Bits, Qubits and Neurons

The near (and not so near) future of Computing

Bill Camp

Preamble

- (This could be used as a course syllabus ☺)

Let’s look at unexpected connections

The Ising Model provides

The simplest non-trivial model of:

- a magnet,
- a ferroelectric,
- a liquid,
- a binary alloy,
- a glass,
- a quantum field theory,
- quantum computing,
- neural nets,

…

The Ising Model in a parallel field

$$H = -\Sigma_{i,j} \{J_{ij}\sigma^z_i\sigma^z_j - \Sigma\, h_i\sigma^z_i\}$$

$$\sigma^z_i = +/- 1$$

Omnis Computatio in tres partes divisus est#

- Turing/VonNeumann Based
- Neuro-Inspired
- Quantum-based

All of computing is divided into three parts-- with apologies to those who have read Caesar

Quantum Computing

Gate and
Circuit-based

Adiabatic Quantum
Relaxation-based

A warp-speed look at quantum mechanics

- Newton got it right if …
- Things weren’t too tiny
- Too heavy
- Or too fast!
- Newton’s physics is deterministic and reversible in the small but irreversible in the large … try unmixing fluids by reversing the paddle.

Demystifying Quantum Physics

- “Things” are described by their “State”, a vector in a linear space of vectors (A Hilbert Space).
- Reversible and Deterministic
  - Until you measure something
- Newton’s physical variables are no longer numbers
  - Position, energy, momentum, angular momentum, dipole moment, …
- They are (self-adjoint) operators in that Hilbert space

Demystifying Quantum Physics

- Self-adjoint operators **A = A***

  (That is, $A_{ij} = A_{ji}^*$) →

- Order matters: Operators don't always commute

  e.g.,$[\mathbf{x},\mathbf{p}] = (\mathbf{x}\,\mathbf{p} - \mathbf{p}\,\mathbf{x}) = ih/2\pi$ Heisenberg!

- Things are measured using inner products
  - Inner product: $<\psi|\phi> = \Sigma_j \psi_j^* \phi_j$ (j=1,N)
  - Measured Energy in state $\psi = <\mathbf{H}> = <\psi|\{\mathbf{H}|\psi>$

Schrödinger's Equation

- $i\delta_t \, |\psi(t)\rangle = \mathbf{H} \, |\psi(t)\rangle$

  (we let $\hbar = h/2\pi = 1$)

- This means that

  $$|\psi(t)\rangle = e^{-i\mathbf{H}t} \, |\psi(0)\rangle$$

- Since $\mathbf{H}=\mathbf{H}^*$, the evolution operator, $\mathbf{U}(t) = e^{-i\mathbf{H}t}$, is unitary:

- $\mathbf{U}^* = \mathbf{U}^{-1} \longrightarrow \mathbf{U}^*\mathbf{U} = \mathbf{I}$

QUBITS

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle$$ is a state with its qubit =0

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle$$ is a state with its qubit =1

For normal bits this is the whole story.

Not so for cubits!

QUBITS

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = |\psi\rangle$$ is a state with its qubit in a superposition of an empty and a full bit:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

This seems weird; but it is true and is the source of (nearly) all of quantum computing's promise!

Quantum parallelism

- We create a quantum circuit using quantum gates.
- Q-gates are unitary operators that operate on the bit-states.
- We can create a starting state for all the bits:
- $|\Psi_0\rangle = |\psi_{0,1}; \psi_{0,2}; \psi_{0,3}; \ldots\psi_{0,j}; \ldots\psi_{0,N}\rangle$
- Where $|\psi_{0,j}\rangle$ (= $\alpha_{0,j}\,|0\rangle + \beta_{0,j}\,|1\rangle$) is the initial state of the jth cubit

Quantum parallelism

To do a calculation we move the N-bit state Ψ_0 through the successive set of L Q-gates:

$|\Psi_f> = \mathbf{U_L U_{L-1} \cdots U_k \cdots U_1} |\Psi_0>$

$|\Psi_f> = |\psi_{f,1}; \psi_{f,2}; \psi_{f,3}; \ldots \psi_{f,j}; \ldots \psi_{f,N}>$

And

$$|\psi_{f,j}> = \alpha_{f,j} |0> + \beta_{f,j} |1>$$

So, we evolve 2^N bit patterns at once!

Unfortunately all that parallelism collapses at output!

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathbf{I}$$

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \boldsymbol{\sigma}^x$$

$$\begin{bmatrix} 0 & -\mathbf{i} \\ \mathbf{i} & 0 \end{bmatrix} = \boldsymbol{\sigma}^y$$

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \boldsymbol{\sigma}^z$$

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \sqrt{2}\bullet\mathbf{H} = \boldsymbol{\sigma}^x + \boldsymbol{\sigma}^z$$

(Hadamard Gate)

$$\begin{bmatrix} 1 & 0 \\ 0 & \mathbf{i} \end{bmatrix} = \mathbf{S}$$

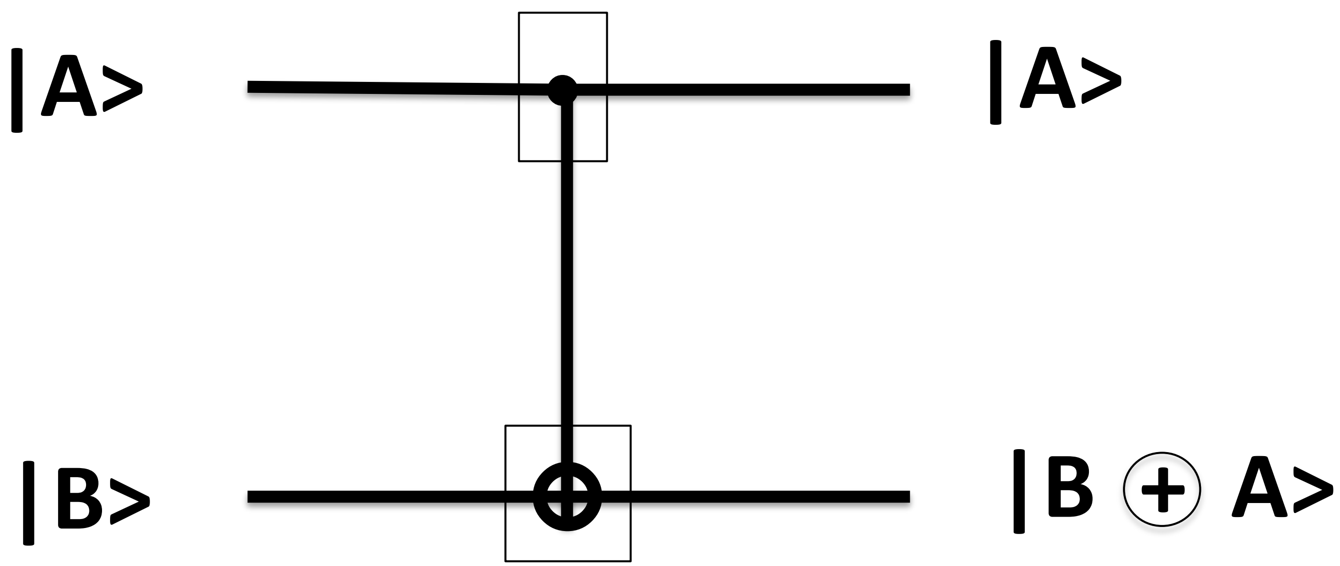
(phase gate)

$$\begin{bmatrix} 1 & 0 \\ 0 & \sqrt{\mathbf{i}} \end{bmatrix} = \mathbf{T}$$

($\pi/8$ gate)

Pauli Operators and 1-bit Q-gates

2-bit Q-gates: C-NOT



2-bit Q-gates: C-NOT

$$\mathbf{U}_{\text{C-NOT}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$\mathbf{U}_{\text{C-NOT}}$ is unitary– check it out!

2-bit Q-gates: C-NOT examples

$$\mathbf{U}_{\text{C-NOT}} \bullet \begin{pmatrix} 0\,0 \\ 0\,1 \\ 1\,0 \\ 1\,1 \end{pmatrix} = \begin{pmatrix} 0\,0 \\ 0\,1 \\ 1\,1 \\ 1\,0 \end{pmatrix}$$

!

2-bit Q-gates: C-NOT general case

$$\mathbf{U}_{\text{C-NOT}} \bullet \begin{pmatrix} \alpha_A\alpha_B \\ \alpha_A\beta_B \\ \beta_A\alpha_B \\ \beta_A\beta_B \end{pmatrix} = \begin{pmatrix} \alpha_A\alpha_B \\ \alpha_A\beta_B \\ \beta_A\beta_B \\ \beta_A\alpha_B \end{pmatrix}$$

!

C-NOT is important

Amazingly C-NOT and the single-bit Q-gates

$$\{I, \sigma^x, \sigma^y, \sigma^z\}$$

are all you need to to create any multi-Q-bit gate.

They form a universal set of building blocks for quantum circuitry

Quantum Noise & Q-ECC

- A real limitation of of quantum computing is noise, which eventually breaks the coherence of quantum superpositions of states.
- Fortunately quantum codes have been invented which, if noise is below a threshold, guarantee coherence.
- Nonetheless, this is not a trivial issue at all!

Is quantum really better?

Sometimes– in principle!

For example,

Thanks to Bob Griffiths' product form for the quantum Fourier transform,

Quantum algorithms can do FTs exponentially faster than the FFT.

Is quantum really better?

Unfortunately it is seemingly impossible to input an arbitrary state to the QFT

It is also impossible in principle to output the QFT

Nonetheless, QFT enables us to do quantum phase estimation, which allows us to do quantum ordering and factoring– exponentially faster than Number theoretic sieves!

Real quantum computers?

- To make quantum computing a reality, we need to be able to
  - Build lots of robust bits
  - Efficiently perform a large set of unitary transformations
  - Prepare initial states reliably
  - Measure outputs

  None of these are currently in good shape for any candidate technology

Takeaway message

- Due to wave function collapse upon measuring output, we only get N-way instead 2^N-way parallelism in the output.
- We cannot query simulations in midstream
- Only in a few cases have quantum algorithms been shown to have huge gains in time to solution
- Quantum seems destined for now to be best for questions that can be with minimal output

Adiabatic Quantum Relaxation

- Create a collection of cubits. Prepare them in the lowest eigenstate of the Hamiltonian of a simple quantum system.
- Slowly transition the system by operating on the qubits with an increasing "perturbation" that effectively removes the original simple Hamiltonian and transitions it to a complex Hamiltonian that represents a "Hard" problem.
- Output the final states.

Topical Example– **Ising Model** in a transverse magnetic field

$$\mathbf{H}(t) = \{1-\tau(t)\}\mathbf{H}_0 + \tau(t)\mathbf{H}_\infty$$

$$\mathbf{H}_0 = -\sum_i h_i \boldsymbol{\sigma}^x_i$$

$$\mathbf{H}_\infty = -\sum_{i,j} J_{ij}\boldsymbol{\sigma}^z_i\boldsymbol{\sigma}^z_j$$

$$[\boldsymbol{\sigma}^z_j, \boldsymbol{\sigma}^x_i] = \boldsymbol{\sigma}^y_j \delta_{j,i}$$

$\tau(0)=0$,
$\tau(\infty)=1$
τ is smoothly Increasing

$(\vec{\sigma} \times \vec{\sigma} = i\vec{\sigma})$!

Ising QAR

- Map the Ising model onto a feasible graph (network), G, with connections representing $\{J_{ij}\}$
- $\{J_{ij}\}$ and G are chosen to map the Ising problem $\mathbf{H}_\infty$ onto the NP-Hard problem to be attacked.
- Evolve the problem on the QAR.

Issues

Adiabatic behavior:

This looks like a continuation method for the ground state of a time dependent Hamiltonian

The starting and ending Hamiltonians do not commute; so they do not share eigenvectors

Ehrenfest's Theorem guarantees success of the adiabatic approximation (continuation):

> if evolution is slow enough and if symmetry issues and level crossings do not violate assumptions.

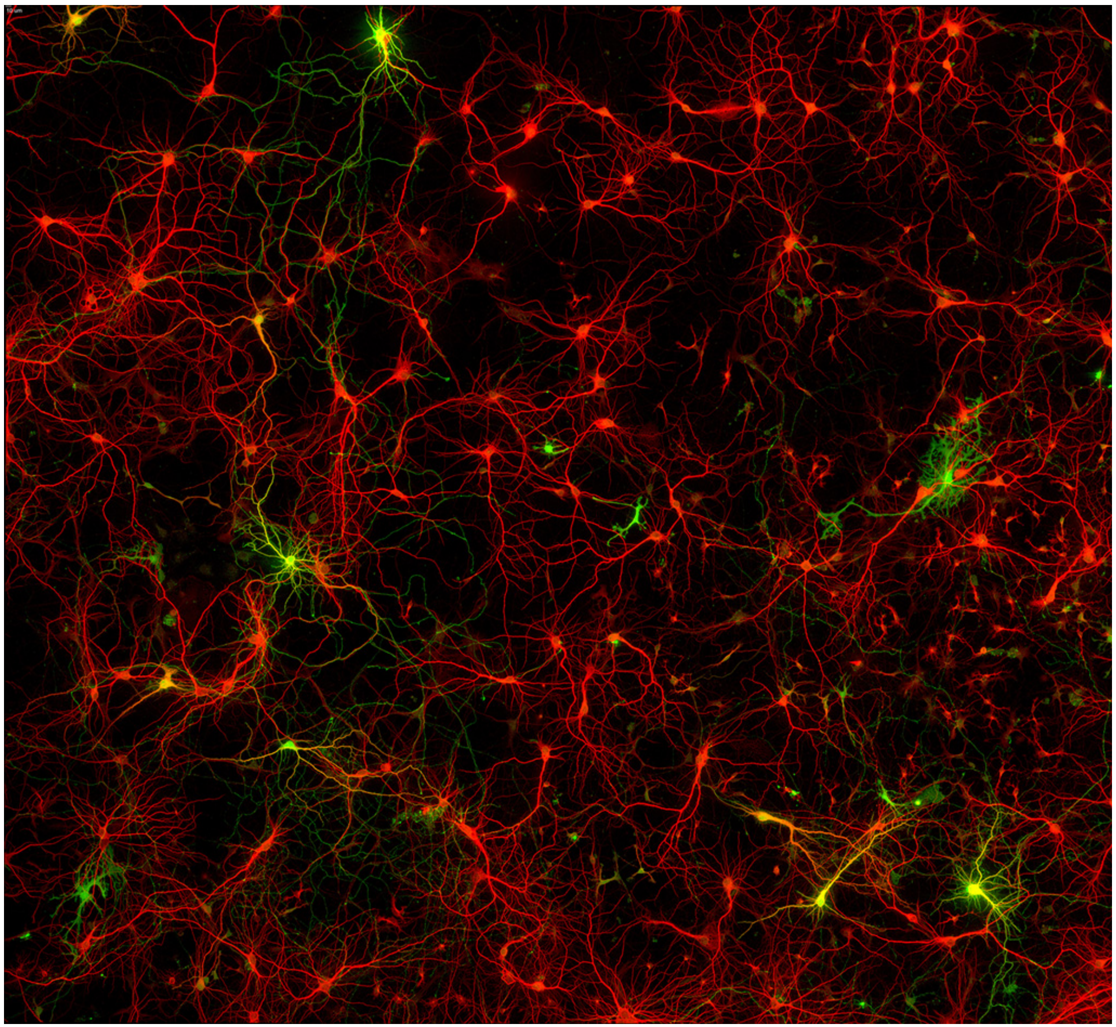
Issues

I am not aware of research that shows definitively when continuation success conditions are met in QARs.

Decoherence due to quantum noise is (in my ken) still poorly understood in QARs

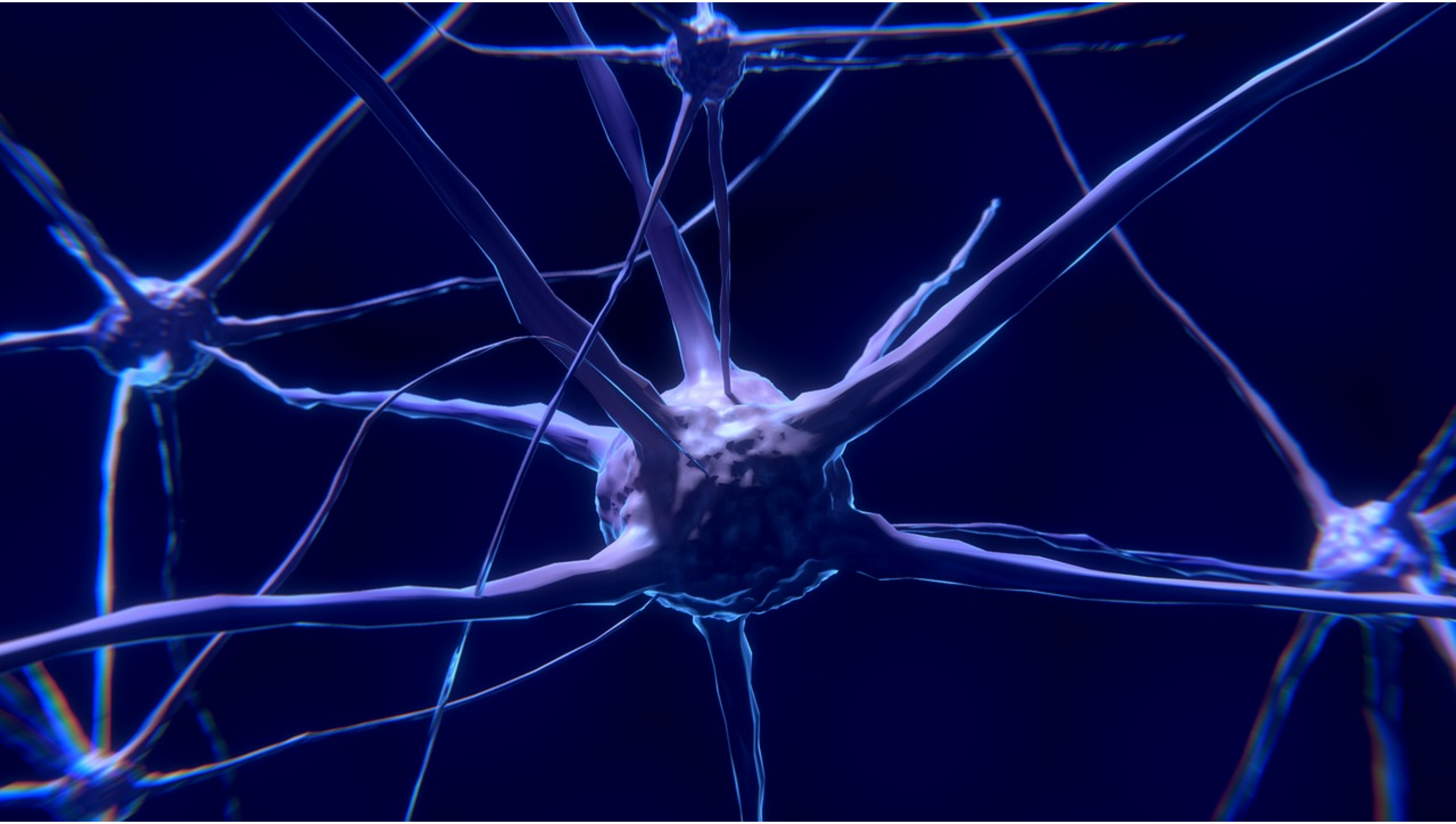
QARs are more limited in scope than circuit quantum computers. How limited is unknown.

Neuro-mimetic computing



Part of a Neural Cortex form a rat brain

BRAIN CELL



Rather more complex than integrating sigmoids, hyberbolic tangents, or rectified linear functions

Nonetheless, it works!

A little history of NN milestones

MACHINE LEARNING/ ARTIFICIAL INTELLIGENCE

- In 1943 McCulloch and Pitts (M-P) introduced the artificial neuron and pointed to pattern classification as front and center to a theory of intelligence!
- In 1949 Hebb proposed that learning changes the morphology of intelligent networks. “The rich get richer and the poor poorer.” (training)
- In 1949, Rosenblatt introduced the perceptron: an M-P network with trainable synapses.
- In 1969, Minsky and Papert pointed out the severe failings of the oringinal perceptron and showed what needed to be done to make it a Turing machine.

MACHINE LEARNING/ ARTIFICIAL INTELLIGENCE

- In 1982, Hopfield introduced the Hopfield net– the basis for many of our modern advances, though not powerful by today's standards.
- In 1982, Hinton et al. introduced Restricted Boltzmann Machines
- In 1984, Fukushima's Neocognitron overcame many of the problems of scale, translation and rotation.
- In 1985, Amit pointed out that the Hopfield Net was in many ways identical to the Ising Spin Glass in its Mean Field Approximation.

MACHINE LEARNING/ ARTIFICIAL INTELLIGENCE

- In 1986, Hinton et al. introduced back propagation as a learning method.
- In 1987, Wolnes discovered that protein folding is also strongly analogous to the behavior of Ising Spin glasses– and discovered minimal frustration and spin funnels.
- In 1987, Pederson and Anderson showed how to speed up SBMs and RBMs by using the Ising Mean Field approximation
- In 2004, LeCun and Bottou emphasized the role of stochastic gradient descent in deep networks.

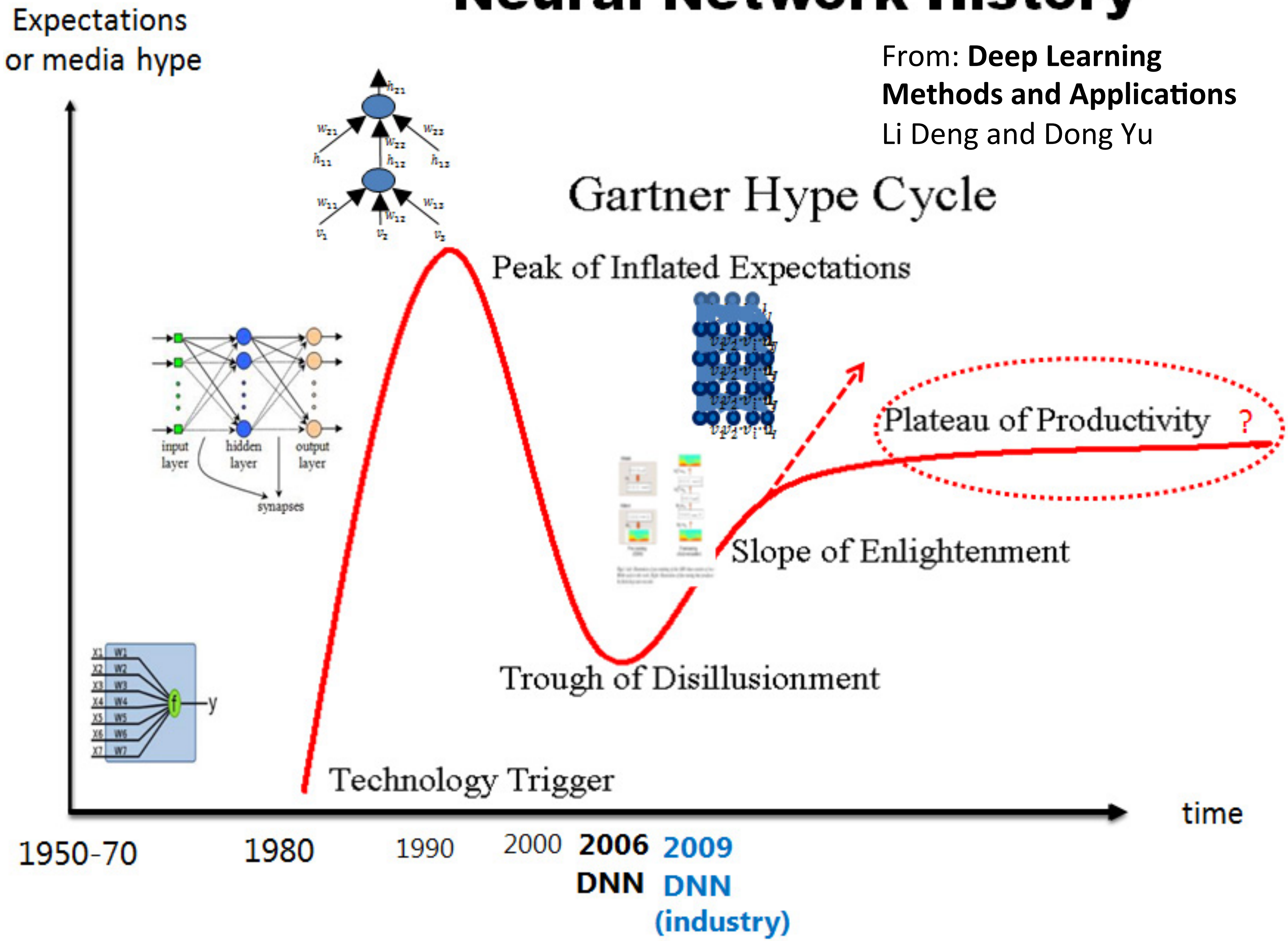
MACHINE LEARNING/ ARTIFICIAL INTELLIGENCE

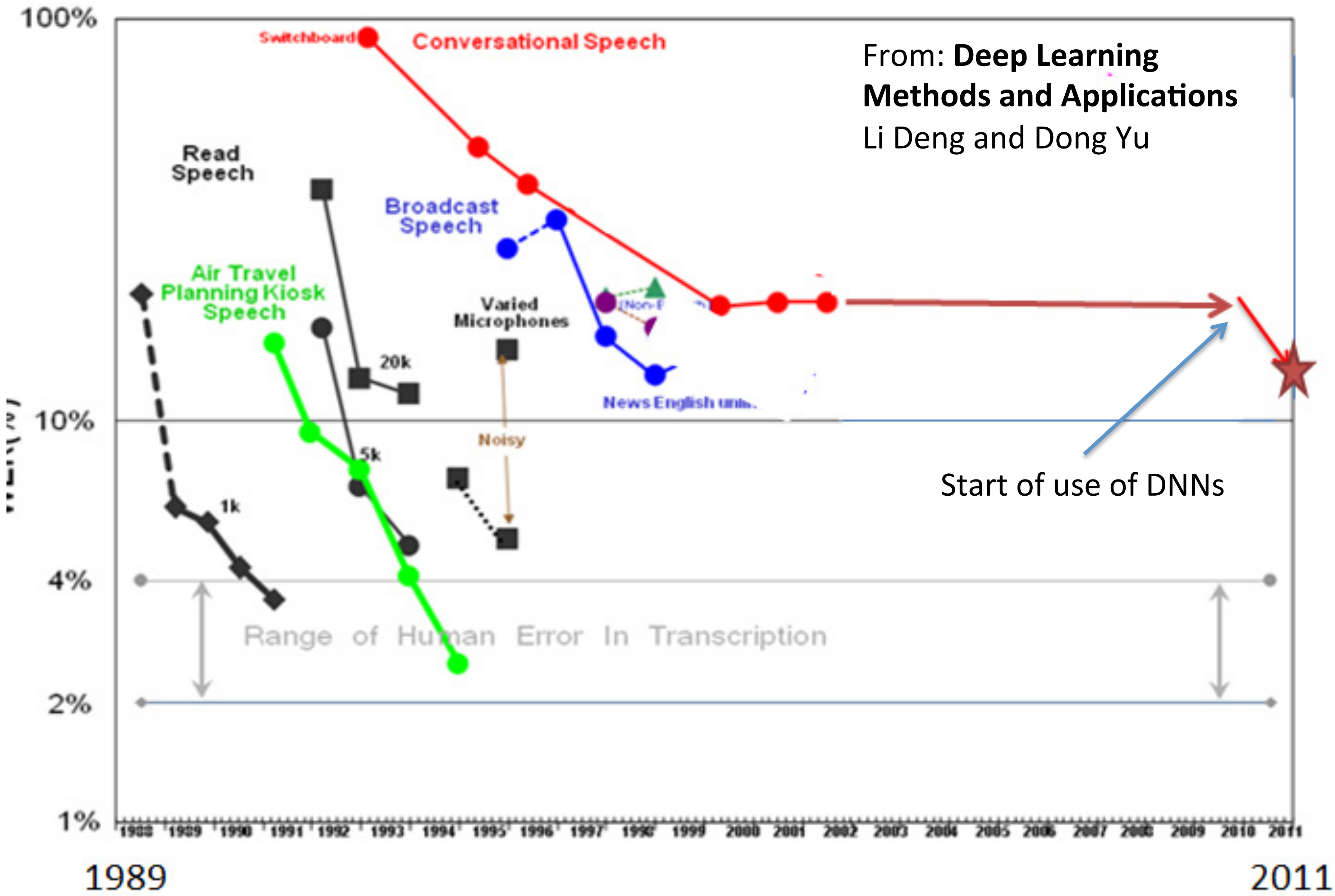
- In 2006, Hinton et al. introduced deep belief networks
- In 2006, Bengio et al. analyzed Deep auto-encoders: Deep networks with greedy layer-by-layer training
- In 2012, Hinton et al. analyzed and emphasized the role of hierarchical abstraction in Deep Neural Networks.

MACHINE LEARNING/AI

- In 2014, Mehta and Schwab showed an exact mapping between Kadanoff's variational approximation for the Renormalization Group approach to the Ising Model and deep neural nets!
- In 2015 LeCun et al demonstrated the relationship of deep NNs to the Ising spin glass in the spherical approximation– Insights galore!

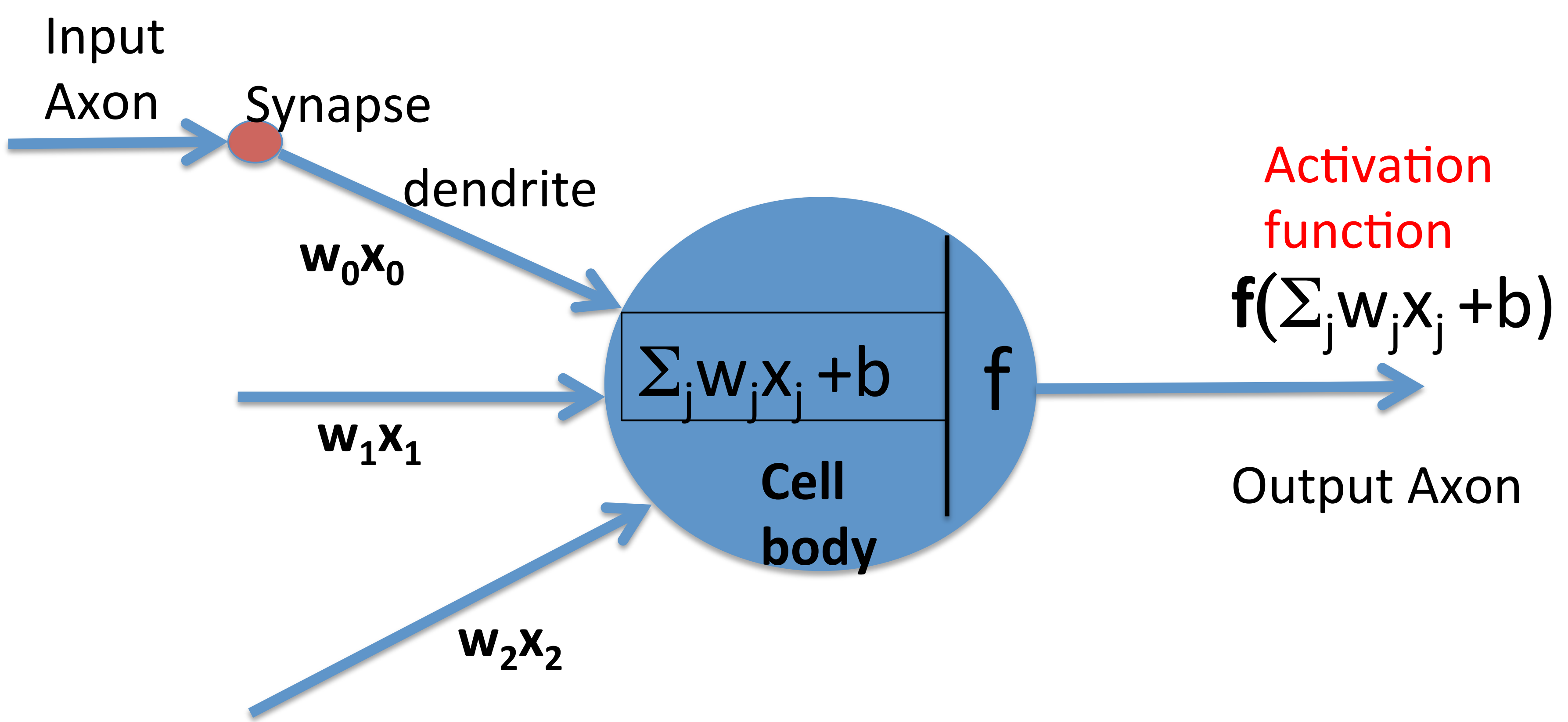
Neural Network History





100%
10%
4%
2%
1%
Switchboard
Conversational Speech
Read
Speech
Broadcast
Speech
Air Travel
Planning Kiosk
Speech
Varied
Microphones
20k
5k
1k
Noisy
News English
Range of Human Error In Transcription
From: **Deep Learning Methods and Applications**
Li Deng and Dong Yu
Start of use of DNNs
1989
2011

Simplified neuron



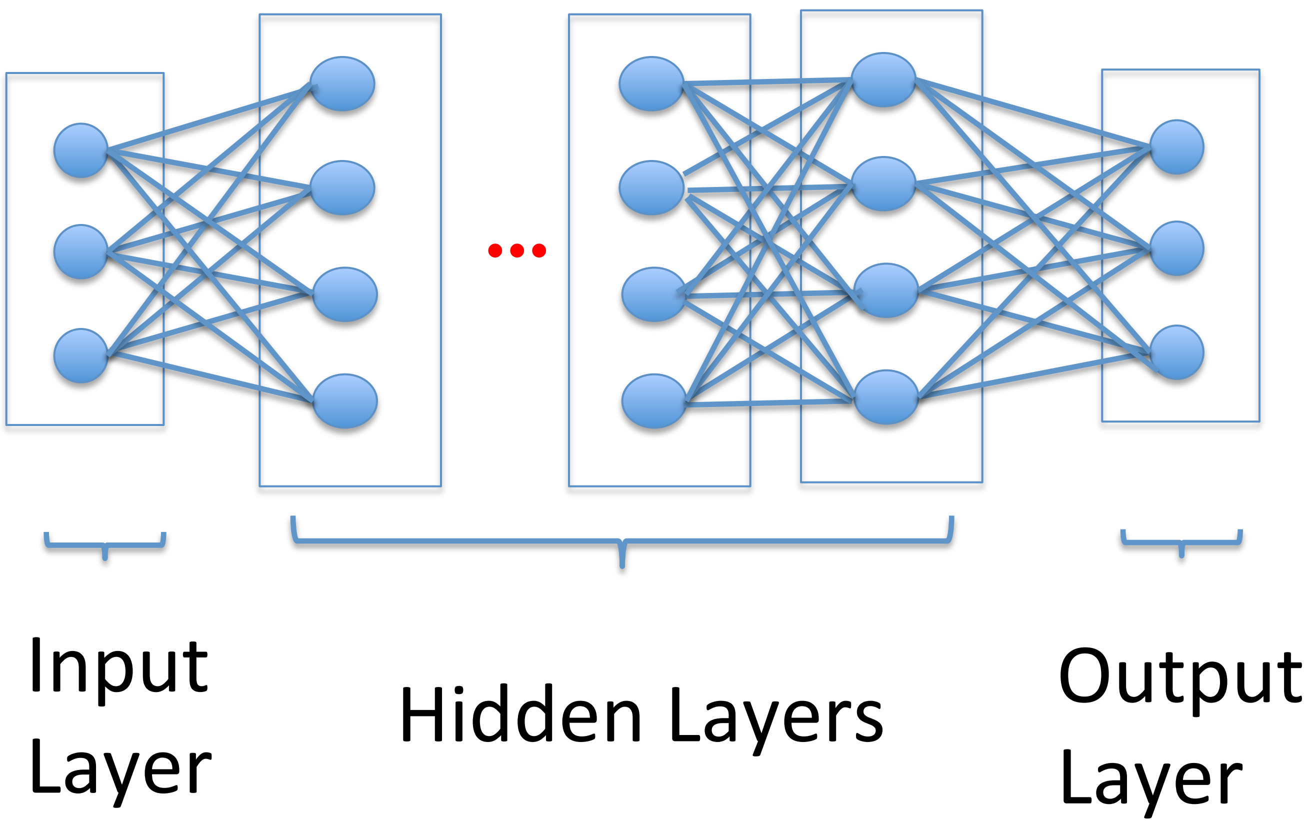
Activation Functions

Sigmoid: $f(x) = 1/(1 + e^{-x})$ (most like biology)

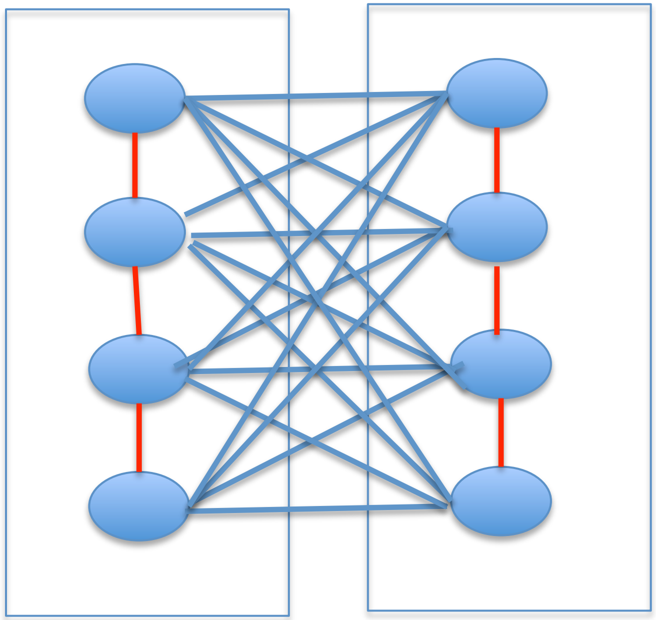
Hyperbolic Tangent: $f(x) = [e^{x} - e^{-x}]/[e^{x} + e^{-x}]$
(most useful for Ising Model comparisons)

ReLU: $f(x) = \Theta(x) \bullet x$ **($\Theta(x)$ is the unit step f^{n})**
(currently most favored)

Prototypical Neural Networks



Two layers of a Network topology for a Stochastic Boltzmann Machine



(Restricted Boltzmann Machine looks like this without the (red) intra-layer connections)

Loss function for an RBM

$H = -\sum_{v,h} \{J_{v,h}\mathbf{t}^z{}_v\mathbf{t}^z{}_h - \sum_{i=v,h} b_\mathbf{i}\mathbf{t}^z{}_i\}$

v is summed over the visible units

v is summed over the hidden units

$\mathbf{t}^z{}_i = 0,1\ (=(2\ \sigma^z{}_i\ -1)/2$

This is the Ising model on a bi-partite graph!

What do we (typically) do with NNs?

- A neural network is presented with training data.
- Its job is to learn that data then extrapolate form it to correctly recognize and classify new data.
- Trivial example: show the NN many pictures of cats and then ask it to classify other pictures by whether they contain cats and perhaps how many cats they contain, and perhaps their breeds, and perhaps their maturity, … .

How NNs do it is fascinating!

- 1. Preparation of data
- 2. Initialization of the net
- 3. Pre-training if used
- 4. Forward propagation: learning by activating neurons
- 5. Back-propagation: changing the weights in the activation functions to minimize the "Loss" function (the error in learning)
- 6. repeat
- Typically there are two visible layers (in, out) and one or more "hidden" layers.
- 7. Output optimal classifications

Deep Neural Networks (DNNs)

- **DNNs** are networks with **many hidden layers**
- (typically 10—20 layers)
- They are **huge**
  - (often 100 million plus network parameters)
- Their input layer is often a **Deep Belief Network** (DBN): a stack of **restricted Boltzmann machines used on learning**

Deep Learning

- In Deep Learning, learning and training is done layer-by-layer from input through many hidden layers to output
- As we go up the stack of layers learning/ training becomes more abstract and powerful.
- This was a real surprise!
- Why it happens is not fully understood

Deep Learning

- Recently, evidence is mounting that the answer lies in two pieces of physics:
  - the Ising Model and its Spin Glasses
  - the resolution of their deep physics via the Renormalization Group Theory (RGT).
- This is also happening in attempts to develop successful folding methods for proteins.

The rest of the story

- I have glossed over nearly 75 years of R&D!
- I haven't told you how RGT works
- I haven't told you how several of us (myself and, independently, Ken Wilson, and a couple of years later Masuo Suzuki) discovered that the Ising Model in d-dimensions is 1-1 and onto a (scalar) quantum field theory in d-1 dimensions (50 years ago!)
- I haven't told you how the Ising model can be used in showing that in an infinite universe there are an infinite # of Bill Camps telling this story to an infinite# of identical audiences
- I haven't discussed Landauer, or Black holes, the holographic theory of memory, information and Maldecena universes.

God willing and the Creek don’t rise, there’s always next year ☺