SYDE 556/750

Simulating Neurobiological Systems
Lecture 3: Representations

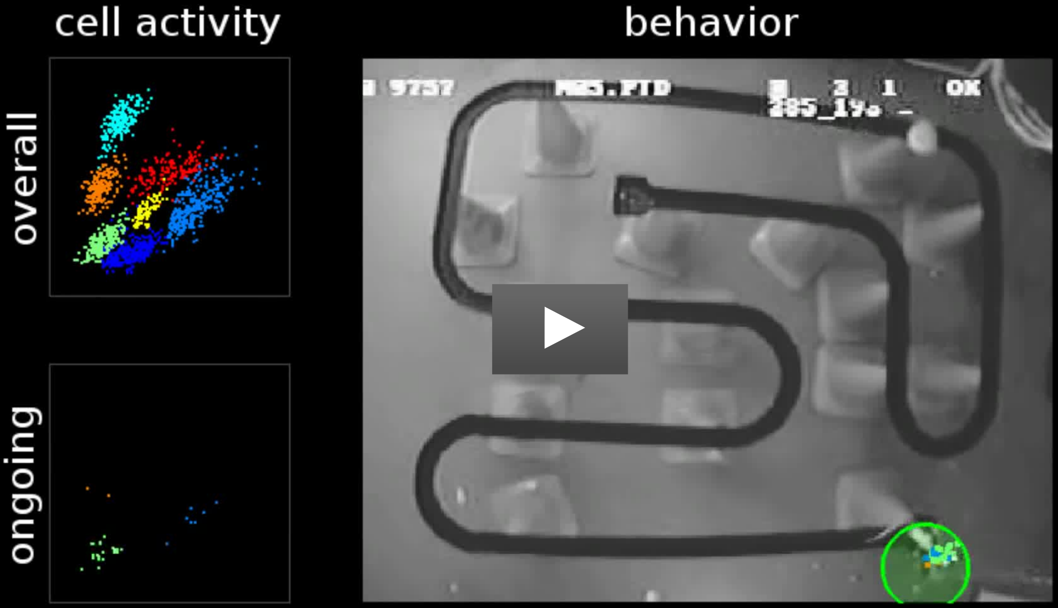
Andreas Stöckel

January 14 & 16, 2020

UNIVERSITY OF WATERLOO | FACULTY OF ENGINEERING

Visual Cortex

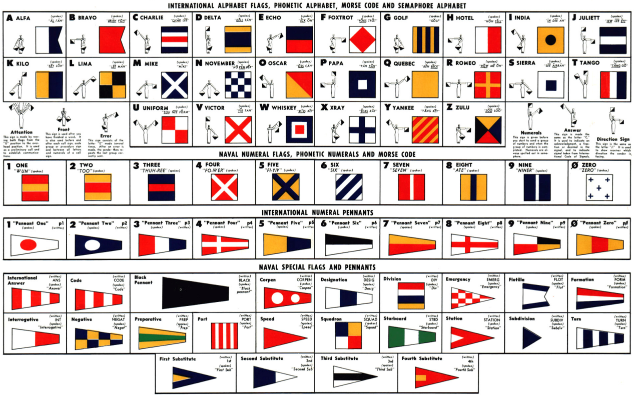
Mapping receptive fields


cell activity
behavior
overall
ongoing

NEF Principle 1: Representation

NEF Principle 1 – Representation
Groups (“populations”, or “ensembles”) of neurons *represent* represent values via nonlinear encoding and linear decoding.

Lossless Codes



A	· −	U	· · −
B	− · · ·	V	· · · −
C	− · − ·	W	· − −
D	− · ·	X	− · · −
E	·	Y	− · − −
F	· · − ·	Z	− − · ·
G	− − ·		
H	· · · ·		
I	· ·		
J	· − − −		
K	− · −	1	· − − − −
L	· − · ·	2	· · − − −
M	− −	3	· · · − −
N	− ·	4	· · · · −
O	− − −	5	· · · · ·
P	· − − ·	6	− · · · ·
Q	− − · −	7	− − · · ·
R	· − ·	8	− − − · ·
S	· · ·	9	− − − − ·
T	−	0	− − − − −

Encoding: $\mathbf{a} = f(\mathbf{x})$ Decoding: $\mathbf{x} = f^{-1}(\mathbf{a})$

Binary numbers: Nonlinear encoding, linear decoding

- Represent a natural number between 0 and $2^n - 1$ as n binary digits.

Binary numbers: Nonlinear encoding, linear decoding

- Represent a natural number between 0 and $2^n - 1$ as n binary digits.
- **Nonlinear encoding**

$$a_i = \big(f(x)\big)_i = \begin{cases} 1 & \text{if } x - 2^i \left\lfloor \frac{x}{2^i} \right\rfloor > 2^{i-1}\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

Binary numbers: Nonlinear encoding, linear decoding

- Represent a natural number between 0 and $2^n - 1$ as n binary digits.
- **Nonlinear encoding**

$$a_i = \big(f(x)\big)_i = \begin{cases} 1 & \text{if } x - 2^i \left\lfloor \frac{x}{2^i} \right\rfloor > 2^{i-1}\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

- **Linear decoding**

$$x = f^{-1}(\mathbf{a}) = \sum_{i=0}^{n-1} 2^i a_i = \mathbf{F}\mathbf{a} = \begin{pmatrix} 1 & 2 & \dots & 2^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}\,.$$

Binary numbers: Nonlinear encoding, linear decoding

- Represent a natural number between 0 and $2^n - 1$ as n binary digits.
- **Nonlinear encoding**

$$a_i = \big(f(x)\big)_i = \begin{cases} 1 & \text{if } x - 2^i \left\lfloor \frac{x}{2^i} \right\rfloor > 2^{i-1}\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

- **Linear decoding**

$$x = f^{-1}(\mathbf{a}) = \sum_{i=0}^{n-1} 2^i a_i = \mathbf{F}\mathbf{a} = \begin{pmatrix} 1 & 2 & \dots & 2^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}.$$

- This is a **distributed code**.

Binary numbers: Nonlinear encoding, linear decoding

- Represent a natural number between 0 and $2^n - 1$ as n binary digits.
- **Nonlinear encoding**

$$a_i = \big(f(x)\big)_i = \begin{cases} 1 & \text{if } x - 2^i \left\lfloor \frac{x}{2^i} \right\rfloor > 2^{i-1}\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

- **Linear decoding**

$$x = f^{-1}(\mathbf{a}) = \sum_{i=0}^{n-1} 2^i a_i = \mathbf{F}\mathbf{a} = \begin{pmatrix} 1 & 2 & \dots & 2^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}\,.$$

- This is a **distributed code**. But, **not robust** against additive noise!

Lossy codes

- **Lossy code**
  Inverse f^{-1} does not exist, instead *approximate* the represented value

$$\text{Encoding:} \quad \mathbf{a} = f(\mathbf{x}) \qquad\qquad \text{Decoding:} \quad \mathbf{x} \approx g(\mathbf{a})$$

Lossy codes

- **Lossy code**
  Inverse f^{-1} does not exist, instead *approximate* the represented value

$$\text{Encoding:} \quad \mathbf{a} = f(\mathbf{x}) \qquad\qquad \text{Decoding:} \quad \mathbf{x} \approx g(\mathbf{a})$$

- **Examples**
  - Audio, image, and video coding schemes
    (MP3, JPEG, H.264)
  - Basis transformation onto first n principal components (PCA)

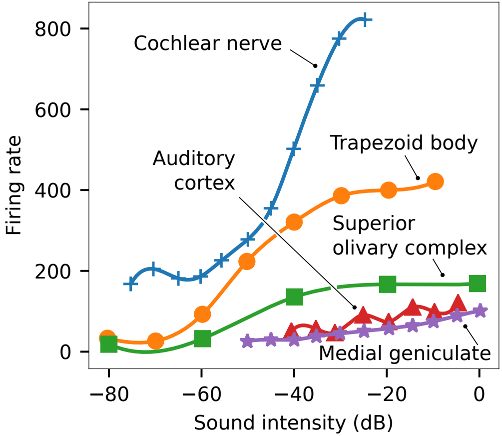
Lossy codes

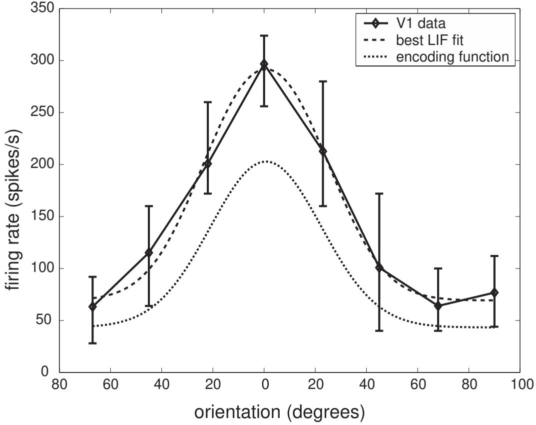
- **Lossy code**
  Inverse f^{-1} does not exist, instead *approximate* the represented value

$$\text{Encoding:} \quad \mathbf{a} = f(\mathbf{x}) \qquad\qquad \text{Decoding:} \quad \mathbf{x} \approx g(\mathbf{a})$$

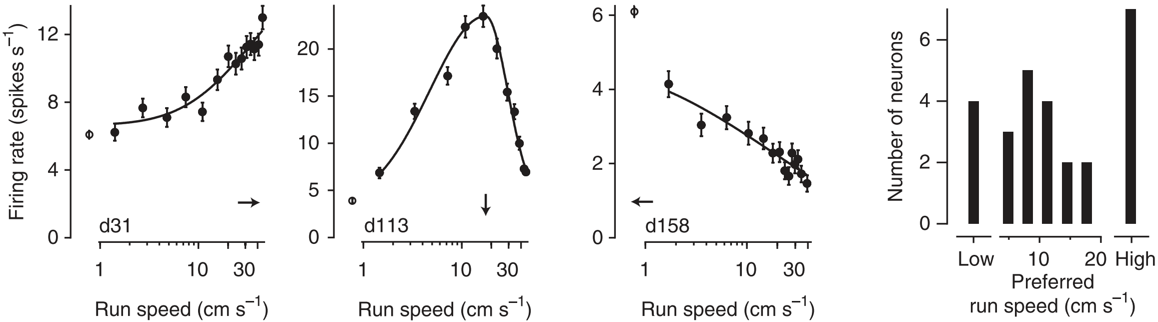
- **Examples**
  - Audio, image, and video coding schemes
    (MP3, JPEG, H.264)
  - Basis transformation onto first n principal components (PCA)
  - **Neural Representations**

Tuning curves (I)

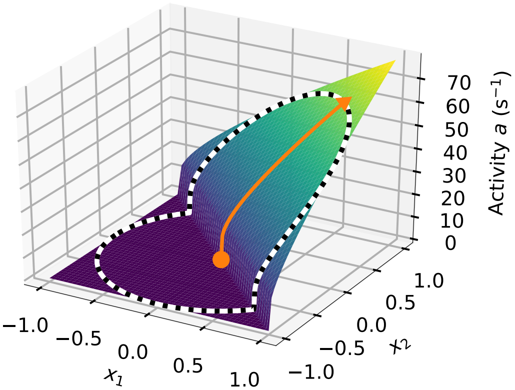




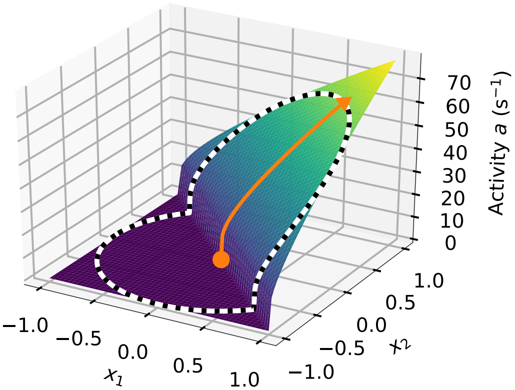
Tuning curves (II)

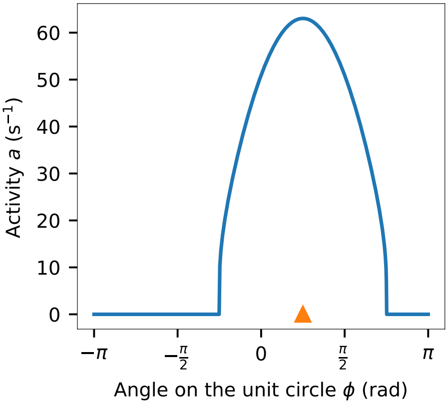


Preferred Directions in Higher Dimensions: Representing 2D Values

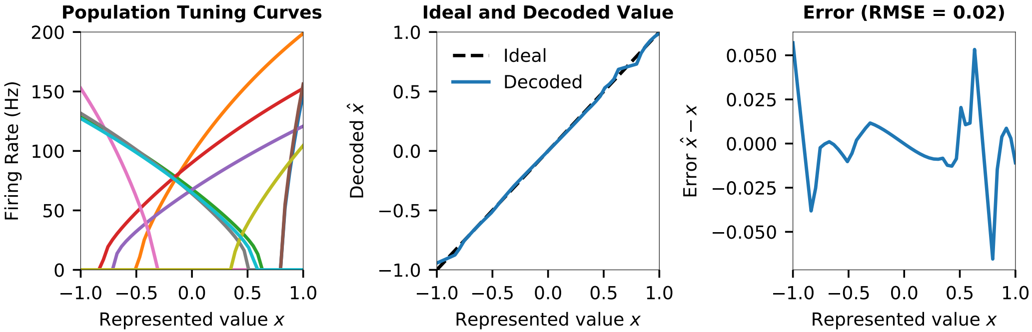


Preferred Directions in Higher Dimensions: Representing 2D Values



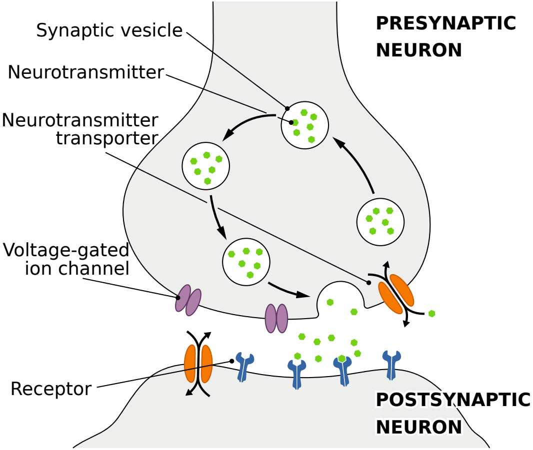


Decoding Without Taking Noise Into Account



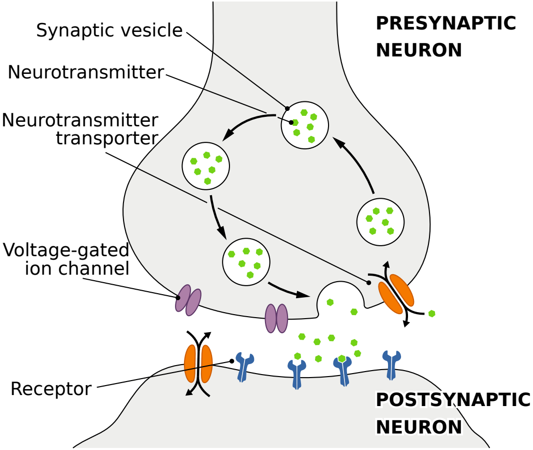
Sources of Noise in Biological Neural Networks

- **Axonal jitter**
  Active axonal spike propagation
- **Vesicle release failure**
  10-30% of pre-synaptic events cause post-synaptic current
- **Neurotransmitter per vesicle**
  Varying amounts of neurotransmitter
- **Ion channel noise**
  Ion-channels are “binary”, stochastic
- **Thermal noise**
- **Network effects**
  Simple, noise-free inhibitory/excitatory networks produce irregular spike trains



Sources of Noise in Biological Neural Networks

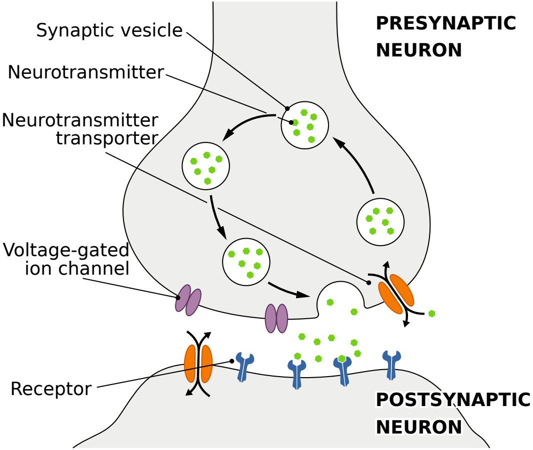
- **Axonal jitter**
  Active axonal spike propagation
- **Vesicle release failure**
  10-30% of pre-synaptic events cause post-synaptic current
- **Neurotransmitter per vesicle**
  Varying amounts of neurotransmitter
- **Ion channel noise**
  Ion-channels are "binary", stochastic
- **Thermal noise**
- **Network effects**
  Simple, noise-free inhibitory/excitatory networks produce irregular spike trains



- **How to model?**

Sources of Noise in Biological Neural Networks

- **Axonal jitter**
  Active axonal spike propagation
- **Vesicle release failure**
  10-30% of pre-synaptic events cause post-synaptic current
- **Neurotransmitter per vesicle**
  Varying amounts of neurotransmitter
- **Ion channel noise**
  Ion-channels are "binary", stochastic
- **Thermal noise**
- **Network effects**
  Simple, noise-free inhibitory/excitatory networks produce irregular spike trains

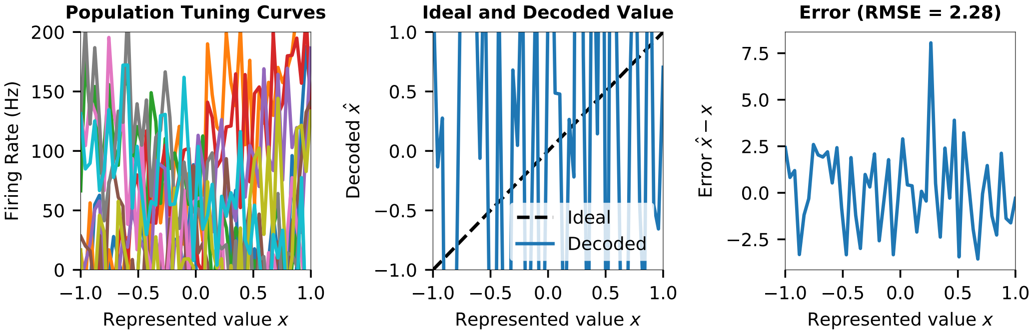


- **How to model?** Gaussian noise

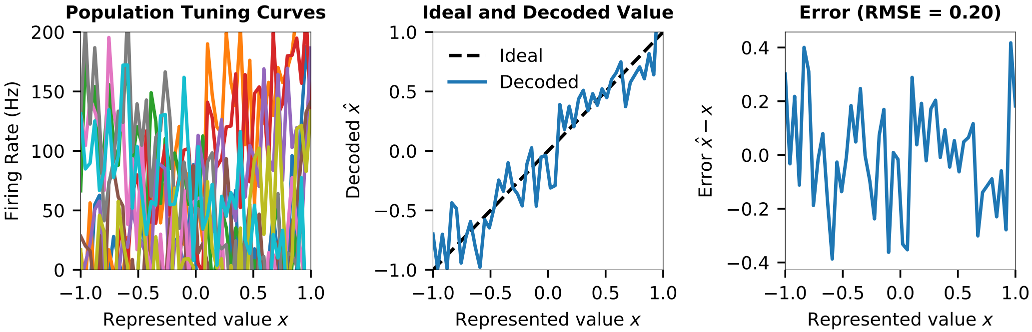
NEF Principle 0: Noise

NEF Principle 0 – Noise
Biological neural systems are subject to significant amounts of noise from various sources. Any analysis of such systems must take the effects of noise into account.

Decoding Noisy $\mathbf{A}$ Without Taking Noise Into Account



Decoding Noisy $\mathbf{A}$ Accounting for Noise

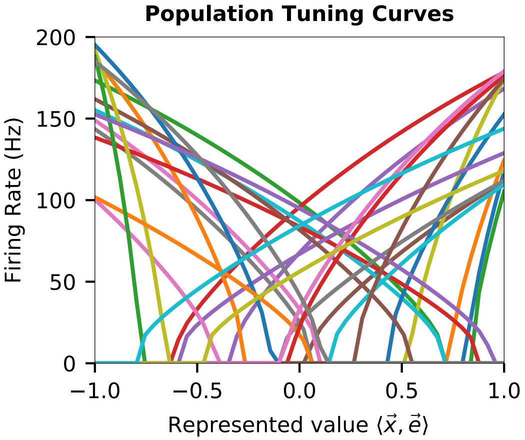


Summary: Building a model of neural representation (Encoding)

Encoding

- Select d, possible range $\mathbf{x} \in \mathbb{X}$, usually $\mathbb{X} = \left\{\mathbf{x} \mid \|\mathbf{x}\| \leq r, \mathbf{x} \in \mathbb{R}^d\right\}$ $(r = 1)$
- Select number of neurons n
- Select tuning curves, maximum rates $\Rightarrow \mathbf{e}_i,\ \alpha_i,\ J_i^{\mathrm{bias}}$
  - Sample $\mathbf{e}_i$ from unit-sphere
  - Uniformly distribute $\alpha_i,\ J_i^{\mathrm{bias}}$
- Encoding equation:

$$a_i(\mathbf{x}) = G\left[\alpha_i \langle \mathbf{e}_i, \mathbf{x} \rangle + J_i^{\mathrm{bias}}\right]$$



Summary: Building a model of neural representation (Decoding)

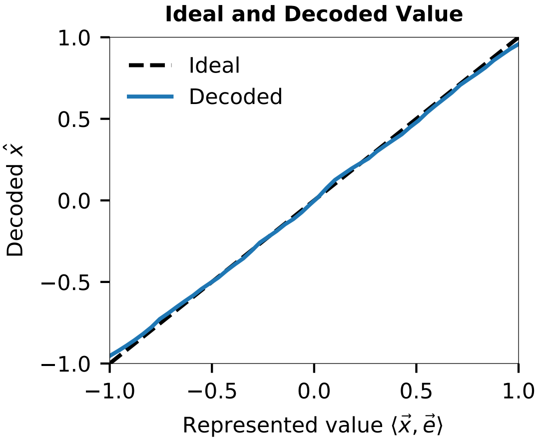
Decoding

- Uniformly sample N samples from $\mathbb{X}$, $\mathbf{X} = \left(\mathbf{x}_1, \dots, \mathbf{x}_N\right)$
- Compute $\mathbf{A}$, where $(\mathbf{A}) = a_i()$
- Decoder computation:

$$\mathbf{D} = \left(\mathbf{A}\mathbf{A}^{\mathrm{T}} + N\sigma^2\mathbf{I}\right)\mathbf{A}\mathbf{X}^T$$

- Decoding equation:

$$\hat{\mathbf{X}} = \mathbf{D}\mathbf{A}$$



Administration

- **Assignment 1 has been released.**

  The due date has been adjusted to January, 30.

- Some new potential times for office hours

  `Mon 15:30-16:30`, `Mon 16:30-17:30`, `Tue 15:00-16:00`,

  `Thu 11:30-12:30` (current slot), `Thu 12:30-13:30`

Image sources

Title slide
"The Ultimate painting."
Author: Clark Richert.
From Wikimedia.