The Active Element Machine

A Simple, Parallel Computing Machine using +, <, and Time on Z

AEMEA
Michael Fiske
San Francisco, California
email: mike@aemea.org

Background & Perspective

B.S. Biology. Emphasis on Neurobiology.

Ph.D. Mathematics. Chaotic Dynamical Systems. Henon Map
$H: P \rightarrow P$ where $H(x, y) = (1 + y - 1.4x^2, 0.3\,x)$.

Post Graduate. Pattern recognition applications.
Handwriting, signature and fingerprint recognition.

Intuition. Current computers are not appropriately designed to effectively implement geometric pattern recognition. Nature computes in parallel and uses time.

What Should a New Computing Machine Do?

1. **Dendritic Integration**
   - Capture useful computational properties of dendrites.   (Wilfrid Rall)
   - Step functions can approximate any measurable function.
   - *Parallel*. The machine uses time.
   - Synapses. What is being computed changes over time.

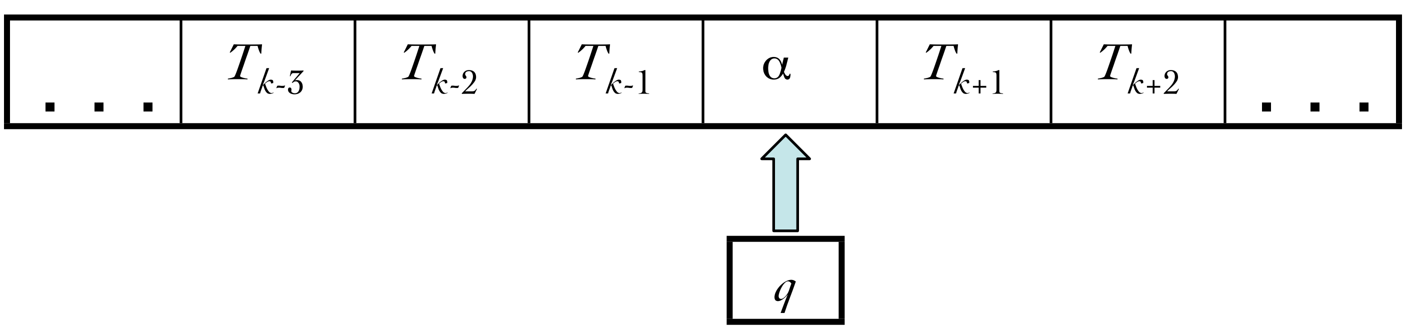
2. **Simple Math. Easy to build in silicon and other hardware**
   - No transfer function as in traditional neural networks.
   - $\boldsymbol{Z} = \{m + k\,\mathrm{dT}$: m, k are integers and dT is a fixed infinitesimal $\}$.
   - Math operators $+$, $>$ and time on $\boldsymbol{Z}$.

3. **Machine and Programming Language**
   - Implicitly programmable: evolution and machine learning.
   - Explicitly programmable: a person can write an AEM program
   - Five commands. Time is in the commands.
   - Machine architecture should be able to change as it is executing.

Turing & Register Machine Model

Turing Machine. The standard computing model.

- Finite set of states $Q = \{q_1, \ldots, q_n\}$. Finite alphabet $A = \{a_1, \ldots, a_m\}$.
- Tape $T : Z \rightarrow A$ with an alphabet symbol on each tape square.
- The Turing program $\eta : Q \times A \rightarrow (Q \cup \{h\}) \times A \times \{-1, +1\}$ is a finite set of rules that stays fixed i.e. the rules do not change as the program executes.
- Computational step: i.e. $\eta(q, \alpha) = (r, \beta, -1)$ or $\eta(q, \alpha) = (r, \beta, +1)$. One rule is selected, based on symbol α and state q. The output replaces α with new symbol β, moves to a new state r and the tape head moves left or right.
- **Computational steps are executed sequentially. No reference to time.**



Active Element Machine computation

1. **Active Elements and Connections**.
   - All elements compute simultaneously.

2. **Connections connect Elements**.
   - Connections determine the messages (pulses) that are sent between elements.

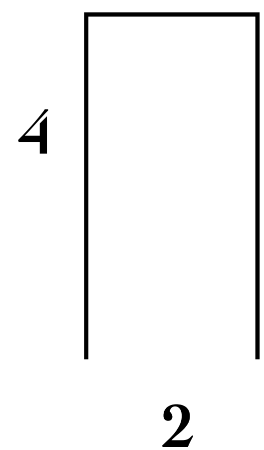
3. **Elements fire and send pulses along Connections**.
   - An element **E** fires at time **s** if the sum of **E**'s input pulses is greater than **E**'s threshold θ and **E**'s refractory period **r** has expired i.e. $\mathbf{s} \geq \mathbf{r} + \mathbf{l}$ where **l** is **E**'s most recent firing time.

AEM computation - Outgoing Pulses

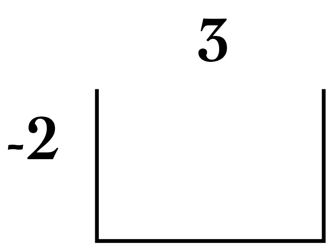
4. **(Connection (Time 4) (From E) (To Y) (Amp -2) (Width 3) (Delay 5)**
   (Connection (Time 4) (From E) (To Z) (Amp 4) (Width 2) (Delay 3)

If **E** fires at time 4, then

- A pulse of time width 3 and amplitude -2 (height - 2) arrives at element **Y** at time 9.
- A pulse of time width 2 and amplitude (height 4) arrives at element **Z** at time 7.



is sent to **Z**



is sent to **Y**

AEM computation - Number Parameters

Extended integers $\{m + k \text{ dT}$: m, k are integers & dT is a fixed infinitesimal$\}$.

Element command parameters:

(Element (Time 3-2dT) (Name E) (Threshold 9) (Refractory 4) (Last 0))

- **(Refractory 4)** The refractory period is a positive integer.
- **(Threshold 9)** The threshold is an integer.
- **(Last 0)** The most recent firing time is an integer.
- **(Time 0)** Time is an extended integer $\boldsymbol{m + k}$ **dT** i.e. 3-2dT

Connection command parameters:

(Connection (Time -1) (From A) (To E) (Amp -7) (Width 4+2dT) (Delay 1-dT)

- **(Amp -7)** Amplitude is an integer.
- **(Width 4+2dT)** Pulse width is $\boldsymbol{m + k}$ **dT** where the standard part $\boldsymbol{m} \geq 1$.
- **(Delay 1-dT)** Transmission time is $\boldsymbol{m + k}$ **dT** where $\boldsymbol{m} \geq 1$.

AEM computation - Input Pulses

(Connection (Time -1) (From A) (To E) (Amp 5) (Width 4+2dT) (Delay 3-dT)

(Connection (Time -1) (From B) (To E) (Amp 9) (Width 6) (Delay 2+dT)

(Connection (Time -1) (From C) (To E) (Amp -4) (Width 5) (Delay 5-dT)

(Connection (Time -1) (From D) (To E) (Amp 2) (Width 4+2dT) (Delay 6-dT)

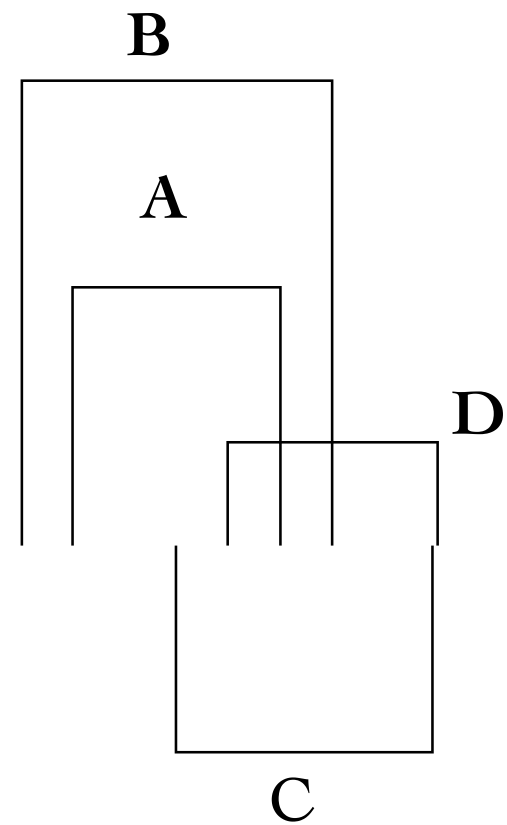
(Fire (Time 0) (Name A))

(Fire (Time 0) (Name B))

(Fire (Time 0) (Name C))

(Fire (Time 0) (Name D))

Time	Sum of Input to E
2	0
3	14
5	10
10	2

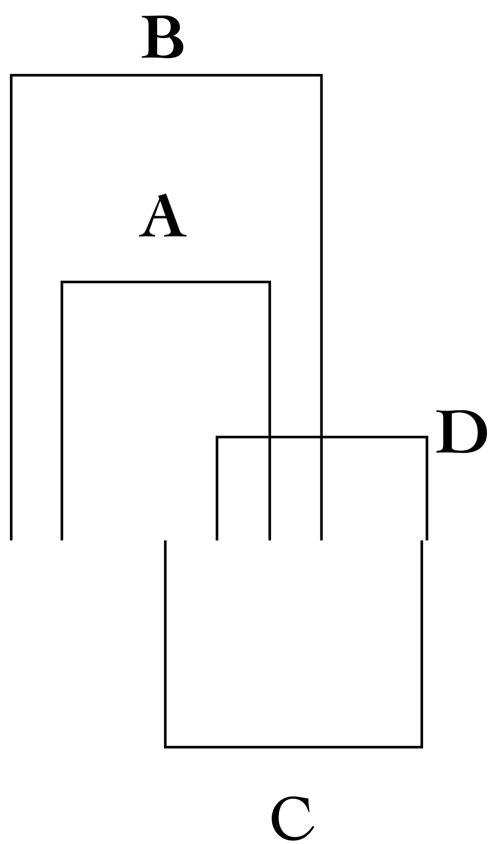


AEM computation - Element **E** Fires

(Pulse (Name A) (Window 3-dT 7+dT) (Amp 5))
(Pulse (Name B) (Window 2+dT 8+dT) (Amp 9))
(Pulse (Name C) (Window 5-dT 10-dT) (Amp -4))
(Pulse (Name D) (Window 6-dT 10+dT) (Amp 2))

(Element (Time 0) (Name E) (Threshold 9) (Refractory 4) (Last 0))

Element E fires at time 4.



Time	Sum of Input to E
4	14
8	7
9	-2
12	0

Element, Connection & Fire Commands

(Element (Time 0) (Name E) (Threshold 9) (Refractory 4) (Last 0))

(Connection (Time -1)(From A) (To E) (Amp 5) (Width 4+2dT) (Delay 3-dT))

Input Active elements: (Fire (Time 0) (Name A))

Keyword dT

- **dT** is an infinitesimal amount of time. **dT** > 0 and **dT** is less than every positive rational. If $m < n$, then m**dT** < n**dT**.

- **dT** helps with concurrency. 3-2**dT** < 3-**dT** < 3 < 3+**dT** < 3+2**dT** For every integer $m > 0$, 2 < 3-m**dT** and 3+m**dT** < 4.

- $st(k + m$**dT**$)$ = k is called the standard part of extended integer $k + m$**dT**.

Keyword clock

- **clock** evaluates to an integer which is the standard part of the current time of the active element machine clock.

Meta command - Dynamic Active Element Machine

(**Meta** (**Name E**) (**Window** b e) (**C** (**Args clock**)))

- If active element **E** fires at time s in window [b, e] where $b \leq s \leq e$ then command (**C clock**) executes at time s.
- If there is no window specified, then if **E** fires at any time s, then **(C s)** executes at time s. (No restrictions on time s when **E** fires.)

Meta Command and Randomness

- At each unit of time 0, 1, 2, . . ., input element **I** fires or does not fire based on a random bit generator.

```
(Program C  (Args  t)
   (Connection (Time t) (From I) (To t) (Amp  2) (Width 1) (Delay 1) )
   (Connection (Time t+1+dT) (From I) (To t) (Amp  0) )
   (Connection (Time t) (From t) (To t) (Amp  2) (Width 1) (Delay 1) )
)

(Element (Time clock) (Name clock)  (Threshold 1) (Refractory 1) (Last -1))
(Meta (Name I)  (C  (Args clock)) )
```

The random bits are **1**, **0**, 1, . . . At time 0, **I** fires. At time 1, **I** doesn't fires. At time **2, I** fires.

Meta & Randomness Execution Time **0**

(Program C (Args t)
(Connection (Time t) (From I) (To t) (Amp 2) (Width 1) (Delay 1))
(Connection (Time t+1+dT) (From I) (To t) (Amp 0))
(Connection (Time t) (From t) (To t) (Amp 2) (Width 1) (Delay 1))
)
(Element (Time clock) (Name clock) (Threshold 1) (Refractory 1) (Last -1))
(Meta (Name I) (C (Args clock)))

At time **0, I** fires.

(Element (Time 0) (Name 0) (Threshold 1) (Refractory 1) (Last -1))

(C (Args 0)) executes because **I** fired at time **0.**

The execution of **(C (Args 0))** causes three commands to execute:

(Connection (Time 0) (From I) (To 0) (Amp 2) (Width 1) (Delay 1))
(Connection (Time 1+dT) (From I) (To 0) (Amp 0))
(Connection (Time 0) (From 0) (To 0) (Amp 2) (Width 1) (Delay 1))

Element **0** continues to fires repeatedly at times 1, 2, 3, . . .

Explanation of Execution at Time **0**

At time **0, I** fires.
(Element (Time 0) (Name 0) (Threshold 1) (Refractory 1) (Last -1))

Because of
(Connection (Time 0) (From I) (To 0) (Amp 2) (Width 1) (Delay 1))
element **0** receives a pulse with amplitude **2** at time **1.** Since element **0**'s threshold is **1,** element **0** fires at time **1.**

The second command
(**Connection** (**Time** 1+dT) (**From** I) (**To** 0) (**Amp** 0))
prevents **I** firing at a time later than 0 from interfering with element **0**'s firing state.

The third command
(**Connection** (**Time 0**) (**From 0**) (**To 0**) (**Amp 2**) (**Width 1**) (**Delay** 1))
creates a connection from element **0** to itself so element **0** continues to fire indefinitely at times 2, 3, . . .

Meta & Randomness Execution Time 1

```
(Program C  (Args  t)
   (Connection (Time t) (From I) (To t) (Amp  2) (Width 1) (Delay 1) )
   (Connection (Time t+1+dT) (From I) (To t) (Amp  0) )
   (Connection (Time t) (From t) (To t) (Amp  2) (Width 1) (Delay 1) )
)
(Element (Time clock) (Name clock)  (Threshold 1) (Refractory 1) (Last -1))
(Meta (Name I)  (C  (Args clock)) )
```

At time 1, I doesn't fire.

(Element (Time 1) (Name 1) (Threshold 1) (Refractory 1) (Last -1))

No connection is established from I to element 1, so element 1 never fires.

Meta & Randomness Execution Time **2**

(Program C (Args t)
(Connection (Time t) (From I) (To t) (Amp 2) (Width 1) (Delay 1))
(Connection (Time t+1+dT) (From I) (To t) (Amp 0))
(Connection (Time t) (From t) (To t) (Amp 2) (Width 1) (Delay 1))
)
(Element (Time clock) (Name clock) (Threshold 1) (Refractory 1) (Last -1))
(Meta (Name I) (C (Args clock)))

At time **2, I** fires.

(Element (Time 2) (Name 2) (Threshold 1) (Refractory 1) (Last -1))

(C (Args 2)) executes because **I** fired at time **2.**

The execution of **(C (Args 2))** causes three commands to execute:

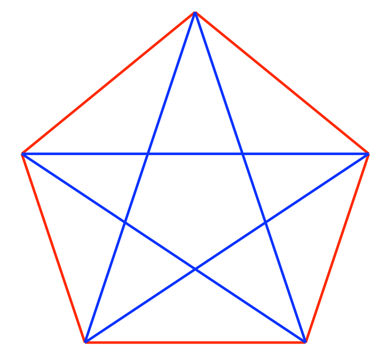
(Connection (Time 2) (From I) (To 2) (Amp 2) (Width 1) (Delay 1))
(Connection (Time 3+dT) (From I) (To 2) (Amp 0))
(Connection (Time 2) (From 2) (To 2) (Amp 2) (Width 1) (Delay 1))

Element **2** continues to fires repeatedly at times 3, 4, 5, . . .

Computing Ramsey Numbers

The Ramsey number $\mathbf{r(j, l)}$ denotes the least integer $\mathbf{n}$ such that if the edges of the complete graph $\mathbf{K_n}$ are $\mathbf{2}$-colored with colors **red** and **blue**, then there always exists a complete subgraph $\mathbf{K_j}$ containing only **red** edges or there exists a complete subgraph $\mathbf{K_k}$ containing only **blue** edges.

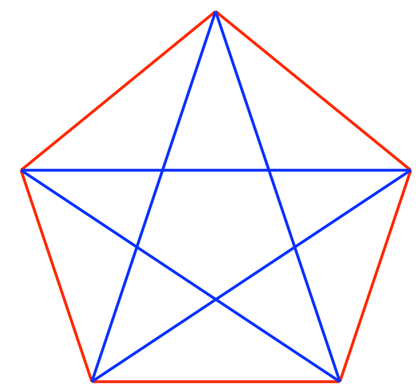
Determining $\mathbf{r(j, k)}$ is an NP-hard problem. $\mathbf{r(5, 5)}$ is unknown.



Paul Erdos asks us to imagine an alien force, vastly more powerful than us, landing on Earth and demanding the value of $\mathbf{r(5, 5)}$ or they will destroy our planet. In this case, Erdos claims that we should marshal all our computers and all our mathematicians and attempt to find the value. But suppose instead that they ask for $\mathbf{r(6, 6)}$. For $\mathbf{r(6, 6)}$, Erdos believes that we should attempt to destroy the aliens.

AEM program that verifies $\mathbf{r(3, 3)} > \mathbf{5}$

Edges $\mathbf{E} = \{\{1, 2\},\{1, 3\},\{1, 4\},\{1, 5\},\{2, 3\},\{2, 4\},\{2, 5\},\{3, 4\}, \{3, 5\}, \{4, 5\}\}$
Triangles $\mathbf{T} = \{ \{1, 2, 3\}, \{1, 2, 4\}, \{1, 2, 5\}, \{1, 3, 4\}, \{1, 3, 5\}, \{1, 4, 5\}, \{2, 3, 4\}, \{2, 3, 5\}, \{2, 4, 5\}, \{3, 4, 5\}\}$



Red edges = $\{ \{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{1, 5\}\}$
Blue edges = $\{ \{1, 3\}, \{1, 4\}, \{2, 4\}, \{2, 5\}, \{3, 5\}\}$

Indices on symbols **B** and **R** denote active elements that correspond to the $\mathbf{K_5}$ graph geometry.

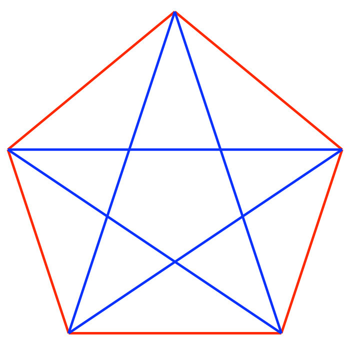
1. Elements representing **red** and **blue** edges are established.

(Element (Time 0) (Name R_12) (Threshold 1) (Refractory 1) (Last -1))
(Element (Time 0) (Name R_23) (Threshold 1) (Refractory 1) (Last -1))
(Element (Time 0) (Name R_34) (Threshold 1) (Refractory 1) (Last -1))
(Element (Time 0) (Name R_45) (Threshold 1) (Refractory 1) (Last -1))
(Element (Time 0) (Name R_15) (Threshold 1) (Refractory 1) (Last -1))

AEM program that verifies **r**(3, 3) > 5

(Element (Time 0) (Name B_13) (Threshold 1) (Refractory 1) (Last -1))
(Element (Time 0) (Name B_14) (Threshold 1) (Refractory 1) (Last -1))
(Element (Time 0) (Name B_24) (Threshold 1) (Refractory 1) (Last -1))
(Element (Time 0) (Name B_25) (Threshold 1) (Refractory 1) (Last -1))
(Element (Time 0) (Name B_35) (Threshold 1) (Refractory 1) (Last -1))

2. Fire element **R_jk** if edge {j, k} is **red**.
(Fire (Time 0) (Name R_12)
(Fire (Time 0) (Name R_23)
(Fire (Time 0) (Name R_34)
(Fire (Time 0) (Name R_45)
(Fire (Time 0) (Name R_15)



3. Fire element **B_jk** if edge {j, k} is **blue**.
(Fire (Time 0) (Name B_13)
(Fire (Time 0) (Name B_14)
(Fire (Time 0) (Name B_24)
(Fire (Time 0) (Name B_25)
(Fire (Time 0) (Name B_35)

AEM program that verifies $r(3, 3) > 5$

4. Meta command cause these elements to keep firing once they have fired.

(Meta (Name 0) (Name R_jk) (Window 0 1)
(Connection (Time 0) (From R_jk) (To R_jk) (Amp 2) (Width 1) (Delay 1)))

(Meta (Name 0) (Name B_jk) (Window 0 1)
(Connection (Time 0) (From B_jk) (To B_jk) (Amp 2) (Width 1) (Delay 1)))

5. For each **{i, j, k}**, determine if a **blue** triangle exists on vertices **{i, j, k}**.

(Connection (Time 0) (From B_ij) (To B_ijk) (Amp 2) (Width 1) (Delay 1)))
(Connection (Time 0) (From B_jk) (To B_ijk) (Amp 2) (Width 1) (Delay 1)))
(Connection (Time 0) (From B_ik) (To B_ijk) (Amp 2) (Width 1) (Delay 1)))

6. For each **{i, j, k}**, determine if a **red** triangle exists on vertices **{i, j, k}**.

(Connection (Time 0) (From R_ij) (To R_ijk) (Amp 2) (Width 1) (Delay 1)))
(Connection (Time 0) (From R_jk) (To R_ijk) (Amp 2) (Width 1) (Delay 1)))
(Connection (Time 0) (From R_ik) (To R_ijk) (Amp 2) (Width 1) (Delay 1)))

AEM program that verifies $\mathbf{r(3, 3)} > \mathbf{5}$

7. For each vertex set **{i, j, k}** in **T** create the following elements.
(Element (Time 0) (Name R_ijk) (Threshold 5) (Refractory 1) (Last -1))
(Element (Time 0) (Name B_ijk) (Threshold 5) (Refractory 1) (Last -1))

A. **R_ijk** only fires when all three elements **R_ij**, **R_jk**, **R_ik** fired one unit of time ago. **B_ijk** only fires when all three elements **B_ij**, **B_jk**, **B_ik** fired one time ago.

B. This AEM computation takes 4 time steps to determine that $\mathbf{r(3, 3)} > \mathbf{5}$

C. This AEM computation uses $2|\mathbf{T}| = 30$ active elements and uses $3|\mathbf{T}| + 3|\mathbf{T}| + |\mathbf{E}| = 70$ connections.

D. Building an AEM program in a similar way on $\mathbf{K_6}$, this AEM determines that $\mathbf{r(3, 3)} = 6$ in 5 time steps. The brute force computation uses $2^{|\mathbf{E}|}(|\mathbf{E}| + 2|\mathbf{T}|) + 1$ elements and $2^{|\mathbf{E}|}(3|\mathbf{T}| + 3|\mathbf{T}| + |\mathbf{E}| + 1)$ connections where $|\mathbf{E}| = 15$ and $|\mathbf{T}| = 20$.

Useful Properties of the AEM

1. **Parallel Computation & Algorithms.**
   - All active elements compute simultaneously.

2. **Avoiding Race Conditions.**
   - Time in the commands helps with coordination. **dT**

3. **Machine can change its rules while executing.**
   - Meta command and time.

4. **Meta command: program complexity can increase with time.**

What is interesting?

1. **Interpretations.**

   Meta Command enables simultaneous, firing representations to dynamically change. (Unlike the register machine.)

2. **Ramsey Number computation is parallelizable.**

   What about other computationally difficult problems?

   http://www.aemea.org/msf/periodic_TM2.pdf

3. **Adding randomness enables useful behavior.**

   How can this be used effectively in machine learning?

What is next?

1. **Machine learning and evolution.**
   - Randomness, the Meta command and Firing Interpretations.
2. **AEM algorithms and mathematical analysis.**
   - Alternatives to von Neumann machine algorithms: Graph algorithms, Dynamic Programming, Distributed networks, Race Conditions, . . .
   - Euler's method on differential equations.
3. **Software implementation on register machine hardware.**
   - Language tools and autonomous systems.
4. **Hardware implementation with AEM friendly architecture.**
   - FPGA, Silicon, Optical, . . .

1. **Pattern recognition.**
    - Object recognition for auto safety.
    - Object recognition for manufacturing quality control & safety.

2. **Weather Forecasting.**

3. **Secure Computation.**