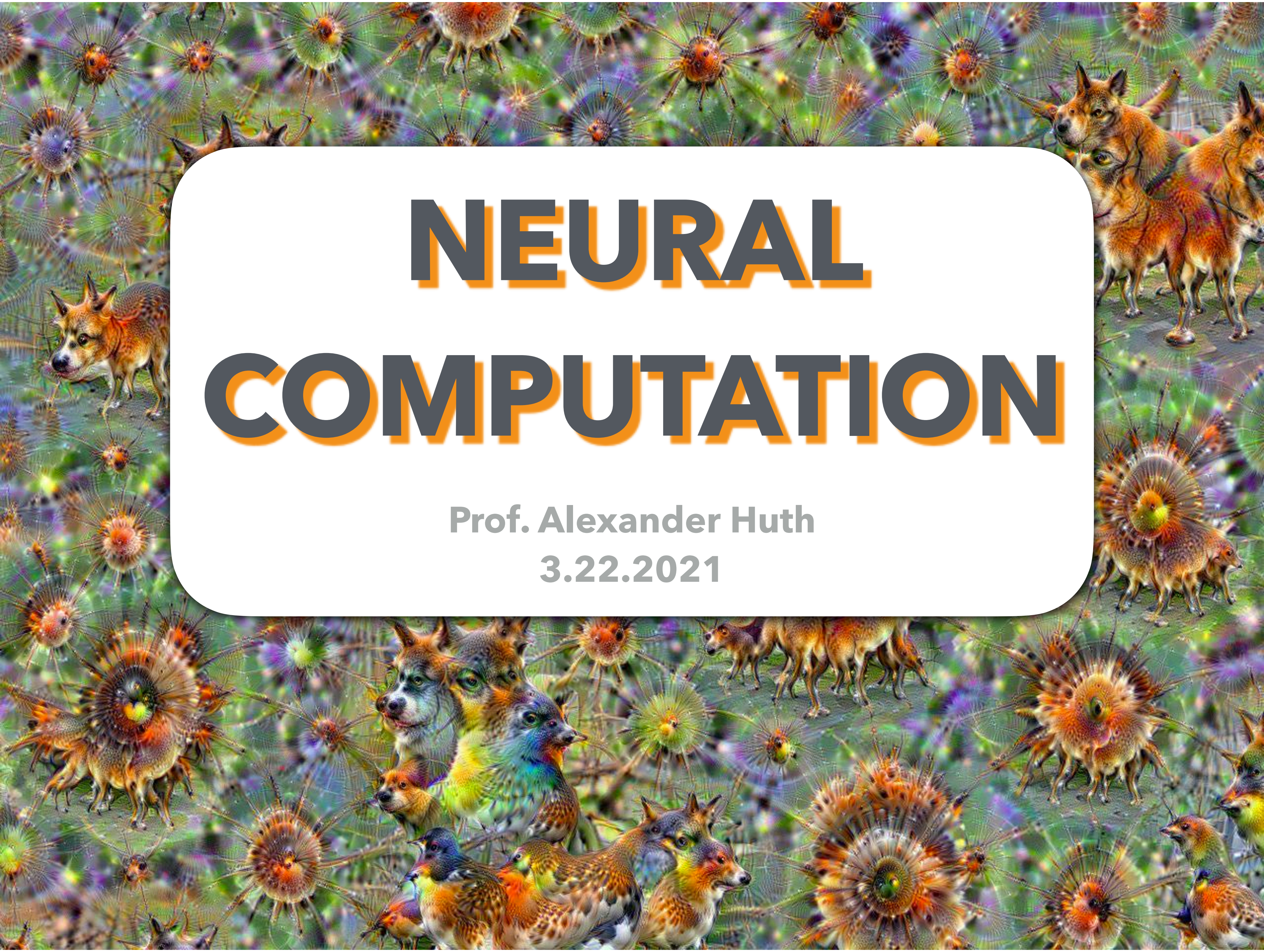


NEURAL COMPUTATION

Prof. Alexander Huth
3.22.2021

PLANNING!

* The next 2.5 weeks:
  * neural networks
  * linking artificial & biological networks
* Homework 1 (the only homework) will be posted Wed. & due in 2 weeks (**April 7**)
* Project proposals will be due **April 12**

VOLTERRA SERIES & KERNEL REGRESSION

RECAP

$$Y = f(X)$$

* System identification
  * Linear $$Y = X\beta$$
  * ***Linearized*** $$Y = \mathbb{L}(X)\beta$$
  * Nonlinear $$Y = \Theta(X)$$

NONLINEAR PROBLEM

x1	0	1	1	0
x2	0	0	1	1
y	0	0	1	0

$$y = f(x_1, x_2)$$

what is f ?

VOLTERRA SERIES

* A finite Volterra series of order P considers every nonlinear combination of up to P variables

$$y = \sum_{n=1}^{P} \sum_{\tau_1=1}^{p} \cdots \sum_{\tau_n=1}^{p} h_n(\tau_1, \ldots, \tau_n) \prod_{j=1}^{n} x_j$$

VOLTERRA SERIES

* A finite Volterra series of order P considers every nonlinear combination of up to P variables

$$y = h_{1,0}x_1 + h_{0,1}x_2 + h_{1,1}x_1x_2 + h_{2,0}x_1^2 + h_{0,2}x_2^2 + h_{2,2}x_1^2x_2^2 + \ldots$$

VOLTERRA SOLUTION!

x1	0	1	1	0
x2	0	0	1	1
y	0	0	1	0

$$y = f(x_1, x_2)$$

$$y = h_{1,0}x_1 + h_{0,1}x_2 + h_{1,1}x_1x_2 + h_{2,0}x_1^2 + h_{0,2}x_2^2 + h_{2,2}x_1^2x_2^2 + \ldots$$

$$h_{1,1} = 1, h_{i,j} = 0 \text{ for all other } i, j$$

VOLTERRA SERIES

* (btw, Volterra series is just a different linearized model…)

* (but it’s one that can capture any nonlinear function!)

VOLTERRA SERIES

* Volterra series have ***nightmarish*** numbers of parameters

* Suppose X’s are 16x16 image patches (i.e. p=256)

* How many coefficients (h’s) are there in a 5th-order Volterra model? (~1 billion!)

KERNEL REGRESSION

FORGET FEATURES, USE SAMPLES!

** Please do not actually forget features*

KERNEL REGRESSION

* Let’s say the y for a new sample is some a combination of the y’s from old samples

* *Example:* image patches

KERNEL REGRESSION

* **Kernel function:** $k(a, b) = \phi(a)^\top \phi(b)$

  tells you how similar a and b are in some "Reproducing kernel Hilbert space", H

KERNEL REGRESSION

*** Representer theorem:**

$$\hat{f} = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \left[||Y - f(X)||_2^2 + \lambda ||f||_{\mathcal{H}}^2 \right]$$

then: $$\hat{f}(z) = \sum_{i=1}^{n} \alpha_i k(z, X_i)$$

i.e. the function value for a new datapoint, z, is a linear combination (with weights alpha) of the kernel similarities between z and existing datapoints in X

KERNEL REGRESSION

* How do we find the alphas?

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmin}} \left[||Y - K\alpha||_2^2 + \lambda \alpha^\top K \alpha \right]$$

where: $K_{ij} = k(X_i, X_j)$

KERNEL REGRESSION

* How do we find the alphas?

$$\hat{\alpha} = (K + \lambda I)^{-1} Y$$

(this is called ***KERNEL RIDGE REGRESSION***)

KERNEL REGRESSION

* Ok fine. But what the heck is *k*?!?

* **Possibility 1:** linear kernel!

$$k(a, b) = a^\top b$$

KERNEL REGRESSION

* **Possibility 1:** linear kernel!

remember: $\hat{f}(z) = \sum_{i=1}^{n} \alpha_i k(z, X_i)$

$$k(a,b) = a^\top b \qquad \Rightarrow K = XX^\top$$

$$\Rightarrow \hat{\alpha} = (XX^\top + \lambda I)^{-1} Y$$

$$\Rightarrow \hat{f}(z) = zX^\top \hat{\alpha} = zX^\top (XX^\top + \lambda I)^{-1} Y$$

KERNEL REGRESSION

* **Possibility 1:** linear kernel!

remember: $\hat{f}(z) = \sum_{i=1}^{n} \alpha_i k(z, X_i)$

$$k(a,b) = a^\top b \quad \Rightarrow K = XX^\top$$

$$\Rightarrow \hat{\alpha} = (XX^\top + \lambda I)^{-1} Y$$

$$\Rightarrow \hat{f}(z) = zX^\top \hat{\alpha} = z X^\top (XX^\top + \lambda I)^{-1} Y$$

what if we just called this part "beta"?

KERNEL REGRESSION

* **Possibility 2:** inhomogeneous polynomial

$$\phi_p(x) = (x_1, x_2, x_1 x_2, \ldots, x_1^p x_2^p)$$

remember: $k(a, b) = \phi(a)^\top \phi(b)$

KERNEL REGRESSION

* **Possibility 2:** inhomogeneous polynomial

$$\phi_p(x) = (x_1, x_2, x_1 x_2, \ldots, x_1^p x_2^p)$$

remember: $k(a,b) = \phi(a)^\top \phi(b)$

Volterra series model!

But with only n parameters!

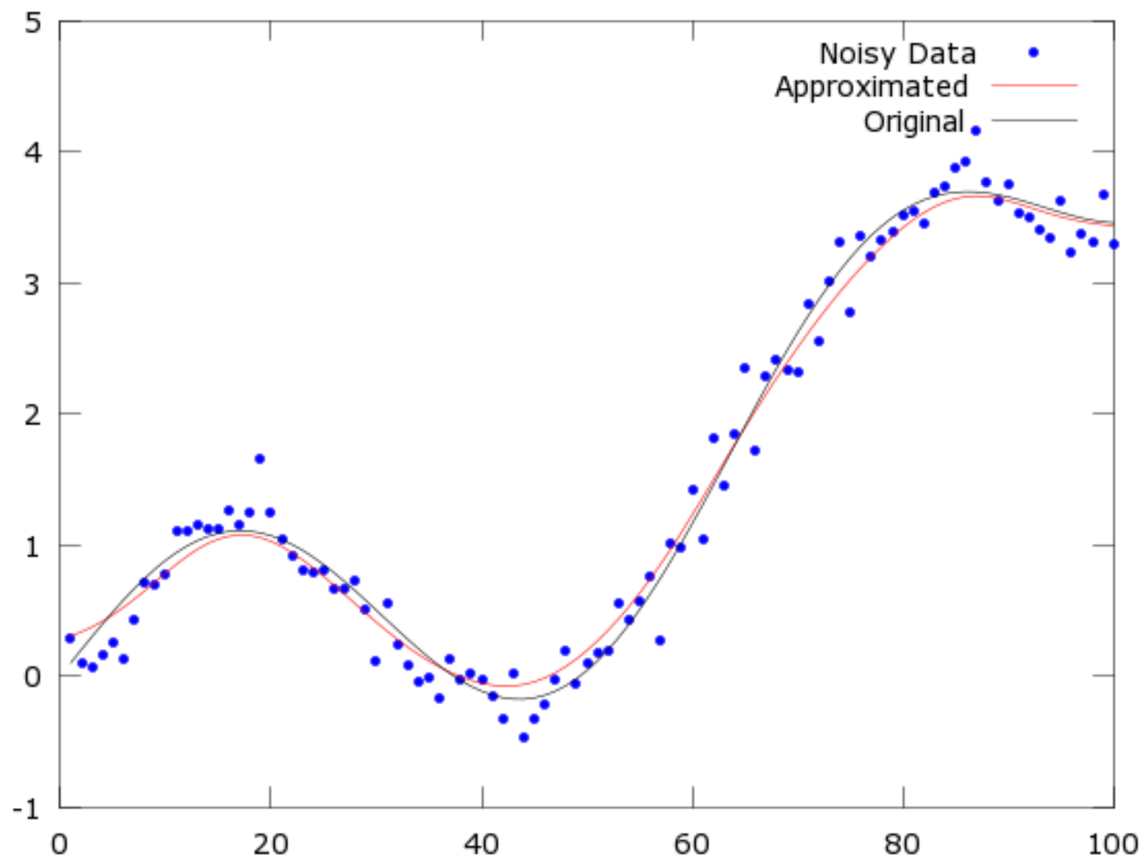
KERNEL REGRESSION

* **Possibility 3:** Radial basis function (RBF)

$$k(a, b) = e^{-||a-b||_2^2/(2\sigma^2)}$$

KERNEL REGRESSION

* **Possibility 3:** Radial basis function (RBF)



KERNEL EFFICIENCY

* Beyond nonlinear applications, kernel regression can also be more efficient in some situations

* Q: What’s the time complexity of kernel regression vs. ridge regression?

KERNEL EFFICIENCY

* Let’s suppose the complexity of dotting an (n x m) matrix with an (m x p) matrix is (nmp)

* And let’s suppose the complexity of inverting an (n x n) matrix is (n^3)

* X is (n x p), Y is (n x m)

KERNEL EFFICIENCY

* What’s the complexity of solving for weights (beta) in ridge regression?

* What’s the complexity of solving for weights in kernel ridge regression?

* Under what conditions is kernel ridge better than ridge, and vice versa?

NEXT TIME

* Neural networks! (well, at least perceptrons)