Model:

$s_1, \dots, s_n$ binary nodes

Fully connected network

Use the following update rule:

$$s_i \begin{cases} \to 1 & \text{if } \sum_{j \neq i} w_{ij} \sigma_j > U_i \leftarrow 0 \\ \to 0 & \qquad\qquad\quad\;\; < U_i \end{cases}$$

Info storage:

want to train network on p states:

$S : [p] \to$ len n bit strings

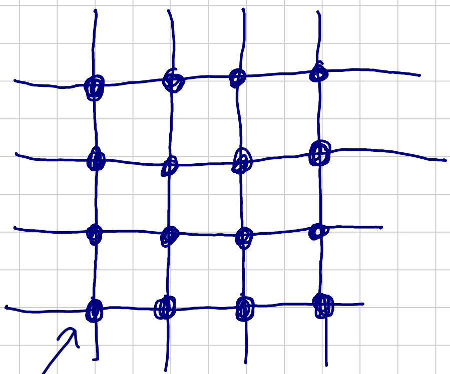
$$S(k) = \begin{pmatrix} s_1(k) \\ \vdots \\ s_n(k) \end{pmatrix}$$

$$w_{ij} = \sum_p (2S_i(p) - 1)(2S_j(p) - 1)$$

ie w_{ij} will be positive if s_i & s_j tend to have the same orientation in the stored states. Negative otherwise.

Ising Model Notes

Have nodes on a 2d lattice that are in 1 of 2 states. (± 1)



Nodes want to be in the same state as their neighbors. Not casual though.

σ_i

Model: $$P(\{\sigma_i\}) = \frac{\exp\{\beta \overbrace{\sum J_{ij}\,\sigma_i\,\sigma_j}^{\text{core}}\}}{C(\beta)}$$

$J_{ij} = 1$ if σ_i & σ_j are neighbors

So $\sum J_{ij}\,\sigma_i\,\sigma_j$ is the product of all neighbors in the network. Neighbors in the same state contribute $+1$ to sum. -1 for diff states.

So $P(\{\sigma_i\})$ is bigger when the neighbors have similiar orientations. β will give more weight to the sum.

Closing Notes:

Many interesting neuro interpts from the properties of the network:

- Can store $n \approx .15\,N$ memories before the error rate gets too high
- Overloading the system will make all memory states irretrievable
- Stored states that are close to each other (by Hamming Dist) will be confused.
- Can quantify error rate using the Normal noise from the $s \neq s'$ terms in $H_i^{s'}$.