notes on Sensing Circuits

- Sense amplifiers sense one bit of the current
- Tree structures to sum them in quick fashion
- A large machine must tolerate a lot of inaccuracy and noise generated by current sensing

D/S

Performance and Energy

RESULTS

Optimization Problems

- 57.75x speedup over single threaded kernel
- 6.19x over PIM (process in memory)
- Many workloads achieved closer to 100x speedup over single threaded kernel

Optimization Problems

- 57.75x speedup over single threaded kernel
- 6.19x over PIM (process in memory)
- Many workloads achieved closer to 100x speedup over single threaded kernel

- 25x less energy used than single threaded kernel
- 5.2x less energy used than PIM

Deep Learning

- Even faster… 68.79x faster than single thread
- 6.89x faster than PIM

Deep Learning

- Even faster… 68.79x faster than single thread
- 6.89x faster than PIM

- 63x less energy than single thread
- 5.3x less energy than PIM