Hierarchical architectures for spiking Winner-Take-All networks

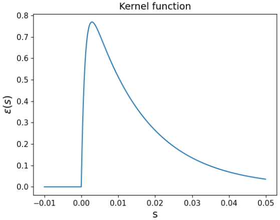
Christoph Rieger,
Supervisor: Univ.-Prof. Dipl.-Ing. Dr.techn. Robert Legenstein

10.01.2025

Outline

Biological background

- Spiking neural networks
  - Resemble biological neural networks closely
  - Generate and propagate neural spikes
- Winner-Take-All networks
- Probabilistic brain
- Synaptic plasticity



Biological background

- Networks are organized in hierarchical structure
- Feedback used for attention / biased competition
- Lee and Mumford found that feedback could let neurons see illusory contour

Kanizsa square, Source: Lee and Mumford

Source: Lee T.S., Mumford D. (2003), “Hierarchical Bayesian inference in the visual cortex.”, In: J Opt Soc Am A Opt Image Sci Vis

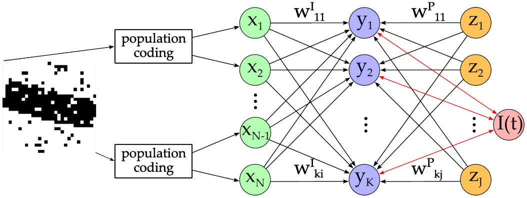
Theoretical background

- Bayesian inference gives the probability of an hypothesis given related evidence
- $P(Y = k|X, Z) = \frac{P(X|Y=k)P(Y=k|Z)}{\Sigma_{k'} P(X|Y=k')P(Y=k'|Z)}$
- Model of Nessler et al. expanded by prior neuron layer
  - Proved mathematically that expansion is valid
- Nessler et al. claimed that synaptic input weights converge towards the log of likelihood, $w^l_{ki} = log(P(x_i = 1|Y = k))$

Source: Nessler et al. (2013), “Bayesian Computation Emerges in Generic Cortical Microcircuits through Spike-Timing-Dependent Plasticity.”, In: PLOS Computational Biology 9.4

The network

- $u_k(t) = \sum_{i=1}^{N} w^I_{ki} \cdot x_i(t) + \sum_{j=1}^{J} w^P_{kj} \cdot z_j(t)$
- $p(y_k \text{ fires at time t}) \propto e^{u_k(t) - I(t)}$
- $w^I_{ki} = log(P(x_i = 1 | Y = k))$
- $w^P_{kj} = log(P(Y = k | Z = j))$



Goals

- Increase the understanding of the network model
- Simulate feedback found in the visual cortex
- Show connection between Bayesian inference and network model

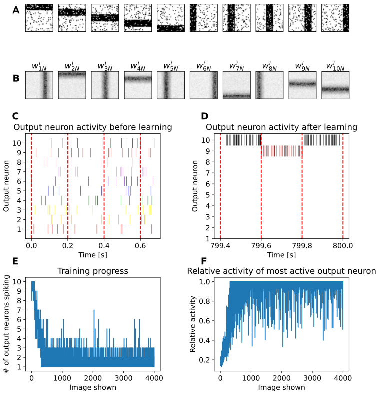
Methodology

- Simulation was performed in Python
- Simulation step size was 1 ms
- Pixels of input images and the prior had a noise level of 10%
- Kullback-Leibler divergence was chosen to evaluate performance of model
  - Compared firing rates of output neurons to analytical Bayesian posterior

Ambiguous visual stimuli 1

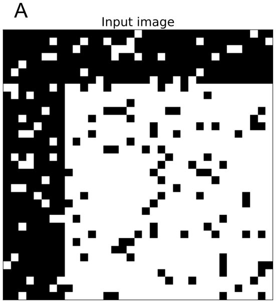
- Network learned to group horizontal and vertical bars into 10 groups
- After training ambiguous images with 1 horizontal and 1 vertical bar were shown
- Network was able to focus on individual bars, due to prior neurons

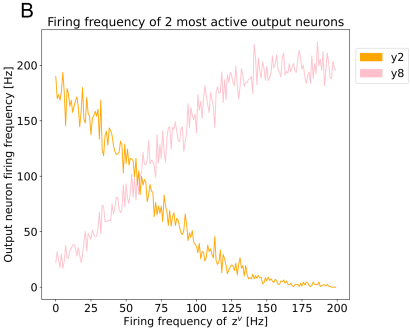
Ambiguous visual stimuli 2



Training plot

Ambiguous visual stimuli 3



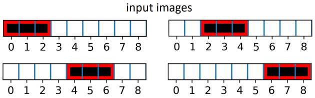


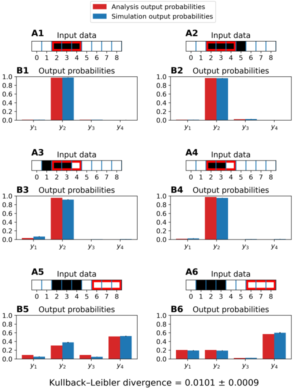
Variable prior activity

Analysis and simulation of the network 1

- Usage of smaller 1-D images to make network easier to analyse
- Mathematical derivation of Bayesian likelihood, prior and posterior
- Derived synaptic weights from Bayesian likelihood and prior
- Simulated network with those weights and fitted hyperparameters
- Compared Bayesian posterior to output of the simulation

Analysis and simulation of the network 2

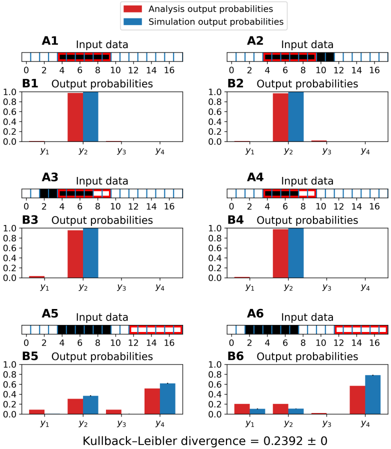




Transferability of hyperparameters 1

- Network was simulated with same hyperparameters of smaller network, to check if they are applicable to any network size
- Input size and prior neuron firing rate was doubled

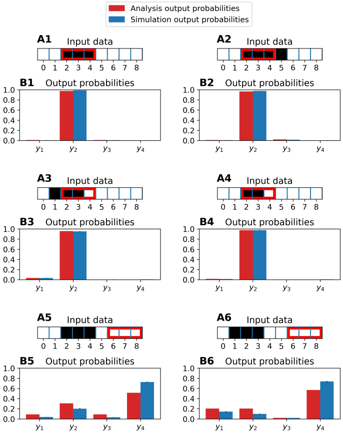
Transferability of hyperparameters 2



Training with predetermined hyperparameters 1

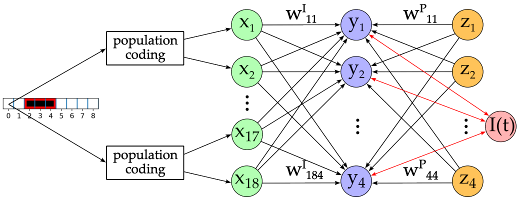
- Determined hyperparameters were used to train weights
- Trained weights were compared to analytically determined weights

Training with predetermined hyperparameters 2



Kullback–Leibler divergence = 0.0342 ± 0.0016

Training with predetermined hyperparameters 3



A $e^{\text{learned input weights}}$

input neurons (active for black pixels); rows: output neurons

0.52	0.49	0.53	0.06	0.1	0.08	0.12	0.11	0.06
0.09	0.07	0.49	0.49	0.51	0.09	0.09	0.09	0.06
0.07	0.08	0.08	0.06	0.51	0.5	0.52	0.07	0.1
0.11	0.1	0.12	0.07	0.09	0.06	0.53	0.46	0.51

B $e^{\text{calculated input weights}}$

input neurons (active for black pixels); rows: output neurons

0.9	0.9	0.9	0.1	0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.9	0.9	0.9	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.1	0.9	0.9	0.9	0.1	0.1
0.1	0.1	0.1	0.1	0.1	0.1	0.9	0.9	0.9

C $e^{\text{learned prior weights}}$

prior neurons; rows: output neurons

1.14	0.02	0.02	0.02
0.03	1.17	0.03	0.01
0.02	0.02	1.15	0.02
0.02	0.03	0.03	1.17

D $e^{\text{calculated prior weights}}$

prior neurons; rows: output neurons

0.9	0.03	0.03	0.03
0.03	0.9	0.03	0.03
0.03	0.03	0.9	0.03
0.03	0.03	0.03	0.9

Results

- Connection between model and Bayesian inference was shown
  - Network outputs spikes according to Bayesian posterior
  - Trained weights converge towards the log of their respective probabilities
- Importance of neural feedback was shown for
  - Attention / Ambiguity resolution
  - Illusory contour effect

Results

- Optimal hyperparameters are dependent on network size
- Training process could not achieve perfectly trained weights

Conclusion

- Thesis provided insight to hierarchical spiking Winner-Take-All network model
- Showed that the network model can simulate effects like attention and changing beliefs through feedback
- Provided ideas on how to further analyse and improve the model

Sources

- Lee T.S., Mumford D. (July 2003). “Hierarchical Bayesian inference in the visual cortex.” In: J Opt Soc Am A Opt Image Sci Vis. DOI: doi:10.1364/josaa.20.001434
- Nessler, Bernhard et al. (Apr. 2013). “Bayesian Computation Emerges in Generic Cortical Microcircuits through Spike-Timing-Dependent Plasticity.” In: PLOS Computational Biology 9.4, pp. 1–30. doi: 10.1371/journal. pcbi.1003037. url: https://doi.org/10.1371/journal.pcbi.1003037