Master‘s Thesis Presentation

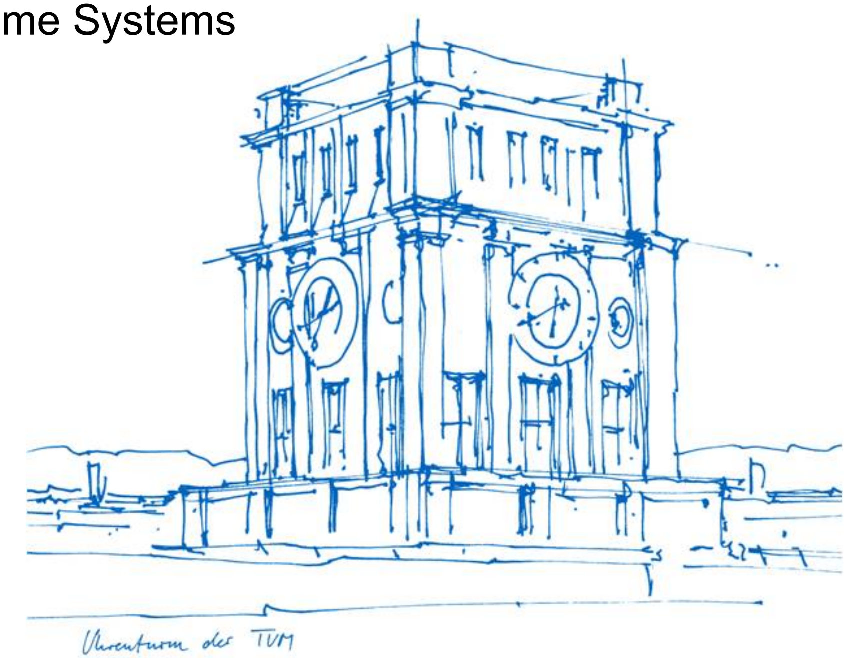
Richard Otto

Technical University of Munich

Department of Informatics

Chair of Robotics, Artificial Intelligence and Real-time Systems

Munich, 30. August 2019



Autonomous Control for Wheel-less Snake Robot based on Neuromorphic Vision Sensor and Spiking Neural Network

Richard Otto

Supervisor: Prof. Dr.-Ing. habil. Alois Christian Knoll

Advisor: Zhuangyi Jiang

Munich, 30. August 2019

Content

Motivation

Autonomous robots have huge utilities:

- Search and Rescue
- Exploration
- Transportation

Key Ability: Real Time Navigation

Developments in Computer Science and Engineering make this goals reachable:

- More computational power
- New mathematical methods and models
- Improvements in sensors and hardware

Problem description

Goal: Target tracking

Platform/Setup:
- Snake-like Robot
- Dynamic Vision Sensor
- Spiking Neural Network
- Neurorobotic Platform

Technique:
- R-STDP learning

Robot

- Snake-like Robot
- Owned by Chair
- Versatile Movement Options
- Slithering Gait used
  - Mimics biological movement
  - Head faces in movement direction
  - but moves from left to right

Z. Jiang [Jia18]

Dynamic Vision Sensor

Tracks Change of Pixel Readings

Sends Event if Change overcomes Threshold

Event: <pixel_ID,event_Time,polarity>

Eventstream is served via a Bus by the Sensor

Advantages over Cameras:

- High temporal resolution
- Less overhead by not using Frames
- Resulting in less computational effort
  - Less energy consumption
  - Higher responsiveness

Spiking Neural Network

Neurons:

- Leaky Integrate and Fire
- Model of biological Neuron
- Fires if Activation Level reaches Threshold
- Activation decays (Leak)

Synapses:

- Reward modulated STDP-Learning
- STDP-Learning used to update eligibility
- Dopamine level corresponds to the Reward

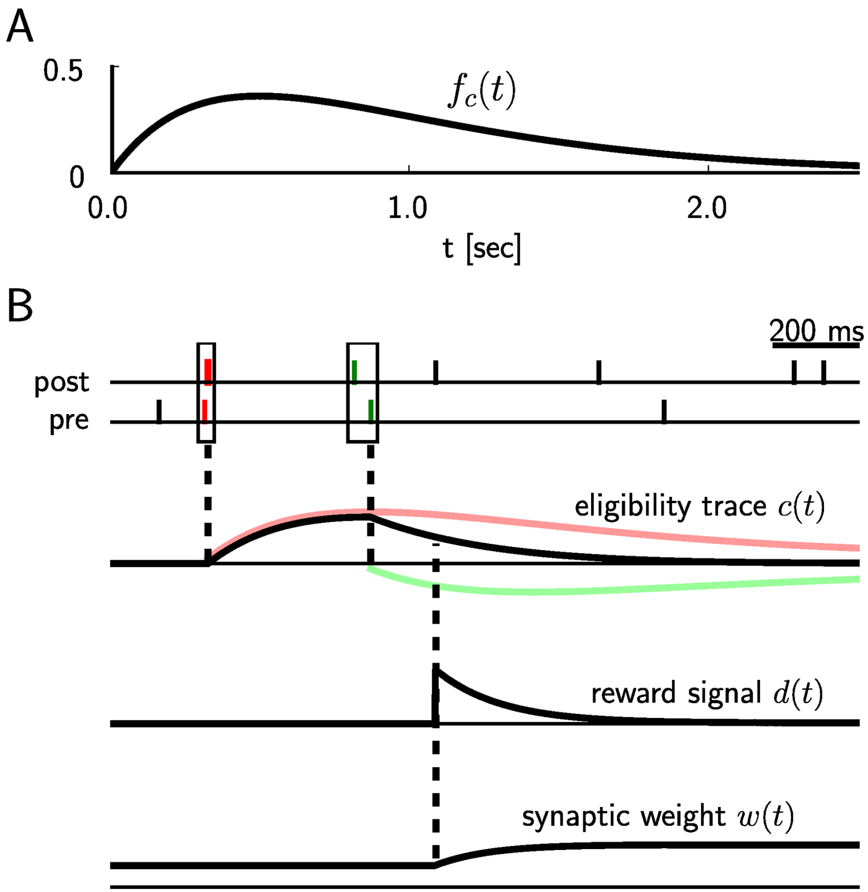
Spiking Neural Network – R-STDP

R-STDP Mechanism (R. Legenstein [L08])

$$\dot{c} = -\frac{c}{\tau_c} + STDP(\Delta t)\delta(t - s_{pre/post})C_1$$

$$\dot{n} = -\frac{n}{\tau_n} + \frac{\delta(t - s_n)}{\tau_n}C_2$$

$$\dot{w} = c(n - b)$$



Neurorobotics Platform

Used for simulating the Robot controlled by an SNN in a closed Loop

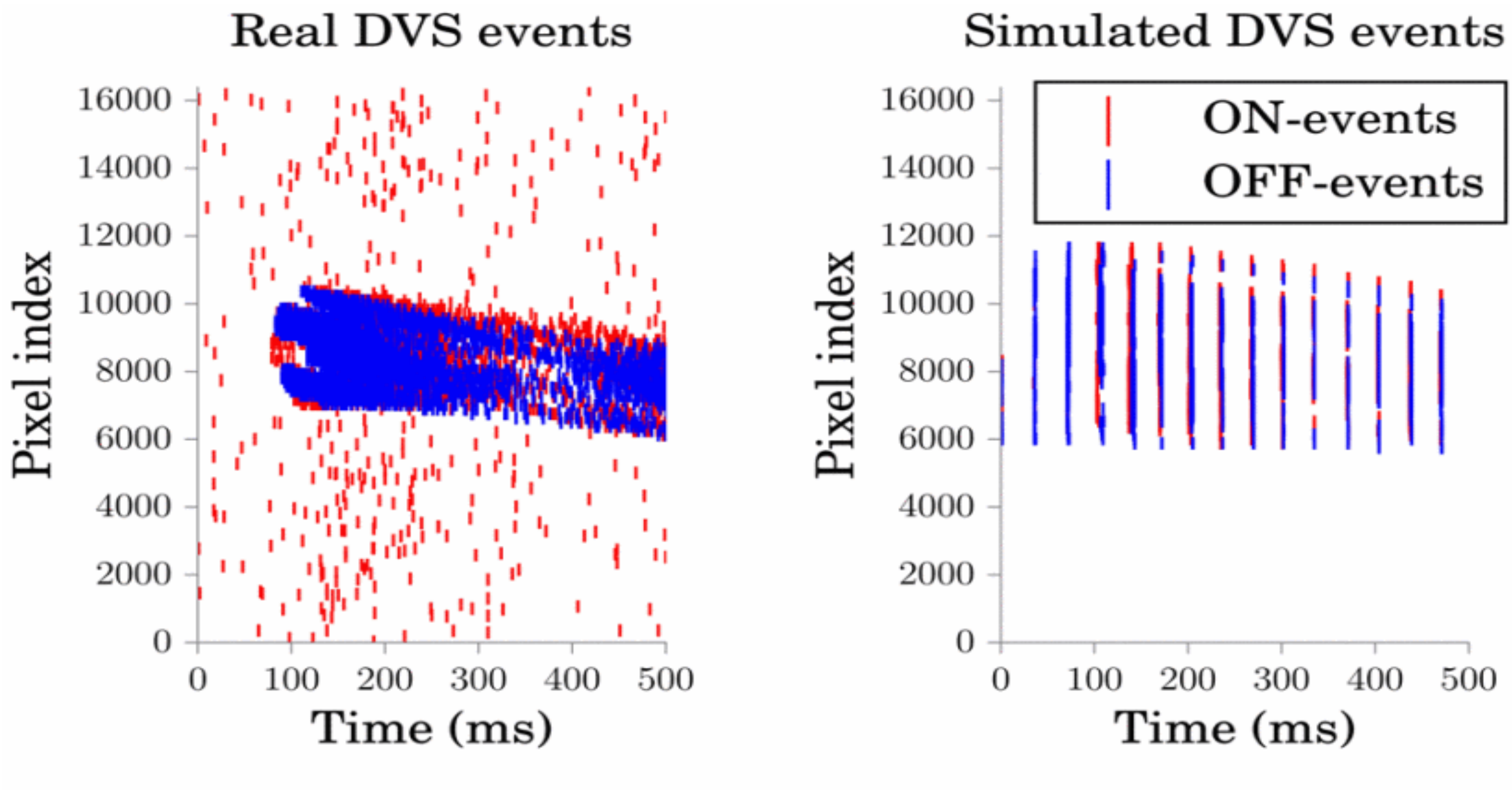
Controls SNN-Simulation and World Simulation

- Simulations are paused every 20ms
- Transfer Functions used for communication between the Simulations

Problem:

Simulated DVS Events all happen at the same time

Neurorobotics Platform



(a) (b)

J. Kaise [Kai+16]

Neurorobotics Platform

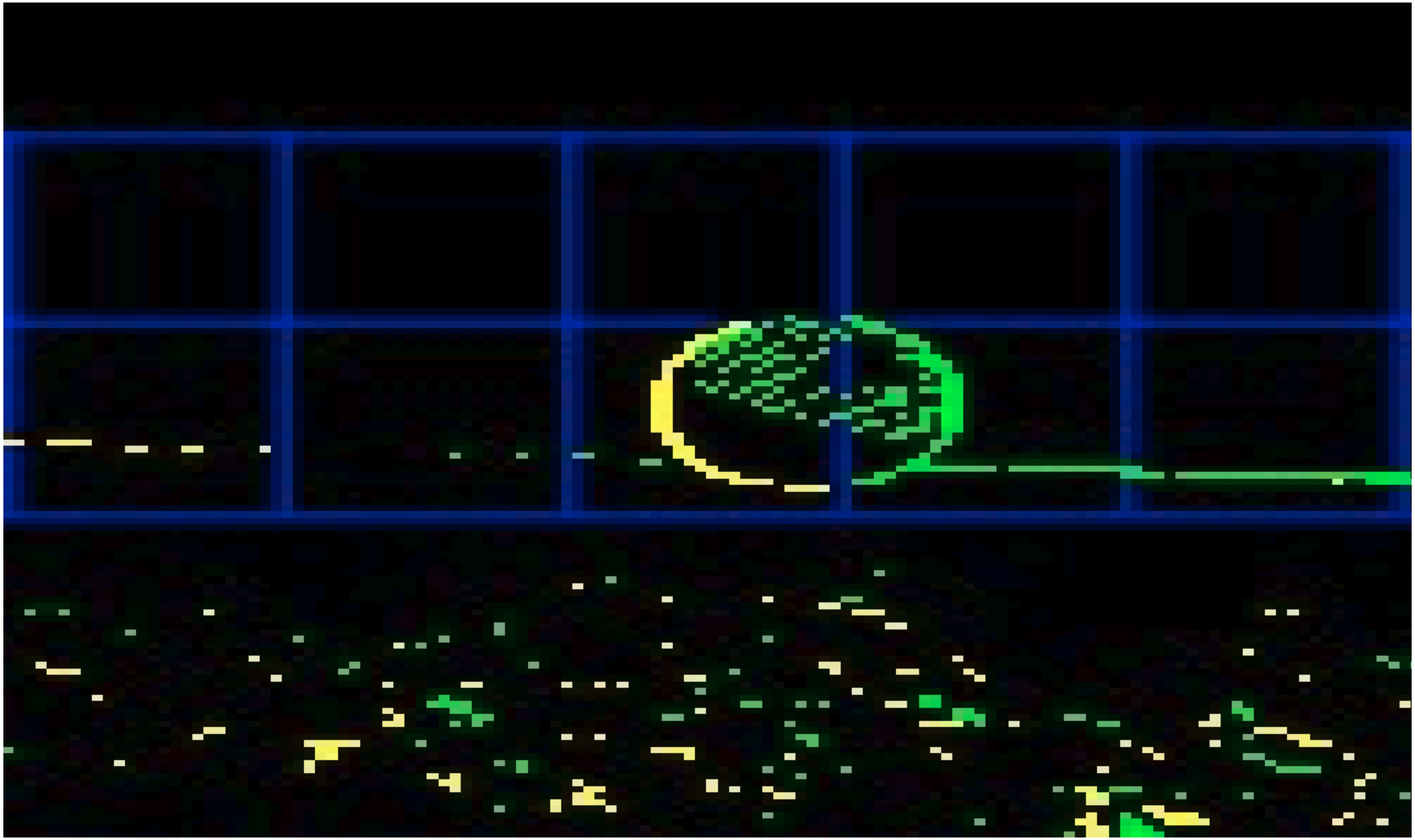
Input Encoding:

Spread events over the next timestep

- Crop input to relevant part
- Divide input into “Superpixels”
- Set Poisson Neuron to fire at a frequency corresponding to the number of events of its Superpixel:

$$f = N_{events} * 50$$

Neurorobotics Platform



Neurorobotics Platform

Output Decoding:

- 2 Neurons used
- Spikes of each get accumulated over last Timestep
- Direction given by:

$$N_d = N_r - N_l$$

$$D_{norm} = N_d / (N_r + N_l)$$

$$D_t = min(1, max(-1, (D_{t-1} + D_{norm} * c_{imp})))$$

Neurorobotics Platform

Reward:

The Angle of the Head to the Target reflects the performance of the Robot

- Reward for the Output Neurons scales linear according to Angle:

$$r_{left/right} = -/+a * c_{rew}$$

- Dopamine Level of Synapses set according to the Reward

Neurorobotics Platform

Reward Propagation:

- Reward unknown for Hidden Layer
- Rewards of the Output Layer has to be used
- Similar to the Backpropagation

$$r_{i,j} = \frac{\sum_{k=1}^{Y_i-1}(r_{i-1,k} * w_{i,j,k})}{max(|w_{i,j,1}|, |w_{i,j,Y_{i-1}-1}|, |w_{i,j,Y_{i-1}}|) * Y_{i-1}}$$

- Simplification in this case:

$$r_h = \frac{r_l * w_l + r_r * w_r}{max(|w_l|, |w_r|) * 2}$$

$$r_h = \frac{r_r * (w_r - w_l)}{max(|w_l|, |w_r|) * 2}$$

Bing, Jiang [Bin+19]

Training the Networks

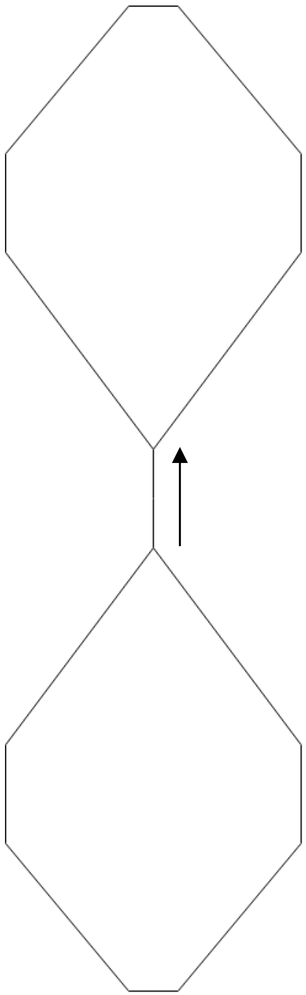
Robot and Ball in empty Environment:

Ball Radius: 30cm

Distance: 1.6m

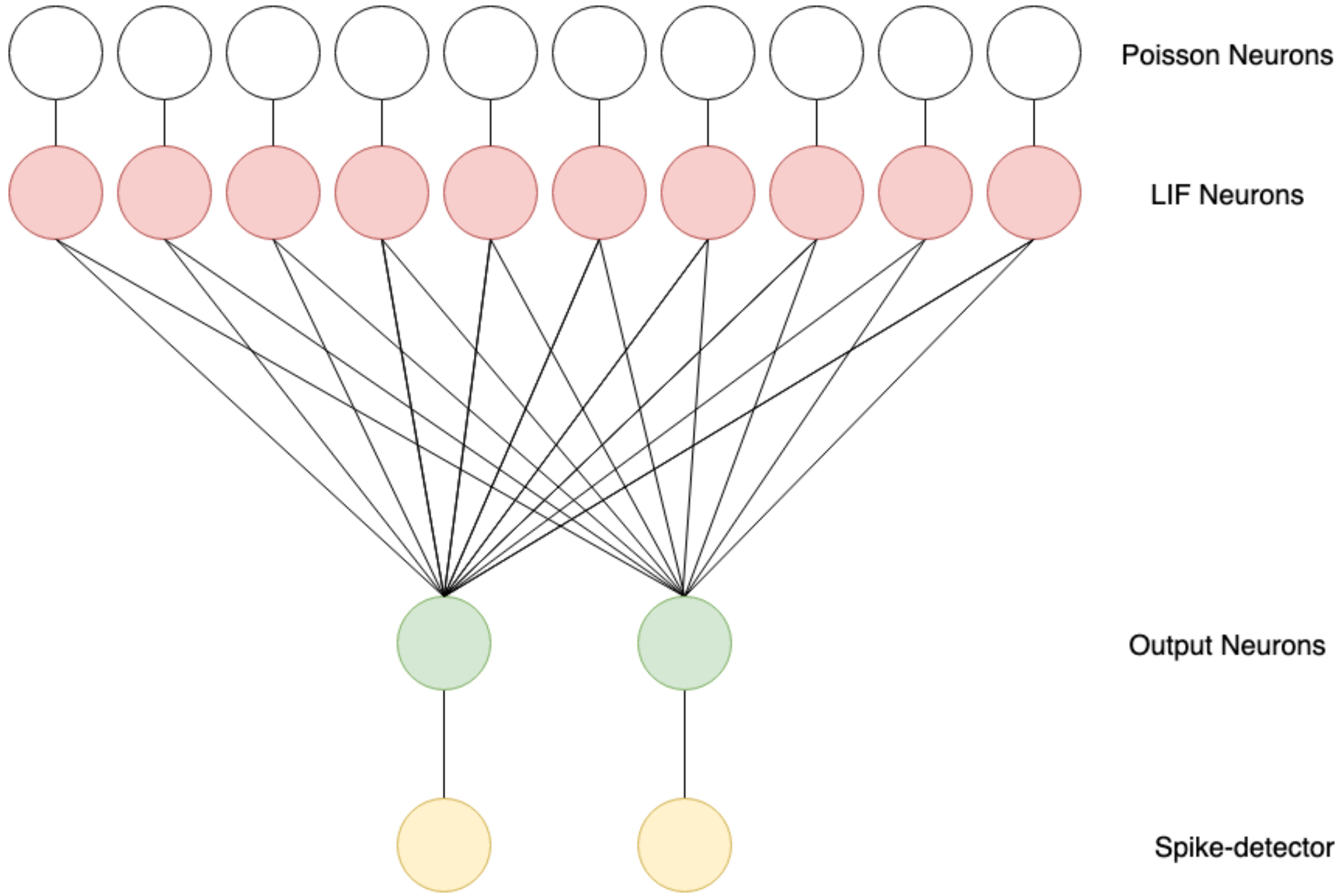
Ball movement in training:

- Eight like shape
- Ball faster than Robot
- Ball keeps distance and waits
- Track mirrored after each Reset



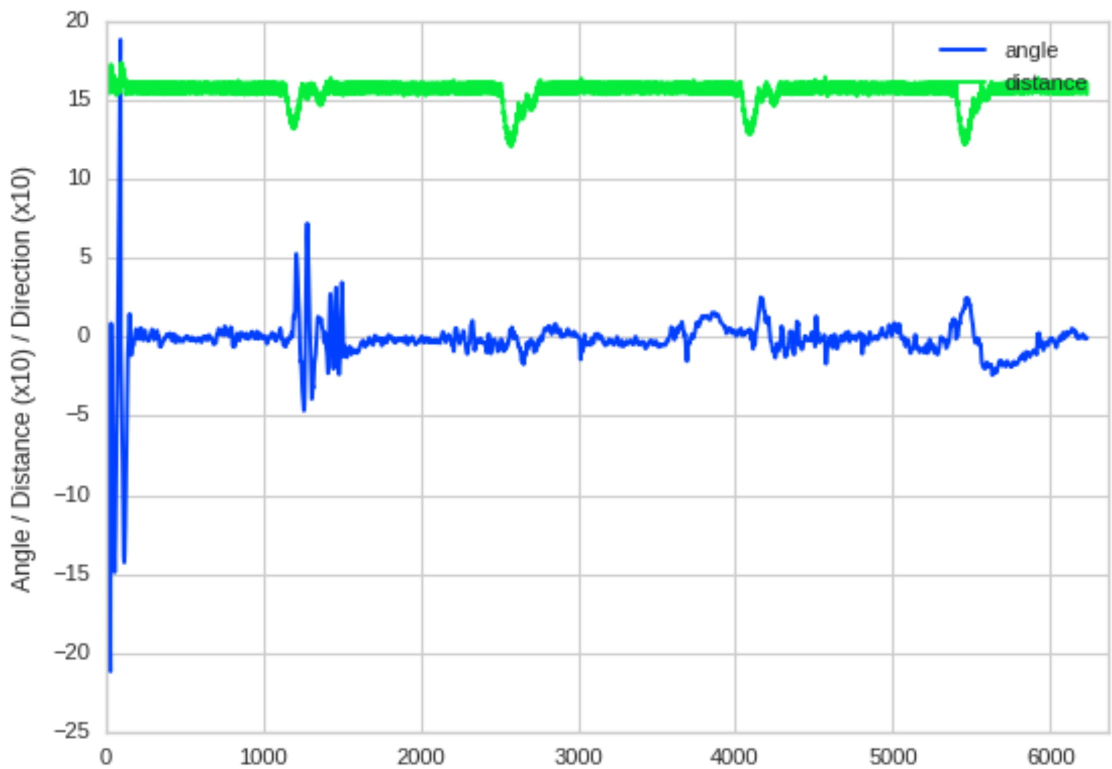
Training the Networks - Basic Network

Topology of the Network



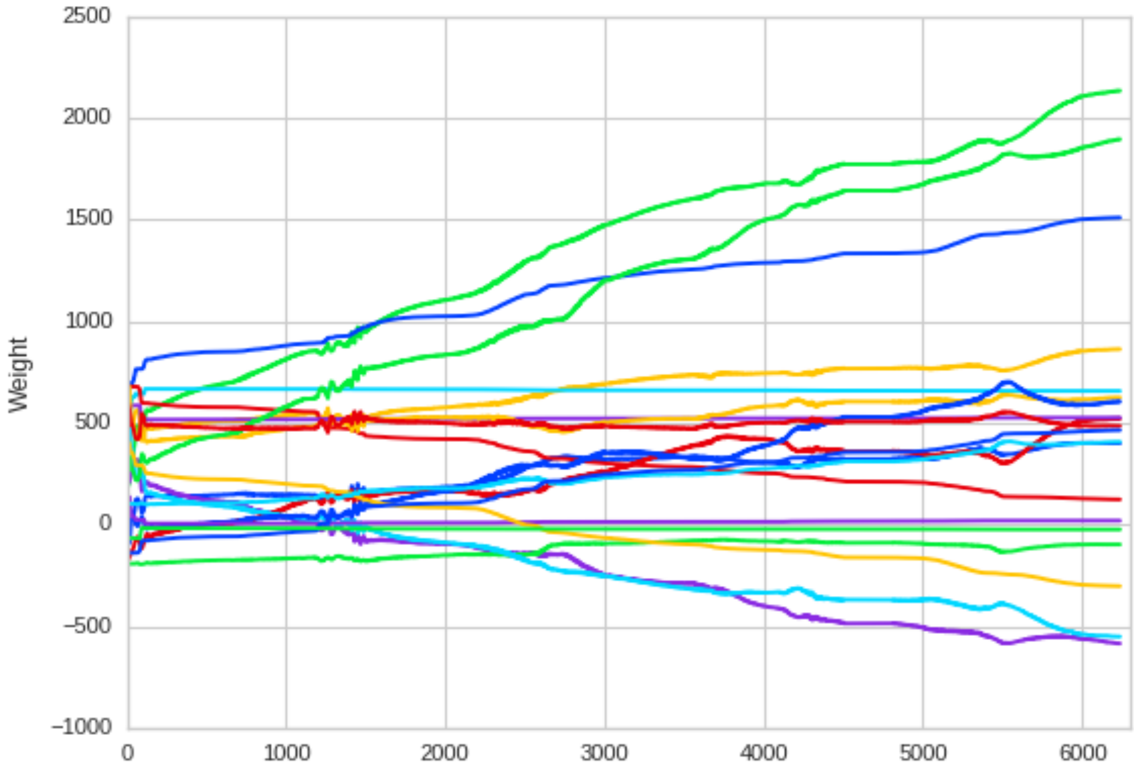
Training the Networks - Basic Network

Performance of the Network



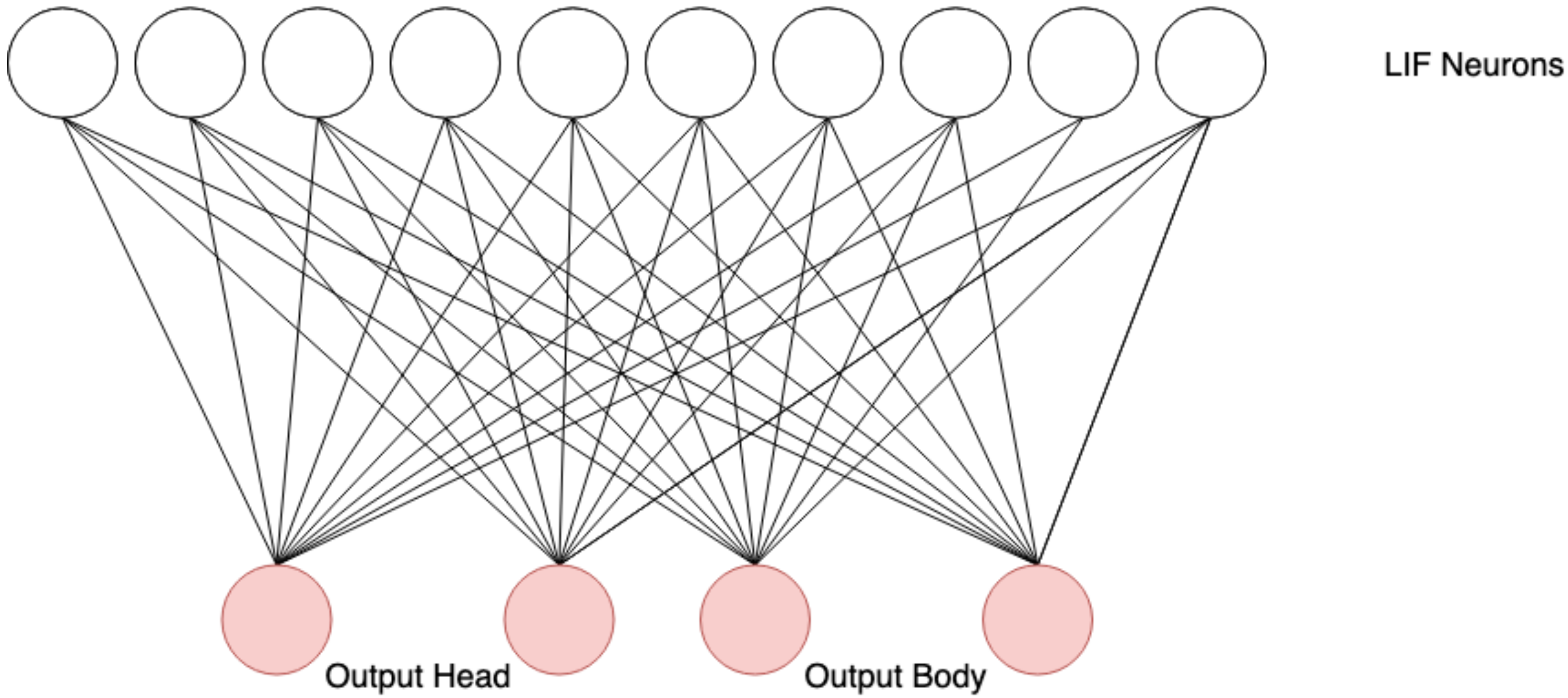
Training the Networks - Basic Network

Plot of Weights in the Network



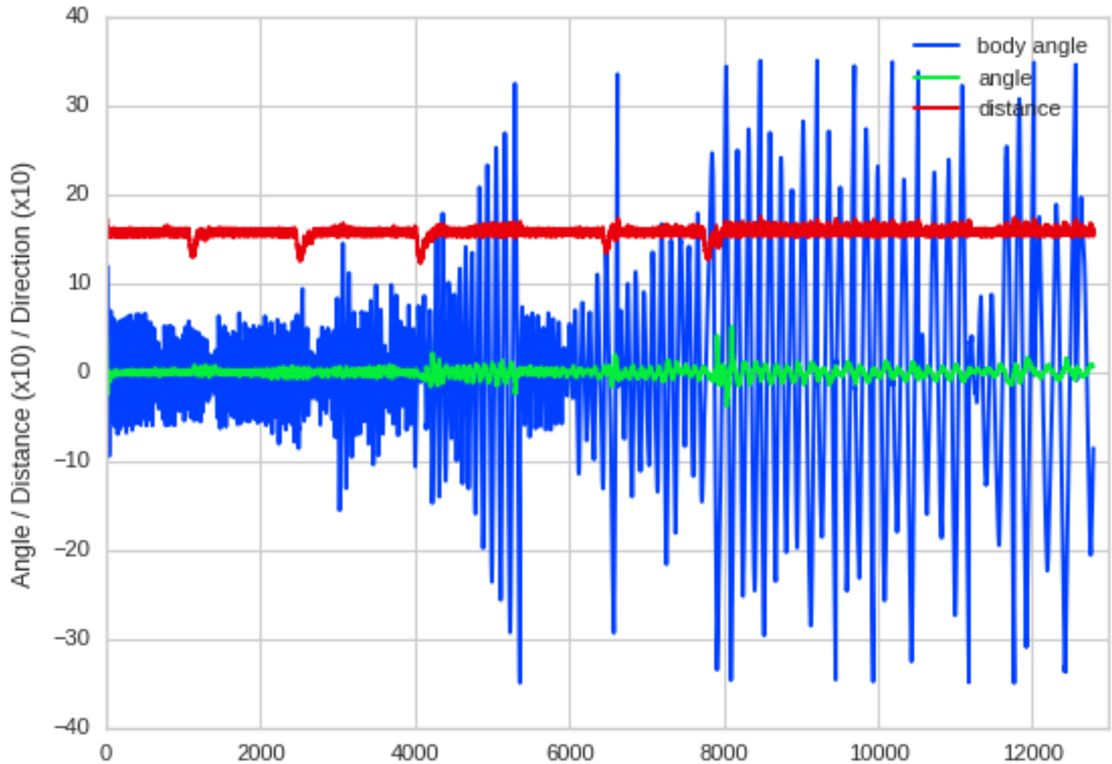
Training the Networks - Adding Head Control

Topology of the Network



Training the Networks - Adding Head Control

Performance of the Network



Training the Networks - Adding Head Control

Additional Topologies:

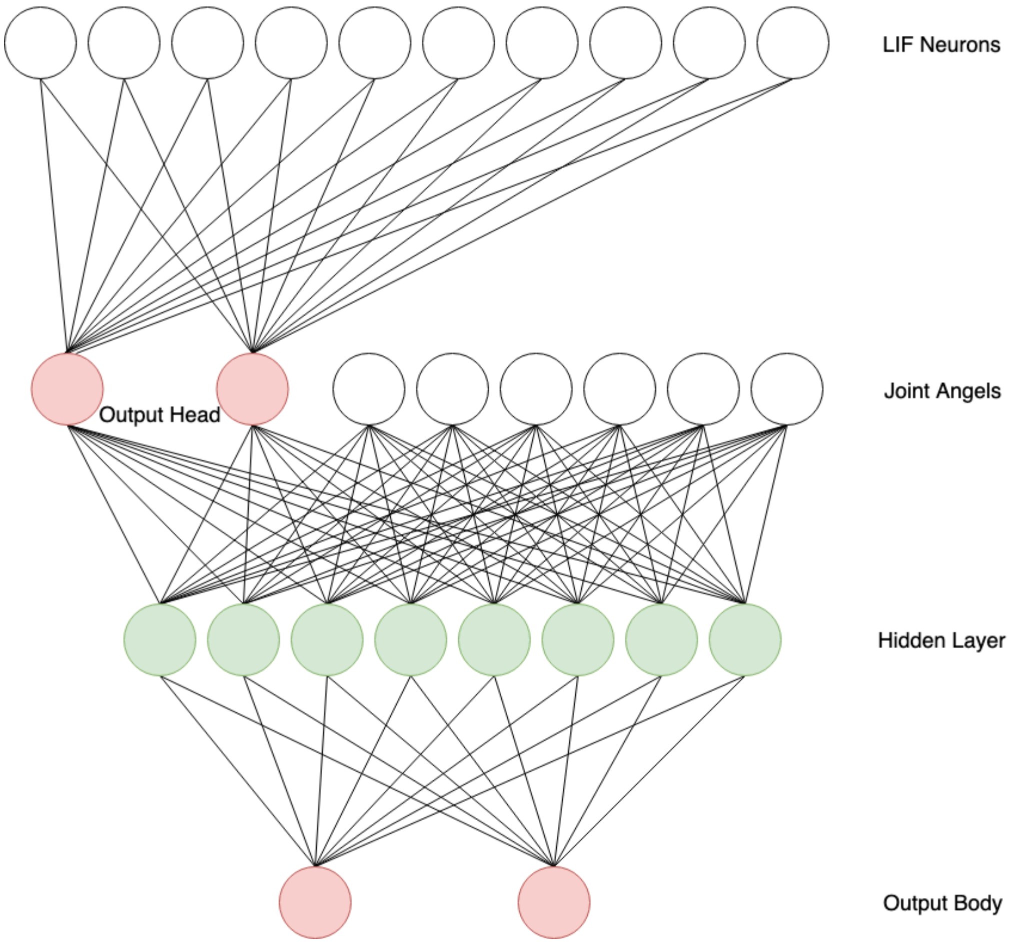
- DVS & Head Output
- Joint Angles & Head Output

Training Results:

- Weights do not stabilize
- Networks unable to solve the task

Training the Networks - Multi Layered

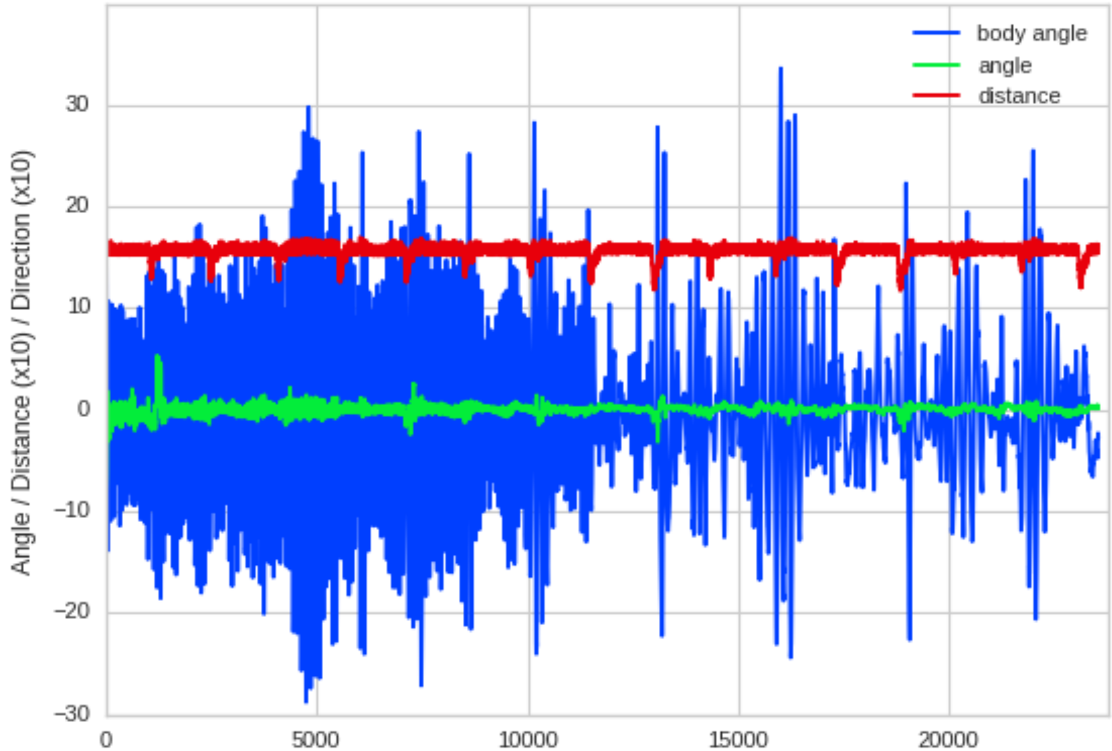
Topology of the Network



Chair of Robotics, Artificial Intelligence and Real-time Systems
Department of Informatics
Technical University of Munich

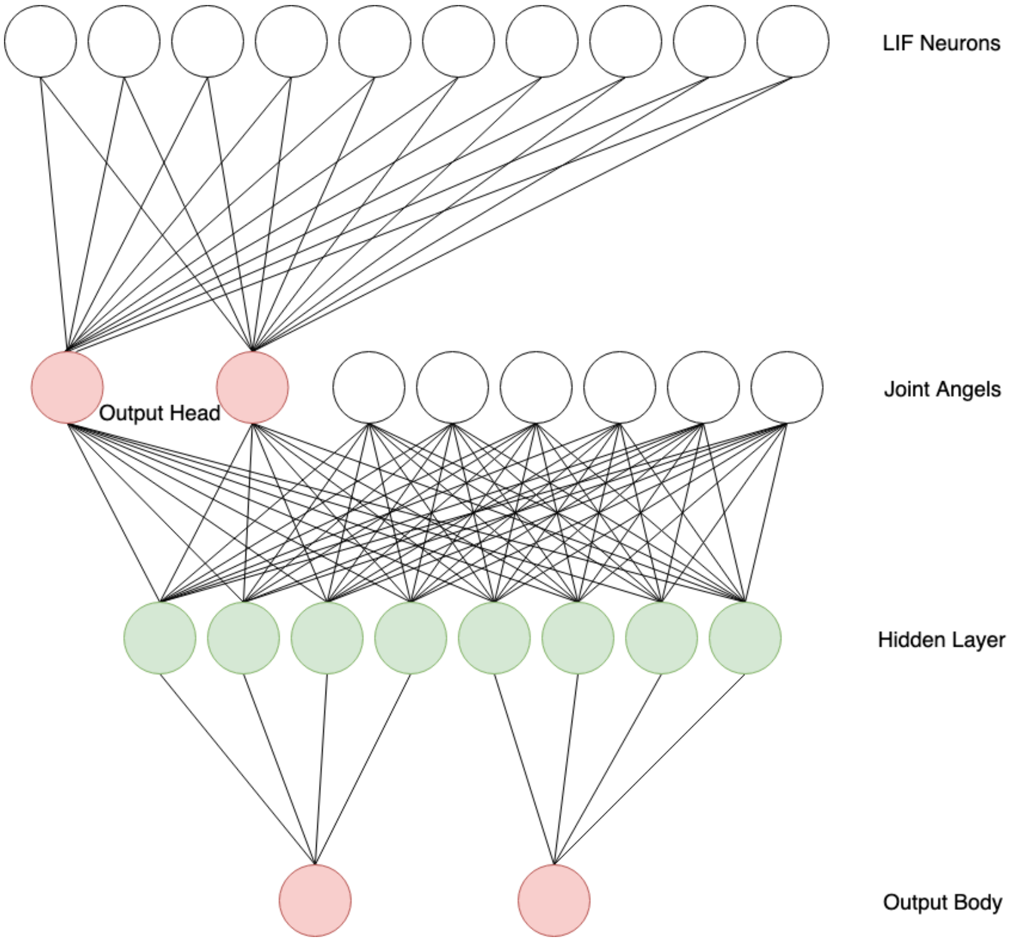
Training the Networks - Multi Layered

Performance of the Network



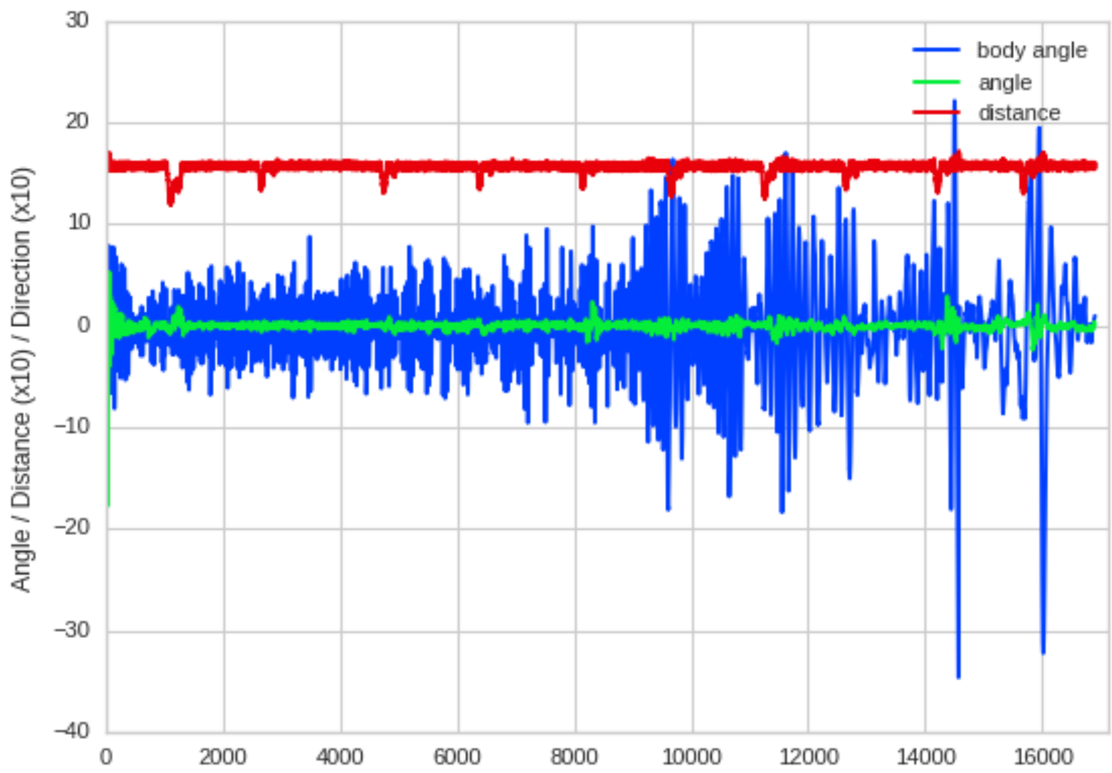
Training the Networks - Split Hidden Layer

Topology of the Network



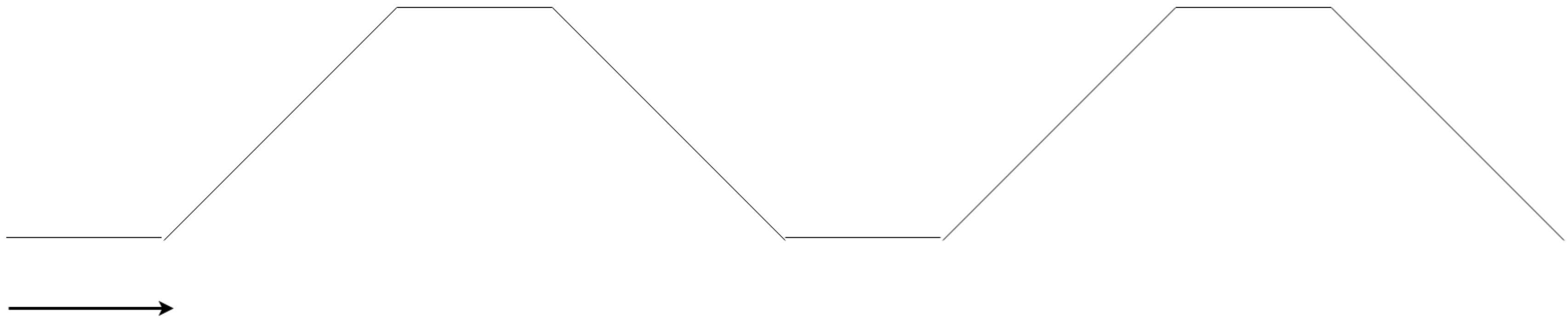
Training the Networks - Split Hidden Layer

Performance of the Network



Testing

New Movement Pattern of Target:



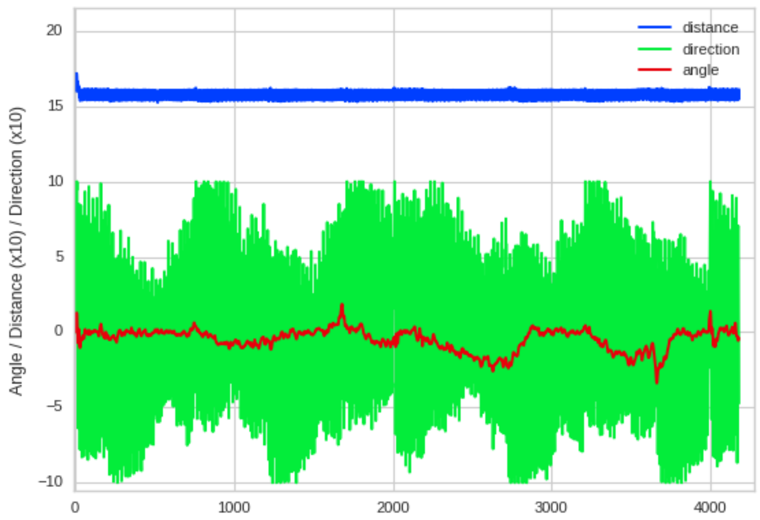
Chair of Robotics, Artificial Intelligence and Real-time Systems
Department of Informatics
Technical University of Munich

Testing – Basic Network

Solves without Fail

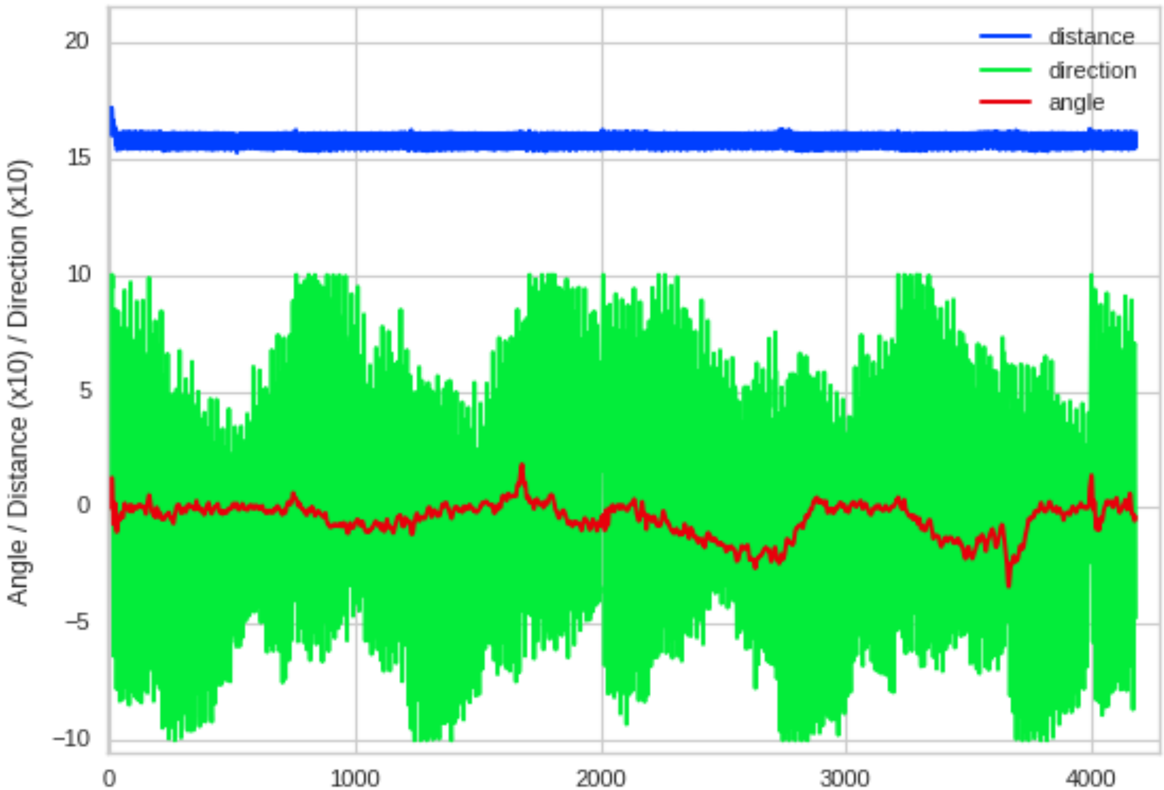
Good tracking of the Target

But high changes in the steering output



Testing – Basic Network

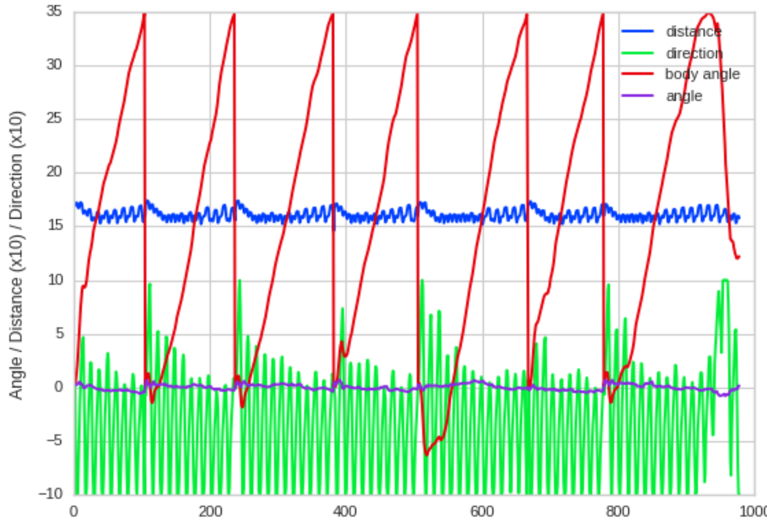
Performance of the Network



Testing – Multi Layered Network

Unable to solve

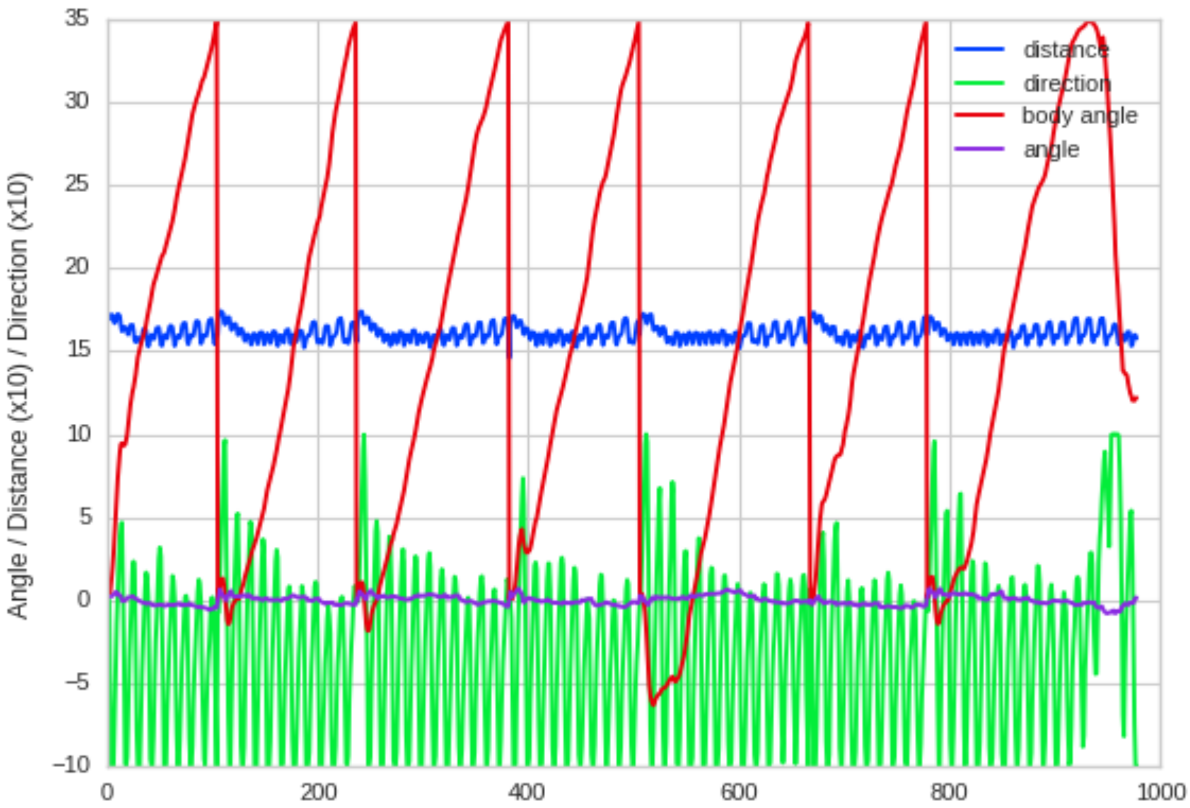
Drifting to the Right



Chair of Robotics, Artificial Intelligence and Real-time Systems
Department of Informatics
Technical University of Munich

Testing – Multi Layered Network

Performance of the Network



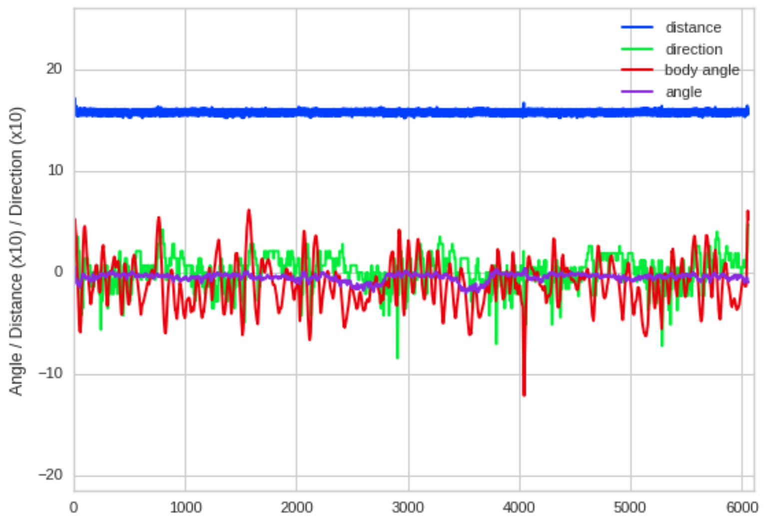
Testing - Split Hidden Layer

Solves without Fail

Very Good tracking of the Target

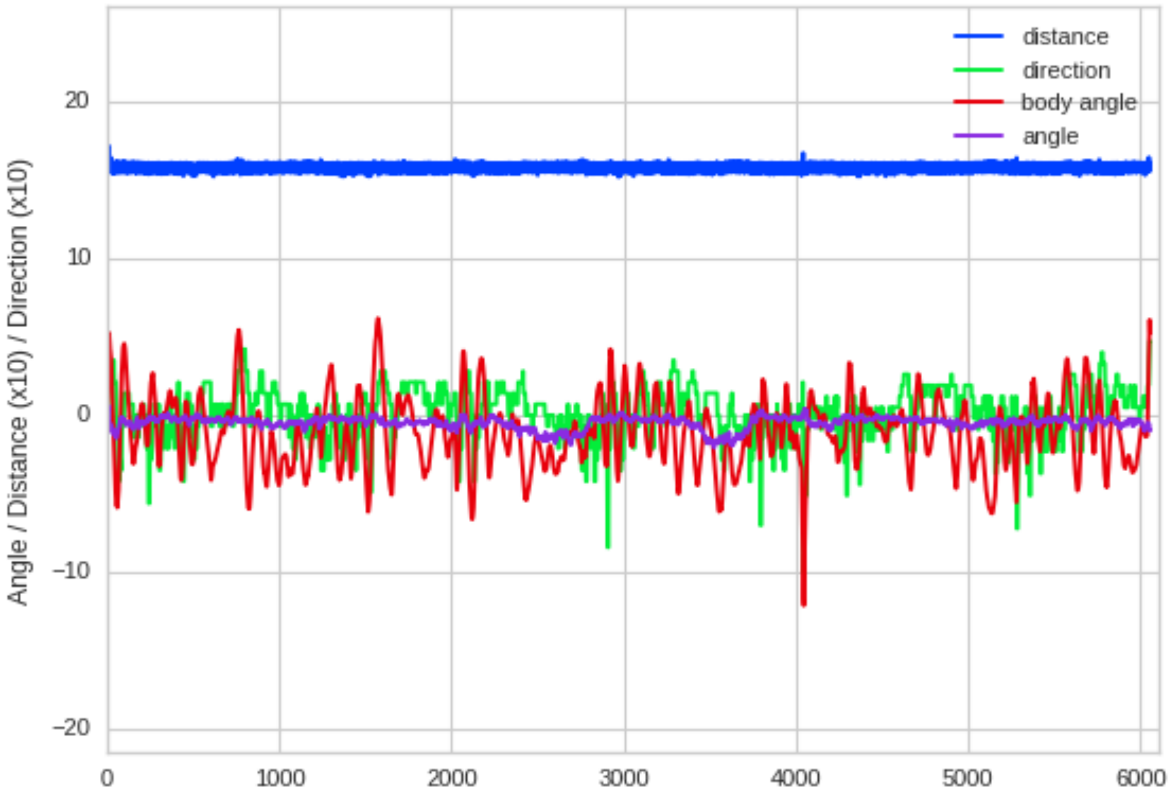
Body Angle to target comparable to Performance of Basic Setup

Control Signal much smoother



Testing - Split Hidden Layer

Performance of the Network



Conclusion

SNN capable of learning to Control Robot

Robust enough to handle Movement of Head

Single Layered Networks are not able to simultaneously control the direction of the Head and the Robot :

- Reduction of Information provided by DVS
- Information States not linear separable any more
- Multi Layered Network needed

Conclusion

Bad Result of Multi Layered Network:

- Network might not have reached a stable state in Training
- Small weight changes in the hidden layer can have a amplified effect on the output
- The topology of the multi layered Network can be reduced to the topology of the network with split hidden layer
- Faster training of topologies designed for the task

Future Work

- Replace Network Controlling Head with more Complex Topology to enable Object detection

- Duplicate DVS Layer to distinguish event Polarity

Thank you for your attention!

Questions?

Sources

[JIA18] Z. Jiang. *Development and Control of a Bio-snake Robot*. Oct. 2018.

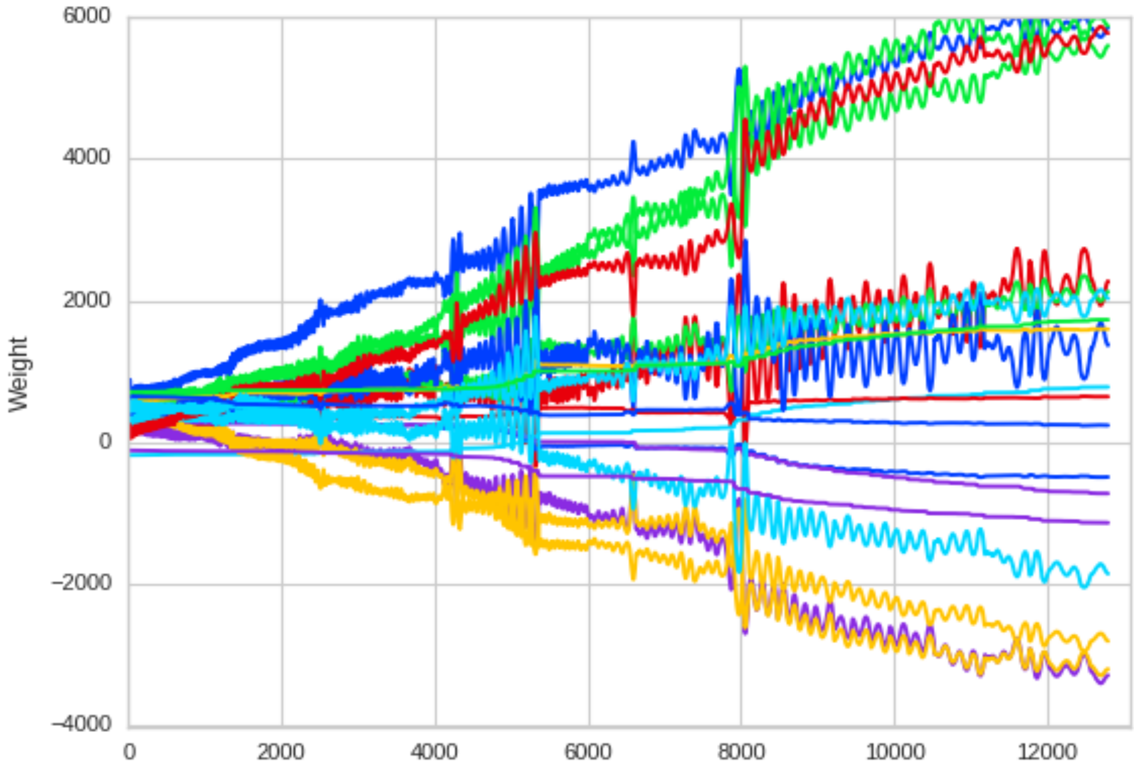
[L08] R. Legenstein, D. Pecevski, and W. Maass. "A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback." In: PLoS computational biology 4.10 (2008), e1000180.

[Kai+16] J. Kaiser, J. C. V. Tieck, C. Hubschneider, P. Wolf, M. Weber, M. Hoff, A. Friedrich, K. Wojtasik, A. Roennau, R. Kohlhaas, et al. "Towards a framework for end-to-end control of a simulated vehicle with spiking neural networks." In: *2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*. IEEE. 2016, pp. 127–134.

[Bin+19] Z. Bing, Z. Jiang, L. Cheng, C. Cai, K. Huang, A. Knoll, et al. "End to End Learning of a Multi-layered SNN Based on R-STDP for a Target Tracking Snake-like Robot, to be appear." In: 2019 IEEE International Conference on Robotics and Automation (ICRA). 2019.

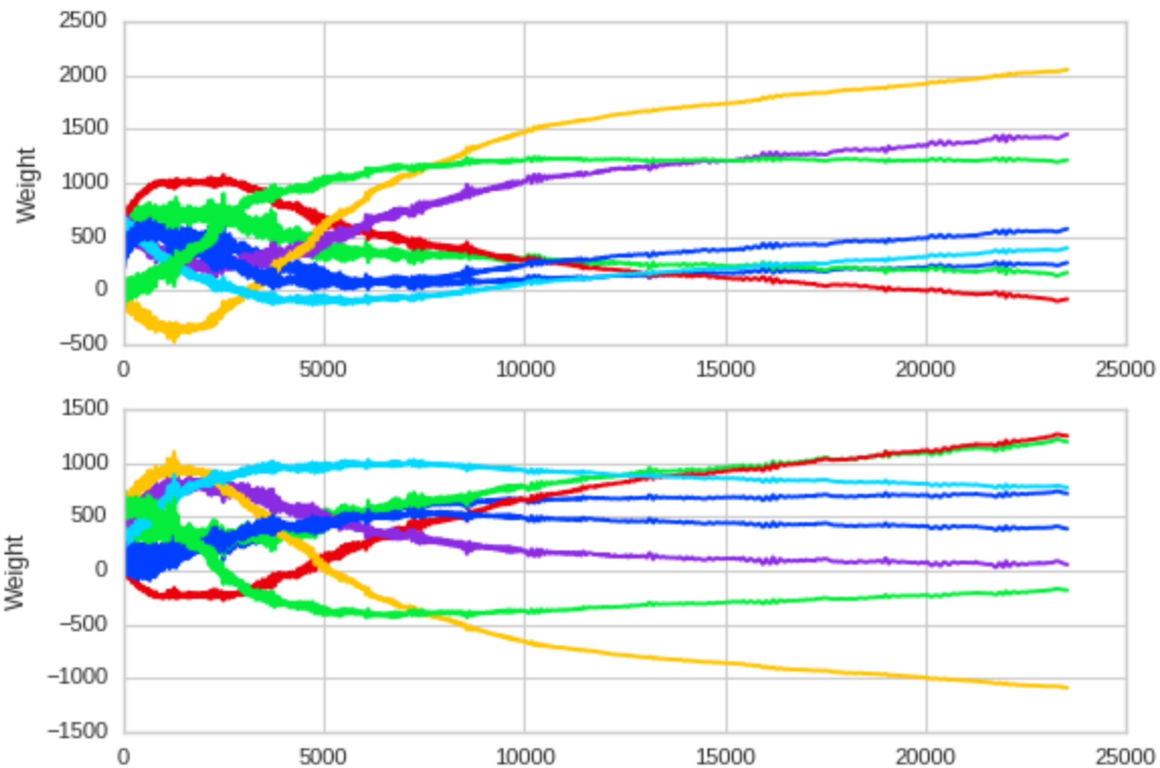
Training the Networks - Adding Head Control

Plot of Weights in the Network



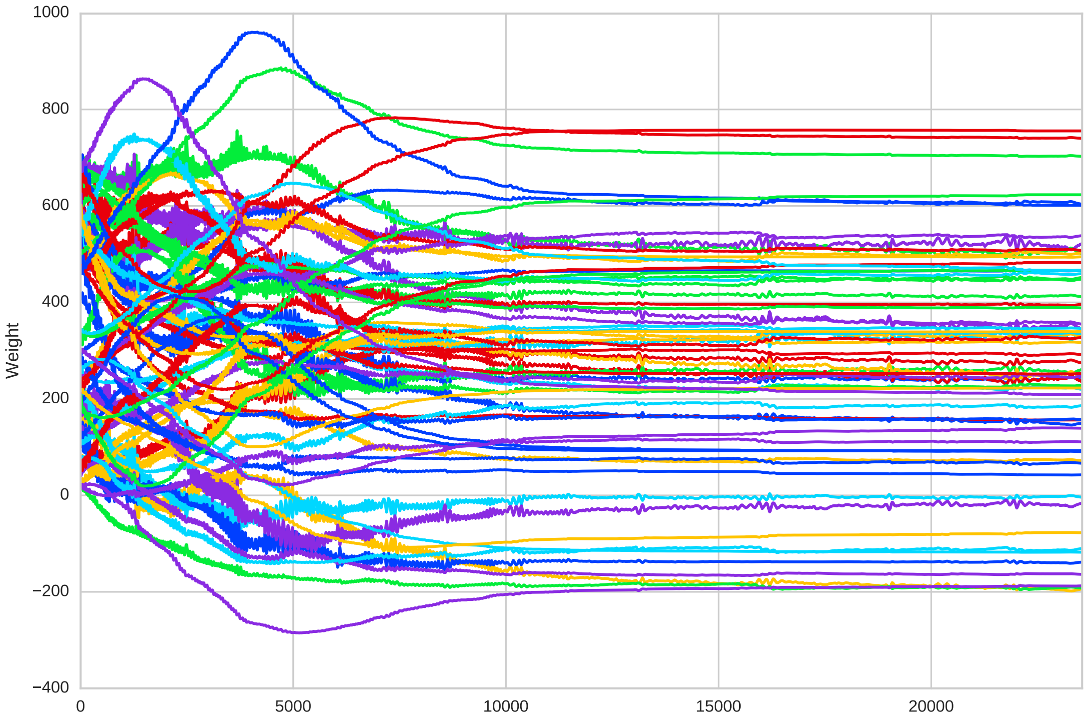
Training the Networks - Multi Layered

Plot of Weights in the Network



Training the Networks - Multi Layered

Plot of Weights in the Network



Training the Networks - Split Hidden Layer

Plot of Weights in the Network