Learning visual motion with recurrent neural networks

Marius Pachitariu

Gatsby Unit, UCL
adviser: Maneesh Sahani

Outline

Marr's three levels of analysis

Levels of analysis

- Computational
- Algorithmic / Representational
- Physical

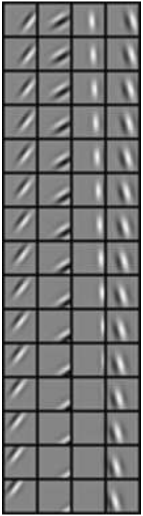
Sequential data types

- Movies
- Spike trains
- Language

Spatio-temporal filters

- Dominant in both visual neuroscience and computer vision.
- Caveats:
  - not real-time/requires copies of the past
    $\rightarrow$ bad for real-world systems, like the brain.

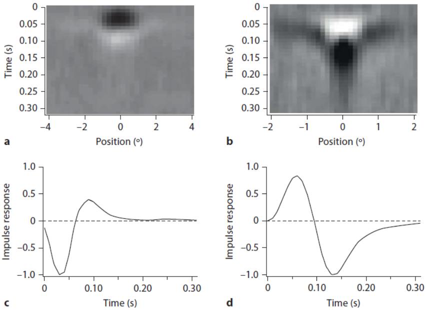
Spatio-temporal filters



- Dominant in both visual neuroscience and computer vision.
- Caveats:
  - not real-time/requires copies of the past
    $\rightarrow$ bad for real-world systems, like the brain.
  - too many parameters
    $\rightarrow$ bad for learning and generalization.
  - high computational complexity
    $\rightarrow$ bad with high-bandwidth data.

Neural candidates for ST filters

- lagged LGN cells (Mastronarde, 1987)

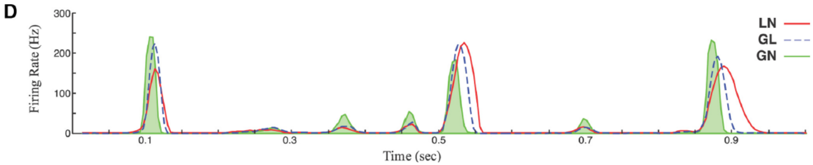


Neural candidates for ST filters

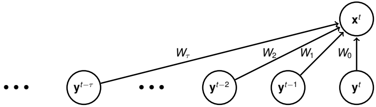
- lagged LGN cells (Mastronarde, 1987)
- but LGN is an information bottleneck

Neural candidates for ST filters

- lagged LGN cells (Mastronarde, 1987)
- but LGN is an information bottleneck
- but LGN responds precisely to natural movies (Butts et al, 2011)



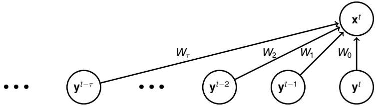
Compact parametrization of ST filters with an RNN



Spatiotemporal filtering

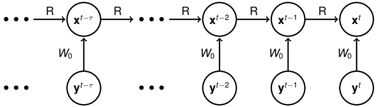
$$\mathbf{x}^t = \sum_{\tau=0}^{\infty} \mathbf{W}_\tau \; \mathbf{y}^{t-\tau}$$

Compact parametrization of ST filters with an RNN



Spatiotemporal filtering

$$\mathbf{x}^t = \sum_{\tau=0}^{\infty} \mathbf{W}_\tau \ \mathbf{y}^{t-\tau}$$



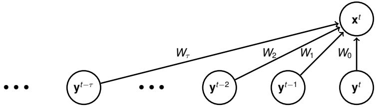
Recurrent neural network

$$\mathbf{W}_\tau = (\mathbf{R})^\tau \ \ \mathbf{W}_0$$

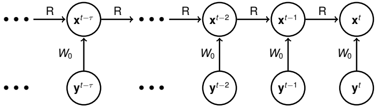
$$\mathbf{x}^t = \mathbf{W}_0 \ \mathbf{y}^t + \mathbf{R} \sum_{\tau=0}^{\infty} (\mathbf{R})^\tau \ \ \mathbf{W}_0 \ \mathbf{y}^{t-1-\tau}$$

$$\mathbf{x}^t = \mathbf{W}_0 \ \mathbf{y}^t + \mathbf{R} \ \mathbf{x}^{t-1}$$

Compact parametrization of ST filters with an RNN



Spatiotemporal filtering



Recurrent neural network

$$\mathbf{x}^t = \sum_{\tau=0}^{\infty} \mathbf{W}_\tau \; \mathbf{y}^{t-\tau}$$

$$\mathbf{W}_\tau = (\mathbf{R})^\tau \; \mathbf{W}_0$$

$$\mathbf{x}^t = \mathbf{W}_0 \; \mathbf{y}^t + \mathbf{R} \sum_{\tau=0}^{\infty} (\mathbf{R})^\tau \; \mathbf{W}_0 \; \mathbf{y}^{t-1-\tau}$$

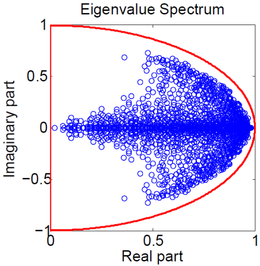
$$\mathbf{x}^t = \mathbf{W}_0 \; \mathbf{y}^t + \mathbf{R} \; \mathbf{x}^{t-1}$$

As a simple example, we fit **R** to a diverse bank of spatiotemporal filters.

Reconstructions of the ST filters are good

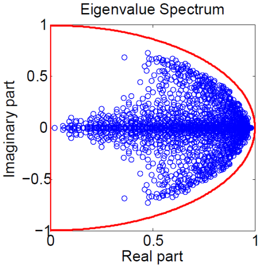
What do the connections look like?

- Spectrum of R



What do the connections look like?

- Spectrum of R



- Strongest connections to a given neuron (animation).

Computational complexity and memory requirements

- l_x by l_y by n_t filters (12 by 12 by 30)
- N (1600) ST filters
- Feedforward flops $= 2N \; l_x^2 \; l_y^2 \; n_t$

Computational complexity and memory requirements

- l_x by l_y by n_t filters (12 by 12 by 30)
- N (1600) ST filters
- Feedforward flops $= 2N\ \ l_x^2\ \ l_y^2\ \ n_t$
- Recurrent flops $= 2N^2 + 2N\ \ l_x^2\ \ l_y^2$

Computational complexity and memory requirements

- l_x by l_y by n_t filters (12 by 12 by 30)
- N (1600) ST filters
- Feedforward flops $= 2N \;\; l_x^2 \;\; l_y^2 \;\; n_t$
- Recurrent flops $= 2N^2 + 2N \;\; l_x^2 \;\; l_y^2$
- 5 % non-zero connections in $\mathbf{R}$.
- Recurrent flops $= 2 \cdot 0.05 \cdot \;\; N^2 + 2N \;\; l_x^2 \;\; l_y^2$

Computational complexity and memory requirements

- l_x by l_y by n_t filters (12 by 12 by 30)
- N (1600) ST filters
- Feedforward flops $= 2N \;\; l_x^2 \;\; l_y^2 \;\; n_t$

- Recurrent flops $= 2N^2 + 2N \;\; l_x^2 \;\; l_y^2$

- 5 % non-zero connections in $\mathbf{R}$.
- Recurrent flops $= 2 \cdot 0.05 \cdot \;\; N^2 + 2N \;\; l_x^2 \;\; l_y^2$

- Recurrent flops $<$ Feedforward flops

Advantages of recurrent neural networks

- the brain already has the hardware

Advantages of recurrent neural networks

- the brain already has the hardware
- do not require copies of the past
  $\rightarrow$ less memory usage
  $\rightarrow$ the brain has short timescales + bottleneck in LGN
  $\rightarrow$ no evidence for true delay lines in cortex

Advantages of recurrent neural networks

- the brain already has the hardware
- do not require copies of the past
  $\rightarrow$ less memory usage
  $\rightarrow$ the brain has short timescales + bottleneck in LGN
  $\rightarrow$ no evidence for true delay lines in cortex
- fewer parameters
  $\rightarrow$ important for learning and generalization

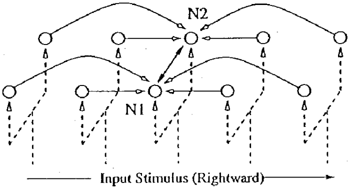
Advantages of recurrent neural networks

- the brain already has the hardware
- do not require copies of the past
  $\rightarrow$ less memory usage
  $\rightarrow$ the brain has short timescales + bottleneck in LGN
  $\rightarrow$ no evidence for true delay lines in cortex
- fewer parameters
  $\rightarrow$ important for learning and generalization
- reduced computational complexity
  $\rightarrow$ good for high bandwidth data

Advantages of recurrent neural networks

- the brain already has the hardware
- do not require copies of the past
  $\rightarrow$ less memory usage
  $\rightarrow$ the brain has short timescales + bottleneck in LGN
  $\rightarrow$ no evidence for true delay lines in cortex
- fewer parameters
  $\rightarrow$ important for learning and generalization
- reduced computational complexity
  $\rightarrow$ good for high bandwidth data
- can integrate over long time periods
  $\rightarrow$ natural visual motion can be slow and noisy

Neural sequence learning via STDP (toy model)



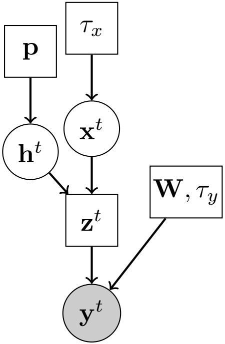
Rao & Sejnowski, NIPS 2000

Spike and Slab Sparse Coding

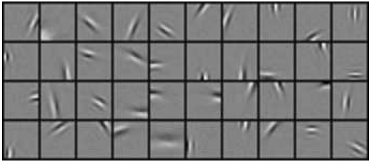
- Olshausen&Millman 2000, Rehn&Sommer 2007, Goodfellow&al 2012

Spike and Slab Sparse Coding

- Olshausen&Millman 2000, Rehn&Sommer 2007, Goodfellow&al 2012



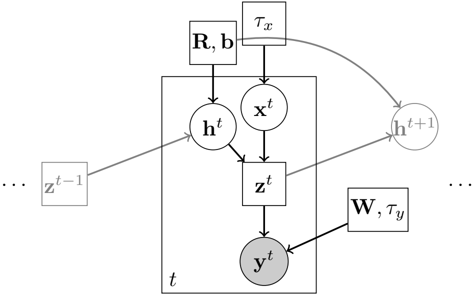
$$\mathbf{h}_k^t = \text{Bernoulli}\left(\mathbf{p}_k\right)$$
$$\mathbf{x}^t = \mathcal{N}\left(0, \tau_x^2 \cdot \mathbf{I}\right)$$
$$\mathbf{z}^t = \mathbf{h}^t \circ \mathbf{x}^t$$
$$\mathbf{y}^t = \mathcal{N}\left(\mathbf{W} \cdot \mathbf{z}^t, \tau_y^2\right)$$



Spike and Slab Recurrent Neural Network

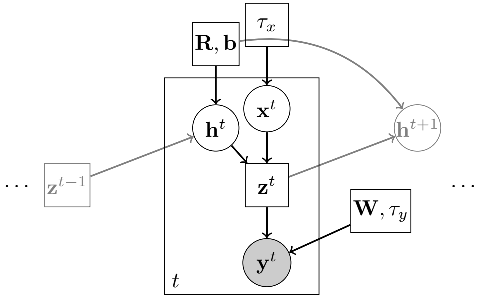
$$\mathbf{P}\left(\mathbf{h}^{t+1}|\mathbf{z}^{t}\right) = \sigma\left(\mathbf{R}\cdot\mathbf{z}^{t} + \mathbf{b}\right)$$

$$\sigma(x) = 1/(1+\exp(-x))$$



Spike and Slab Recurrent Neural Network

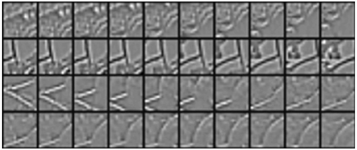
$$\mathbf{P}\left(\mathbf{h}^{t+1}|\mathbf{z}^{t}\right) = \sigma\left(\mathbf{R}\cdot\mathbf{z}^{t} + \mathbf{b}\right)$$
$$\sigma(x) = 1/(1+\exp(-x))$$



For approximate inference:

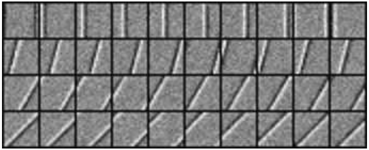
- Assuming we have set $\hat{\mathbf{x}}^t, \hat{\mathbf{h}}^t$ for $t = 1$ to T.
- At $T+1$ we only need to solve an SC problem.

Natural movies and artificial stimuli



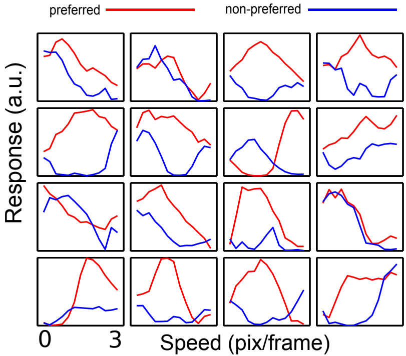
Training data (ZCA whitened)

Full single frame of training data



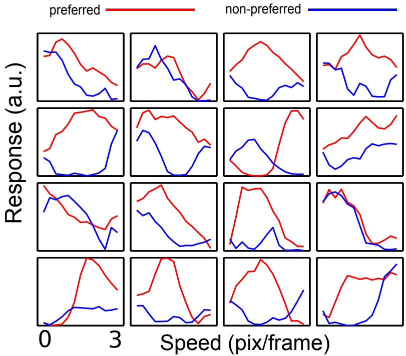
Test data (whitened)

Results - Speed tuning

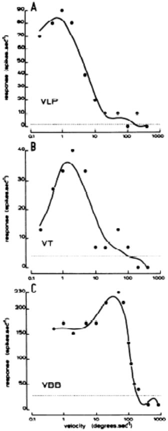


Rectified neural responses $\max(\mathbf{z}, 0)$ to drifting square gratings.

Results - Speed tuning

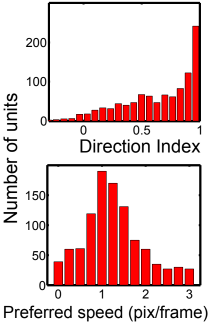


Rectified neural responses $\max(\mathbf{z}, 0)$ to drifting square gratings.



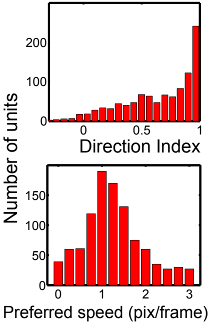
Orban et al, 1986

Results - Direction selectivity indices

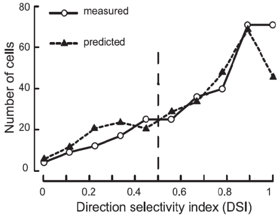


$$\mathrm{DI} = 1 - r_{\text{non-pref}}/r_{\text{pref}}$$

Results - Direction selectivity indices

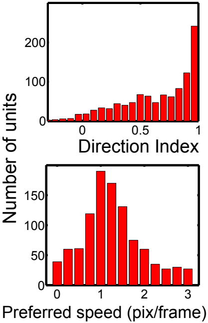


$$\mathrm{DI} = 1 - r_{\text{non-pref}} / r_{\text{pref}}$$

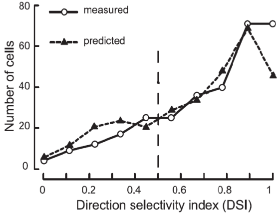


Peterson et al, 2004

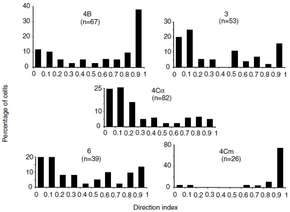
Results - Direction selectivity indices



$$\mathrm{DI} = 1 - r_{\text{non-pref}} / r_{\text{pref}}$$

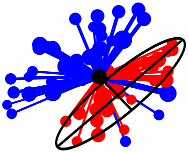


Peterson et al, 2004



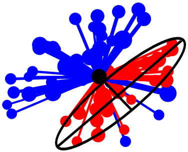
Gur et al, 2007

Results - connectomics in silico

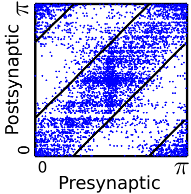


Largest outgoing connections of one unit

Results - connectomics in silico

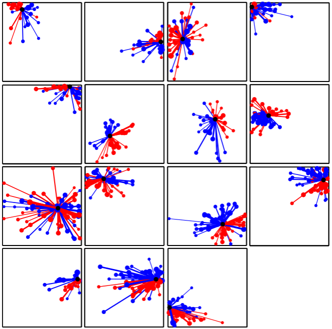


Largest outgoing connections of one unit



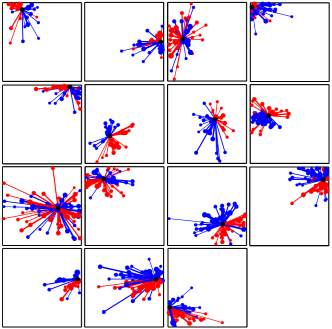
Connected units are co-oriented

Results - connectomics in silico

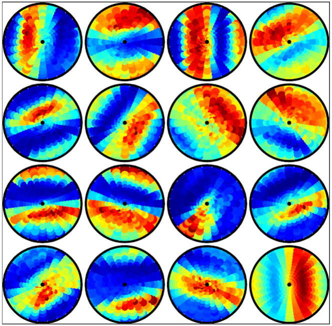


Outgoing connections of 15 randomly chosen DS units (and animation during learning)

Results - connectomics in silico

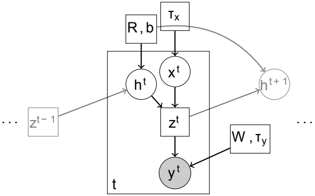


Outgoing connections of 15 randomly chosen DS units (and animation during learning)



Responses to small drifting Gabors - polar plots

Inference and learning



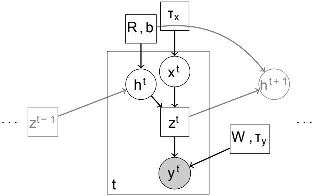
$$\mathcal{L}_{\text{ss-RNN}} = \sum_t \mathcal{L}^t_{\text{ss-RNN}}$$

$$\mathcal{L}^t_{\text{ss-RNN}} = \text{const} - \|\mathbf{y}^t - \mathbf{W}(\mathbf{x}^t \circ \mathbf{h}^t)\|^2/2\tau_y^2 - \|\mathbf{x}^t\|^2/2\tau_x^2 +$$

$$+ \sum_{j=1}^{N} h_j^t \log \sigma \left(\mathbf{R}\left(\mathbf{h}^{t-1} \circ \mathbf{x}^{t-1}\right) + \mathbf{b}\right)_j$$

$$+ \sum_{j=1}^{N} (1 - h_j^t) \log \left(1 - \sigma\left(\mathbf{R}\left(\mathbf{h}^{t-1} \circ \mathbf{x}^{t-1}\right) + \mathbf{b}\right)_j\right)$$

Inference and learning



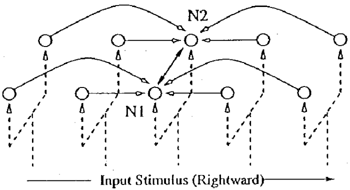
$$\mathcal{L}_{\text{ss-RNN}} = \sum_t \mathcal{L}^t_{\text{ss-RNN}}$$

$$\mathcal{L}^t_{\text{ss-RNN}} = \text{const} - \|\mathbf{y}^t - \mathbf{W}(\mathbf{x}^t \circ \mathbf{h}^t)\|^2/2\tau_y^2 - \|\mathbf{x}^t\|^2/2\tau_x^2 + \\ + \sum_{j=1}^{N} h_j^t \log \sigma\left(\mathbf{R}\left(\mathbf{h}^{t-1} \circ \mathbf{x}^{t-1}\right) + \mathbf{b}\right)_j \\ + \sum_{j=1}^{N} (1 - h_j^t) \log\left(1 - \sigma\left(\mathbf{R}\left(\mathbf{h}^{t-1} \circ \mathbf{x}^{t-1}\right) + \mathbf{b}\right)_j\right)$$

For approximate inference, we use *greedy filtering*:

- Assuming we have already set $\hat{\mathbf{x}}^t, \hat{\mathbf{h}}^t$ for $t = 1$ to T.
- At step $T + 1$ we only need to solve an sparse coding problem given by the slice $\mathcal{L}^{T+1}_{\text{ss-RNN}}$.
- We solve the SC problem with standard matching pursuit / coordinate descent methods.

Learning of ss-RNN



Rao & Sejnowski, NIPS 2000

Gradients for learning **R** are similar to the STDP learning rule used in Rao & Sejnowski, 2000.

$$\frac{\partial \mathcal{L}^t_{\text{ss-RNN}}}{\partial R_{jk}} = \left(h_k^{t-1} x_k^{t-1}\right) \cdot \left(h_j^t - \sigma\left(\mathbf{R}\left(\mathbf{h}^t \circ \mathbf{x}^t\right) + \mathbf{b}\right)_j\right).$$

DS is learned, OS is not?

- Orientation selectivity (OS) and ocular dominance (OD) do not require visual experience
- Visual deprivation has little impact on OS and OD

DS is learned, OS is not?

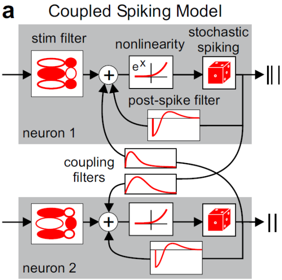
- Orientation selectivity (OS) and ocular dominance (OD) do not require visual experience
- Visual deprivation has little impact on OS and OD
- However, DS does require visual experience in ferrets (Li et al, 2006)

Conclusions

- Recurrent neural networks can analyze visual motion in an online fashion without delayed inputs.
- Formulating a generative model allows learning the recurrent connections via an STDP rule.
- As a model of V1, the RNN makes testable predictions about the lateral connectivity of neurons.
- Responses to stimuli may however be similar to those of spatiotemporal filters.

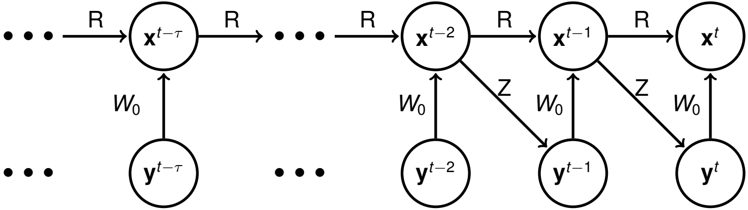
Statement of the problem

The equivalent of spatiotemporal filters: Generalized Linear Models (GLMs)



Pillow et al, 2008.

Recurrent Generalized Linear Models

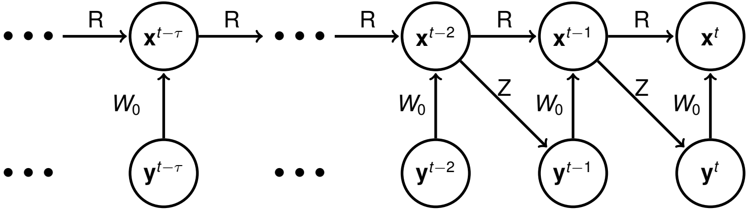


predictive RNN

$$\mathbf{x}^t = \mathbf{W}_0 \;\; \mathbf{y}^t + \mathbf{R} \;\; \mathbf{x}^{t-1}$$

$$\mathbf{y}^t \perp\!\!\!\perp \{\mathbf{y}^{t-1}, \mathbf{y}^{t-2}, \ldots\} \;\mid\; \mathbf{x}^t$$

Recurrent Generalized Linear Models

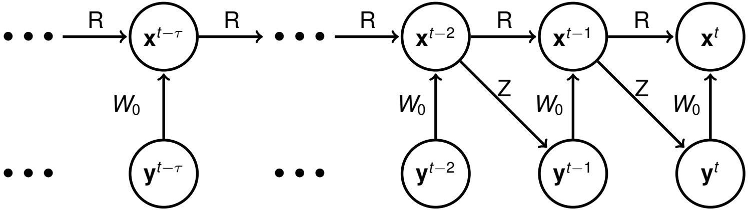


predictive RNN

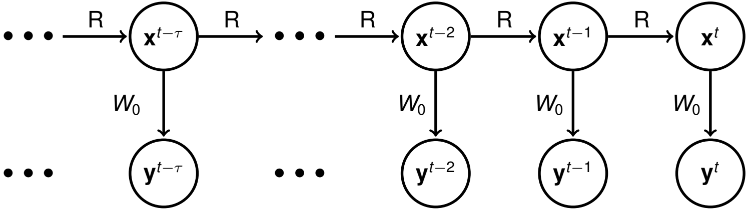
$$\mathbf{x}^t = \mathbf{W}_0 \ \mathbf{y}^t + \mathbf{R} \ \mathbf{x}^{t-1}$$
$$\mathbf{y}^t \perp\!\!\!\perp \{\mathbf{y}^{t-1}, \mathbf{y}^{t-2}, \ldots\} \mid \mathbf{x}^t$$

Similar to state-of-the-art language models: Sutskever et al, 2011, Mikolov et al, 2011

Relationship to linear dynamical system (LDS)



R-GLM



LDS

Problem: cannot have instantaneous connections in a GLM

- where are instantaneous correlations coming from?

Problem: cannot have instantaneous connections in a GLM

- where are instantaneous correlations coming from?
- can add Ising observation model but
  - partition function

$$p(\mathbf{x}) = \frac{1}{Z} e^{-\mathbf{x}^T \mathbf{A}\mathbf{x}/2 - \mathbf{x}^T \mathbf{b}}$$

  - not available for Poisson observations
  - cannot add nonlinear link function like in GLM

Our solution: sequential prediction of each neuron

- constrain $\mathbf{A}$ to be strictly lower triangular

$$p(\mathbf{x}) = \mathrm{Poisson}\left(f\left(\mathbf{A}\mathbf{x} + \mathbf{b}\right)\right)$$

Our solution: sequential prediction of each neuron

- constrain $\mathbf{A}$ to be strictly lower triangular

$$p(\mathbf{x}) = \text{Poisson}\left(f\left(\mathbf{A}\mathbf{x} + \mathbf{b}\right)\right)$$

- Can do the same with Gaussian observation noise.
  $\rightarrow$ Equivalent to full covariance Gaussians.

Our solution: sequential prediction of each neuron

- constrain $\mathbf{A}$ to be strictly lower triangular

$$p(\mathbf{x}) = \text{Poisson}\left(f\left(\mathbf{A}\mathbf{x} + \mathbf{b}\right)\right)$$

- Can do the same with Gaussian observation noise.
  $\rightarrow$ Equivalent to full covariance Gaussians.
- What about the ordering?

Our solution: sequential prediction of each neuron

- constrain $\mathbf{A}$ to be strictly lower triangular

$$p(\mathbf{x}) = \text{Poisson}\left(f\left(\mathbf{A}\mathbf{x} + \mathbf{b}\right)\right)$$

- Can do the same with Gaussian observation noise.
  $\rightarrow$ Equivalent to full covariance Gaussians.
- What about the ordering?
- Can do the same with Bernoulli observation noise.
  $\rightarrow$ Performance matches Ising model.

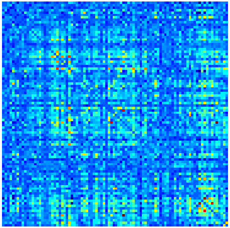
Our solution: sequential prediction of each neuron

- constrain $\mathbf{A}$ to be strictly lower triangular

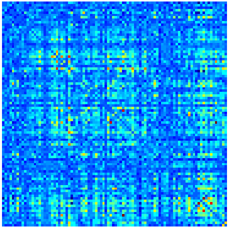
$$p(\mathbf{x}) = \text{Poisson}\left(f\left(\mathbf{A}\mathbf{x} + \mathbf{b}\right)\right)$$

- Can do the same with Gaussian observation noise.
  $\rightarrow$ Equivalent to full covariance Gaussians.
- What about the ordering?
- Can do the same with Bernoulli observation noise.
  $\rightarrow$ Performance matches Ising model.
- Similar to recent image models: Theis et al, 2011 and Larochelle&Murray, 2011.

Sampling the correlated Poisson model



Data correlations

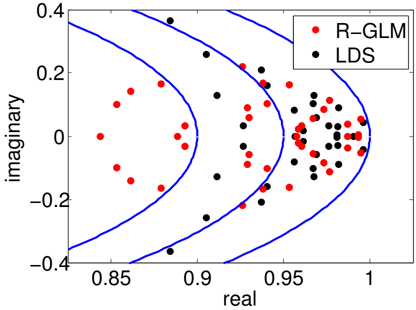


Model correlations

Instantaneous noise and recurrence increase the likelihood

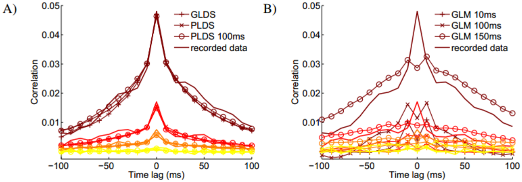
	Likelihood (bits/spike)
fully independent	- 3.15
correlated Poisson	+ 0.175
GLM	+ 0.225
GLM with correlated Poisson	+ 0.03
R-GLM with correlated Poisson	+ 0.03

R-GLM learns long timescales



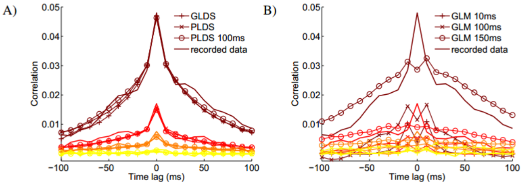
Eigenvalues of the recurrent matrix

Samples of model reproduce spatiotemporal correlations in data

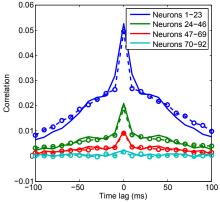


LDS and GLM

Samples of model reproduce spatiotemporal correlations in data

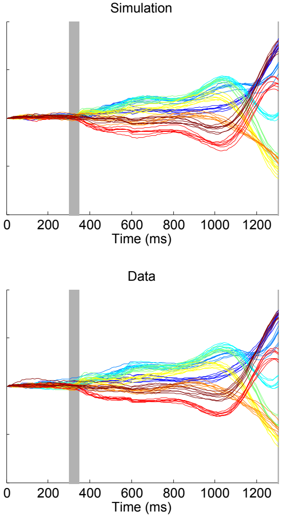


LDS and GLM



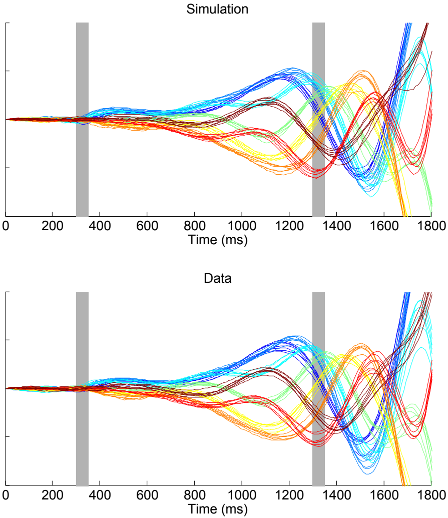
R-GLM

Hidden units integrate info about stimulus



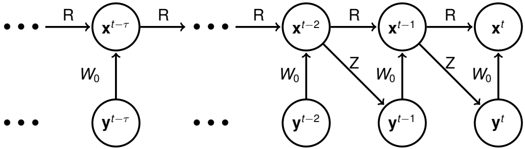
Delay period

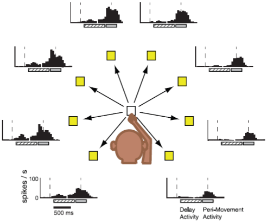
Hidden units generate dynamics



Delay + Movement period

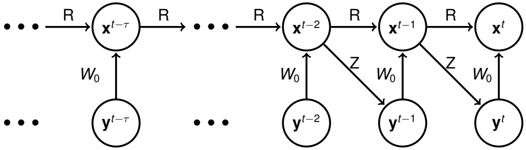
Joint estimation with hand position improves decoding

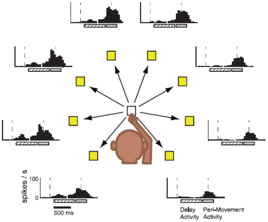




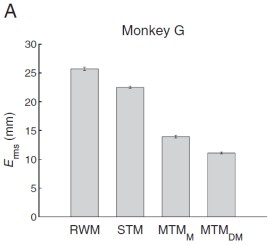
Task

Joint estimation with hand position improves decoding



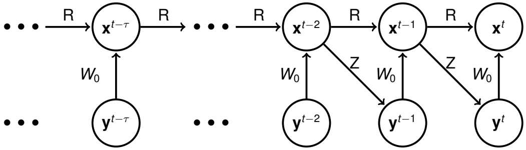


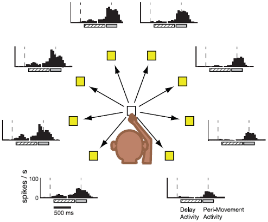
Task



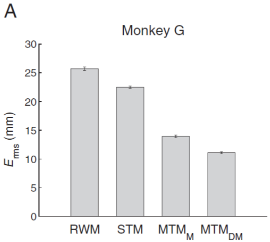
Mixture of trajectories model

Joint estimation with hand position improves decoding





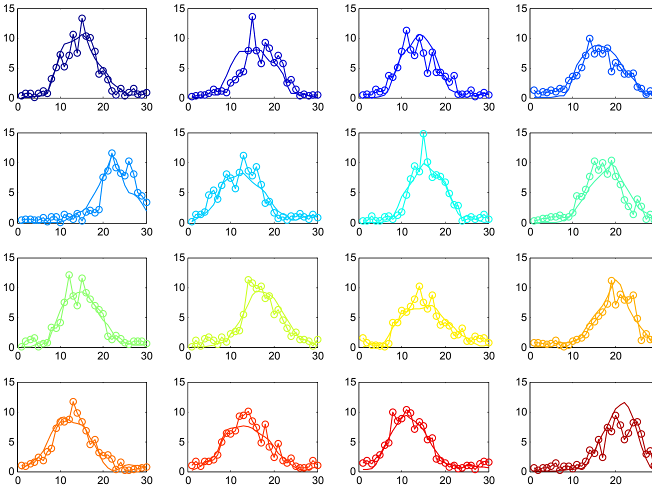
Task



Mixture of trajectories model

Our result: 6.45 mm.

Speed profiles



Conclusions

- Recurrent GLMs with correlated Poisson observations improve statistical models of spike trains.
- The low dimensional parametrization improves decoding of hand trajectories from neural data.

- This work was funded by the Gatsby Charitable Foundation.

The Bayesian Sampling hypothesis

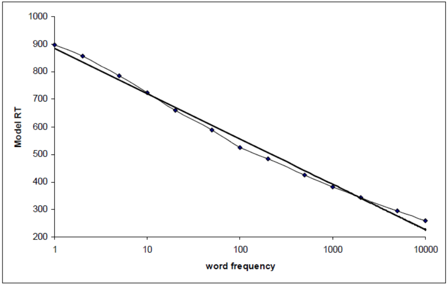
- Bayesian brain hypothesis
- how does the brain do inference: sampling
- interesting data from V1 supports Bayesian sampling hypothesis

The Bayesian Sampling hypothesis

- Bayesian brain hypothesis
- how does the brain do inference: sampling
- interesting data from V1 supports Bayesian sampling hypothesis
- visual word reading time $\sim -\log(\mathrm{P}(\mathrm{word}))$

The Bayesian Sampling hypothesis

- Bayesian brain hypothesis
- how does the brain do inference: sampling
- interesting data from V1 supports Bayesian sampling hypothesis
- visual word reading time $\sim -\log(P(\text{word}))$
- Bayesian reader (Norris, 2006) collects visual samples until $P(\text{word}|\text{visual samples})$ is large.



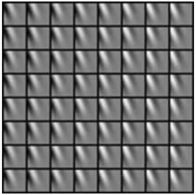
The sequential Bayesian Reader

- Hypothesis: visual word reading time $\sim$ P(word|history)
- Need two ingredients
  - Data
  - good language models

Language modelling: statement of the problem

The Great Gatsby, by F. Scott Fitzgerald
In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since.
"Whenever you feel like criticizing any one," he told me, "just remember that all the people in this world haven't had the advantages that you've had."

The equivalent of spatio-temporal filters and GLMs: N-grams



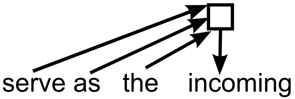
4-gram (frequency)

- serve as the incoming (92)
- serve as the incubator (99)
- serve as the independent (794)
- serve as the index (223)
- serve as the indication (72)
- serve as the indicator (120)

Neural network language models

4-gram (frequency)

- serve as the incoming (92)
- serve as the incubator (99)
- serve as the independent (794)
- serve as the index (223)
- serve as the indication (72)
- serve as the indicator (120)

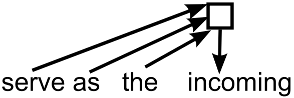


Neural Network Language Model

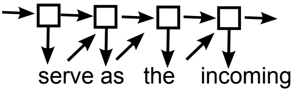
$$\mathbf{h}^{t_0} = \mathbf{f}\left(\sum_{t=1}^{t=\infty} \mathbf{W}^{\mathsf{T}}\, \mathbf{I}_{t_0-t}\right)$$

$$\mathrm{P}(\mathbf{I}_{t0}) = \mathrm{softmax}(\mathbf{Z}^{\mathsf{T}} \mathbf{h}^{t_0}).$$

Recurrent neural network language models



Neural Network Language Model



Recurrent Neural Network Language Model

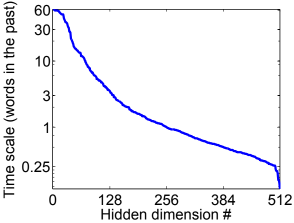
- State of the art (Mikolov et al 2011).
- Our simplification: linear RNN (R-GLM).

$$\mathbf{h}^{\mathbf{t_0}} = \mathbf{R}\ \mathbf{h}^{\mathbf{t_0 - 1}} + \mathbf{W_0^T}\ \mathbf{I}_{\mathbf{t_0 - t}}.$$

$$\mathbf{h}^{\mathbf{t_0}} = \sum_{\mathbf{t=0}}^{\mathbf{t=\infty}} \mathbf{R}^{\mathbf{t}}\ \mathbf{I}_{\mathbf{t_0 - t}}.$$

What are the relevant time scales of language?

- R-GLM learns caching.



The timescales of language

- Long time scales are good: dynamic R-GLM further adapts parameters at test time.

Perplexity results on Penn Corpus (930k tokens, 10k vocab) - single models

	Single	+KN5+cache	x10	x10+KN5+cache
5-gram Kneser-Ney[1]	141.2	125.7		
feedforward NNLM[1]	140.2	106.6		
Log-bilinear LM[1]	144.5	105.8		
RNN[1]	124.7	97.5	102.1	89.4
dynamic RNN[1]	123.2	98.0	101.0	90.0
R-GLM(no reg)	137			
R-GLM(L1 reg)	125			
R-GLM ([2]DO&CN)	**102**	**94**	98.8	92.5
dynamic R-GLM([2]DO&CN)	**98.4**	**90.7**	95.1	89.1

[1] copied from Tomas Mikolov thesis
[2] trained with random dropouts and column normalization

Conclusions

- None yet. Need to collect data.