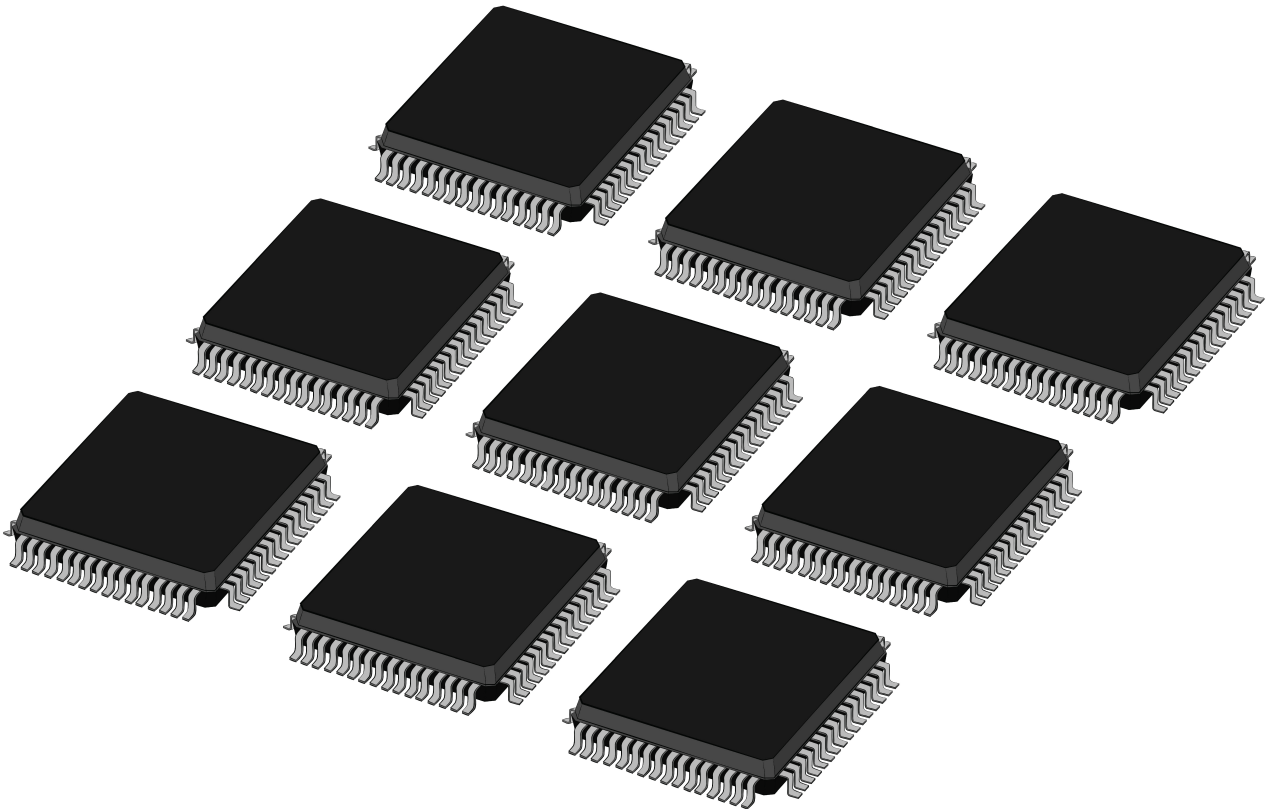


Traditional Computing Theory

QueenField



Paco Reina Campo

Abstract

Traditional Computing Theory.

Contents

1	Mechanics	5
1.1	Newtonian Mechanics	5
1.1.1	First Newton Law	5
1.1.2	Second Newton Law	5
1.1.3	Third Newton Law	6
1.2	Lagrangian Mechanics	6
1.3	Hamiltonian Mechanics	6
2	Information	7
2.1	Bit	7
2.2	Logic Gate	7
2.2.1	YES/NOT Gate	7
2.2.2	AND/NAND Gate	8
2.2.3	OR/NOR Gate	8
2.2.4	XOR/XNOR Gate	8
2.3	Combinational Logic	8
2.3.1	Arithmetic Circuits	8
2.3.2	Logic Circuits	8
2.4	Finite State Machine	8
2.5	Pushdown Automaton	9
3	Neural Network	11
3.1	Feedforward Neural Network	11
3.2	Long Short Term Memory Neural Network	11
3.3	Transformer Neural Network	12
4	Turing Machine	14
4.1	Neural Turing Machine	15
4.1.1	Feedforward Neural Turing Machine	17
4.1.2	LSTM Neural Turing Machine	17
4.1.3	Transformer Neural Turing Machine	17
4.2	Differentiable Neural Computer	17
4.2.1	Feedforward Differentiable Neural Computer	20
4.2.2	LSTM Differentiable Neural Computer	21
4.2.3	Transformer Differentiable Neural Computer	21
5	Computer Architecture	22
5.1	von Neumann Architecture	22
5.1.1	RISC-V	22
5.1.2	MSP430	22
5.2	Harvard Architecture	22
5.2.1	RISC-V	22
5.2.2	OpenRISC	22
6	Advanced Computer Architecture	24
6.1	Processing Unit	24
6.1.1	SISD	24
6.1.2	SIMD	24
6.1.3	MISD	24

6.1.4	MIMD	24
6.2	System on Chip	24
6.2.1	Bus on Chip	25
6.2.2	Network on Chip	25
6.3	Multi-Processor System on Chip	25

List of Tables

2.1	Finite State Machine Definitions	9
2.2	Pushdown Automaton Definitions	9
4.1	Turing Machine Definitions	14

List of Figures

2.1 Automata Theory 7

Chapter 1

Mechanics

1.1 Newtonian Mechanics

1.1.1 First Newton Law

$$\vec{F} = \vec{0} \rightarrow \frac{d\vec{p}}{dt} = \vec{0}$$

1.1.2 Second Newton Law

$$\vec{p} = m\vec{v}$$

$$\vec{F} = \frac{d\vec{p}}{dt}$$

1.1.3 Third Newton Law

.....

$$\vec{p} = \vec{p}_1 + \vec{p}_2$$

$$\frac{d\vec{p}}{dt} = \frac{d\vec{p}_1}{dt} + \frac{d\vec{p}_2}{dt}$$

.....

1.2 Lagrangian Mechanics

.....

$$\frac{d}{dt} \left(\frac{\partial L(q_i(t), \dot{q}_i(t), t)}{\partial \dot{q}_i} \right) - \frac{\partial L(q_i(t), \dot{q}_i(t), t)}{\partial q_i} = 0$$

.....

1.3 Hamiltonian Mechanics

.....

$$H(\mathbf{q}, \mathbf{p}, t) = \sum_i \dot{q}_i p_i - L(q_j, \dot{q}_j, t)$$

.....

Chapter 2

Information

.....

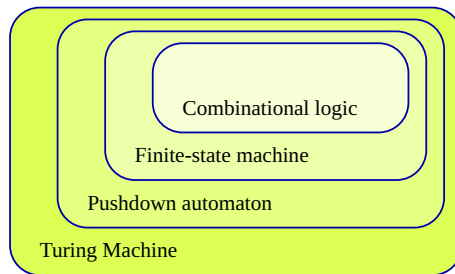


Figure 2.1: Automata Theory

.....

2.1 Bit

.....

2.2 Logic Gate

.....

2.2.1 YES/NOT Gate

.....

2.2.2 AND/NAND Gate

.. .. .
.. .. .
.. .. .
.. .. .

2.2.3 OR/NOR Gate

.. .. .
.. .. .
.. .. .
.. .. .

2.2.4 XOR/XNOR Gate

.. .. .
.. .. .
.. .. .
.. .. .

2.3 Combinational Logic

.. .. .
.. .. .
.. .. .
.. .. .

2.3.1 Arithmetic Circuits

.. .. .
.. .. .
.. .. .
.. .. .

2.3.2 Logic Circuits

.. .. .
.. .. .
.. .. .
.. .. .

2.4 Finite State Machine

.. .. .
.. .. .
.. .. .
.. .. .

$$T = \{Q, \Sigma, \delta, q_0, F\}$$

.. .. .
.. .. .
.. .. .
.. .. .

$$Q \subseteq H$$

.. .. .
.. .. .

$$\delta : Q \times \Sigma \otimes Q \rightarrow Q$$

Table 2.1: Finite State Machine Definitions

Element	Definitions
Q	Finite Non-Empty Set of States
Σ	Input Alphabet. A Finite Non-Empty Set of Symbols
δ	State Transition Function
$q_0 \in Q$	Initial State of Set of States
F	Set of Final States. A (Possibly Empty) Subset of Set of States

2.5 Pushdown Automaton

$$T = \{Q, \Sigma, \Gamma, \delta, q_0, \gamma_0, F\}$$

$$Q \subseteq H$$

$$\delta : \Sigma \times Q \otimes \Gamma \rightarrow \Sigma \times Q \otimes \Gamma \times \{L, R\}$$

Table 2.2: Pushdown Automaton Definitions

Element	Definitions
Q	Finite Non-Empty Set of States
Σ	Input Alphabet. A Finite Non-Empty Set of Symbols

Element	Definitions
Γ	Stack Alphabet. A Finite Non-Empty Set of Symbols
δ	State Transition Function
$q_0 \in Q$	Initial State of Set of States
$\gamma_0 \in \Gamma$	Initial Symbol of Stack Alphabet
F	Set of Final States. A (Possibly Empty) Subset of Set of States

.....
.....
.....
.....

Chapter 3

Neural Network

3.1 Feedforward Neural Network

$$h_t = \sigma_g(W_h \cdot x_t + U_h \cdot h_{t-1} + b_h)$$

$$y_t = \sigma_g(W_y \cdot h_t + b_y)$$

$$h_t = \sigma_g(W_h \star x_t + U_h \star h_{t-1} + b_h)$$

$$y_t = \sigma_g(W_y \star h_t + b_y)$$

3.2 Long Short Term Memory Neural Network

$$a_t = \sigma_g(W_a \cdot x_t + U_a \cdot h_{t-1} + b_a)$$

$$f_t = \sigma_g(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f)$$

$$i_t = \sigma_g(W_i \cdot x_t + U_i \cdot h_{t-1} + b_i)$$

$$o_t = \sigma_g(W_o \cdot x_t + U_o \cdot h_{t-1} + b_o)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ a_t$$

$$h_t = o_t \circ \sigma_g(c_t)$$

$$a_t = \sigma_g(W_a \star x_t + U_a \star h_{t-1} + b_a)$$

$$f_t = \sigma_g(W_f \star x_t + U_f \star h_{t-1} + b_f)$$

$$i_t = \sigma_g(W_i \star x_t + U_i \star h_{t-1} + b_i)$$

$$o_t = \sigma_g(W_o \star x_t + U_o \star h_{t-1} + b_o)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ a_t$$

$$h_t = o_t \circ \sigma_g(c_t)$$

3.3 Transformer Neural Network

$$\text{attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V$$

- Query vector

$$q_i = x_i W_Q$$

- Key Vector

$$k_i = x_i W_K$$

[illegible]

- Value Vector

[illegible]

$$v_i = x_i W_V$$

[illegible]

Chapter 4

Turing Machine

.....

$$T = \{Q, \Sigma, \Gamma, \delta, q_0, \gamma_0, F\}$$

.....

$$Q \subseteq H$$

.....

$$\delta : \Sigma \times Q \otimes \Gamma \rightarrow \Sigma \times Q \otimes \Gamma \times \{L, R\}$$

.....

Table 4.1: Turing Machine Definitions

Element	Definitions
Q	Finite Non-Empty Set of States
Σ	Input Alphabet. A Finite Non-Empty Set of Symbols
Γ	Stack Alphabet. A Finite Non-Empty Set of Symbols
δ	State Transition Function
$q_0 \in Q$	Initial State of Set of States
$\gamma_0 \in \Gamma$	Initial Symbol of Stack Alphabet
F	Set of Final States. A (Possibly Empty) Subset of Set of States

.....

4.1 Neural Turing Machine

- Definitions

$$\mathcal{D}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|}$$

- Reading

$$\sum_{i=0}^{M-1} w_t(i) = 1$$

$$0 \leq w_t(i) \leq 1$$

$$\mathbf{r}_t \leftarrow \sum_{i=0}^{M-1} w_t(i) \mathbf{M}_t(i)$$

- Writing

$$\tilde{\mathbf{M}}_t(i) \leftarrow \mathbf{M}_{t-1}(i) [1 - w_t(i) \mathbf{e}_t]$$

$$\mathbf{M}_t(i) \leftarrow \tilde{\mathbf{M}}_t(i) + w_t(i) \mathbf{a}_t$$

- Addressing

$$w_t^c(i) \leftarrow \frac{\exp\left(\beta_t \mathcal{D}[\mathbf{k}_t, \mathbf{M}_t(i)]\right)}{\sum_{j=0}^{N-1} \exp\left(\beta_t \mathcal{D}[\mathbf{k}_t, \mathbf{M}_t(j)]\right)}$$

$$\mathbf{w}_t^g \leftarrow g_t \mathbf{w}_t^c + (1 - g_t) \mathbf{w}_{t-1}$$

$$\tilde{w}_t(i) \leftarrow \sum_{j=0}^{N-1} w_t^g(j) s_t(i - j)$$

$$w_t(i) \leftarrow \frac{\tilde{w}_t(i)^{\gamma_t}}{\sum_{j=0}^{N-1} \tilde{w}_t(j)^{\gamma_t}}$$

- Interfaces

$$\xi_t = W_\xi[h_t^1; \dots; h_t^L] = [\mathbf{k}_t^w; \hat{\beta}_t^w; \hat{\mathbf{e}}_t; \mathbf{v}_t; \hat{g}_t^a; \hat{g}_t^w]$$

$$\rho_t = W_\rho[h_t^1, \dots; h_t^L] = [\mathbf{k}_t^{r,1}, \dots; \mathbf{k}_t^{r,R}; \hat{\beta}_t^{r,1}, \dots; \hat{\beta}_t^{r,R}; \hat{f}_t^1, \dots; \hat{f}_t^R; \hat{\pi}_t^1, \dots; \hat{\pi}_t^R]$$

- Output Vector

$$\mathbf{y}_t = W_y \mathbf{h}_t + W_r^i \mathbf{r}_t^i$$

4.1.1 Feedforward Neural Turing Machine

4.1.2 LSTM Neural Turing Machine

4.1.3 Transformer Neural Turing Machine

4.2 Differentiable Neural Computer

- Definitions

$$\mathcal{D}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|}$$

```

.. .....
.....
.....
.....

```

$$\mathcal{C}(M, \mathbf{k}, \beta)[i] = \frac{\exp\{\mathcal{D}(\mathbf{k}, M[i, \cdot])\beta\}}{\sum_j \exp\{\mathcal{D}(\mathbf{k}, M[j, \cdot])\beta\}}$$

```

.. .....
.....
.....
.....

```

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

```

.. .....
.....
.....
.....

```

$$\text{oneplus}(x) = 1 + \log(1 + e^x)$$

```

.. .....
.....
.....
.....

```

$$\text{softmax}(\mathbf{x})_j = \frac{e^{x_j}}{\sum_{k=1}^K e^{x_k}}$$

```

.. .....
.....
.....
.....

```

- Addressing

```

.. .....
.....
.....
.....

```

$$M_t = M_{t-1} \circ (E - \mathbf{w}_t^w \mathbf{e}_t^\top) + \mathbf{w}_t^w \mathbf{v}_t^\top$$

```

.. .....
.....
.....
.....

```

$$\mathbf{u}_t = (\mathbf{u}_{t-1} + \mathbf{w}_{t-1}^w - \mathbf{u}_{t-1} \circ \mathbf{w}_{t-1}^w) \circ \psi_t$$

```

.. .....
.....
.....
.....

```

$$\mathbf{p}_t = \left(1 - \sum_i \mathbf{w}_t^{wv}[i]\right) \mathbf{p}_{t-1} + \mathbf{w}_t^{wv}$$

$$L_t = (\mathbf{1} - \mathbf{I}) \left[(1 - \mathbf{w}_t^w[i] - \mathbf{w}_t^j) L_{t-1}[i, j] + \mathbf{w}_t^w[i] \mathbf{p}_{t-1}^j \right]$$

$$\mathbf{w}_t^w = g_t^w [g_t^a \mathbf{a}_t + (1 - g_t^a) \mathbf{c}_t^w]$$

$$\mathbf{w}_t^{r,i} = \pi_t^i[1] \mathbf{b}_t^i + \pi_t^i[2] \mathbf{c}_t^{r,i} + \pi_t^i[3] \mathbf{f}_t^i$$

$$\mathbf{r}_t^i = M_t^\top \mathbf{w}_t^{r,i}$$

$$\mathbf{a}_t[\phi_t[j]] = (1 - \mathbf{u}_t[\phi_t[j]]) \prod_{i=1}^{j-1} \mathbf{u}_t[\phi_t[i]]$$

$$\mathbf{c}_t^w = \mathcal{C}(M_{t-1}, \mathbf{k}_t^w, \beta_t^w)$$

$$\mathbf{c}_t^{r,i} = \mathcal{C}(M_{t-1}, \mathbf{k}_t^{r,i}, \beta_t^{r,i})$$

$$\mathbf{f}_t^i = L_t \mathbf{w}_{t-1}^{r,i}$$

$$\mathbf{b}_t^i = L_t^\top \mathbf{w}_{t-1}^{r,i}$$

$$\psi_t = \prod_{i=1}^R (1 - f_t^i \mathbf{w}_{t-1}^{r,i})$$

- Interfaces

$$\xi_t = W_\xi[h_t^1; \dots; h_t^L] = [\mathbf{k}_t^w; \hat{\beta}_t^w; \hat{\mathbf{e}}_t; \mathbf{v}_t; \hat{g}_t^a; \hat{g}_t^w]$$

$$\rho_t = W_\rho[h_t^1; \dots; h_t^L] = [\mathbf{k}_t^{r,1}; \dots; \mathbf{k}_t^{r,R}; \hat{\beta}_t^{r,1}; \dots; \hat{\beta}_t^{r,R}; \hat{f}_t^1; \dots; \hat{f}_t^R; \hat{\pi}_t^1; \dots; \hat{\pi}_t^R]$$

- Output Vector

$$\mathbf{y}_t = W_y \mathbf{h}_t + W_r^i \mathbf{r}_t^i$$

4.2.1 Feedforward Differentiable Neural Computer

4.2.2 LSTM Differentiable Neural Computer

.....
.....
.....
.....

4.2.3 Transformer Differentiable Neural Computer

.....
.....
.....
.....

Chapter 5

Computer Architecture

..
.....
.....
.....

5.1 von Neumann Architecture

..
.....
.....
.....

5.1.1 RISC-V

..
.....
.....
.....

5.1.2 MSP430

..
.....
.....
.....

5.2 Harvard Architecture

..
.....
.....
.....

5.2.1 RISC-V

..
.....
.....
.....

5.2.2 OpenRISC

..
.....

*** ***** ** ***** **** ***** * ***** ** ** ***** ***** ** ***** ***** ***** ***** ***** ***** ***** ***** ***** ***** *****
*** ***** ** *****

Chapter 6

Advanced Computer Architecture

..
.....
.....
.....

6.1 Processing Unit

..
.....
.....
.....

6.1.1 SISD

..
.....
.....
.....

6.1.2 SIMD

..
.....
.....
.....

6.1.3 MISD

..
.....
.....
.....

6.1.4 MIMD

..
.....
.....
.....

6.2 System on Chip

..
.....

... ..
... ..

6.2.1 Bus on Chip

.. ..
... ..
... ..
... ..

6.2.2 Network on Chip

.. ..
... ..
... ..
... ..

6.3 Multi-Processor System on Chip

.. ..
... ..
... ..
... ..