

# Space-Time Computing with Temporal Neural Networks

# Synthesis Lectures on Computer Architecture

## Editor

**Margaret Martonosi**, *Princeton University*

*Synthesis Lectures on Computer Architecture* publishes 50- to 100-page publications on topics pertaining to the science and art of designing, analyzing, selecting, and interconnecting hardware components to create computers that meet functional, performance, and cost goals. The scope will largely follow the purview of premier computer architecture conferences, such as ISCA, HPCA, MICRO, and ASPLOS.

## Space-Time Computing with Temporal Neural Networks

James E. Smith

May 2017

## Hardware and Software Support for Virtualization

Edouard Bugnion, Jason Nieh, and Dan Tsafir

February 2017

## Datacenter Design and Management: A Computer Architect's Perspective

Benjamin C. Lee

February 2016

## A Primer on Compression in the Memory Hierarchy

Somayeh Sardashti, Angelos Arelakis, Per Stenström, and David A. Wood

December 2015

## Research Infrastructures for Hardware Accelerators

Yakun Sophia Shao and David Brooks

November 2015

## Analyzing Analytics

Rajesh Bordawekar, Bob Blainey, and Ruchir Puri

November 2015

### Customizable Computing

Yu-Ting Chen, Jason Cong, Michael Gill, Glenn Reinman, and Bingjun Xiao  
July 2015

### Die-stacking Architecture

Yuan Xie and Jishen Zhao  
June 2015

### Single-Instruction Multiple-Data Execution

Christopher J. Hughes  
May 2015

### Power-Efficient Computer Architectures: Recent Advances

Magnus Själander, Margaret Martonosi, and Stefanos Kaxiras  
December 2014

### FPGA-Accelerated Simulation of Computer Systems

Hari Angepat, Derek Chiou, Eric S. Chung, and James C. Hoe  
August 2014

### A Primer on Hardware Prefetching

Babak Falsafi and Thomas F. Wenisch  
May 2014

### On-Chip Photonic Interconnects: A Computer Architect's Perspective

Christopher J. Nitta, Matthew K. Farrens, and Venkatesh Akella  
October 2013

### Optimization and Mathematical Modeling in Computer Architecture

Tony Nowatzki, Michael Ferris, Karthikeyan Sankaralingam, Cristian Estan, Nilay Vaish, and David Wood  
September 2013

### Security Basics for Computer Architects

Ruby B. Lee  
September 2013

### The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, Second edition

Luiz André Barroso, Jimmy Clidaras, and Urs Hölzle  
July 2013

### Shared-Memory Synchronization

Michael L. Scott

June 2013

### Resilient Architecture Design for Voltage Variation

Vijay Janapa Reddi and Meeta Sharma Gupta

June 2013

### Multithreading Architecture

Mario Nemirovsky and Dean M. Tullsen

January 2013

### Performance Analysis and Tuning for General Purpose Graphics Processing Units (GPGPU)

Hyesoon Kim, Richard Vuduc, Sara Bagsorkhi, Jee Choi, and Wen-mei Hwu

November 2012

### Automatic Parallelization: An Overview of Fundamental Compiler Techniques

Samuel P. Midkiff

January 2012

### Phase Change Memory: From Devices to Systems

Moinuddin K. Qureshi, Sudhanva Gurumurthi, and Bipin Rajendran

November 2011

### Multi-Core Cache Hierarchies

Rajeev Balasubramonian, Norman P. Jouppi, and Naveen Muralimanohar

November 2011

### A Primer on Memory Consistency and Cache Coherence

Daniel J. Sorin, Mark D. Hill, and David A. Wood

November 2011

### Dynamic Binary Modification: Tools, Techniques, and Applications

Kim Hazelwood

March 2011

### Quantum Computing for Computer Architects, Second Edition

Tzvetan S. Metodji, Arvin I. Faruque, and Frederic T. Chong

March 2011

### High Performance Datacenter Networks: Architectures, Algorithms, and Opportunities

Dennis Abts and John Kim

March 2011

[Processor Microarchitecture: An Implementation Perspective](#)

Antonio González, Fernando Latorre, and Grigorios Magklis

December 2010

[Transactional Memory, 2nd edition](#)

Tim Harris , James Larus, and Ravi Rajwar

December 2010

[Computer Architecture Performance Evaluation Methods](#)

Lieven Eeckhout

December 2010

[Introduction to Reconfigurable Supercomputing](#)

Marco Lanzagorta, Stephen Bique, and Robert Rosenberg

2009

[On-Chip Networks](#)

Natalie Enright Jerger and Li-Shiuan Peh

2009

[The Memory System: You Can't Avoid It, You Can't Ignore It, You Can't Fake It](#)

Bruce Jacob

2009

[Fault Tolerant Computer Architecture N](#)

Daniel J. Sorin

2009

[The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines](#)

Luiz André Barroso and Urs Hölzle

2009

[Computer Architecture Techniques for Power-Efficiency](#)

Stefanos Kaxiras and Margaret Martonosi

2008

[Chip Multiprocessor Architecture: Techniques to Improve Throughput and Latency](#)

Kunle Olukotun, Lance Hammond, and James Laudon

2007

[Transactional Memory](#)

James R. Larus and Ravi Rajwar

2006

Quantum Computing for Computer Architects  
Tzvetan S. Metodi and Frederic T. Chong  
2006

© Springer Nature Switzerland AG 2022

Reprint of original edition ©Morgan & Claypool 2017

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopy, recording, or any other except for brief quotations in printed reviews, without the prior permission of the publisher.

Space-Time Computing with Temporal Neural Networks  
James E. Smith

ISBN: 978-3-031-00626-5 print

ISBN: 978-3-031-01754-4 ebook

DOI 10.1007/978-3-031-01754-4

A Publication in the Springer series

*SYNTHESIS LECTURES ON COMPUTER ARCHITECTURE* #39

Series Editor: Margaret Martonosi, Princeton University

Series ISSN: 1935-3235 Print 1935-3243 Electronic

# Space-Time Computing with Temporal Neural Networks

**James E. Smith**

Professor Emeritus, University of Wisconsin

*SYNTHESIS LECTURES ON COMPUTER ARCHITECTURE #39*



## ABSTRACT

Understanding and implementing the brain's computational paradigm is the one true grand challenge facing computer researchers. Not only are the brain's computational capabilities far beyond those of conventional computers, its energy efficiency is truly remarkable. This book, written from the perspective of a computer designer and targeted at computer researchers, is intended to give both background and lay out a course of action for studying the brain's computational paradigm. It contains a mix of concepts and ideas drawn from computational neuroscience, combined with those of the author.

As background, relevant biological features are described in terms of their computational and communication properties. The brain's neocortex is constructed of massively interconnected neurons that compute and communicate via voltage spikes, and a strong argument can be made that precise spike timing is an essential element of the paradigm. Drawing from the biological features, a mathematics-based computational paradigm is constructed. The key feature is spiking neurons that perform communication and processing in space-time, with emphasis on time. In these paradigms, time is used as a freely available resource for both communication and computation.

Neuron models are first discussed in general, and one is chosen for detailed development. Using the model, single-neuron computation is first explored. Neuron inputs are encoded as spike patterns, and the neuron is trained to identify input pattern similarities. Individual neurons are building blocks for constructing larger ensembles, referred to as "columns". These columns are trained in an unsupervised manner and operate collectively to perform the basic cognitive function of pattern clustering. Similar input patterns are mapped to a much smaller set of similar output patterns, thereby dividing the input patterns into identifiable clusters. Larger cognitive systems are formed by combining columns into a hierarchical architecture. These higher level architectures are the subject of ongoing study, and progress to date is described in detail in later chapters. Simulation plays a major role in model development, and the simulation infrastructure developed by the author is described.

## KEYWORDS

spiking neural networks, temporal models, unsupervised learning, classification, neuron models, computing theory

# Contents

|  |              |
|--|--------------|
| <b>Figure Credits</b> .....  | <b>xvii</b>  |
| <b>Preface 2019</b> .....  | <b>xxi</b>   |
| <b>Preface 2017</b> .....  | <b>xxiii</b> |
| <b>Acknowledgments</b> .....   | <b>xxvii</b> |
| <b>Part I: Introduction to Space-Time Computing and Temporal Neural Networks</b> ..... | <b>1</b>     |
| <b>1 Introduction</b> .....  | <b>3</b>     |
| 1.1 Basics of Neuron Operation .....   | 4            |
| 1.2 Space-time Communication and Computation .....                                     | 9            |
| 1.2.1 Communication .....  | 9            |
| 1.2.2 Computation .....  | 11           |
| 1.2.3 Discussion .....   | 13           |
| 1.3 Background: Neural Network Models .....  | 14           |
| 1.3.1 Rate Coding .....  | 15           |
| 1.3.2 Temporal Coding .....  | 16           |
| 1.3.3 Rate Processing .....  | 17           |
| 1.3.4 Spike Processing .....   | 18           |
| 1.3.5 Summary and Taxonomy .....   | 19           |
| 1.4 Background: Machine Learning .....   | 20           |
| 1.5 Approach: Interaction of Computer Engineering and Neuroscience .....               | 21           |
| 1.6 Bottom-Up Analysis: A Guiding Analogy .....  | 23           |
| 1.7 Overview .....   | 24           |
| <b>2 Space-Time Computing</b> .....  | <b>27</b>    |
| 2.1 Definition of Terms .....  | 27           |
| 2.2 Feedforward Computing Networks .....   | 28           |
| 2.3 General TNN Model .....  | 31           |
| 2.4 Space-time Computing Systems .....   | 32           |
| 2.5 Implications of Invariance .....   | 35           |
| 2.6 TNN System Architecture .....  | 37           |
| 2.6.1 Training .....   | 38           |

|          |  |           |
|----------|--|-----------|
| 2.6.2    | Computation (Evaluation)                           | 38        |
| 2.6.3    | Encoding   | 38        |
| 2.6.4    | Decoding   | 39        |
| 2.7      | Summary: Meta-Architecture                         | 41        |
| 2.7.1    | Simulation   | 42        |
| 2.7.2    | Implied Functions                                  | 43        |
| 2.8      | Special Case: Feedforward McCulloch-Pitts Networks | 44        |
| 2.9      | Race Logic   | 45        |
| <b>3</b> | <b>Biological Overview</b>                         | <b>47</b> |
| 3.1      | Overall Brain Structure (Very Brief)               | 47        |
| 3.2      | Neurons  | 48        |
| 3.2.1    | Synapses   | 50        |
| 3.2.2    | Synaptic Plasticity                                | 50        |
| 3.2.3    | Frequency-Current Relationship                     | 51        |
| 3.2.4    | Inhibition   | 53        |
| 3.3      | Hierarchy and Columnar Organization                | 55        |
| 3.3.1    | Neurons  | 55        |
| 3.3.2    | Columns (Micro-Columns)                            | 55        |
| 3.3.3    | Macro-Columns                                      | 56        |
| 3.3.4    | Regions  | 57        |
| 3.3.5    | Lobes  | 57        |
| 3.3.6    | Uniformity   | 59        |
| 3.4      | Inter-Neuron Connections                           | 60        |
| 3.4.1    | Path Distances                                     | 62        |
| 3.4.2    | Propagation Velocities                             | 62        |
| 3.4.3    | Transmission Delays                                | 63        |
| 3.4.4    | Numbers of Connections                             | 63        |
| 3.4.5    | Attenuation of Excitatory Responses                | 64        |
| 3.4.6    | Connections Summary                                | 64        |
| 3.5      | Sensory Processing                                 | 64        |
| 3.5.1    | Receptive Fields                                   | 65        |
| 3.5.2    | Saccades and Whisks                                | 67        |
| 3.5.3    | Vision Pathway                                     | 67        |
| 3.5.4    | Waves of Spikes                                    | 68        |
| 3.5.5    | Feedforward Processing Path                        | 69        |
| 3.5.6    | Precision  | 71        |

|   |   |           |
|---|---|-----------|
| 3.5.7   | Information Content   | 72        |
| 3.5.8   | Neural Processing   | 73        |
| 3.6   | Oscillations  | 75        |
| 3.6.1   | Theta Oscillations  | 76        |
| 3.6.2   | Gamma Oscillations  | 77        |
| <b>Part II: Modeling Temporal Neural Networks</b> |   | <b>79</b> |
| <b>4</b>  | <b>Connecting TNNs with Biology</b>                                 | <b>81</b> |
| 4.1   | Communication via Voltage Spikes                                    | 81        |
| 4.2   | Columns and Spike Bundles   | 82        |
| 4.3   | Spike Synchronization   | 83        |
| 4.3.1   | Aperiodic Synchronization: Saccades, Whisks, and Sniffs             | 83        |
| 4.3.2   | Periodic Synchronization  | 84        |
| 4.4   | First Spikes Carry Information                                      | 85        |
| 4.5   | Feedforward Processing  | 87        |
| 4.6   | Simplifications Summary   | 88        |
| 4.7   | Plasticity and Training   | 88        |
| 4.8   | Fault Tolerance and Temporal Stability                              | 89        |
| 4.8.1   | Interwoven Fault Tolerance  | 90        |
| 4.8.2   | Temporal Stability  | 91        |
| 4.8.3   | Noise (Or Lack Thereof)   | 92        |
| 4.9   | Discussion: Reconciling Biological Complexity with Model Simplicity | 93        |
| 4.10  | Prototype Architecture Overview                                     | 94        |
| <b>5</b>  | <b>Neuron Modeling</b>  | <b>97</b> |
| 5.1   | Basic Models  | 97        |
| 5.1.1   | Hodgkin Huxley Neuron Model   | 97        |
| 5.1.2   | Derivation of the Leaky Integrate and Fire (LIF) Model              | 98        |
| 5.1.3   | Spike Response Model (SRM0)   | 99        |
| 5.2   | Modeling Synaptic Connections                                       | 101       |
| 5.3   | Excitatory Neuron Implementation                                    | 103       |
| 5.4   | The Menagerie of LIF Neurons  | 105       |
| 5.4.1   | Synaptic Conductance Model  | 105       |
| 5.4.2   | Biexponential SRM0 Model  | 105       |
| 5.4.3   | Single Stage SRM0   | 106       |
| 5.4.4   | Linear Leak Integrate and Fire (LLIF)                               | 106       |
| 5.5   | Other Neuron Models   | 107       |

|          |   |            |
|----------|---|------------|
| 5.5.1    | Alpha Function .....                                      | 107        |
| 5.5.2    | Quadratic Integrate-and-Fire .....                        | 107        |
| 5.6      | Synaptic Plasticity and Training .....                    | 108        |
| <b>6</b> | <b>Computing with Excitatory Neurons .....</b>            | <b>111</b> |
| 6.1      | Single Neuron Clustering .....                            | 111        |
| 6.1.1    | Definitions .....   | 112        |
| 6.1.2    | Excitatory Neuron Function, Approximate Description ..... | 112        |
| 6.1.3    | Looking Ahead .....                                       | 113        |
| 6.2      | Spike Coding .....  | 114        |
| 6.2.1    | Volleys .....   | 114        |
| 6.2.2    | Nonlinear Mappings .....                                  | 115        |
| 6.2.3    | Distance Functions .....                                  | 116        |
| 6.3      | Prior Work: Radial Basis Function (RBF) Neurons .....     | 117        |
| 6.4      | Excitatory Neuron I: Training Mode .....                  | 121        |
| 6.4.1    | Modeling Excitatory Response Functions .....              | 121        |
| 6.4.2    | Training Set .....  | 121        |
| 6.4.3    | STDP Update Rule .....                                    | 122        |
| 6.4.4    | Weight Stabilization .....                                | 123        |
| 6.5      | Excitatory Neuron I: Compound Response Functions .....    | 126        |
| 6.6      | Excitatory Neuron Model II .....                          | 128        |
| 6.6.1    | Neuron Model Derivation .....                             | 129        |
| 6.6.2    | Training Mode .....                                       | 133        |
| 6.6.3    | Evaluation Mode .....                                     | 134        |
| 6.7      | Attenuation of Excitatory Responses .....                 | 138        |
| 6.8      | Threshold Detection .....                                 | 139        |
| 6.9      | Excitatory Neuron Model II Summary .....                  | 140        |
| <b>7</b> | <b>System Architecture .....</b>                          | <b>143</b> |
| 7.1      | Overview .....  | 143        |
| 7.2      | Interconnection Structure .....                           | 145        |
| 7.3      | Input Encoding .....                                      | 146        |
| 7.4      | Excitatory Column Operation .....                         | 149        |
| 7.4.1    | Evaluation .....  | 149        |
| 7.4.2    | Training .....  | 149        |
| 7.4.3    | Unsupervised Synaptic Weight Training .....               | 150        |
| 7.4.4    | Supervised Weight Training .....                          | 151        |
| 7.5      | Inhibition .....  | 152        |

|  |  |            |
|--|--|------------|
| 7.5.1  | Feedback Inhibition                                      | 153        |
| 7.5.2  | Lateral Inhibition                                       | 154        |
| 7.5.3  | Feedforward Inhibition                                   | 155        |
| 7.6  | Volley Decoding and Analysis                             | 157        |
| 7.6.1  | Temporal Flattening                                      | 158        |
| 7.6.2  | Decoding to Estimate Clustering Quality                  | 159        |
| 7.6.3  | Decoding for Classification                              | 161        |
| 7.7  | Training Inhibition                                      | 162        |
| 7.7.1  | FFI: Establishing $t_F$ and $k_F$                        | 162        |
| 7.7.2  | LI: Establishing $t_L$ and $k_L$                         | 163        |
| 7.7.3  | Excitatory Neuron Training in the Presence of Inhibition | 164        |
| <b>Part III: Extended Design Study: Clustering the MNIST Dataset</b> |  | <b>165</b> |
| <b>8</b>   | <b>Simulator Implementation</b>                          | <b>167</b> |
| 8.1  | Simulator Overview                                       | 167        |
| 8.2  | Inter-Unit Communication                                 | 168        |
| 8.3  | Simulating Time  | 169        |
| 8.4  | Synaptic Weight Training                                 | 170        |
| 8.5  | Evaluation   | 170        |
| 8.5.1  | EC block   | 170        |
| 8.5.2  | IC block   | 171        |
| 8.5.3  | VA block   | 171        |
| 8.6  | Design Methodology                                       | 171        |
| <b>9</b>   | <b>Clustering the MNIST Dataset</b>                      | <b>177</b> |
| 9.1  | MNIST Workload   | 177        |
| 9.2  | Prototype Clustering Architecture                        | 177        |
| 9.3  | OnOff Encoding   | 180        |
| 9.4  | Intra-CC Network   | 180        |
| 9.5  | Excitatory Column (EC)                                   | 181        |
| 9.6  | Lateral Inhibition                                       | 186        |
| 9.7  | 144 RFs  | 188        |
| 9.8  | Feedforward Inhibition                                   | 189        |
| 9.9  | Layer 1 Result Summary                                   | 194        |
| 9.10   | Related Work   | 195        |
| 9.11   | Considering Layer 2                                      | 197        |
| <b>10</b>  | <b>Summary and Conclusions</b>                           | <b>201</b> |

|                                   |            |
|-----------------------------------|------------|
| <b>References</b> . . . . .       | <b>205</b> |
| <b>Author Biography</b> . . . . . | <b>215</b> |

## Figure Credits

|   |    |
|---|----|
| Photo courtesy of the Archives, California Institute of Technology . . . . .  | 3  |
| Figure 3.1: From: <a href="http://www.thebrain.mcgill.ca">www.thebrain.mcgill.ca</a> . Used with permission . . . . .   | 48 |
| Figure 3.2: Reprinted from <i>Progress in Neurobiology</i> , 39, DeFelipe, J., and Fariñas, I.<br>“The pyramidal neuron of the cerebral cortex: morphological and chemical character-<br>istics of the synaptic inputs.” 563–607, Copyright © 1992, with permission from<br>Elsevier . . . . .  | 49 |
| Figure 3.3: Based on: Gerstner, W., and W. M. Kistler. <i>Spiking Neuron Models: Single Neurons, Populations, Plasticity</i> , Cambridge University Press. Copyright © 2002 Cambridge University Press . . . . .  | 51 |
| Figure 3.7: (a) From: Hill, Sean L., et al., “Statistical connectivity provides a sufficient foundation for specific functional connectivity in neocortical neural microcircuits.” <i>Proceedings of the National Academy of Sciences</i> 109, no. 42. Copyright © 2012 National Academy of Sciences. Used with permission. (b) From: Peters, Alan, and Claire Sethares. “Myelinated axons and the pyramidal cell modules in monkey primary visual cortex.” <i>Journal of Comparative Neurology</i> 365, no. 2: 232–255. Copyright © 1996 Wiley-Liss, Inc. Used with permission . . . . . | 58 |
| Figure 3.7: (d) From: Felleman, Daniel J., and David C. Van Essen. “Distributed hierarchical processing in the primate cerebral cortex.” <i>Cerebral Cortex</i> 1, no. 1 (1991): 1–47., by permission of Oxford University Press . . . . .  | 59 |
| Figure 3.8: From: Perin, Rodrigo, Thomas K. Berger, and Henry Markram. “A synaptic organizing principle for cortical neuronal groups.” <i>Proceedings of the National Academy of Sciences</i> 108, no. 13: 5419–5424. Copyright © 2011 National Academy of Sciences. Used with permission . . . . .   | 61 |
| Figure 3.9: From: Hill, Sean L., et al., “Statistical connectivity provides a sufficient foundation for specific functional connectivity in neocortical neural microcircuits.” <i>Proceedings of the National Academy of Sciences</i> 109, no. 42. Copyright © 2012 National Academy of Sciences. Used with permission . . . . .  | 62 |
| Figure 3.10: From: Bakkum, Douglas J., Zenas C. Chao, and Steve M. Potter, “Long-term activity-dependent plasticity of action potential propagation delay and amplitude in cortical networks.” <i>PLOS One</i> 3.5. Copyright © 2008 Bakkum et al. Used with permission . . . . .   | 62 |



|   |    |
|---|----|
| <b>Figure 3.11:</b> From: Hill, Sean L., et al., “Statistical connectivity provides a sufficient foundation for specific functional connectivity in neocortical neural microcircuits.” <i>Proceedings of the National Academy of Sciences</i> 109, no. 42. Copyright © 2012 National Academy of Sciences. Used with permission . . . . .                | 63 |
| <b>Figure 3.12:</b> From: Fauth, Michael, Florentin Wörgötter, and Christian Tetzlaff. “The Formation of Multi-synaptic Connections by the Interaction of Synaptic and Structural Plasticity and Their Functional Consequences.” <i>PLoS Computational Biology</i> 11, no. 1. Copyright © 2015 Fauth et al. Used with permission. . . . .               | 64 |
| <b>Figure 3.13:</b> Courtesy of Glycoforum, <a href="http://www.glycoforum.gr.jp">www.glycoforum.gr.jp</a> . . . . .  | 65 |
| <b>Figure 3.16:</b> Based on: Maldonado, Pedro, Cecilia Babul, Wolf Singer, Eugenio Rodriguez, Denise Berger, and Sonja Grün. “Synchronization of neuronal responses in primary visual cortex of monkeys viewing natural images.” <i>Journal of Neurophysiology</i> 100, no. 3: 1523–1532. Copyright © 2008 by the American Physiological Society . . . | 68 |
| <b>Figure 3.17:</b> Based on: Thorpe, Simon J., and Michel Imbert. “Biological constraints on connectionist modelling.” <i>Connectionism in Perspective</i> 63–92. Copyright © 1989 Elsevier . . . . .  | 69 |
| <b>Figure 3.18:</b> Based on: Petersen, Rasmus S., Stefano Panzeri, and Mathew E. Diamond, “Population Coding of Stimulus Location in Rat Somatosensory Cortex.” <i>Neuron</i> 32.3: 503–514. Copyright © 2001 Cell Press . . . . .   | 71 |
| <b>Figure 3.19:</b> Based On: Kermany, Einat, Asaf Gal, Vladimir Lyakhov, Ron Meir, Shimon Marom, and Danny Eytan, “Tradeoffs and Constraints on Neural Representation in Networks of Cortical Neurons.” <i>The Journal of Neuroscience</i> 30.28 (2010): 9588–9596. Copyright © 2010 Society for Neuroscience . . . . .                                | 73 |
| <b>Figure 3.20:</b> Based on: Hikosaka, Okihide, Yoriko Takikawa, and Reiko Kawagoe. “Role of the basal ganglia in the control of purposive saccadic eye movements.” <i>Physiological Reviews</i> 80, no. 3: 953–978. Copyright © 2000 The American Physiological Society. . . . .  | 74 |
| <b>Figure 3.21:</b> Based on: Lee, Jungah, HyungGoo R. Kim, and Choongkil Lee. “Trial-to-trial variability of spike response of V1 and saccadic response time.” <i>Journal of Neurophysiology</i> 104, no. 5: 2556–2572. Copyright © 2010 The American Physiological Society. . . . .   | 75 |
| <b>Figure 3.22:</b> From: Carlo, C. Nikoosh, and Charles F. Stevens. “Structural uniformity of neocortex, revisited.” <i>Proceedings of the National Academy of Sciences</i> 110, no. 4: 1488–1493. Copyright © 2013 National Academy of Sciences. Used with permission . . . . .   | 77 |

- Figure 6.5: Based on: Natschläger, Thomas, and Berthold Ruf. “Spatial and temporal pattern analysis via spiking neurons.” *Network: Computation in Neural Systems* 9, no. 3: 319–332. Copyright © 1998 Taylor & Francis . . . . . 120
- Figure 9.18: From Kheradpisheh, Saeed Reza, Mohammad Ganjtabesh, and Timothée Masquelier. “Bio-inspired unsupervised learning of visual features leads to robust invariant object recognition.” *Neurocomputing* 205 (2016): 382–392 . . . . . 196

## Preface 2019

Since the time of this book's publication, the author's approach to developing Temporal Neural Networks (TNNs) has continued to evolve. In general, this evolution has been in the direction of increasing simplicity rather than increasing complexity. The following paragraphs summarize the most significant changes.

The biggest change in approach has been with respect to synaptic modeling and training. Both in [Part III](#) and in the lead-up material in [Part II](#), emphasis is placed on compound synapses. A compound synapse is the composition of multiple simple synapses that connect the same two neurons. Each connection has a different delay. Compound synapses are both biologically plausible and computationally very expressive. This also makes them difficult to work with when developing a new computing paradigm. Consequently, the author has begun working with simpler synapse neurons in order to better understand the detailed operation of STDP before proceeding to compound synapses.

On a related matter, the averaging approach to synaptic training has been abandoned in favor of a more conventional Spike Timing Dependent Plasticity (STDP) approach, wherein the excitatory column, lateral inhibitory column, and STDP work closely together in a coordinated way, using only information local to each synapse and its associated neuron body. The averaging method was intended to simplify and streamline the simulation process. However, this approach does not generate synaptic weights that are similar to weights produced by conventional STDP. Unlike compound synapses which may be an avenue for productive future research, the averaging technique for training does not appear to be viable.

A more conventional STDP approach is also advantageous because it is naturally amenable to continual learning, in which both evaluation (inference) and training are intertwined ongoing processes. In the long run, this feature may prove to be one of the most important aspects of TNNs. Localized STDP is an essential element of the emergent learning behavior that will be crucial to the eventual success of this enterprise, and devising efficient, robust, localized STDP is a hard problem.

With regard to training inhibitory blocks, the author no longer uses the Pareto optimizing approach used in [Section 7.7](#). Although this method may eventually prove to be useful for some types of TNNs, it does not appear to be necessary for the TNNs under consideration here. Currently, inhibition parameters are manually specified, and the same parameters hold for all inhibitory columns in the same network layer.

With regard to input encoding, biologically plausible “OnOff” encoding computes the difference between a center pixel and the average of its surround. At the time this book was originally written in 2017, it was felt that similar computational properties could be achieved with an encoding composed of both the positive and negative of an image (Section 7.3). However, a closer approximation to true OnOff encoding has been found to work better. Furthermore, OnOff encoding is relatively easy to compute, so there are no apparent advantages to the positive-negative approach.

Space-time algebra and its association with Generalized Race Logic (GRL) were added late in the development of the book. This aspect of the work appears to be very promising, and a fuller development can be found in the paper: James E. Smith (2018). “Space-time algebra: A model for neocortical computation.” In *Proceedings of the 45th Annual International Symposium on Computer Architecture*, pp. 289–300, DOI: [10.1109/ISCA.2018.00033](https://doi.org/10.1109/ISCA.2018.00033).

With regard to new areas for TNN research that are not discussed in the book, dendritic computation provides significant potential for innovation. With dendritic computation, input spikes coming into the same dendrite can interact in ways that implement simple functions. For example, in terms of space-time algebra, max and min functions may be implemented in the dendrites, prior to STDP. This opens up the possibility of operations that are akin to pooling operations in conventional machine learning systems.

J. E. Smith

April 30, 2019

## Preface 2017

Understanding, and then replicating, the basic computing paradigms at work in the brain will be a monumental breakthrough in both computer engineering and theoretical neuroscience. I believe that the breakthrough (or a series of breakthroughs) will occur in the next 40 years, perhaps significantly sooner. This means that it will happen during the professional lifetime of anyone beginning a career in computer engineering today. For some perspective, consider the advances in computation that can be accomplished in 40 years.

When I started working in computer architecture and hardware design in 1972, the highest performing computer was still being constructed from discrete transistors and resistors. The CDC 7600, which began shipping in 1969, had a clock frequency that was two orders of magnitude slower than today's computers, and it had approximately 512 KB of main memory. Now, 40+ years later, we have clock frequencies measured in GHz and main memories in GB. The last 40 years has been an era of incredible technology advances, primarily in silicon, coupled with major engineering and architecture refinements that exploit the silicon advances.

It is even more interesting to consider the 40 years prior to 1972. That was a period of fundamental invention. In the early 1930s, Church, Gödel, and Turing were just coming onto the scene. Less than 30 years prior to 1972, in 1945, von Neumann wrote his famous report. Twenty years prior, in 1952, Seymour Cray was a newly minted engineer. Just before our 1972 division point, the CDC 7600 had a very fast in-order instruction pipeline and used a RISC instruction set (although the term hadn't yet been coined). Also in the first 40 years, cache memory, micro-coding, and issuing instructions out-of-order had already been implemented in commercially available computers; so had virtual memory and multi-threading.

To summarize: the most recent 40 years of computer architecture, spanning an entire career, has largely been a period of application and technology-driven *refinement*. In contrast, the 40 years before that was an era of great *invention*—the time when the giants walked the earth. Based on an admittedly self-taught understanding of neuroscience, I believe we are at the threshold of another 40 years of great invention—inventing an entirely new class of computing paradigms.

Computer architects and engineers have a number of roles to play in the discovery of new computing paradigms as used in the brain's neocortex. One role is developing large scale, Big Data platforms to manage and analyze all the information that will be generated by connectome-related projects. Another role is developing special purpose computing machines to support high performance and/or efficient implementations of models proposed by theoretical neuroscientists.

The role that I emphasize, however, is as an active participant in formulating the underlying theory of computation. That is, computer architects and engineers should be actively engaged in proposing, testing, and improving plausible computational methods that are fundamentally similar to those found in the neocortex.

Computer architecture and engineering, in the broad sense, encompasses CMOS circuits, logic design, computer organization, instruction set architecture, system software, and application software. Someone knowledgeable in computer architecture and engineering has a significant understanding of the entire spectrum of computing technologies from physical hardware to high-level software. This is a perspective that no other research specialization has.

I am sure that many computer architects and engineers would love to work on the extremely challenging, far-reaching problem of understanding and implementing paradigms as done in the brain. Unfortunately, there is a significant barrier to entry. That barrier is the daunting mountain of neuroscience literature combined with the well-established scientific culture that has grown up alongside it (e.g., vocabulary, representation style, mathematical style). This isn't insignificant, by the way: the language you use shapes the way you think.

So, how to overcome this barrier? Answering that question is the over-arching theme of this book.

First, it requires a lot of reading from the mountain of neuroscience literature; there is no shortcut, but the papers cited herein can provide some guidance. Then, by taking a bottom-up approach, computer architects and engineers will achieve their best leverage. At the bottom is the abstraction from biological neural systems to a mathematics-based formulation. In conventional computers, the analogous abstraction is from CMOS circuits to Boolean algebra. A computer architect, with some perspective and insight, can start with biological circuits (as complicated as they may seem) and model/abstract them to a practical mathematics-based computational method.

This book contains a description of relevant biological features as background. Then drawing from these biological features, a mathematics-based computational paradigm is constructed. The key feature is spiking neurons that perform communication and processing in *space-time*, with emphasis on *time*. In these paradigms, time is used as a freely available resource for communication and computation. Along the way, a prototype architecture for implementing feedforward data clustering is developed and evaluated.

Although a number of new ideas are described, many of the concepts and ideas in this book are not original with the author. Lots of ideas have been proposed and explored over the decades. At this point, there is much to be gained by carefully choosing from existing concepts and ideas, then combining them in new and interesting ways—engineering, in other words.

The particular modeling choices made in this book are but one set of possibilities. It is not even clear that the methods explored in this book are eventually going to work as planned. At the

time of this writing, the ongoing design study in the penultimate chapter ends with an incomplete prototype neural network design.

There is no doubt many other approaches and modeling choices that could be, and should be, explored. Eventually, someone will find just the right combination of ideas—and there is every reason to expect that person will be a computer engineer.

\*\*\*

If the reader's goal is to achieve a solid understanding of spike-based *Neural Computation*, then this book alone is not enough. It is important to read material from the literature concurrently. There is a long list of references at the end of this book; too long to be a practical supplementary reading list. Following is a shorter, annotated list of background material. None of the listed papers is standalone; rather, each provides a good touchstone for a particular area of related research.

### Neuron-Level Biology:

Just about any introductory textbook will do. Better yet, use a search engine to find any of a number of excellent online articles, many illustrated with nice graphics.

### Circuit-Level Biology:

Mountcastle, Vernon B. "The columnar organization of the neocortex." *Brain* 120, no. 4 (1997): 701–722.

Buxhoeveden, Daniel P. and Manuel F. Casanova. "The minicolumn hypothesis in neuroscience." *Brain* 125, no. 5 (2002): 935–951.

Hill, Sean L., et al. "Statistical connectivity provides a sufficient foundation for specific functional connectivity in neocortical neural microcircuits." *Proceedings of the National Academy of Sciences* (2012): E2885–E2894.

*The paper by Mountcastle is a classic, mostly summarizing his groundbreaking work on the column hypothesis. The paper by Buxhoeveden and Casanova is an excellent review article. The paper by Hill et al. is from the Markram group in Switzerland; it is experimental work that attempts to answer the right kinds of questions regarding connectivity.*

### Modeling:

Morrison, Abigail, Markus Diesmann, and Wulfram Gerstner. "Phenomenological models of synaptic plasticity based on spike timing." *Biological Cybernetics* 98, no. 6 (2008): 459–478.

*The operation of synapses is critical to the computational paradigm, and this is an excellent modeling paper specifically directed at synapses and synaptic plasticity. This and other work by Gerstner and group should be at the top of any reading list on neuron modeling.*

**Theory:**

Maass, Wolfgang. “Computing with spiking neurons.” In *Pulsed Neural Networks*, W. Maass and C. M. Bishop, editors, pages 55, 85. MIT Press (Cambridge), 1999.

*Maass did some seminal theoretical research in spiking neural networks. This paper summarizes much of that work along with related research by others.*

**Computation:**

Masquelier, T., and Simon J. Thorpe. “Unsupervised learning of visual features through spike timing dependent plasticity.” *PLoS Computational Biology* 3, no. 2 (2007): e31.

Bohte, Sander M., et al., “Unsupervised clustering with spiking neurons by sparse temporal coding and multilayer RBF networks.” *IEEE Transactions on Neural Networks*, 13, no. 2 (2002): 426–435.

Karnani, Mahesh, et al. “A blanket of inhibition: Functional inferences from dense inhibitory connectivity.” *Current Opinion in Neurobiology* 26 (2014): 96–102.

*Simon Thorpe is a pioneer in spiking neural networks of the type described in this book. All the work by Thorpe and associates is interesting reading, not just the paper listed here. The work by Bohte et al., builds on earlier radial basis function research, which should also be read. The paper by Karnani et al. is a nice discussion of inhibition and the modeling thereof.*

**Machine Learning:**

Ciresan, Dan, et al. “Flexible, high performance convolutional neural networks for image classification.” *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*. 2 (2011): 1237–1242.

*The neural networks being developed in this book fit within the broad discipline of machine learning. Consequently, there are some similarities with conventional machine learning approaches. This paper describes a deep convolutional neural network of about the same scale as the networks studied here.*

**Meta-Theory:**

Chaitin, Gregory. *META MATH! The Quest for Omega*. Vintage, 2008.

*The book by Chaitin is fairly easy-to-read and is imbued with the concepts and philosophy behind algorithmic information theory. When studying a computing paradigm that is not human-designed, it is good to have a “meta-” perspective.*



## Acknowledgments

I would like to thank Raquel for her love and great patience. Without her unwavering support, writing this book would not have been possible. Five years ago, Mikko Lipasti started me down this research path, along with Atif Hashmi and Andy Nere, and for that I will always be thankful. I am grateful to Mark Hill for providing the initial impetus for this book and Margaret Martonosi, his successor as editor of this series, for her insightful suggestions which had a major impact on the book's eventual direction. I thank Mario Nemirovsky for providing very helpful advice along the way, and Ravi Nair, Tim Sherwood, Abhishek Bhattacharjee, and an anonymous reviewer for their many helpful comments and suggestions. Thanks to Deb Gabriel and the staff at Morgan & Claypool for their efforts. Mike Morgan deserves a special thanks not only for his support of this book, but also for providing a forum for authors to express and disseminate their ideas.