ARTIFICIAL NEURAL NETWORKS

Prof. Alexander Huth
3.12.2020

RECAP:
NONLINEAR METHODS

A 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 ?

RECAP:
NONLINEAR METHODS

- **Volterra series**

  - Generic, *expensive*

- **Kernel regression** *(samples, not features!)*

  - Generic (w/ RBF or poly. kernel),
    ~linear regression (w/ linear kernel),
    ~Volterra series (w/ Volterra kernel)

  - *Not expensive!*

RECAP:
KERNEL REGRESSION

$$\hat{f}(z) = \sum_{i=1}^{n \leftarrow \textit{sum across datapoints}} \underset{\substack{\uparrow \\ \textit{weight}}}{\alpha_i} \underset{\substack{\uparrow \\ \textit{kernel} \\ \textit{function}}}{k}(z, X_i)$$

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

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

RECAP:
KERNEL REGRESSION

- **Possible Kernel:** inhomogeneous polynomial

$$k(a,b) = (a^\top b + 1)^p$$

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

What is phi?

RECAP: KERNEL REGRESSION

Volterra series model!

But with only n parameters!

* **Possible Kernel:** inhomogeneous polynomial

$$k(a,b) = (a^\top b + 1)^p$$

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

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

RECAP:
KERNEL EFFICIENCY

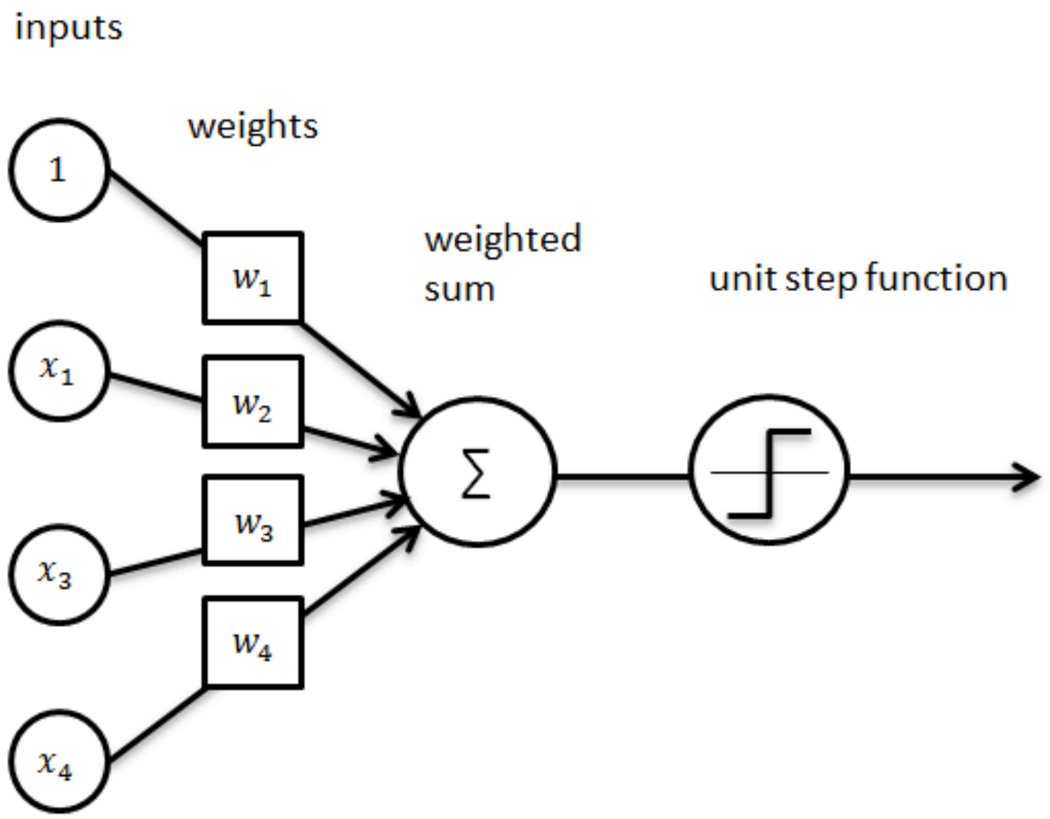
* What’s the complexity of solving for weights (beta) in ridge regression?

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

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

ARTIFICIAL NEURAL NETWORK

* Simplest version: a *perceptron*



$$y_i \in \{0, 1\}$$

$$X_i \in \mathcal{R}^p$$

$$\hat{y}_i = H(X_i w)$$

h/t http://ataspinar.com/2016/12/22/the-perceptron/

PERCEPTRON LEARNING

* Due to Frank Rosenblatt (1958)

* Usually defined as online learning rule (assuming each example will be fed in one-by-one)

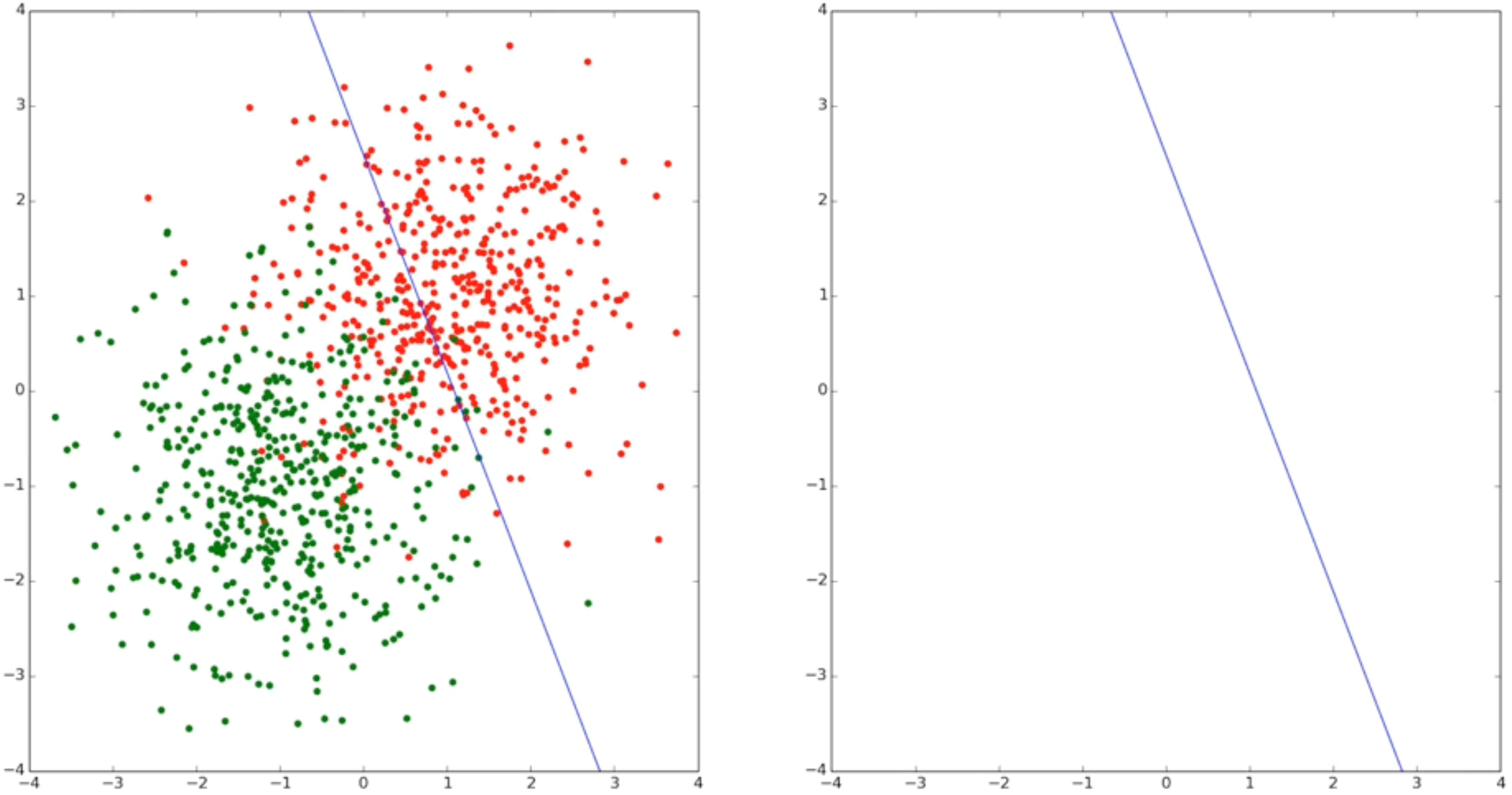
PERCEPTRON LEARNING

At iteration *t* in the learning process:

$$\hat{y}_i^{(t)} = H(X_i w^{(t)})$$

$$w_j^{(t+1)} = w_j^{(t)} + (y_i - \hat{y}_i^{(t)})X_{ij}$$

PERCEPTRON LEARNING



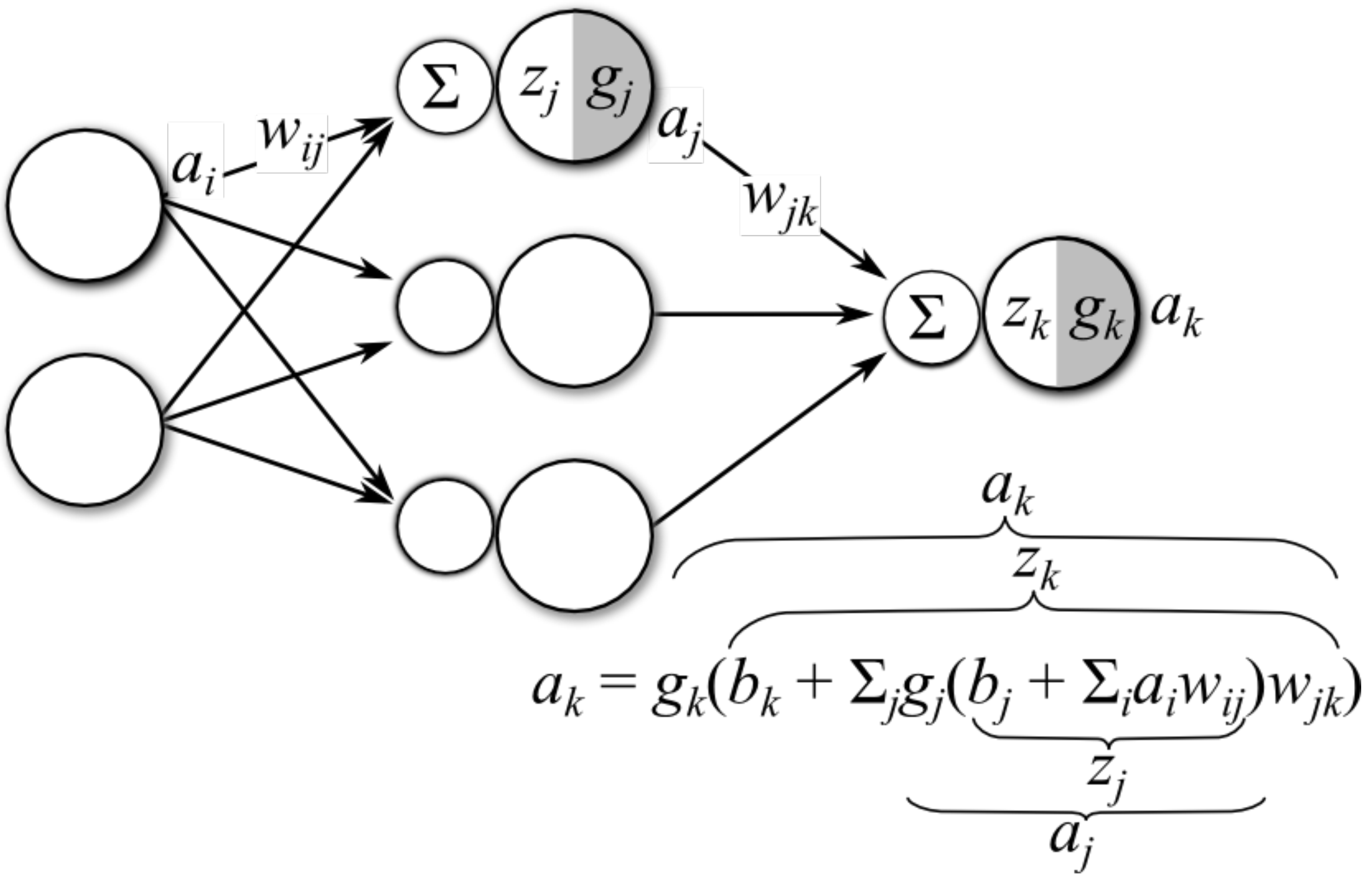
h/t https://github.com/ayusek/Perceptron-Animation

PERCEPTRONS

* “You dumdums, single perceptrons can’t even solve something as simple as XOR, how could they possibly recognize images or walk or talk or be conscious?!”
– Minsky & Papert, basically, 1969

* –> “AI winter” of 1974-1980

MULTILAYER ARTIFICIAL NEURAL NETWORKS



$$a_k = \overbrace{g_k(\overbrace{b_k + \Sigma_j \underbrace{g_j(\underbrace{b_j + \Sigma_i a_i w_{ij}}_{z_j})}_{a_j} w_{jk}}^{z_k})}^{a_k}$$

w = weights

a = activations

z = total input

g = activ. fxn

b = bias

h/t Dustin Stansbury

MULTILAYER ARTIFICIAL NEURAL NETWORKS

* Let y-hat = a_k (the output),
  $X_i=a_i$ (the input)

* Suppose g’s are fixed

* How do we learn the w’s and b’s?

MULTILAYER ARTIFICIAL NEURAL NETWORKS

* Backpropagation!

MULTILAYER ARTIFICIAL NEURAL NETWORKS

- Suppose number of hidden units = number of training samples

- Suppose w_{ij} = X_j

- *Suddenly*.. a kernel regression neural network, with k(a,b) = tanh(a.b)

h/t Michael Oliver

NEXT TIME

* Next class will be on March 31

* It will almost certainly be online (via Zoom, in all likelihood)

* Since I don’t think making announcements on GitHub is very reliable, I’m going to also post announcements on Canvas