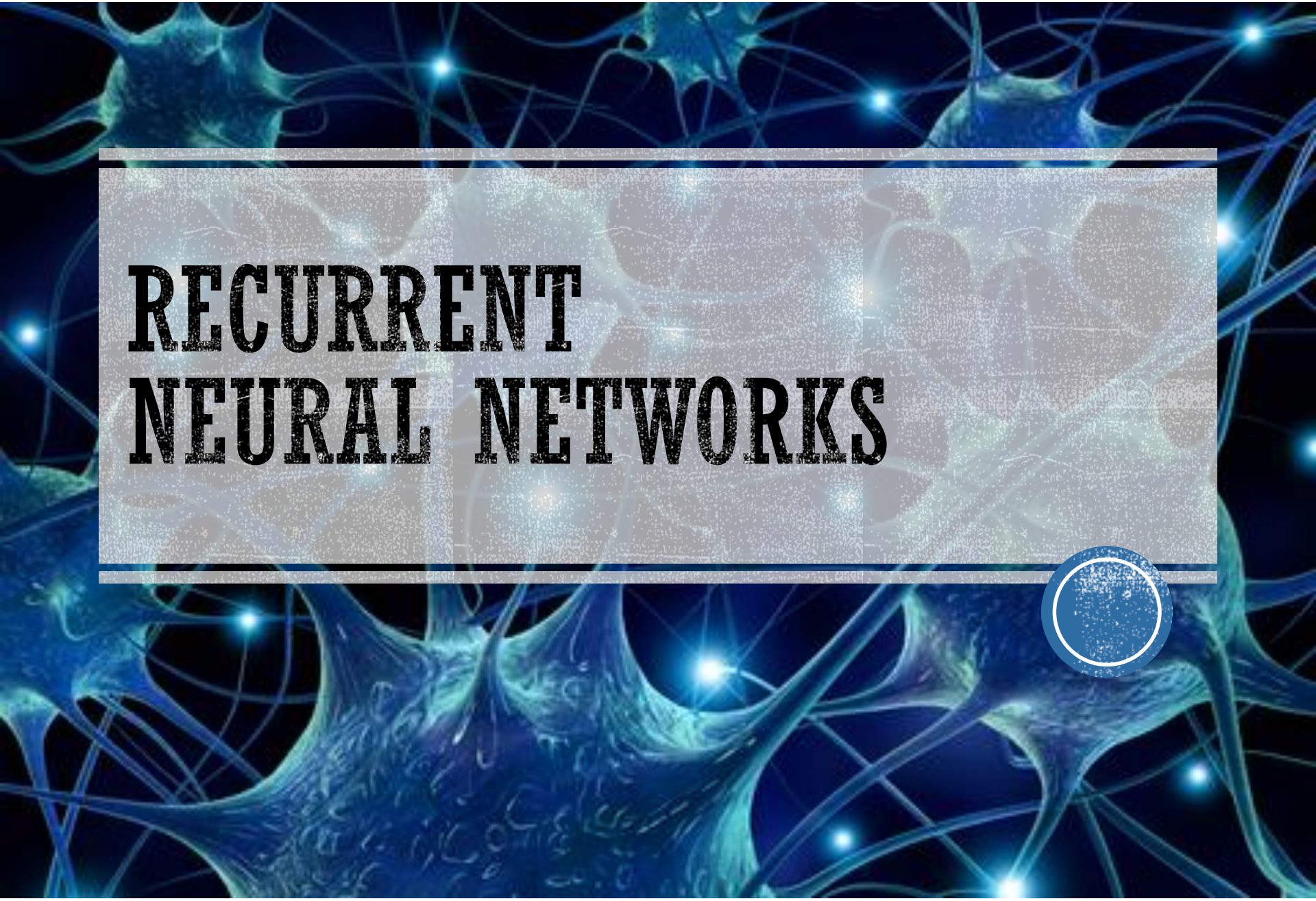
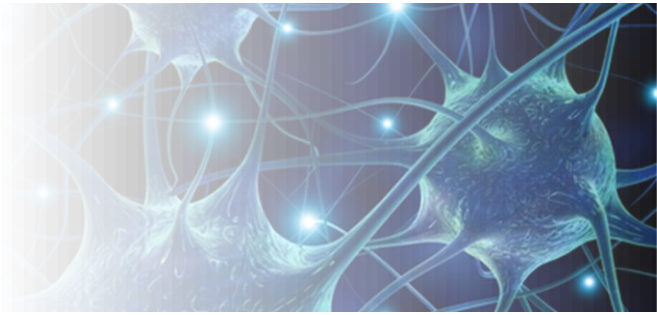

RECURRENT
NEURAL NETWORKS

WHY?

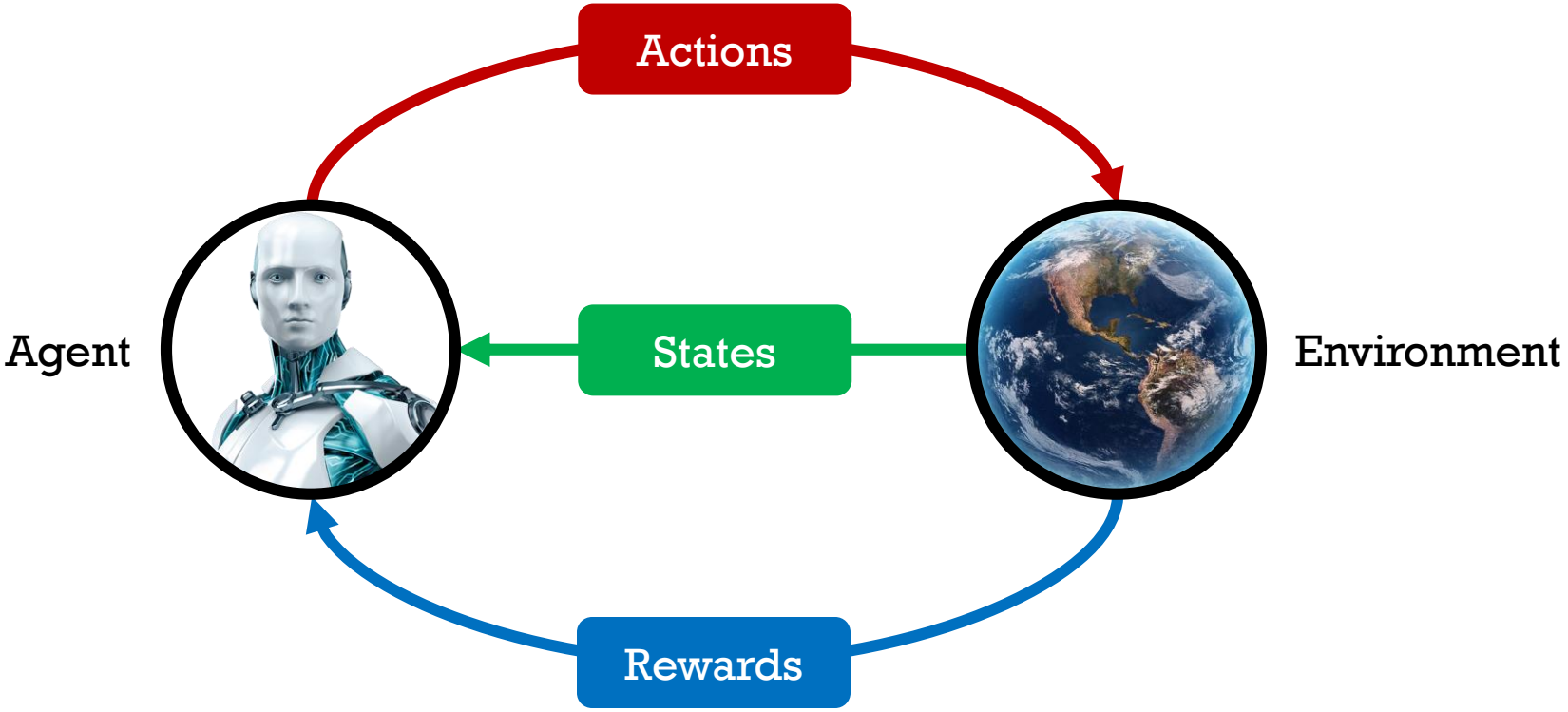


SPEECH TRANSLATION

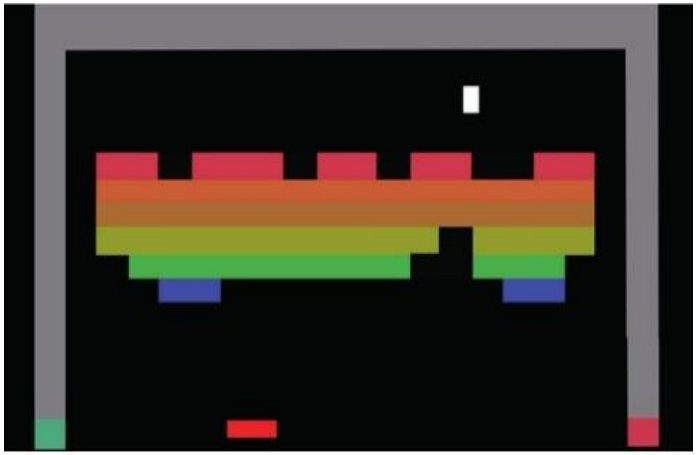
From Hidden Markov Models to Recurrent Neural Networks

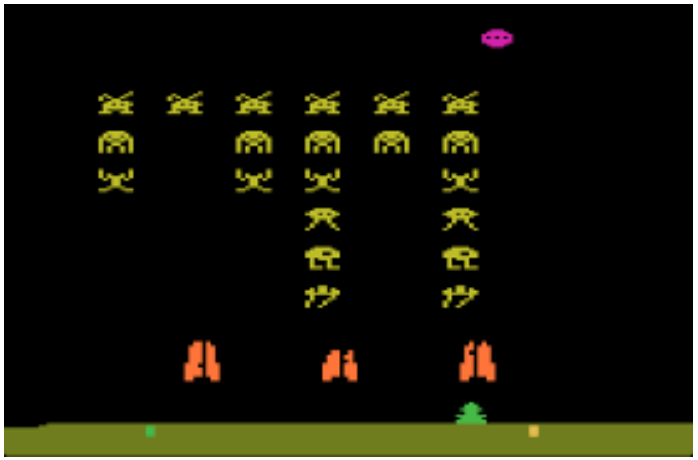
https://www.youtube.com/watch?v=mWTySUGXR2k

REINFORCEMENT LEARNING



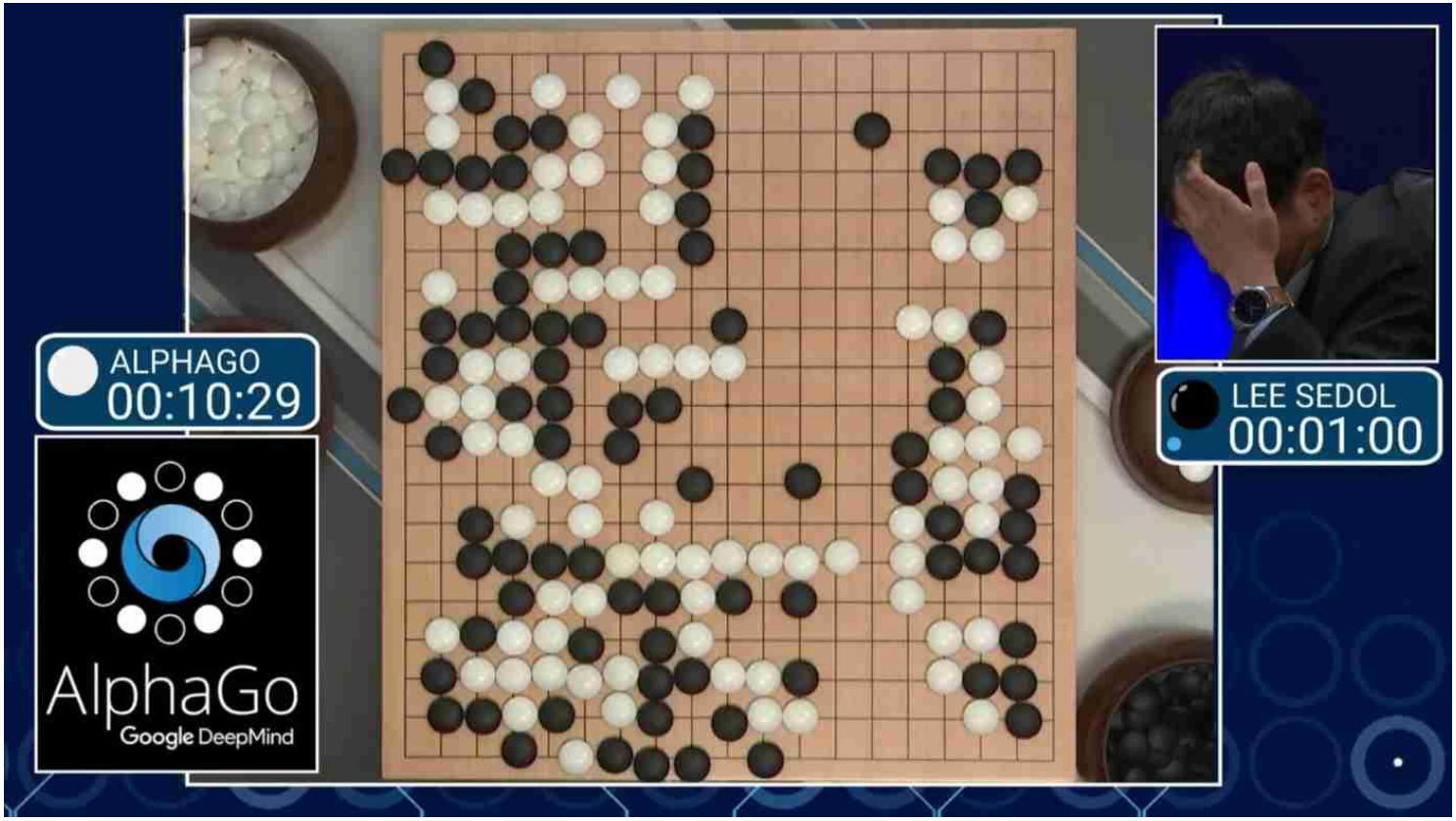
ATARI GAMES





https://www.youtube.com/watch?v=rbsqaJwpu6A

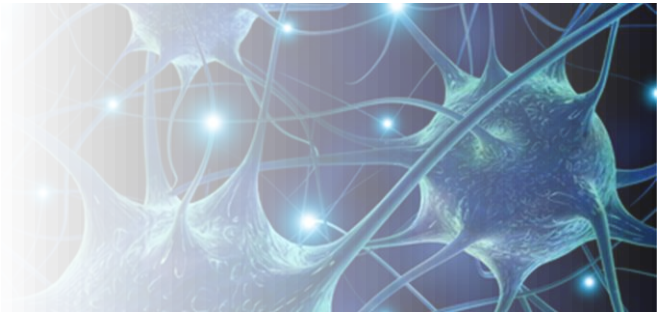
ALPHAGO



https://www.youtube.com/watch?v=SUbqykXVx0A

SELF-DRIVING CARS

WHAT?

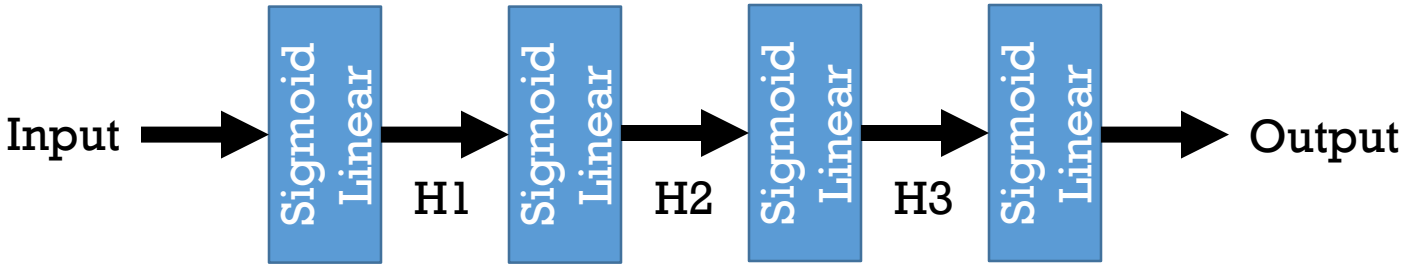


NOTATION

Function (Compute)

Data (Transport)

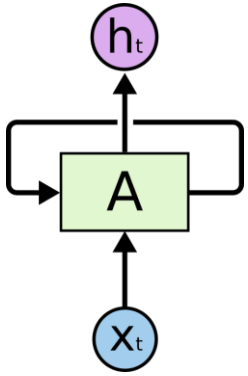
Example. "Four-Layer"(Weighted) Neural Network.



RECURRENT NEURAL NETWORKS

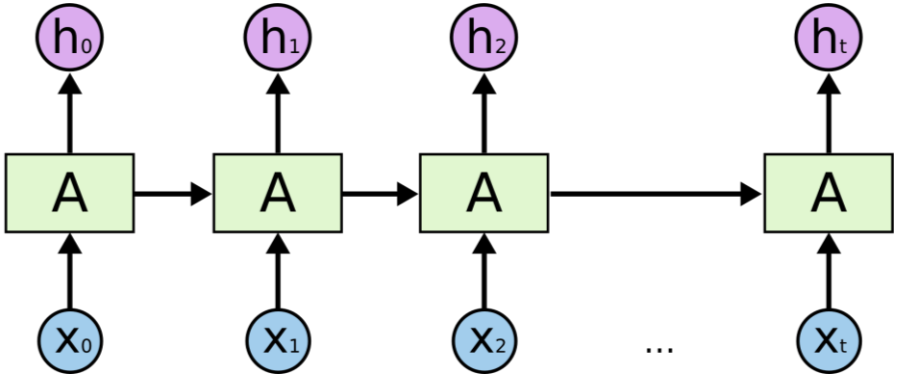
Definition.

Neural networks where the connections form a directed cycle. As opposed to feedforward neural networks.



Redefinition.

May be unrolled as a feedforward network with shared weights propagating through time.

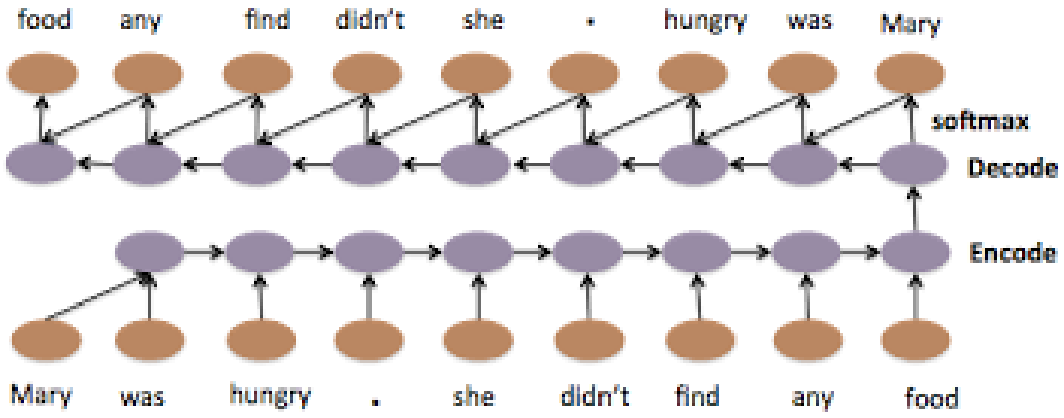


http://colah.github.io/posts/2015-08-Understanding-LSTMs/

RECURRENT NEURAL NETWORKS

Temporal State.

- RNN dynamics encode some time-varying state.
- Perfect for modeling sequences, e.g. speech recognition, natural language processing, financial modeling.



PROBLEM WITH VERY DEEP NETWORKS

Vanishing or Exploding Gradient Problem.

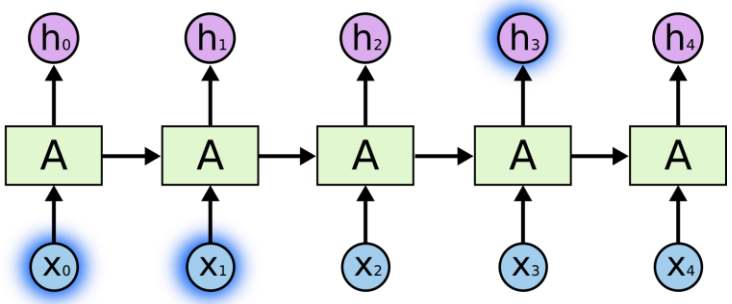
- Difficult to train very deep networks because of issues with backpropagation algorithm
- Error signals $\delta^{(l)}$ either shrink exponentially, or grow exponentially with the number of layers

Recurrent Neural Networks are Very Deep!

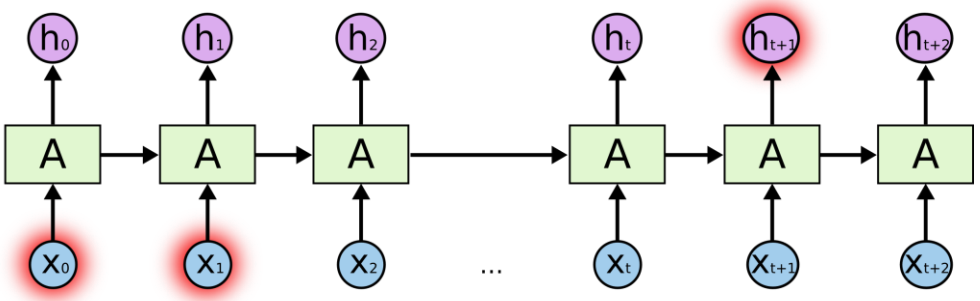
- We want RNNs to learn long-term dependencies
- But for a long time, nobody knew how to train them

LONG-TERM DEPENDENCIES

"the clouds are in the *sky*"

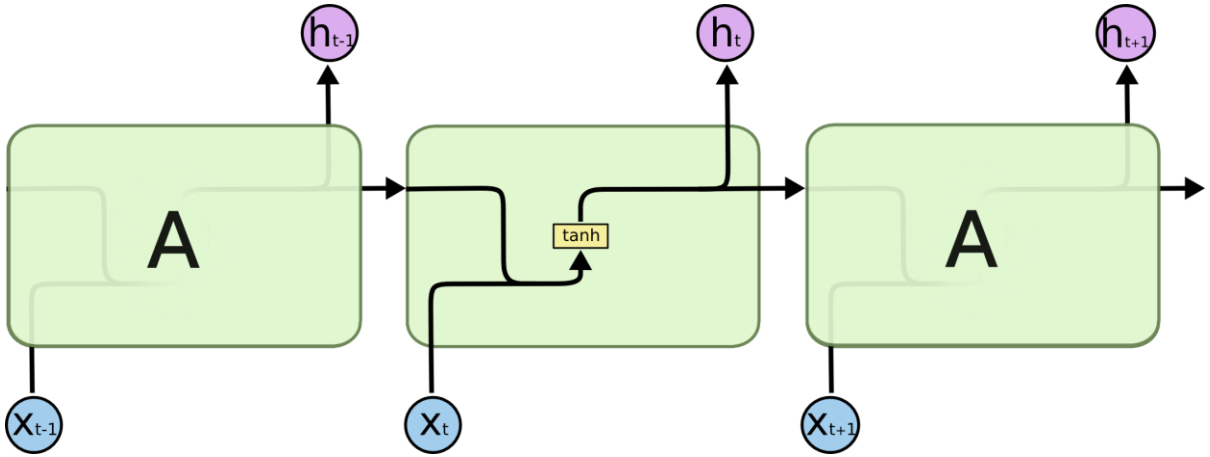


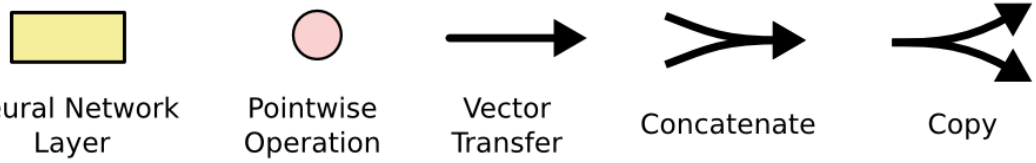
I grew up in France… I speak fluent *French*.



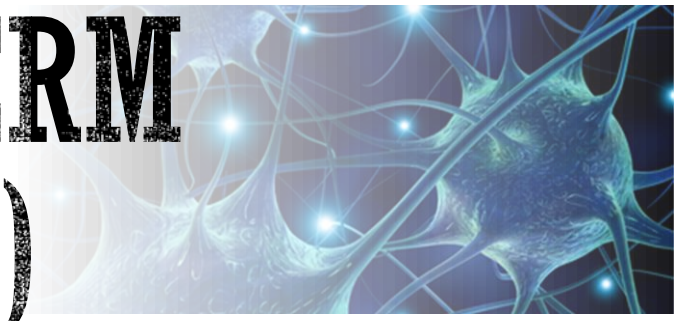
STANDARD RNN

Single Layer RNN.



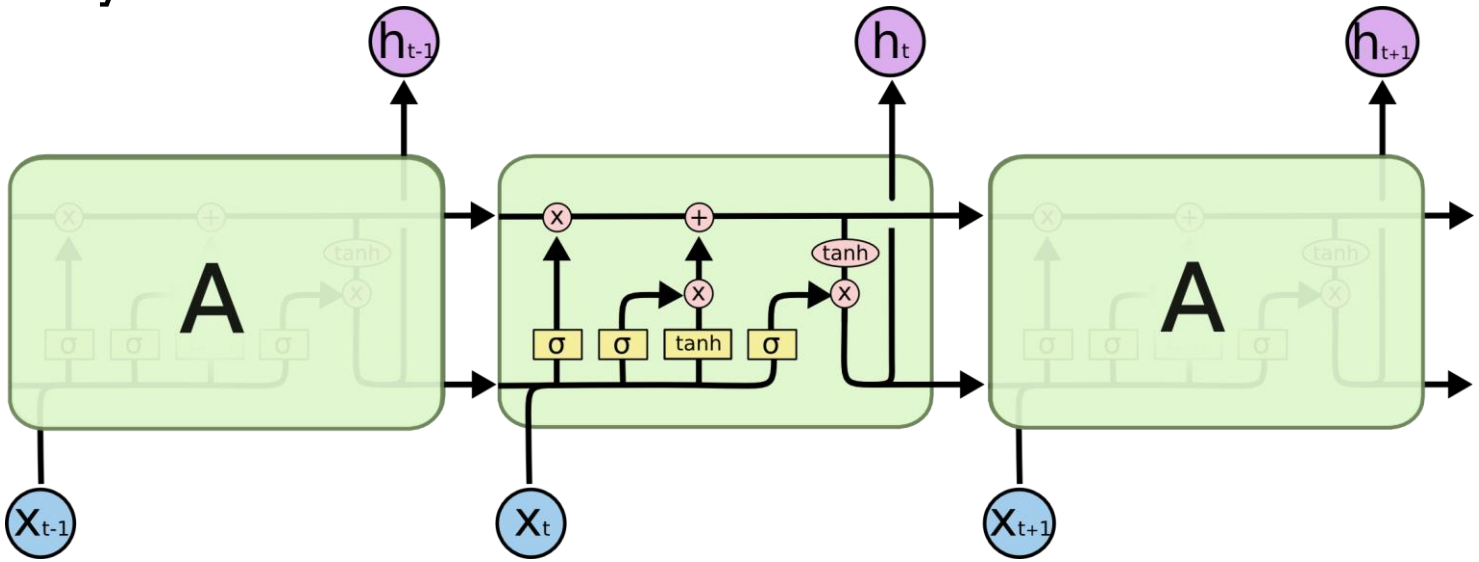


LONG SHORT TERM MEMORY (LSTM)



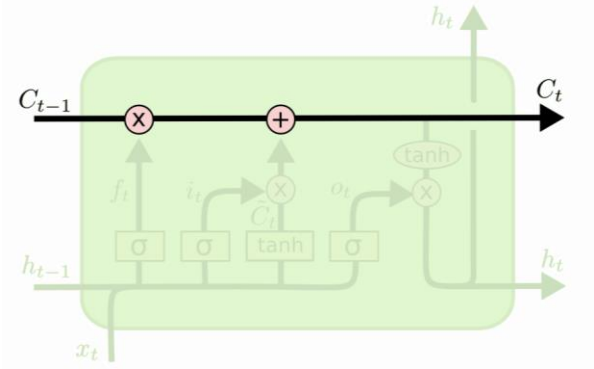
LONG SHORT TERM MEMORY

Four-Layer RNN.

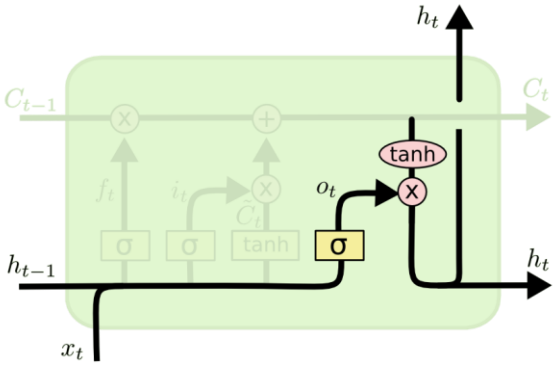


STATES AND GATES

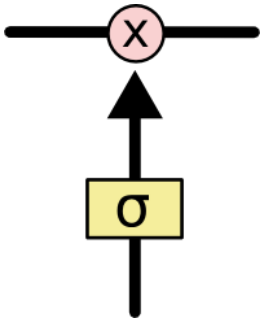
Cell State



Output State



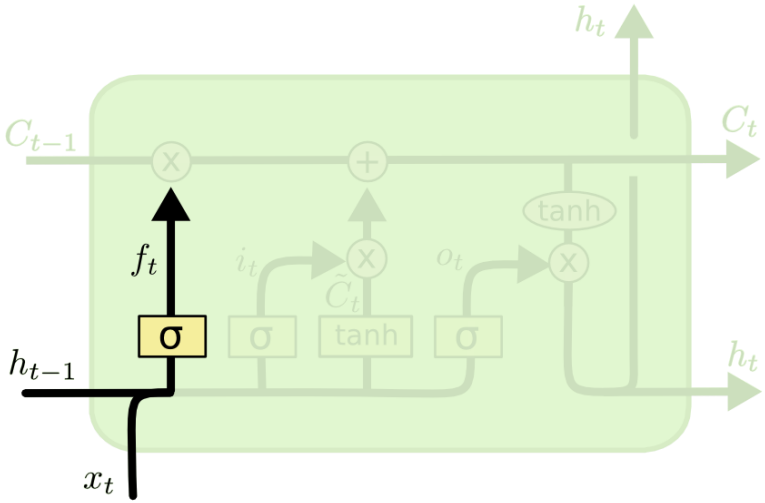
Gates



Output of sigmoid
is between 0 and 1
0 – forget past state
1 – keep past state

LAYER 1 – FORGET PAST STATE

Example. Forget gender of subject when subject changes.

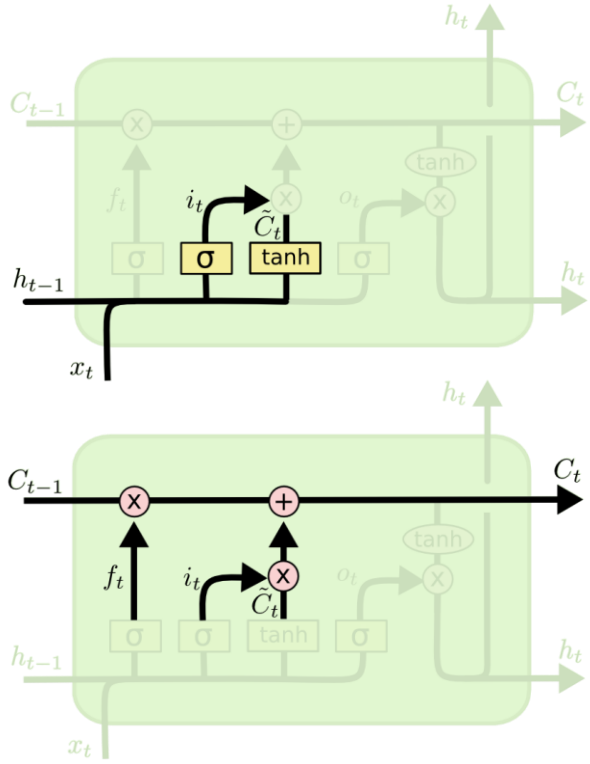


How much to forget?

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

LAYER 2,3 – ADD NEW INFORMATION

Example. Add gender of new subject.



How much to add?

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

What info to add?

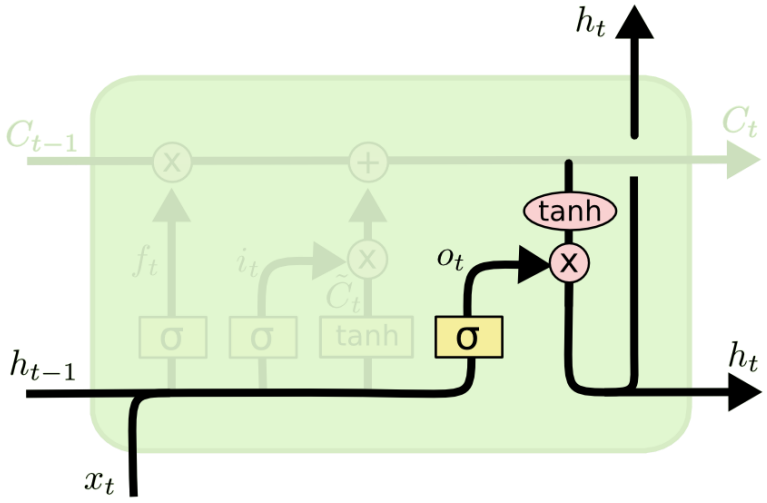
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

New cell state.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

LAYER 4 – SELECTIVE OUTPUT

Example. Singular or plural form of verb? Output and forward.



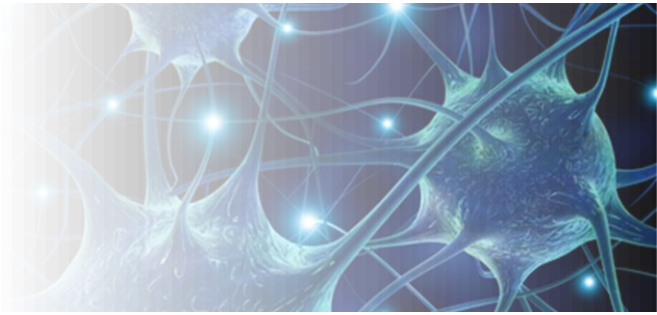
How much to output?

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

What to output?

$$h_t = o_t * \tanh\left(C_t\right)$$

ASSOCIATIVE MEMORIES



ENERGY-BASED LEARNING

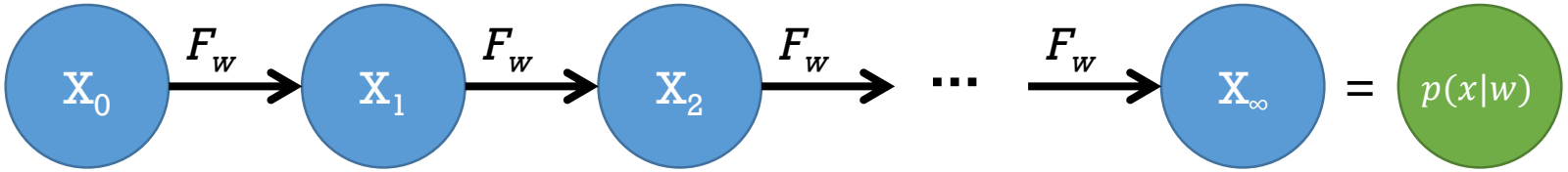
- Discrete model – states x, parameters w
- Probabilities defined in terms of *energy* $f(x|w)$

$$p(x|\omega) = \frac{e^{-f(x|\omega)}}{Z(\omega)} \quad \text{where } Z(\omega) = \sum_x e^{-f(x|\omega)}$$

- Energy wells represent likely states of the model

ENERGY-BASED LEARNING

- Partition function $Z(w)$ often difficult to compute, but it is easy to sample from model distributions using Markov Chain Monte Carlo (MCMC) techniques
- Let $X_0 = (x_1, x_2, \ldots, x_N)$ denote the data
- Given parameter w, let F_w be a corresponding MCMC update that acts on the state space, i.e. $X_{i+1} = F_w(X_i)$.



- Loosely, the X_i converges to the model distribution $p(x|w)$

ISING MODEL (1924)

Magnetic spins on a lattice.

- Energy of a configuration σ

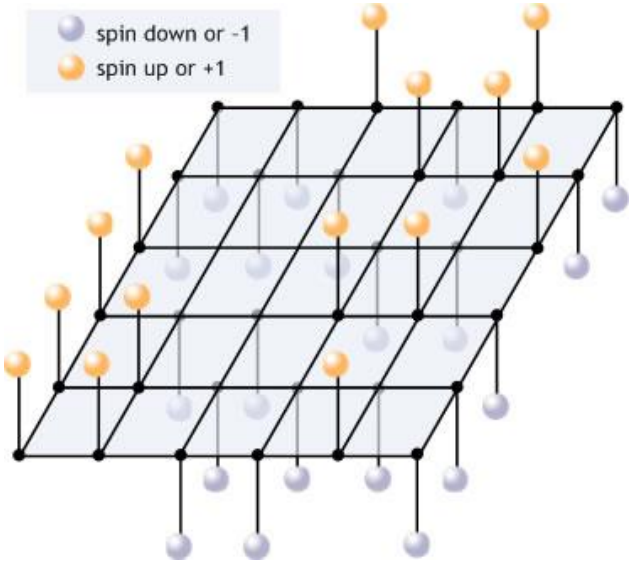
$$H(\sigma) = -\sum_{\langle i\ j\rangle} J_{ij}\sigma_i\sigma_j - \mu\sum_j h_j\sigma_j$$

- Probability of observing σ

$$P_\beta(\sigma) = \frac{e^{-\beta H(\sigma)}}{Z_\beta}$$

- Normalization constant

$$Z_\beta = \sum_\sigma e^{-\beta H(\sigma)}$$



HOPFIELD NETWORKS (1982)

Using Ising Model to mimic neural networks.

- Energy

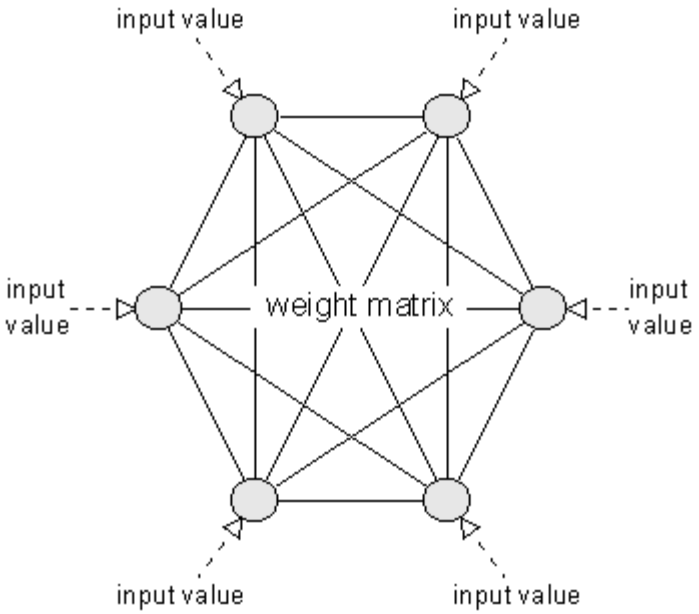
$$E = -\frac{1}{2}\sum_{i,j} w_{ij} s_i s_j + \sum_i \theta_i s_i$$

- Update Rule

$$s_i \leftarrow \begin{cases} +1 & \text{if } \sum_j w_{ij} s_j \geq \theta_i, \\ -1 & \text{otherwise.} \end{cases}$$

- Learning Rule (Hebbian)

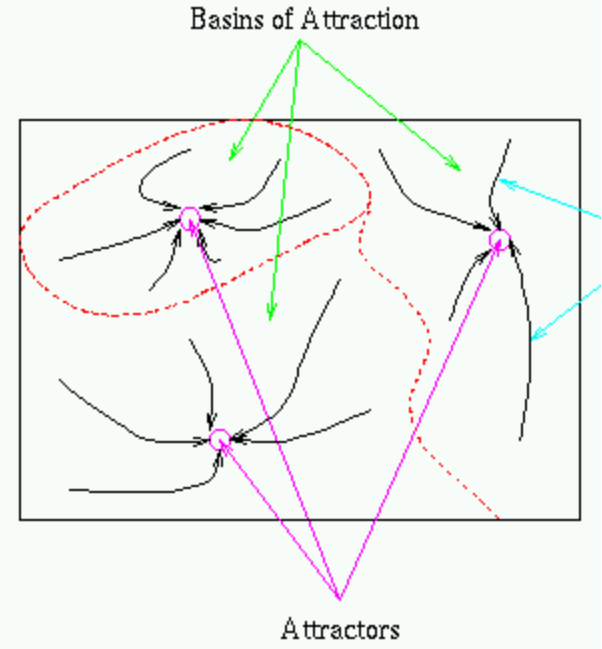
$$w_{ij} = \frac{1}{n}\sum_{\mu=1}^{n} \epsilon_i^\mu \epsilon_j^\mu$$

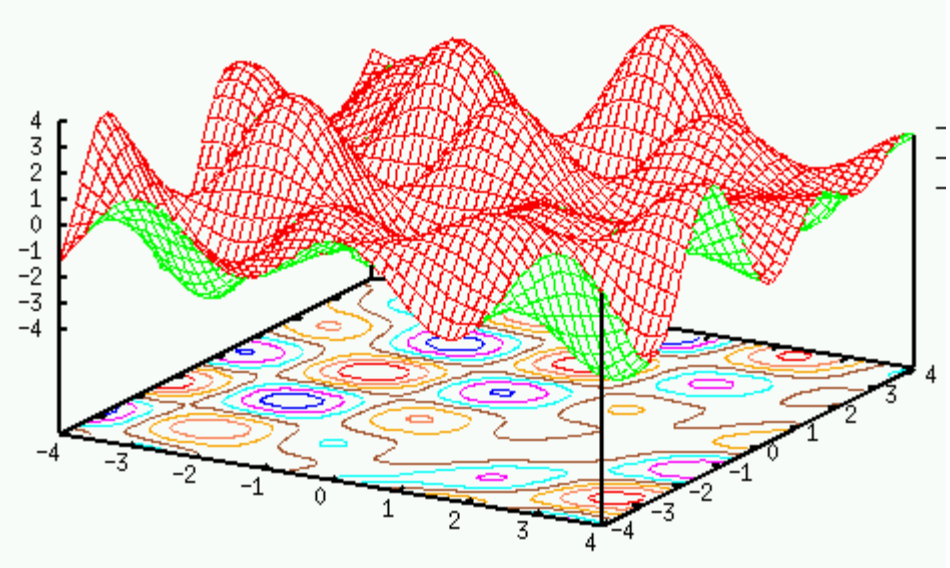


CONVERGENCE

Theorem. The update rule always converges.

Proof. The energy decreases with each update.





BOLTZMANN MACHINES (1985)

Stochastic version of Hopfield networks.

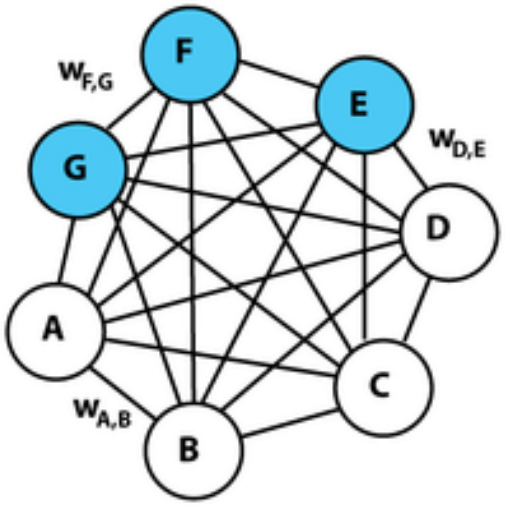
- Energy

$$E = -\frac{1}{2}\sum_{i,j} w_{ij} s_i s_j + \sum_i \theta_i s_i$$

- Gibbs sampling

$$\mathbb{P}(x_i = 1 | x_{-i}) = \text{sigmoid}\left(\sum_{j \neq i} w_{ij} x_j + b_i\right)$$

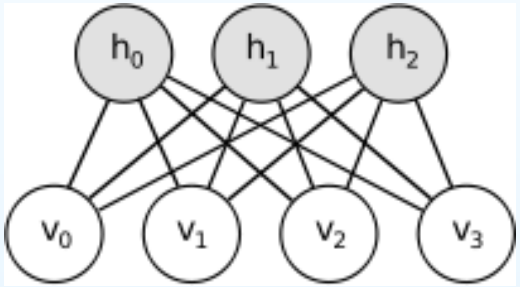
- Learning (typically not easy or too slow)



RESTRICTED BOLTZMANN MACHINES

- **Binary-valued neurons (Geoffrey Hinton)**

Undirected discrete graphical model with binary states.
Graph G is bipartite with two layers: observed and hidden nodes.
Parameters: weights ω_{ij} for each edge (h_i, v_j),
bias b_i for each h_i, bias c_j for each v_j.



$$p(x|W,b) \propto e^{-f(x|W,b)}$$
$$f(x|W,b) = \sum_{i<j} W_{ij} x_i x_j + \sum_i b_i x_i$$

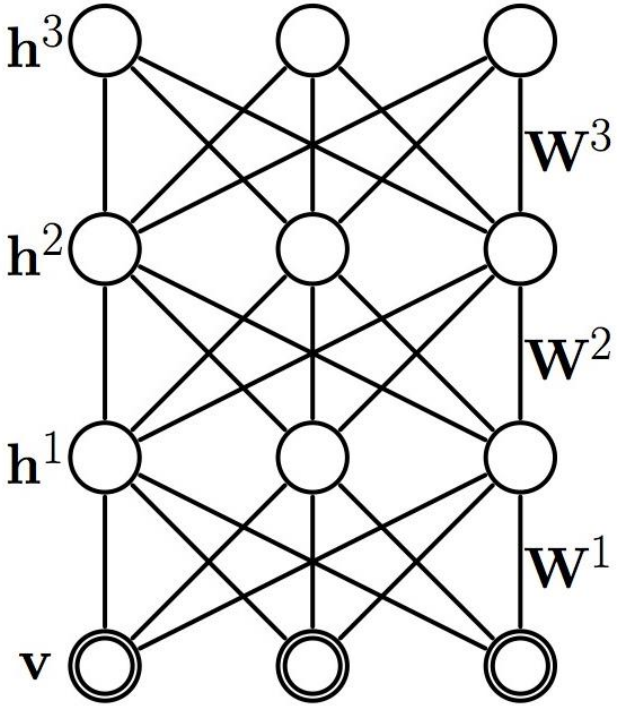
$$\mathbb{P}(h_i = 1|v) = \mathrm{sig}(b_i + \sum_j \omega_{ij} v_j), \quad \mathrm{sig}(x) = \frac{1}{1+e^{-x}}$$

CONTRASTIVE DIVERGENCE

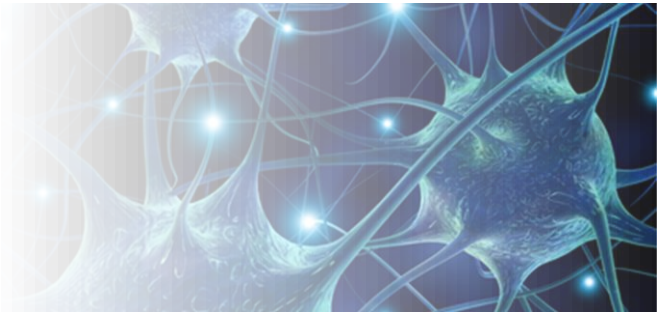
- MLE minimizes $$\ell(\omega) = \log Z(\omega) + \frac{1}{N}\sum_i f(x_i|\omega)$$
- Gradient $$\begin{aligned}\frac{\partial \ell}{\partial \omega} &= \mathbb{E}_{X_0}\left(\frac{\partial f(x|\omega)}{\partial \omega}\right) + \frac{\partial \log Z(\omega)}{\partial \omega} \\ &= \mathbb{E}_{X_0}\left(\frac{\partial f(x|\omega)}{\partial \omega}\right) - \mathbb{E}_{X_\infty}\left(\frac{\partial f(x|\omega)}{\partial \omega}\right)\end{aligned}$$
- CD relaxes it to $$\frac{\partial \ell}{\partial \omega} = \mathbb{E}_{X_0}\left(\frac{\partial f(x|\omega)}{\partial \omega}\right) - \mathbb{E}_{X_1}\left(\frac{\partial f(x|\omega)}{\partial \omega}\right)$$
- This relaxation seems to work well in experiments.

Hinton, Geoffrey E. "Training products of experts by minimizing contrastive divergence." Neural computation 14.8 (2002): 1771-1800.

DEEP BOLTZMANN MACHINES

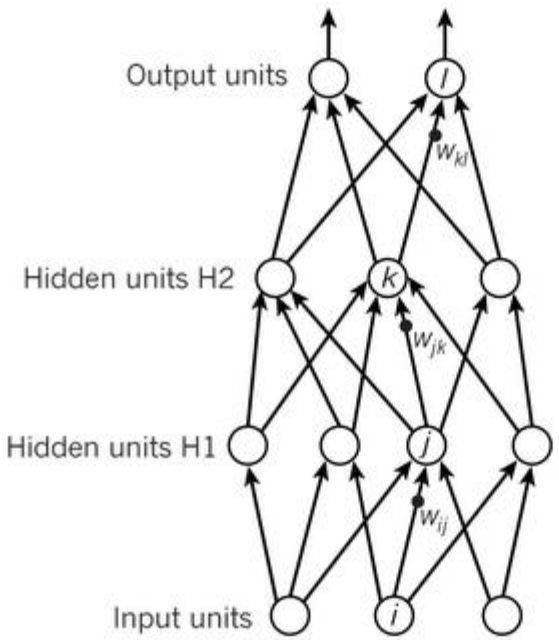


BIOLOGICAL PLAUSIBILITY



FEED-FORWARD NETWORKS

Forward Propagation



$$y_l = f(z_l)$$
$$z_l = \sum_{k \,\varepsilon\, \text{H2}} w_{kl}\, y_k$$

$$y_k = f(z_k)$$
$$z_k = \sum_{j \,\varepsilon\, \text{H1}} w_{jk}\, y_j$$

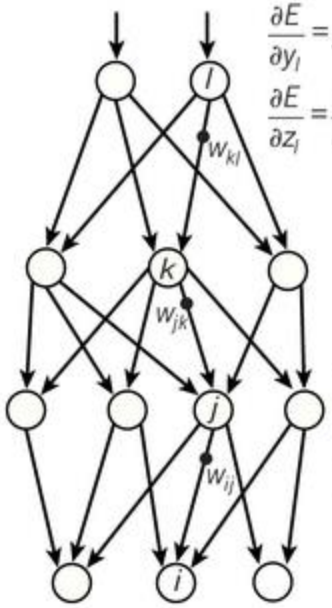
$$y_j = f(z_j)$$
$$z_j = \sum_{i \,\varepsilon\, \text{Input}} w_{ij}\, x_i$$

Back Propagation

$$\frac{\partial E}{\partial y_l} = y_l - t_l$$
$$\frac{\partial E}{\partial z_l} = \frac{\partial E}{\partial y_l}\frac{\partial y_l}{\partial z_l}$$

$$\frac{\partial E}{\partial y_k} = \sum_{l \,\varepsilon\, \text{out}} w_{kl}\frac{\partial E}{\partial z_l}$$

$$\frac{\partial E}{\partial z_k} = \frac{\partial E}{\partial y_k}\frac{\partial y_k}{\partial z_k}$$



$$\frac{\partial E}{\partial y_j} = \sum_{k \,\varepsilon\, \text{H2}} w_{jk}\frac{\partial E}{\partial z_k}$$

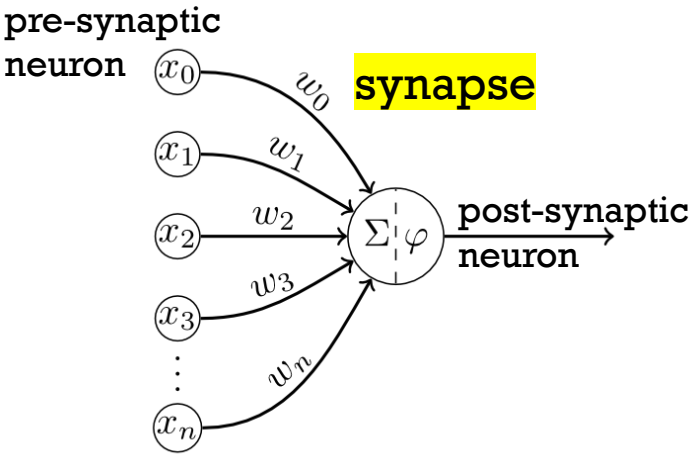
$$\frac{\partial E}{\partial z_j} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial z_j}$$

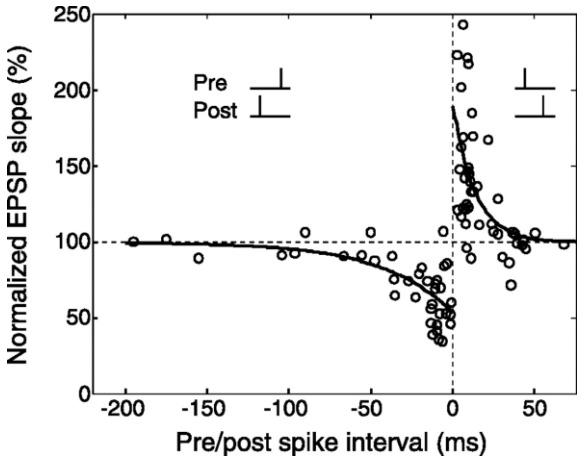
ISSUES WITH BACKPROPAGATION

Backpropagation	Biological Neuron
Only linear operations used	Both linear and non-linear operations used
Feedback paths need to know derivatives of feedforward nonlinearities	No such mechanism observed
Feedback paths need to know weights of feedforward paths	No such mechanism observed
Alternating feedback and feedforward phases need to be precisely clocked	No such mechanism observed
Neurons communicate real values	Neurons communicate binary spikes
Target outputs required	Target outputs often not required

Bengio, Yoshua, et al. "Towards biologically plausible deep learning." *arXiv preprint arXiv:1502.04156* (2015).

SPIKE-TIMING-DEPENDENT PLASTICITY



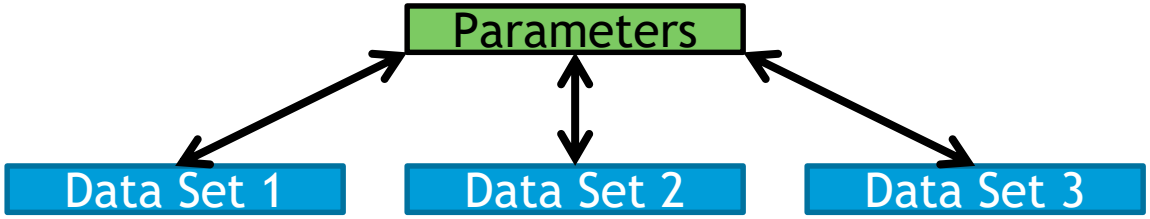


- If pre-synaptic spike occurs before post-synaptic spike:
  synaptic weight is strengthened,
  synaptic change is inversely proportional to spike interval.

- If pre-synaptic spike occurs after post-synaptic spike:
  synaptic weight is weakened,
  synaptic change is inversely proportional to spike interval.

Bi, Guo-qiang, and Mu-ming Poo. "Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type." *Journal of neuroscience* 18.24 (1998): 10464-10472.

HARDWARE ACCELERATION



Costly to train massive neural networks


Yann LeCun, NYU

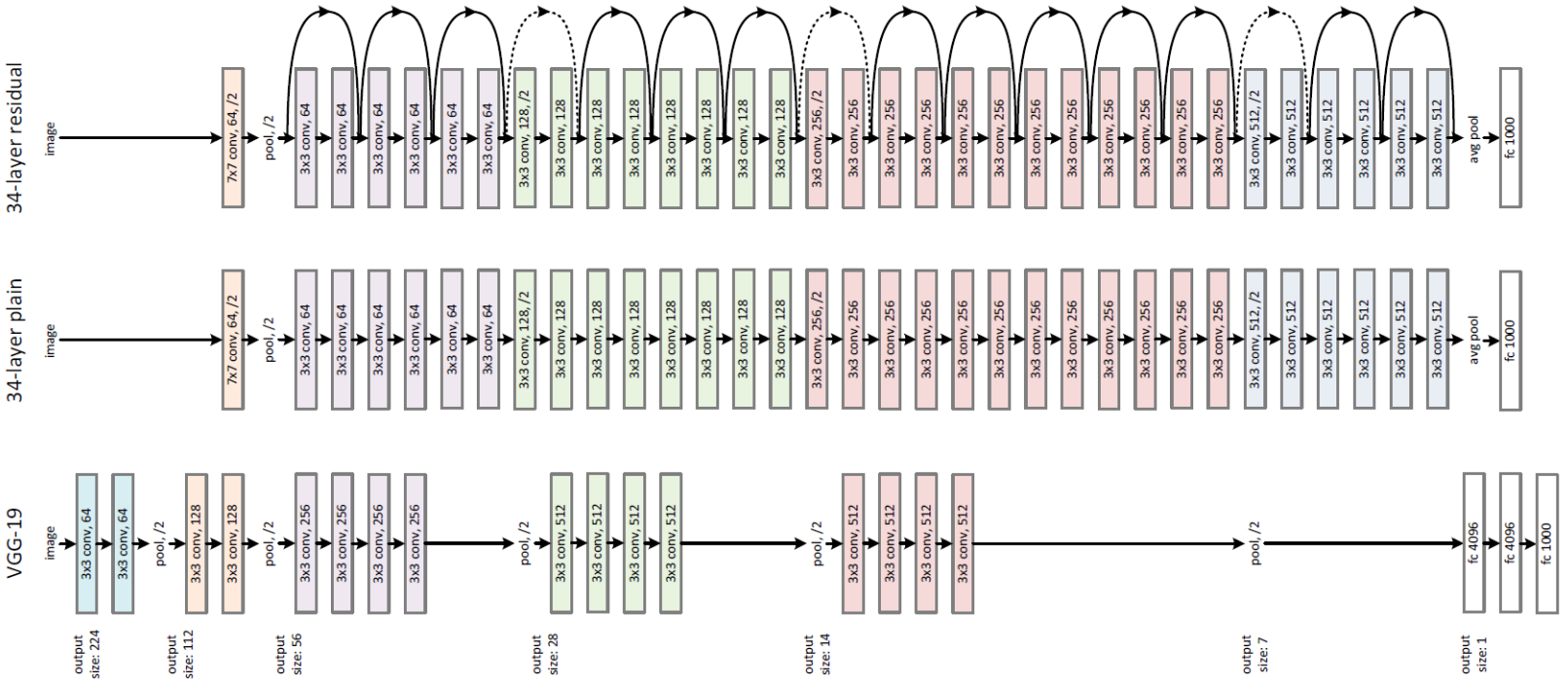
"What we're missing is the ability to parallelize the training of the network, mostly because the **communication is the killer**."

"A lot of people are trying to solve this problem at Baidu, Facebook, Google, Microsoft and elsewhere […] it is **both a hardware and software issue**."

"Eventually a lot of the deep learning task will be done on the device, which will keep pushing the need for **on-board neural network accelerators** that operate at very low power and can be trained and used quickly."

http://www.theplatform.net/2015/05/11/deep-learning-pioneer-pushing-gpu-neural-network-limits/

VERY DEEP NETWORKS

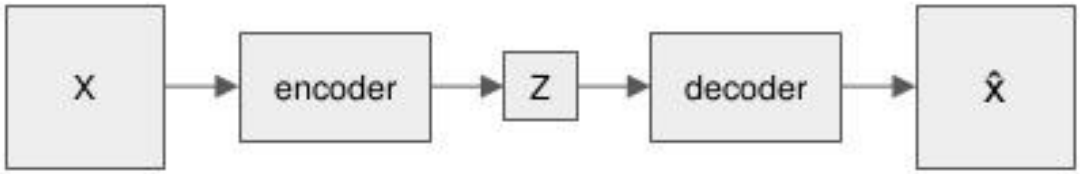


2014 VGG Nets.

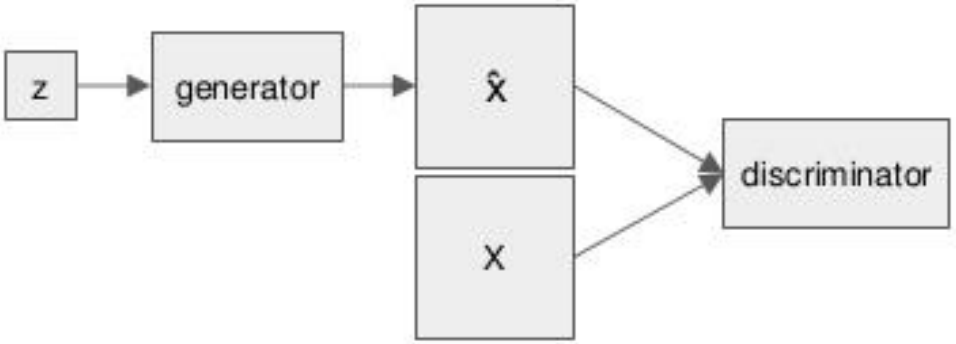
2015 Deep Residual Learning.

ONE-SHOT LEARNING

Two promising approaches



Variational Autoencoders (VAE)
Kingma and Welling [1312.6114]



Generative Adversarial Networks (GAN)
Goodfellow et al. [1406.2661]