AGENDA

- Establish what is “Intelligence”
- Design two Neurocomputers
- Extend Machine Intelligence to Robotics
5 Questions to help define what is “Intelligence”

- What can a human do?
- What can a lesser animal do? (a spider for example)
- What can a conventional computer program do?
- What can symbolic Artificial Intelligence (AI) do?
- What can an artificial Neural Network (NN) do?
Two major Machine Intelligence fields of study

- **Symbolic AI programs**
  - Heuristics, inference, hypothesis-testing, and forms of knowledge representation
  - “Expert Systems”
  - Predicate Calculus, PROLOG, LISP

- **Artificial Neural Networks**
  - Connectionist architecture (hardware or software) similar to biological brain
  - Trained (learns) by changing strength of inter-neuron connections
  - Satisfy multiple constraints
Example Neural Network ("back propagation")

<table>
<thead>
<tr>
<th>X1</th>
<th>X2</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Example Neural Network (“back propagation”)

Table:

<table>
<thead>
<tr>
<th>X1</th>
<th>X2</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Example Neural Network ("back propagation")

LEARN by changing "weights" of connections between neurons:

\[ \Delta W_{jk} = \eta \times [(d_k - O_k) \times O_k \times (1 - O_k)] \times O_j \]

\[ \Delta W_{ij} = \eta \times (O_j \times (1 - O_j)) \times \sum_k [(d_k - O_k) \times O_k \times (1 - O_k) \times W_{jk}] \times O_i \]
Example Neural Network ("back propagation")

Weights changing (struggling) to find values that satisfy all exemplars simultaneously.

One Epoch = one forward feed of all exemplars through network AND one change of all weights via backpropagated error (= desired output – actual) for each of exemplar.

Many more Matlab examples programmed in J. Wunderlich courses.
Neural Network **learning** for this type

Neural Net is gradient decent minimization on an Error Surface

where “\(d\)” = Desired Output” and “\(O\)” = present output

\[ E = (d - O) \]
… where the neuron transfer function needs to be continuously differentiable

Transfer function = $1/(1+\exp[-\text{sum}])$
... and learning is most noticeable (and mathematically magnified) when neuron input summation is very small (where \( j \) is the middle “hidden” layer and \( k \) is the output layer)
i.e., it’s in a “deciding position”

\[
O_j = f(\text{sum}_j)
\]

\[
O_k = f(\text{sum}_k)
\]

\[
O_j' = f'(\text{sum}_j) = O_j(1 - O_j)
\]

\[
O_k' = f'(\text{sum}_k) = O_k(1 - O_k)
\]
5 Questions, 42 Mental Abilities to help define what is “Intelligence”

- Answer:
  1. What can a human do?
  2. What can a spider do?
  3. What can conventional computer do?
  4. What can symbolic AI do?
  5. What can Neural Network do?

- for:
  - 20 “Basic Animal Abilities”
  - 22 “Complex Abilities”
Humans rely heavily (perhaps too much) on Standardized tests (SAT, GRE, MCAT, LAT, etc) for these

Bug solves placing web.
- Knowledge of environment and prey

Conventional and intelligent machines
- Solve problems and retain knowledge
- Only differ in memory capacity, method of storage, and class of solvable problems
Animals learn & adapt to new environments
In real and evolutionary time
Conventional computers don’t (need human)
Symbolic A.I. can reason
  Somewhat adaptable to new input
Neural networks better
  Generalize when presented new inputs
  Extremely fast in real-time when embedded
“Mobility,” “Acquisition,” “Protection”
- Essential for animal survival

- Partially solved by conventional and intelligent machines
  - Robotic Motor control
  - Power Supplies
  - Firewalls

(4) Motor Coordination
(5) Acquire Energy
(6) Protect self
Animals sense, think quickly/instinctually
- Anticipate outcomes
- Predict based on extrapolating info

Conventional and *intelligent* machines do this
(with the exception of instinct)
- Neural networks outperform symbolic A.I. when dealing with new stimulus
  - Can be much faster (especially if embedded)
(12) Communication

- Animals, conventional computers, and *intelligent machines* all communicate.
- Nothing close to human natural language processing
  - Symbolic A.I. attempted for decades
- Neural networks very successful in speech recognition (IBM ViaVoice, DragonSpeak, etc.)
  - Including understanding “context”
  - And perhaps not loosing meaning in translations
- Krushchev’s *“We will bury you”* was actually *“Whether you like it or not, history is on our side. We will dig you in“* or *“We will outlive you”*

VIDEO: [http://www.youtube.com/watch?v=OwJHg9UBNPE](http://www.youtube.com/watch?v=OwJHg9UBNPE)
"To derive or induce a general principle or concept from particulars"

Animals do well

Other than simple sorting, conventional computers and code typically don’t

- Give specific responses to specific inputs

Symbolic AI only do somewhat

Neural networks good at

- Generalize so outputs produced to “best fit” (classify) a set of inputs (even when differ from what trained with)
(14) Associate,
(15) Recognize patterns

- All animals do well
- No animal surpasses human’s ability to associate concepts and memories
- Conventional computers correlate data and recognize patterns
- **Symbolic AI** does better
  - But limited by fixed structure
  - “State-space” fixed regardless of efficiency of search
- **Neural networks** even better
  - Widely used for recognizing image and speech
Evolution insures animals partially function when some subsystems fail (including brain)

Conventional computers can’t
- Bit error can cause “lock-up”

Symbolic AI also likely to not function when the underlying computer hardware fails

Neural networks robust under partial failure
- Will partially function if embedded NN hardware or NN software simulation looses some neurons or inter-neuron connections; however if NN is running on a computer that fails, it will fail
Autonomous thought,

- Animals free to make decisions
- Conventional and symbolic AI programs respond in pre-programmed way, but, for example, robotic autonomous global path-planners are very much capable of incorporating symbolic AI
- Neural networks can be also be part of these planners
Animals free to make decisions
  - Struggle with programming dictated by genes
    - Including drive to reproduce

Far in the future, if ever, for *intelligent* machines
Some uncertainty with all animals

- *Allows free will?*

Conventional and *symbolic AI* don’t have free will, but are generally repeatable, predictable, stable, and therefore traceable

*Neural networks* can produce unexpected results, especially with never-seen input, and even slightly different results with identical inputs

- This is why FDA has denied permitting their use in pharmaceutical quality control
Multitask

- Biological evolution yields brains with multiple subsystems
  - Regulation
    - Pulmonary, respiratory, temperature, and motor control
  - Pre-processing
    - Visual cortex, etc.
  - Higher reasoning
  - But we may be overloading!

- Conventional computers becoming better

- In *symbolic AI* when written for multi-processor

- Significant for *neural network* learning
(21) Abstraction
(22) Intuition
(23) Common sense

- Abstract: “.. only intrinsic form ... no attempt at pictorial representation or narrative content”
- Intuition: "Knowing without conscious reasoning”
- Common sense: "Sound and prudent but often unsophisticated judgment "

- All Unlikely for a bug
- Conventional and symbolic AI programs can’t do
- Neural network’s generalize (abstraction?)
Manipulate tools

- Evolved animals
  - Humans became bipedal so they could use front legs for manipulation of tools

- Spider can’t envision extensions of its appendages

- Conventional and intelligent machines can
  - Signals to actuators to position & orient tools (robotic-arm)
  - Not only what arm holds, but arm itself is a tool to manipulate physical world
Spiders, unlike humans, don’t typically:
- Consider every way to react
- Recognize when one scenario infers another
- Solve problems by testing multiple hypotheses

Conventional computers do (somewhat)
- Symbolic AI (especially “Expert Systems”) do exactly this!
- Most neural networks can’t
  - But somewhat infer results
Humans override “animal drives” and develop rules to maintain civilization

Bugs don’t

Conventional computers don’t

**Symbolic AI** programs can incorporate rules
  - And therefore implied “values”

**Neural networks** can’t
  - However could be trained to respond “ethically”
Selective awareness (filtering)

- Animals focus on task
  - Ignoring distractions
- Find images semi-obscured by camouflage or clutter
- Conventional and symbolic AI programs pre-process
  - Signal and image processing
- Neural networks perform well with "fuzzy" input
Tracing mental thoughts less “exact” than tracing execution of conventional or symbolic AI program

- Neural networks less open to inspection
  - Many compromises made in changing inter-neuron weights during learning
To feel, imagine & create, have passion & ambition, and experiment through playful curiosity is still primarily human.

Play needed for evolution.

Unlikely for bugs.

No conventional or *intelligent* machine fully does this YET, but we’re getting closer every day. These processes require multiple parts of our brains.
Empathizing feelings of others, taking risks for others, and displaying leadership (vision, compassion, motivation) is primarily human.

- Unlikely for a bug
- No conventional or intelligent machines do this yet, and it’s highly unlikely free people would want this.
Humans see themselves, their lives, their influence on others, their influence on the future, and their mortality

Unlikely for bugs

Conventional computers don’t

Intelligent machines becoming self-aware???

But machine is “immortal” if ample supply of replacement parts
Humans play, work, raise children, and wage war as teams, and can collectively share beliefs.

Other animals work as a collective (not spiders).

Conventional computers (including mobile devices), and *intelligent* machines can be networked:

- This has lead to group psychology!
- Could social networking be a "*macro*" extension of the "*micro*" distributive brain-center processes needed for our most complex mental abilities?
2013 Sustainability Conference in Japan with my son Joseph:
TALK: http://users.etown.edu/w/wunderjt/Green_Social_Designs_Japan_TALK_19_PLUS.pdf
2013 Sustainability Conference in Japan with my son Joseph:


TALK: [http://users.etown.edu/w/wunderjt/Green_Social_Designs_Japan_TALK_19_PLUS.pdf](http://users.etown.edu/w/wunderjt/Green_Social_Designs_Japan_TALK_19_PLUS.pdf)
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Acquire and retain knowledge</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>Solve problems</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>
| 3 | Learn and adapt | yes | yes | no | somewhat | yes | Evolution  
| 4 | Motor coordination | yes | yes | somewhat | somewhat | somewhat | Survival  
| 5 | Acquire energy | yes | yes | somewhat | somewhat | somewhat | Survival  
| 6 | Protect self | yes | yes | somewhat | somewhat | somewhat | Survival  
| 7 | Sensory processing | yes | yes | yes | yes | yes |  
| 8 | Real-time thought | yes | yes | yes | yes | yes |  
| 9 | React instinctively | yes | yes | no | not yet | not yet |  
| 10 | Anticipate | yes | yes | yes | yes | yes |  
| 11 | Predict | yes | yes | yes | yes | yes |  
| 12 | Communicate | yes | yes | yes | yes | yes |  
| 13 | Generalize | yes | yes | no | somewhat | yes |  
| 14 | Associate | yes | yes | somewhat | somewhat | somewhat | yes |  
| 15 | Recognition patterns | yes | yes | somewhat | somewhat | somewhat | yes |  
| 16 | Robust under partial failure | yes | yes | no | no | yes |  
| 17 | Autonomous thought | yes | yes | if programmed | somewhat | soon | How tightly to hold the leash?  
| 18 | Drive to reproduce | yes | yes | no | not yet | not yet |  
| 19 | Stability, repeatability, predictability | somewhat | somewhat | yes | yes | somewhat | Uncertainty  
| 20 | Multitask | to a point | yes | yes | no | yes |  

**BASIC ANIMAL ABILITIES:**
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>21 Abstraction</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>somewhat</td>
</tr>
<tr>
<td>22 Intuition</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>no</td>
<td>not yet</td>
<td>soon</td>
</tr>
<tr>
<td>23 Common sense</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>not yet</td>
<td>soon</td>
<td></td>
</tr>
<tr>
<td>24 Manipulate tools</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>Evolution</td>
</tr>
<tr>
<td>25 Heuristics</td>
<td>yes</td>
<td>yes</td>
<td>somewhat</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>26 Inference</td>
<td>yes</td>
<td>yes</td>
<td>somewhat</td>
<td>yes</td>
<td>somewhat</td>
<td></td>
</tr>
<tr>
<td>27 Hypothesis testing</td>
<td>yes</td>
<td>somewhat</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>28 Self-discipline, impulse-control</td>
<td>yes</td>
<td>unlikely</td>
<td>somewhat</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>29 Ethical behavior</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>somewhat</td>
<td>somewhat</td>
<td>If coded/trained</td>
</tr>
<tr>
<td>30 Selective awareness (filtering)</td>
<td>to a point</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>31 Open to inspection</td>
<td>somewhat</td>
<td>somewhat</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>somewhat</td>
</tr>
<tr>
<td>32 Emotions</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>not yet</td>
<td>soon</td>
<td></td>
</tr>
<tr>
<td>33 Imagination</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>not yet</td>
<td>soon</td>
<td></td>
</tr>
<tr>
<td>34 Creativity</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>not yet</td>
<td>soon</td>
<td></td>
</tr>
<tr>
<td>35 Passion</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>not yet</td>
<td>soon</td>
<td></td>
</tr>
<tr>
<td>36 Playfulness</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>not yet</td>
<td>soon</td>
<td></td>
</tr>
<tr>
<td>37 Empathy</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>not yet</td>
<td>soon</td>
<td></td>
</tr>
<tr>
<td>38 Courage</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>not yet</td>
<td>soon</td>
<td></td>
</tr>
<tr>
<td>39 Leadership</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>not yet</td>
<td>not yet</td>
<td></td>
</tr>
<tr>
<td>40 Self awareness</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>not yet</td>
<td>not yet</td>
<td></td>
</tr>
<tr>
<td>42 Group psychology, Social Networking, and Living in the Cloud(s)</td>
<td>yes</td>
<td>unlikely</td>
<td>somewhat</td>
<td>somewhat</td>
<td>somewhat</td>
<td>Networking, Crowd-sourcing, Socially-networked Design</td>
</tr>
</tbody>
</table>
Two approaches:

1. Circuits similar to brain biology ("Bottom-Up" design)

2. Produce results similar to human thought ("Top-Down" design)
   - Symbolic A.I. Heuristics, Inference, Hypothesis Testing
   - Neural Network Vector and Matrix Calculations
“Bottom-Up” circuits similar to brain biology (e.g., Artificial Dendritic Tree)

by Wunderlich and Elias, 1992
<table>
<thead>
<tr>
<th>&quot;LEVEL&quot;</th>
<th>DEVICES</th>
<th>MACHINE INTELLIGENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedded</td>
<td>Microcontroller, Microprocessor</td>
<td>High-speed real-time-learning neural network. Not for “Symbolic AI”</td>
</tr>
<tr>
<td></td>
<td>ASIC</td>
<td></td>
</tr>
<tr>
<td>PC</td>
<td>Microprocessor</td>
<td>ok for neural network simulations and “symbolic AI programming”</td>
</tr>
<tr>
<td>PC Server or Workstation</td>
<td>Multiple microprocessors</td>
<td>Good for neural network simulations and “symbolic AI programming”</td>
</tr>
<tr>
<td></td>
<td>Silicon Graphics, SUN, IBM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RS6000</td>
<td></td>
</tr>
<tr>
<td>Mini-Computer</td>
<td>IBM AS400, Amdahl, HP, Hitachi</td>
<td>Good for neural network simulations and “symbolic AI programming”</td>
</tr>
<tr>
<td>Super-Computer</td>
<td>SMP: IBM S/390, MPP: IBM SP2</td>
<td>Great for neural network simulations and “symbolic AI programming”</td>
</tr>
<tr>
<td></td>
<td>(e.g., “Deep Blue”), Vector:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CRAY</td>
<td></td>
</tr>
</tbody>
</table>


Wunderlich, J.T. (2008). **Two single-chip neurocomputer designs; one bottom-up, one top-down.** *(invited journal paper in peer-review)*
TOP-DOWN VS. BOTTOM-UP NEUROCOMPUTER DESIGN

JOSEPH T. WUNDERLICH, Ph.D.
Agenda

- Top-down vs. bottom-up
  - Concepts
  - Mathematics
  - Learning
  - Two examples
    - Bottom-up artificial dendritic tree VLSI chip
    - Digital neurocomputer chip for top-down neural networks
Biological neuron for “bottom-up” neurocomputer design

LEARN BY STRENGTHENING, WEAKENING, AND GROWING NEW CONNECTIONS

STIMULI FROM ENVIRONMENT AND OTHER NEURON OUTPUTS

DENDRITIC TREE

NEURON

OUTPUT TO OTHER NEURONS
Behavioral model for “top-down” neurocomputer design

- Actual Response
- Error
- Desired Response

Learn by adapting to minimize error

Network

Desired response

Actual response

Learn by adapting to minimize error

Stimuli
Analog circuit representation of biological model

\[ R_m = \text{Neuron membrane resistance} \]
\[ C_m = \text{Neuron membrane capacitance} \]
\[ R_a = \text{Serial axial cytoplasm resistance} \]
\[ V_b = \text{Resting neuron membrane voltage} \]
\[ V(X,t) = \text{Neuron membrane voltage at location X and time t as a result of a current density } I(X,t) \]

\[ R_m \cdot C_m \cdot \frac{\partial V(X,t)}{\partial t} = \frac{R_m}{R_a} \cdot \frac{\partial^2 V(X,t)}{\partial X^2} - V + R_m \cdot I(X,t) \]
Analog circuit for "bottom-up" design

\[ V_{out} = \frac{V_I}{R_a + R_m} \]

- Output is a transient output voltage spike
- Learning involves varying inhibitions and excitations to minimize error between actual and desired output

**Symbols:**
- \( R_m \) = Neuron membrane resistance
- \( C_m \) = Neuron membrane capacitance
- \( R_a \) = Serial axial cytoplasm resistance
- \( V_b \) = Resting neuron membrane voltage
- \( V_E \) = Excitation voltage (i.e., \( V_E > V_b \))
- \( V_I \) = Inhibitory voltage (i.e., \( V_I=GND \))
Mathematical models considered for “top-down” neurocomputer

- Backpropagation
- MADALINE III
- Hopfield
- BOLTZMANN MACHINE,
- BAM (Bi-directional Associative Memory)
- NEOCOGNITRON
Layered neural network for "top-down" design

Sigmoid transfer function:

\[ O_j = \frac{1}{1 + e^{-(j\text{BIAS} \times W_{j\text{BIAS}}) + \sum_i (-O_i \times W_{ij})}} \]

Learn by changing weights as a function of error:

\[ \Delta W_{jk} = \eta \times [(d_k - O_k) \times O_k \times (1 - O_k)] \times O_j \]

\[ \Delta W_{k\text{BIAS}} = \eta \times [(d_k - O_k) \times O_k \times (1 - O_k)] \times k\text{BIAS} \]

\[ \Delta W_{ij} = \eta \times (O_j \times (1 - O_j)) \times \sum_k [(d_k - O_k) \times O_k \times (1 - O_k) \times W_{jk}] \times O_i \]

\[ \Delta W_{j\text{BIAS}} = \eta \times (O_j \times (1 - O_j)) \times \sum_k [(d_k - O_k) \times O_k \times (1 - O_k) \times W_{jk}] \times j\text{BIAS} \]

**SIMPLE “XOR” EXAMPLE**

<table>
<thead>
<tr>
<th>Input/Output Exemplars</th>
<th></th>
<th>Desired Output (d_k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X_1</td>
<td>X_2</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Possible backprop neurocomputer chip implementations

- Discrete analog components for all computations
- Digital circuits for all computations except transfer function implemented as serial or parallel analog circuits
  - Analog circuits can suffer from drift, fabrication inconsistencies, and conversion delays
- Digital circuits for all computations including transfer function implemented as serial or parallel look-up tables
  - Parallel high-precision look-up tables use much real-estate
- Parallel vector-register digital circuits for all computations including a polynomial approximation of the transfer function (CHOOSEN APPROACH)
  - Scalable, fast, single-chip with on-chip learning
Polynomial approximation techniques considered

- Taylor Series
- Hermite
- Divided Difference
- Cubic Spline
  - This eliminated since piecewise requires separate calculations for each part of spline
First recall that learning is most noticeable (and mathematically magnified) when neuron input summation is very small. So we must maintain the highest of precision in a range of summated inputs centered around zero. The allowable width of the range is found via NN simulations concurrent to the numerical method techniques used to find a polynomial approximation of the Sigmoid transfer function.

\[ O_j = f(\text{sum}_j) \]

\[ O_k = f(\text{sum}_k) \]

\[ O_j' = f'(\text{sum}_j) = O_j(1 - O_j) \]

\[ O_k' = f'(\text{sum}_k) = O_k(1 - O_k) \]
Neuron output for Taylor polynomial approximations of sigmoid:

\[
\frac{1}{1+e^{-x}} \text{ expanded about } x_0 = 0
\]

\[
P_{Taylor}(x) = f(x_0) + f'(x-x_0) + f''(x_0) \left[ \frac{(x-x_0)^2}{2!} \right] + \ldots + f^{(n)}(x_0) \left[ \frac{(x-x_0)^n}{n!} \right]
\]
Error for Taylor series polynomial approximations of sigmoid: $1/(1+e^{-x})$

\[
\text{Error}_{Taylor} = f(x) - P_{Taylor}(x) = (x - x_0)^{(n+1)} \left[ \frac{f^{(n+1)}(\xi(x))}{(n + 1)!} \right]
\]

for some number $\xi(x)$ between $x$ and $x_0$.
Neuron output for Hermite polynomial approximations of sigmoid: 
$1/(1+e^{-x})$ with evaluation points at $x = [-6 \text{ to } +6]$ at 0.25 intervals

$$P_{\text{Hermite}}(x) = \sum_{j=0}^{n} f(x)H_{n,j}(x) + \sum_{j=0}^{n} f''(x)\hat{H}_{n,j}(x)$$

$H_{n,j}(x) = \left[1-2(x-x_j)L'_{n,j}(x)\right]L^2_{n,j}(x)$

$\hat{H}_{n,j}(x) = (x-x_j)L^2_{n,j}(x)$

$L_{n,j}(x) = \prod_{i=0}^{n} \frac{(x-x_i)}{(x_j-x_i)}$
Error for 12th and 13th degree Hermite polynomial approximation of sigmoid: \(1/(1+e^{-x})\)

\[
Error_{Hermite} = f(x) - P_{Hermite}(x) = \left[ \prod_{i=0}^{n} (x - x_i)^2 \right] f^{(2n+2)}(\xi)
\]

for some number between adjacent points in \((x_0, x_1, \ldots, x_n)\)
Neuron output with Divided-Difference polynomial approx of sigmoid: $1/(1+e^{-x})$ with evaluation points at $x = [-6, -5, -4, -3, -2, -1, -0.5, 0.5, 1, 2, 3, 4, 5, 6]$

$$P_{DivDiff}(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots + a_n(x - x_0)(x - x_1)\cdots(x - x_{n-1})$$

$$a_0 = f[x_0] = f(x_0)$$

$$a_1 = f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0}$$

$$a_2 = f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{1}{x_2 - x_0} \left[ \frac{f(x_2) - f(x_0)}{x_2 - x_1} - \frac{f(x_2) - f(x_1)}{x_1 - x_0} \right]$$

$$= \frac{f(x_0)}{(x_0 - x_1)(x_0 - x_2)} + \frac{f(x_1)}{(x_1 - x_0)(x_1 - x_2)} + \frac{f(x_2)}{(x_2 - x_0)(x_2 - x_1)}$$

$$a_n = f[x_0, x_1, \ldots, x_n] = \frac{f(x_0)}{(x_0 - x_1)\cdots(x_0 - x_n)} + \frac{f(x_1)}{(x_1 - x_0)\cdots(x_1 - x_n)} + \ldots + \frac{f(x_n)}{(x_n - x_0)\cdots(x_n - x_{n-1})}$$
Approximation error for 12th degree
Divided-Difference polynomial approximations
of sigmoid: $1/(1+e^{-x})$
Backpropagation learning “XOR” using standard sigmoid

"XOR": Neuron function #0, Weight set #2, Learn rate=1. Stop tol=0.1
Backpropagation learning “XOR” using 10th degree Taylor polynomial approximation of $e^{-x}$ in sigmoid: $1/(1+e^{-x})$. Expanded about $x_0 = 0$, and "clipped" outside of domain $-3.5 > X < 3.5$
Backpropagation learning "XOR" using 12th degree Divided-Difference polynomial approximation of sigmoid, and "clipping" outside of domain $-5.25 > X < 5.25$ to demonstrate robustness of methodology (i.e., 12th degree poly has same accuracy for $-10 > X > 10$)
This top-down design is:

- Vector-register
- Entirely digital
- **Fully parallel** by using a polynomial approximation of sigmoid to allow parallel *on-chip learning*
- (U.S. Patent Disclosure recorded in 1991 for J. Wunderlich design)
- Transistor-count estimate to implement 100 Neurons is 1,000,000
“Bottom-up” neurocomputer
(artificial dendritic tree VLSI chip)
by Wunderlich and Elias, 1992

LEARN BY STRENGTHENING,
WEAKENING, AND
GROWING NEW CONNECTIONS

STIMULI FROM ENVIRONMENT AND OTHER NEURON OUTPUTS

DENDRITIC TREE

OUTPUT TO OTHER NEURONS

NEURON
Bottom-up more difficult to implement

- **Useful** bottom-up designs difficult to realize
  - 3D biological brains vs. 2D IC’s
    - Connection density problems
  - In biological learning, connections not only strengthened or weakened, but are **grown**

![Diagram of neural connections](image)
Read more at:


- Wunderlich, J.T. (2008). **Two single-chip neurocomputer designs; one bottom-up, one top-down.** (invited journal paper in peer-review)
Merging bottom-up and top-down techniques for pre-processing sensory data
- Visual
- Auditory
- Olfactory

And using top-down techniques for higher reasoning
- including combining neural networks with symbolic AI programming

Add machine *intelligence* into robots .......
Europa Rover

Europa is a moon of Jupiter covered in ice

Future NASA/ESA mission is to get to liquid ocean beneath the ice

Image from: http://photojournal.jpl.nasa.gov/catalog/PIA00502
Europa Rover

Europa is much further from Earth than Mars.

And therefore much more autonomous robotics will be needed.

Image from: http://photojournal.jpl.nasa.gov/catalog/PIA00502
Europa  (Cracks in the ice – and an ocean below)

Spots are possibly from warm ocean interacting with surface

Future space missions?

Much intelligence may likely be needed.

VIDEO:  http://www.youtube.com/watch?v=Q3C5sc8b3xM
Asimo’s “Intelligence”

“Intelligence is a technology and a strategy for robust and flexible problem solving in complex environments (both natural and artificial) under the constraints of limited resources (e.g. time, energy). ……. Our brains are designed to achieve autonomous adaptation to a changing world……rapidly and stably self-organize its successful behaviors in response to an unpredictably changing, or non-stationary, world.

Our approach is based on the assumption that the essence of brain computing is not in local processing or learning algorithm but in way brain organizes processing. …….rather than modeling isolated subsystems, large-scale computational models of complete functional blocks at several interacting levels of complexity have to be investigated.

The simulation of large-scale hypotheses on brain function is limited by the available technology. Therefore, at the Honda Research Institute Europe, we investigate different levels in parallel and convey fundamental results between these levels in order to circumvent the incorporation of all complexity levels in one system set-up.“

Prof. Dr. Edgar Koerner
Honda Research Institute Europe GmbH

Asimo’s “Intelligence”

“We target control architectures that are required for brain processing at the following levels:

(1) The control of growth processes and development by gene-regulatory networks
(2) The detailed cortical columnar architectures for self-referential control for storing experience
(3) The behavior based dynamic allocation of systems resources for predictive visual scene analysis
(4) The global behavior control architecture for autonomous interaction of our humanoid robot research platform ASIMO

Step by step we implemented nested control loops for reflexes, attention modulated behavior, on-line learning from sensory experience, and generating predictions for interaction. This enables ASIMO to learn to recognize objects through interaction with humans, to learn associations between acoustic and visual objects, as well as to associate sound with behavioral concepts for interaction, and to demonstrate first steps of prediction driven behavior.”

Prof. Dr. Edgar Koerner
Honda Research Institute Europe GmbH

More on designing autonomy at:
http://users.etown.edu/w/wunderjt/ROME_2011_PAPER9_434_Reading.pdf


<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Acquire and retain knowledge</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>Solve problems</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>Learn and adapt</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>somewhat</td>
<td>yes</td>
<td>Evolution</td>
</tr>
<tr>
<td>Motor coordination</td>
<td>yes</td>
<td>yes</td>
<td>somewhat</td>
<td>somewhat</td>
<td>somewhat</td>
<td>Survival</td>
</tr>
<tr>
<td>Acquire energy</td>
<td>yes</td>
<td>yes</td>
<td>somewhat</td>
<td>somewhat</td>
<td>somewhat</td>
<td>Survival</td>
</tr>
<tr>
<td>Protect self</td>
<td>yes</td>
<td>yes</td>
<td>somewhat</td>
<td>somewhat</td>
<td>somewhat</td>
<td></td>
</tr>
<tr>
<td>Sensory processing</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>Real-time thought</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>React instinctively</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>not yet</td>
<td>not yet</td>
<td></td>
</tr>
<tr>
<td>Anticipate</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>Predict</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>Communicate</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>Generalize</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>somewhat</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>Associate</td>
<td>yes</td>
<td>yes</td>
<td>somewhat</td>
<td>somewhat</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>Recognition patterns</td>
<td>yes</td>
<td>yes</td>
<td>somewhat</td>
<td>somewhat</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>Robust under partial failure</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>Autonomous thought</td>
<td>yes</td>
<td>yes</td>
<td>if programmed</td>
<td>somewhat</td>
<td>soon</td>
<td>How tightly to hold the leash?</td>
</tr>
<tr>
<td>Drive to reproduce</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>not yet</td>
<td>not yet</td>
<td></td>
</tr>
<tr>
<td>Stability, repeatability, predictability</td>
<td>somewhat</td>
<td>somewhat</td>
<td>yes</td>
<td>yes</td>
<td>somewhat</td>
<td>Uncertainty</td>
</tr>
<tr>
<td>Multitask</td>
<td>to a point</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>Abstraction</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>no</td>
<td>somewhat</td>
</tr>
<tr>
<td>22</td>
<td>Intuition</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>not yet</td>
<td>soon</td>
</tr>
<tr>
<td>23</td>
<td>Common sense</td>
<td>yes</td>
<td>yes</td>
<td>not yet</td>
<td>soon</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Manipulate tools</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>Evolution</td>
</tr>
<tr>
<td>25</td>
<td>Heuristics</td>
<td>yes</td>
<td>yes</td>
<td>somewhat</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>26</td>
<td>Inference</td>
<td>yes</td>
<td>yes</td>
<td>somewhat</td>
<td>yes</td>
<td>somewhat</td>
</tr>
<tr>
<td>27</td>
<td>Hypothesis testing</td>
<td>yes</td>
<td>somewhat</td>
<td>yes</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Self-discipline, impulse-control</td>
<td>yes</td>
<td>unlikely</td>
<td>somewhat</td>
<td>some</td>
<td>no</td>
</tr>
<tr>
<td>29</td>
<td>Ethical behavior</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>somewhat</td>
<td>If coded/trained</td>
</tr>
<tr>
<td>30</td>
<td>Selective awareness (filtering)</td>
<td>to a point</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>Open to inspection</td>
<td>somewhat</td>
<td>somewhat</td>
<td>yes</td>
<td>yes</td>
<td>somewhat</td>
</tr>
<tr>
<td>32</td>
<td>Emotions</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>not yet</td>
<td>soon</td>
</tr>
<tr>
<td>33</td>
<td>Imagination</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>not yet</td>
<td>soon</td>
</tr>
<tr>
<td>34</td>
<td>Creativity</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>not yet</td>
<td>soon</td>
</tr>
<tr>
<td>35</td>
<td>Passion</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>not yet</td>
<td>soon</td>
</tr>
<tr>
<td>36</td>
<td>Playfulness</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>not yet</td>
<td>soon</td>
</tr>
<tr>
<td>37</td>
<td>Empathy</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>not yet</td>
<td>soon</td>
</tr>
<tr>
<td>38</td>
<td>Courage</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>not yet</td>
<td>soon</td>
</tr>
<tr>
<td>39</td>
<td>Leadership</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>not yet</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>Self awareness</td>
<td>yes</td>
<td>unlikely</td>
<td>no</td>
<td>not yet</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>Awareness of mortality</td>
<td>yes</td>
<td>unlikely</td>
<td>immortal?</td>
<td>Immortal?</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>Group psychology, Social Networking, and Living in the Cloud(s)</td>
<td>yes</td>
<td>unlikely</td>
<td>somewhat</td>
<td>somewhat</td>
<td>Replaceable parts, Networking, Crowd-sourcing, Socially-networked Design</td>
</tr>
</tbody>
</table>
More on Networking, Crowd-sourcing, and Socially-networked Design at:
TALK: http://users.etown.edu/w/wunderjt/Green_Social_Designs_Japan_TALK_19_PLUS.pdf

2012 Elizabethtown College student socially-networked architecture

2013 Key-note talk in Osaka, Japan

**PAPER:** [http://users.etown.edu/w/wunderjt/Green_Social_Designs_Japan_paper_19.pdf](http://users.etown.edu/w/wunderjt/Green_Social_Designs_Japan_paper_19.pdf)

**TALK:** [http://users.etown.edu/w/wunderjt/Green_Social_Designs_Japan_TALK_19_PLUS.pdf](http://users.etown.edu/w/wunderjt/Green_Social_Designs_Japan_TALK_19_PLUS.pdf)
A collaboration between the United Nations and Mojang, the developers of Minecraft
Minecraft to aid UN regeneration projects

Development plans for 300 places around the world are being modelled in Minecraft so residents can help decide how the locations will change.

Called Block by Block, the programme is part of a collaboration between Minecraft-maker Mojang and UN Habitat.

Urban locations will be recreated on computer using Minecraft allowing residents to take a virtual tour.

Residents will be able to take a virtual stroll around the Minecraft models.

They will also be able to change the model and help decide how regeneration cash should be spent.
Humans play, work, raise children, and wage war as teams, and can collectively share beliefs.

Many other animals work as a collective.

Conventional computers (including mobile devices), and *intelligent* machines can be networked.

Could social networking be a "macro" extension of the "micro" distributive brain processes needed for our most complex mental abilities? – and possibly enhance civilization with a balance of autonomy and group harmony.

**Minecraft to aid UN regeneration projects**

Development plans for 300 places around the world are being modelled in Minecraft so residents can help decide how the locations will change.

Called Block by Block, the programme is part of a collaboration between Minecraft-maker Mojang and UN Habitat.

Urban locations will be recreated on computer using Minecraft allowing residents to take a virtual tour.

They will also be able to change the model and help decide how the locations will change.

Residents will be able to take a virtual stroll around the Minecraft models.
Watch PBS Frontline Video "Digital Nation"
http://www.pbs.org/wgbh/pages/frontline/digitalnation/view/

and question Mental Abilities:

#20 "MULTITASKING"
#30 “SELECTIVE AWARENESS (FILTERING)”
#42 "GROUP PSYCHOLOGY, SOCIAL NETWORKING, and LIVING IN THE CLOUD(s)"

Then read:

http://users.etown.edu/w/wunderjt/ITALY_2009/PUBLICATION_SUBMITPAPERdefininglimitsREVISED16b.pdf