Slowness: An Objective for Spike-Timing-Dependent Plasticity?

Henning Sprekeler, Christian Michaelis and Laurenz Wiskott

Stefan Klampfl
Computational Intelligence Seminar C, WS 2006/07
January 23, 2007
Outline

Motivation

Slow Feature Analysis (SFA)

Continuous model neuron
- Further definitions
- Reformulation of slowness objective
- Hebbian learning on filtered signals
- Relation to other learning rules

Linear Poisson neuron
- STDP can perform SFA
- Learning windows
- Interpretation of learning windows

Summary
Motivation

- The generation of invariant representations is a major challenge for the neural system of an organism.
- Invariances found in the brain are at least partially the product of learning.
- One powerful principle for the learning of invariances is temporal stability or slowness: Slower signals are more likely to carry information about invariant features.
- Slow Feature Analysis (SFA) (Wiskott and Sejnowski, 2002) is an unsupervised learning algorithm based on this slowness principle.
Motivation

- SFA has been successfully applied to the learning of translation, rotation and other invariances in a model of the visual system (Wiskott and Sejnowski, 2002)
- Applied to natural image sequences SFA reproduces a wide range of features of complex cells in primary visual cortex (Berkes and Wiskott, 2003)
- On an abstract level SFA seems to capture an important aspect of cortical information processing, but as a technical algorithm it is biologically rather implausible
- This work shows how this kind of computation can be realized by a spiking neuron
Slow Feature Analysis (SFA)

- Given a multidimensional input signal $x(t)$ and a function space $\mathcal{F}$, SFA finds the input-output function $g \in \mathcal{F}$ which generates the most slowly varying output signal $y = g(x(t))$
- Note that $g(x(t))$ is an instantaneous function (i.e., no low-pass filtering)
- Measure of slowness: variance of the time derivative $\langle \dot{y}^2(t) \rangle$
- Minimize $\langle \dot{y}^2(t) \rangle$ under the constraints $\langle y(t) \rangle = 0$ and $\langle y^2(t) \rangle = 1$ (to avoid trivial constant solution)
- Multiple slow features can be extracted using the decorrelation constraint $\langle y_i(t)y_j(t) \rangle = 0$ for $i \neq j$
Consider the special case

- where the input $\mathbf{x}$ is whitened (i.e., $\langle \mathbf{x} \rangle = 0$ and $\langle \mathbf{x} \mathbf{x}^T \rangle = \mathbf{I}$)
- $\mathcal{F}$ is the set of linear functions $y = g(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$

In this case the problem reduces to minimizing

$$\langle \dot{y}^2 \rangle = \mathbf{w}^T \langle \dot{\mathbf{x}} \dot{\mathbf{x}}^T \rangle \mathbf{w}$$

under the constraints

$$\langle \mathbf{y} \rangle = \mathbf{w}^T \langle \mathbf{x} \rangle = 0$$

$$\langle \mathbf{y}^2 \rangle = \mathbf{w}^T \langle \mathbf{x} \mathbf{x}^T \rangle \mathbf{w} = \mathbf{w}^T \mathbf{w} = 1$$

The solution is the normed eigenvector of the matrix $\langle \dot{\mathbf{x}} \dot{\mathbf{x}}^T \rangle$ that corresponds to the smallest eigenvalue
Continuous model neuron

Linear continuous neuron model

\[ a^{\text{out}}(t) = \sum_{i=1}^{n} w_i a^{\text{in}}_i(t) \]

Assume the inputs \( a^{\text{in}}_i(t) \) are approximately whitened on a sufficiently large interval \([t_a; t_b] \):

\[
\int_{t_a}^{t_b} a^{\text{in}}_i(t) dt \approx 0 \quad \text{(zero mean)}
\]

\[
\int_{t_a}^{t_b} (a^{\text{in}}_i(t))^2 dt \approx 1 \quad \text{(unit variance)}
\]

\[
\int_{t_a}^{t_b} a^{\text{in}}_i(t) a^{\text{in}}_j(t) dt \approx 0 \quad \text{(decorrelation)}
\]

can be achieved by a normalization and decorrelation step.
Continuous model neuron

Assume the output signal \( a^{out}(t) = \sum_{i=1}^{n} w_i a^{in}_i(t) \) is normalized to unit variance

\[
\int_{t_a}^{t_b} a^{out}(t) dt \approx 0 \quad \text{(zero mean)}
\]

\[
\int_{t_a}^{t_b} (a^{out}(t))^2 dt = \sum_{i=1}^{n} w_i^2 \approx 1 \quad \text{(unit variance)}
\]

⇒ weight vector is normalized to length 1
Further definitions

- Convolution and cross-correlation of two functions $f(t)$ and $g(t)$:

  \[ [f \circ g](t) := \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau \]

  \[ [f \star g](t) := \int_{-\infty}^{\infty} f(\tau)g(t + \tau)d\tau \]

- Windowed signals:

  \[ \hat{s}(t) := \begin{cases} s(t) & \text{if } t \in [t_a; t_b] \\ 0 & \text{otherwise} \end{cases} \]

  \[ \Rightarrow \int_{t_a}^{t_b} s(t)dt = \int_{-\infty}^{\infty} \hat{s}(t)dt \]
Further definitions

- Fourier transform $\mathcal{F}_s(\nu)$ and power spectrum $\mathcal{P}_s(\nu)$ of a signal $s(t)$:

  Fourier transform: 
  \[ \mathcal{F}_s(\nu) = \int_{-\infty}^{\infty} s(t) e^{-2\pi i \nu t} dt \]

  \[ s(t) = \int_{-\infty}^{\infty} \mathcal{F}_s(\nu) e^{2\pi i \nu t} d\nu \]

  Power spectrum: 
  \[ \mathcal{P}_s(\nu) = \mathcal{F}_s(\nu) \overline{\mathcal{F}_s(\nu)} \]

  Assume that input signals do not have a significant power above $\nu_{\text{max}}$
Reformulation of slowness objective

- SFA minimizes variance of time derivative $\int \dot{a}^{out}(t)^2 dt$
- Time derivative is difficult to extract for spiking neurons
- Replace the time derivative by a low-pass filtering:
  
  $$\int_{-\infty}^{\infty} \dot{a}^{out}(t)^2 dt = \int_{-\infty}^{\infty} P_{\dot{a}^{out}}(\nu) d\nu = 4\pi^2 \int_{-\infty}^{\infty} \nu^2 P_{a^{out}}(\nu) d\nu$$

  $\Leftrightarrow$ maximize

  $$\int_{-\infty}^{\infty} (\nu^2_{max} - \nu^2) P_{a^{out}}(\nu) d\nu = \int_{-\infty}^{\infty} \max(0, \nu^2_{max} - \nu^2) P_{a^{out}}(\nu) d\nu$$

  $$=: P_{SFA}(\nu)$$

  $$= \int_{-\infty}^{\infty} (\lfloor f_{SFA} \circ a^{out} \rfloor(t))^2 dt$$

- Either minimize the variance of the time derivative or maximize the variance of a low-pass filtered output signal
Reformulation of slowness objective

- **whitened input**
- **time derivative**
- **low-pass filter**
- **choose direction of minimal variance**
- **choose direction of maximal variance**
- **power spectrum of time derivative**
- **power spectrum of low-pass filter**
Hebbian learning on filtered signals

- Standard Hebbian learning maximizes the variance of the output signal
- Apply Hebbian learning to the filtered input and output signals

\[
[f_{SFA} \circ a^{out}](t) = \left[ f_{SFA} \circ \sum_{i=1}^{n} w_i a_i^{in} \right](t) = \sum_{i=1}^{n} w_i [f_{SFA} \circ a_i^{in}](t)
\]

- Filtered Hebbian learning rule:

\[
\dot{w}_i = \gamma [f_{SFA} \circ a_i^{in}](t) f_{SFA} \circ a^{out}](t)
\]

- optimizes slowness under the constraint of unit variance and normalized weights
Hebbian learning on filtered signals

neural activity
processing

learning

low-pass

\[ \Delta w \]
in x out

low-pass
Hebbian learning on filtered signals

- Use arbitrary filters $f^{in}$ and $f^{out}$

$$\Delta w_i := \int_{t_a}^{t_b} \dot{w}_i \, dt$$

$$= \gamma \int_{t_a}^{t_b} [f^{in} \circ a^{in}_i](t)[f^{out} \circ a^{out}](t) \, dt$$

$$\approx \gamma \int_{-\infty}^{\infty} [f^{in} \circ \hat{a}^{in}_i](t)[f^{out} \circ \hat{a}^{out}](t) \, dt$$

$$= \gamma \int_{-\infty}^{\infty} [f^{in} \ast f^{out}](t)[\hat{a}^{out} \ast \hat{a}^{in}_i](t) \, dt$$

as long as $\mathcal{F}_{f^{in} \ast f^{out}} = \mathcal{P}_{f^{SFA}}$
Relation to other learning rules

- Hebbian learning on low-pass filtered signals is the basis of several other models for unsupervised learning of invariances, e.g., trace rule (Földièk, 1991)

- output signal filtered with $f(t) = \theta(t)\gamma \exp(-\gamma t)$

\[
\text{trace rule: } \quad \max \int_{-\infty}^{\infty} F(\nu) P_{a_{out}}(\nu) d\nu \\
F(\nu) = \frac{\gamma}{\gamma^2 + (2\pi\nu)^2}
\]

\[
\text{SFA: } \quad \max \int_{-\infty}^{\infty} P_{SFA}(\nu) P_{a_{out}}(\nu) d\nu \\
P_{SFA}(\nu) = \max(0, \nu^2_{max} - \nu^2)
\]

- The difference lies in the choice of the low-pass filter

- SFA can be interpreted as a quadratic approximation of the trace rule
Linear Poisson neuron

- Input spike trains $S_{i}^{in}(t)$ are modeled as inhomogeneous Poisson processes with instantaneous rates $r_{i}^{in}(t)$, drawn from ensembles $E_{i}^{in}$:

$$r_{i}^{in}(t) := c_{i}^{in} + a_{i}^{in}(t) \quad \langle S_{i}^{in}(t) \rangle_{E_{i}^{in}} = r_{i}^{in}(t)$$

- Output spike train $S^{out}(t)$ is another inhomogeneous Poisson process, drawn from ensemble $E^{out}$ given a realization of input spike trains $S_{i}^{in}$:

$$m(t) := r_{0} + \sum_{i=1}^{n} w_{i}[\epsilon \circ S_{i}^{in}](t) \quad \langle S^{out}(t) \rangle_{E^{out}|\{S_{i}^{in}\}} = m(t)$$

$$r^{out}(t) := \langle S^{out}(t) \rangle_{E^{out},E^{in}} = c^{out} + [\epsilon \circ a^{out}](t)$$
STDP can perform SFA

- The STDP weight change can be modeled as

\[
\Delta w_i = \gamma \int_{t_a}^{t_b} W(t - t') S_{in}^i(t) S_{out}^{out}(t') dt dt'
\]

\[
\approx \int_{-\infty}^{\infty} W(t - t') \hat{S}_{in}^i(t) \hat{S}_{out}^{out}(t') dt dt'
\]

- Taking the ensemble average yields

\[
\langle \Delta w_i \rangle_{E_{in}, E_{out}} \approx \gamma \int_{-\infty}^{\infty} [W \circ \epsilon](t)[\hat{a}^{out} \star \hat{a}^{in}_i](t) dt
\]

- For the dynamics of the learning process the convolution of the learning window \( W \) with the EPSP \( \epsilon \) is relevant
STDP can perform SFA

- Compare equations

\[
\Delta w_i = \gamma \int_{-\infty}^{\infty} [f^{in} * f^{out}](t)[\hat{a}^{out} * \hat{a}^{in}_i](t)dt
\]

\[
\langle \Delta w_i \rangle_{E^{in}, E^{out}} \approx \gamma \int_{-\infty}^{\infty} [W \circ \epsilon](t)[\hat{a}^{out} * \hat{a}^{in}_i](t)dt
\]

- STDP generates the same weight distribution in an ensemble-averaged sense as the continuous model with a specific learning window

\[
[W \circ \epsilon](t) = [f^{in} * f^{out}](t) =: W_0(t)
\]

\[
\mathcal{F}_{W_0}(\nu) = \mathcal{F}_{f^{in} * f^{out}}(\nu) = \mathcal{P}_{f_{SFA}}(\nu)
\]

- Given \( \mathcal{P}_{f_{SFA}} \) and EPSP \( \epsilon \) the learning window \( W \) is uniquely determined
Learning windows

- Power spectrum $P_{f\textsc{sfa}}(\nu) = \max(0, \nu_{\text{max}}^2 - \nu^2) \overset{!}{=} \mathcal{F}W_0(\nu)$

- EPSP $\epsilon(t) = \theta(t)e^{-\frac{t}{\tau}}$

- Calculate learning window $W$ by inverse Fourier transform
  \[ (\mathcal{F}W_0(\nu) = \mathcal{F}W(\nu)\mathcal{F}\epsilon(\nu)) \]

\[
W(t) = \frac{d}{dt}W_0(t) + \frac{1}{\tau}W_0(t)
\]

  \begin{align*}
  &\text{antisymmetric} &\text{symmetric}
  \\
  \end{align*}

- $W(t)$ is a linear combination of a symmetric and an antisymmetric component

- $\tau \ll \frac{1}{\nu_{\text{max}}}$ symmetric part dominates

- $\tau \gg \frac{1}{\nu_{\text{max}}}$ antisymmetric part dominates
Learning windows

\[ \nu_{\text{max}} = \frac{1}{(40 \text{ms})} \]
Learning windows

- It is reasonable to choose $\tau$ (time constant of EPSP) and $1/\nu_{max}$ to have the same order of magnitude ($40\text{ms}$)

- If EPSP is short enough to resolve the fastest input components, the learning window is symmetric

- If EPSP is too long to fully resolve the temporal structure of the input, the learning window is antisymmetric

- Synapses are potentiated when the onset of an EPSP precedes a postsynaptic spike and depressed when this temporal order is reversed

- If $P_{f_{SFA}}$ is replaced by a smoothened Cauchy function (trace rule) the learning window has a double-exponential shape
Learning windows

\[ W_i = P_{SFA} \]

data from (Bi and Poo, 1998)
Interpretation of learning windows

Why are these learning windows optimal for learning slowness?

- **Case 1: Symmetric learning window**
  - EPSP is short, $r^{out}$ is instantaneous function of input rates
  - Synapses which transmit a spike in a certain time window around a postsynaptic spike get strengthened
  - equilibrates firing rates for neighboring times, favors temporally slow output signals

- **Case 2: Antisymmetric learning window**
  - EPSP is long, $r^{out}$ is generated by low-pass filtering of $a^{out}$
  - Deconvolution is needed to reconstruct $a^{out}$
  - Learning window has to “invert” the effect of the EPSP

- The learning window for long EPSPs is the temporal derivative of the learning window for short EPSPs

- The antisymmetry of the STDP learning window may not be a causality detector, but rather a mechanism for compensating intrinsic low-pass filters, such as the EPSP
Summary

- SFA is one algorithm making use of the slowness principle for learning invariances, but a way of implementing it with neural circuits is still missing.

- In this work it has been shown analytically that such an implementation is possible in both continuous and spiking neurons.

- For linear continuous model neurons, Hebbian learning on low-pass filtered versions of input and output signals learns the slowest direction in the input signals (similar to trace rule).

- For a spiking Poisson neuron, it is possible in an ensemble-averaged sense to reproduce the behavior of the continuous neuron by means of STDP.
Summary

- The presented model is not a complete implementation of SFA; additional steps are necessary
  - nonlinear expansion of the input space
  - whitening of the expanded input signals
  - normalizing the weights
- Spiking neuron model is simplified because it has only a linear relationship between input and output firing rate
- Simulations are necessary to verify the results and analyze them for more realistic neuron models
- These results close the gap between slowness as an abstract learning principle and biologically plausible STDP learning rules and offer a new interpretation of the standard STDP learning window
Slow feature analysis yields a rich repertoire of complex cell properties.

Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type.

Learning invariance from transformation sequences.

Slowness: An objective for spike-timing-dependent plasticity?

Wiskott, L. and Sejnowski, T. J. (2002).
Slow feature analysis: unsupervised learning of invariances.