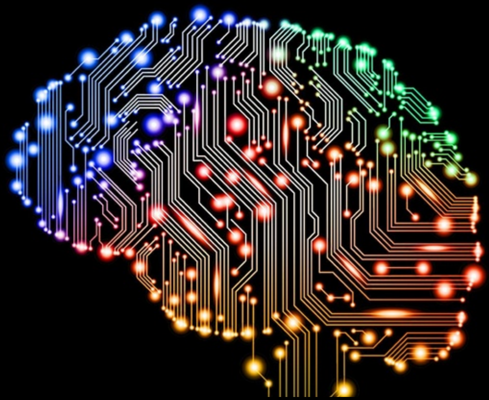
Kernel theories of networks and their use in neuroscience

Kameron Decker Harris

University of Washington
Paul G. Allen School of Comp Science & Engineering, Biology

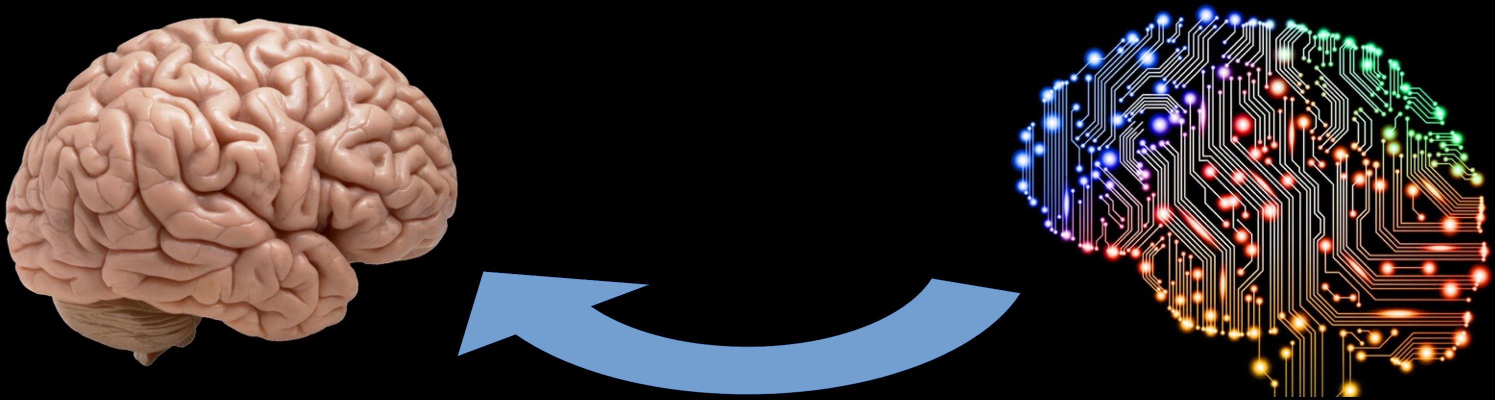
Western Washington University
Computer Science

Neuro



CS

slide idea: Guillaume Lajoie, others
images: flickr/_DJ_, neurobanter.com



Neuro

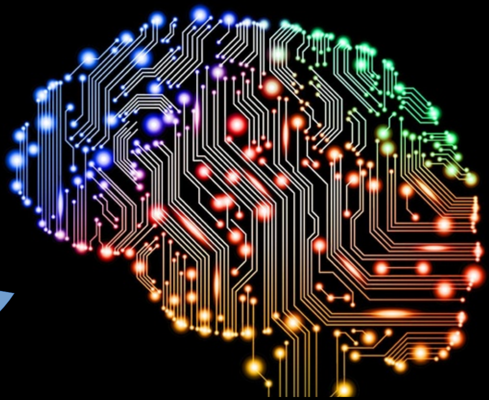
CS

slide idea: Guillaume Lajoie, others
images: flickr/_DJ_, neurobanter.com

slide idea: Guillaume Lajoie, others
images: flickr/_DJ_, neurobanter.com

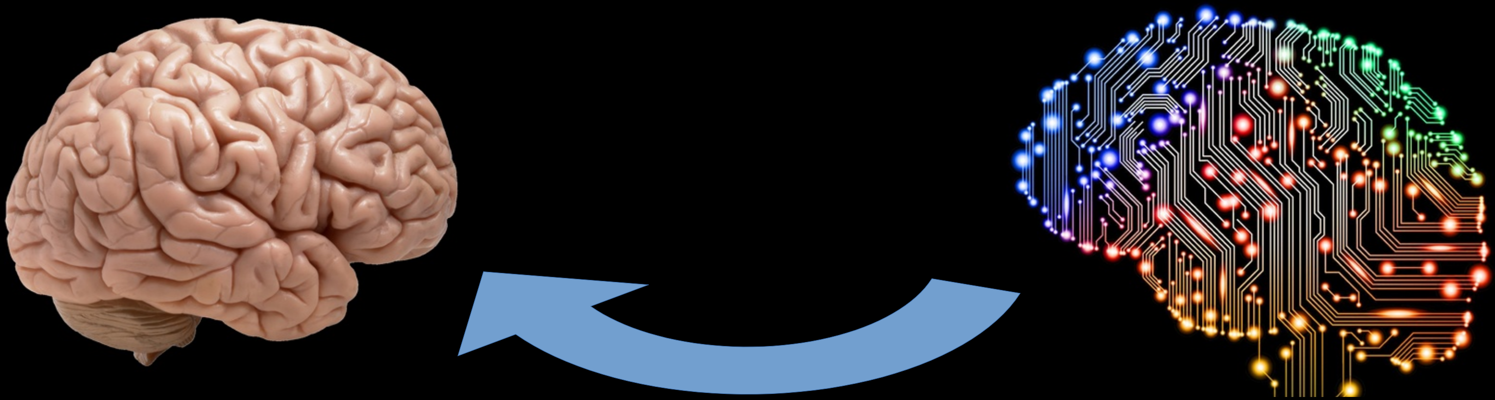
Neuro

Big data management
Analysis of experiments



CS

slide idea: Guillaume Lajoie, others
images: flickr/_DJ_, neurobanter.com



Neuro

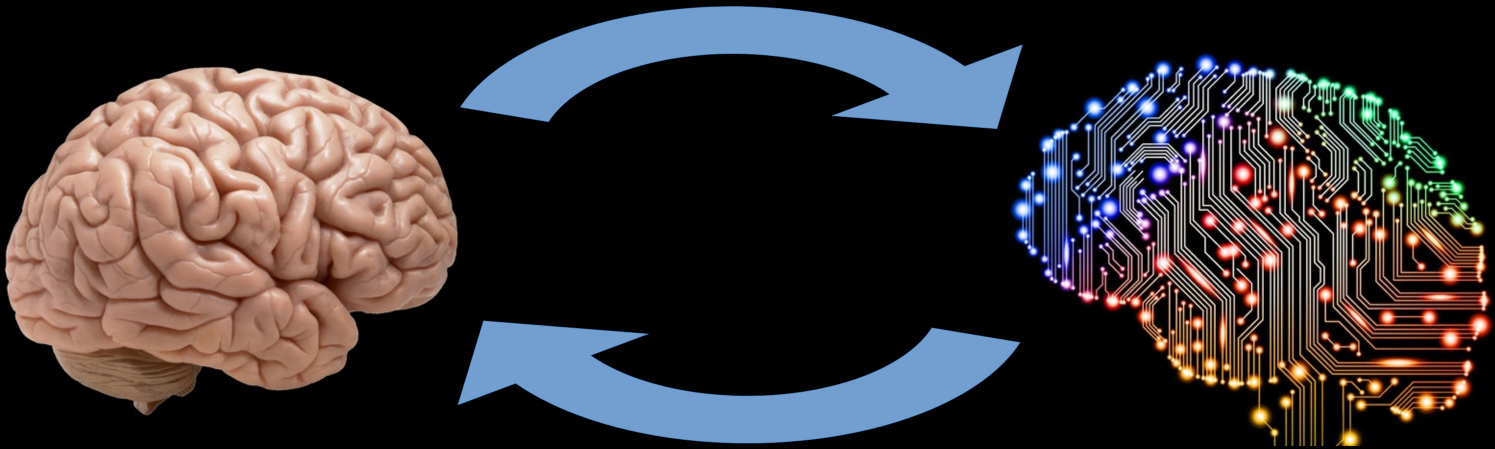
CS

Big data management

Analysis of experiments

Artificial neural networks

slide idea: Guillaume Lajoie, others
images: flickr/_DJ_, neurobanter.com



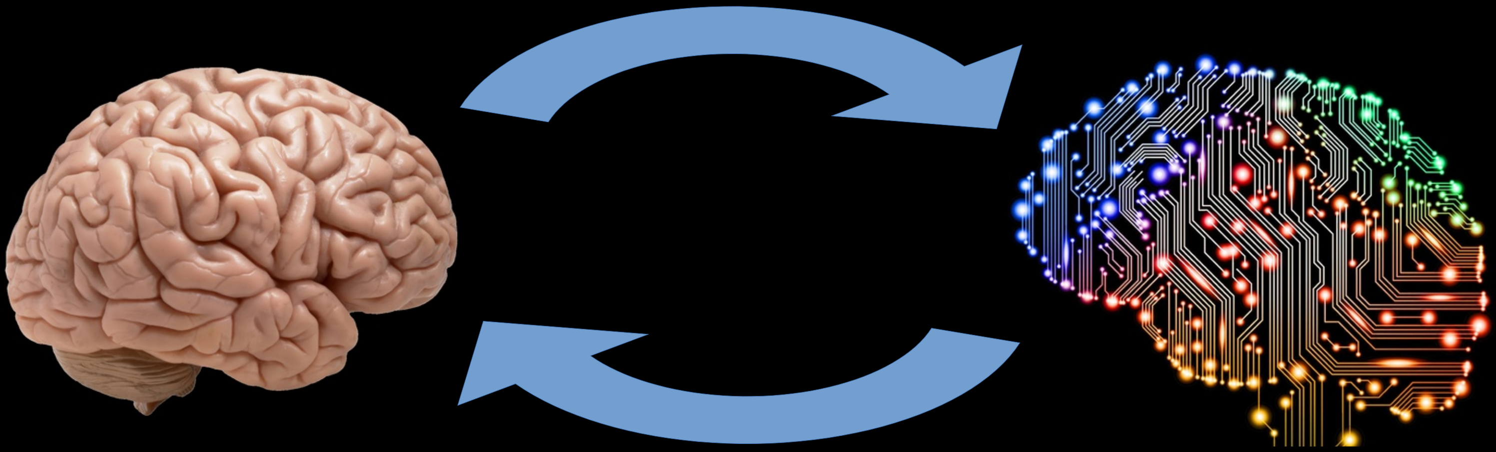
Neuro

CS

Big data management
Analysis of experiments
Artificial neural networks

slide idea: Guillaume Lajoie, others
images: flickr/_DJ_, neurobanter.com

Brain-inspired algorithms



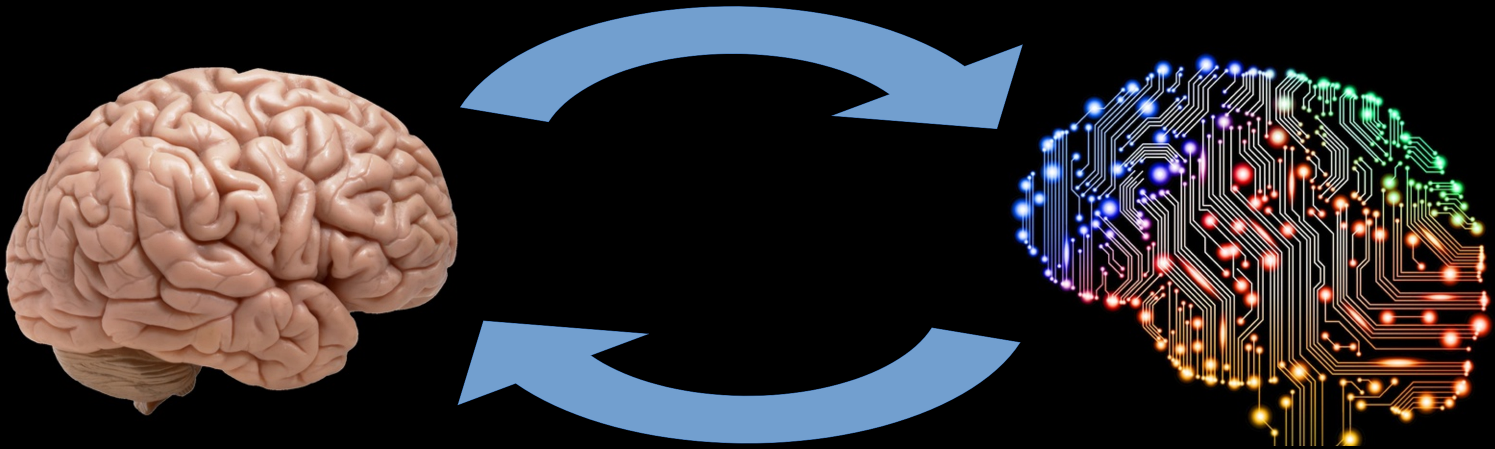
Neuro

CS

Big data management
Analysis of experiments
Artificial neural networks

slide idea: Guillaume Lajoie, others
images: flickr/_DJ_, neurobanter.com

Brain-inspired algorithms
Emphasis on dynamics



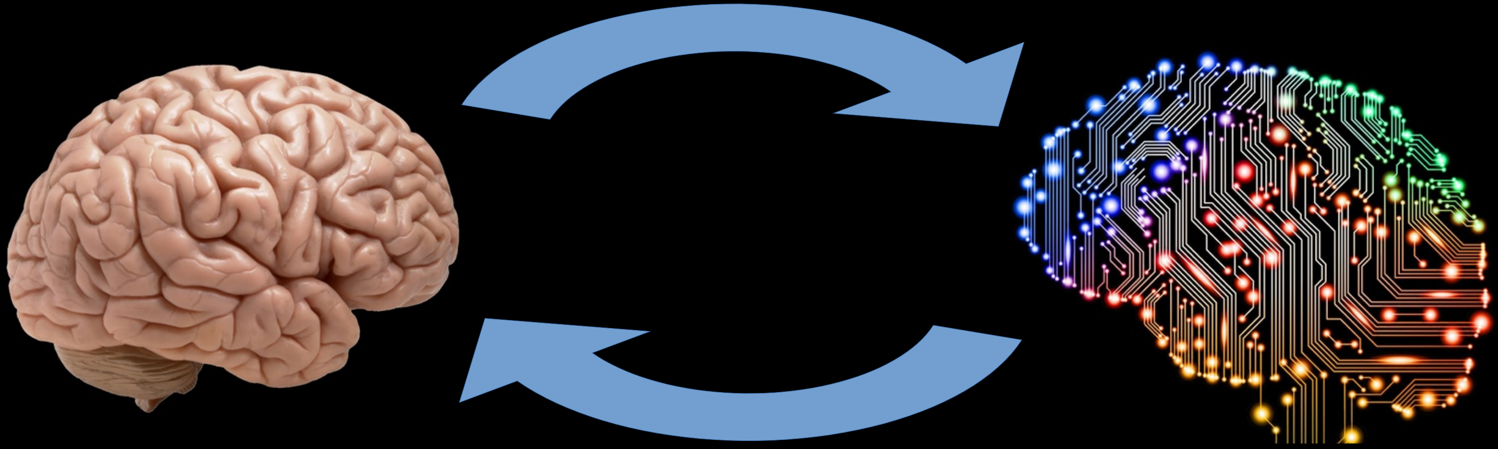
Neuro

CS

Big data management
Analysis of experiments
Artificial neural networks

slide idea: Guillaume Lajoie, others
images: flickr/_DJ_, neurobanter.com

Brain-inspired algorithms
Emphasis on dynamics
Alternate models of computation



Neuro

CS

Big data management
Analysis of experiments
Artificial neural networks

slide idea: Guillaume Lajoie, others
images: flickr/_DJ_, neurobanter.com

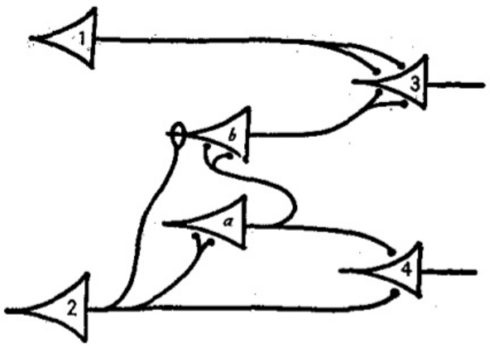
Overview

- Context: structure, randomness, & neuroscience
- Kernel theory:
  - Mathematical framework
  - Network sparsity → additive functions
  - Tuning curves → basis change
- Conclusions

Overview

- **Context: structure, randomness, & neuroscience**
- Kernel theory:
  - Mathematical framework
  - Network sparsity → additive functions
  - Tuning curves → basis change
- Conclusions

Structure and function



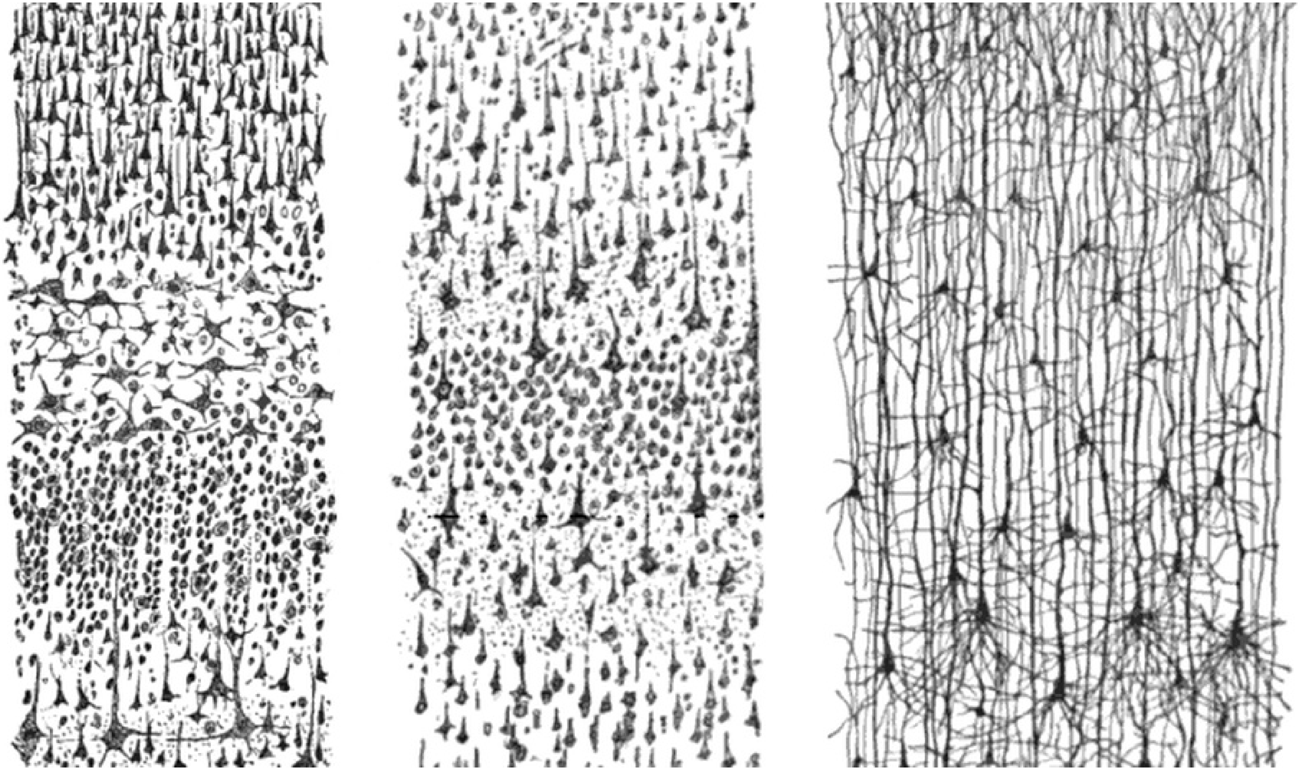
A LOGICAL CALCULUS OF THE
IDEAS IMMANENT IN NERVOUS ACTIVITY

WARREN S. MCCULLOCH AND WALTER PITTS

<u>1943</u>

“ for any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes ”

The messy reality of neuronal structure



Ramón y Cajal

Random vs stereotyped is not a dichotomy

Random vs stereotyped is not a dichotomy

- Some connections are stereotyped: CPGs, *Drosophila* Gal4 lines

Random vs stereotyped is not a dichotomy

- Some connections are stereotyped: CPGs, *Drosophila* Gal4 lines
- Some connections are nearly random: mushroom body, cortex?
  - Sophie Caron et al.

Random vs stereotyped is not a dichotomy

- Some connections are stereotyped: CPGs, *Drosophila* Gal4 lines
- Some connections are nearly random: mushroom body, cortex?
  - Sophie Caron et al.

Hypotheses:

Random vs stereotyped is not a dichotomy

- Some connections are stereotyped: CPGs, *Drosophila* Gal4 lines
- Some connections are nearly random: mushroom body, cortex?
  - Sophie Caron et al.

Hypotheses:

- Connections in many large networks are well-described by *random distributions* with *structure*

Random vs stereotyped is not a dichotomy

- Some connections are stereotyped: CPGs, *Drosophila* Gal4 lines
- Some connections are nearly random: mushroom body, cortex?
  - Sophie Caron et al.

Hypotheses:

- Connections in many large networks are well-described by *random distributions* with *structure*
- Plasticity and evolution modify these distributions

Overview

- Context: structure, randomness, & neuroscience
- **Kernel theory:**
  - **Mathematical framework**
  - Network sparsity → additive functions
  - Tuning curves → basis change
- Conclusions

Kernels: introduction & definition

$\mathcal{X}$ input vector l-dimensional $x \in \mathcal{X} \subseteq \mathbb{R}^l$ compact

$k(x, x')$ kernel function computes "similarity"

$k(x,x') = \langle \phi(x), \phi(x') \rangle$

requirements:

- $k(x,x') = k(x',x)$
- $K = \left(k(x_i, x_j)\right)_{i,j=1}^{n}$ $n \times n$ kernel matrix
  symmetric + positive definite
  $c^T K c > 0$
- k continuous

See book by Shawe-Taylor & Christianini

Examples of kernels

Examples of kernels

1) Linear

$$k(x, x') = \langle x, x' \rangle = \sum_{i=1}^{l} x_i x'_i$$

Examples of kernels

1) Linear $$k(x, x') = \langle x, x' \rangle = \sum_{i=1}^{l} x_i x'_i$$

2) Polynomial $$k(x, x') = (c + \langle x, x' \rangle)^d$$

Examples of kernels

1) Linear

$$k(x, x') = \langle x, x' \rangle = \sum_{i=1}^{l} x_i x'_i$$

2) Polynomial

$$k(x, x') = (c + \langle x, x' \rangle)^d$$

3) Radial basis function

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

universal

bandwidth

Eigendecomposition: Mercer's theorem

For a kernel which is symmetric, positive definite, continuous on a compact domain:

PD matrix

$$k(x, x') = \sum_{i=1}^{\infty} \lambda_i \psi_i(x) \psi_i(x')$$

eigenvalues

$\geqslant 0$

eigenvectors

orthonormal basis for $L_2(\mathcal{X})$

Utility of kernel algorithms

Utility of kernel algorithms

- Kernels implicitly represent inputs in higher-dimensional features space called *reproducing kernel Hilbert space (RKHS)*

$$k(x, x') = \langle \phi(x), \phi(x') \rangle$$

Utility of kernel algorithms

- Kernels implicitly represent inputs in higher-dimensional features space called *reproducing kernel Hilbert space (RKHS)*

$$k(x, x') = \langle \phi(x), \phi(x') \rangle$$

- Algorithms can leverage this and just work with kernel matrix: SVM, **ridge regression**, PCA, etc.

K n×n

$$f(x) = \mathbf{y}(K + \alpha I)^{-1}\kappa(x), \quad \kappa(x) = (k(x_i, x))_{i=1}^{n}$$

n+n matrix Solve

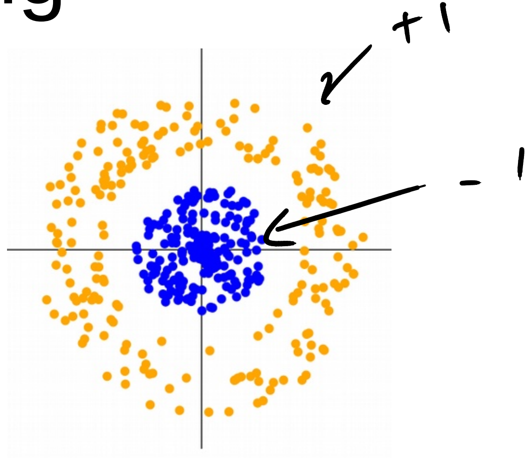
$\mathrm{Tr}\left(K (K+\alpha I)^{-1} \right)$

$\alpha \nearrow$ dim $\searrow$

Kernels and learning

Given: examples $(\mathbf{x}^i, y_i)_{i=1}^n$

Find: f so that $f(\mathbf{x}) \approx y$

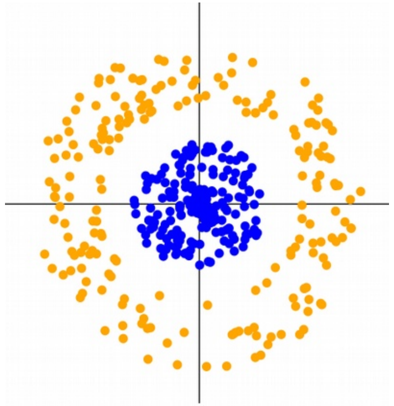


Shawe-Taylor & Christianini; Wahba; Bartlett & Mendelson

Kernels and learning

Given: examples $(\mathbf{x}^i, y_i)_{i=1}^n$

Find: f so that $f(\mathbf{x}) \approx y$



Key parameters

RKHS norm of the target function

$$f \in \mathcal{H} : \quad \|f\|_{\mathcal{H}}$$

Dimensionality of the kernel matrix a.k.a. *Rademacher complexity*

significant eigenvalues of *K* ← depended on X

Shawe-Taylor & Christianini; Wahba; Bartlett & Mendelson

Kernel theory of networks

Related review: “Randomness in neural networks” by Scardapane & Wang (2017)

Kernel theory of networks

Developed for big datasets, i.e. *n* huge $\vec{y}(K + \alpha I)^{-1}$

- Rahimi & Recht (2008) **random features** … sketching *K*
- older work in Gaussian processes by Neal (1996), Williams (1997)

Related review: “Randomness in neural networks” by Scardapane & Wang (2017)

Kernel theory of networks

Developed for big datasets, i.e. *n* huge

- Rahimi & Recht (2008) **random features** … sketching *K*
- older work in Gaussian processes by Neal (1996), Williams (1997)

Exciting work tries to understand success of ANNs trained via gradient descent

- neural tangent kernel (NTK) – Jacot, Gabriel, Hongler, 2018; Arora et al, 2019
- convolutional kernel networks (CKN) – Mallat; Bruna; Harchaoui; Chizat et al
- interpolation & double descent – Belkin et al; Mei & Montanari

Related review: “Randomness in neural networks” by Scardapane & Wang (2017)

Kernel theory of networks

Developed for big datasets, i.e. *n* huge

- Rahimi & Recht (2008) **random features** … sketching *K*
- older work in Gaussian processes by Neal (1996), Williams (1997)

Exciting work tries to understand success of ANNs trained via gradient descent

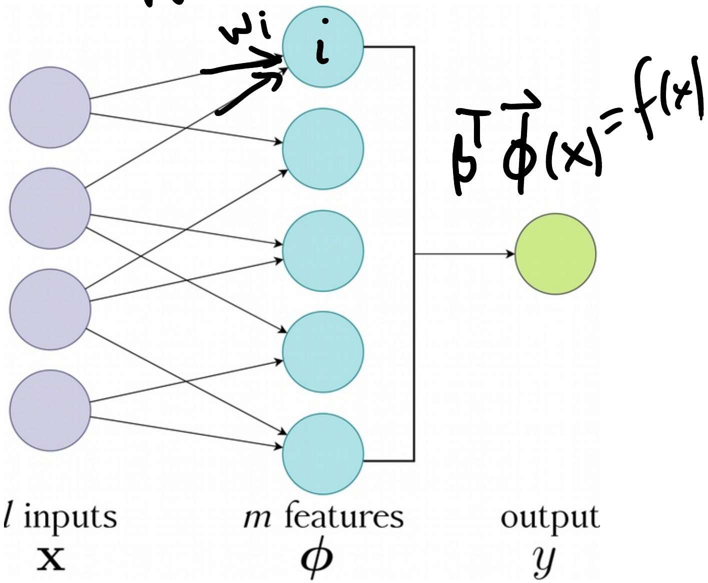
- neural tangent kernel (NTK) – Jacot, Gabriel, Hongler, 2018; Arora et al, 2019
- convolutional kernel networks (CKN) – Mallat; Bruna; Harchaoui; Chizat et al
- interpolation & double descent – Belkin et al; Mei & Montanari

Barely applied in neuroscience, so far

Related review: "Randomness in neural networks" by Scardapane & Wang (2017)

From random network to kernel

W



class. ± 1

reg. $\mathbb{R}$

$$\underbrace{\phi_i(x) = h(w_i^T x)}$$

Assume $w_i \sim \mu$ iid random

Then assuming h Lipschitz, 2^{nd} moments

$$\frac{1}{m}\sum_{i=1}^{m} \phi_i(x)\,\phi_i(x')$$

$$= \frac{1}{m}\vec{\phi}(x)^T \vec{\phi}(x')$$

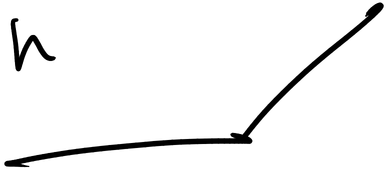
$$m \to \infty \quad \longrightarrow \mathbb{E}[\phi(x)\,\phi(x')] = \int h(\underbrace{w^T x}_{\text{scalar}})\,h(w^T x')\, d\mu(w)$$

scalars

$$\equiv k(x, x')$$

Convergence rate to kernel

h



Harris, 2019 (informal version)

Claim *The random map* $\frac{1}{m}\phi(\mathbf{x})^\intercal\phi(\mathbf{x}')$ *with* κ*-Lipschitz nonlinearity uniformly approximates* $k_D^{\mathrm{dist}}(\mathbf{x}, \mathbf{x}')$ *to error* ϵ *using* $m = \Omega(\frac{l\kappa^2}{\epsilon^2}\log\frac{C}{\epsilon})$ *many features*

Examples of random feature kernels

Examples of random feature kernels

Gaussian $w \sim N(0, \sigma^{-2} I)$ + Fourier $\exp(i w^\intercal x)$

FT of Gaussian

= Gaussian

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

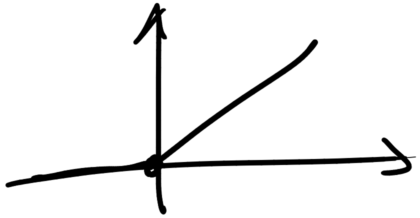
RBF kernel
Rahimi & Recht, 2008

Examples of random feature kernels

Gaussian $w \sim N(0, \sigma^{-2} I)$ + Fourier $\exp(i w^\intercal x)$

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

RBF kernel
Rahimi & Recht, 2008



Gaussian + rectified polynomial

$$k(x, x') = \|x\|^p \|x'\|^p \, J_p\left(\arccos \frac{x^\intercal x'}{\|x\| \|x'\|}\right)$$

dot product kernel
(RBF on sphere)
Cho & Saul, 2009

Examples of random feature kernels

Gaussian $w \sim N(0, \sigma^{-2} I)$ + Fourier $\exp(i w^\intercal x)$

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

RBF kernel
Rahimi & Recht, 2008

Gaussian + rectified polynomial

$$k(x, x') = \|x\|^p \|x'\|^p \, J_p\left(\arccos \frac{x^\intercal x'}{\|x\| \|x'\|}\right)$$

dot product kernel
(RBF on sphere)
Cho & Saul, 2009

Almost always take uncorrelated, Gaussian weights
No network or input <u>structure</u>

The rest of the talk

Build in structure that occurs in neural systems

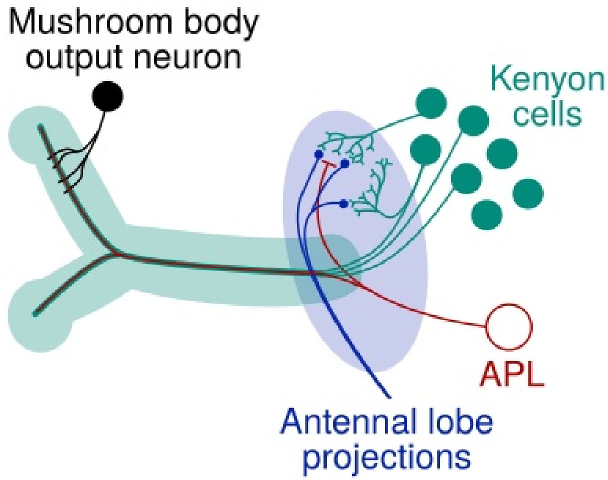
Show what the kernel theory says

Overview

- Context: structure, randomness, & neuroscience
- **Kernel theory:**
  - Mathematical framework
  - **Network sparsity → additive functions**
  - Tuning curves → basis change
- Conclusions

Harris. “Additive function approximation in the brain.” NeurIPS Neuro/AI workshop, 2019
Litwin-Kumar, Harris, Axel, Sompolinsky, Abbott. “Optimal degrees of synaptic connectivity.” Neuron, 2017

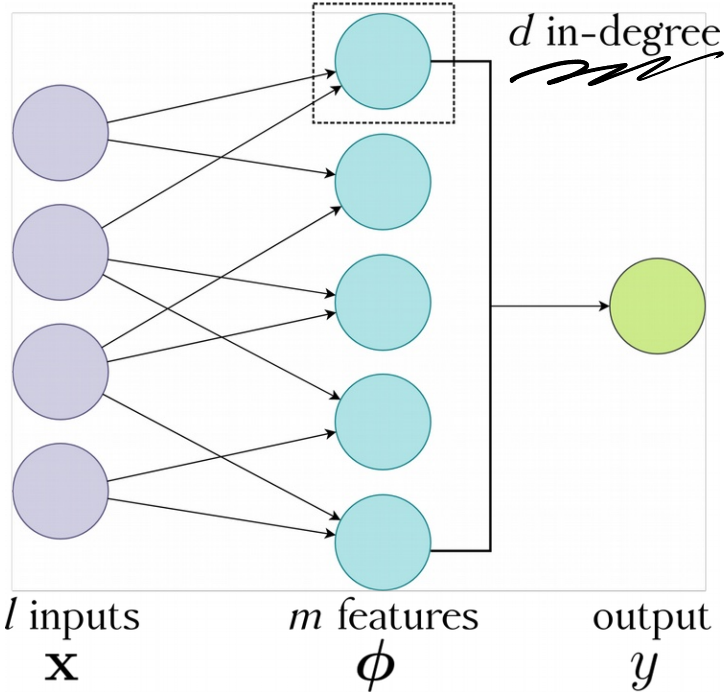
Olfactory network of *Drosophila*



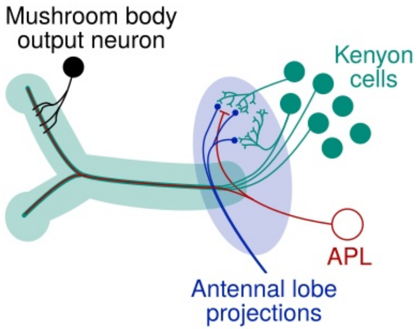
Output neuron decides:
good smell or bad smell?

Common brain network structure: 2-layer sparse expansion

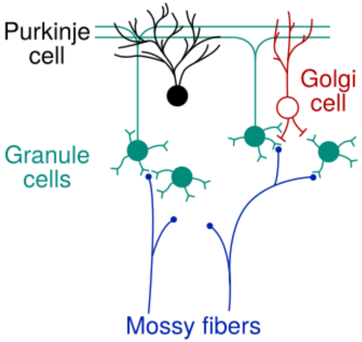
$d < l$



Mushroom body
d = 7



Cerebellum
d = 4

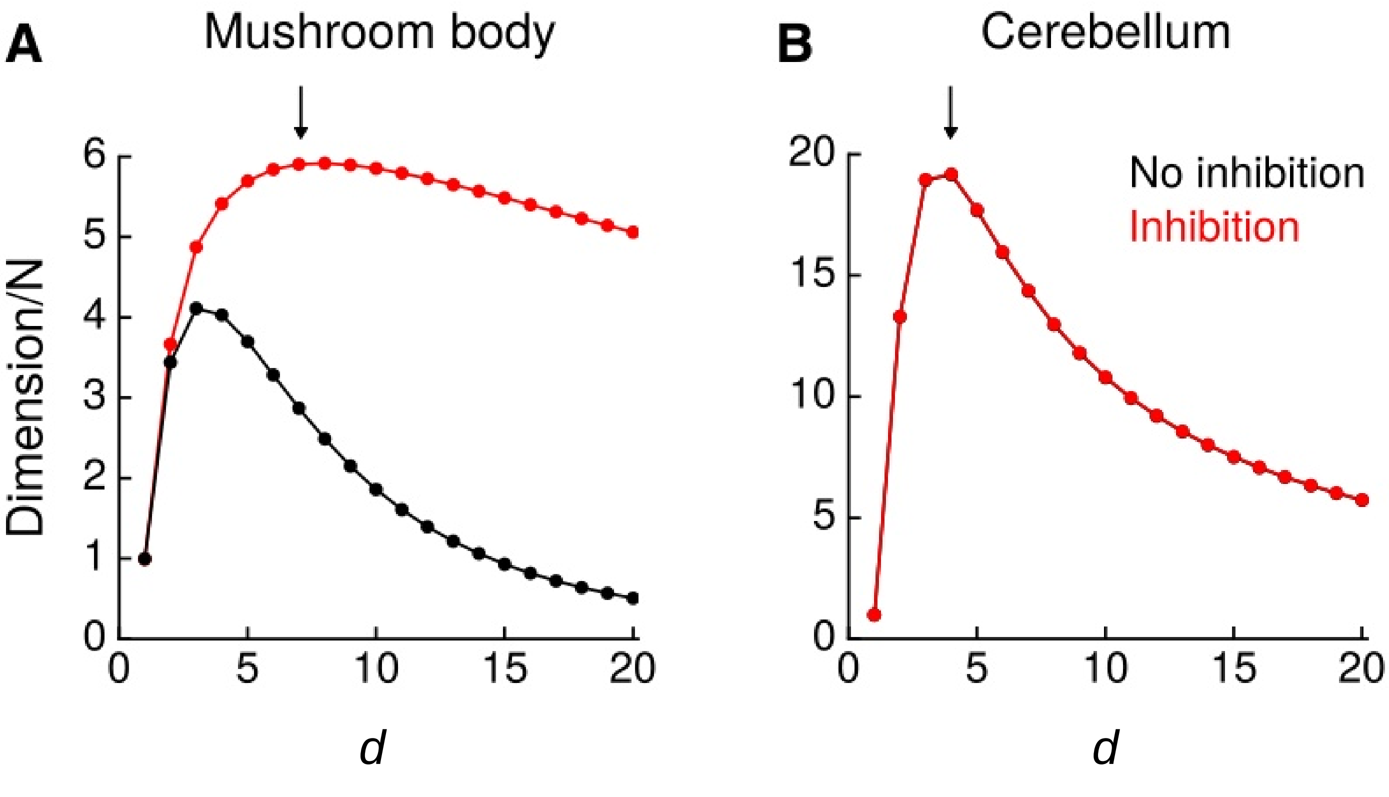


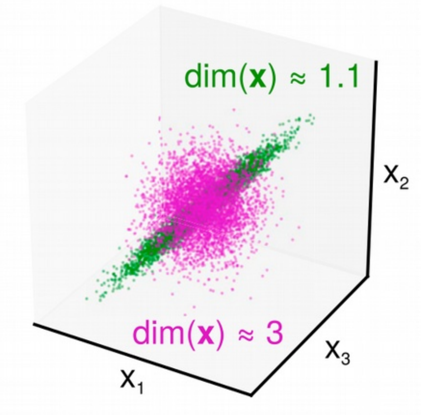
Litwin-Kumar, Harris, Axel, Sompolinsky, Abbott

Marr; Albus; Hansel & van Vreeswijk; Rigotti et al; Barak et al; Babadi & Sompolinsky; Cayco-Gajic, Clopath, Silver; Dasgupta, Stevens, Navlakha

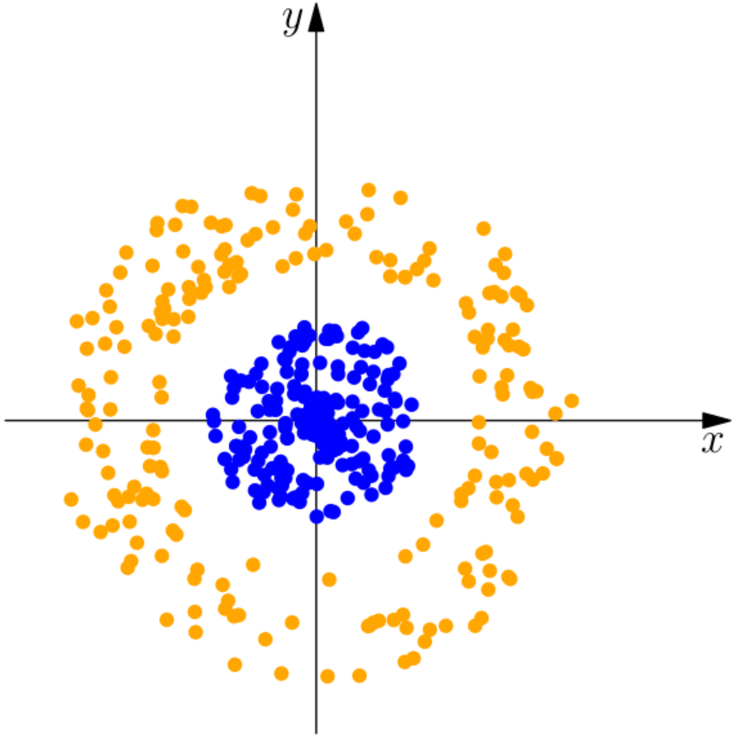
Sparsity under constraints = max dimensionality

Arrows = avg degree observed in brains

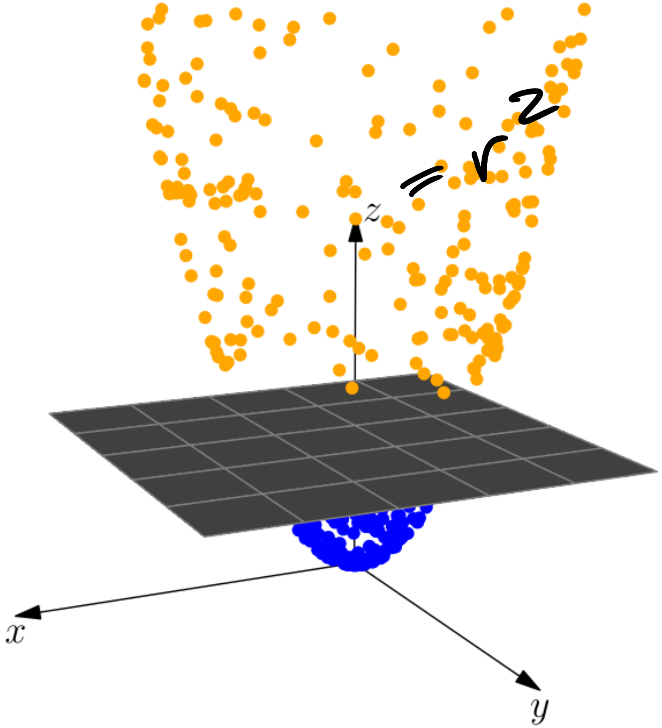




Dimensionality can help **or hurt** learning

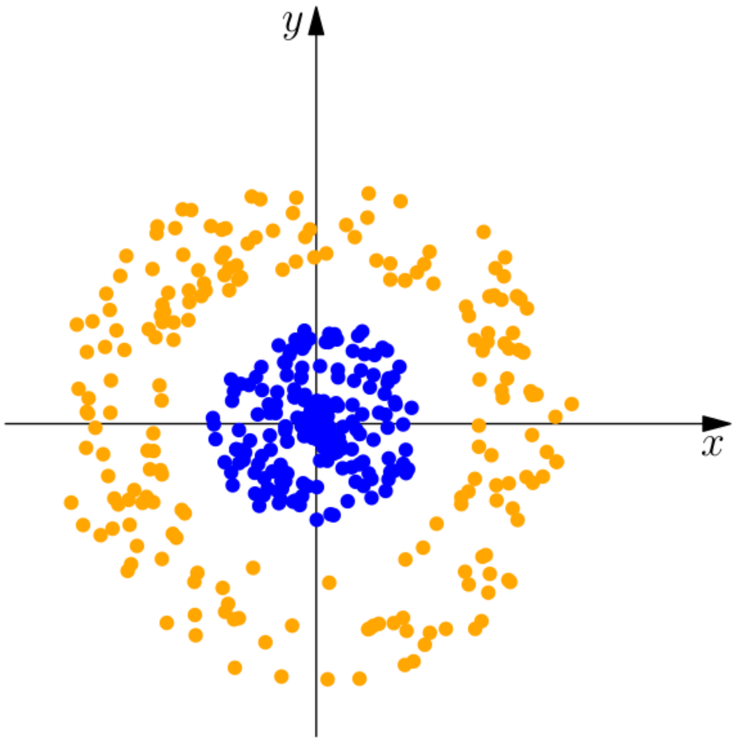


Nonlinear map
2-D to 3-D

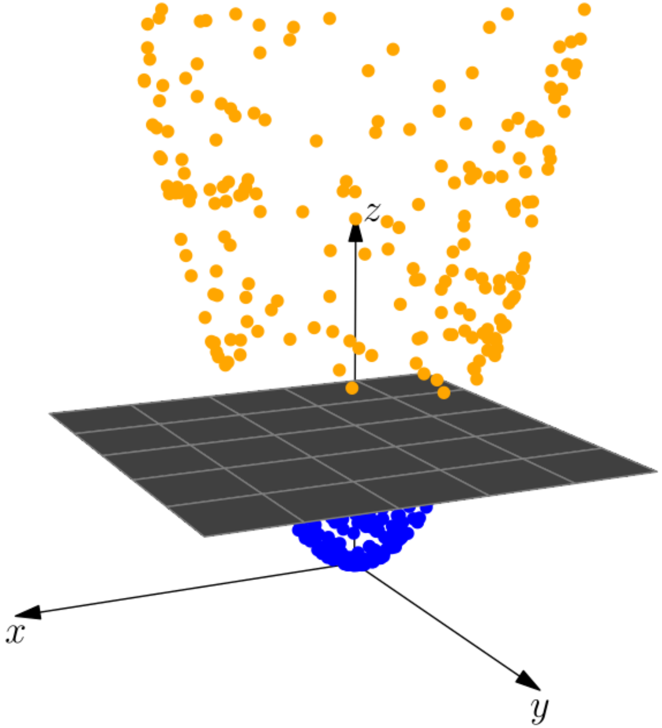


https://www.researchgate.net/publication/262800142_Lecture_Notes_on_the_Kernel_Trick_I

Dimensionality can help **or hurt** learning



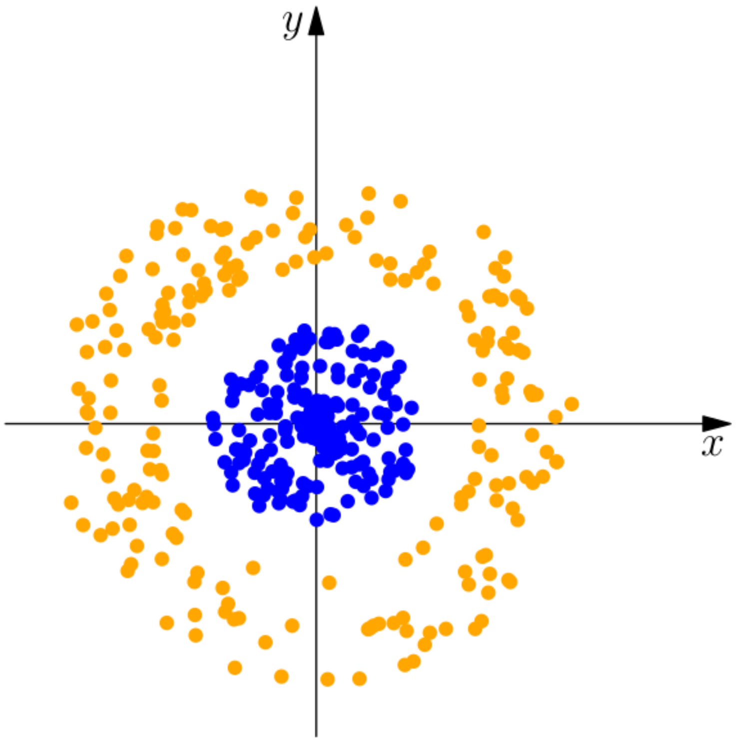
Nonlinear map
2-D to 3-D



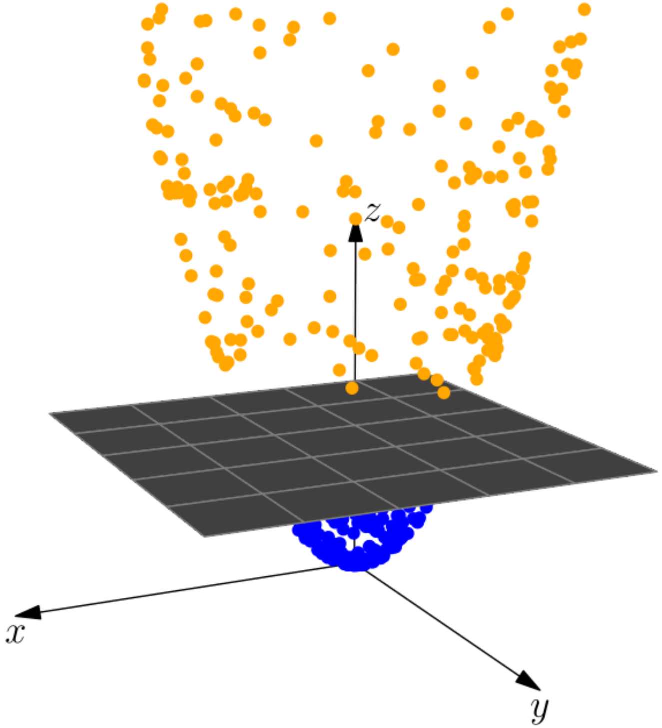
Expectation:

https://www.researchgate.net/publication/262800142_Lecture_Notes_on_the_Kernel_Trick_I

Dimensionality can help **or hurt** learning



Nonlinear map
2-D to 3-D

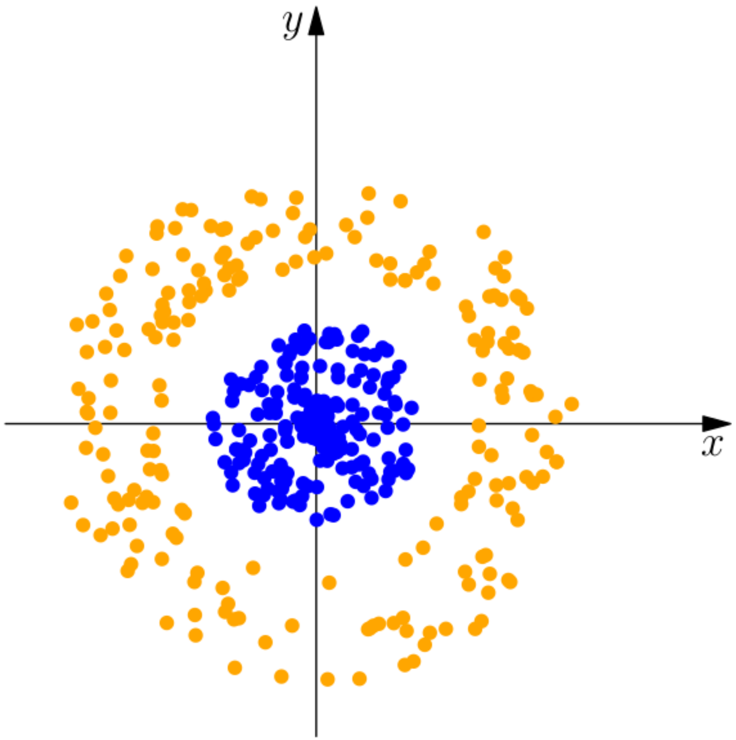


Expectation:

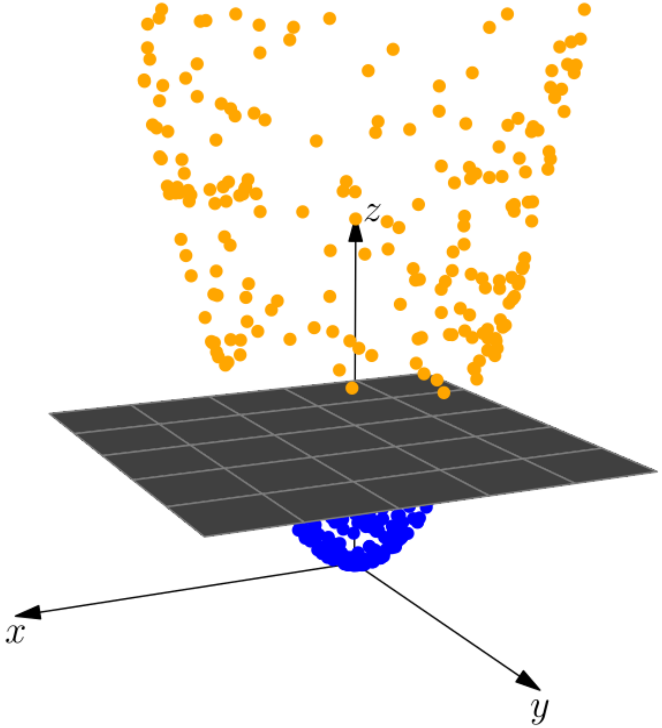
- dimensionality ↑ pattern separation ↑

https://www.researchgate.net/publication/262800142_Lecture_Notes_on_the_Kernel_Trick_I

Dimensionality can help **or hurt** learning



Nonlinear map
2-D to 3-D



Expectation:

- dimensionality ↑ pattern separation ↑
- dimensionality ↑ generalization ↓

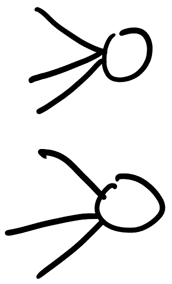
https://www.researchgate.net/publication/262800142_Lecture_Notes_on_the_Kernel_Trick_I

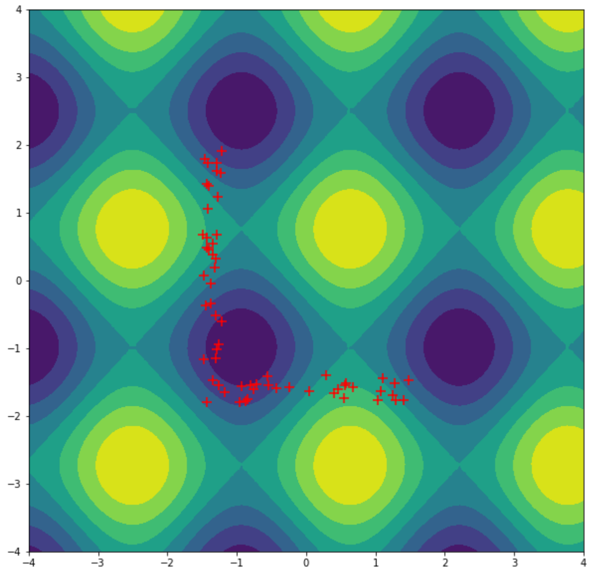
Kernels: *sparse networks* = ***additive*** functions

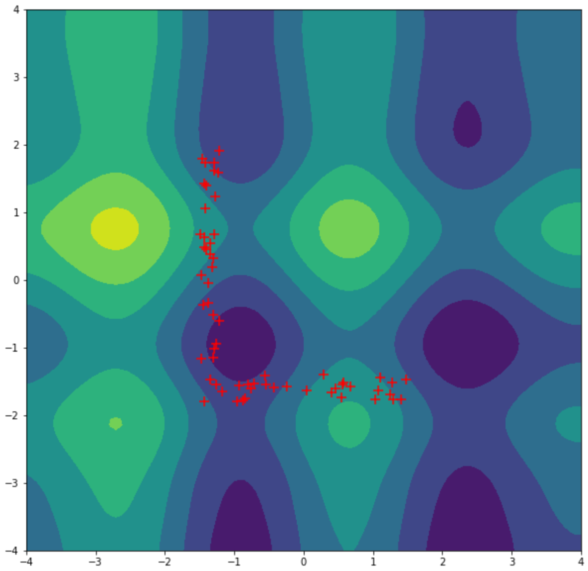
Additive functions are constrained, hence low-dimensional (Stone, 1985 & ‘86)

$d = 3$: $$f(\mathbf{x}) = f_1(x_1, x_3, x_4) + f_2(x_1, x_4, x_{11}) + \ldots$$

$= \ell$ -dim

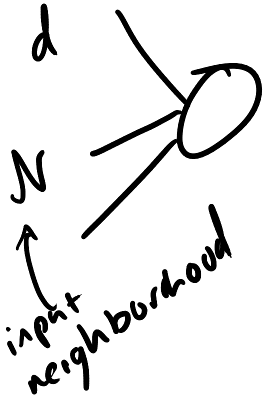






Sparse network kernels

$$k(x,x') = \mathbb{E}\left[\phi(x)\phi(x')\right] = \mathbb{E}\left[\mathbb{E}\left[\phi(x_{\mathcal{N}})\phi(x'_{\mathcal{N}}) \mid \mathcal{N}\right]\right]$$



$d=2$

$\ell=4$

$\binom{4}{2}$

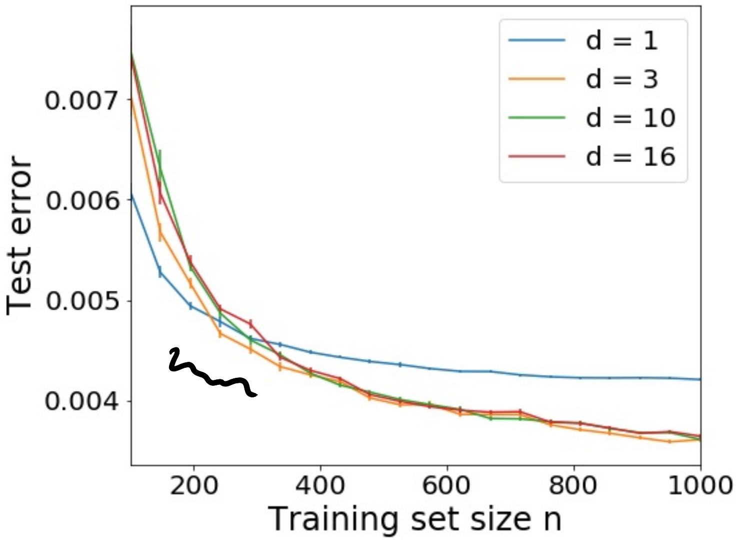
$$k_d^{reg}(x,x') = \frac{1}{\binom{\ell}{d}} \sum_{\substack{\mathcal{N}:|\mathcal{N}|=d \\ \subset[\ell]}} k_d(x_{\mathcal{N}}, x'_{\mathcal{N}})$$

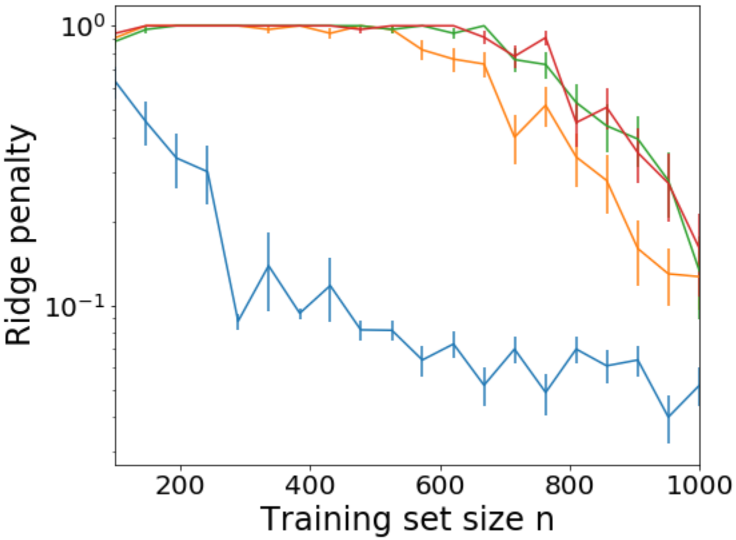
additive model order d

$d \sim 7$

$$\mathcal{H} = \mathcal{H}_1 \oplus \mathcal{H}_2 \oplus \cdots$$

Simulations confirm sparsity advantage





Target function:
Random linear + degree 3 polynomial

Overview

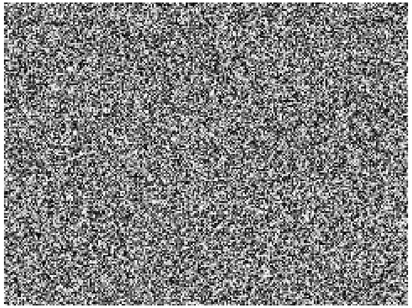
- Context: structure, randomness, & neuroscience
- Kernel theory:
  - Mathematical framework
  - Network sparsity → additive functions
  - **Tuning curves → basis change**
- Conclusions

Biraj Pandey
NSF Grad Fellow
Applied Math

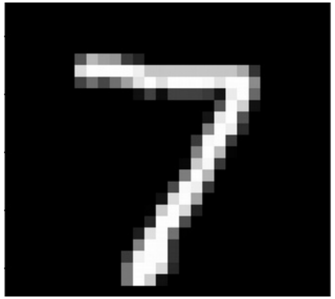
Bing Brunton
Biology

“Random features for structured input”

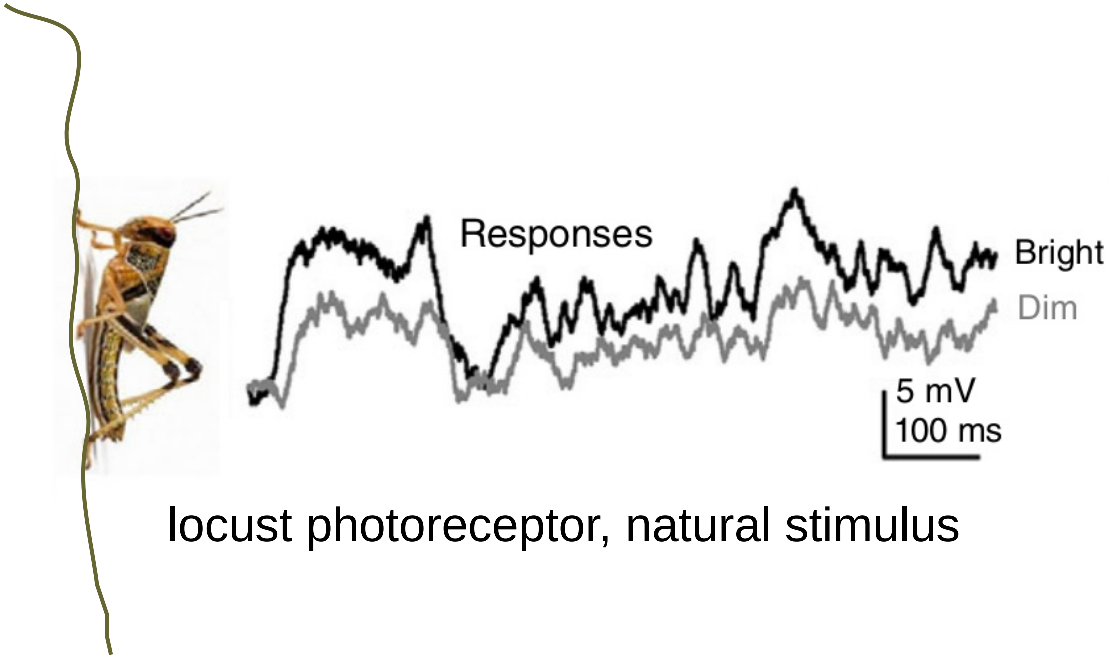
Defining “structured input”



white noise



MNIST

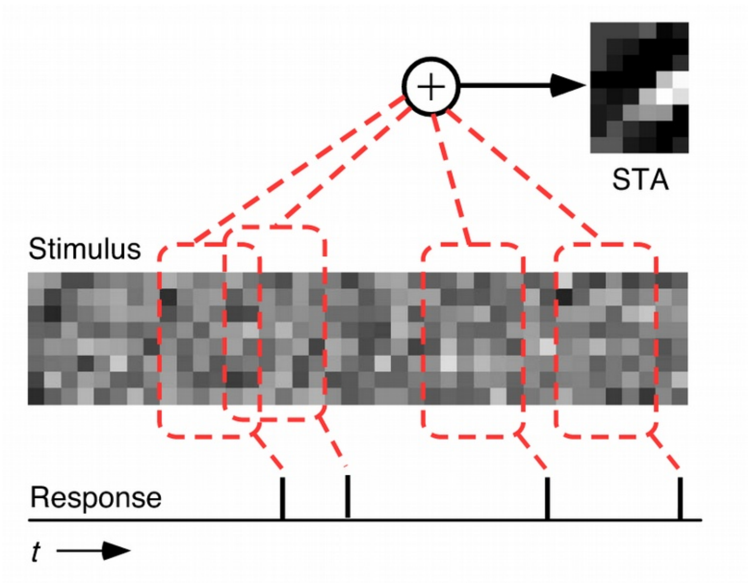


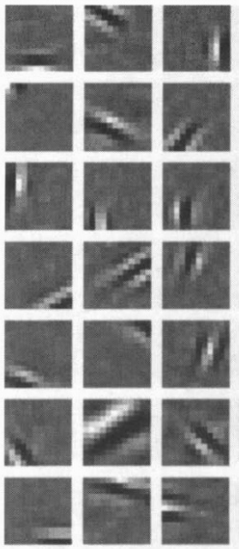
locust photoreceptor, natural stimulus

LeCun; Juusola & Song, 2017

Tuning curves occuring in nature

Stimulus that best drives a neuron, i.e. its **receptive field**

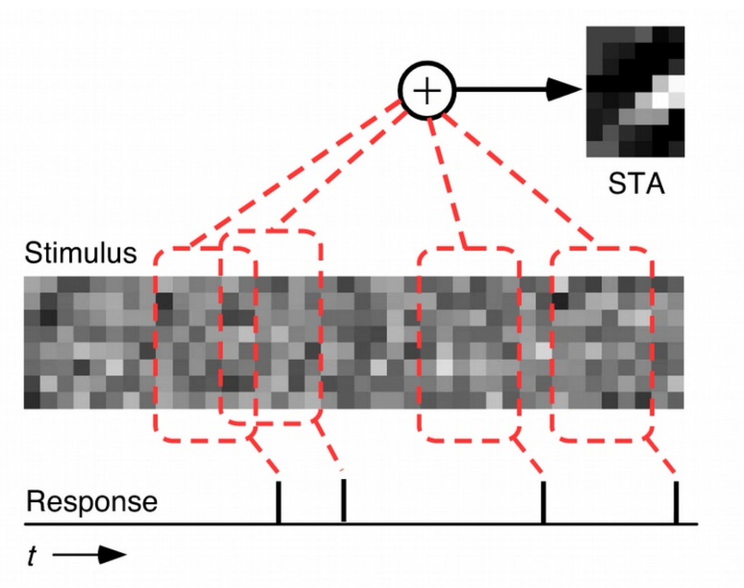


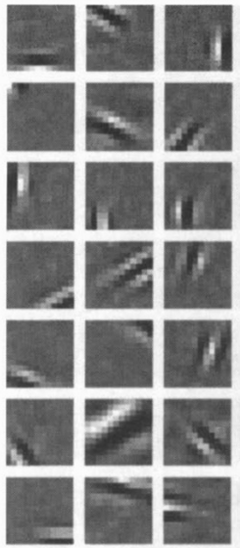


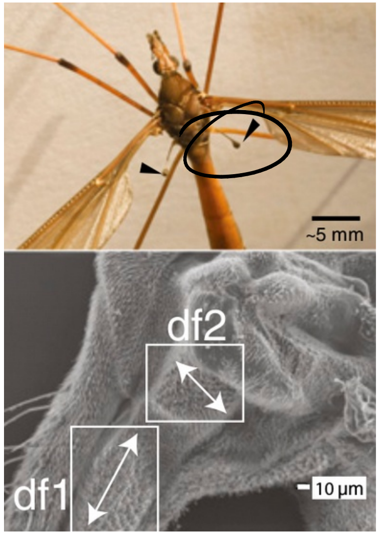
Schwartz et al; Olshausen & Field; Fox, Fairhall, Daniel

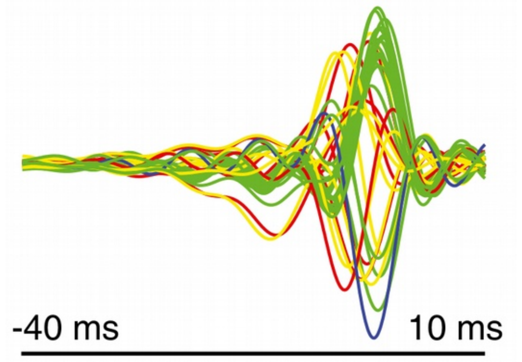
Tuning curves occuring in nature

Stimulus that best drives a neuron, i.e. its **receptive field**







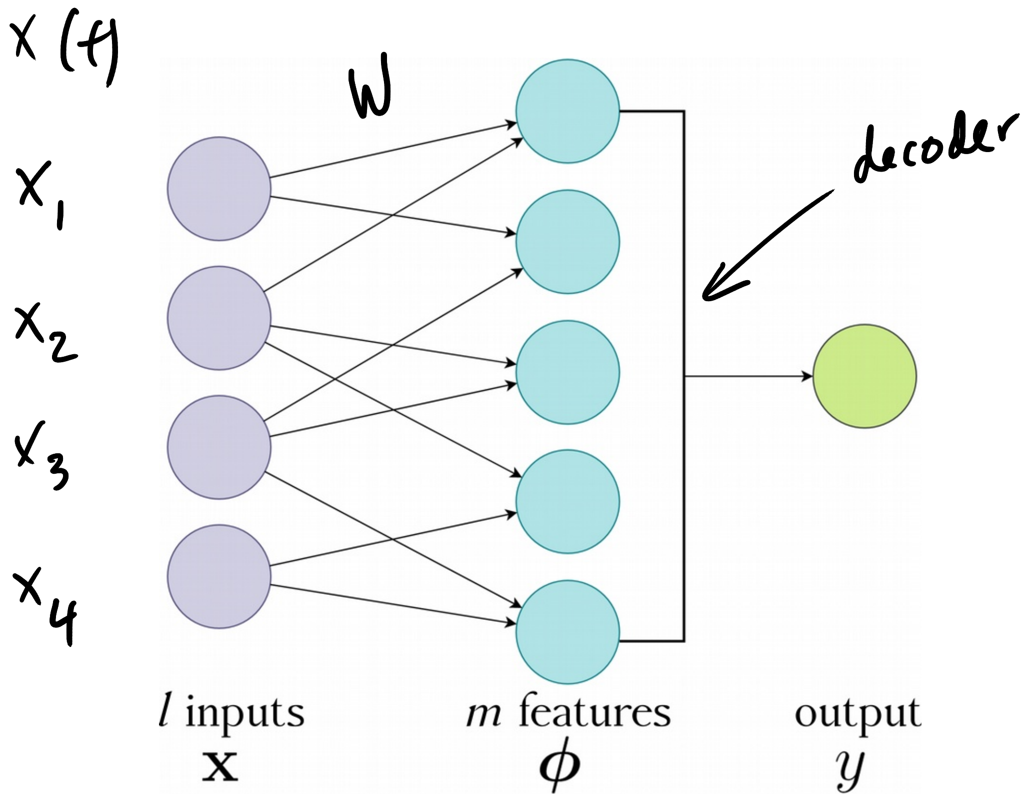


Schwartz et al; Olshausen & Field; Fox, Fairhall, Daniel

Linear-nonlinear model neurons

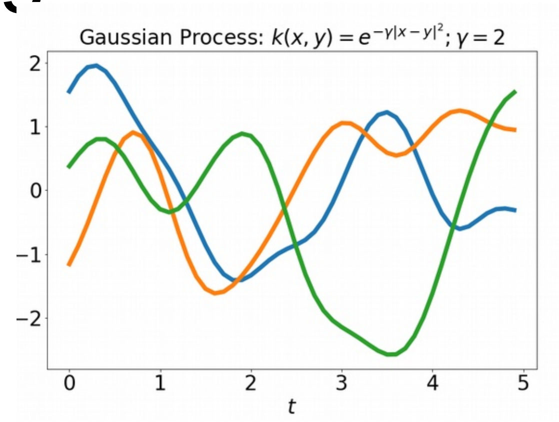
weights onto inputs

= linear filters



Theory of random tuning curves

Model **tuning curves** as random function drawn from a Gaussian process (GP)



$w(t)$ $\quad \mathbb{E}[w(t)] = 0$

$C(t,t') = \mathbb{E}[w(t)\, w(t')]$

<u>Mercer's thm</u>: $w(t) = \sum_{i=1}^{\infty} \sqrt{\lambda_i}\, G_i\, \psi_i(t)$ ← eigenfunctions

$G_i \sim \mathcal{N}(0,1)$

<u>Discrete</u>

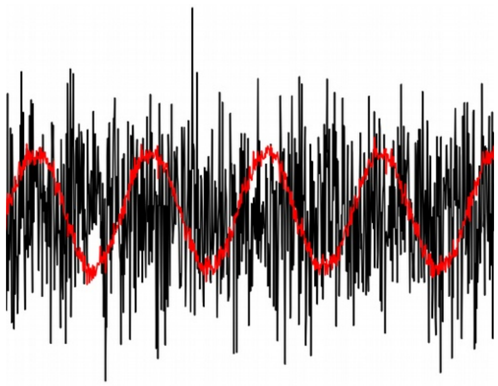
diagonal weights

$w = U D g$

orthogonal basis

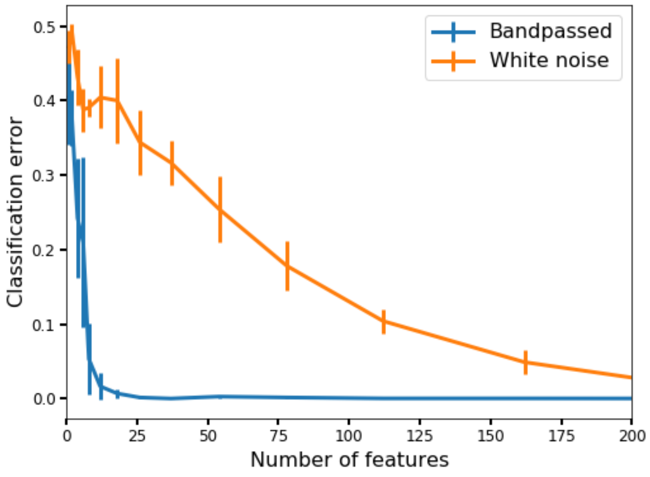
$w^T x = g^T D U^T x = g^T D \tilde{x}$ new basis

Example: frequency detection in timeseries

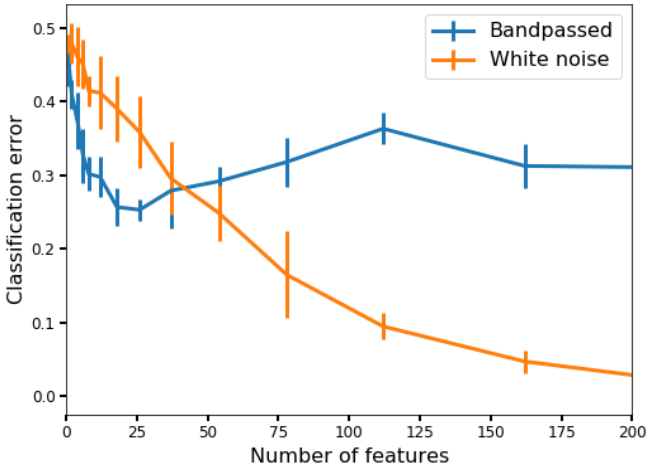


50 Hz signal
noise

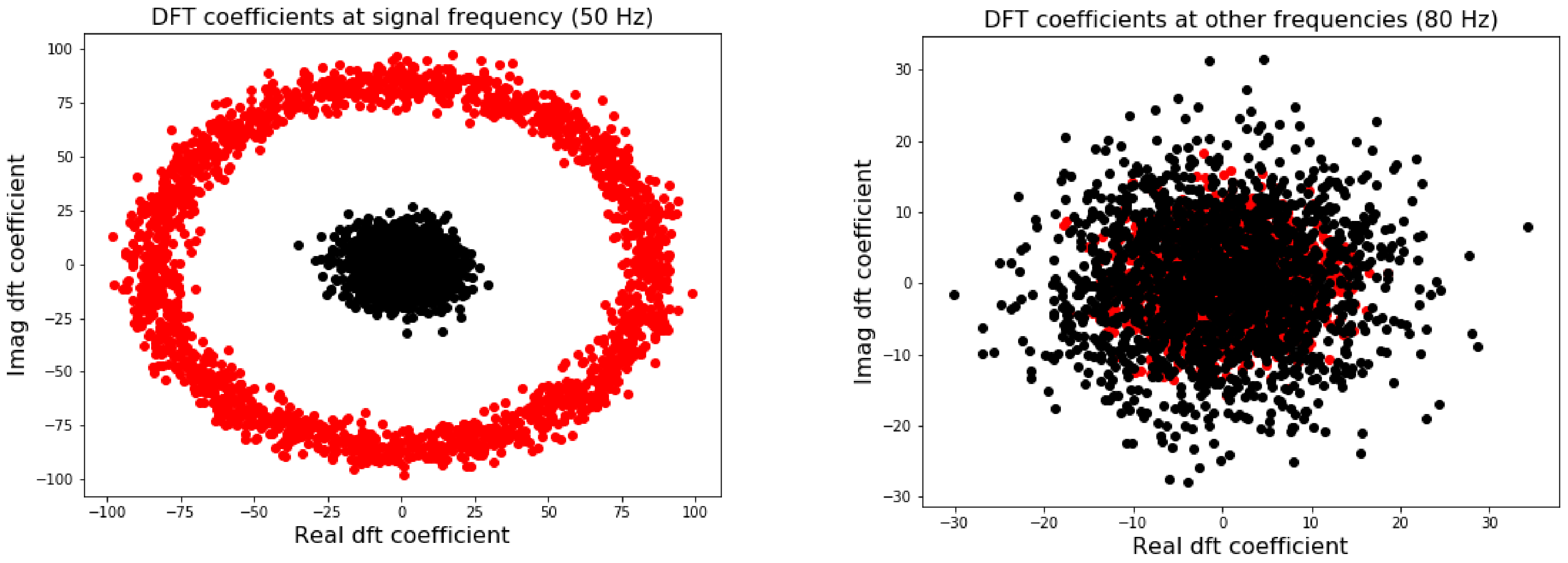
20 – 80 Hz
bandpass



70 – 120 Hz
bandpass

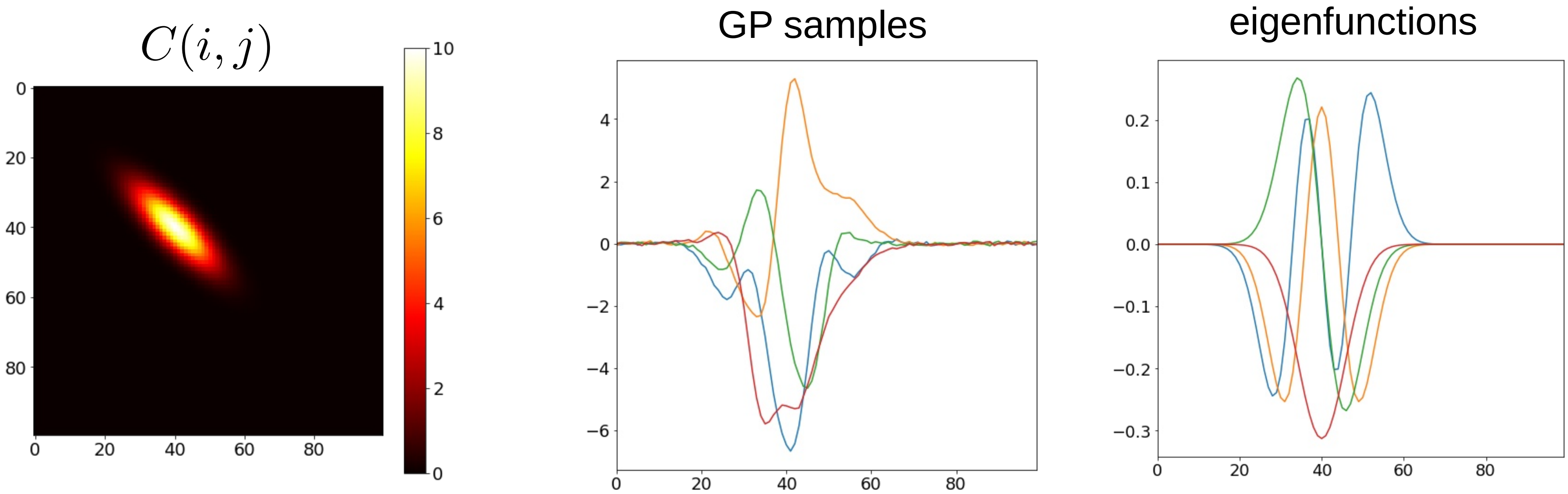


Fourier analysis of this test case



need a kernel which is good for
data in Fourier basis

Example: wavelet basis via non-stationary GPs



Overview

- Context: structure, randomness, & neuroscience
- Kernel theory:
  - Mathematical framework
  - Network sparsity → additive functions
  - Tuning curves → basis change
- **Conclusions**

Conclusions

Conclusions

- Kernel representation of random networks is powerful
  - statistical learning theory: **Which functions are easy to learn?**
  - neuro can learn from ML (dimensionality) and vice-versa (structure)

Conclusions

- Kernel representation of random networks is powerful
  - statistical learning theory: **Which functions are easy to learn?**
  - neuro can learn from ML (dimensionality) and vice-versa (structure)
- Sparse network leads to additive kernels
  - classical way to model “big data”

Conclusions

- Kernel representation of random networks is powerful
  - statistical learning theory: **Which functions are easy to learn?**
  - neuro can learn from ML (dimensionality) and vice-versa (structure)
- Sparse network leads to additive kernels
  - classical way to model “big data”
- Random tuning curves could explain variability seen
  - despite randomness, may represent inputs in Fourier/wavelet bases

Conclusions

- Kernel representation of random networks is powerful
  - statistical learning theory: **Which functions are easy to learn?**
  - neuro can learn from ML (dimensionality) and vice-versa (structure)
- Sparse network leads to additive kernels
  - classical way to model “big data”
- Random tuning curves could explain variability seen
  - despite randomness, may represent inputs in Fourier/wavelet bases
- Many future directions
  - feedback, temporal dynamics, unsupervised settings

Thank you!

- Funding: Washington Research Foundation
- Collaborators:
  - Biraj Pandey, Bing Brunton
  - Marjorie Xie, Ashok Litwin-Kumar, Larry Abbott, Richard Axel, Haim Sompolinsky
- Thanks to Raj Rao, Kamesh Krishnamurthy, Yian Ma, & Francis Bach for discussions

Meaning of dimensionality in statistical learning

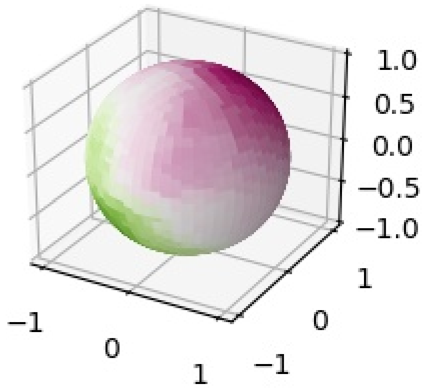
- Eigenvalue decay of kernel matrix *K* that depends on
  - kernel function
  - distribution of data *x*

$$\mathbb{E}(y - \hat{y})^2 = (\text{bias})^2 + \underbrace{\text{variance}}_{\leq \frac{\sigma_y^2}{n}\,\mathbf{dim}}$$

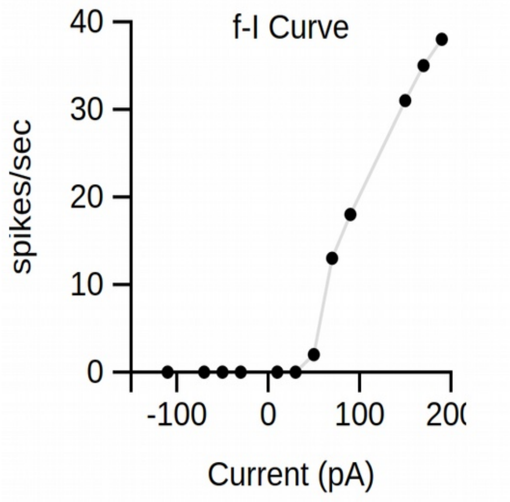
* but not the
“participation ratio”

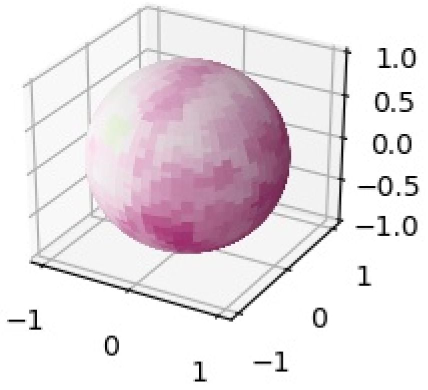
Kernels highlight importance of “preprocessing”

- Antennal lobe glomeruli provide
  - pooling of ORN inputs
  - divisive normalization
- RBFs on the unit sphere

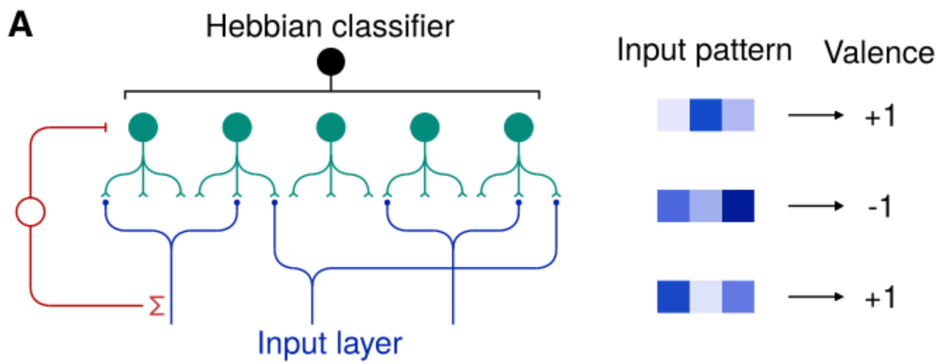


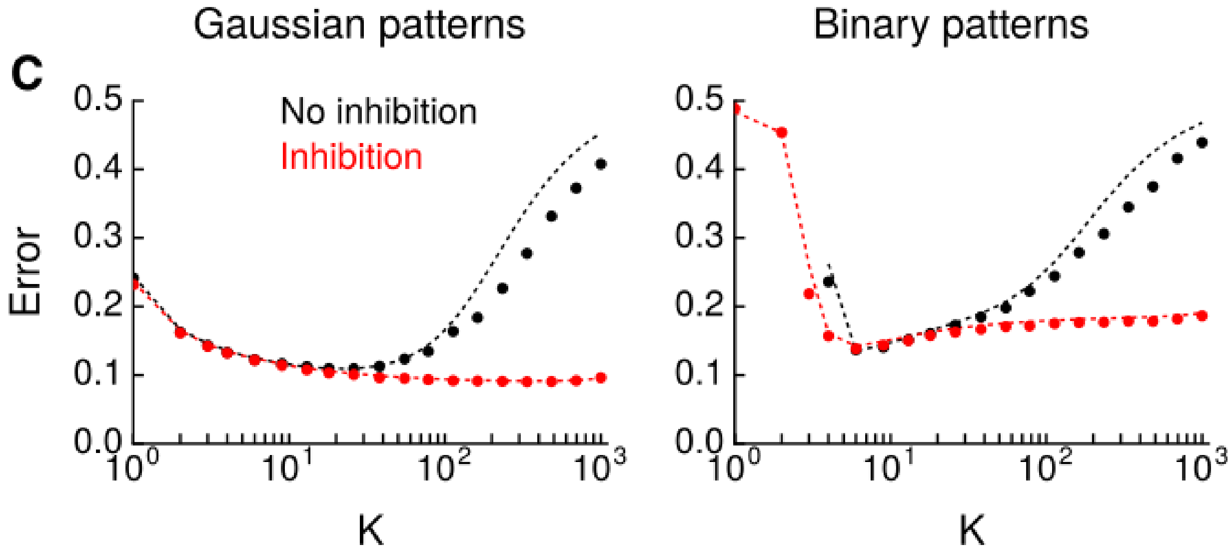
$$\phi_i(\mathbf{x}) = h(\mathbf{w}_i^\intercal \mathbf{x})$$





Sparsity can improve classification





Classifier must
remember & denoise

Uncorrelated random
- Input patterns
- Binary valence
- Binary noise

$$\mathrm{SNR} \cong \frac{\dim(\mathbf{m}) \cdot (1-\Delta)^2}{P}$$

Noise
amplification

Learning input-mixed weights most useful only in dense networks