SenseMaker

IST2001-34712

Neuro-IT Presentation

Alicante 2003
## Project Partners

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ISEL</td>
<td>University of Ulster (UK) - Intelligent Systems Engineering Laboratory</td>
</tr>
<tr>
<td>2</td>
<td>UNIC-CNRS</td>
<td>Unité de Neurosciences Intégratives et Computationnelles, Centre National de la Recherche Scientifique (France)</td>
</tr>
<tr>
<td>3</td>
<td>UHEI</td>
<td>Ruprecht-Karls-Universität Heidelberg (Germany), Kirchhoff-Institut für Physik</td>
</tr>
<tr>
<td>4</td>
<td>TCD</td>
<td>Trinity College Dublin (Ireland) Visual Cognition Group, Institute for Neuroscience, Department of Psychology</td>
</tr>
<tr>
<td>5</td>
<td>IXL-CNRS</td>
<td>ENSEIRB-CNRS Université Bordeaux 1 (France), IXL Laboratory</td>
</tr>
</tbody>
</table>
Project Overview
BIOLOGICAL PRINCIPLES

- SYSTEM LEVEL: BRAIN-LIKE ARCHITECTURE
- NEURAL PROCESSING LEVEL
- « NOISE » AS A COMPUTATIONAL PRINCIPLE
- CONSTRAINTS ON ARTIFICIAL NETWORK ARCHITECTURE
- LOW-LEVEL IMPLEMENTATION: THE TWO-RING PROBLEM
BIOLOGICAL PRINCIPLES

- SYSTEM LEVEL: BRAIN-LIKE ARCHITECTURE
- NEURAL PROCESSING LEVEL
- « NOISE » AS A COMPUTATIONAL PRINCIPLE
- CONSTRAINTS ON ARTIFICIAL NETWORK ARCHITECTURE
- LOW-LEVEL IMPLEMENTATION: THE TWO-RING PROBLEM
BIOLOGICAL PRINCIPLES: BRAIN-LIKE ARCHITECTURE

[Diagram of brain-like architecture with labels for different areas such as High-level perception, Association cortex, Sensory cortex, Subcortical, Rétine, Mystical pad, Inner ear, V1, S1, A1, Thalamus, and Primary cortex.]
BIOLOGICAL PRINCIPLES: ADAPTIVE PROPERTIES

Diagram of neural pathways and cortical areas.
BIOLOGICAL PRINCIPLES: SENSORY SUBSTITUTION
BIOLOGICAL PRINCIPLES

- **SYSTEM LEVEL**: BRAIN-LIKE ARCHITECTURE
- **NEURAL PROCESSING LEVEL**
- **"NOISE" AS A COMPUTATIONAL PRINCIPLE**
- **CONSTRAINTS ON ARTIFICIAL NETWORK ARCHITECTURE**
- **LOW-LEVEL IMPLEMENTATION: THE TWO-RING PROBLEM**
BIOLOGICAL PRINCIPLES: Rate vs. Time Coding

SPARSE NOISE

DENSE NOISE

CONTRAST EDGE

GRATING

INCREASED LEVELS OF COMPUTATIONAL LOAD
**BIOLOGICAL PRINCIPLES: Time Coding**

<table>
<thead>
<tr>
<th>Trial</th>
<th>PSTH</th>
<th>1/σ(V_m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#N</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**INVARINANCE OF THE FIRING PATTERN WHEN ITERATING THE SAME DENSE NOISE SEED**
BIOLOGICAL PRINCIPLES: SPIKE-TIMING DEPENDENT PLASTICITY (S.T.D.P.)

(Bi and Poo, Ann. Rev. Neurosci., 2001)
FUNCTIONAL CORRELATE OF LTP: INDUCTION OF AN ECTOPIC RECEPTIVE FIELD

\[ \tau = 95-145 \text{ ms} \]

BEFORE

AFTER

CORTICAL SUBTHRESHOLD \( (V_m) \) RECEPTIVE FIELD

THALAMIC SPIKE RF

BIOLGICAL PRINCIPLES: IN VIVO SPIKE-TIMING DEPENDENT POTENTIATION
BIOLOGICAL PRINCIPLES: STDP ALGORITHM

A

\[ P(t) \]

Pre

\[ Q(t) \]

Post

LTP

Amount of change: \( P(t_i - t_{i-1}) \)

LTD

Amount of change: \( Q(t_i - t_{i-1}) \)

no change

B

Relative change in synaptic strength

\[ \text{spike timing (ms)} \]

\[ 1 \]

\[ 0.6 \]

\[ 0.2 \]

\[ -0.2 \]

\[ -0.6 \]

\[ -100 \]

\[ -50 \]

\[ 0 \]

\[ 50 \]

\[ 100 \]

C

\[ \Delta t < 0 \]

\[ \Delta t > 0 \]

\[ \text{Change in EPSC amplitude (\%)} \]

\[ \text{Spike timing (ms)} \]

\[ -60 \]

\[ -40 \]

\[ -20 \]

\[ 0 \]

\[ 20 \]

\[ 40 \]

\[ 60 \]
BIOLOGICAL PRINCIPLES

- SYSTEM LEVEL: BRAIN-LIKE ARCHITECTURE
- NEURAL PROCESSING LEVEL
- "NOISE" AS A COMPUTATIONAL PRINCIPLE
- CONSTRAINTS ON ARTIFICIAL NETWORK ARCHITECTURE
- LOW-LEVEL IMPLEMENTATION: THE TWO-RING PROBLEM
BIOLOGICAL PRINCIPLES: NOISE CONTROL OF NEURONAL RESPONSE PROBABILITY FUNCTION

Synaptic noise, a new principle for attentional facilitation?
Hybrid artificial-biological neural networks *in vitro*

Computational Models of Background Synaptic Noise

PC-based simulator

DSP Board

Acquisition

Equations solving

Signal out

\[ \Delta t \]

Amplification and current injection (dynamic clamp)

\[ I_{\text{syn}} = g_{\text{syn}}(V_{\text{biol}} - E_{\text{syn}}) \]

BIOLOGICAL PRINCIPLES: NOISE CONTROL OF NEURONAL RESPONSE PROBABILITY FUNCTION

Input-output probability

Normalized Synaptic Conductance

Input-output probability

Background Synaptic Noise

Hybrid noise

Hybrid artificial-biological neural networks

Sensemaker (IST-2001-34712) Neuro-IT Presentation
BIOLOGICAL PRINCIPLES

- SYSTEM LEVEL: BRAIN-LIKE ARCHITECTURE
- NEURAL PROCESSING LEVEL
- "NOISE" AS A COMPUTATIONAL PRINCIPLE
- CONSTRAINTS ON ARTIFICIAL NETWORK ARCHITECTURE
- LOW-LEVEL IMPLEMENTATION: THE TWO-RING PROBLEM
BIOLOGICAL PRINCIPLES: CONSTRAINTS ON ARTIFICIAL NEURAL NETWORK ARCHITECTURES

A generation 1 / 2

B generation 3

C generation 4

PERCEPTRONS
HOPFIELD / BOLTZMANN
MCCulloch-Pitts NEURONS
ACTIVATION FUNCTION

SPIKING RECURRENT NEURAL NETWORKS
INTEGRATE-AND-FIRE NEURONS

ANY-TIME COMPUTING LIQUID COMPUTATION
TIME CODING AND STDP
BIOLOGICAL PRINCIPLES

- SYSTEM LEVEL: BRAIN-LIKE ARCHITECTURE
- NEURAL PROCESSING LEVEL
- « NOISE » AS A COMPUTATIONAL PRINCIPLE
- CONSTRAINTS ON ARTIFICIAL NETWORK ARCHITECTURE
- LOW-LEVEL IMPLEMENTATION: THE TWO-RING PROBLEM
IMPLEMENTATION OF LOW-LEVEL ADAPTIVE PRINCIPLES
IMPLEMENTATION OF LOW-LEVEL ADAPTIVE PRINCIPLES: THE TWO-RING PROBLEM

Space code

Time code

Action code

Vision

Touch

Motor/Proprioceptive
IMPLEMENTATION OF LOW-LEVEL ADAPTIVE PRINCIPLES: THE TWO-RING PROBLEM
VLSI Development for the SenseMaker Project

**Task**
Design, construction and operation of VLSI based substrates for biologically inspired neural computation with sensory input and (possibly) bio-compatible output

**Concept**
Follow 2 parallel strategies and transfer results:

- close biological model - low complexity
- simplified biological model - high complexity

Ruprecht-Karls-Universität Heidelberg
Kirchhoff-Institut für Physik

Université Bordeaux
ENSEIRB-CNRS

Sensemaker (IST- 2001 - 34712) Neuro-IT Presentation
SenseMaker Roadmap from Biology to VLSI

- mixed signal perceptron
- distributed system
- mixed signal SNN
- single neuron analog SNN
- multi neuron analog SNN

VLSI Implementation of biological principles

Design and Construction | Experiments

Design and Construction | Experiments

Design and Construction | Experiments

high complexity
mixed signal
high speed
simplified biological models
scalability

low complexity
fully analog
biological speed
complex biological models
configurability
A mixed-signal Recurrent Perceptron FPNN System

interconnected network boards

network ASIC

prog. logic (FPGA)

network

synapses and neurons

network block

128 x 64 synapses

weight inputs

outputs

inputs
Mixed Signal Perceptron: Design Considerations

Hardware
- FPNN chip with mixed signal neural network function
- Analog weights, digital communication
- 0.35 µm CMOS technology
- 128 input neurons and 64 output neurons in 4 blocks per chip
- 32768 synapses with size 6µm x 12 µm
- Programmable hardware logic (FPGA) which accesses chip purely digitally and accelerate the low-level functions of the training algorithms
- Dedicated PCI card, so the hardware can be used with a common PC.

Software
- Low-level hardware abstraction layer (HAL)
- User interface

System is fully operational for experiments today
Microphotograph of the 0.35 µm VLSI Chip (HAGEN)
The Experimental Setup

- Neural network chip
- FPNN on adapter board
- Local memory to store the neural weights and the training data
- Programmable hardware to execute low-level parts of the training algorithms
- User interface and higher level parts of the training algorithms run on the PC
- PCI bus allows using the hardware with a standard PC
Distributed Neural Processing: Schematic Functional Overview

- **New card**
- **Host PC for monitoring and control**
- **Hardwired links on the backplane connect all modules**
- **Fast backplane interconnection**
Building Large Neural Networks

Example of how the FPGAs on the cards can be programmed to use the hardwired Backplane connections to build a large neural network.
Picture of Setup under Construction
Distributed Neural Processing: Picture of a network card

Connectors for future neural network chips

FPNN Socket

Backplane Connector

FPGA

SRAM
Final step towards high complexity FPNN: Implementing Low-Level Biology directly into mixed signal VLSI

Actual Integrate and Fire Model as currently planned

\[ c_m \frac{dV}{dt} = g_m(V - E_l) + \sum_k p_k g_k(V - E_x) + \sum_l p_l g_l(V - E_i) \]

- sum over excitatory synapses \( k \)
- sum over inhibitory synapses \( l \)

Synapse Model Simplifications

- \( p(t) \) only zero or one
  - spike width modulated by short-term-depression circuit of presynaptic neuron
- \( g \) 0 to \( g_{\text{max}} \) with digital (most likely 4 bit) resolution
Overview of the SNN Chip - Current Planning

- technology: UMC 0.18\(\mu\)m, 6 metal, 1 poly
- chipsize: 5 x 5 mm\(^2\) (Europractice constraints)
- 4096 neurons, 524288 synapses
- programmable synapse/neuron ratio
- fully analog network core
- continuous time network operation
- scale factor 10\(^{-6}\) to 10\(^{-5}\) : 1 ns chiptime equals 0.1 to 1 ms realtime
- short-term depression: analog on-chip
- spike-time-dependent-plasticity: digital off-chip
- event (i.e. spike) based external interface
- synaptic weights stored on-chip in SRAM located in the synapses
Neuron Response with High Random Background
(full circuit simulation)
SenseMaker Roadmap from Biology to VLSI

- mixed signal perceptron
- distributed system
- mixed signal SNN
- single neuron analog SNN
- multi neuron analog SNN

VLSI Implementation of biological principles

Design and Construction
Experiments

- high complexity
- mixed signal
- high speed
- simplified biological models

- low complexity
- fully analog
- biological speed
- complex biological models
Conductance based spiking neuron model
(Hodgkin-Huxley formalism)

\[ C_{\text{mem}} \frac{dV_{\text{mem}}}{dt} + i_{\text{Na}} + i_{\text{K}} + i_{\text{M}} + \sum i_{\text{syn}} + i_{\text{leak}} = 0 \]

\[ i_{\text{Na}} = g_{\text{Na}} \cdot m^3 \cdot h^1 \cdot (V_{\text{mem}} - V_{\text{eqNa}}) \]

\[ i_{\text{K}} = g_{\text{K}} \cdot n^4 \cdot (V_{\text{mem}} - V_{\text{eqK}}) \]

\[ i_{\text{M}} = g_{\text{M}} \cdot m_p \cdot (V_{\text{mem}} - V_{\text{eqM}}) \]

\[ i_{\text{leak}} = g_{\text{leak}} \cdot (V_{\text{mem}} - V_{\text{leak}}) \]

\[ i_{\text{syn}} = g_{\text{syn}} \cdot r \cdot (V_{\text{mem}} - V_{\text{syn}}) \]

\[ \tau_m \cdot \frac{dm}{dt} = m_\infty - m \]

\[ m_\infty = \frac{1}{1 + \exp \left( \frac{\text{offset}_m - V_{\text{mem}}}{\text{slope}_m} \right)} \]

Identical for all state variables
\((m, h, n, m_p)\)
» AMS BiCMOS 0.8mm
» ~ 10000 transistors
» 11 mm²
» 1 neuron
» 2 synapses (inhib., exci.)
» 68 pads
Excitatory neuron

ASIC « Neuron » software simulation

ASIC measurement

time dependence OK
2 cortical neurons

Stimulation current

Pyramidal Excitatory Neuron
Ampa excitatory synapse
Inhibitory Neuron
GabaA inhibitory synapse

Reciprocal synapses (average weights)

Reciprocal synapses (strong weights)
Sensor Development: High dynamic range CMOS sensor

Technical data:
- sensor: $n^+$ - substrate diode
- technology: 0.25mm CMOS (IBM)
- pixel size: 7.5mm x 7.5mm
- die size: 2mm x 4mm
- resolution: 170 x 170 pixels

Special features:
- adjustable dynamic range
- averaging of neighbouring pixels (2x2, 4x4, 8x8) (resolution matching)
- movable foveal region of high dynamic range (87x87)

submitted for production in June 03
High level design environment required with common approach
- SMU hardware
- Commercial FPGA devices

Matlab
- Standard tool for high level language and graphical exploration of various signal processing and network topologies
- Routes have now been developed and verified to target both hardware platforms
Matlab to SMU Hardware

Matlab

Network Training

SMU Support Software

Software Interface

SMU

Hardware Implementation
Programmable Hardware Implementation

Matlab to Commercial FPGAs

MATLAB Environment

Library

Simulink Blockset

Xilinx Blockset

SenseMaker SNN Blockset

Abstract SNN models

Reconfigurable Hardware

Simulink

System Generator

VHDL Code

Xilinx Integrated Software Environment (ISE)

--Synthesis compiler

-- VHDL Simulator

--FPGA Place & Route

Bitstream

Sensemaker (IST- 2001 - 34712) Neuro-IT Presentation
Programmable Hardware Implementation

**SenseMaker SNN Blockset**

- Synapses with programmable weights and synapse response functions
- Dynamic Threshold Functions
- Spike Generators
- Spike Train Integrators
- Dendritic Integrators
- STDP learning circuits
- => Enables building of Multi-layer SNN
Programmable Hardware Implementation

Generic Computational Representation of a Spiking Neuron

\[ \varepsilon(\Delta t) = \frac{\Delta t}{\tau} e^{1 - \frac{\Delta t}{\tau}} \]  
\( \Delta t = t - t_j^{(k)} \)

Spike train integrator
\[ y_j(t) = \sum_k w_{ij}^{(k)} \varepsilon(t - t_j^{(k)}) \]

Dendritic integrator
\[ x_i(t) = \sum_j y_j(t) \]

Threshold / Firing
\[ \Theta(t) \]
Programmable Hardware Implementation

SpikeProp Learning Algorithm

Learning rule for output layer

$$
\Delta w_{ij}^k(t_j^a) = -\eta \frac{y_i^k(t_j^a)(t_j^d - t_j^a)}{\sum_{i \in \Gamma_j} \sum_l w_{ij}^l \frac{\partial y_j^l(t_j^a)}{\partial t_j^a}}
$$

Learning rule for hidden layers

$$
\Delta w_{hi}^k = -\eta y_h^k(t_i^a) \delta_i = -\eta \frac{y_h^k(t_i^a) \sum_j \delta_j \sum_k w_{ij}^k \frac{\partial y_i^k(t_j^a)}{\partial t_i^a}}{\sum_{n \in \Gamma_i} \sum_l w_{ni}^l \frac{\partial y_n^l(t_i^a)}{\partial t_i^a}}
$$

where

$$
\delta_i = \frac{\sum_{j \in \Gamma_i} \delta_j \left\{ \sum_k w_{ij}^k \frac{\partial y_i^k(t_j^a)}{\partial t_i^a} \right\}}{\sum_{h \in \Gamma_i} \sum_l w_{hi}^l \frac{\partial y_h^l(t_i^a)}{\partial t_i^a}}
$$
Example 1: Modified Spikeprop Applied to Iris Data

Programmable Hardware Implementation

Encoding neurons - Spike train

In1: 4.3~7.9
In2: 2.0~4.4
In3: 1.0~6.9
In4: 0.1~2.5

Class 1  2  3

Spiking neurons
Matlab Implementation of SNN

- 6 connections to each of the 9 input neurons.
- Each connection contains 48 synapses.
- Connections to 6 neurons.
Programmable Hardware Implementation

**Synthesis Report**

*Device utilization summary* (This report is generated by Xilinx ISE 5.1i)

**Selected Device**: The Virtex-II XCV6000

- Number of Slices: 8454 out of 33792, 25%
- Number of Slice Flip Flops: 5267 out of 67584, 7%
- Number of 4 input LUTs: 10310 out of 67584, 15%
- Number of BRAMs: 60 out of 144, 41%
- Number of MULT18X18s: 7 out of 144, 4%
- Number of GCLKs: 1 out of 16, 6%

**Timing Summary**: Speed Grade: -6
- Minimum period: 23.092ns (Maximum Frequency: 43.305MHz)
- Minimum input arrival time before clock: 2.588ns
- Maximum output required time after clock: 14.229ns

**Potential Network Size**

The Virtex-II XCV6000 provides 33,792 slices. An approximate analysis indicates that the chip can thus support a network with dimensions of the order of $10^2$ neurons and $10^4$ synapses.
System Architecture

- Three phases of implementation SMS1, SMS2, SMS3

- **SMS1**
  - Based on SMU1 PCI card
  - McCulloch Pitts neurons
  - Proof of concept

- **SMS2**
  - Further development of McCulloch Pitts neurons on SMU2
  - Development of digital SNNs on commercial FPGA systems
  - Essential intermediate step towards large spiking neuron assembly
  - Exploration of high level controller architecture on commercial FPGAs
  - Implementation of two ring problem
  - Investigation of various training algorithms
System Architecture

**SMS3**

- SMU3, commercial FPGA & IXL analog board
- Spiking neuron based
- Communication via high speed serial links
- IXL analog output board used to generate biologically compatible signals for interfacing to hybrid silicon–biological systems
System Architecture

**SMS3**

- High Level Controller
  - Programmable Logic System (Virtex Pro/Virtex II)
  - Backplane
  - High Speed Link (Virtex - Virtex Rocket I/O)
  - Backplane
  - Comms Unit
  - SMU3
  - IXL Board

- Digital Sensor Input
- Digital Transducer Output
- Analog Transducer Output

Sensemaker (IST- 2001 - 34712) Neuro-IT Presentation
System Architecture

High Level Controller
System Architecture

SOFNN High Level Controller

Input layer -> RBF layer -> Normalized layer -> Weighted layer -> Output layer

SOFNN High Level Controller

MF1j(c1j, σ1j)
MF2j(c2j, σ2j)
MFrj(c_rj, σ_rj)

x1
x2
xr

µij
µ2j
µ_rj

φ
**Fuzzy Neural High Level Controller**

- **Adding a neuron- rule based approach**
  - Two criteria to judge re adding a RBF neuron
  - error criterion considers the generalization performance of the overall network
  - if-part criterion evaluates whether existing fuzzy rules or RBF neurons can cover and cluster the input vector suitably.

- **Pruning a neuron**
  - Combines the Optimal Brain Surgeon approach with RLS algorithm
  - Deletes the least important neuron if the performance of the network is within the desired tolerance limit
Presentation Summary

- Presentation has covered:
  - Low-level biological principles
  - Design of a mixed analog/digital simulator for biologically-realistic neuron networks
  - VLSI based implementations of neural network systems and sensors
  - Programmable hardware implementations
  - System architecture