Neuromorphic z Vertex Trigger Studies for the Belle II Detector

Sebastian Skambraks

July 23, 2012

Introduction

Goals

- build a z-vertex trigger
- high precision (spatial resolution)
- fast decision (under 5 μs)

Method

- neural networks as classifier (MLP, LSM)
- CDC data as input (Wire ID & Drift Time)
- classifiers tested on simplified MC data

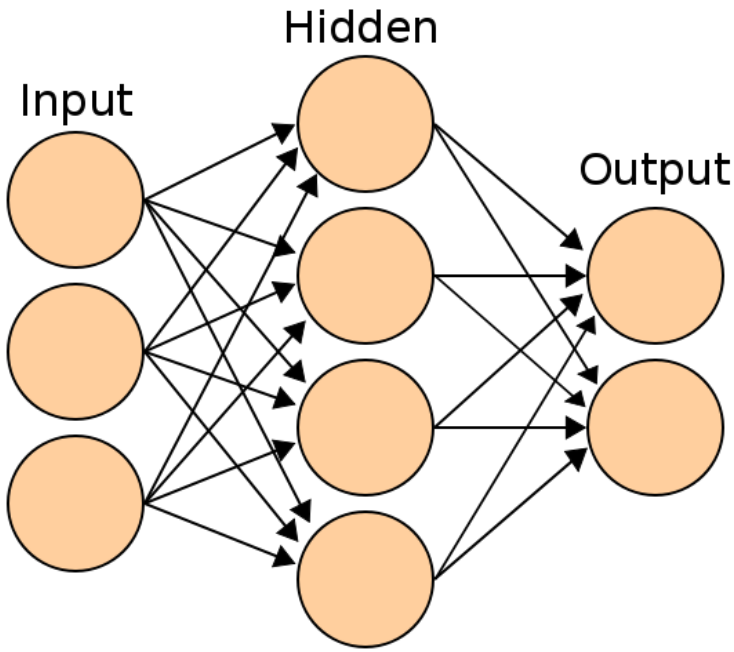
MLP - Multi Layer Perceptron [1]

$$y(\vec{x}) = g(\vec{x} \cdot \vec{w}) \tag{1}$$

with y: neuron output, g: activation function, $\vec{x}$: inputs, $\vec{w}$: weights.

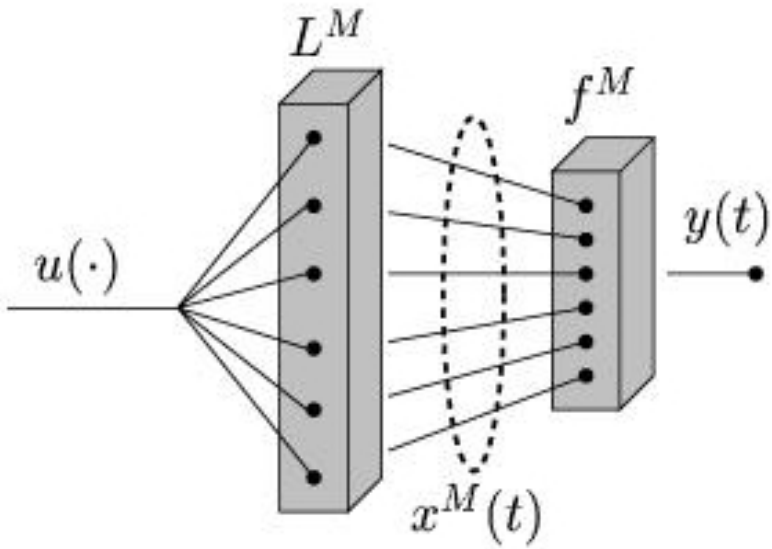
used configuration:

- 3 Layer Network, feedforward
- activation functions: tanh
- size hidden = 2 * size input
- 1 output neuron
- training of weights by backpropagation



LSM - Liquid State Machine [2]

A LSM consists of a liquid and a readout. The liquid serves as a **filter**, transforming the input onto a **higher dimensional space** and thus making it better separable (e.g. linear). The readout does the classification.



- u: input
- L^M: liquid
- x^M: liquid state, $x^M(t) = (L^M u)(t)$
- f^M: readout for liquid
- y: result, $y(t) = f^M(x^M(t))$

Liquid conditions:

1. Separation by the liquid
2. Approximation by the readout

Spiking Neurons - Integrate and Fire (IF) Model [4]

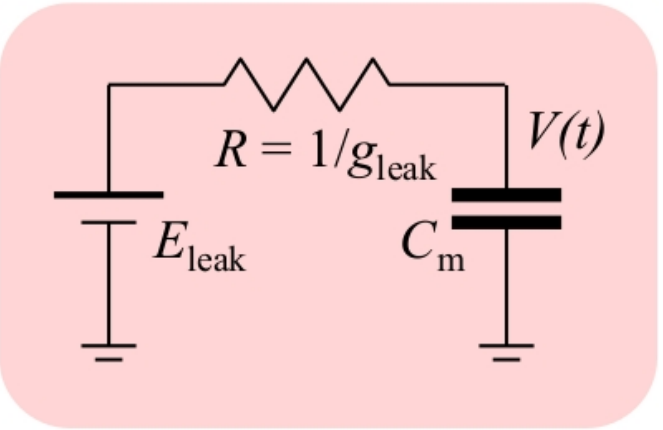
Biologically inspired model

recurrent connected network serves as **liquid**

Differential equation for the membrane potential:

$$\underbrace{C\frac{dV}{dt}}_{I_m} = \underbrace{-g_L(V-E_L)}_{I_l} + \sum_j \underbrace{p_j g_j (V-E_e)}_{I_j} + \sum_k \underbrace{p_k g_k (V-E_i)}_{I_k} \qquad (2)$$

with C membrane capacitance,
g_i conductances,
E_i reversal Potentials,
$I_{m,l,k,j}$: membrane, leak, and synapse currents
p_i: input spike shapes



- $V(t)$ represents the state of a neuron
- numerical integration can be implemented on hardware (RC - circuits)

LSM

Used configuration

- liquid neuron type: IF facets1 (cond exp) [6]
- size (liquid) $= 3 \cdot$ size (input)
- randomly and recurrently connected
- readout: Tempotron [5]- a supervised learning method for a spiking neuron

Liquid idea - example

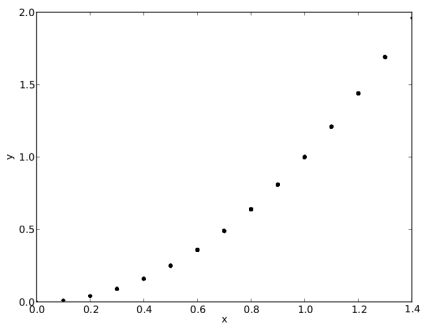
Simplification by transformation to a higher dimensional space.
Example: Process described by a quadratic equation $f : x \to \mathbb{R}$:

$$f(x) = a \cdot x^2 + b \cdot x + c \quad (3)$$

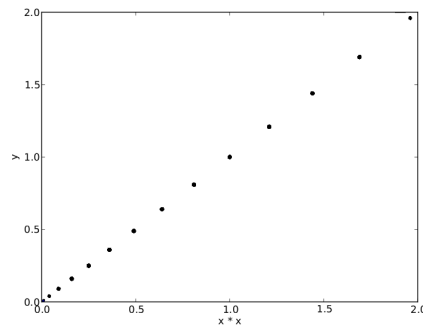
with the substition $X_1 = x^2$, $X_2 = x$ the equation becomes:

$$f(X_1, X_2) = a \cdot X_1 + b \cdot X_2 + c \quad (4)$$

$f(x)$



$f(x^2)$



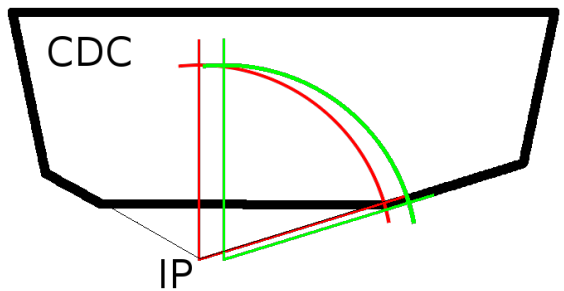
Data

- single tracks, 6GeV $\mu^{\pm}$
- generated at discrete z positions : z = (-10, -5, 0, 5, 10)cm

constraints on direction (to reduce the number of hit wires):

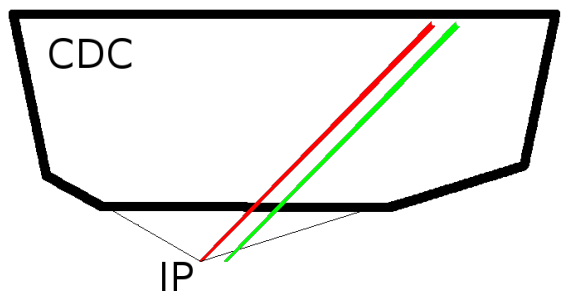
slice

- $\theta = 17°..90°$
- $\Delta\phi = 0.5°$



cone

- $\theta = 45°..46°$
- $\Delta\phi = 0.5°$



Experimental Setup

Parameter Space is searched for some of the free Parameters

MLP - Multi Layer Perceptron

- optimized for: training algorithm, input format
- input scale: $[200, 0]$ns $\rightarrow [0, 1]$
- not fired $\rightarrow 0$

LSM - Liquid State Machine

- optimized for: liquid: injected current; tempotron: threshold potential, membrane time constant
- input scale: $[0, 200]$ns $\rightarrow [0, 25]$ms
- simulation time: 25 ms
- current injection

Selection of results

parameter combinations with the highest accuracy are shown

accuracy

$$acc = \frac{N_{cor}}{N} \tag{5}$$

with N: number of trained events, N_{cor}: number of correctly classified events. The size of both classes is equal.

error

- error for LSM: standard deviation of accuracy over 1000 events
- error for MLP: mean squared error, used in the training

plots

- LSM x axis: number of trained events
- MLP x axis: number of trained epochs
- y axis: accuracy

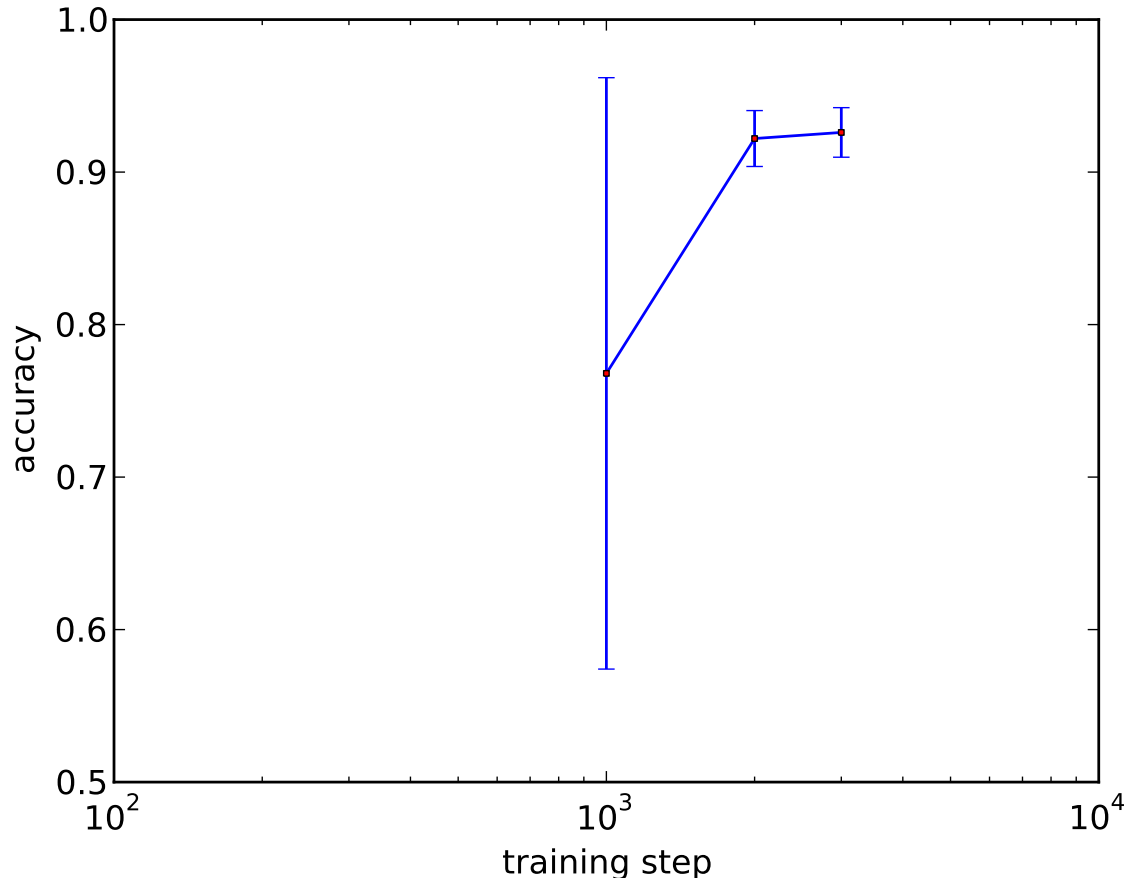


Figure: LSM, data: belle1 cone, data, $dc = 0.6nA$, $V_{thresh} = -61.4V$u, $\tau_m = 0.8ms$, z=0cm vs. z=10cm. Final accuracy = 92.5, $\sigma = 1.9$.

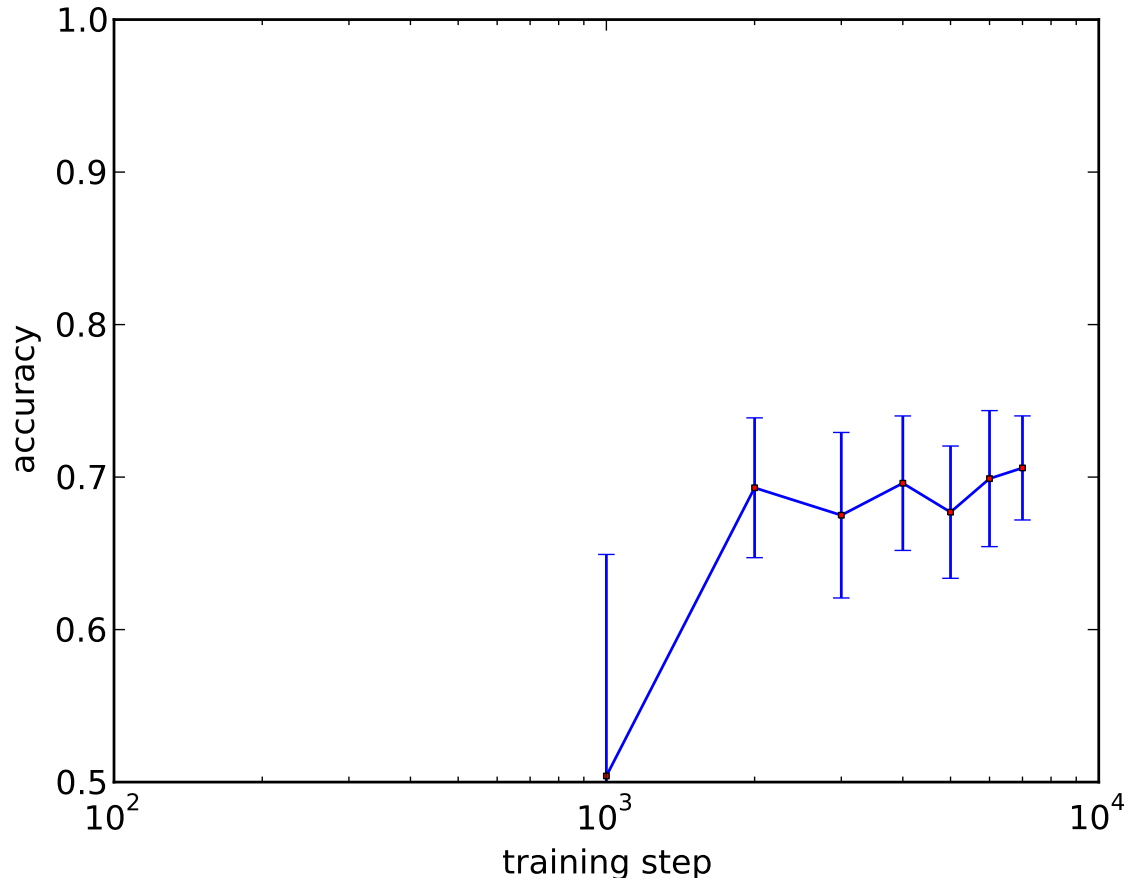


Figure: LSM, data: belle2 cone, $dc = 1.4nA$, $V_{thresh} = -60.2V$, $\tau_m = 1.0ms$, z=0cm vs. z=10cm. Final accuracy = 70, $\sigma = 5$.

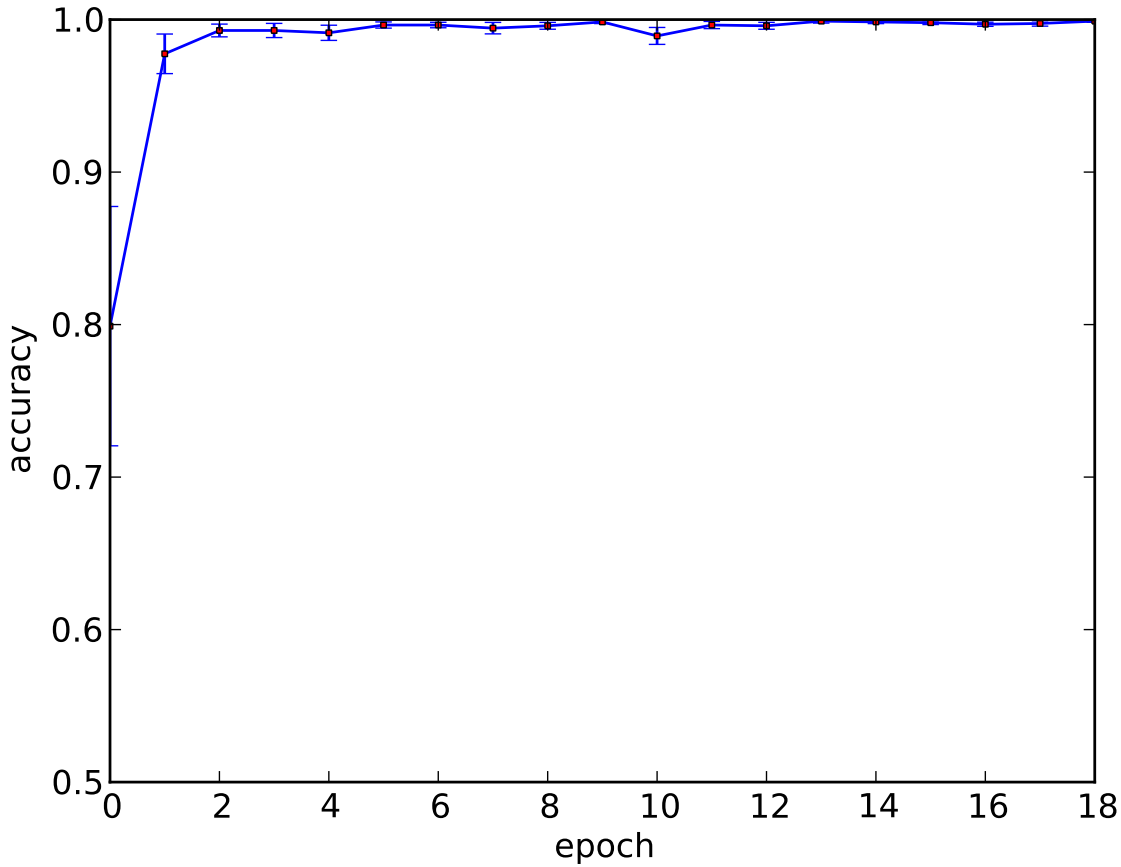


Figure: MLP, data: cone belle1, position: z = 0cm vs. z = 5cm, algorithm: incremental backprop. Final accuracy = 99.81, mse = 0.0002.

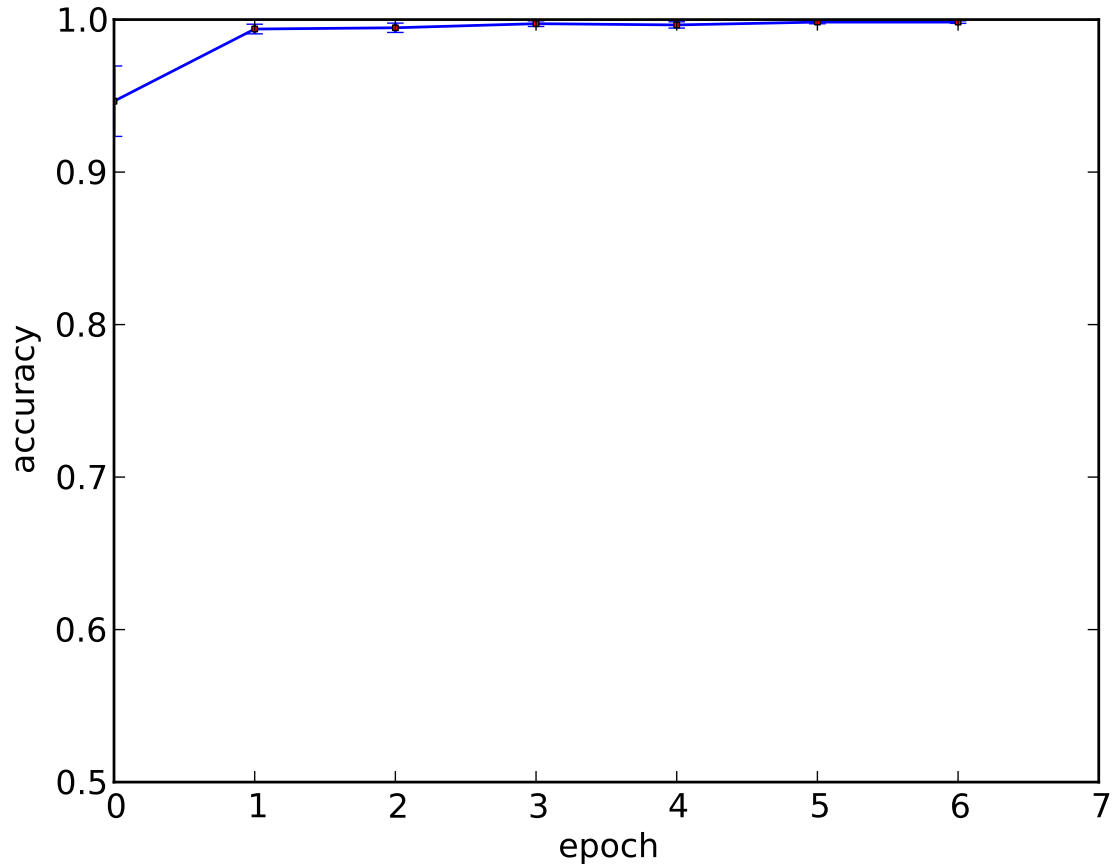


Figure: MLP, data: cone belle2, position: z = 0cm vs. z = 5cm, algorithm: incremental backprop. Final accuracy = 99.02, mse = 0.0003.

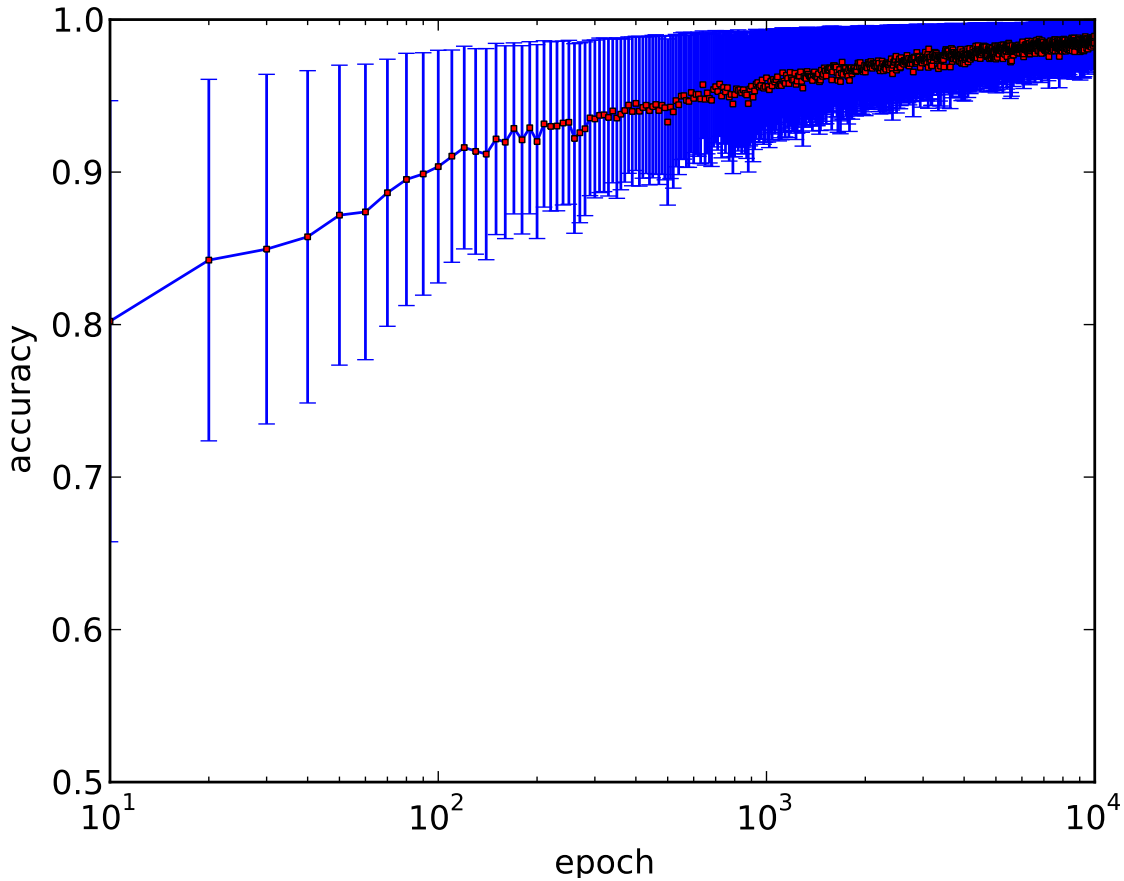


Figure: MLP, data: slice belle1, position: z = 0cm vs. z = 5cm, algorithm: incremental backprop. Final accuracy = 95.4, mse = 0.4.

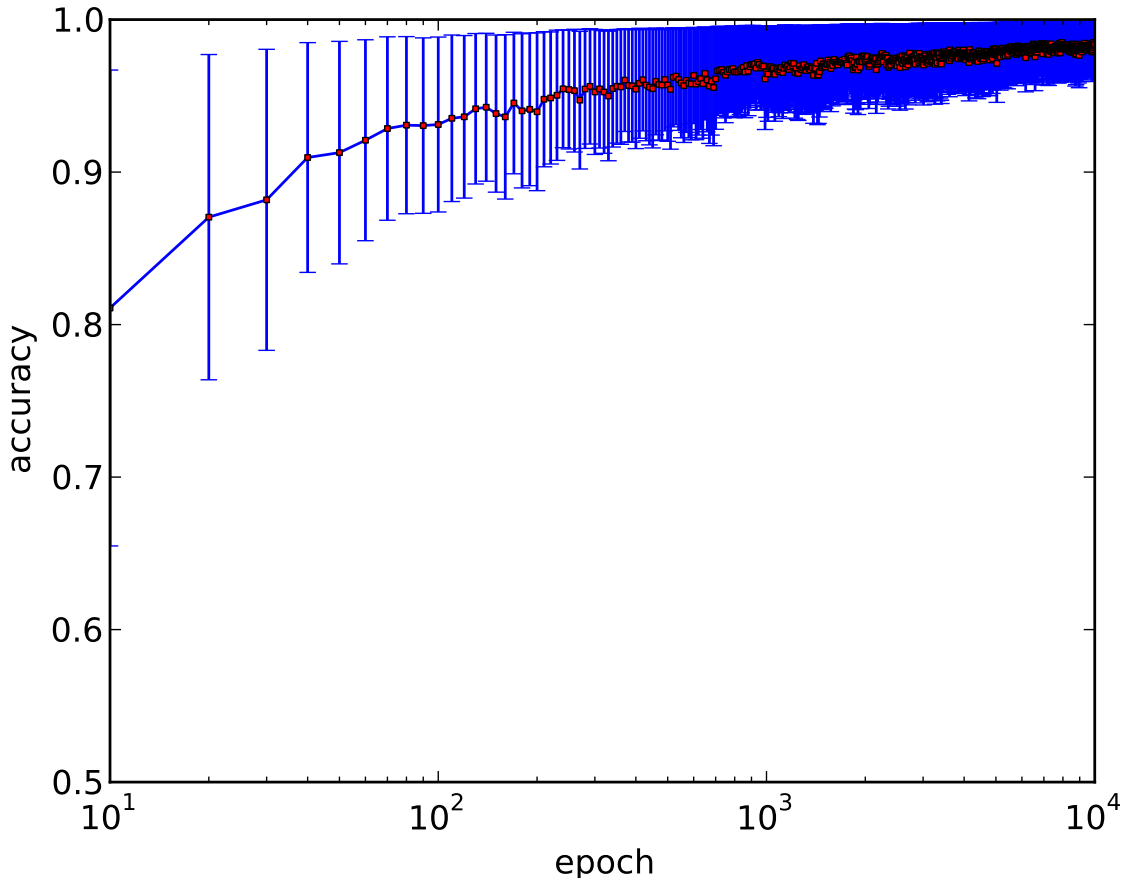


Figure: MLP, data: slice belle2, position: z = 0cm vs. z = 5cm, algorithm: incremental backprop. Final accuracy = 96.4, mse = 0.4.

Conclusion

Summary

- MLP accuracy better than LSM
- LSM has many free parameters $\rightarrow$ still work to do
- positive first results encourage further studies

outlook

- include all wires
- use more realistic particles, e.g. 4 μ events
- test several liquid architectures

References

- C. M. Bishop, Neural Networks for Pattern Recognition, Oxford University press, ISBN 978-0-19-853864-6, 1995
- W. Maas, Liquid State Machines: Motivaiton, Theory and Applications, Institute for Theoretical Computer Science, Graz University of Technology, 2010
- Belle Collaboration, Belle II Technical Design Report, arXiv:1011.0352v1, 2010
- Brette, Gerstner, Adaptive Exponential Integrate-and-Fire Model as an Effective Description of Neuronal Activity, J Neurophysiol. 94(5):3637-42., 2005
- Gütig, Sompolinsky, The tempotron: a neuron that learns spike timing-based decisions, Nature Neuroscience, doi:10.1038/nn1643, 2006
- J. Schemmel, D. Brüderle, K. Maier, B. Ostendorf, Modeling Synaptic Plasticity within Networks of Highly Accelerated I & F Neurons, Kirchhoff Institute for Physics, University of Heidelberg, 2005

Backup

Tempotron - readout [5]

supervised training of a spiking neuron.
Membrane potential:

$$V(t) = \sum_i \omega_i \sum_{t_i} K(t - t_i) + V_{rest} \quad (6)$$

$$K(t - t_i) = V_0 \cdot \left(e^{-\frac{t-t_i}{\tau}} - e^{-\frac{t-t_i}{\tau_s}} \right) \quad (7)$$

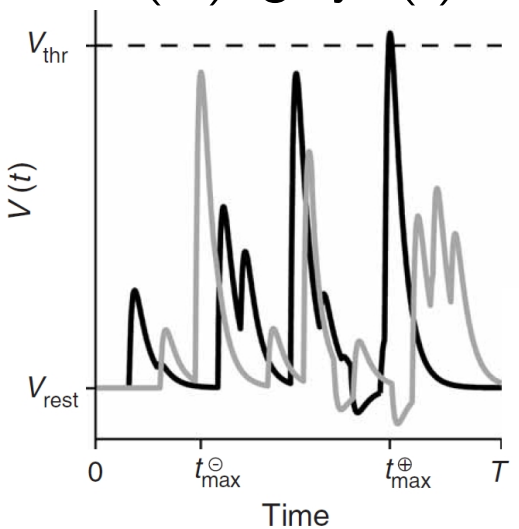
- $V(t)$: membrane potential
- ω_i: weight for the i-th input
- t, t_i: current- , i-th spike- time
- V_{rest}: resting potential
- $K(t - t_i)$: kernel function
- τ, τ_s: membrane and synaptic time constants

$$E_{\pm} = \pm(V_{thr} - V(t_{max}))\Theta(\pm V_{thr} - V(t_{max}))$$

$$\Delta w_i = \alpha \sum_{t_i < t_{max}} K(t_i - t_{max})$$

- $E_{\pm}$: cost function
- t_{max}: time of maximum PSP
- Δw: amount of weight update
- α: learn rate

black: (+); grey: (-)



Tempotron decission, source: [5]

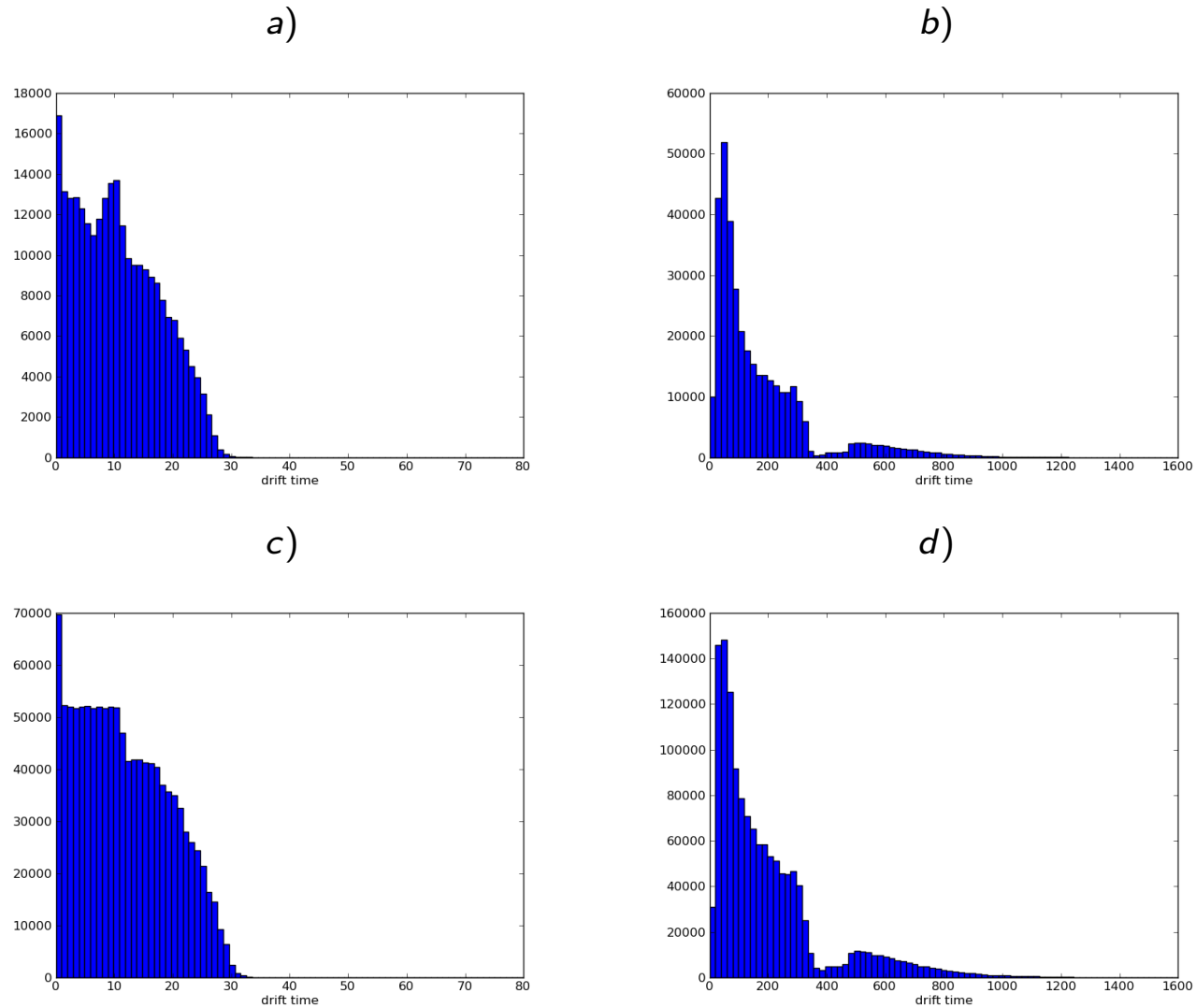


Figure: Drift time distributions for $z = 0cm$. a) belle2 slice, b) belle1 slice, c) belle2 muquad, d) belle1 muquad. The timescale for Belle2 data is 8ns (1 clock cycle), for Belle1 it is 0.5ns.