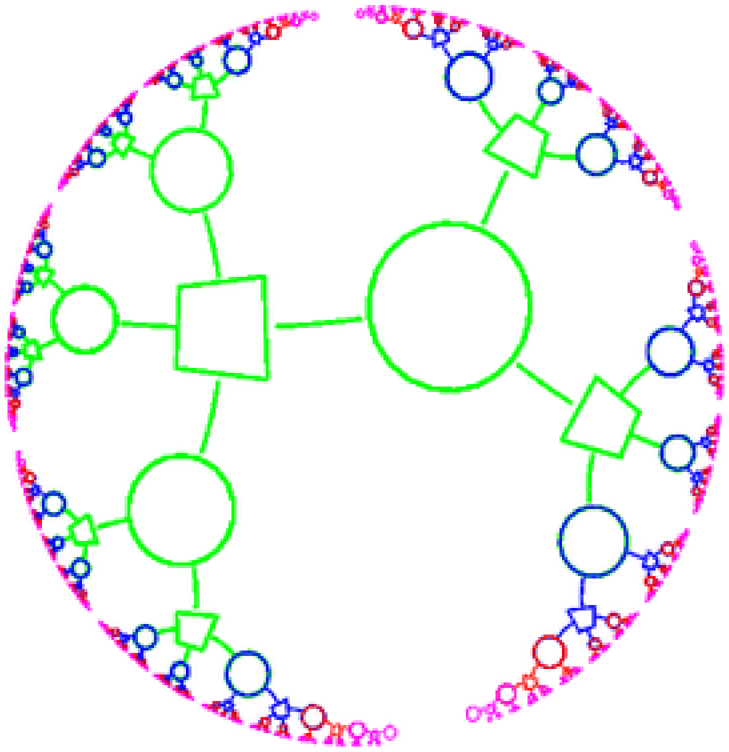
Neural Networks, part II

- Content addressable memories

State-of-the-art

error-correcting codes

Information theory, pattern recognition, and neural networks



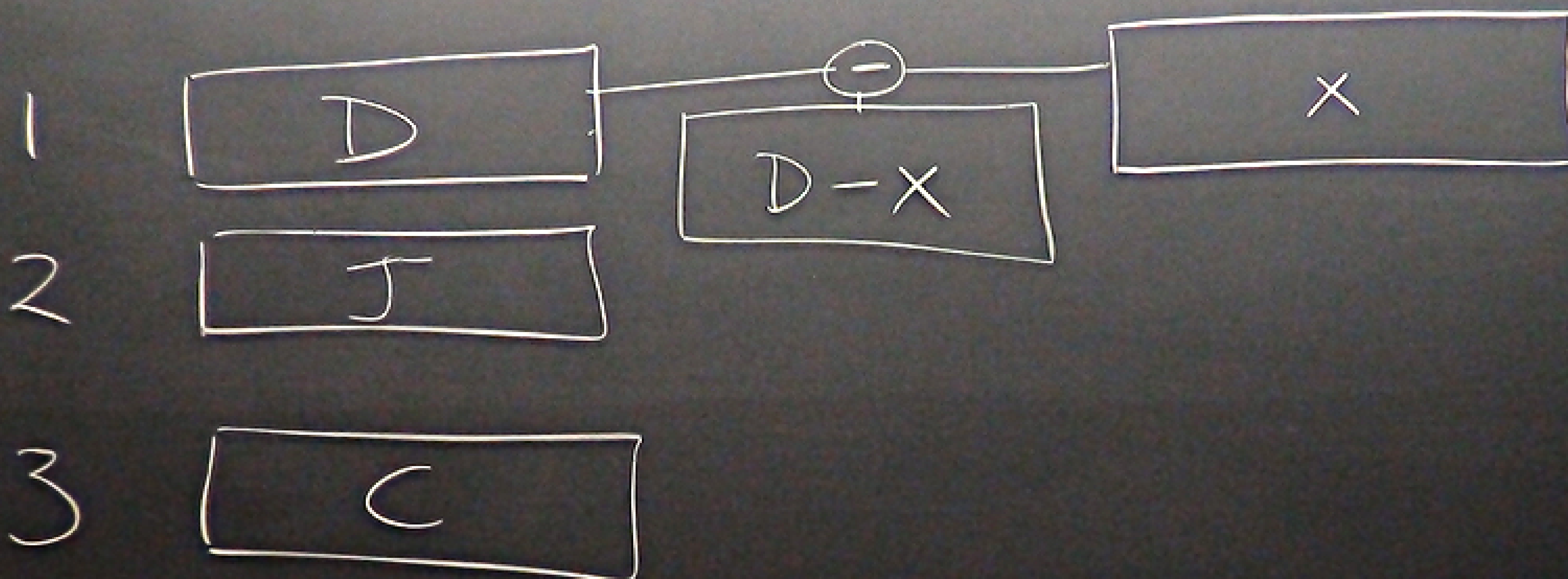
- Source coding (Data compression)
- Noisy-channel coding
- Inference + probabilistic methods
  - 9-10 Inference
  - 11 Clustering
  - 12 Monte Carlo methods
  - 13 Advanced Monte Carlo methods
  - 14 Variational methods
- Neural networks
  - 15 Introduction to feedforward neural networks
  - 16 Content-addressable memories
- State-of-the-art error-correcting codes

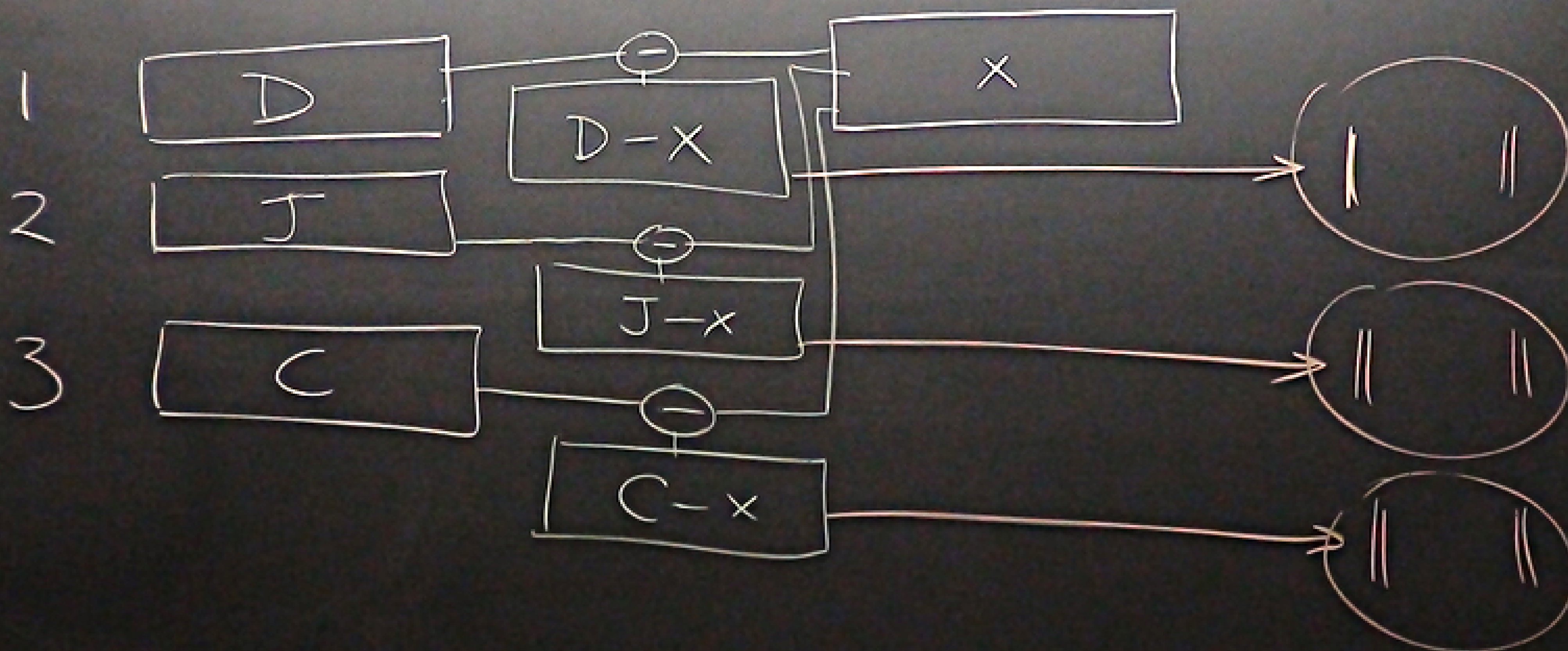
www.inference.phy.cam.ac.uk/itprnn/
www.inference.phy.cam.ac.uk/itila/

An orthodox approach to content-addressable memory

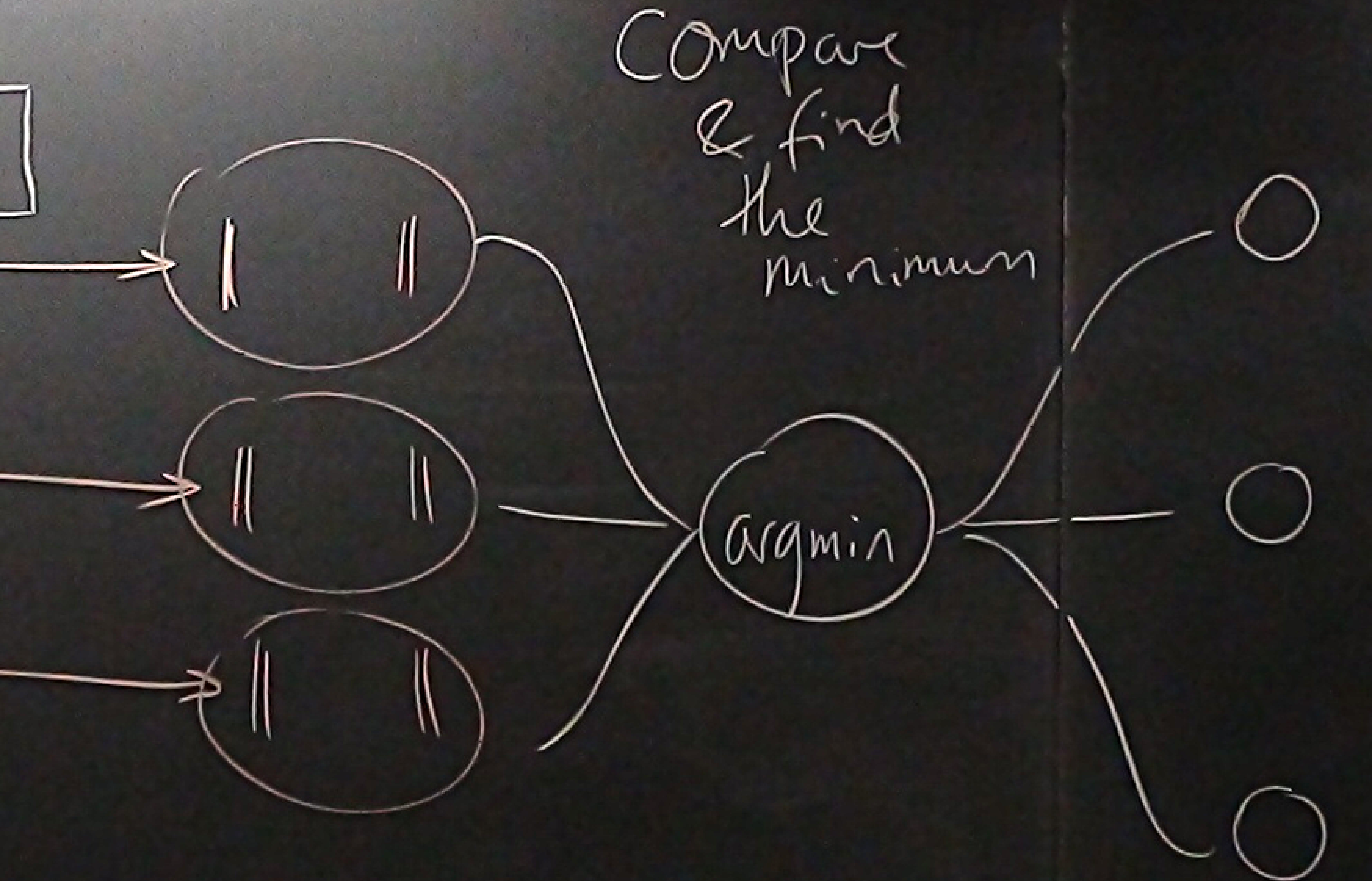
note that this is a brute-force decoder

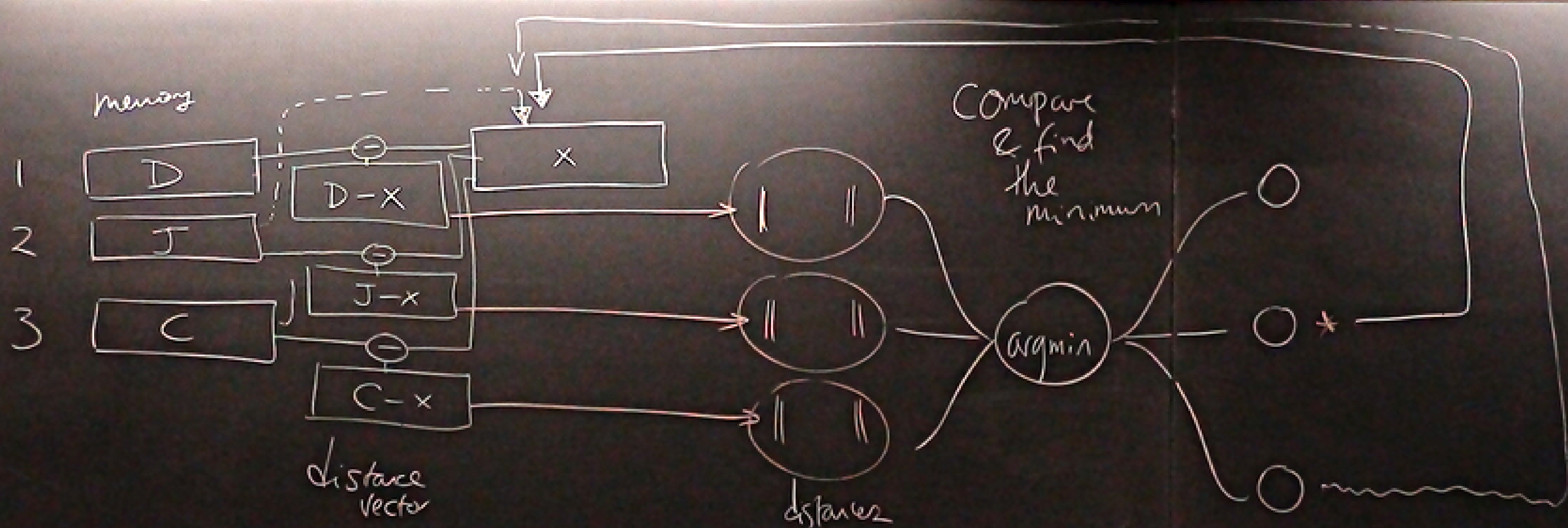
for the error-correcting code whose codewords are the required memories

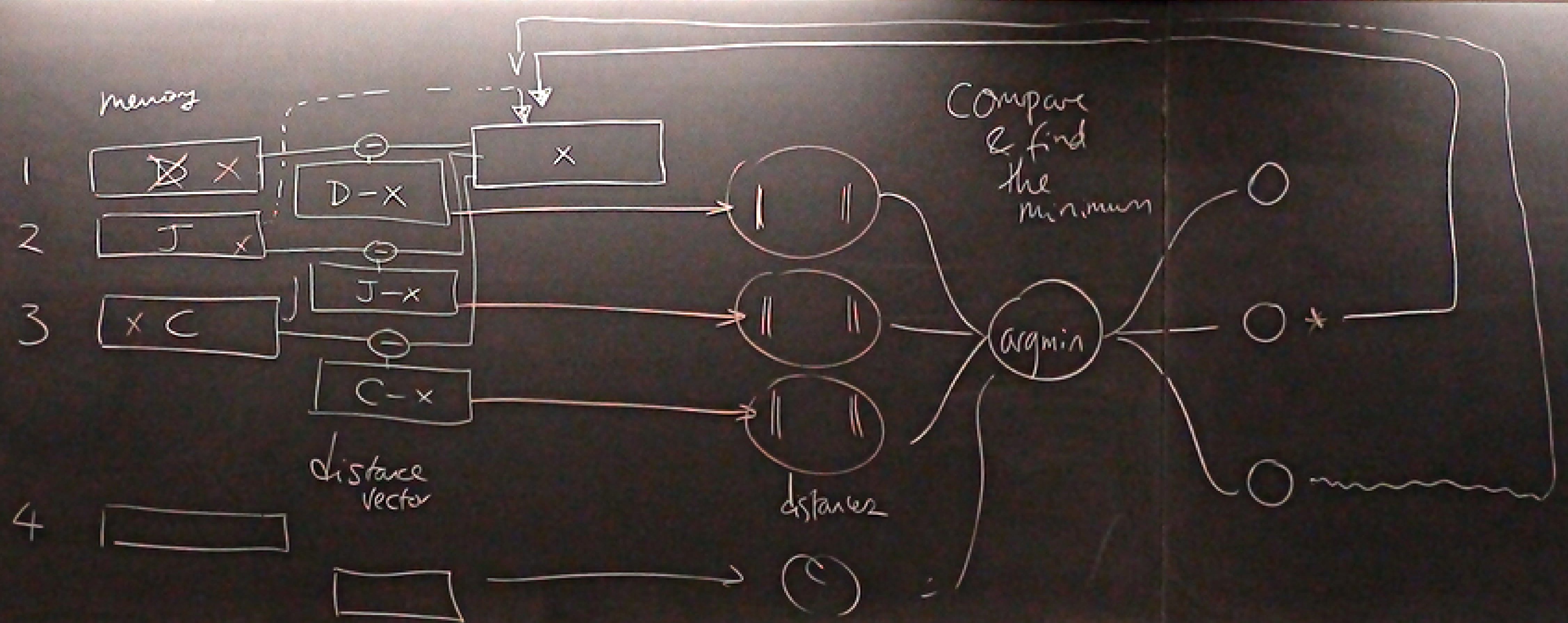

1
D
D-X
X
2
J
3
C



1
2
3
D
J
C
X
D-X
J-X
C-X


Compare
& find
the
minimum
argmin


memory
1
2
3
D
J
C
D-X
J-x
C-x
x
distance vector
distances
Compare & find the minimum
argmin
*


memory
1
2
3
4
D-x
J-x
C-x
x
distance vector
distances
compare & find the minimum
argmin

An orthodox approach to content-addressable memory

note that this is a brute-force decoder

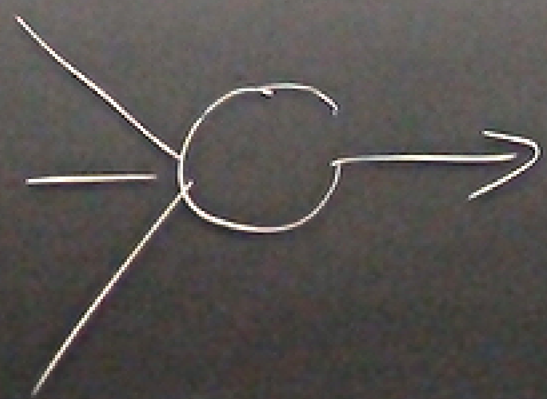
for the error-correcting code whose codewords are the required memories

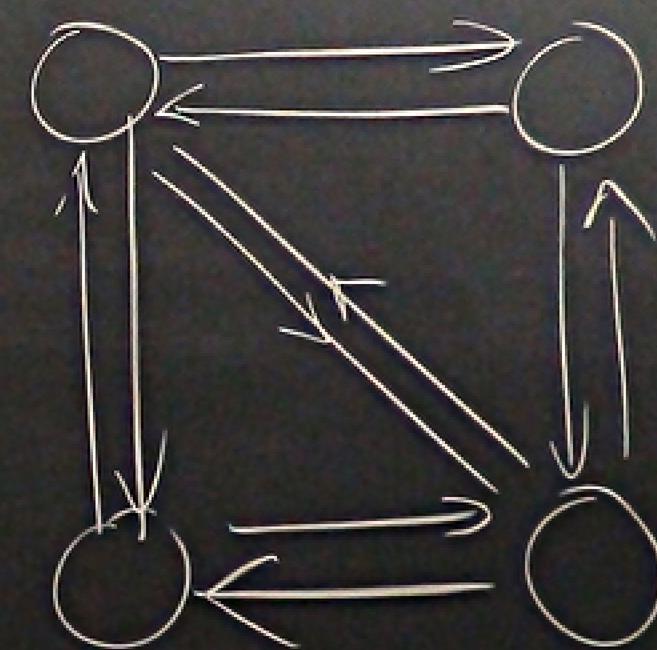
Neural Networks, Part II:

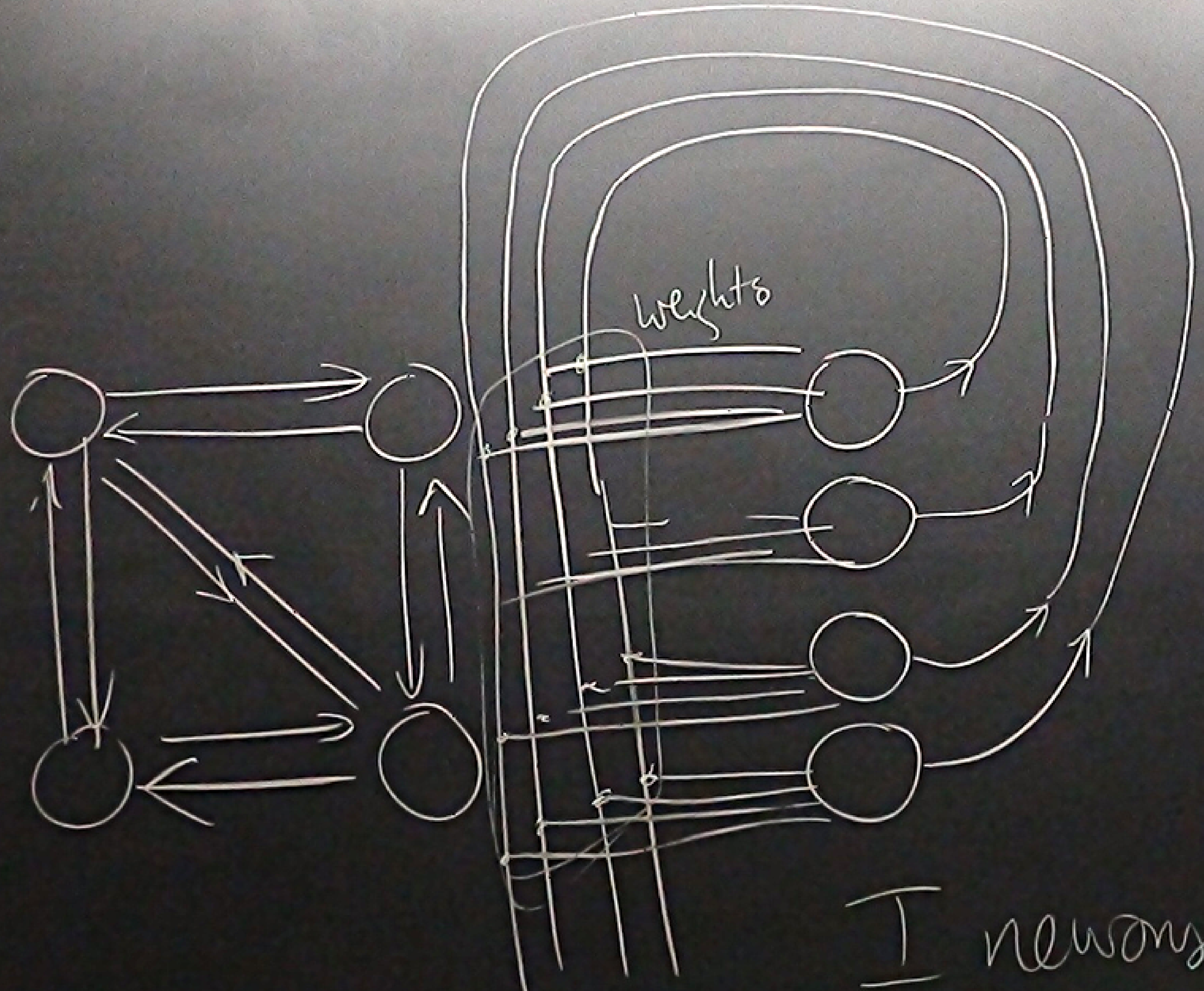
Feedback Networks

The Hopfield Network

Chapter 42






weights
I neurons

The Hopfield network

Architecture

Feedback network

Symmetric weights $w_{ij} = w_{ji}, \quad w_{ii} = 0$

Activity rule

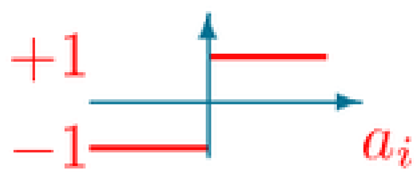
Continuous Hopfield network:

$$a_i = \sum_j w_{ij} x_j + \theta_i$$

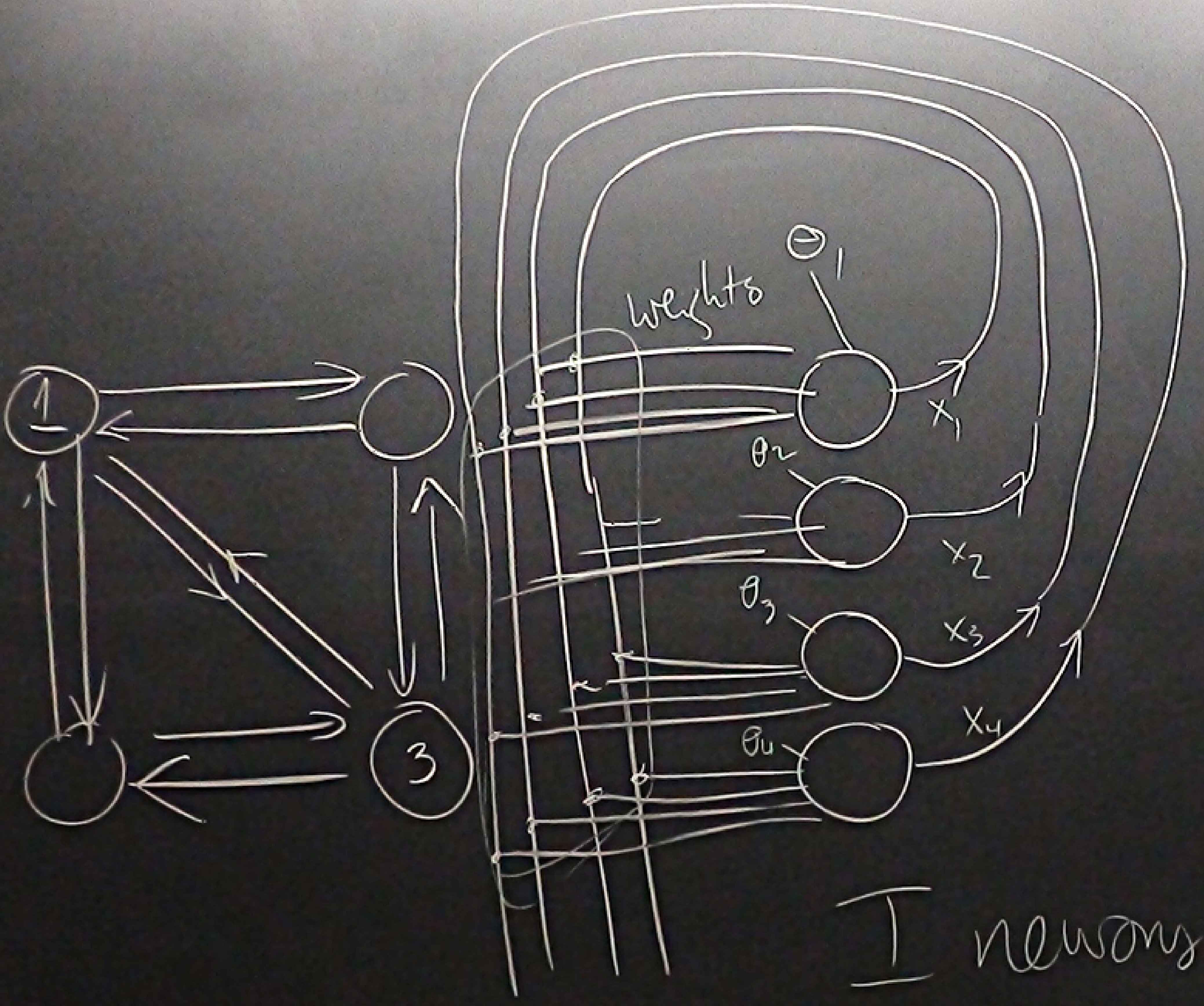
$$x_i = \tanh a_i$$

Binary Hopfield network:

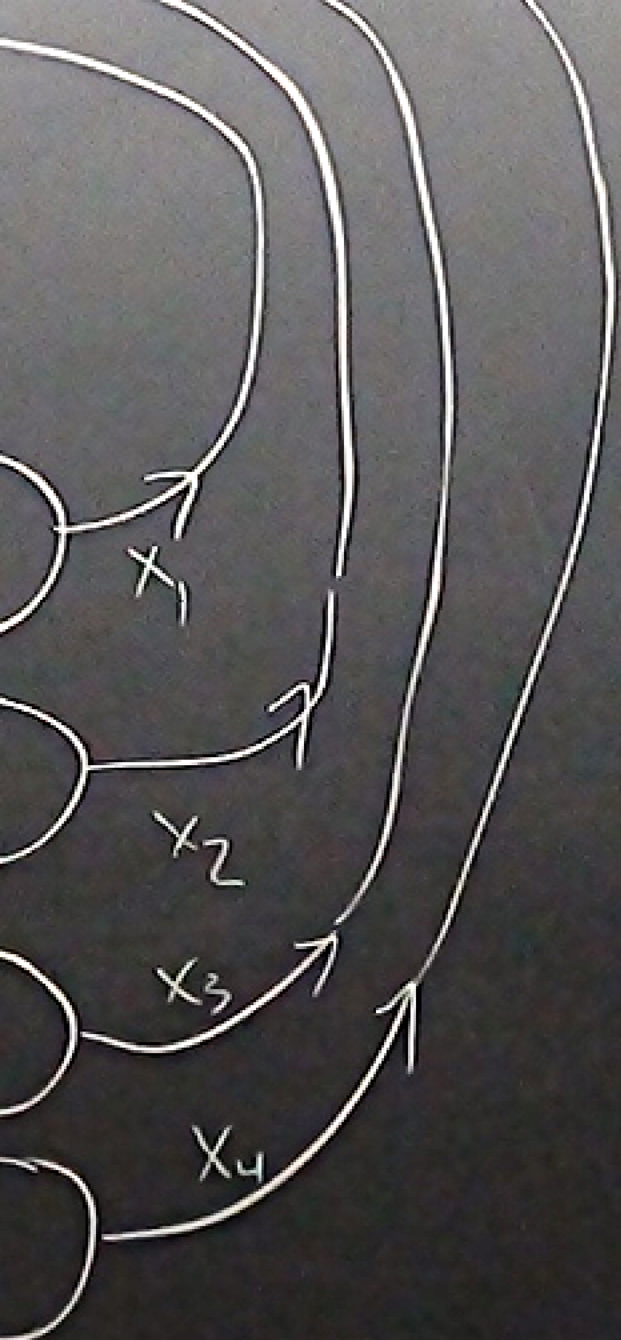
$x_i =$

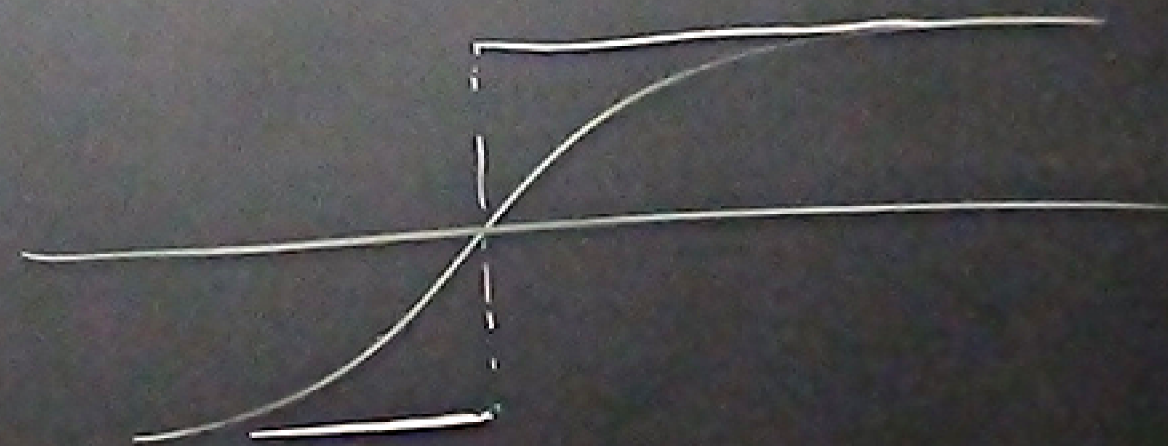


John Hopfield



1
3
weights
θ1
θ2
θ3
θ4
x1
x2
x3
x4
I neurons





$I \times I$ weight ma

I neurons

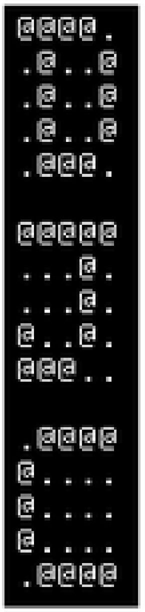
weight matrix

$$\frac{25 \times 24}{2} = \# \text{ parameters}$$

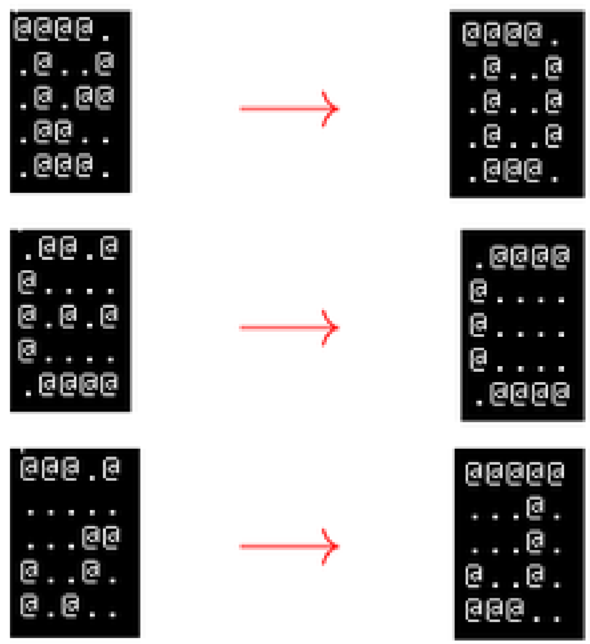
$W_{ij} =$

$I = 25$

Content-addressable memory challenge



- Make a dynamical system
  - 25 dynamical variables, 300 parameters (4-bit precision)
  - that has attracting fixed points at desired memories
  - such that noisy versions are automatically cleaned up



  - and such that new memories can be added incrementally
  - AND robust to corruption of more than half the parameters

Hopfield network and Hebbian learning

Architecture

Feedback network

Symmetric weights $w_{ij} = w_{ji}, \quad w_{ii} = 0$

Activity rule

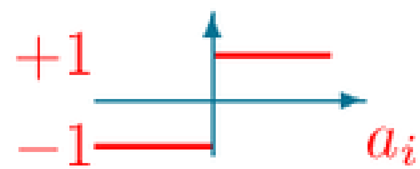
Continuous Hopfield network:

$$a_i = \sum_j w_{ij} x_j + \theta_i$$

$$x_i = \tanh a_i$$

Binary Hopfield network:

$x_i =$



Learning rule

Hebb rule

$$w_{ij} = \sum_{n=1}^{N} x_i^{(n)} x_j^{(n)}$$

patterns to learn

$N=3$

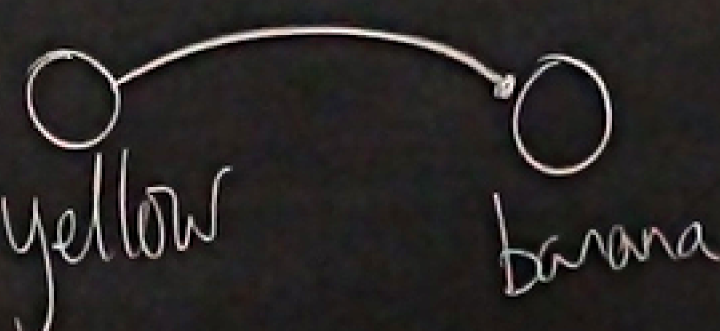
$$W_{ij} = \sum_{n=1}^{N} x_i^{(n)} x_j^{(n)}$$

b

j

c

M $N=4$



McGurk effect

From Wikipedia, the free encyclopedia

The **McGurk effect** is a perceptual phenomenon that demonstrates an interaction b
perception. The illusion occurs when the auditory component of one sound is paired
another sound, leading to the perception of a third sound.[1] The visual information a
speak changes the way they hear the sound.[2] People who are used to watching dub
who are not susceptible to the McGurk effect because they have, to some extent, lea
are getting from the mouths of the "speakers".[3] If a person is getting poor quality a
visual information, they may be more likely to experience the McGurk effect.[4] Integ
information may also influence whether a person will experience the effect. People t
have been shown to be more susceptible to the effect.[2] Many people are affected d
based on many factors, brain damages or disorders.

Contents [hide]

WIKIPEDIA
The Free Encyclopedia
Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact Wikipedia
Toolbox
Print/export
Languages
Deutsch
Français

Hebbian learning

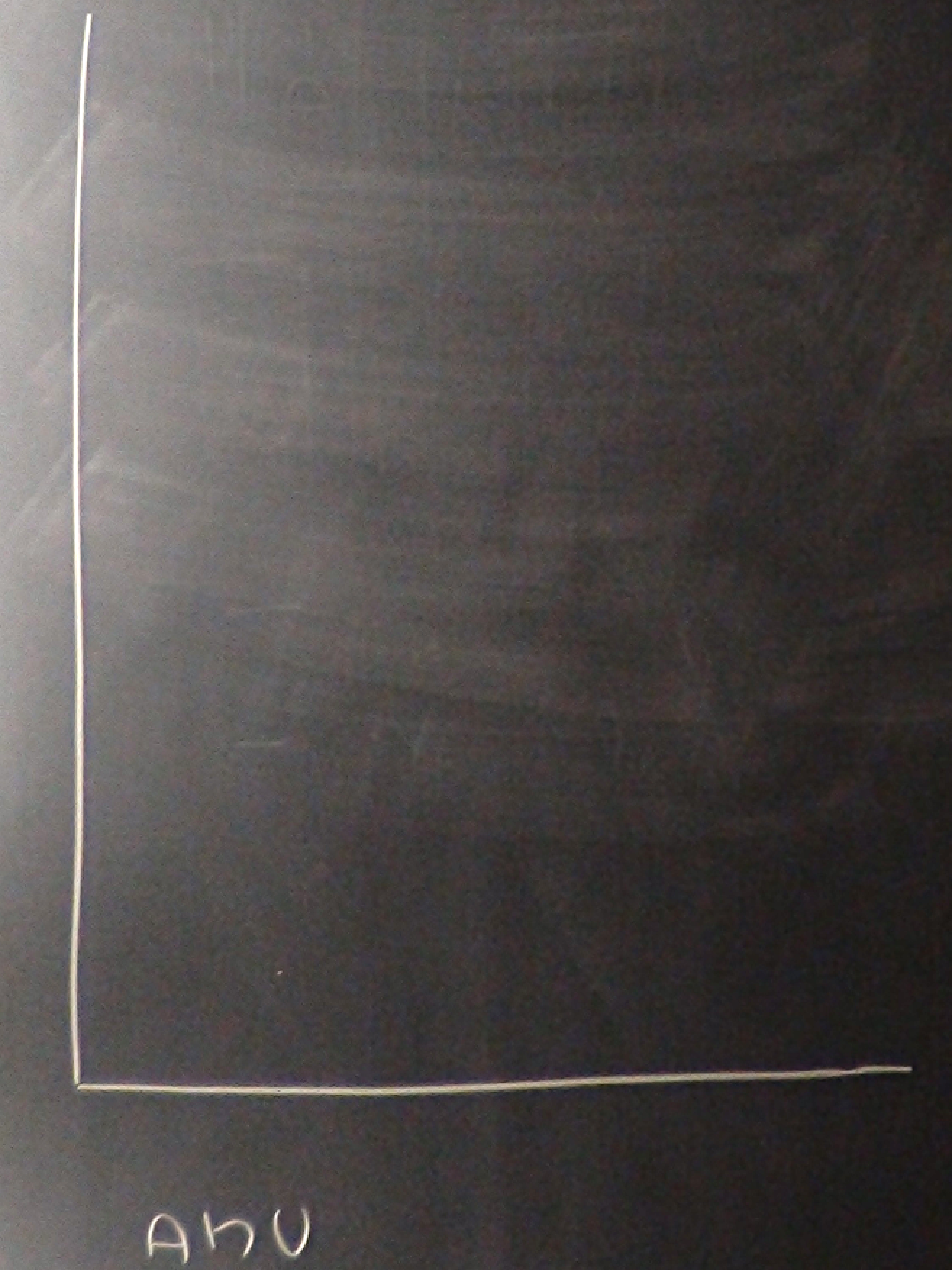
Desired memories

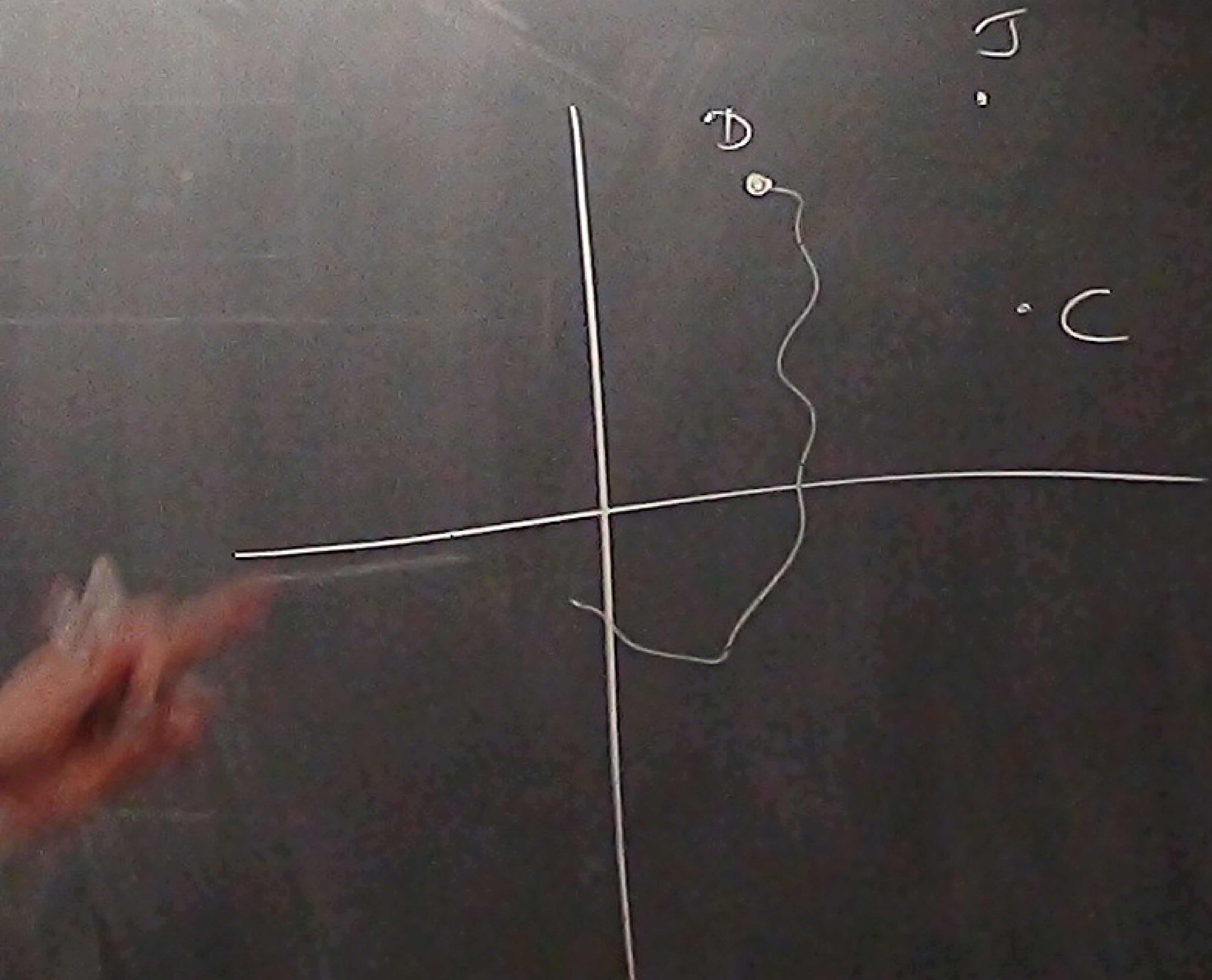
```
@@@@.
.@..@
.@..@
.@..@
.@@@.

@@@@@
...@.
...@.
@..@.
@@@..

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

Weights

```
W:
  . 1 1 1-1-3 1-1 1 1-3 1-1 1 1-1 1-1 1 1 1 1 1-1-3
  1 . 3 3 1-1-1-3-1-1-1-1-3-1-1 1-1-3-1-1-1 3 3 1-1
  1 3 . 3 1-1-1-3-1-1-1-1-3-1-1 1-1-3-1-1-1 3 3 1-1
  1 3 3 . 1-1-1-3-1-1-1-1-3-1-1 1-1-3-1-1-1 3 3 1-1
 -1 1 1 1 . 1-3-1 1-3 1-3-1 1-3 3-3-1 1-3 1 1 1-1 1
 -3-1-1-1 1 .-1 1-1-1 3-1 1-1-1 1-1 1-1-1-1-1-1 1 3
  1-1-1-1-3-1 . 1-1 3-1 3 1-1 3-3 3 1-1 3-1-1-1 1-1
 -1-3-3-3-1 1 1 . 1 1 1 1 3 1 1-1 1 3 1 1 1-3-3-1 1
  1-1-1-1 1-1-1 1 .-1-1-1 1 3-1 1-1 1 3-1 3-1-1-3-1
  1-1-1-1-3-1 3 1-1 .-1 3 1-1 3-3 3 1-1 3-1-1-1 1-1
 -3-1-1-1 1 3-1 1-1-1 .-1 1-1-1 1-1 1-1-1-1-1-1 1 3
  1-1-1-1-3-1 3 1-1 3-1 . 1-1 3-3 3 1-1 3-1-1-1 1-1
 -1-3-3-3-1 1 1 3 1 1 1 1 . 1 1-1 1 3 1 1 1-3-3-1 1
  1-1-1-1 1-1-1 1 3-1-1-1 1 .-1 1-1 1 3-1 3-1-1-3-1
  1-1-1-1-3-1 3 1-1 3-1 3 1-1 .-3 3 1-1 3-1-1-1 1-1
 -1 1 1 1 3 1-3-1 1-3 1-3-1 1-3 .-3-1 1-3 1 1 1-1 1
  1-1-1-1-3-1 3 1-1 3-1 3 1-1 3-3 . 1-1 3-1-1-1 1-1
 -1-3-3-3-1 1 1 3 1 1 1 1 3 1 1-1 1 . 1 1 1-3-3-1 1
  1-1-1-1 1-1-1 1 3-1-1-1 1 3-1 1-1 1 .-1 3-1-1-3-1
  1-1-1-1-3-1 3 1-1 3-1 3 1-1 3-3 3 1-1 .-1-1-1 1-1
  1-1-1-1 1-1-1 1 3-1-1-1 1 3-1 1-1 1 3-1 .-1-1-3-1
  1 3 3 3 1-1-1-3-1-1-1-1-3-1-1 1-1-3-1-1-1 . 3 1-1
  1 3 3 3 1-1-1-3-1-1-1-1-3-1-1 1-1-3-1-1-1 3 . 1-1
 -1 1 1 1-1 1 1-1-3 1 1 1-1-3 1-1 1-1-3 1-3 1 1 . 1
 -3-1-1-1 1 3-1 1-1-1 3-1 1-1-1 1-1 1-1-1-1-1-1 1 .
```

cd ~/itp/hopfield ; source LONGDEMO


ANU

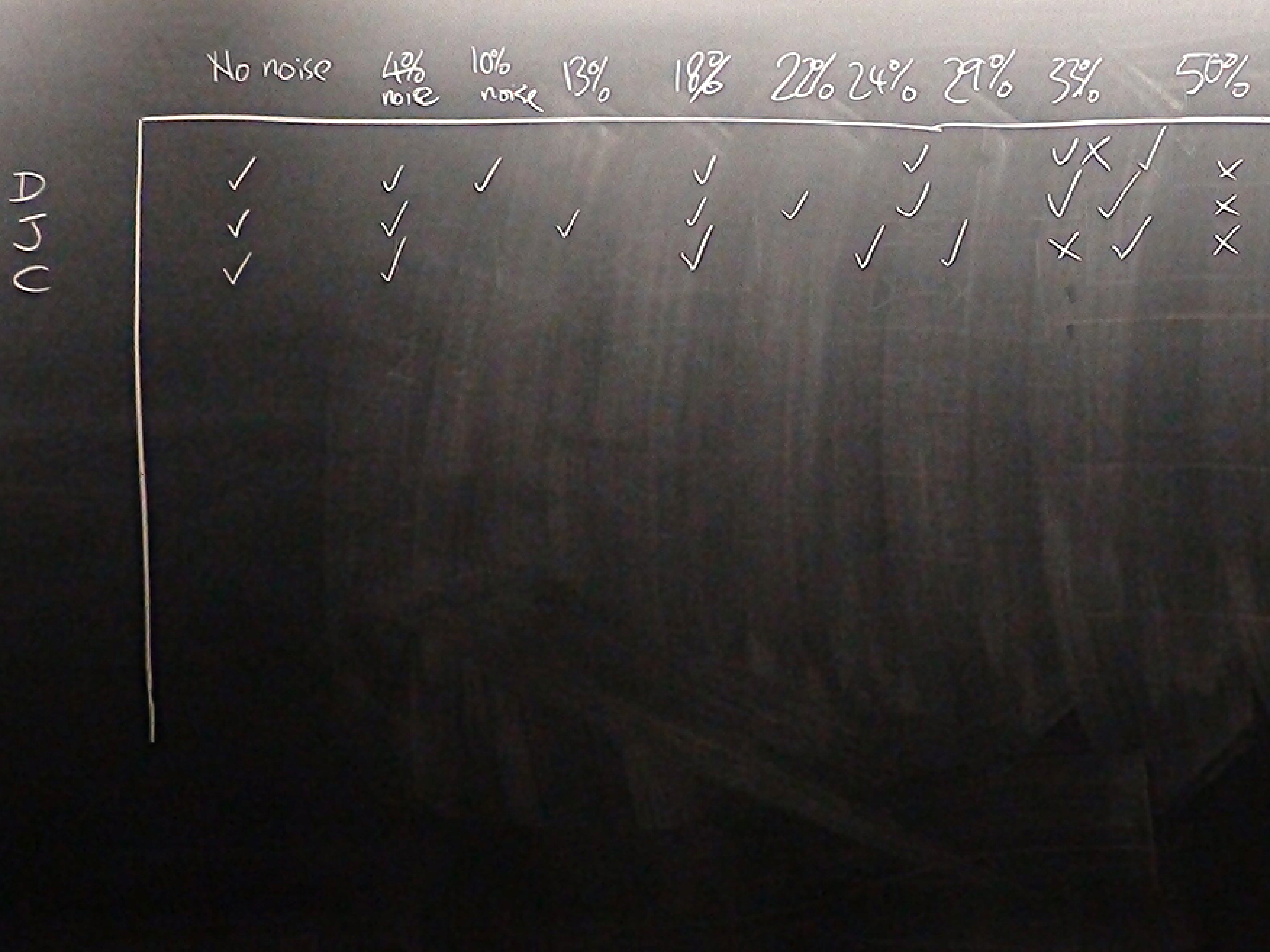

J
D
C

	No noise
DJC	✓

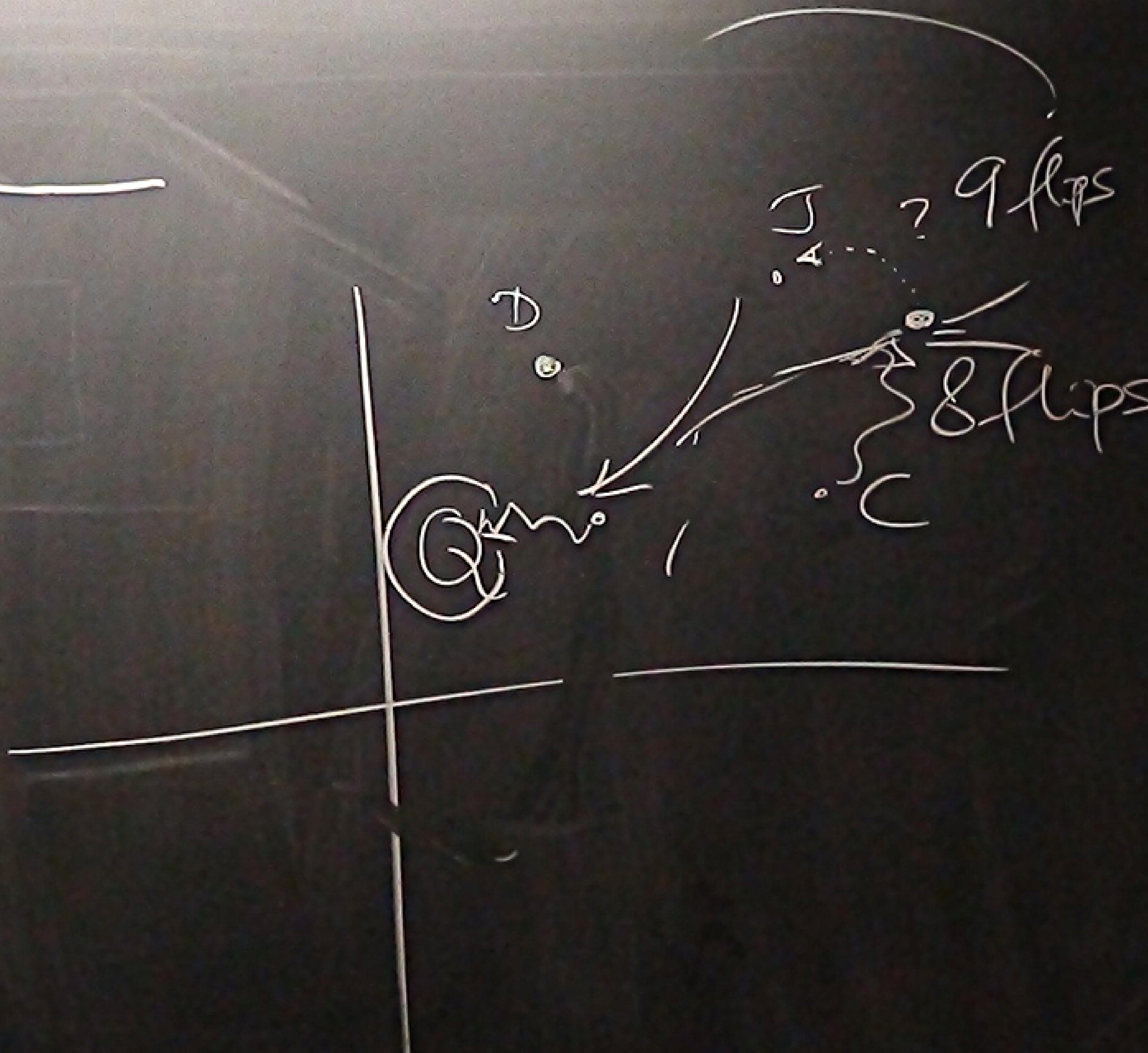
	No noise	4% noie	10% noise	13%
D	✓	✓	✓	
J	✓	✓		✓
C	✓	✓		

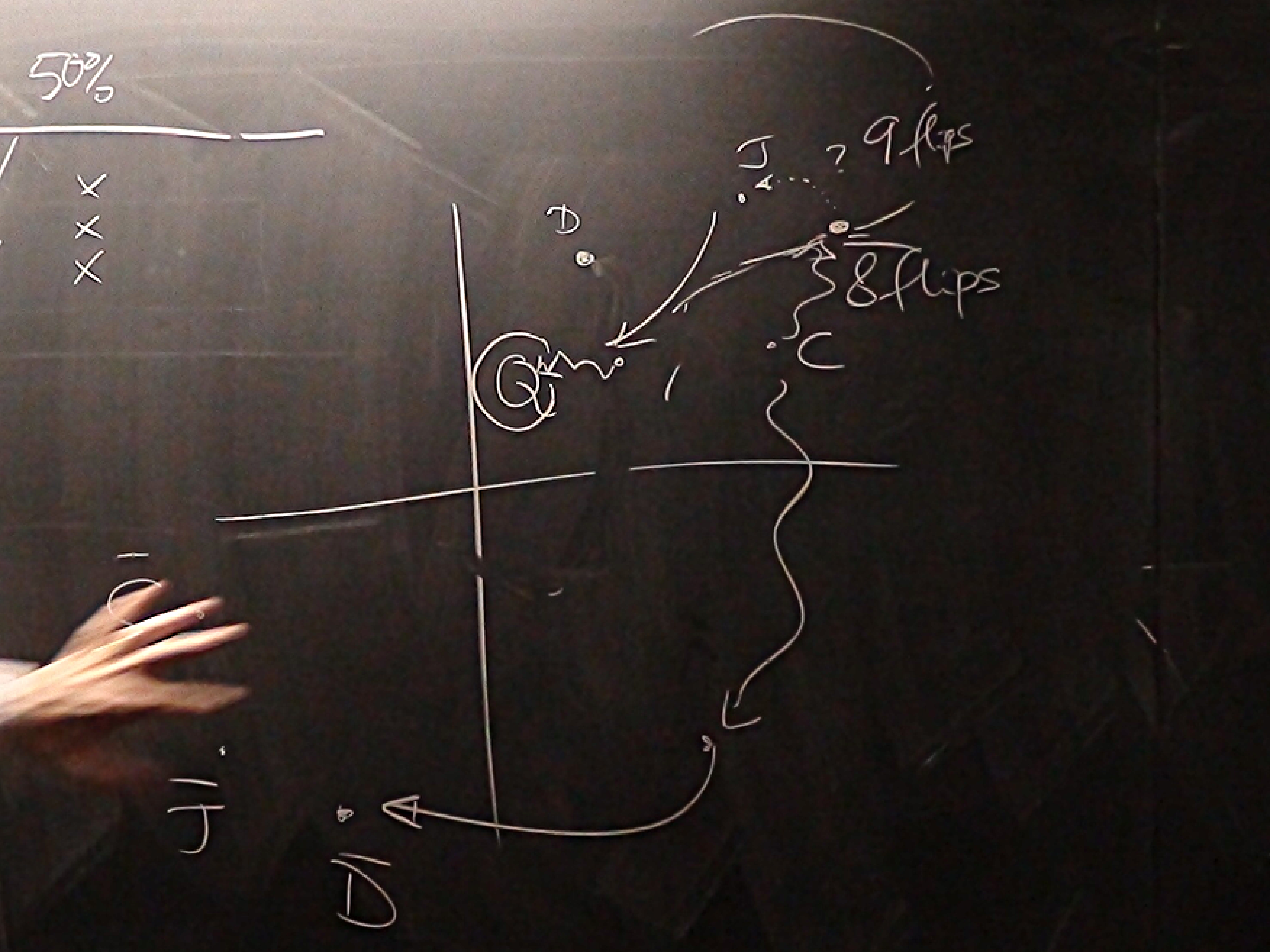
	No noise	4% noie	10% noise	13%	18%	22%	24%	29%	50%
D	✓	✓	✓		✓		✓		✗
J	✓	✓		✓	✓	✓	✓		✗
C	✓	✓			✓		✓	✓	✗

$Q = \square .$


No noise
4% noise
10% noise
13%
18%
22%
24%
29%
33%
50%
D
J
C

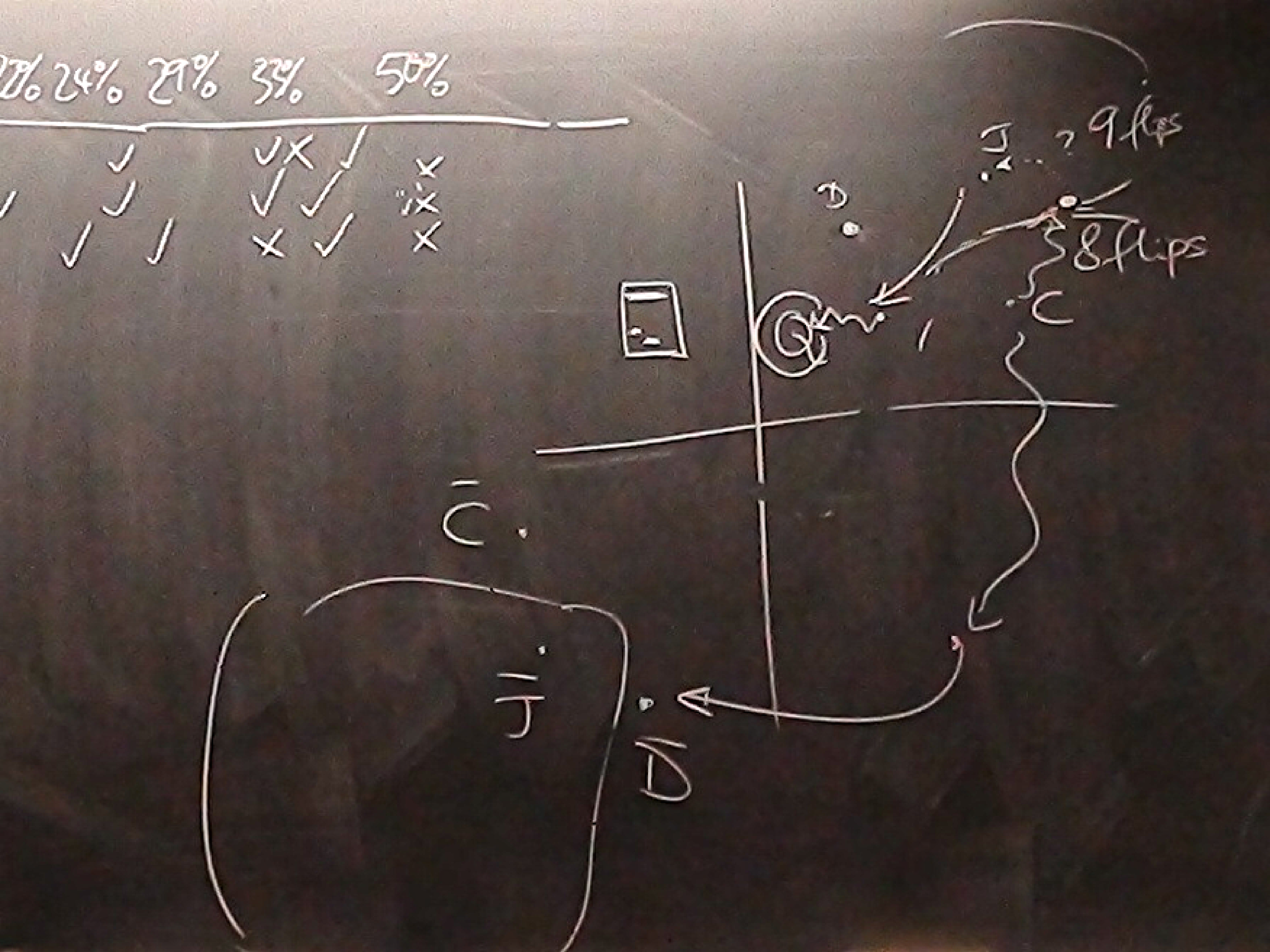
0%



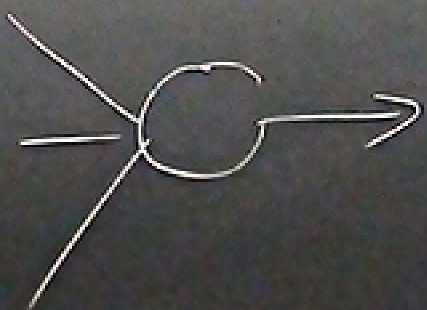

50%
9 flips
8 flips
J
?
D
Q
C
J̄
D̄

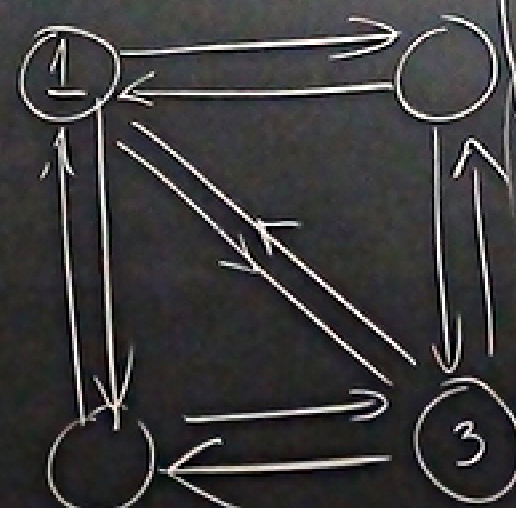
patte

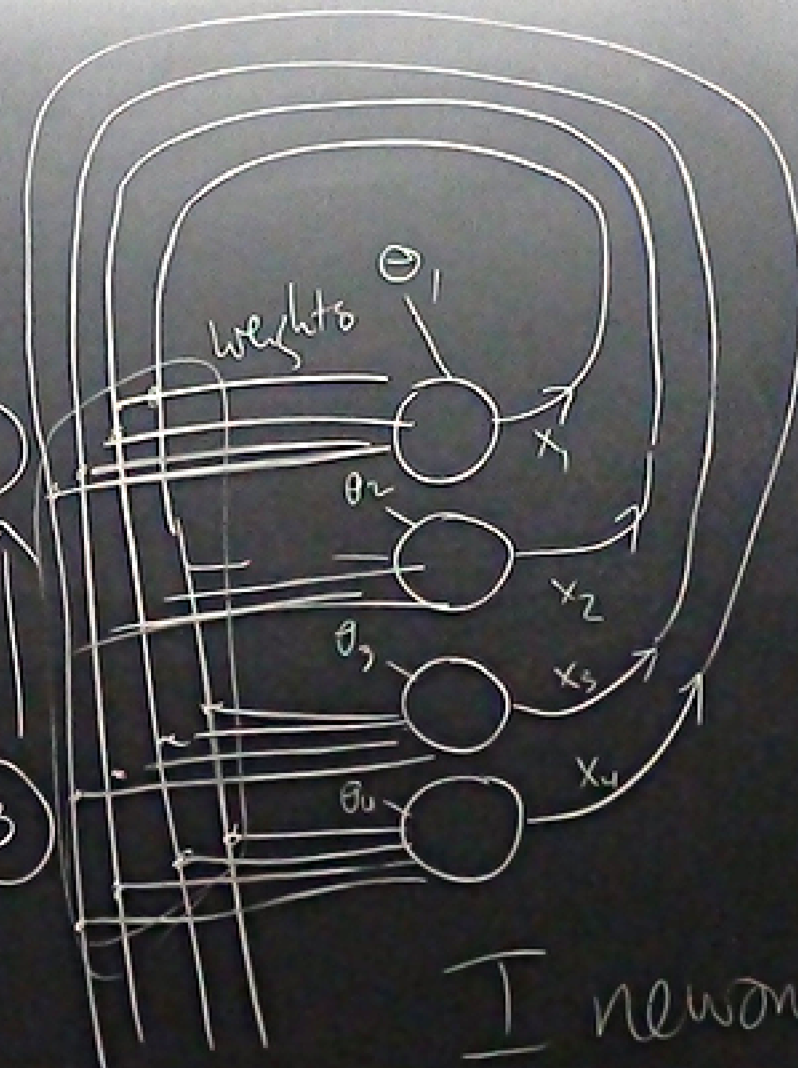
$$W_{ij} = \sum_{n=1}^{N} x_i^{(n)} x_j^{(n)}$$


24% 29% 35% 50%
J
9 flips
D
8 flips
C
Q
C̄
J̄
D̄

	No noise	4% noise	10% noise	13%	18%	22%	24%	29%	33%	50%
DJC	✓ ✓ ✓	✓ ✓ ✓	✓	✓	✓ ✓ ✓	✓	✓ ✓ ✓	✓	✓ ✗ ✓ ✓ ✓ ✗ ✓	✗ ✗ ✗







I neurons

patterns to learn

N=3

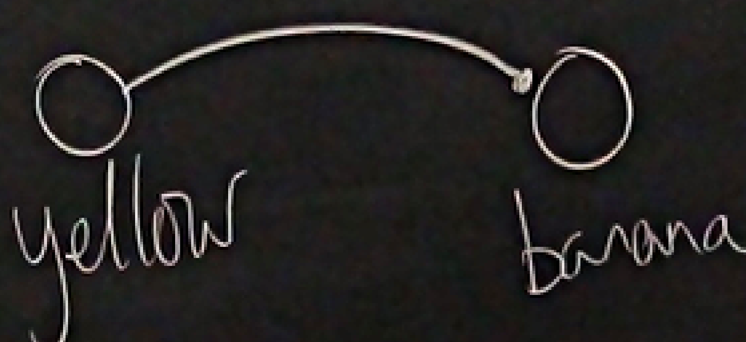
$$W_{ij} = \sum_{n=1}^{N} x_i^{(n)} x_j^{(n)}$$

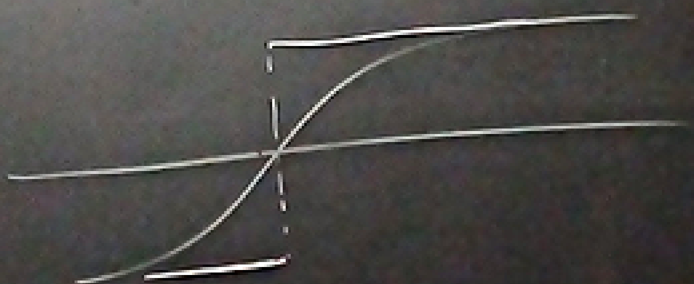
b

j

c

M N=4





$I \times I$ weight matrix

$$\frac{25 \times 24}{2} = \# \text{ parameters}$$

$I = 25$

	No noise	4% noise	10% noise	13%	18%
D	✓	✓	✓		✓
J	✓	✓		✓	✓
C	✓	✓			✓
Corrupt weights					

		No noise	4% noise	10% noise	13%	18%	22%	2
	D	✓	✓	✓		✓		
	J	✓	✓		✓	✓	✓	
	C	✓	✓			✓		
Corrupt weights 79/300	D				✓			
	J		✓		✓			
	C				✓			

0% 15% 29% Noise

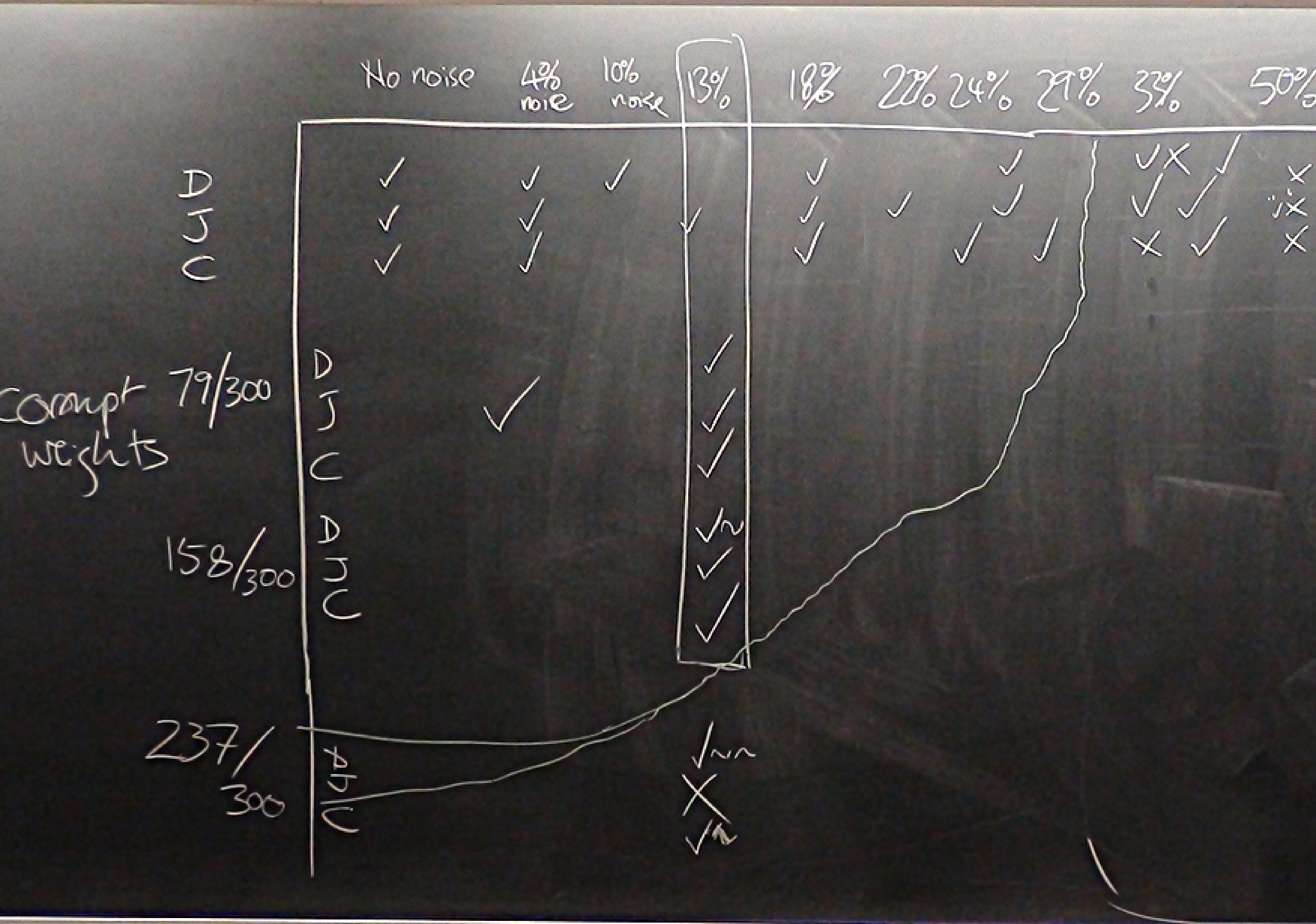
ADC ✓ ✓ ✓ ✓ ✓ ✓

ADCM

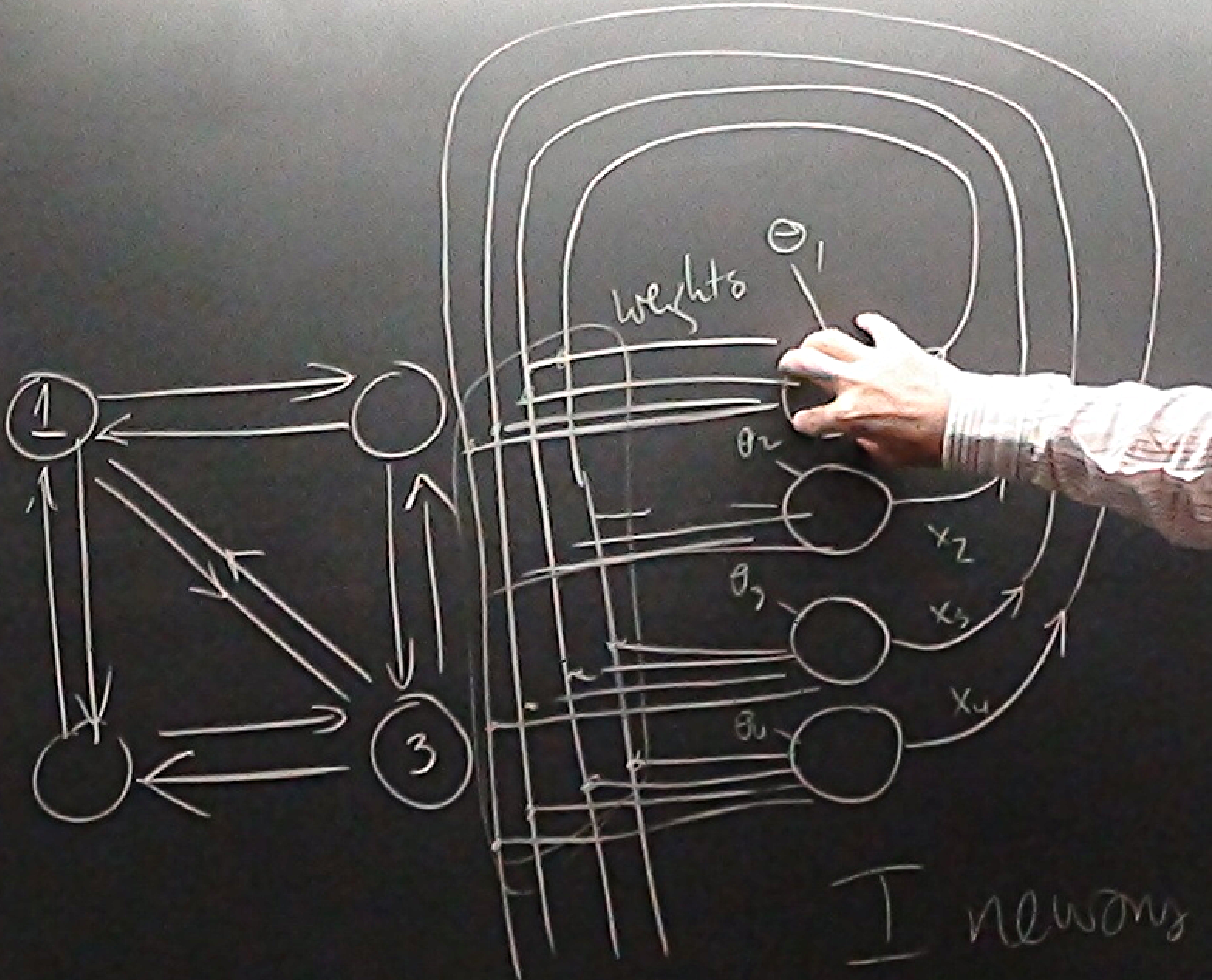
N = 4

N = 5

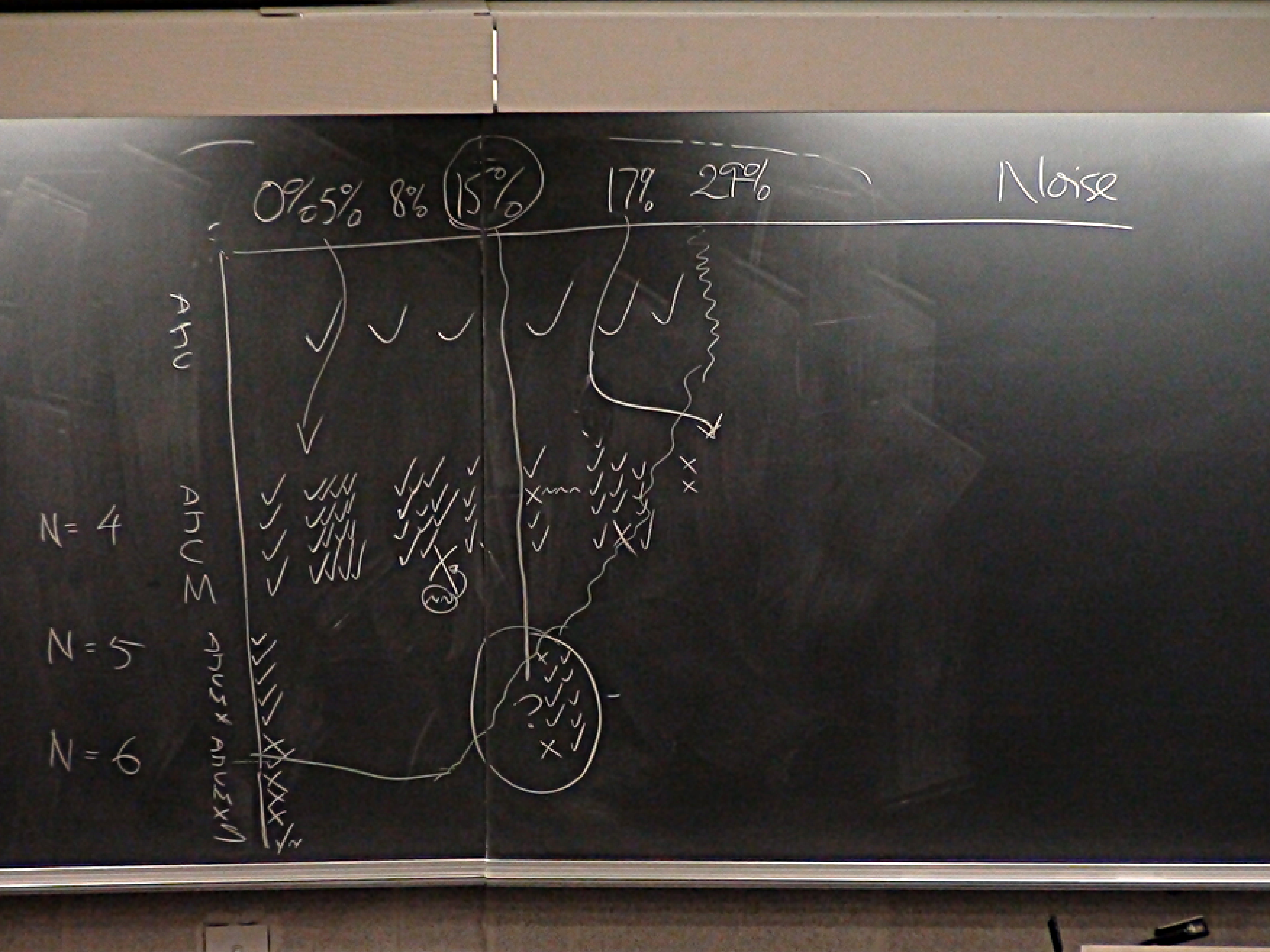
N = 6



No noise
4% noise
10% noise
13%
18%
22%
24%
29%
33%
50%
D
J
C
corrupt weights
79/300
D
J
C
158/300
D
J
C
237/300
D
J
C



1
3
weights
θ_1
θ_2
θ_3
θ_4
x_2
x_3
x_4
I neurons


Noise
0% 5% 8% 15% 17% 29%
AHU
AHUM
N = 4
N = 5
N = 6

The Hopfield network

- Why does Hopfield net have stable states at all?
- What is the capacity of the Hopfield net?
- What else can we do with them?

Q: Where have we seen the Hopfield net before?

$$w_{ij} = w_{ji}, \quad w_{ii} = 0$$

Dynamics

$$a_i = \sum_j w_{ij} x_j + \theta_i$$
$$x_i = \tanh a_i$$

Hint:

$$a_m = \sum_n J_{mn} \bar{x}_n + h_m$$
$$\bar{x}_n = \tanh a_n$$

Variational methods

When we approximated the spin system whose energy function was

$$E(\mathbf{x};\mathbf{J}) = -\frac{1}{2}\sum_{m,n} J_{mn}x_m x_n - \sum_n h_n x_n$$

with a separable distribution

$$Q(\mathbf{x};\mathbf{a}) = \frac{1}{Z_Q}\exp\left(\sum_n a_n x_n\right)$$

and optimized Q so as to minimize the variational free energy

$$\beta\tilde{F}(\mathbf{a}) = \beta\sum_{\mathbf{x}} Q(\mathbf{x};\mathbf{a})E(\mathbf{x};\mathbf{J}) - \sum_{\mathbf{x}} Q(\mathbf{x};\mathbf{a})\ln\frac{1}{Q(\mathbf{x};\mathbf{a})},$$

we found that the iterative equations

$$a_m = \beta\left(\sum_n J_{mn}\bar{x}_n + h_m\right) \quad \text{and} \quad \bar{x}_n = \tanh(a_n)$$

were guaranteed to decrease the variational free energy

$$\beta\tilde{F}(\mathbf{a}) = \beta\left(-\frac{1}{2}\sum_{m,n} J_{mn}\bar{x}_m\bar{x}_n - \sum_n h_n\bar{x}_n\right) - \sum_n H_2^{(e)}(q_n).$$

HN $\equiv$ VFE min $\underset{\text{"Q"}}{\underline{\text{approx}}}$ to

$$P(x) = \frac{e^{\sum_{ij} x_i w_{ij} x_j}}{Z}$$

$\boxed{\text{Boltzmann machine}}$

learning algorithm for the BM

$\longrightarrow$ Hebb rule!

Hopfield net dynamics

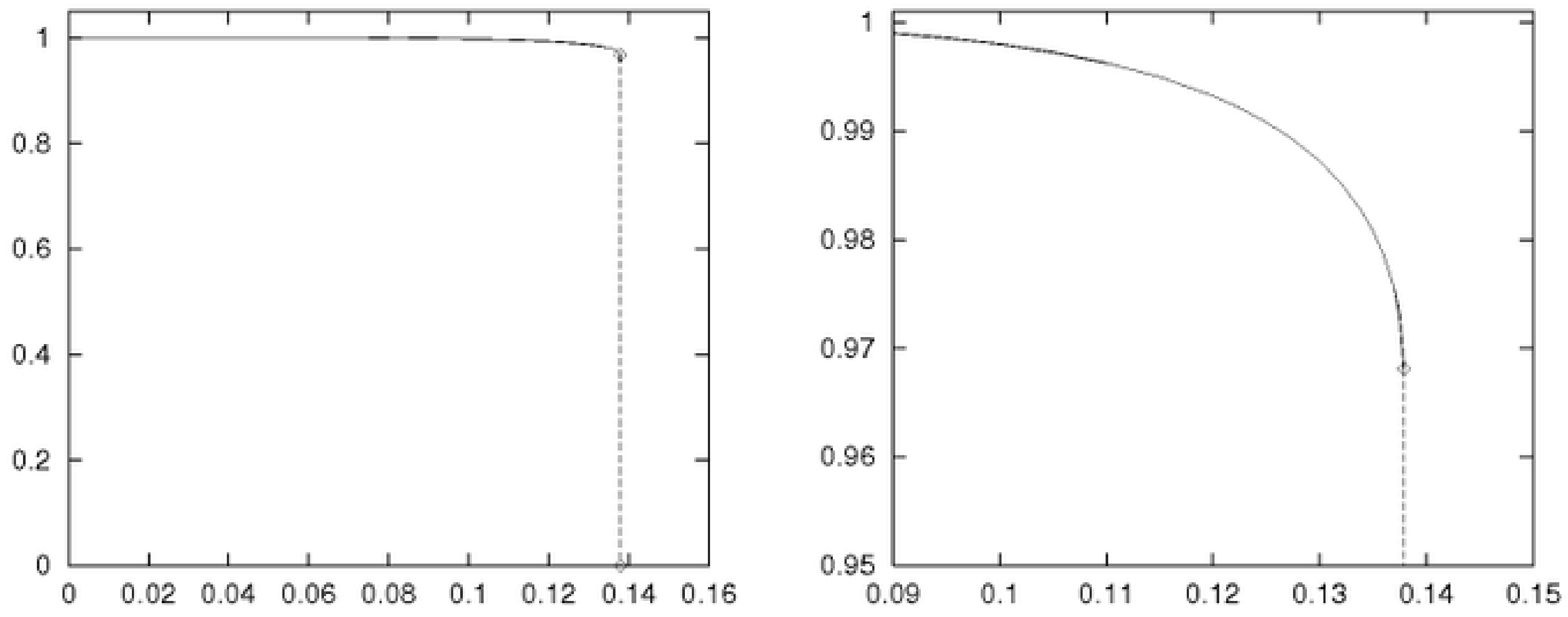
minimize $\tilde{F}$

$\therefore$ $\exists$ fixed points

& dynamics

Capacity of Hopfield network

Overlap between a desired memory and the nearest stable state



N/I N/I

Capacity for random patterns: $N_{\text{crit}} = 0.138\,I$

0.24 bits per weight.

I neurons

N patterns

$\#$parameters $\sim \frac{I^2}{2}$

I neurons

N patterns

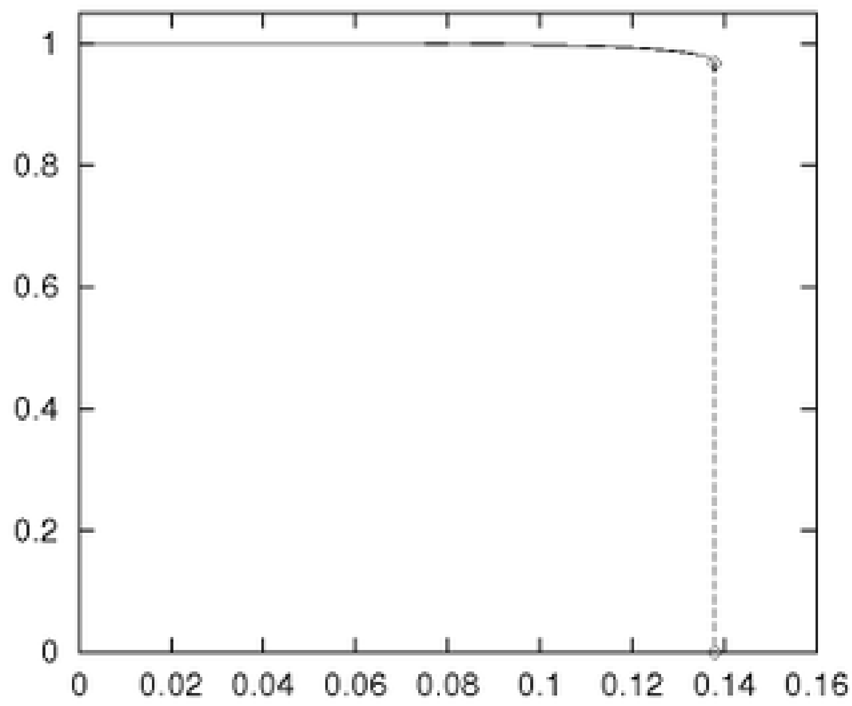
#parameters $\sim \frac{I^2}{2}$

$N_{crit} = 0.138\, I$

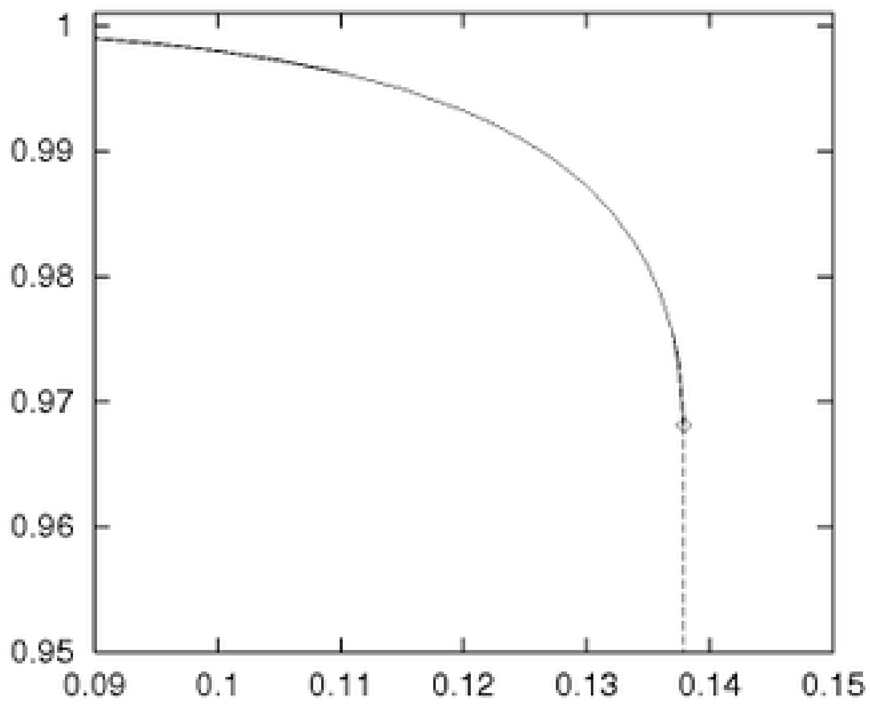
#bits $\simeq N_{crit} \times I \simeq 0.138\, I^2$

Capacity of Hopfield network

Overlap between a desired memory and the nearest stable state



N/I



N/I

Capacity for random patterns: $N_{\mathrm{crit}} = 0.138\, I$

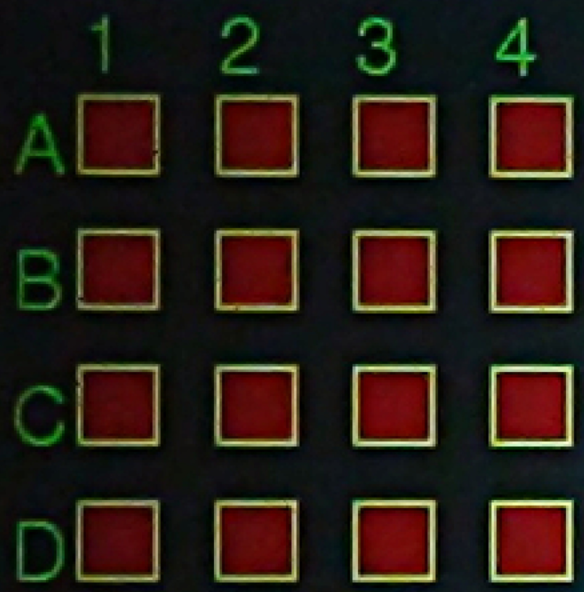
0.24 bits per weight.

The Hopfield network

- Why does Hopfield net have stable states at all?
- What is the capacity of the Hopfield net?
- What else can we do with them?

Energy: 0.0 Actvn: 0.0

Bias: < > -8 penalty1: < > 5 penalty2: < > 7



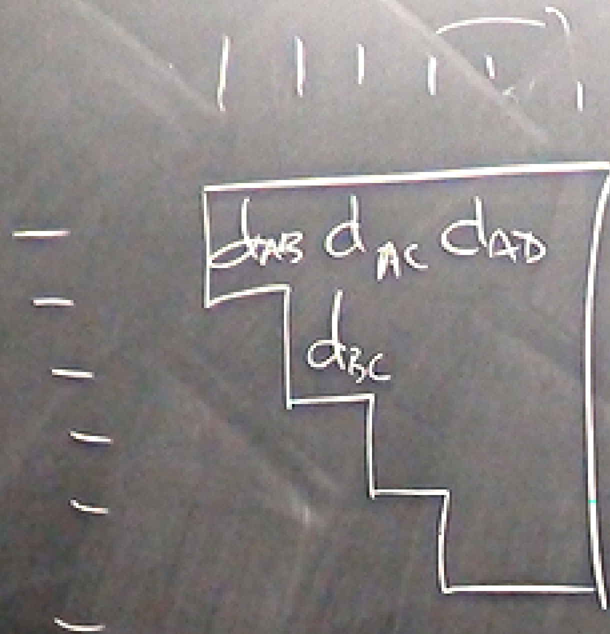
	A	B	C	D
A		1	4	6
B	1		1	4
C	4	1		1
D	6	4	1	

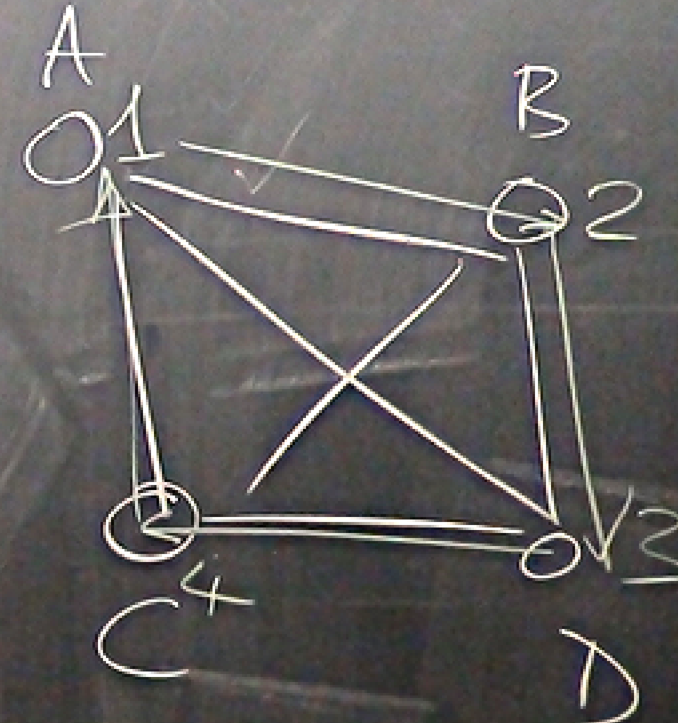
Negative Weights

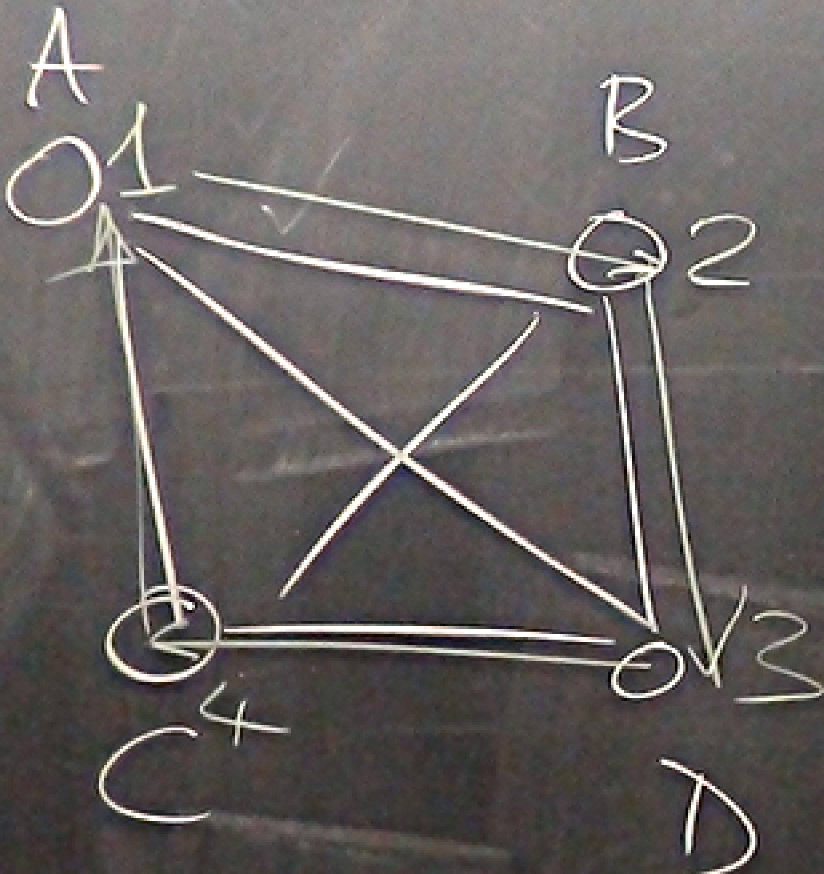
	7	7	7	5	1	0	1	5	4	0	4	5	6	0	6
7		7	7	1	5	1	0	4	5	4	0	6	5	6	0
7	7		7	0	1	5	1	0	4	5	4	0	6	5	6
7	7	7		1	0	1	5	4	0	4	5	6	0	6	5
5	1	0	1		7	7	7	5	1	0	1	5	4	0	4
1	5	1	0	7		7	7	1	5	1	0	4	5	4	0
0	1	5	1	7	7		7	0	1	5	1	0	4	5	4
1	0	1	5	7	7	7		1	0	1	5	4	0	4	5
5	4	0	4	5	1	0	1		7	7	7	5	1	0	1
4	5	4	0	1	5	1	0	7		7	7	1	5	1	0
0	4	5	4	0	1	5	1	7	7		7	0	1	5	1
4	0	4	5	1	0	1	5	7	7	7		1	0	1	5
5	6	0	6	5	4	0	4	5	1	0	1		7	7	7
6	5	6	0	4	5	4	0	1	5	1	0	7		7	7
0	6	5	6	0	4	5	4	0	1	5	1	7	7		7
6	0	6	5	4	0	4	5	1	0	1	5	7	7	7	

Dismiss

TSP







NP complete problem

NP complete problem.

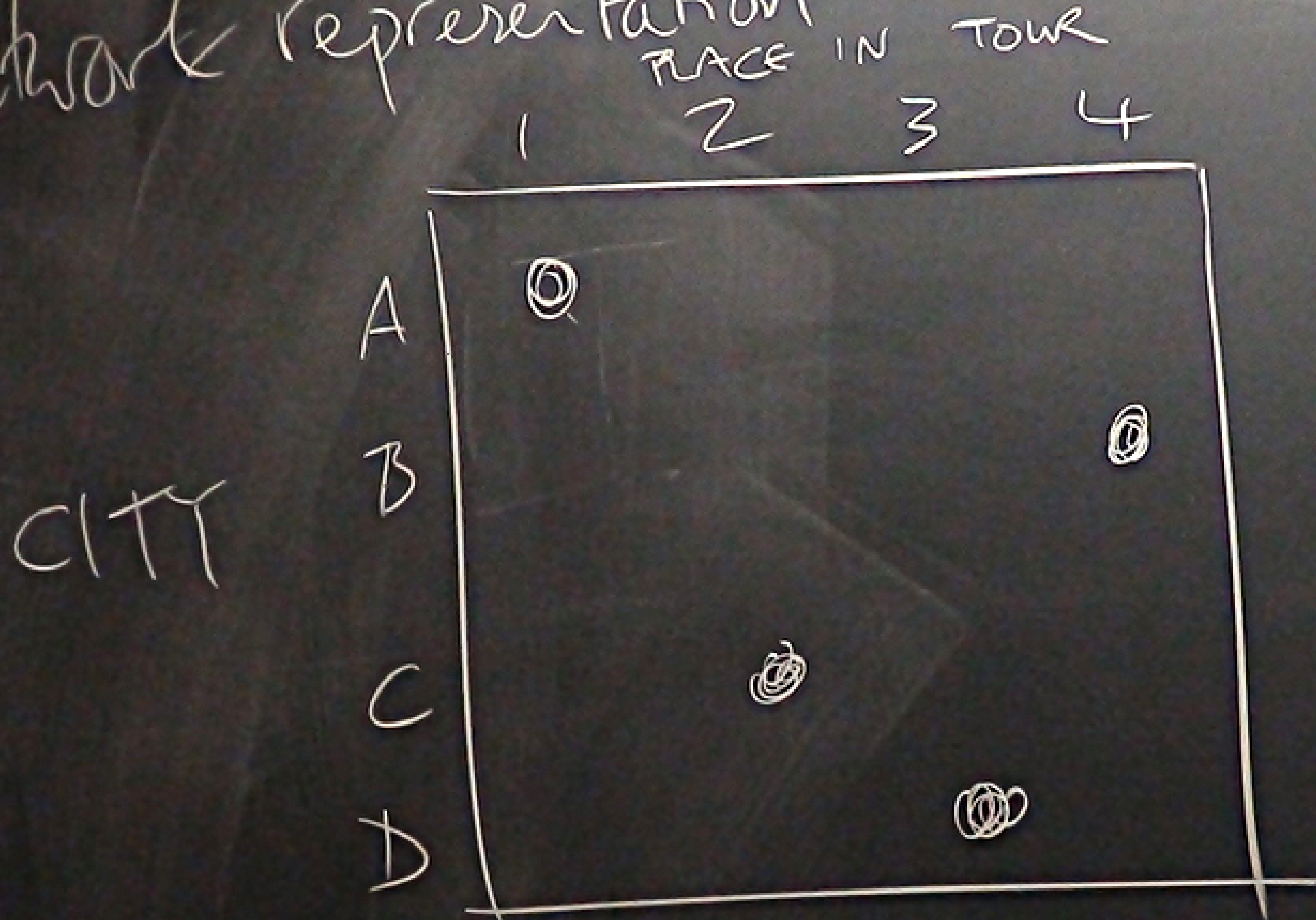
goal:

find a permutation

A, B, C, D

that minimizes total distance

Network representation



ACDB

NP complete problem.

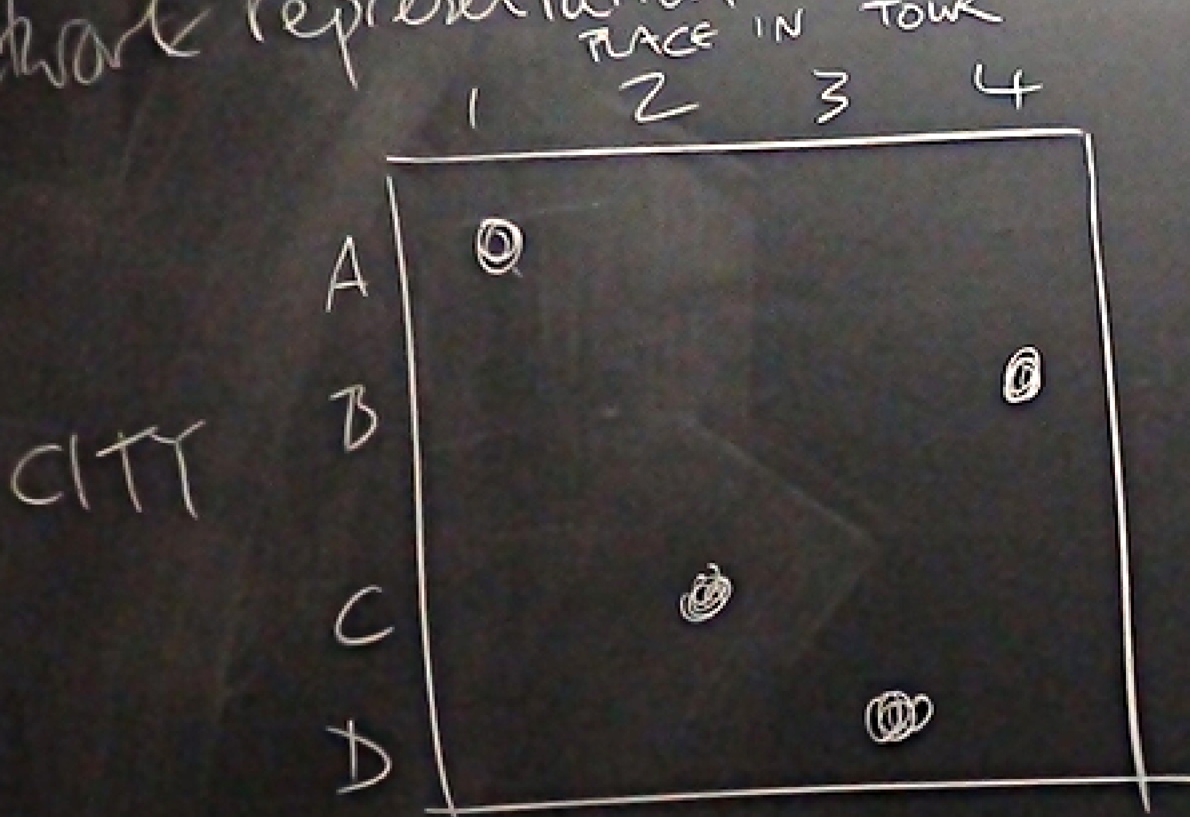
goal:

find a permutation

A, B, C, D

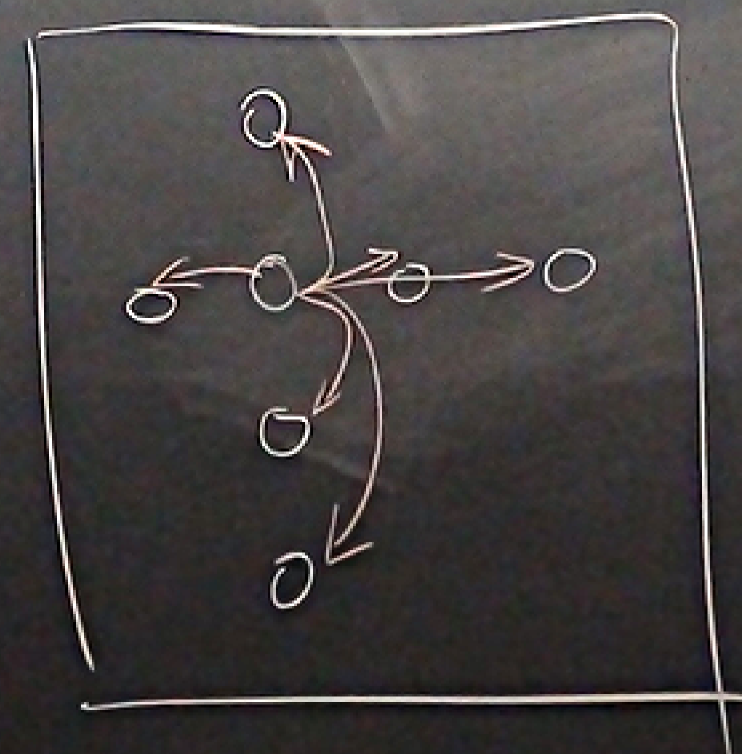
that minimizes total distance

Network representation

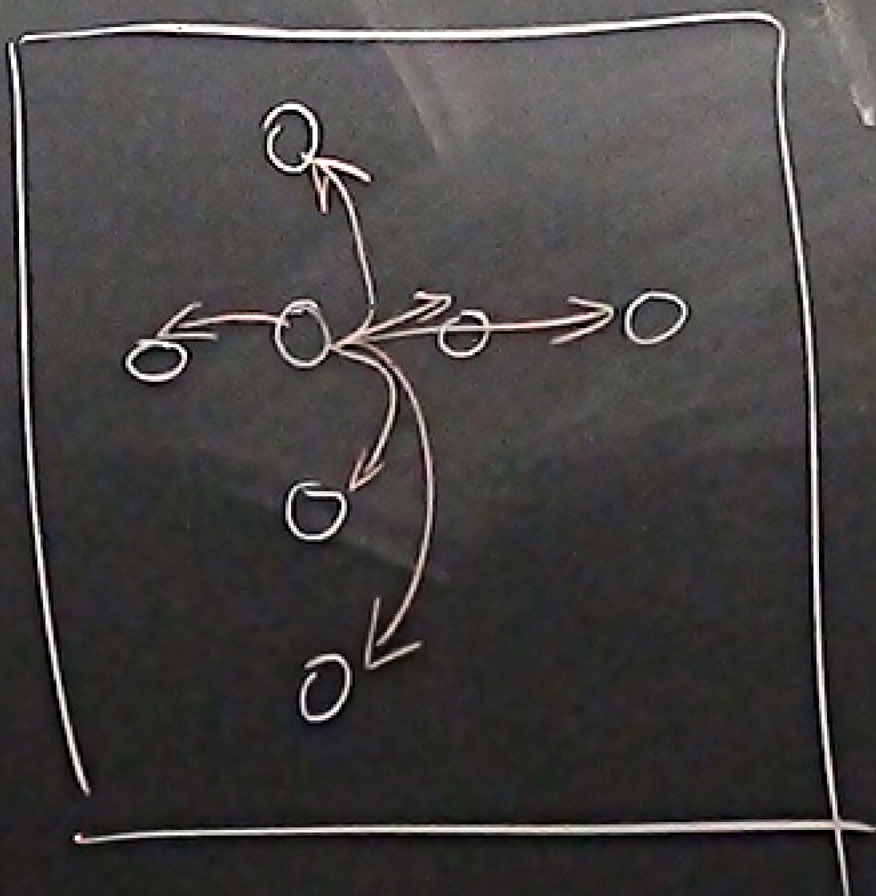


ACDB

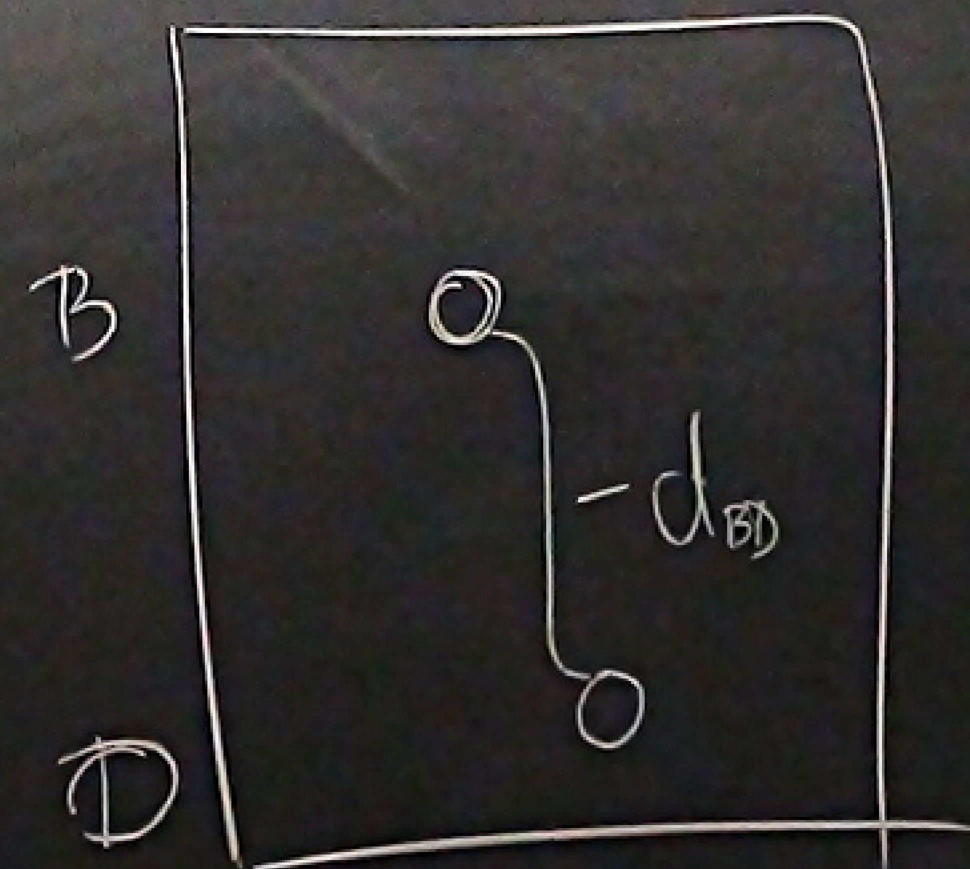
Permutation



Permutation



2 3



TSP Demonstration

Energy: 0.0 Actvn:

Bias: < > -8 penalty1: < > 5 penalty2: < > 7

	1	2	3	4
A				
B				
C				
D				

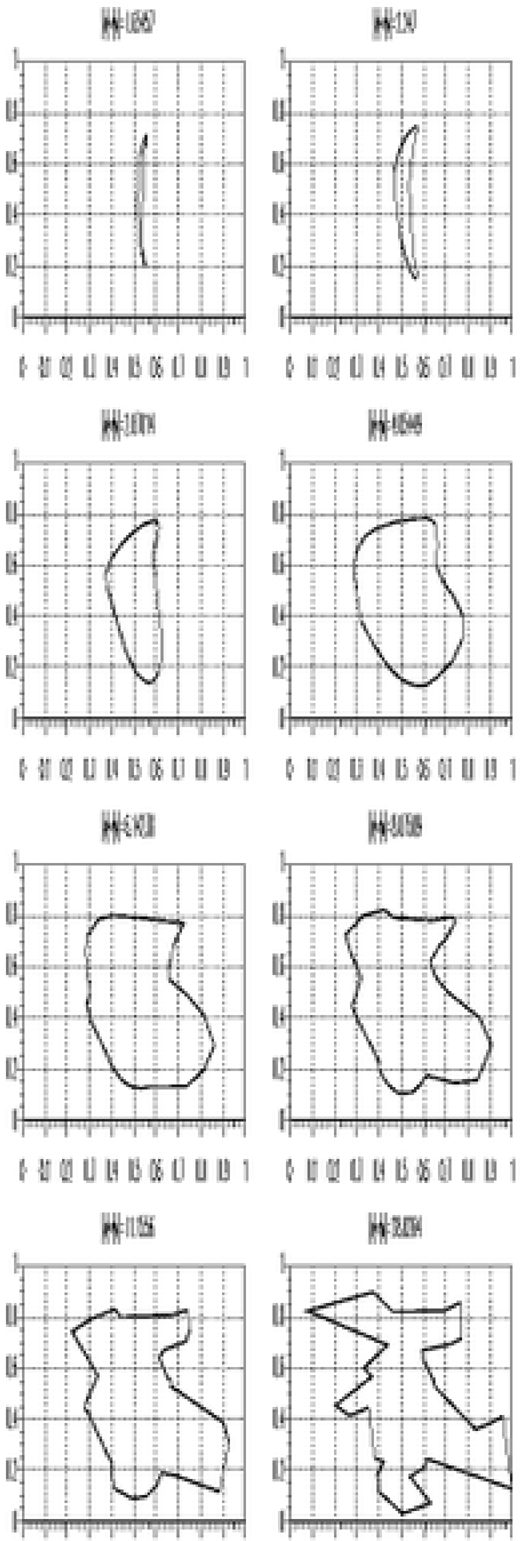
	A	B	C	D
A		1	4	6
B	1		1	4
C	4	1		1
D	6	4	1	

Negative Weights

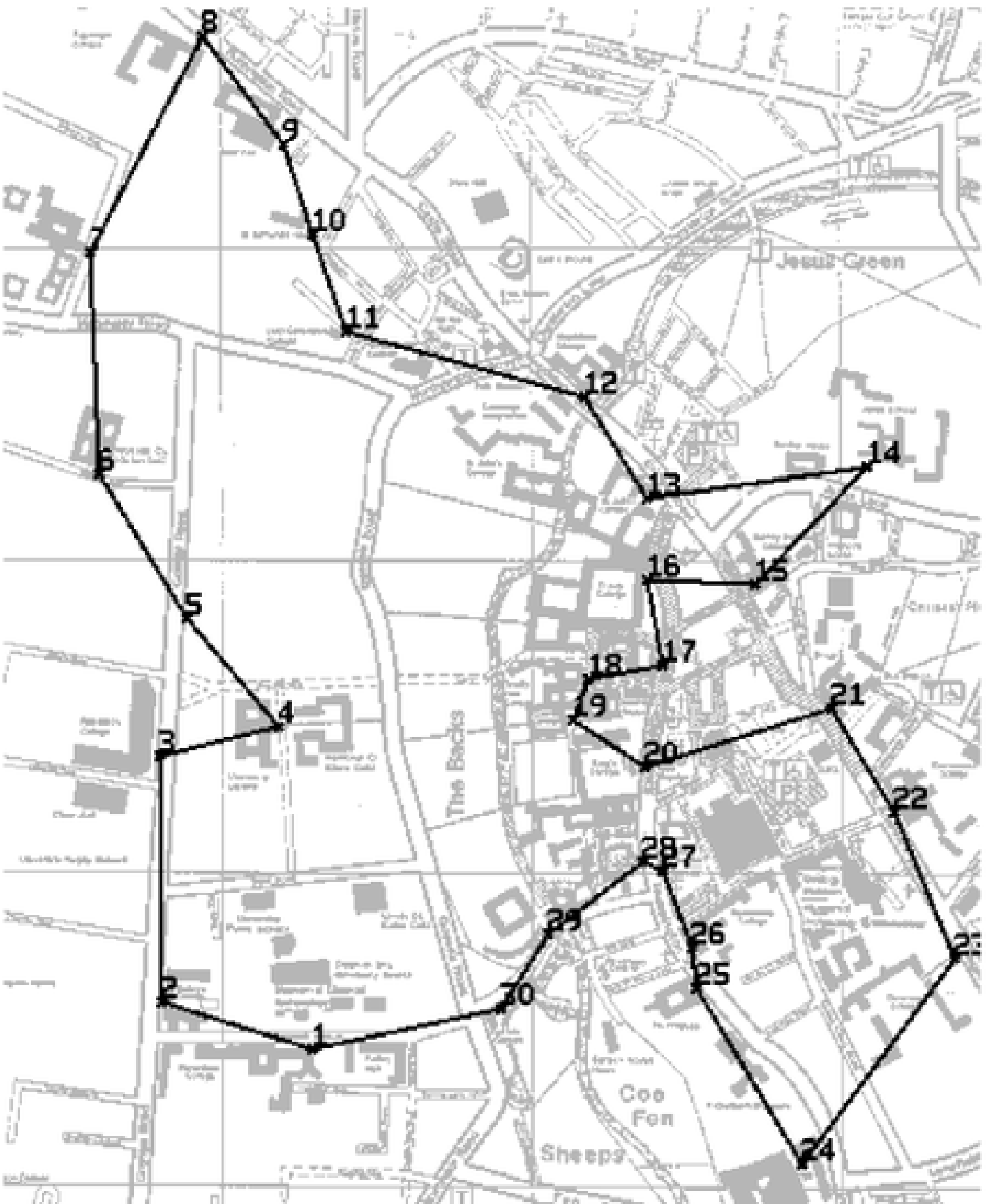
	7	7	7	5	1	0	1	5	4	0	4	5	6	0	6
7		7	7	1	5	1	0	4	5	4	0	6	5	6	0
7	7		7	0	1	5	1	0	4	5	4	0	6	5	6
7	7	7		1	0	1	5	4	0	4	5	6	0	6	5
5	1	0	1		7	7	7	5	1	0	1	5	4	0	4
1	5	1	0	7		7	7	1	5	1	0	4	5	4	0
0	1	5	1	7	7		7	0	1	5	1	0	4	5	4
1	0	1	5	7	7	7		1	0	1	5	4	0	4	5
5	4	0	4	5	1	0	1		7	7	7	[illegible]	1	0	1
4	5	4	0	1	5	1	0	7		7	7	1	5	1	0
0	4	5	4	0	1	5	1	7	7		7	0	1	5	1
4	0	4	5	1	0	1	5	7	7	7		1	0	1	5
5	6	0	6	5	4	0	4	5	1	0	1		7	7	7
6	5	6	0	4	5	4	0	1	5	1	0	7		7	7
0	6	5	6	0	4	5	4	0	1	5	1	7	7		7
6	0	6	5	4	0	4	5	1	0	1	5	7	7	7	

Dismiss

Hopfield networks for optimization



"The Travelling Scholar Problem" (Sree Aiyer, 1991)



Q: What is this?

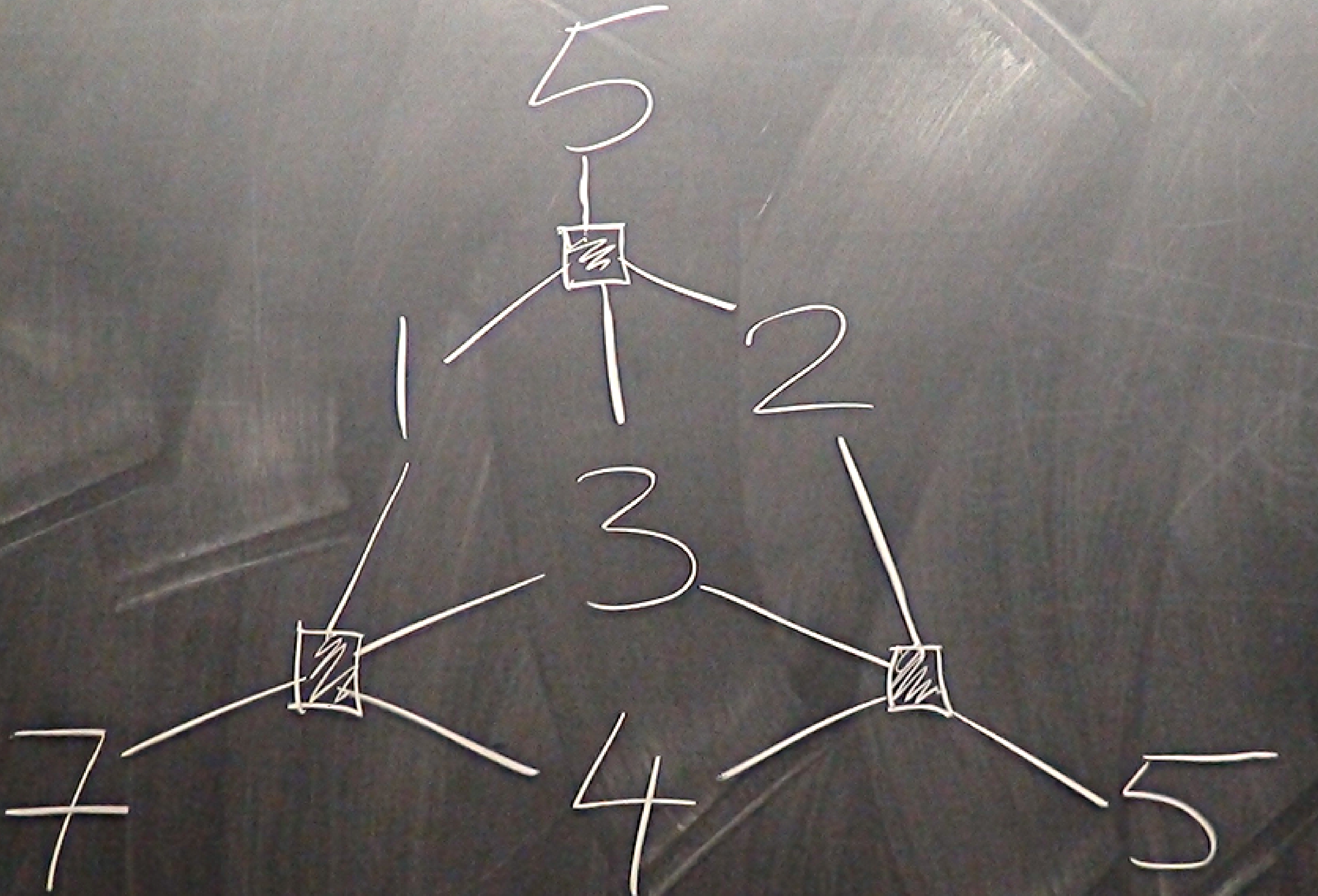
$$P_0(\mathbf{x}) = \frac{1}{Z_0} e^{\beta[x_1x_2x_3x_5 + x_2x_3x_4x_6 + x_1x_3x_4x_7]}$$

where $\mathbf{x} = (x_1, x_2, x_3, \ldots, x_7)$, $x_n \in \pm 1$

and $\beta =$ large

& What is this?

$$P_1(\mathbf{x}) = \frac{1}{Z_1} e^{\beta\,[x_1x_2x_3x_5 + x_2x_3x_4x_6 + x_1x_3x_4x_7] + \sum_{n=1}^{N} b_n x_n}$$


5
1
2
3
7
4
5

Q: What is this?

$$P_0(\mathbf{x}) = \frac{1}{Z_0} e^{\beta[x_1 x_2 x_3 x_5 + x_2 x_3 x_4 x_6 + x_1 x_3 x_4 x_7]}$$

where $\mathbf{x} = (x_1, x_2, x_3, \ldots, x_7)$, $x_n \in \pm 1$

and $\beta =$ large

& What is this?

$$P_1(\mathbf{x}) = \frac{1}{Z_1} e^{\beta\,[x_1 x_2 x_3 x_5 + x_2 x_3 x_4 x_6 + x_1 x_3 x_4 x_7] + \sum_{n=1}^{N} b_n x_n}$$

(7,4) Hamming Code

$$\underline{s} \xrightarrow{\text{Encode}} \underline{t} \xrightarrow{\text{BSC}} \underline{r}$$

$$P_0(\underline{t})$$

$$\underline{r} = \underline{t} \oplus \underline{n}$$

$$P(\underline{t}|\underline{r}) = \frac{\overset{\prod_{1}^{N} P(r_n|t_n)}{\overset{\uparrow}{P(\underline{r}|\underline{t})}}\;\overset{P_0(\underline{t})}{\overset{\uparrow}{P(\underline{t})}}}{Z}$$

$$\prod_{1}^{N} P(r_n|t_n) \qquad P_0(\underline{t})$$

$$\frac{P(\underline{r}|\underline{t})\,P(\underline{t})}{Z}$$

$$\frac{e^{b_n t_n}}{\tilde{Z}_n}$$

Q: What is this?

$$P_0(\mathbf{x}) = \frac{1}{Z_0} e^{\beta[x_1x_2x_3x_5 + x_2x_3x_4x_6 + x_1x_3x_4x_7]}$$

where $\mathbf{x} = (x_1, x_2, x_3, \ldots, x_7)$, $x_n \in \pm 1$

and β = large

& What is this?

$$P_1(\mathbf{x}) = \frac{1}{Z_1} e^{\beta\,[x_1x_2x_3x_5 + x_2x_3x_4x_6 + x_1x_3x_4x_7] + \sum_{n=1}^{N} b_n x_n}$$

$$P(\underline{x}) = \frac{e^{\sum_{i<j} w_{ij} x_i x_j + \sum \theta_i x_i}}{Z}$$

approx this using a VFE min $Q(\underline{x})$

Idea

$$P_1(\mathbf{x}) = \frac{1}{Z_1} e^{\beta\left[x_1x_2x_3x_5 + x_2x_3x_4x_6 + x_1x_3x_4x_7\right] + \sum_{n=1}^{N} b_n x_n}$$

- Solve the decoding problem

$$\max_{\mathbf{x}} P_1(\mathbf{x})$$

using approximate methods?

eg, Approximate $P_1(\mathbf{x})$ by $Q(\mathbf{x})$
using a variational method

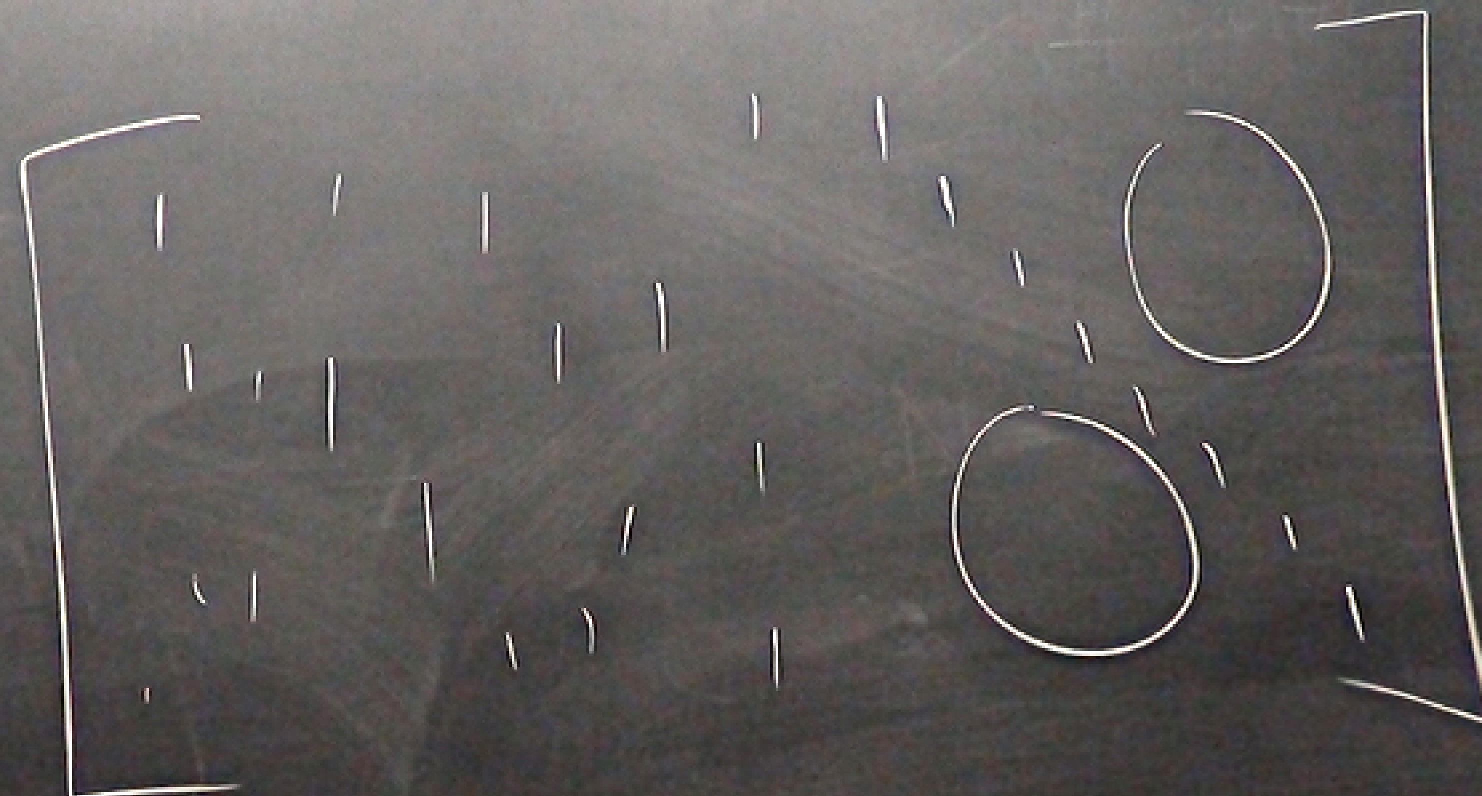
Electronics Letters, March 1995

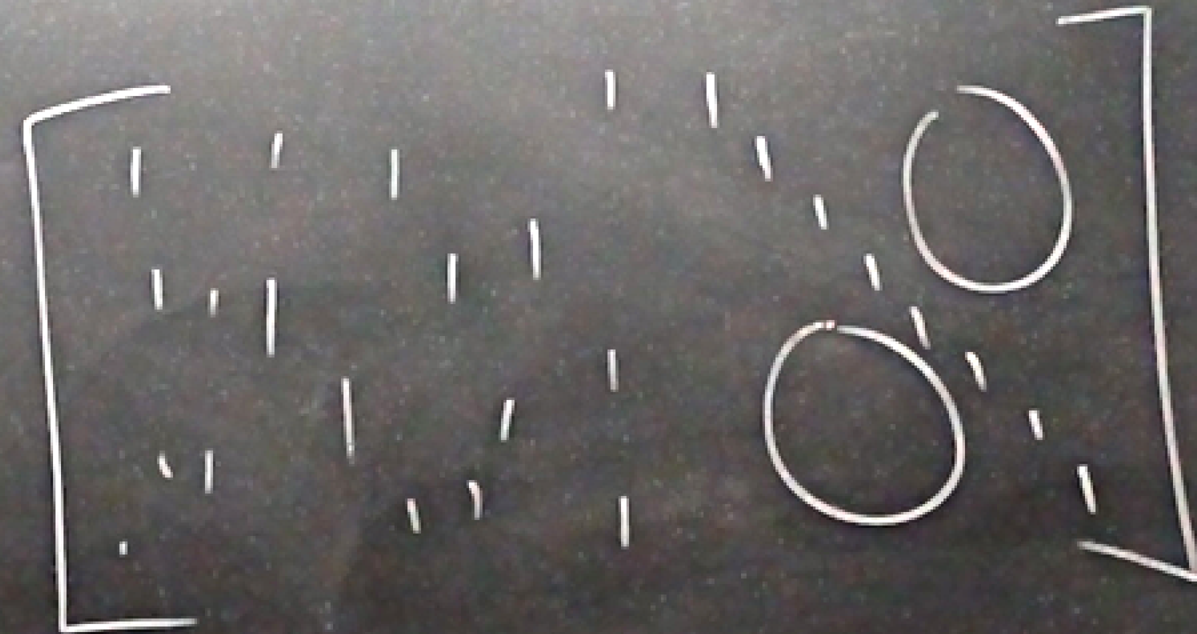
Free energy minimisation algorithm for decoding and cryptanalysis

D.J.C. MacKay

Indexing terms: Decoding, Cryptography

An algorithm is derived for inferring a binary vector **s** given nois
observations of **As** modulo 2, where **A** is a binary matrix. Th
binary vector is replaced by a vector of probabilities, optimised b
free energy minimisation. Experiments on the inference of th
state of a linear feedback shift register indicate that this algorithr
supersedes the Meier and Staffelbach polynomial algorithm.

$H =$ 

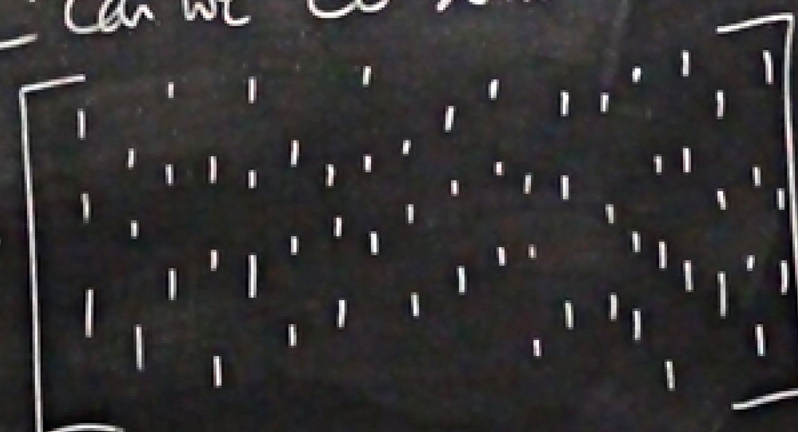
$H =$ 

$Q(\underline{x})$

can we do better than VFEmin?

$H =$ 

can we do better than this

$H =$ 

can we do better than VFE_{min}?

$H =$ 

can we do bette than VFEmin? ✓

sum-product algorithm

Electronics Letters, August 1996

Near Shannon limit performance of low density parity check codes

D.J.C. MacKay and R.M. Neal

[Low Density Parity Check Codes: Gallager 1962]

Indexing terms: Probabilistic decoding, Error correction codes

The authors report the empirical performance of Gallager's lo
density parity check codes on Gaussian channels. They show tha
performance substantially better than that of standar
convolutional and concatenated codes can be achieved; indeed th
performance is almost as close to the Shannon limit as that c
turbo codes.


Error probability
Achievable
C
1
Rate

Low Density Parity Check Code

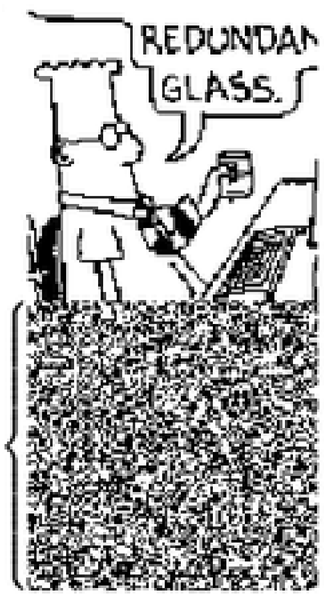
Ch 47

We demonstrate a large code that encodes $K = 10000$ source bits into $N = 20000$ transmitted bits.

Each parity bit depends on about 5000 source bits.

The encoder is derived from a very sparse 10000×20000 matrix $\mathbf{H}$ with three 1s per column.

TRANSMITTED:

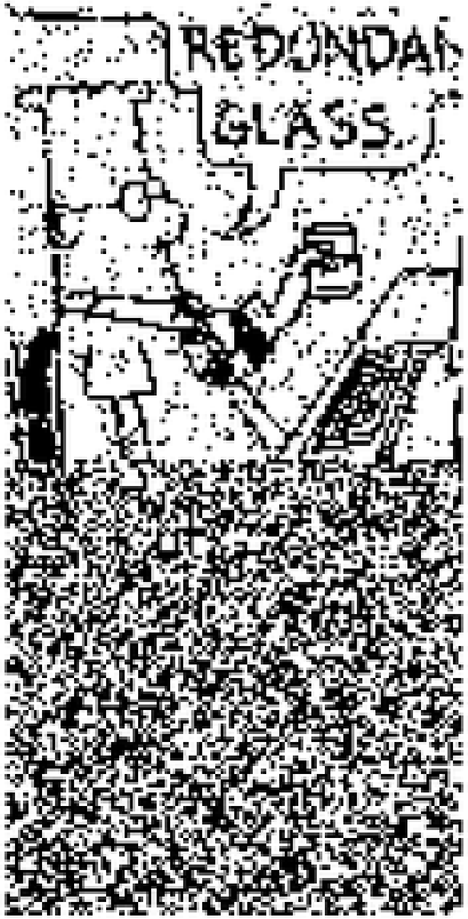


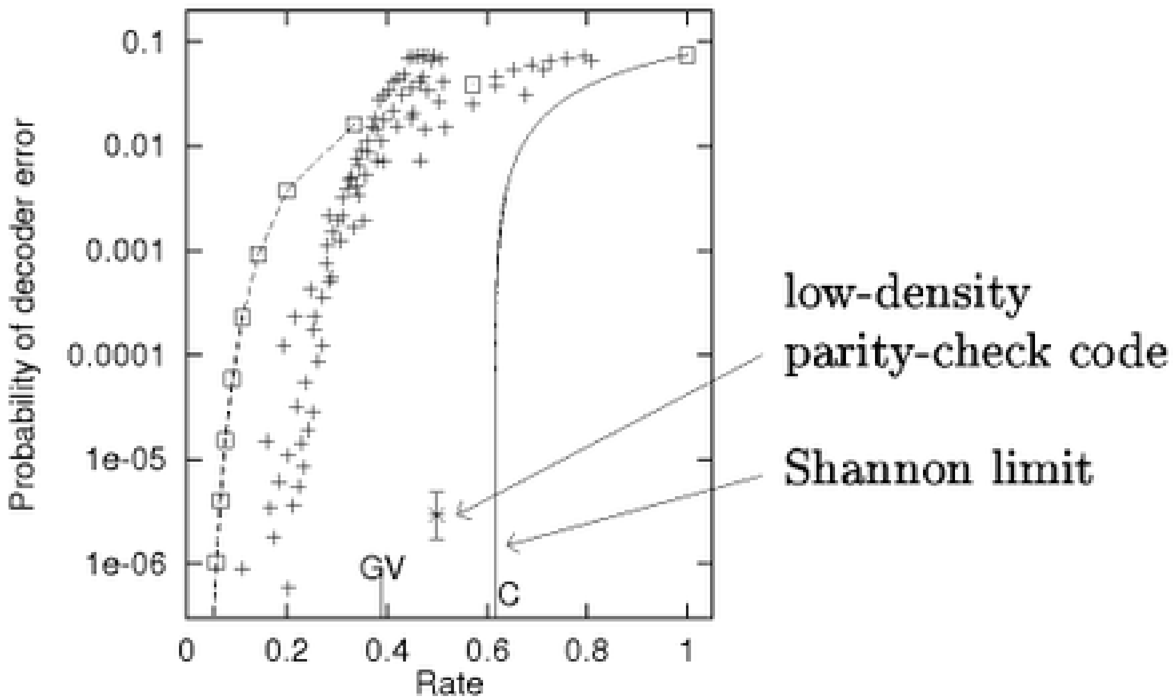
parity bits

$\mathbf{H} \;=$ 

Low Density Parity Check Code (f = 7.5%)

Iterative probabilistic decoding





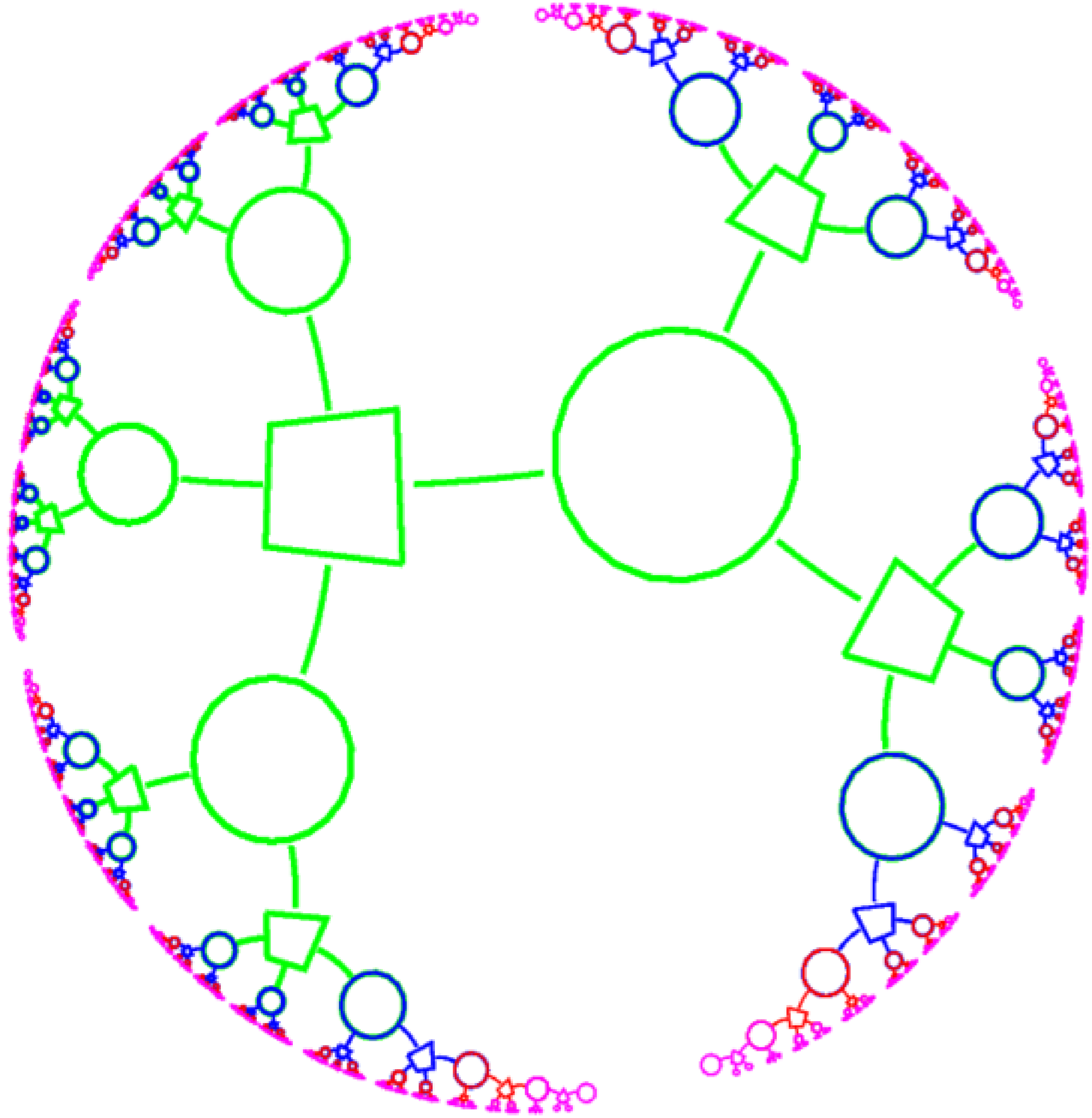
International Symposium on Information Theory 1997

One paper on low-density parity-check codes

International Symposium on Information Theory 2002

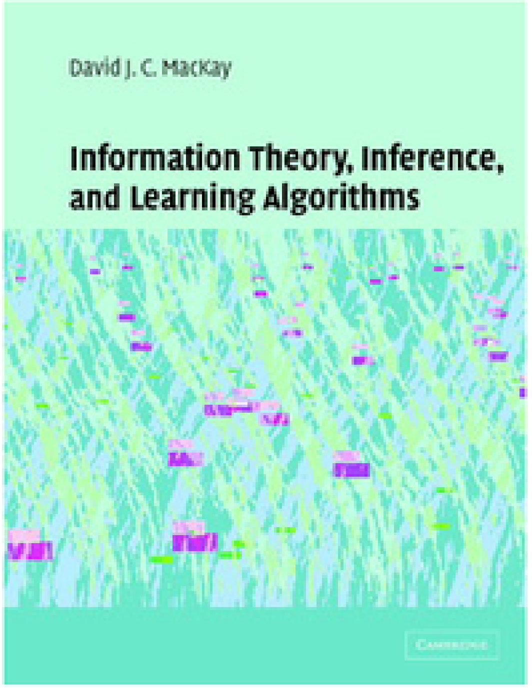
6 sessions on LDPC codes; 10 on sparse-graph codes

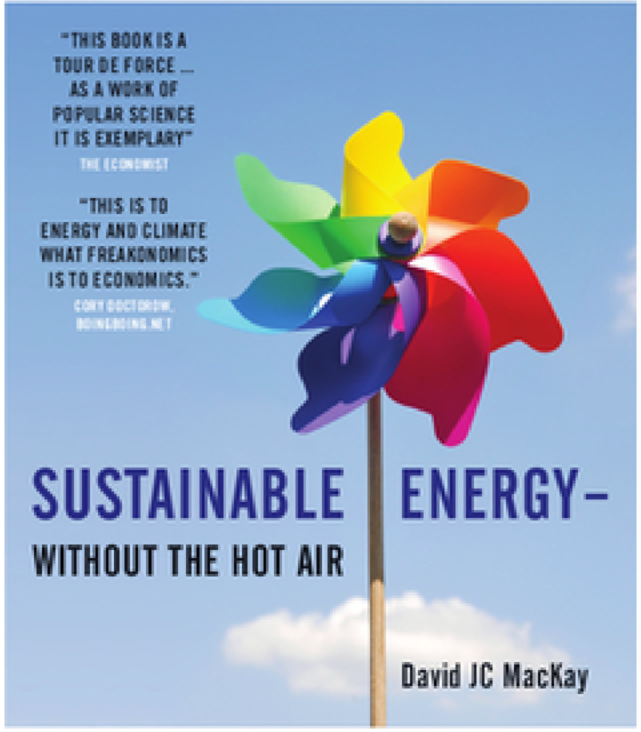
Analysis of LDPC Codes I 755, 511, 567, 263 R. Michael TANNER	Iterative Decoding I 805, 465, 240, 467 Tom RICHARDSON	Decoding Techniques I 485, 424, 649, 181 Hans-Andrea LOELIGER	Construction of LDPC Codes I 725, 378, 787, 711 Frank R. KSCHISCHANG	Convolutional Codes I 591, 115, 358, 631 Joachim ROSENTHAL
Analysis of LDPC Codes II 142, 276, 123, 410 Evangelos ELEFTHERIOU	Iterative Decoding II 351, 468, 329 Kamil ZIGANGIROV	Coding and Statistical Physics 325, 135, 129 David FORNEY	Construction of LDPC Codes II 157, 562, 87, 31 Marc P. C. FOSSORIER	Convolutional Codes II 322, 750, 420, 118 Ajay DHOLAKIA
Turbo Code Design 417, 677, 535, 166 Johannes B. HUBER	Applications of Iterative Decoding I 743, 502, 667, 443 Tolga DUMAN		Soft-In Soft-Out Decoding 706, 307, 439, 233 Joachim HAGENAUER	Decoding Methods for LDPC Codes 215, 659, 596, 798 Thomas E FUJA
Analysis of Turbo Codes 266, 560, 507, 228 Daniel J. COSTELLO	Applications of Iterative Decoding II 565, 561, 614, 237 Steven W. MCLAUGHLIN		Decoding Algorithms 92, 303, 685, 499 Andrew SINGER	Construction of LDPC Codes III 599, 510, 146, 647, 585 Thomas MITTELHOLZER



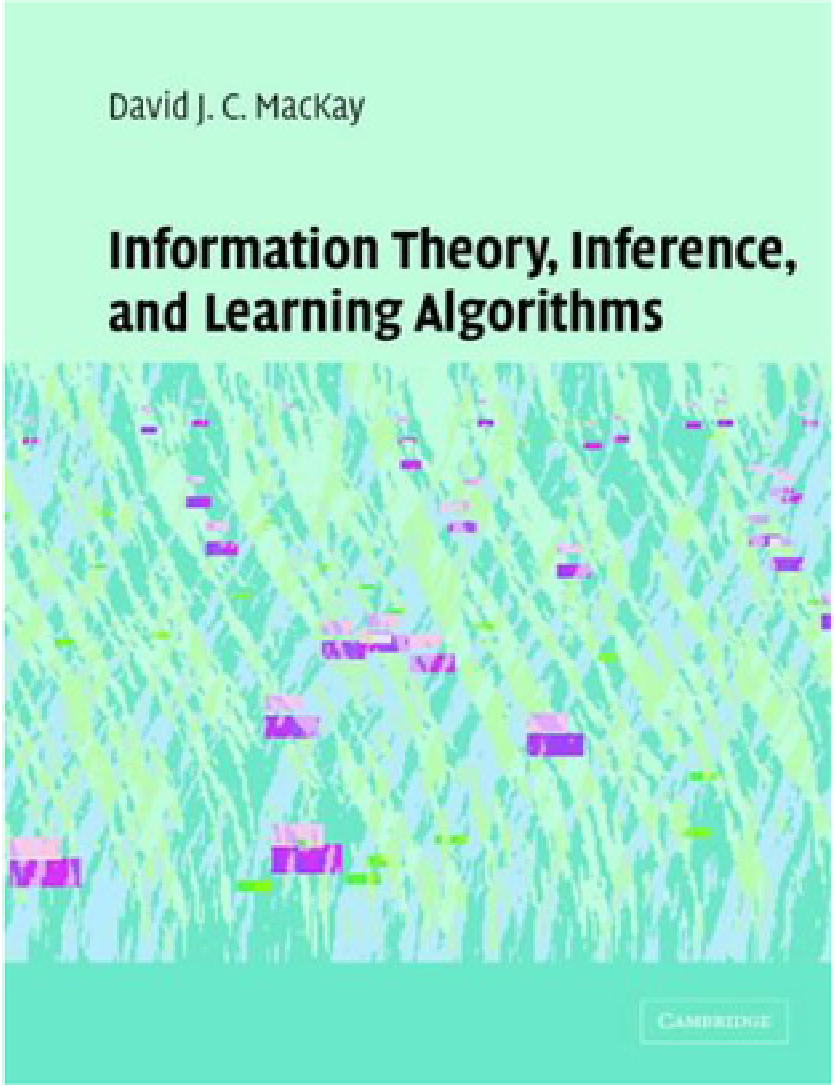
Information Theory

Sustainable Energy

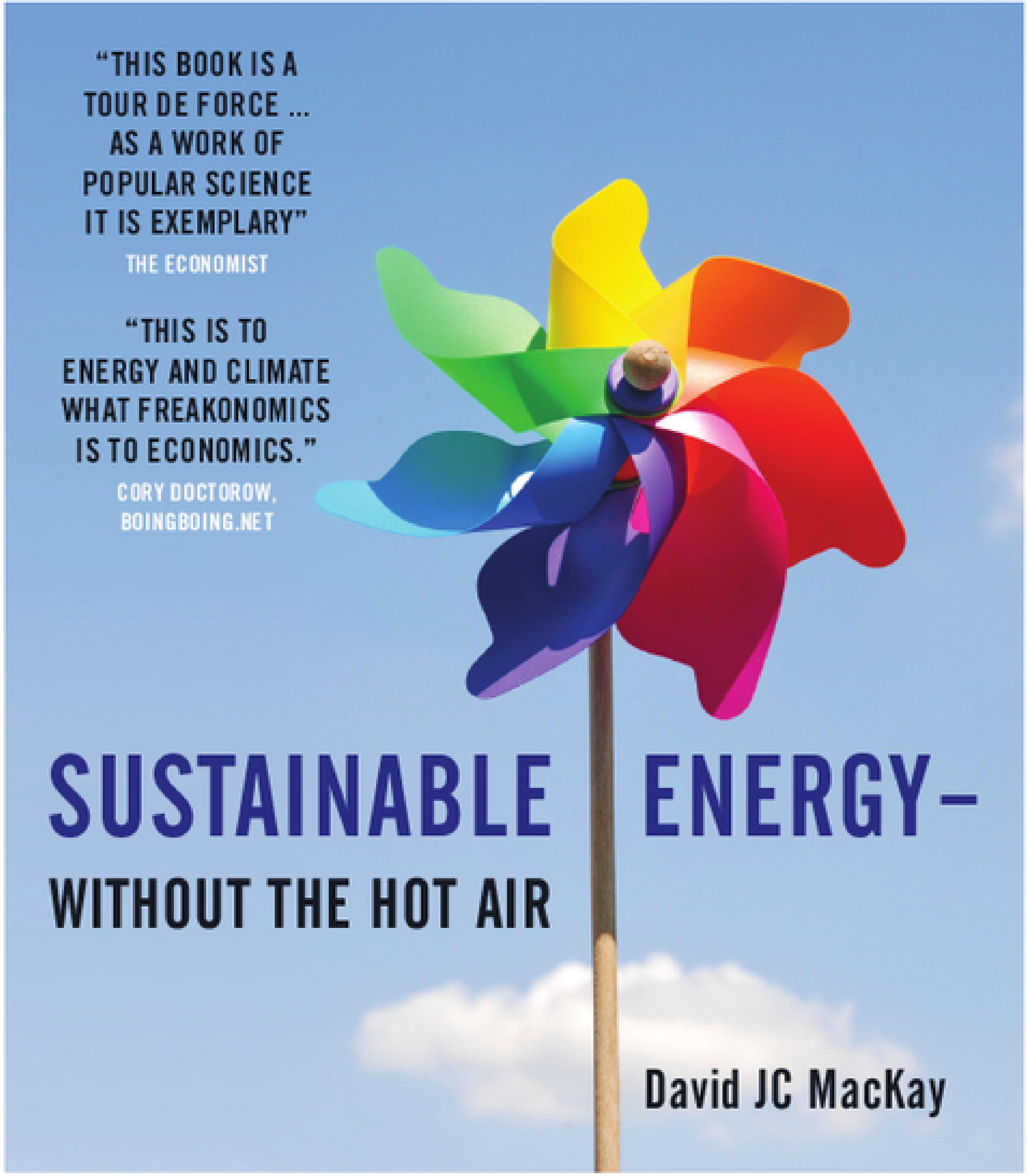




Both available free online



www.inference.phy.cam.ac.uk/itila/



www.withouthotair.com