
CONTENTS

1	Introduction	1
1.1	Building self-behaving robots	1
1.2	Neuromorphic cognitive systems	2
1.3	Neural-inspired hardware	4
1.4	Learning algorithms and neural networks for pattern recognition	5
1.5	Randomly connected neurons	6
1.6	Synaptic plasticity in real-time hardware	7
1.7	Motivation of this thesis	8
1.8	Thesis overview	9
2	Neuromorphic VLSI	11
2.1	Introduction	12
2.2	Neural dynamics in analog VLSI	12
2.3	Silicon neurons	14
2.4	Silicon synapses	16
2.4.1	Homeostatic plasticity: synaptic scaling	17
2.5	Synaptic plasticity: spike-based learning circuits	17
2.6	From circuits to networks	20
2.6.1	Recurrent neural networks	20
2.6.2	Distributed multi-chip networks	23
2.7	Experimental results	24
2.7.1	Synaptic and neural dynamics	24
2.7.2	Spike-based learning	25
2.8	Discussion	27
2.9	Conclusions	29
3	The software ecosystem	31
3.1	Introduction	32
3.2	A software front-end for real-time neuromorphic systems	33

3.3	Asynchronous communication back-end	34
3.4	Neural network definition front-end	36
3.5	Mapping models into spiking hardware	38
3.6	The pyTune tool-set	39
3.7	Methods	40
3.7.1	Chip functional blocks	40
3.7.2	Neural network topology	41
3.7.3	Automatically tuning system parameters	41
3.8	Results	42
3.9	Discussion	44
3.9.1	Towards a neuromorphic kernel	44
3.9.2	A black-box approach to configuration	44
3.10	Conclusions	45
4	Configuring the hardware	47
4.1	Introduction	48
4.2	Neuron dynamics	49
4.2.1	Statistical properties of the VLSI hardware neuron	49
4.2.2	Depolarization distribution	50
4.3	Synaptic dynamics and plasticity	50
4.3.1	Measuring transition probabilities	54
4.4	Exponential dynamics of learning synapses: bug or feature?	54
4.5	Discussion	56
4.6	Conclusions	58
5	The neural network model	59
5.1	Introduction	60
5.2	The feed-forward network with random projections	61
5.3	Ensembles of hardware classifiers: theory and simulations	63
5.3.1	Abstract model of stochastic learning synapses	63
5.3.2	Phenomenological model of the spiking implementation	64
5.3.3	The interplay between synaptic scaling and stop-learning	66
5.3.4	Examples from simulations	67
5.3.5	Combinations of weak classifiers	71
5.4	Building the neural network in VLSI	73
5.4.1	Random projections with a probabilistic mapper	73
5.4.2	Testing the VLSI perceptrons	74
5.5	Discussion	75
5.5.1	Aggregating techniques and the stochastic learning	75
5.5.2	The use of high-level simulations to calibrate low-level parameters	76
5.5.3	Similarities with boosted SVMs	76
5.5.4	Performance scaling with the number of aggregated predictors	77
5.6	Conclusions	77

6	Classification	79
6.1	Introduction	79
6.2	Aggregating the linear classifiers	81
6.3	Effects of circuit mismatch	82
6.4	Improved predictions by corrupting the decision thresholds	82
6.5	Discussion	83
6.5.1	Competition emerging from lateral inhibition	84
6.5.2	Introducing readout noise for improving readouts	85
6.6	Conclusions	85
7	Discussion	87
7.0.1	Classification systems in neuromorphic hardware	87
7.1	What is the best pre-processing?	88
7.1.1	Towards low-power cognitive hardware	89
7.2	Computational costs	90
7.2.1	Support-vector-machines	91
7.3	Learning and representational power	91
7.4	Learning attractor networks	92
7.5	Spatio-temporal patterns	93
7.6	Limitations of modern computers	93
7.7	Hardware synapses and learning	94
7.8	Biological plausibility	96
7.8.1	Evidence for randomly connected neurons	96
7.8.2	Sensory fusion and reward in the thalamus	96
7.9	The de-coupling role of stochasticity and mismatch	96
8	Conclusions	99
8.1	Outlook	100
8.2	Lessons learned	101
	Appendices	103
A	Other experiments	105
A.1	Sharpening tuning curves	105
A.2	Neuromorphic olfaction	106
A.3	Considerations on STDP	108
A.3.1	Studying coincidences	109