# AGI Grand Unification

## □□

- □□□□□□□□ □□□□ □□□□□□□□□□□□□ Richard Sutton □□□□□□ □□□□□□□□□
- □ □□□□□□ □□□□□□□□□□□□□□□ □□□□□□ □□ □□ □□□ □□□□□□ □ AGI □□□□□□ = □□□□□□ □□□□□□□ □□□□□□□□□□□□□□□□□□□□□ AGI □□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ mathematical object□□□□□□□□□□□□□ □□□ □□□□□ □□□□□□□□□□□□□□□□□□□□□ □□ (sampling), □□□□□□□□□□□□□□□□□
- Hopfield □□□□□ □□□□□□ □□□□□ (energy landscape)□□□□□□□□□□ implicit □ □□□□□□ □□ Hopfield □□□ learning□□□□□□□□□□□□□□ □□□□□□□ Hopfield □□□□□□□□ □□ □□□ □□□□□□□□ Boltzmann machine□□□□□ EBM (Energy-Based Models).
- □□ "Hopfield Network is All You Need" □□[1]□□□ Hopfield □□□ state update rule □ Transformer □□[2]□ □□□□□□□□□□□□ Transformer□□□□□ Hopfield □□□□□□□□
- Transformer □ softmax □□□□□□ □□□□□□ "winner-takes-all" □□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□ □□□Transformer □□ □□□□□□□□□□□□□□□□□□□□□ AGI.

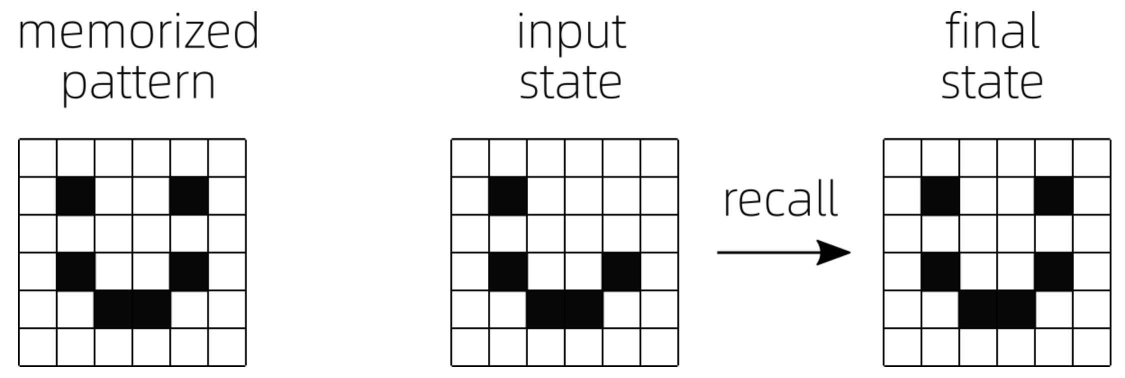
---

[1] □□ Eric Zeng □□□□□□□□□

[2] □□□□□ state update rule □□□ learning update rule. □□ □□ Hopfield □□□ □□ □□□ □□ □□ Hopfield □□□□□□□□□

# Hopfield Networks

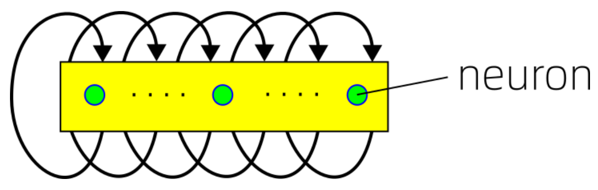
Hopfield networks are a type of particularly simple, fully-connected neural networks, capable of associative memory retrieval:




(1)

Memory patterns are determined by connection weights of the network, which define an energy function. The Hopfield network is actually the Ising model of statistical physics applied to neural networks.

One can flatten a Hopfield network and draw it like this:




(2)

Note that it is not a feed-forward network; its input and output neurons are the same.

## Classical Hopfield Network

Our notation follows *Hopfield Network is All You Need* [2021].

$\mathbf{x}^i$ = memory patterns. There are $N$ of them, $x^i_s$ is the $s$-th bit of the $i$-th pattern.

$\mathbf{X} = (\mathbf{x}^1, ..., \mathbf{x}^N)$ is the matrix containing all memory patterns.

$\boldsymbol{\xi}$ = state of the Hopfield net, $\xi_s$ = activation state of the $s$-th neuron.

Connection weight between the $s$-th and $t$-th neurons:

$$\boxed{\text{Weights}} \quad T_{s,t} = \sum_i x^i_s x^i_t \tag{3}$$

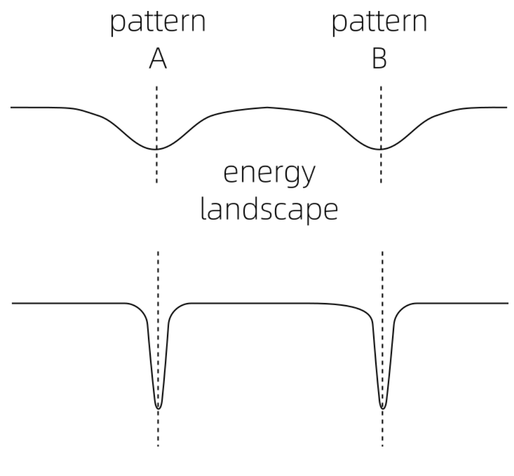
Refering to figure (1), the weights actually record the co-activation of pixel pairs.

Total energy (Hamiltonian):

$$\boxed{\text{Energy}} \quad E = -\frac{1}{2} \sum_s \sum_{t \neq s} \xi_s T_{s,t} \xi_t \tag{4}$$

## Modern Hopfield Network

In the classical Hopfield network, when two patterns A and B are too close together, they would interfere, so the storage capacity is limited. Modern Hopfield networks modify the energy function such that interference is reduced, resulting in greater storage capacity:




(5)

The new energy function:

$$\boxed{\text{New Energy}} \quad E = -\sum_i F(\boldsymbol{\xi}^T \mathbf{x}^i) \tag{6}$$

Note: $\boldsymbol{\xi}^T \mathbf{x}^i = \sum_s \xi_s x^i_s$ , $F$ = interaction function.

[Demircigil et al 2017] proposes to use the exponential function for $F$.

State update rule:

$$\boldsymbol{\xi}^{\text{new}} = \mathbf{X} \,\text{softmax}(\beta \mathbf{X}^T \boldsymbol{\xi}) \tag{7}$$

This corresponds to the update rule of the conventional Transformer:

$$\mathbf{Z} = \text{softmax}\left( \frac{1}{\sqrt{d_k}} \mathbf{Q}\,\mathbf{K}^T \right) \mathbf{V} \tag{8}$$

## Hopfield-Transformer □□

□□ Transformer's state update rule:

$$\mathbf{Z} = \text{softmax}\left( \frac{1}{\sqrt{d_k}} \mathbf{Q}\,\mathbf{K}^T \right) \mathbf{V} \tag{9}$$

Modern Hopfield network's state update rule:

$$\mathbf{Z} = \text{softmax}\left( \beta \hat{\mathbf{R}} \hat{\mathbf{Y}}^T \right) \hat{\mathbf{Y}}^T \tag{10}$$

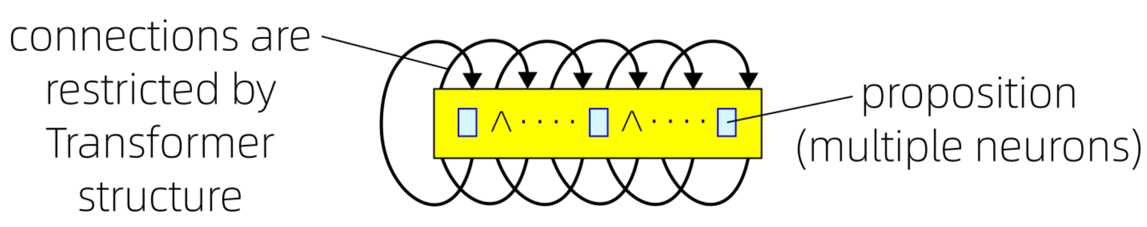
□□ $\hat{\mathbf{R}}, \hat{\mathbf{Y}}, \hat{\mathbf{X}}$ □ $\mathbf{R}, \mathbf{Y}, \mathbf{X}$ □□□□□□□□ $\mathbf{W}$'s □□□□□□□□□□□□□□□□□□□

Query patterns = $\mathbf{R} = (\mathbf{r}_1, ..., \mathbf{r}_M)$ □□□□□□□ □□□□ $M$ □□□□□□□□□□ Transformer □ $M$ □□□ □□□□□□□□□□□□ Hopfield □□ □□□□□□□□□ Hopfield □□ Transformer □□□□□

$\mathbf{Y}$ □□ Hopfield □□□□ patterns□□□□□ Transformer □ keys □□□□

□ Self-Attention □□$\mathbf{R} = \mathbf{Y}$.

□□ (2) □□□□□□ □ Transformer □□ □□□ Hopfield □□□




(11)

# Boltzmann □

□□□ □□□ Boltzmann machine □ □□ Hopfield network □□□□

Let $O = (O_1, ..., O_n)$ be the **state vector**.

$W = \{W_{s,t}\}$ are connection **weights**.

**State update rule**: $i$-th unit is set to 1 with probability

$$\frac{1}{1 + e^{-S_i/T}} \tag{12}$$

where $T$ is a temperature.

□□□□□□ update rule□□ Hopfield □□ □□ □□□□□

$$P(O) = P(O|W) = \frac{e^{-\mathcal{E}(O)/T}}{Z} \quad \boxed{\text{Boltzmann distribution}} \tag{13}$$

where partition function $Z = \sum_U e^{-\mathcal{E}(U)/T}$

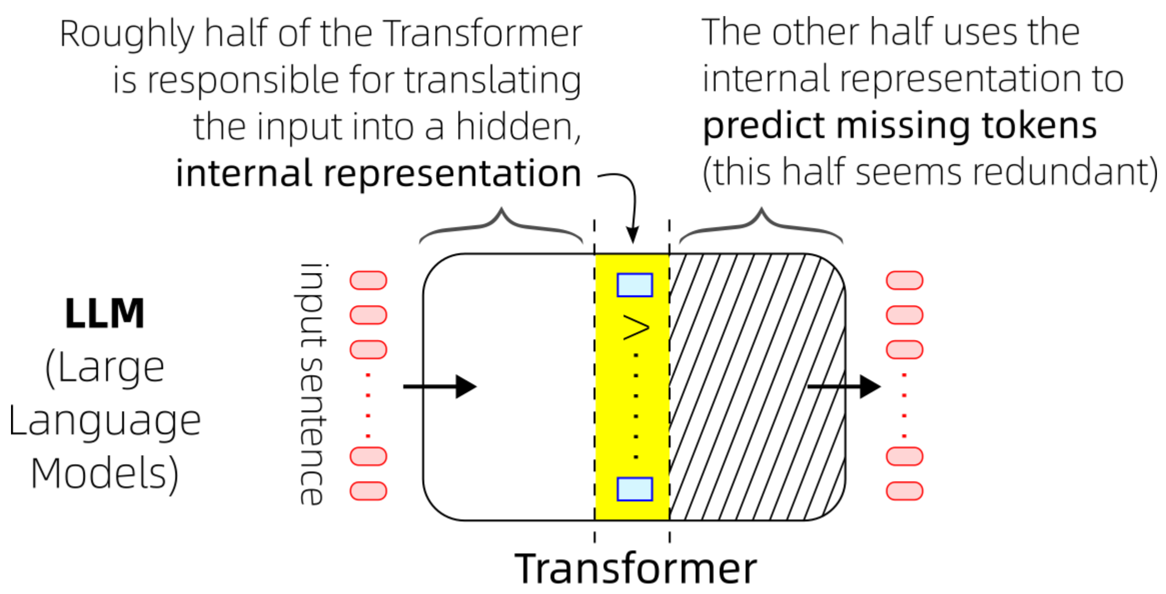
TO-DO: □□ □□□□ Hopfield network □ Boltzmann state update rule.

TO-DO: □□ Hopfield-Boltzmann machine □ **learning** update rule. □□□□□□□ □□ □ update □□

# □□□□□□ □□

## LLM □□□□□□□□□□

LLM (□□□□□□) □□□ □□□ (auto-regression, □□□□□□□□□□□□) □□□□□□ □□□□□□□Transformer □□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□ □□□□□ □□□□□ □□□□□□□□ □□□□□□□□□□□

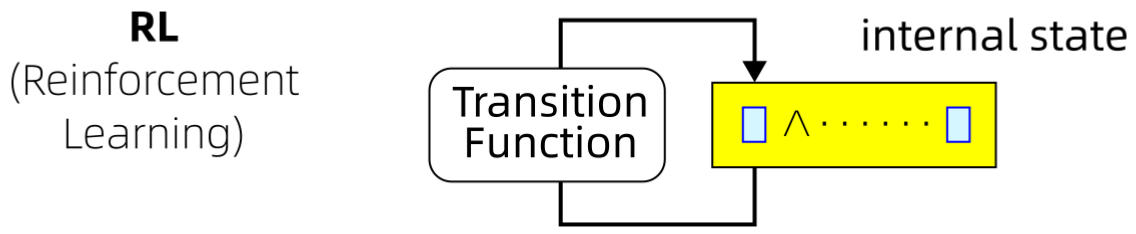



(14)

□□□□□□□□□□□□□□□□□□□□□ □□□□□ hidden representation□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□ □□□□□□□□□□□□□□□ AGI □□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□ □□□□ □□ □□□□□□□□□□□□□□□□□□□□□□□□□□ AGI □□□□□□□




(15)

□□ transition function □□□□□□ Transformer □□□□

## Transformer → Hopfield → Boltzmann

□□□□□□□□□□□ □□ □ □□ □□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□ □□□□□ □□□□□□□□□□□□ □□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□

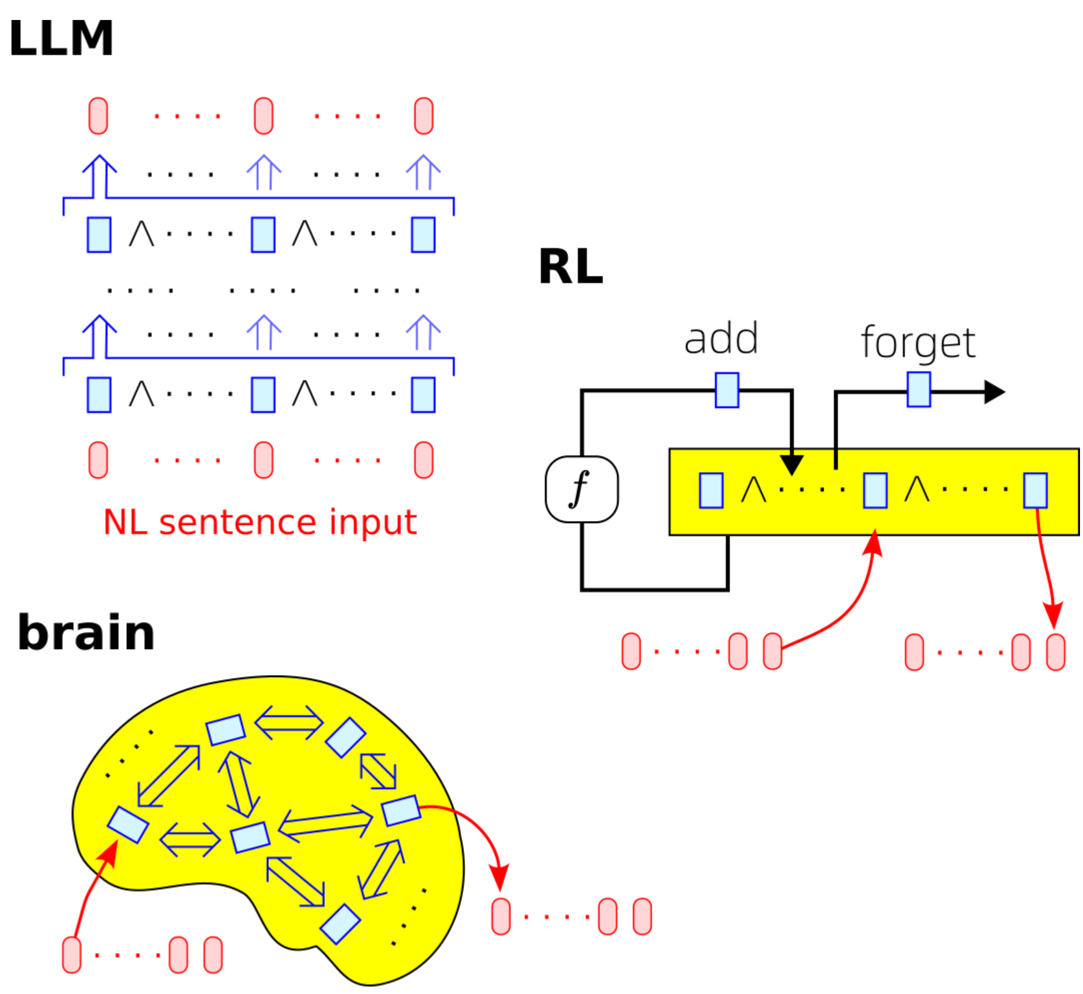
□□□□□□ Transformer □□□□ □□□□□ □□□□□□Transformer □□□ token□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□ □□□ trick □□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□ *"Hopfield Network is All You Need"* □□□□ Hochreiter et al □□□Transformer □ Hopfield □□□□□□□□

Hopfield □□□□□□□□□□□□ □□□□□□ Boltzmann machine. □□□□□□□□□□□□□□□□□□□□□ Hinton□Bengio□LeCun □□ □□□□□□□□□□□□□□□□□□□ □□□□□ □□□□ EBM (energy-based models).

Transformer □□□□□□□□ Hopfield □□□□□□□ EBM. □□□□□□□□□□ Bellman □□□□□□□□□□□ learning update rule, □□□□□□ AGI □□□□□□□

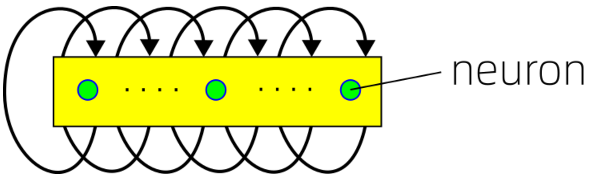
□□□□□□□□□□□□□□ □ RL □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□ "areas" □□□□ □□□□ □□□□ □□□□□□□□□□□□□□□□□□□□ □□ □□□□□□□□□




(16)

# □□ Transformer + RL □ Road Map

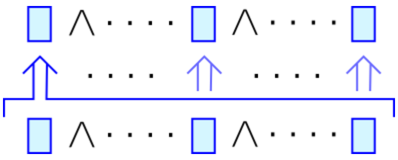
□□□□□□□□□Hopfield □□ □ fully-connected, □□□□□□□□




(17)

### □1□

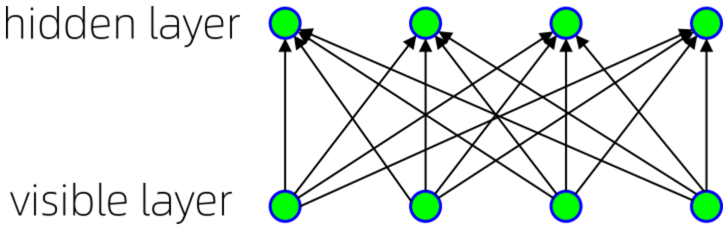
□□□Hopfield Network Is All You Need□□□(17) □□ □□□ Transformer □ forward inference□□□ Hopfield □□□□□□□□□□□□□□□□□□□□□□□



(18)

### □2□

□□□□ □ Hopfield → Boltzmann□□□□□□ □□□□□ □ Boltzmann machine □□□□□ intractable □□□□ restricted Boltzmann machine, □□□□□ bipartite graph□□□ □□ □□□□□□□□□ factorize□

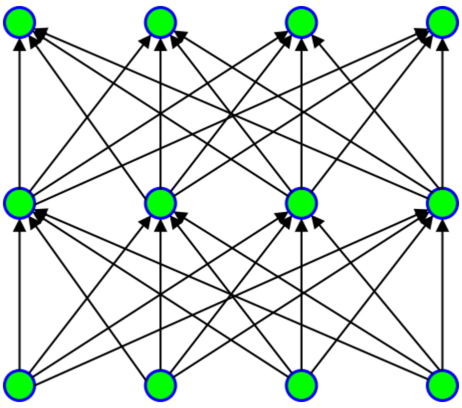



(19)

□□□□□□(19) □ □□□□□ □□ Transformer□

### □3□

□□□□□□□□ Boltzmann machine □□□□□□□□ □□□□□ Transformer□□□ □□□□□



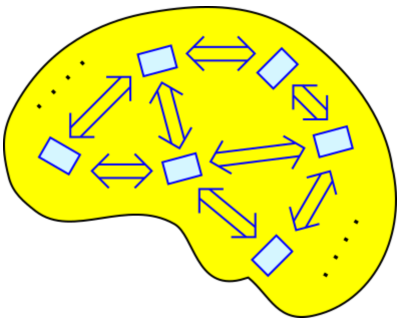
(20)

□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□ AGI □□□

### □4□

□□□□□□□□□□□□□□□□□□□□□□□

□(20) □□□□□□□□ “areas” □□□

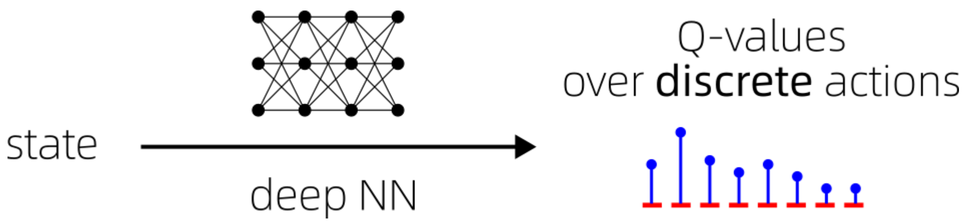


(21)

□□□□ □(20) □□□□□ areas □□□□□□ □□□□ □□□□□□□□□□□□□□□□□□□□□ AGI □□□□□ □□□□□ areas□areas □□□ Transformer □□□ □□ Transformer □□□□□ □□□□□□□□□□□□□□

# RL update rule

□□□Deep Q Learning (DQN) □□□□□□□□□ □□□ action space□□□□□□□□□□□□




(22)

□□ Q-value □□ $Q(s,a)$ = □□□ $s$ □□□ $a$ □□□□ □□□□□ □□□□ $s$ □□□□□□□□□□□□□□□□ $a$ □□□□□□□□□□□□□□ $Q(s,a)$ □□□□□□□□□□□□□□□□□□□□□□

(s,a) □□□□□ □□□□□□□□ Energy-Based Model (EBM) □□□□□□□□□□□□ Hinton & Sallans 2004 [1], □□□□□□□□□□□□□□□□□ RL □□[1]□ □□□□□□ □ free energy[2] $F$ □□ Q-value: $Q(s,a) \approx -F(s,a)$.

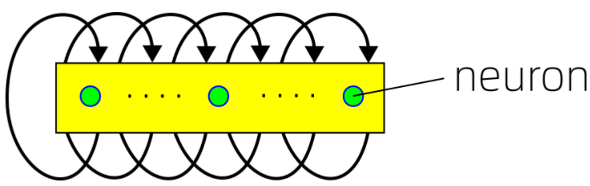
□□□□ □ Bellman update □□ □□ $s$ □□□ $R(s,a)$ □□□□□□□, □□□□□□□□ Q-Learning □ temporal difference update:

$$Q(s,a) \mathrel{+}= \eta \left[ R(s,a) + \gamma \max_{a'} Q'(s',a') - Q(s,a) \right] \tag{23}$$

( $\eta$ = learning rate, $\gamma$ = discount factor, $s$ = state, $a$ = action )
$Q$ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ $Q$ □□□□□□□□ □□ Hopfield □ Boltzmann □□ □□□ □□□□ $s \to s'$ □□□ □□□ $F$ □□□□□□□

$Q$ □ update rule □□□□□ Hopfield □ Boltzmann □□□□□□ update rule. □□□□□□□□□ Hinton & Sallans □□□ □□□ naïve □□□ □□□□□□□□□□□□□□□ back-prop□□□□□□□□□□□

□□□□□□□Boltzmann machine □□□□□□□□□□




(24)

□□□ □(22) □□□□□□□□□□□ □□□□ □ loop □□□□□□□□□□□□□□ back-prop □□□□□□ □□□□□□□□□□□□□□□□□ □□□□□□.....

**□□□□□□**

1. Hopfield / Boltzmann spin □□ set of next propositions.
2. □□□□□ propositions □ $P(x_i|\mathbf{X})$, □□□□□□
3.

# References

[1] Brian Sallans and Geoffrey E. Hinton. "Reinforcement Learning with Factored States and Actions". In: J. Mach. Learn. Res. 5 (Dec. 2004), pp. 1063–1088. ISSN: 1532-4435.

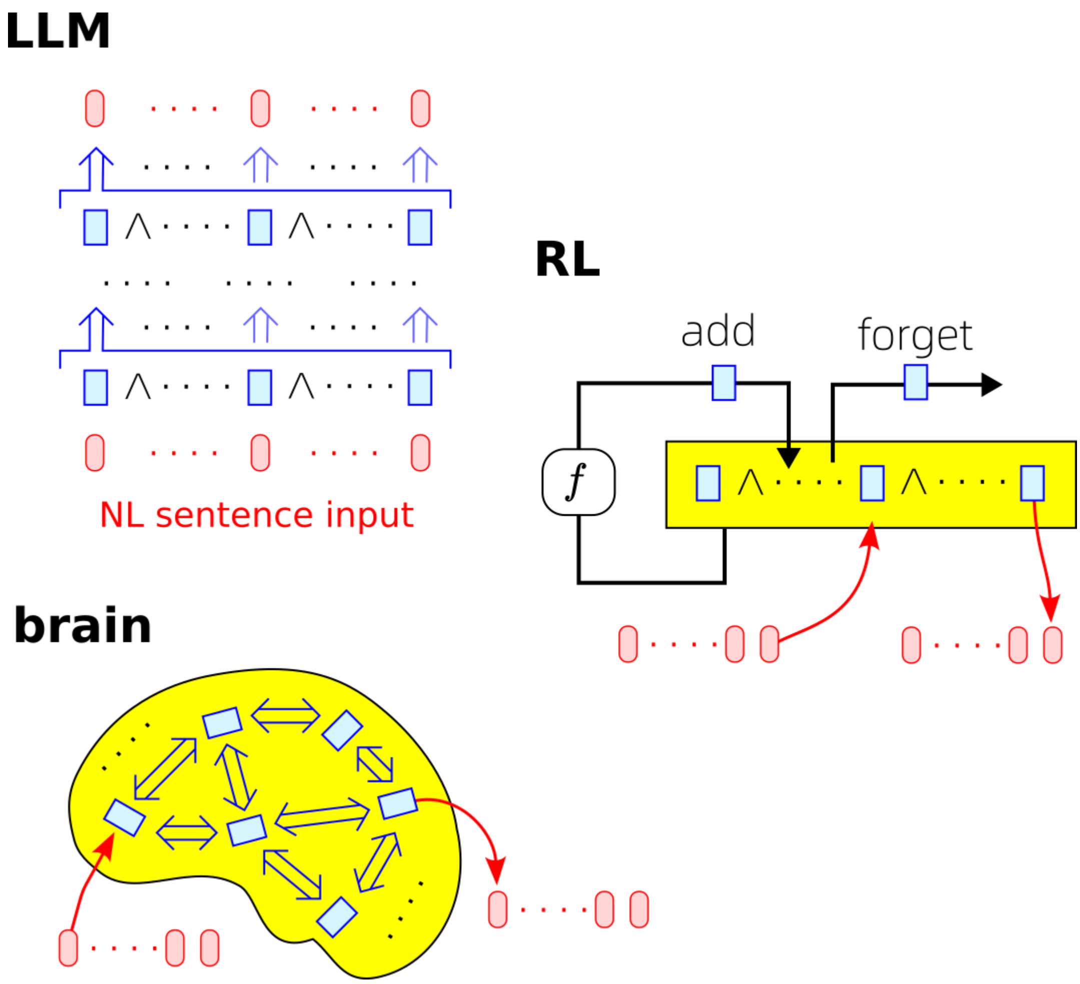
[1] □□□□□□Quantum Enhancements for Deep Reinforcement Learning in Large Spaces□ [2021]□□□□□□□□□□ □□□□□□□□□□ deep RL □□□□□□

[2] □□ free energy □□□□ Hopfield □ Boltzmann □□□□□ visible ($v_i$) □ hidden ($h_i$) □□□□□□ Energy $E$ □□□□□□ $v_i$ □ $h_i$. Free energy □ $E$ □ $h_i$ □□□□□ marginalization: $F(v) = \sum_h E(v,h)$.

# Personal Notes

□□ Ngiam *et al* 2011□□ Andrew Ng □□□□□□□□□ Deep Energy Models (DEMs).

□□□□□□□□□ LLM, RL □ □□□ □□□□□□ □□□□□□□□□




(25)

□ model-based RL □□□□□□□□ PILCO□□□□ □□□□□□ □□□□□□□□□□□□□□ □□□□□□□□□□□ □□□□□□□□□□ □□□□□□□□□□□□ □□ model-based □ model-free □□□□□□□ □ □□□□□ □□ □□□□□□□□□ □□□ □□□□□□□ □ sensory-based inference □□□

□□□□□□ □□□□□ update□□□□□□□ □□□□□□

RL □□□weights □□ next state□utility □□ next state□weights □□ utility = energy = probability distribution over next states. □ current state □ utility value □□□□□□□□

State □ action □□□□□□□□□ □□□□ □ □□□□□ □□□□□

□□□□□□□□□□□ □□□□□□□□

□□□□□□□□□□□□....

## □□