TENSOR NETWORKS ♡ LOGIC

Tensor networks since Aristotle

Logic: the study of valid/invalid reasoning

keywords: (deductive) reasoning, inference, information flow

Logic overlaps with all the following disciplines:

- philosophy (how to reason about ethics, government, ...)
- linguistics (which sentences are valid given a grammar/syntax)
- mathematics (axiomatization, theorem proving, game theory)
- computer science (computability, complexity)

In the western world, rules of logical inference were first introduced by Aristotle the Stageirite (384 – 322 BCE)

→ syllogistic logic

Logical inference:

"All men are mortal" ← premises (→ universal)
"Socrates is a man" (→ existential)
___
"Socrates is mortal" ← conclusion

propositions — can be true or false

Propositional logic

Chrysippus of Soli (c.279 – c.206 BCE); Boole (1815-1864)

- atomic propositions or variables: e.g. $p \in \{0, 1\}$ elementary propositions
  "false" "true"
- logical operators or relations:

symbol	read	name
$\neg$	NOT	negation
$\wedge$	AND	conjunction
$\vee$	OR	disjunction

An assignment of n variables:

$\vec{x} \in \{0,1\}^n$, $\vec{x} = (x_1, x_2, \dots, x_n)$

n variables $\to$ 2^n assignments

Truth tables:

φ	$\neg\varphi$
0	1
1	0

φ	ψ	$\varphi \wedge \psi$	$\varphi \vee \psi$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Logical tensors:

Relation R:

$$aRb = \begin{pmatrix} (a=0)R(b=0) & (a=0)R(b=1) \\ (a=1)R(b=0) & (a=1)R(b=1) \end{pmatrix}$$

e.g.:

$$a \vee b \Rightarrow \begin{array}{cc} & \begin{array}{cc} b= & \\ 0 & 1 \end{array} \\ a = \left\{ \begin{array}{c} 0 \\ 1 \end{array} \right. & \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \end{array}$$

$$a \vee b \vee c \Rightarrow \begin{array}{cc} & \begin{array}{cc} c= & \\ 0 & 1 \end{array} \\ a = \left\{ \begin{array}{c} 0 \quad b = \left\{ \begin{array}{c} 0 \\ 1 \end{array} \right. \\ \hline 1 \quad b = \left\{ \begin{array}{c} 0 \\ 1 \end{array} \right. \end{array} \right. & \left(\begin{array}{cc} 0 & 1 \\ 1 & 1 \\ \hline 1 & 1 \\ 1 & 1 \end{array} \right) \end{array}$$

Conjunctive normal form

$$F = \bigwedge_{i=1}^{m} C_i, \quad C_i = (x_j \vee x_k \vee \dots)$$

formula, clause, possibly negated

Boolean satisfiability (SAT) $P \overset{?}{=} NP$

NP-completeness

LOGIC $\Leftrightarrow$ COMPUTATION

each clause added eliminates assignments that satisfy the formula

Question: is there assignment $\vec{x}$ that makes $F = 1$?

Two-element Boolean algebra — turning logic into (multi-)linear algebra

$1+0=1$

$\boxed{1+1=1}$

$1\cdot 0 = 0\cdot 0 = 0$

$1\cdot 1 = 1$

Boolean semiring: no additive inverse — allows us to express propositional logic

$\longrightarrow \quad \vee \equiv +$

$\wedge \equiv \cdot$

a	b	$a\vee b$	$a+b$	$a\wedge b$	$a\cdot b$
0	0	0	0	0	0
0	1	1	1	0	0
1	0	1	1	0	0
1	1	1	1	1	1

✓

Composition of relations (shared variables) → inference! De Morgan (1806–1871)

E.g.: $F=(a\vee b)\wedge(\neg b\vee c)$ → Relationship between a and c?

Truth table:

a	b	c	$a\vee b$	$\neg b\vee c$	$(a\vee b)\wedge(\neg b\vee c)$
0	0	0	0	1	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	0	0
1	1	1	1	1	1

a and c cannot be simultaneously 0

$\Rightarrow \boxed{a\vee c}$

Logical tensors:

$$a=\begin{array}{c|cc} & b=0 & b=1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 1 \end{array} \cdot\; b=\begin{array}{c|cc} & c=0 & c=1 \\ \hline 0 & 1 & 1 \\ 1 & 0 & 1 \end{array} = a=\begin{array}{c|cc} & c=0 & c=1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 1 \end{array} \Rightarrow \boxed{a\vee c}$$

$$0 = 0\cdot 1 + 1\cdot 0$$

$$= \underbrace{((0\vee 0)}_{0}\wedge(1\vee 0))\vee((0\vee 1)\wedge\underbrace{(0\vee 0)}_{0})$$

(a, b, $\neg b$, c with $\neg b = 0$; a, b, $\neg b$, c with $\neg b = 1$)

Tensor contraction

$\equiv$!

Logical inference

Syllogistic logic

first formal system of inference (Aristotle)

- "All A are B" becomes: $\neg a \vee b$ universal premise

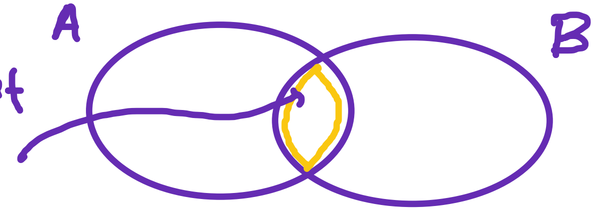
  there is no situation where a is true and b is not

y cannot be here

- "Some A are B" becomes: $a \wedge b$ existential premise

  there must exist at least one situation where a and b are simultaneously true

there must be at least one y here



Logical tensors:

$$\neg a \vee b: \quad \begin{array}{cc} & b= \\ & \begin{array}{cc} 0 & 1 \end{array} \\ a=\begin{array}{c} 0 \\ 1 \end{array} & \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \end{array}, \qquad a \wedge b: \quad \begin{array}{cc} & b= \\ & \begin{array}{cc} 0 & 1 \end{array} \\ a=\begin{array}{c} 0 \\ 1 \end{array} & \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \end{array}$$

"All men (a) are mortal (b)" $\longrightarrow$ $\neg a \vee b$

"Socrates (c) is a man (a)" $\longrightarrow$ $a \wedge c$

$$\Rightarrow F = (\neg a \vee b) \wedge (a \wedge c)$$

here we have grouped two propositions into a relation

need to infer: "Socrates is a man"

Inference:

$$\text{is Socrates?} \begin{array}{cc} & \text{is man?} \\ & \begin{array}{cc} 0 & 1 \end{array} \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \end{array} \cdot \text{is man?} \begin{array}{cc} & \text{is mortal?} \\ & \begin{array}{cc} 0 & 1 \end{array} \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \end{array} = \text{is Socrates?} \begin{array}{cc} & \text{is mortal?} \\ & \begin{array}{cc} 0 & 1 \end{array} \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \end{array}$$

contraction!

Morale: though TNs state-of-the-art tool, they formulate ancient rules of logical inference

Redundancy Let us consider the following example

$$F=(a\vee b)\wedge(b\vee c) \Rightarrow \begin{array}{c|cc} & b=0 & 1 \\ \hline a=0 & 0 & 1 \\ 1 & 1 & 1 \end{array} \cdot \begin{array}{c|cc} & c=0 & 1 \\ \hline b=0 & 0 & 1 \\ 1 & 1 & 1 \end{array} = \begin{array}{c|cc} & c=0 & 1 \\ \hline a=0 & 1 & 1 \\ 1 & 1 & 1 \end{array} \Rightarrow$$

a, c independent variable b redundant logical matrix has rank $=1$

↳ $\boxed{\text{rank decomposition reveals redundancy in a system of constraints}}$ role of SVD in TN algos

Enumeration: now let's allow $1+1=2$

$$F=(a\vee b)\wedge(b\vee c) \Rightarrow \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}\cdot\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$$

Now we can ask: How many assignments $\vec{x}$ make $F=1$?

↳ combinatorial enumeration complexity class #P; #P-complete; harder than NP

↳ weighted enumeration & probabilistic logic graphical models; stat mech of disordered systems

partition functions

↳ spin glasses → Nobel!

E.g.:

$$\underbrace{W}_{\text{transition/transfer tensor (matrix)}} \cdot \underbrace{\vec{p}}_{\text{probability distribution}} = \sum_{b=0}^{1} W_{ab}\, p_b = \begin{array}{c|cc} & b=0 & 1 \\ \hline a=0 & 1-p & p \\ 1 & p & 1-p \end{array} \begin{array}{c|c} & \\ \hline b=0 & p' \\ 1 & 1-p' \end{array} = \underbrace{\begin{array}{c|c} & \\ \hline a=0 & (1-p)p' + p(1-p') \\ 1 & p\cdot p' + (1-p)(1-p') \end{array}}_{\text{updated probability distribution}}, \quad \sum_b W_{ab} = 1 \;\; \forall a$$

$p, p' \in [0,1]$

↳ expresses Markov chain Monte Carlo

↳ transfer matrix methods (coarse-graining, renormalisation)

Partition function

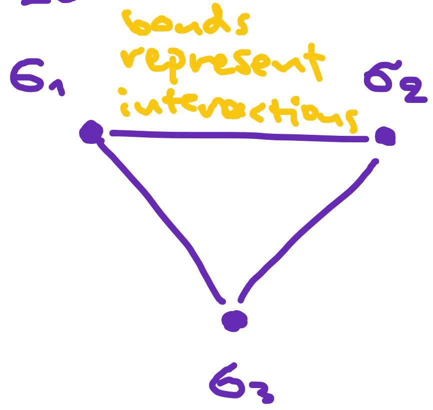
Key quantity in classical & quantum statistical mechanics

$$Z = \sum_{s} e^{-\beta E(s)} \equiv \text{tensor network!}$$

s ← (micro)states

$E(s)$: energy functional, $\beta = 1/k_B T$

Example: AFM Ising triangle



$\sigma_i = \pm 1$

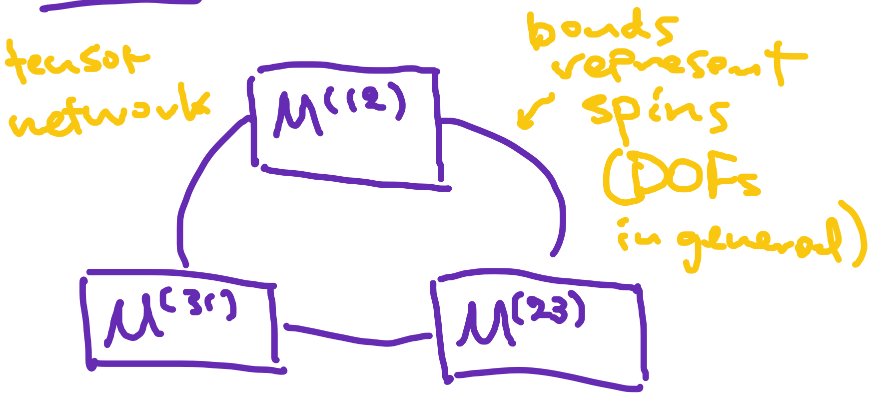
state: $s = \vec{\sigma} = (\sigma_1, \sigma_2, \sigma_3)$

energy: $E(\vec{\sigma}) = J\sigma_1\sigma_2 + J\sigma_2\sigma_3 + J\sigma_3\sigma_1$

$$\Rightarrow Z = \sum_{\sigma_1,\sigma_2,\sigma_3 = \pm 1} e^{-\beta E(\vec{\sigma})} = 2e^{-3\beta J} + 6e^{\beta J}$$

Define: tensor $M^{(ij)}$ for each bond, e.g.:

$$M^{(12)} = \begin{array}{c|cc} & \sigma_2 = +1 & -1 \\ \hline \sigma_1 = +1 & e^{-\beta J} & e^{\beta J} \\ -1 & e^{\beta J} & e^{-\beta J} \end{array}$$



$$Z = \mathrm{Tr}\left(M^{(12)} M^{(23)} M^{(31)}\right)$$

$$= \sum_{\sigma_1,\sigma_2,\sigma_3 = \pm 1} \left[M^{(12)}\right]_{\sigma_1\sigma_2} \left[M^{(23)}\right]_{\sigma_2\sigma_3} \left[M^{(31)}\right]_{\sigma_3\sigma_1}$$

$$= \mathrm{Tr}\begin{pmatrix} e^{-3\beta J} + 3e^{\beta J} & 3e^{-\beta J} + e^{3\beta J} \\ 3e^{-\beta J} + e^{3\beta J} & e^{-3\beta J} + 3e^{\beta J} \end{pmatrix} = 2e^{-3\beta J} + 6e^{\beta J} \quad ✓$$

What if spin σ_i participates in more than 2 terms? → COPY tensor!