Oscillatory Recurrent Gated Neural Integrator Circuits (ORGaNICs)

David J. Heeger
New York University

Heeger PNAS (2017)
Heeger & Mackey arXiv (2018)

Part I: Working memory

Working memory: cognitive psychology

Cognitive process that is responsible for temporarily **maintaining** and **manipulating** information.

Example from language:

Problem of long-term dependencies

Working memory: cognitive psychology

Cognitive process that is responsible for temporarily **maintaining** and **manipulating** information.

Example from language:

The athlete realized his goals, which were formed during childhood, to qualify for this year's Olympic team, …

Problem of long-term dependencies

Working memory: cognitive psychology

Cognitive process that is responsible for temporarily **maintaining** and **manipulating** information.

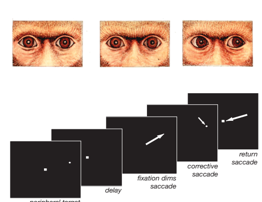
Example from language:

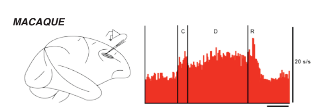
The athlete realized his goals, which were formed during childhood, to qualify for this year's Olympic team, … (quickly/were unattainable).

Problem of long-term dependencies

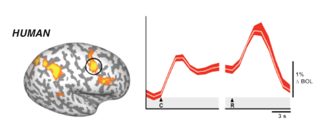
Working memory: neuroscience

Example: oculomotor delayed-response (ODR) task





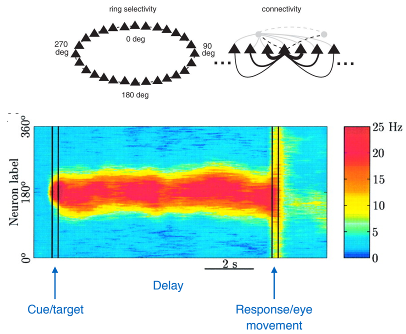
Funahashi, Bruce & Goldman-Rakic (1989)



Srimal & Curtis (2008)

Working memory models

(sustained delay-period activity)



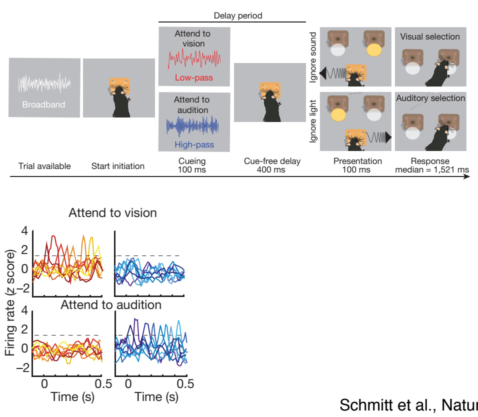
Compte Brunel, Goldman-Rakic, & Wang (2000)

Weaknesses of the delayed-response / sustained delay-period activity paradigm

Working memory involves **maintenance and manipulation**, but most of the neuroscience focuses only on maintenance.

Neural activity can exhibit complicated dynamics during a delay period.

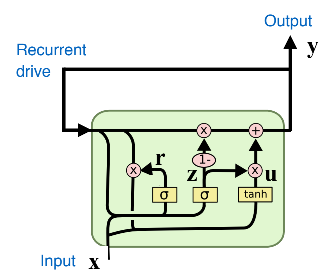
Sequential activity



Schmitt et al., Nature (2017)

Working memory: AI

Long short term memory networks (LSTMs):



$$\tau_i \frac{dy_i}{dt} = -y_i + u_i$$

$$\tau_i = 1/z_i$$

$$\mathbf{u} = \tanh\left(\mathbf{W}_{yx}\mathbf{x} + \mathbf{W}_{yy}\mathbf{v} + \mathbf{b}_u\right)$$

$$v_i = r_i y_i$$

$$\mathbf{z} = \sigma\left(\mathbf{W}_{zx}\mathbf{x} + \mathbf{W}_{zy}\mathbf{y} + \mathbf{b}_z\right)$$

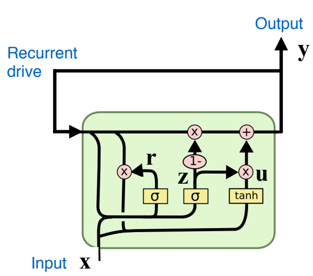
$$\mathbf{r} = \sigma\left(\mathbf{W}_{rx}\mathbf{x} + \mathbf{W}_{ry}\mathbf{y} + \mathbf{b}_r\right)$$

$$\sigma(x) = \frac{e^x}{e^x + 1}$$

Colah's blog, Understanding LSTM Networks (2015)
see also:
Hochreiter & Schmidhuber (1997)
Cho et al. (2014)

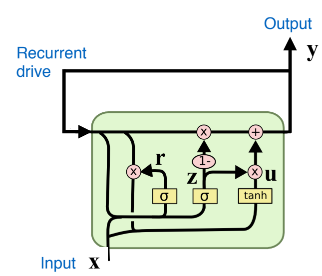
Working memory: AI

Long short term memory networks (LSTMs):



Working memory: AI

Long short term memory networks (LSTMs):



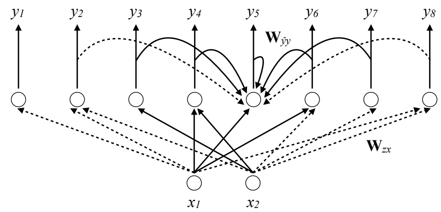
Insight: gated integration and reset

Learning to generate Shakespeare

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

Karpathy's blog, The Unreasonable Effectiveness of Recurrent Neural Networks (2015)

Leaky neural integrator



Input drive · Recurrent drive · Encoding weight matrix · Recurrent weight matrix

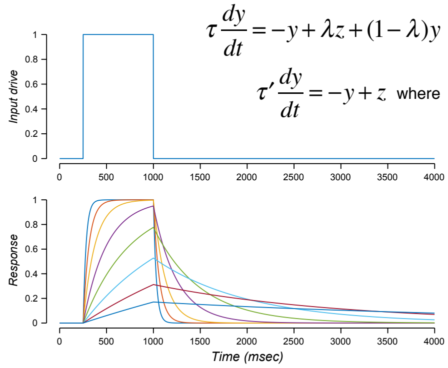
$$\tau \frac{d\mathbf{y}}{dt} = -\mathbf{y} + \lambda \mathbf{z} + (1-\lambda)\hat{\mathbf{y}} \qquad \mathbf{z} = \mathbf{W}_{zx}\mathbf{x} \qquad \hat{\mathbf{y}} = \mathbf{W}_{\hat{y}y}\mathbf{y}$$

$$0 < \lambda < 1$$

Effective time constant

$$\tau \frac{dy}{dt} = -y + \lambda z + (1-\lambda) y$$

$$\tau' \frac{dy}{dt} = -y + z \quad \text{where} \quad \tau' = \frac{\tau}{\lambda}$$



ORGaNICs

$$\tau_y \frac{dy_j}{dt} = -y_j + \left(\frac{b_j^+}{1+b_j^+}\right) z_j + \left(\frac{1}{1+a_j^+}\right) \hat{y}_j$$

Input drive: $\mathbf{z} = \mathbf{W}_{zx}\mathbf{x} + \mathbf{c}_z$ (Encoding weights, Input)

Recurrent drive: $\hat{\mathbf{y}} = \mathbf{W}_{\hat{y}y}\mathbf{y} + \mathbf{c}_{\hat{y}}$ (Recurrent weights, Output)

Readout: $\mathbf{W}_{ry}\mathbf{y}$ followed (optionally) by output nonlinearity (e.g., rectification)

Modulators depend on inputs and outputs:

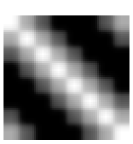
$$\tau_a \frac{d\mathbf{a}}{dt} = -\mathbf{a} + \mathbf{W}_{ax}\mathbf{x} + \mathbf{W}_{ay}\mathbf{y} + \mathbf{c}_a$$

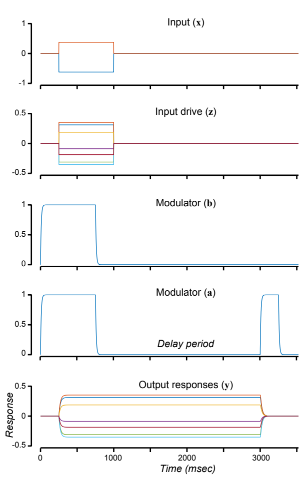
$$\tau_b \frac{d\mathbf{b}}{dt} = -\mathbf{b} + \mathbf{W}_{bx}\mathbf{x} + \mathbf{W}_{by}\mathbf{y} + \mathbf{c}_b$$

Superscript "+" means rectifying nonlinearity

Sustained delay-period activity

Center-surround recurrent weight matrix:





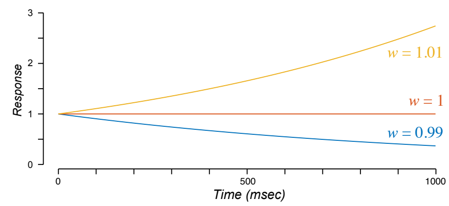
Response dynamics during delay period

Response dynamics during delay period (when $\mathbf{a} = \mathbf{b} = \mathbf{0}$):

$$\tau_y \frac{d\mathbf{y}}{dt} = -\mathbf{y} + \mathbf{W}_{\hat{y}y}\mathbf{y}$$

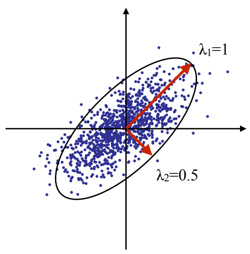
Example recurrent weight matrix: $\mathbf{W}_{\hat{y}y} = \begin{pmatrix} 0.99 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1.01 \end{pmatrix}$

Initial responses $\mathbf{y} = \mathbf{y}_0 = \mathbf{1}$ at the beginning of the delay period.

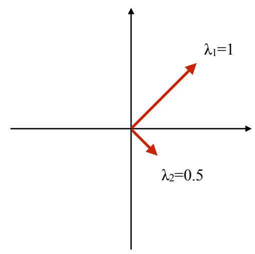


Stability depends on the eigenvectors of recurrent weight matrix

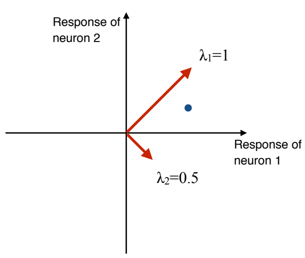
What's an eigenvector?



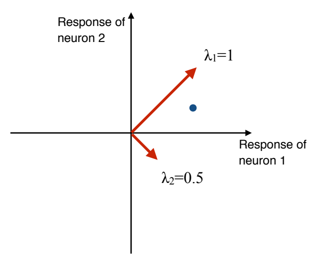
Stability depends on the eigenvectors of recurrent weight matrix



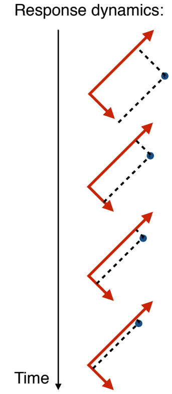
Stability depends on the eigenvectors of recurrent weight matrix



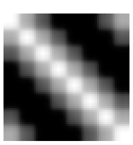
Stability depends on the eigenvectors of recurrent weight matrix



Response dynamics:



Encoding & readout weights



Recurrent weight matrix has **two** eigenvalues = 1 and others < 1.

Representational dimensionality D = 2 (i.e., 2D continuous attractor).

Encoding / embedding weights: $\mathbf{z} = \mathbf{W}_{zx}\mathbf{x} = \mathbf{V}\mathbf{x}$

Readout / decoding weights: $\hat{\mathbf{x}} = \mathbf{W}_{ry}\mathbf{y} = \mathbf{V}^t\mathbf{y}$

$\mathbf{V}$: columns are the eigenvectors of the recurrent weight matrix with corresponding eigenvalues = 1

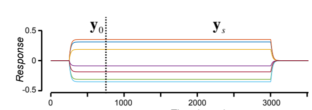
Readout derivation

Encoding / embedding (during target presentation):

$$\mathbf{y}_0 = \mathbf{V}\mathbf{x}_0$$

Steady-state responses during delay period:

$$\mathbf{y}_s = \mathbf{V}\mathbf{p}$$
$$\mathbf{p} = \mathbf{V}^t\mathbf{y}_0$$

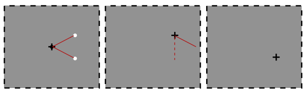


Readout during delay period (after reaching steady state):

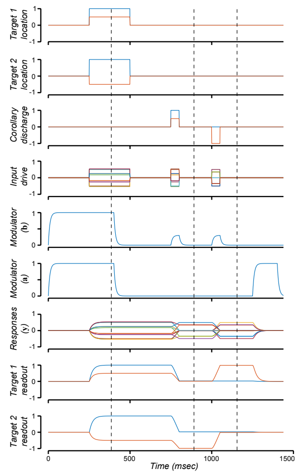
$$\hat{\mathbf{x}} = \mathbf{V}^t\mathbf{y}_s$$
$$= \mathbf{V}^t\mathbf{V}\mathbf{p} = \mathbf{V}^t\mathbf{V}\mathbf{V}^t\mathbf{y}_0 = \mathbf{V}^t\mathbf{V}\mathbf{V}^t\mathbf{V}\mathbf{x}_0 = \mathbf{x}_0$$

Manipulation with gated integration

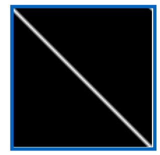
Double-step saccade task



Recurrent weight matrix



Sequential activity

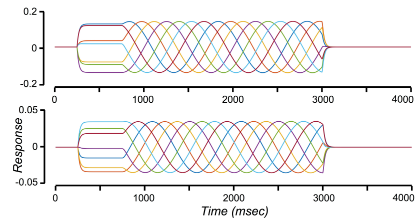


Encoding weights: $\mathbf{z} = \mathbf{W}_{zx}\mathbf{x} = \mathbf{V}\mathbf{x}$

Readout weights: $\hat{\mathbf{x}} = \mathbf{W}_{ry}\mathbf{y} = \mathbf{V}^t\mathbf{y}$

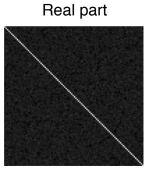
$\mathbf{V}$: columns are complex-valued eigenvectors with eigenvalues that have real part = 1. Imaginary part determines oscillation frequency.

Representational dimensionality D = 2.

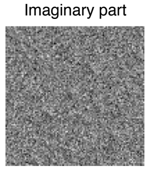


Complex dynamics

Real part

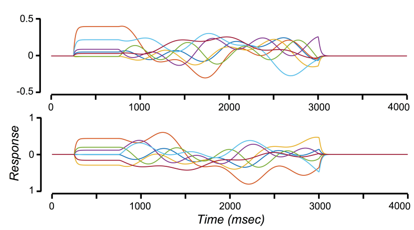


Imaginary part



Complex-valued recurrent weight matrix

Representational dimensionality D = 10.



Stability and E:I balance

Generalization with different intrinsic time constants:

$$\tau_{y_j}\frac{dy_j}{dt} = -y_j + \left(\frac{b_j^+}{1+b_j^+}\right) z_j + \left(\frac{1}{1+a_j^+}\right)\hat{y}_j$$

Response dynamics during delay period (when $\mathbf{a} = \mathbf{b} = \mathbf{0}$):

$$d(\boldsymbol{\tau}_y)\frac{d\mathbf{y}}{dt} = -\mathbf{y} + \mathbf{W}_{\hat{y}y}\mathbf{y}$$

Stability depends on the eigenvalues of this matrix:

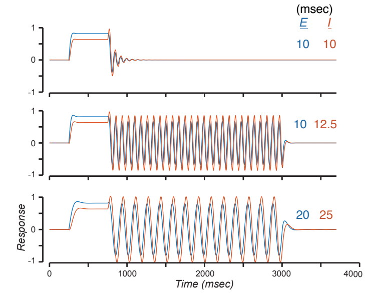
$$\mathbf{W}'_{\hat{y}y} = d\left(\tfrac{1}{\tau_y}\right)\left(\mathbf{W}_{\hat{y}y} - \mathbf{I}\right)$$

Oscillation frequencies:

$$f_i = \tfrac{1000}{2\pi}\,\mathrm{Im}(\lambda_i)$$

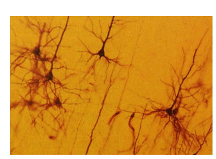
Example recurrent weight matrix:

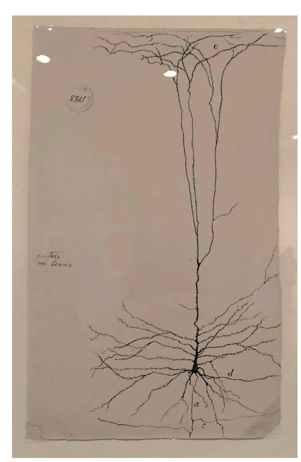
$$\mathbf{W}_{\hat{y}y} = \begin{pmatrix} 2 & -1 \\ 2 & -0.25 \end{pmatrix}$$



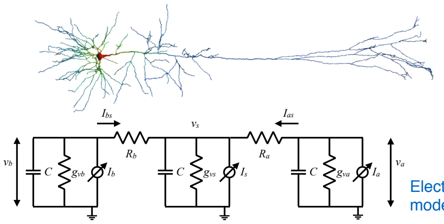
Part II: Biophysical implementation

Pyramidal cells





Biophysical implementation: output responses



Electrical-circuit model

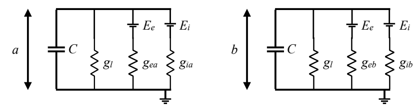
$$\left(C\frac{dv_b}{dt}+g_{vb}v_b-I_b\right)+\left(C\frac{dv_s}{dt}+g_{vs}v_s-I_s\right)+\left(C\frac{dv_a}{dt}+g_{va}v_a-I_a\right)=0$$

Steady state: $$gv_s=\left(\frac{1}{1+R_bg_{vb}}\right)I_b+I_s+\left(\frac{1}{1+R_ag_{va}}\right)I_a$$

b a

- Input drive   Input drive   Recurrent drive

Biophysical implementation: modulators

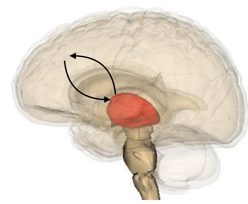


$$C\frac{da}{dt}=-g_l(a-E_l)-g_{ea}(a-E_e)-g_{ia}(a-E_i)$$

With $E_l=0, E_s=0, E_e=1, E_i=-1$

$$\tau\frac{da}{dt}=-a+\frac{g_{ea}-g_{ia}}{g}$$ where $g=(g_l+g_{ea}+g_{ia})$ and $\tau=\frac{C}{g}$

Linear sum followed by sigmoid



Thalamocortical loops: Schmitt et al., Nature (2017) and Guo et al., Nature (2017).

Part III: Canonical computation: sensory & motor processing

Motor preparation and motor control

Backside double McTwist 1260
(Shawn White, 2018)

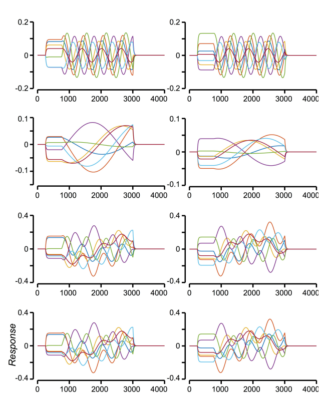
Motor preparation and motor control

Responses when input drives 1st eigenvector

Responses when input drives 2nd eigenvector

Responses when input drives both eigenvectors

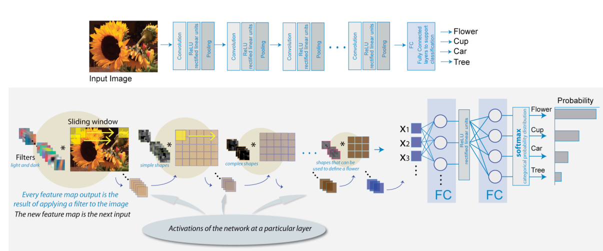
Sum of top two panels



Readout: weighted sum followed (optionally) by output nonlinearity (e.g., rectification).

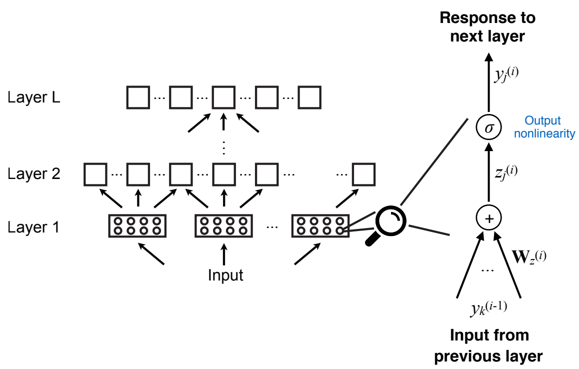
Sensory processing

Stack 'em with convolutional encoding weights, like a deep net:

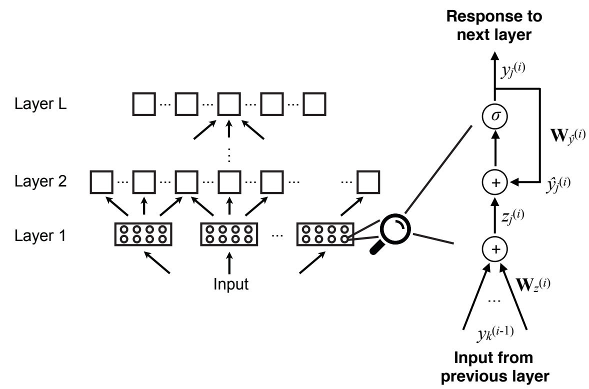


https://www.mathworks.com/help/nnet/convolutional-neural-networks.html

Conventional feedforward network



ORGaNICs feedforward network



Part IV: Prediction

ORGaNICs (revisited)

Global optimization:

Output, Input drive, Output, Recurrent drive

$$E = \tfrac{1}{2}\int_t \sum_j \left(\tfrac{b_j^+}{1+b_j^+}\right)\left[y_j - z_j\right]^2 + \left(\tfrac{1}{1+b_j^+}\right)\left[y_j - \left(\tfrac{1}{1+\alpha_j^+}\right)\hat{y}_j\right]^2$$

Local computation:

$$\tau_y \frac{dy_j}{dt} = -\frac{dE}{dy_j}$$

$$= -y_j + \left(\tfrac{b_j^+}{1+b_j^+}\right)z_j + \left(\tfrac{1}{1+b_j^+}\right)\left(\tfrac{1}{1+\alpha_j^+}\right)\hat{y}_j$$

$$= -y_j + \left(\tfrac{b_j^+}{1+b_j^+}\right)z_j + \left(\tfrac{1}{1+a_j^+}\right)\hat{y}_j \quad \text{where } \left(1+a_j^+\right) = \left(1+b_j^+\right)\left(1+\alpha_j^+\right)$$

Time-series prediction: global optimization

Summed responses, Input drive, Output responses, Recurrent drive

$$E = \tfrac{1}{2}\int_t \sum_j \left(\tfrac{b_j^+}{1+b_j^+}\right)\left[\sum_k \mathrm{Re}\left(y_k\right) - x\right]^2 + \left(\tfrac{1}{1+b_j^+}\right)\left[y_j - \left(\tfrac{1}{1+\alpha_j^+}\right)\hat{y}_j\right]^2$$

Recurrent drive:

$$\hat{\mathbf{y}} = \mathbf{W}_{\hat{y}y}\mathbf{y}$$

Recurrent weights, Output

Diagonal recurrent weight matrix:

$$w_j = 1 + i2\pi\omega_j\tau_y$$

Frequency

Time-series prediction: local computation

$$\tau_y \frac{dy_j}{dt} = -\frac{dE}{dy_j}$$

$$= -y_j + \left(\tfrac{b_j^+}{1+b_j^+}\right)x + \left(\tfrac{1}{1+a_j^+}\right)\hat{y}_j + \left(\tfrac{b_j^+}{1+b_j^+}\right)\left[y_j - \sum_k \mathrm{Re}\left(y_k\right)\right]$$

where $\left(1+a_j^+\right) = \left(1+b_j^+\right)\left(1+\alpha_j^+\right)$

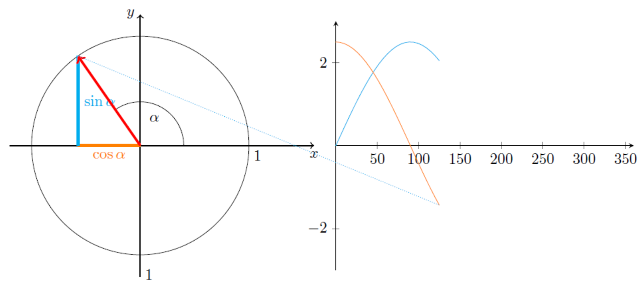
Time-series prediction: local computation

$$\tau_y \frac{dy_j}{dt} = -\frac{dE}{dy_j}$$

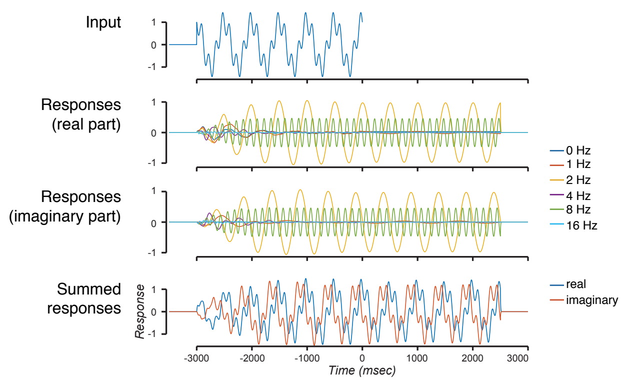
$$= -y_j + \left(\frac{b_j^+}{1+b_j^+}\right) x + \left(\frac{1}{1+a_j^+}\right) \hat{y}_j$$

where $(1+a_j^+) = (1+b_j^+)(1+\alpha_j^+)$

Predictive basis functions



Time-series prediction



Processing delays mean the brain has to make predictions:

Processing delays mean the brain has to make predictions:

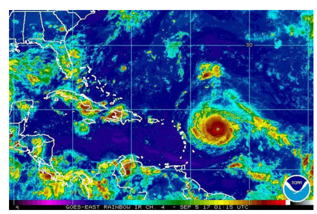
Comprehension relies on prediction

Comprehension relies on prediction

Events unfold over time

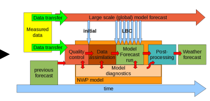
Events unfold over time

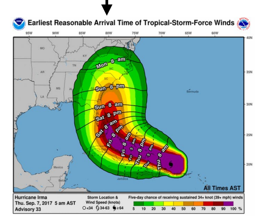
Prediction requires a model



Observation

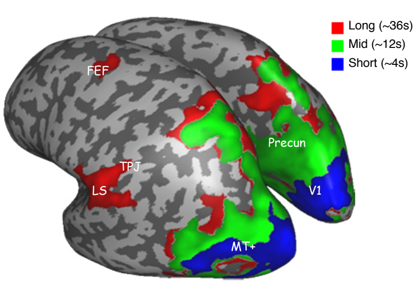
Numerical model





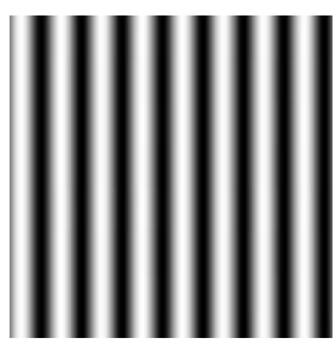
Prediction

Hierarchy of processing time scales

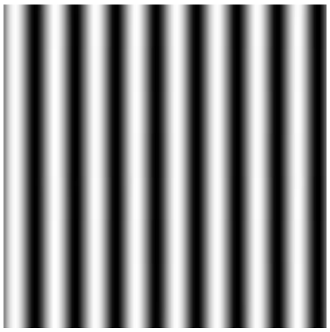


Hasson et al., J Neurosci (2008)
see also:
- Honey et al., Neuron (2012)
- Farbood et al., Frontiers (2015)

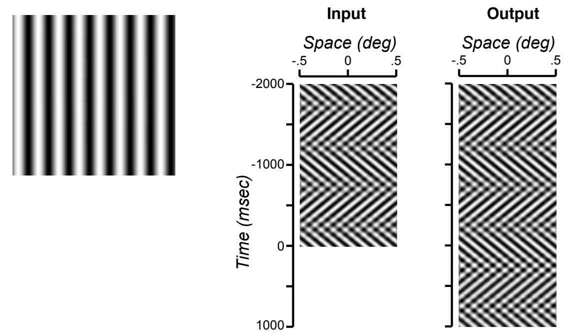
Motion prediction



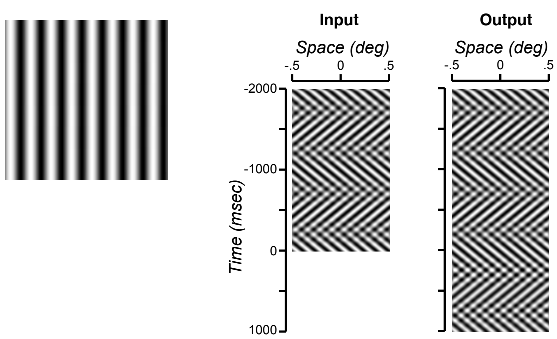
Motion prediction



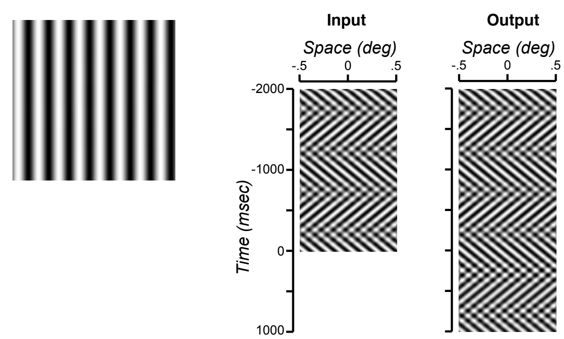
Motion prediction



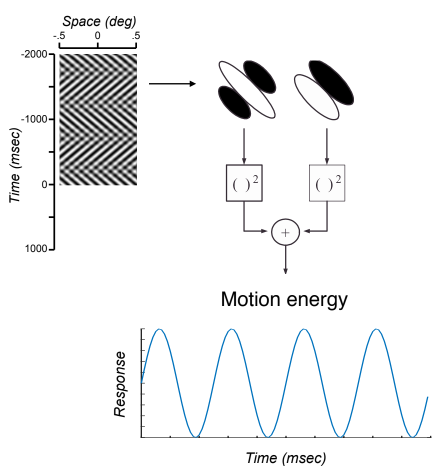
Motion prediction



Motion prediction



Feedforward motion estimation



Time-series prediction (revisited)

Summed responses | Input drive | Output responses | Recurrent drive

$$E = \tfrac{1}{2}\sum_t \sum_j \lambda \left[\sum_k \mathrm{Re}(y_k) - x\right]^2 + (1-\lambda)\left[y_j - \hat{y}_j\right]^2$$

$$\tau_y \frac{dy_j}{dt} = -\frac{dE}{dy_j} = -\lambda\left[\sum_k \mathrm{Re}(y_k) - x\right] - (1-\lambda)\left(y_j - \hat{y}_j\right)$$

where $\lambda = \left(\frac{b_j^+}{1+b_j^+}\right)$ and $\alpha = 0$

Global optimization

$$E = \frac{1}{2}\sum_{i=1}^{L}\sum_{n}\sum_{t}\alpha^{(i)}(t)\lambda^{(i)}(t)\left[\left(\sum_{m}\mathrm{Re}\left(y_{nm}^{(i)}(x,t)\right)\right) - z_n^{(i)}(x,t)\right]^2 + \frac{1}{2}\sum_{i=1}^{L}\sum_{n}\sum_{t}\alpha^{(i)}(t)\left(1-\lambda^{(i)}(t)\right)\left[\sum_{m}\left(y_{nm}^{(i)}(x,t) - \hat{y}_{nm}^{(i)}(x,t)\right)^2\right]$$

State · Summed responses · Input drive · Output responses · Prior/prediction

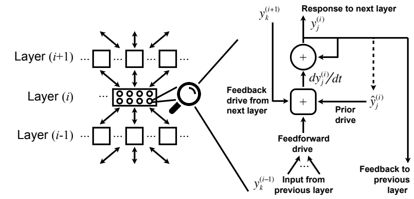
$\hat{y}_{nm}^{(i)}(x,t) = y_{nm}^{(i)}(x,t-\Delta t)\, w_m^{(i)}$ Prior/prediction from previous time step

$w_m^{(i)}(\Delta t) = e^{i 2\pi\omega_m^{(i)}\Delta t}$ Weights specified by predictive basis functions

$z_j^{(i)} = \frac{1}{2}\left(v_j^{(i)}\right)^2$ Quadratic output nonlinearity

$v_j^{(i)} = \sum_k w_{jk}^{(i-1)} y_k^{(i-1)}$ Weighted sum / convolution

Local computation



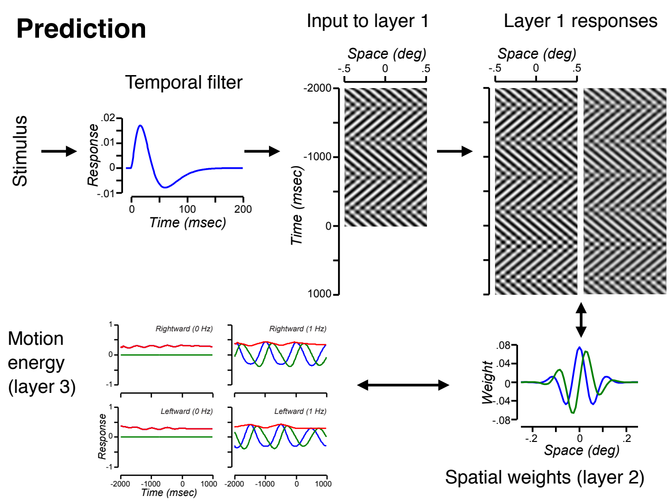
$$\tau\frac{dy_{nm}^{(i)}}{dt} = -\frac{dE}{dy_{nm}^{(i)}} = -\alpha^{(i)}\lambda^{(i)} f_n^{(i)} + \alpha^{(i+1)}\lambda^{(i+1)} b_n^{(i)} - \alpha^{(i)}\left(1-\lambda^{(i)}\right)p_{nm}^{(i)}$$

$f_n^{(i)} = \left(\sum_m \mathrm{Re}\left(y_{nm}^{(i)}(x,t)\right)\right) - z_n^{(i)}(x,t)$ Feedforward drive

$p_{nm}^{(i)} = y_{nm}^{(i)}(x,t) - \hat{y}_{nm}^{(i)}(x,t)$ Prior drive

$b_n^{(i)} = \sum_k\left[\left(\sum_m \mathrm{Re}\left(y_{km}^{(i+1)}(x,t)\right)\right) - z_k^{(i+1)}(x,t)\right] v_k^{(i+1)} w_{kn}^{(i)}$ Feedback drive

Prediction



Summary

ORGaNICs (straightforward extension of leaky neural integrators):

- Sensory processing
- Motor preparation and motor control
- Executive control (working memory, controlling attention).
- Prediction
- Inference in a multi-layered recurrent neural net

Conceptual framework:

- Gated integration & reset
- Effective time constant
- Dimensionality
- Stability / E:I balance
- Time warping via τ

Implications for neuroscience

1) Working memory/executive functions, motor preparation/control, and sensory processing may share a common computational foundation.
2) Working memory > short-term memory.
3) Complex dynamics:
   - Unified model for sustained delay-period activity, sequential activity, and complex dynamics.
   - Read out in spite of complex dynamics.
4) Experiments:
   - Example of testable prediction: thalamic input changes the effective time constant and recurrent gain of a PFC neuron.
   - New conceptual framework / new paradigm: gated integration, reset, effective time constant.

Implications for AI

1) Go complex: simple harmonic motion is everywhere!
2) Stability:
   - Avoid exploding gradients by rescaling recurrent weight matrix after each gradient update (s.t. largest eigenvalue = 1).
   - Avoid vanishing gradients by using rectification instead of saturating nonlinearities.
3) Reset & update gates = gated integration, reset, effective time-constant.
4) Warp time by scaling the intrinsic time constants.
5) Neuromorphic (analog VLSI) implementation.

Funding: None (your name here)

Special thanks: Wayne Mackey, Mike Halassa, Eero Simoncelli, Mike Landy, Roozbeh Kiani, Charlie Burlingham, Rui Costa, Walter Senn.

Funding: None (your name here)

Special thanks: Wayne Mackey, Mike Halassa, Eero Simoncelli, Mike Landy, Roozbeh Kiani, Charlie Burlingham, Rui Costa, Walter Senn.

Thank you

ORGaNICs

$$\tau_y \frac{dy_j}{dt} = -y_j + \left(\frac{b_j^+}{1+b_j^+}\right) z_j + \left(\frac{1}{1+a_j^+}\right) \hat{y}_j$$

Input drive:

$$\mathbf{z} = \mathbf{W}_{zx}\mathbf{x} + \mathbf{c}_z$$

Encoding weights — Input

Recurrent drive:

$$\hat{\mathbf{y}} = \mathbf{W}_{\hat{y}y}\mathbf{y} + \mathbf{c}_{\hat{y}}$$

Recurrent weights — Output

Readout:

$$\mathbf{W}_{ry}\mathbf{y}$$

followed (optionally) by output nonlinearity (e.g., rectification)

Modulators depend on inputs and outputs:

$$\tau_a \frac{d\mathbf{a}}{dt} = -\mathbf{a} + \mathbf{W}_{ax}\mathbf{x} + \mathbf{W}_{ay}\mathbf{y} + \mathbf{c}_a$$

$$\tau_b \frac{d\mathbf{b}}{dt} = -\mathbf{b} + \mathbf{W}_{bx}\mathbf{x} + \mathbf{W}_{by}\mathbf{y} + \mathbf{c}_b$$

Superscript “+” means rectifying nonlinearity