Pattern Classification and NNet applications with memristive crossbar circuits

Fabien ALIBART
D. Strukov’s group, ECE-UCSB
Now at IEMN-CNRS, France
Outline

• Introduction: Neural Network with memristive devices
• Engineering and switching dynamics
• High precision analog tuning
• Nnet Circuits
  • MAC operation
  • Hopfield A/D converter
  • Pattern classifier
• Conclusion
Introduction: Why ANNet

Saturation of clock frequency + Energy consumption

Increase of Fault (nanoscale engineering)

New needs for computing Recognition, Mining, Synthesis (Intel)

Von Neumann bottleneck

SEMICONDUCTOR TECHNOLOGY CHALLENGES

Shift toward a new paradigm for computation

BIO-INSPIRED COMPUTING to match the brain performances (low power consumption, fault tolerant, performances for RMS)
Introduction: NNET basic principle

Biological neural network

Bio-inspired system: in the brain, we learn by creating (weighting) synaptic connections between neurons from different experiences. After, we can react to unknown situations which are similar to the learning ones.

The memory is in the processing unit (Direct solution to Von Neumann bottleneck!)
Introduction: NNET roadmap

$10^{11}$ neurons / $10^{15}$ synapses
Memristive TiO2 devices

- Non-linear I-V
- Analog non-volatile memory (any state between $R_{on}$ and $R_{off}$ is accessible and stable)
- “smooth” RESET transition
- “sharp” SET transition
- “plateau” of non disturbing voltage pulses

Main challenge: how to deal with dispersion
Engineering improve yield and decrease dispersion (better control of forming)
Switching

3 parameter for switching:
- Voltage
- Time
- Initial state
Switching

3 parameters for switching:
- Voltage
- Time
- Initial state

But still a large dispersion!!
Switching

Restricted analysis (2 parameters):
- Voltage
- Initial state

Partial conclusion: engineering is not enough to deal with dispersion
Algorithm

Objective: tune the device to a desired state by pulses of voltage

The perfect picture:

- Switching dynamic
- Statistics

Device modeling

Which pulse for a given state?

Tuning of the device/evaluation

Feedback control

A more simple picture: sweep V or t

DILEMA

Using Voltage as the parameter: potentially fast but low precision
Using t as the parameter: good accuracy but may required infinite time
In addition, we need to deal with dispersion!
Algorithm

F. Alibart et al. Nanotechnology, 23 075201, 2012

Start
- inputs: desired state \( I_{\text{desired}} \), desired accuracy \( A_{\text{desired}} \)
- initialize: write voltage to small non-disturbing value \( V_{\text{WRITE}} = 200 \text{ mV} \), voltage step \( V_{\text{STEP}} = 10 \text{ mV} \)

Read
- (apply \( V_{\text{READ}} = 200 \text{ mV} \) and read current \( I_{\text{current}} \))

Processing
- if \( I_{\text{desired}} - I_{\text{current}} < A_{\text{desired}} \)
  - yes
  - Finish
- if \( V_{\text{WRITE}} = V_{\text{READ}} \) and sign \( I_{\text{oldsign}} = \text{sign} \)
  - check for overshoot and set the sign of increment, i.e.
    \[ \text{sign} = \text{sign} \left( I_{\text{current}} - I_{\text{desired}} \right) / \left( V_{\text{WRITE}} - V_{\text{READ}} \right) \]
    if \( V_{\text{WRITE}} 
eq V_{\text{READ}} \)
  - no

Write
- apply pulse \( V_{\text{WRITE}} \)

Increase Weight
- (apply \( V_{\text{WRITE}} = V_{\text{WRITE}} + \text{sign} \times V_{\text{STEP}} \))

Decrease Weight
- (apply \( V_{\text{WRITE}} = V_{\text{WRITE}} - \text{sign} \times V_{\text{STEP}} \))

Stand-by (Read only)
- (apply \( V_{\text{READ}} = 0 \))

Resistance state

Current @ -200mV (A)

Time (s)

60\mu A
30\mu A
15\mu A
Algorithm

- 1% accuracy equivalent to 8-bit
- Intermediate state are non-volatile
- Can be improved with better modeling (more elaborated feedback)
- Accuracy is limited by noise (evidence of RTS)

F. Alibart et al. Nanotechnology, 23 075201, 2012
Circuit: MAC operation

- $R_1 = 3333\, \text{ohm}$
- $R_2 = 3333\, \text{ohm}$
- $R_1 = 3333\, \text{ohm}$
- $R_2 = 6666\, \text{ohm}$
- $R_1 = 6666\, \text{ohm}$
- $R_2 = 13333\, \text{ohm}$
- $R_1 = 13333\, \text{ohm}$
- $R_2 = 13333\, \text{ohm}$
- $R_1 = 13333\, \text{ohm}$

F. Alibart et al. Nanotechnology, 23 075201, 2012
Circuit: Hopfield Network

Basically an associative memory:
Energy function

\[ E = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} T_{ij} V_i V_j - \sum_{i=1}^{N} V_i I_i. \]

Suitable also for optimization problem, speech recognition...

An example: Analog (input) to digital (output) conversion
Circuit: Hopfield Network

TiO2 memristive devices

Analog input: Amplitude: 1.75V, offset: 0.85V, f=1Hz

Variable Resistors:
- R1 = 210k
- R2 = 45.8k
- R3 = 10.72k
- R4 = 2.39k

Memristors:
- 0.95uA@0.2V
- 4.37uA@0.2V
- 18.66uA@0.2V
- 83.7uA@0.2V

Digital output

Analog input

Voltage (V)

Time (s)
The perceptron task is to realize classification:
Considering a set of data \( \{(V_1, \ldots, V_n)\} \), if this ensemble of data is composed of two separable group A and B, it is possible to define a set of \((R_1, \ldots, R_n)\) that verifies:

\[
\sum V_i / R_i > \text{threshold} \quad \text{if} \quad (V_1 \ldots V_n) \text{ is } A \quad \text{(i.e. } \text{out=}1) \\
\sum V_i / R_i < \text{threshold} \quad \text{if} \quad (V_1 \ldots V_n) \text{ is } B \quad \text{(i.e. } \text{out=}0)
\]

The set of weight is determined by optimization procedure on “real” datas, or “training” datas (No analytical solution). This stage of programming is called “learning” (the system is trained to react properly by adjusting the weights). The performances are then evaluated on a testing set of datas (operation of the system).
Pattern classification

Seminal work
Bernard Widrow
Marcian Hoff

Fig. 4. Patterns for classification experiment.

Fig. 5. Adaptive-element performance curve.

<table>
<thead>
<tr>
<th>EXPERIMENT</th>
<th>PATTERNS ADAPTED</th>
<th>NUMBER OF ERRORS</th>
<th>MISADJUSTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>95, 79, 07, 60</td>
<td>25</td>
<td>M = 32.36% = 108%</td>
</tr>
<tr>
<td>2</td>
<td>70, 49, 32, 34</td>
<td>19</td>
<td>M = 17.95% = 58%</td>
</tr>
<tr>
<td>3</td>
<td>45, 12, 84, 63</td>
<td>20</td>
<td>M = 20.12% = 67%</td>
</tr>
<tr>
<td>4</td>
<td>07, 42, 45, 66</td>
<td>28</td>
<td>M = 35.12% = 133%</td>
</tr>
</tbody>
</table>

Fig. 8. Training with noisy patterns.

memistor
Pattern classification

Input set of patterns “X”
Desired output: \( d = +1 \)

Input set of patterns “T”
Desired output: \( d = -1 \)

Learning rule:

\[
\begin{pmatrix}
\sum_x \sum_y \\
\end{pmatrix} = \begin{pmatrix}
w_0^+ & w_1^+ & w_2^+ & w_3^+ & w_4^+ & w_5^+ & w_6^+ & w_7^+ & w_8^+ & w_9^+ \\
w_0^- & w_1^- & w_2^- & w_3^- & w_4^- & w_5^- & w_6^- & w_7^- & w_8^- & w_9^- \\
\end{pmatrix}
\]

memristors
Pattern classification: setup

Arbitrary waveform generator B1530

Current measurement B1530 (fast IV mode)

Ground (GNDU, Agilent)

Agilent B1500

Switching matrix (Agilent E5250A)

Wires implementing crossbar circuit

Chip packaged wire bonded memristive devices
Pattern classification: Ex situ

We want to use our “high precision algorithm” to tune the state of each device sequentially. The final weights are calculated on a software-based precursor network. Challenge: we need to import the weight into the Xbar array (XTALK).

Solution: half bias trick.

High precision algorithm

- $V_{\text{switch}}$
- $-V_{\text{switch}}$

Read pulse

Write pulse

Current @ 200mV

Vread

Number of pulses

Vread

Slight Xtalk

5-bit accuracy still available
Pattern classification: Ex situ

- 3-bit accuracy is enough for this classification task
- Can be improve by delta rule instead of perceptron rule
Pattern classification: In situ

- 4-step pulse sequence
- Training in parallel of all the devices

Input set of patterns “X”
Desired output: \(d = +1\)

Input set of patterns “T”
Desired output: \(d = -1\)

Learning rule:
\[
\Delta w_i^\pm = \pm \alpha (d^{(p)} - y^{(p)})
\]
Pattern classification: In situ

Figure 4

- 7 epochs
  - $V_{training} = 1V$
  - Number of pattern #

- 10 epochs
  - $V_{training} = 0.9V$

- Initial
- $I_+ - I_-$ (A)

- $V_{training} = 1V$
- Training events #

- $V_{training} = 0.9V$
- $\omega_{10}$
- $\omega_9$
- $\omega_8$
- $\omega_7$
- $\omega_6$
- $\omega_5$
- $\omega_4$
- $\omega_3$
- $\omega_2$
- $\omega_1$

(A @ 200mV)
Pattern classification: In situ

**A**

- **$V_{\text{TRAINING}}$**
  - INITIAL
  - $T \rightarrow -1$  $X \rightarrow +1$
  - $T \rightarrow 1$  $X \rightarrow -1$
  - $X \rightarrow 1$  $T \rightarrow +1$

- **$I_+ - I_-$** (A)

- **Number of Pattern #**
  - 0.9V
  - 1V
  - 1.1V

**B**

- **class inversion**

- **$\omega$**
  - $\omega_{10}$
  - $\omega_{9}$
  - $\omega_{8}$
  - $\omega_{7}$
  - $\omega_{6}$
  - $\omega_{5}$
  - $\omega_{4}$
  - $\omega_{3}$
  - $\omega_{2}$
  - $\omega_{1}$

- **$V_{\text{TRAINING}}$**
  - 0.9V
  - 1V
  - 1.1V

**RECONFIGURABLE**
conclusion

• Physics of the memristive devices to be continued (modeling will improve algorithms)
• Memristive Xbar is promising for larger scale NNET circuit implementation (forming free devices is required)
• Heterogeneous CMOS/Xbar is the next technological challenge (implementation of CMOL concept)
Acknowledgments

Open position:
• 1 Postdoc at UCSB
• 1 Postdoc at IEMN

Supervisor: Dmitri Strukov
ADC work: Ligang Gao, Farnood Merrikh-Bayat, Xinjie Guo
Pattern classification: Elham Zamanidoost
Material science: Brian Hoskins