10417/10617 Intermediate Deep Learning: Fall2019

Russ Salakhutdinov

Machine Learning Department
rsalakhu@cs.cmu.edu

https://deeplearning-cmu-10417.github.io/

Midterm Review

Midterm Review

- Polynomial curve fitting – generalization, overfitting
- Loss functions for regression

$$\mathbb{E}[L] = \int \int \big(t - y(\mathbf{x})\big)^2 p(\mathbf{x}, t) \mathrm{d}\mathbf{x} \mathrm{d}t.$$

- Generalization / Overfitting
- Statistical Decision Theory

Midterm Review

- Bernoulli, Multinomial random variables (mean, variances)
- Multivariate Gaussian distribution (form, mean, covariance)
- Maximum likelihood estimation for these distributions.

- Exponential family / Maximum likelihood estimation / sufficient statistics for exponential family.

$$p(\mathbf{x}|\boldsymbol{\eta}) = h(\mathbf{x})g(\boldsymbol{\eta})\exp\left\{\boldsymbol{\eta}^{\mathrm{T}}\mathbf{u}(\mathbf{x})\right\}$$

- Linear basis function models / maximum likelihood and least squares:

$$\ln p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \sum_{i=1}^{N} \ln \mathcal{N}(t_n|\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n), \beta)$$

$$= -\frac{\beta}{2}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)\right)^2 + \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi).$$

$$\mathbf{w}_{\mathrm{ML}} = \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}$$

Midterm Review

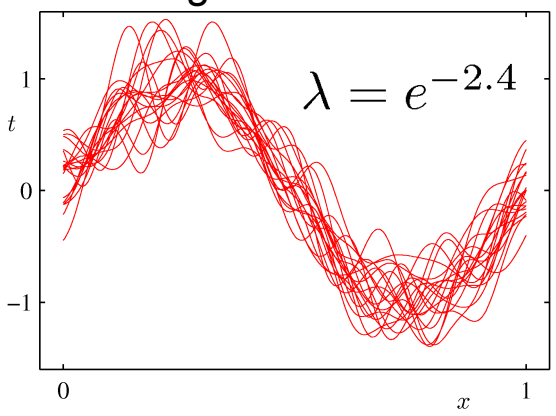
- Regularized least squares:

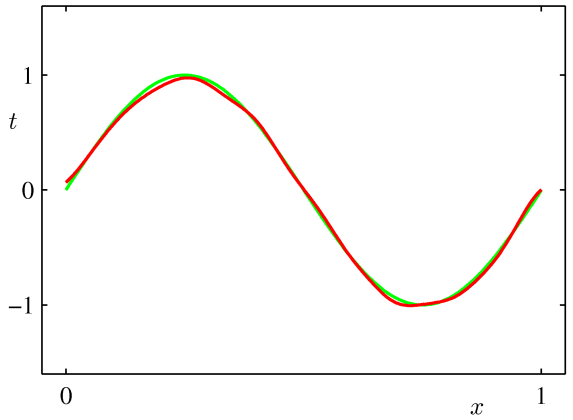
Ridge regression

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \qquad \mathbf{w} = \left(\lambda\mathbf{I} + \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}.$$

- Bias-variance decomposition.

High variance



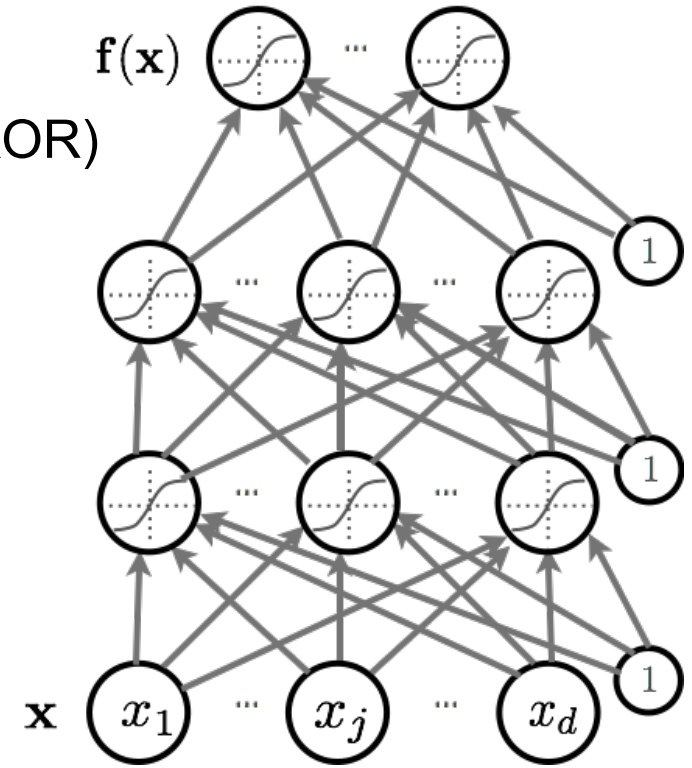


Low bias

- Gradient Descend, SGD, Parameter Update Rules

Neural Networks

- How neural networks predict f(x) given an input x:
  - Forward propagation
  - Types of units
  - Capacity of neural networks (AND, OR, XOR)
- How to train neural nets:
  - Loss function
  - Backpropagation with gradient descent
- More recent techniques:
  - Dropout
  - Batch normalization
  - Unsupervised Pre-training



Artificial Neuron: Logistic Regression

- Neuron pre-activation (or input activation):

$$a(\mathbf{x}) = b + \sum_i w_i x_i = b + \mathbf{w}^\top \mathbf{x}$$

- Neuron output activation:

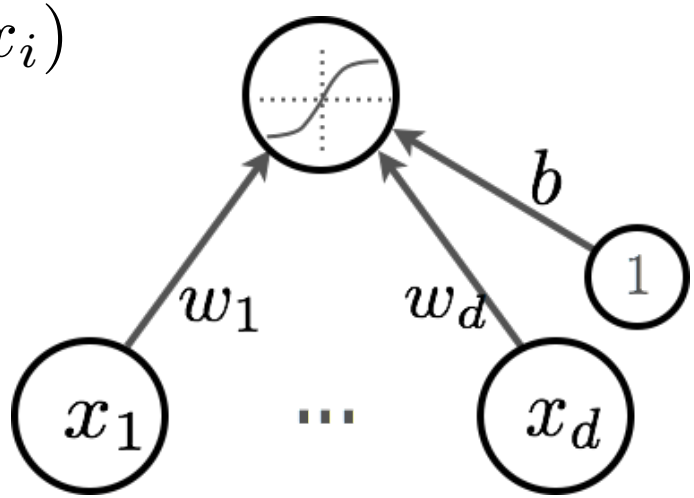
$$h(\mathbf{x}) = g(a(\mathbf{x})) = g(b + \sum_i w_i x_i)$$

where

$\mathbf{w}$ are the weights (parameters)

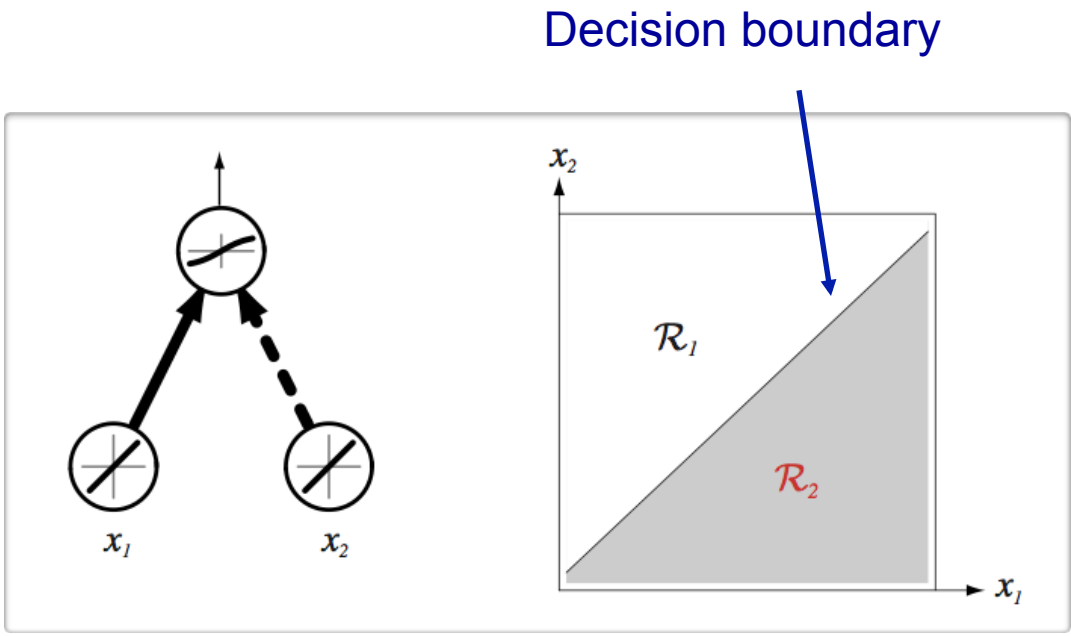
b is the bias term

$g(\cdot)$ is called the activation function



Decision Boundary of a Neuron

- Binary classification:
  - With sigmoid, one can interpret neuron as estimating $p(y = 1|\mathbf{x})$
  - Interpret as a logistic classifier
  - If activation is greater than 0.5, predict 1
  - Otherwise predict 0

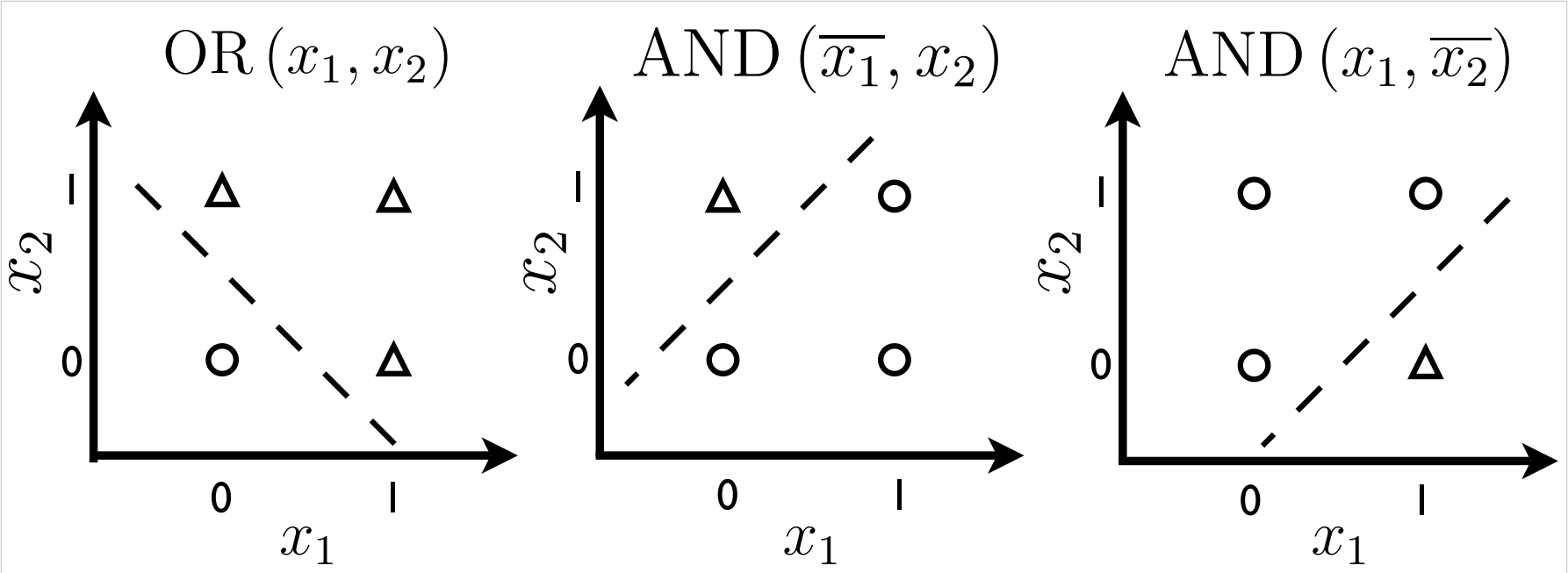


(from Pascal Vincent's slides)

Same idea can be applied to a tanh activation

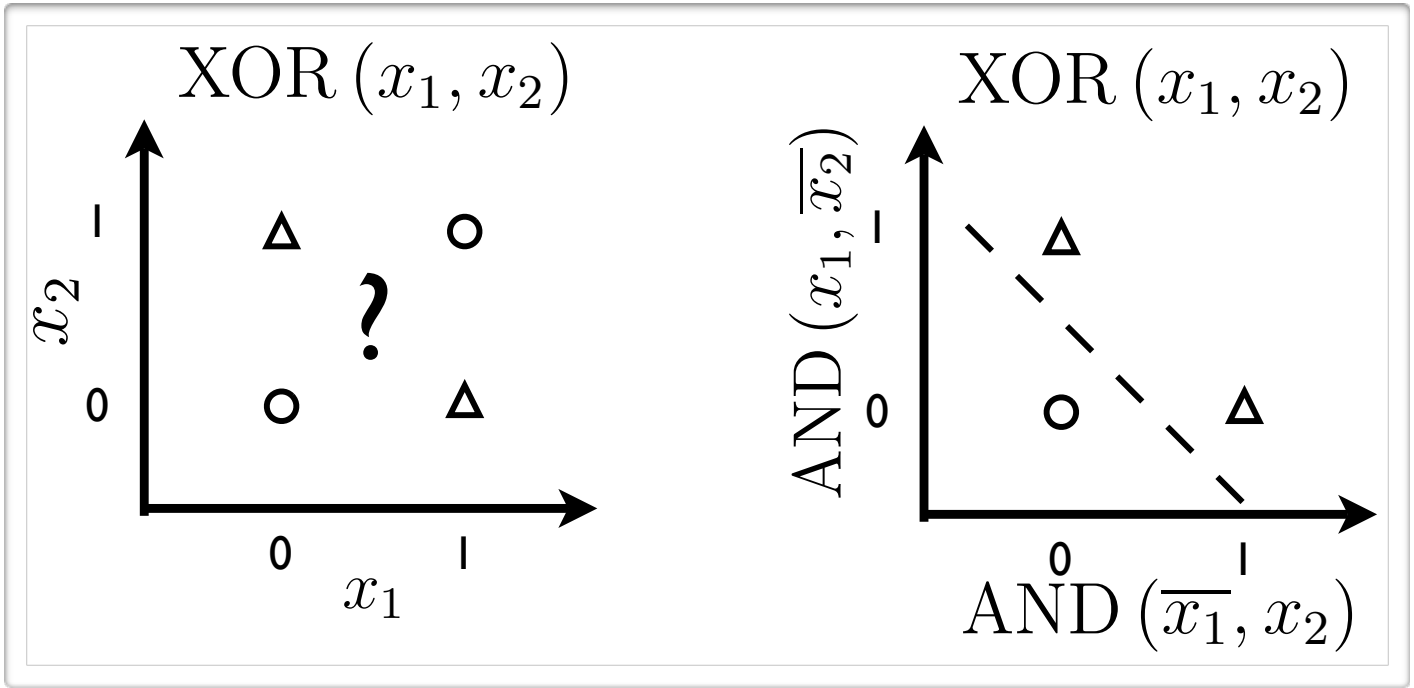
Capacity of a Single Neuron

- Can solve linearly separable problems.



Capacity of a Single Neuron

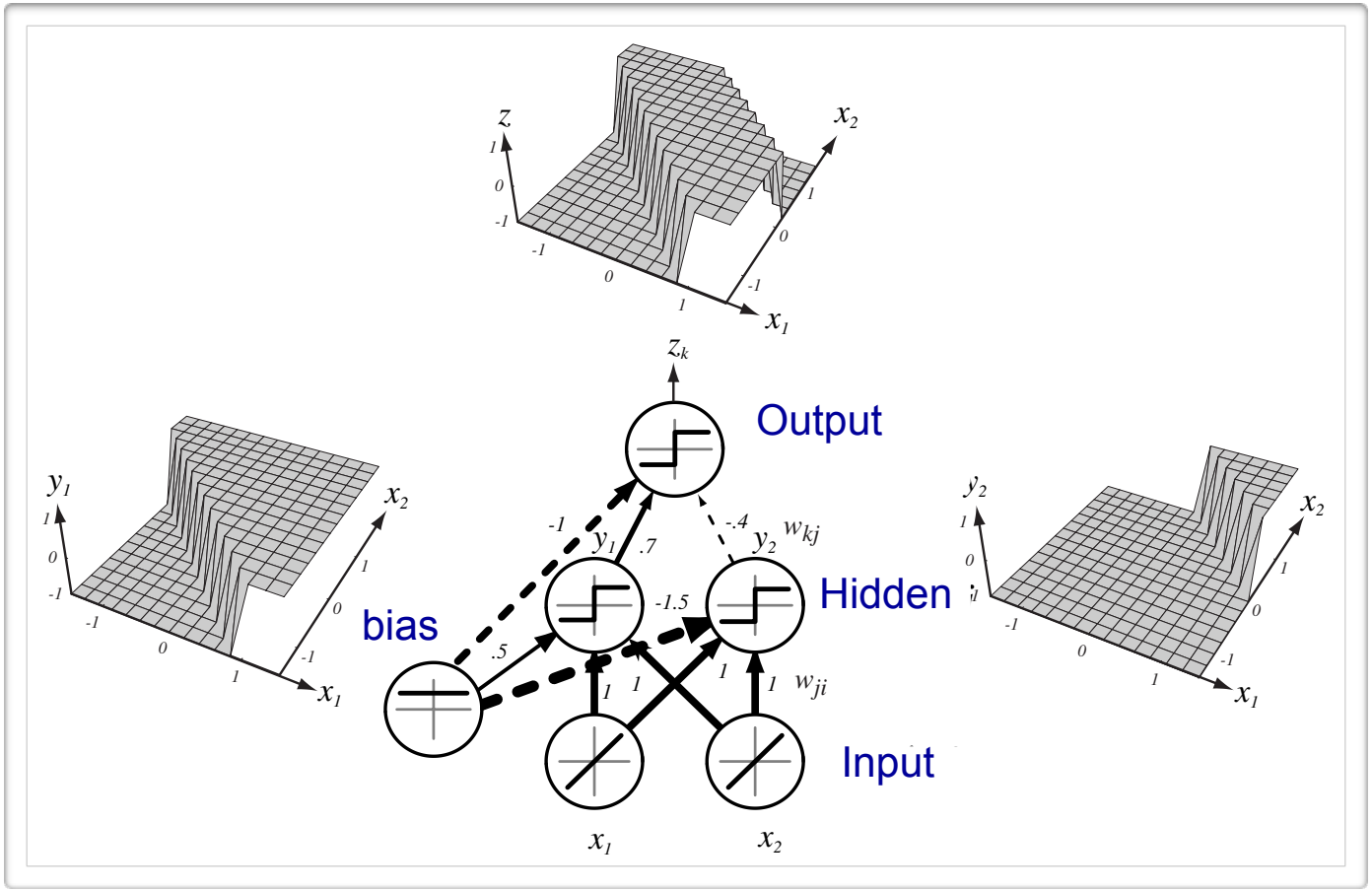
- Can not solve non-linearly separable problems.



- Need to transform the input into a better representation.
- Remember basis functions!

Capacity of Neural Nets

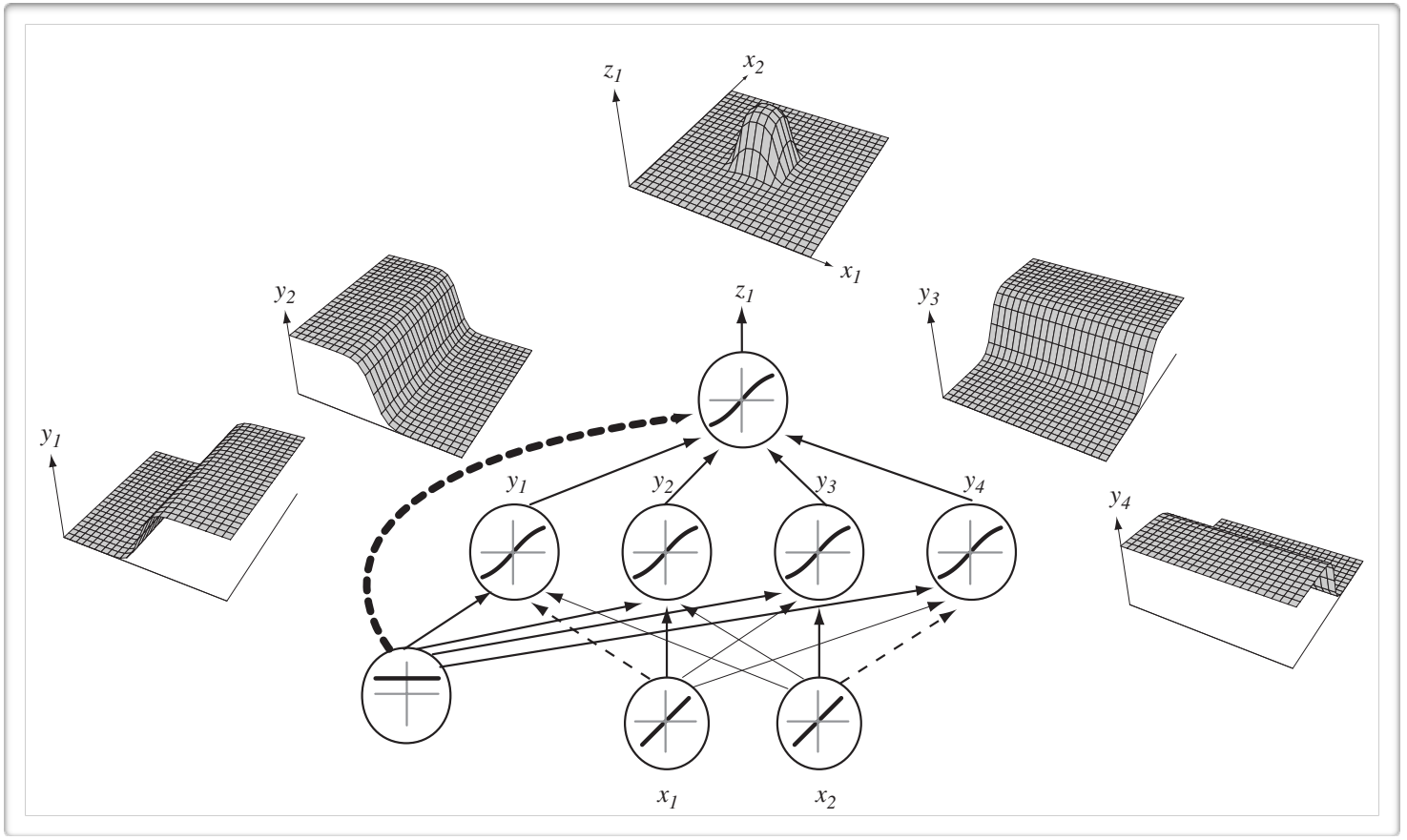
- Consider a single layer neural network



(from Pascal Vincent's slides)

Capacity of Neural Nets

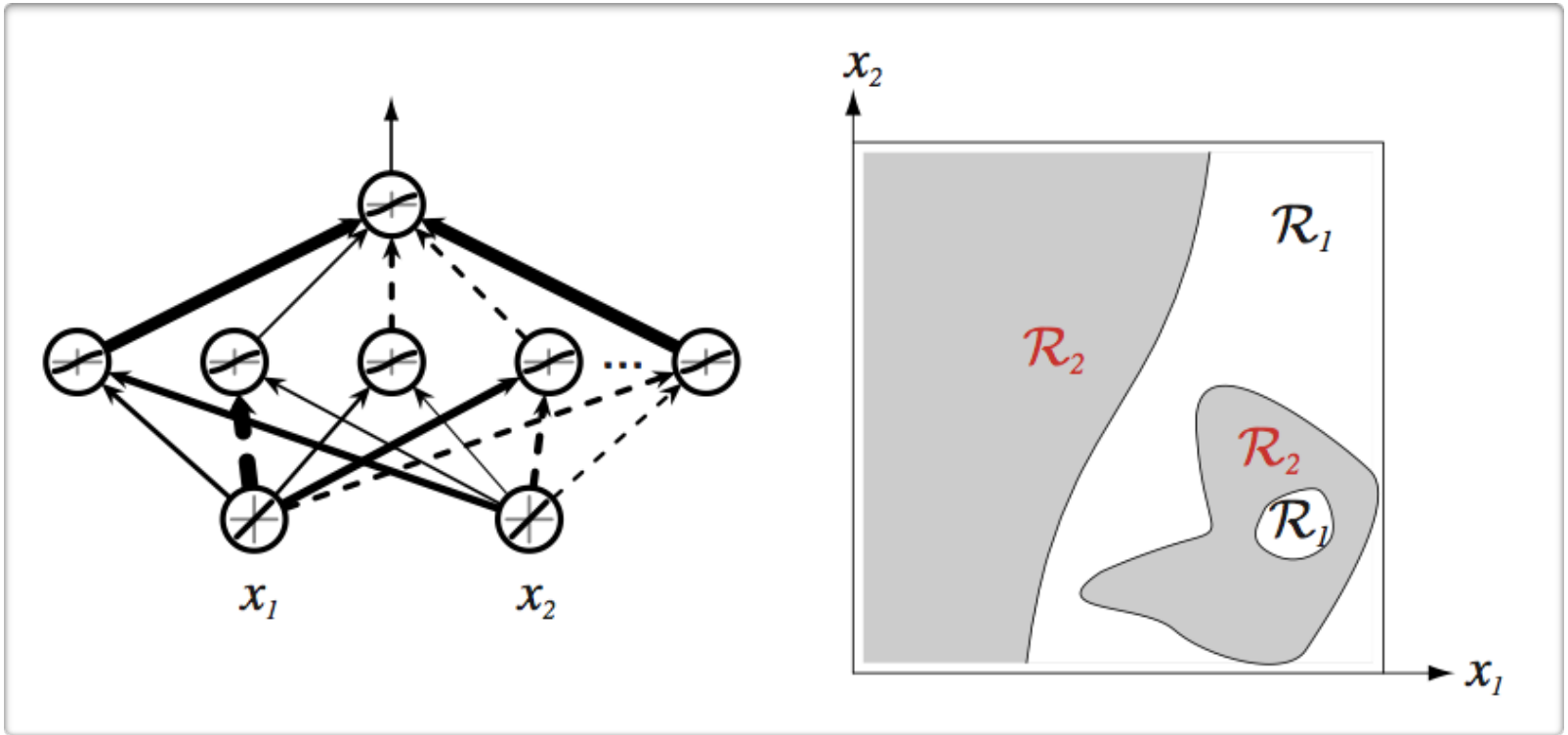
- Consider a single layer neural network



(from Pascal Vincent's slides)

Capacity of Neural Nets

- Consider a single layer neural network



(from Pascal Vincent's slides)

Neural Networks

- SGD Training, cross entropy loss, ReLU activations
- Classification with neural networks
- Regularization, Dropout, Batchnorm
- Forward Propagation and Backprop (computing derivatives)

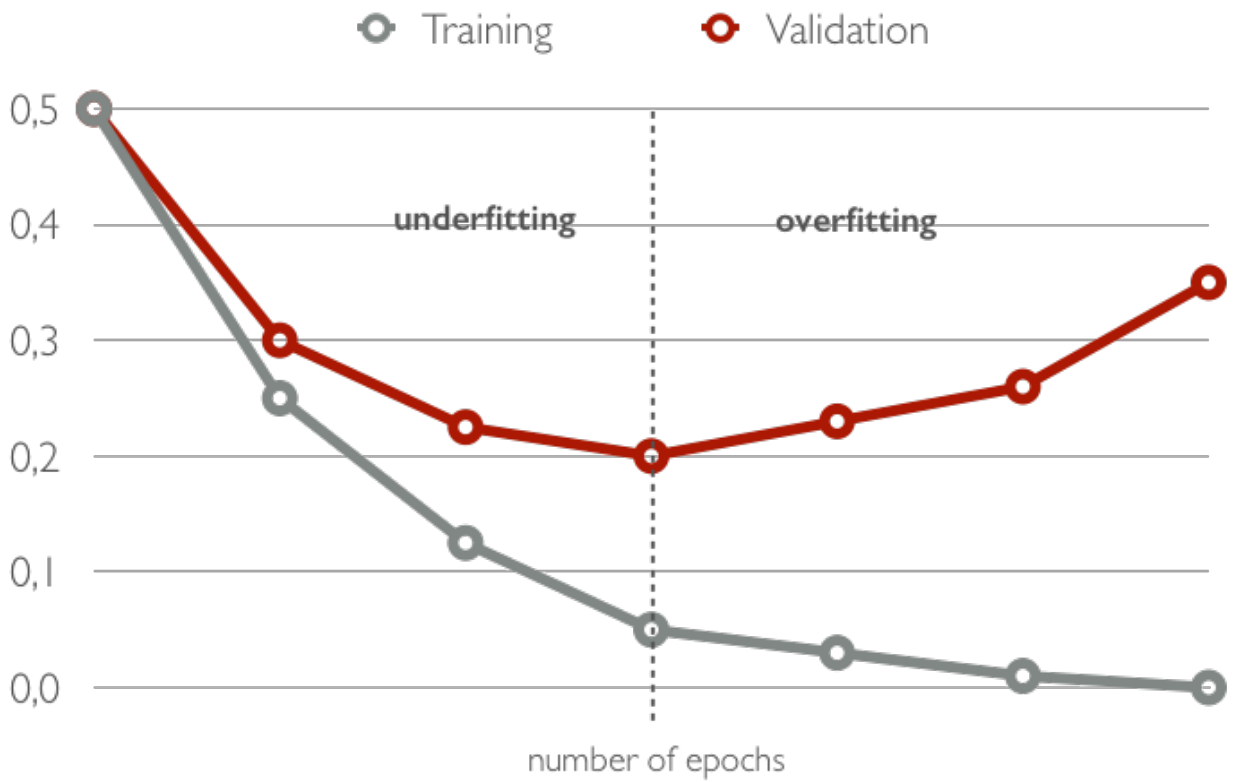
Model Selection

- Training Protocol:
  - Train your model on the Training Set $\mathcal{D}^{\text{train}}$
  - For model selection, use Validation Set $\mathcal{D}^{\text{valid}}$
    - Hyper-parameter search: hidden layer size, learning rate, number of iterations/epochs, etc.
  - Estimate generalization performance using the Test Set $\mathcal{D}^{\text{test}}$

- Remember: Generalization is the behavior of the model on **unseen examples**.

Early Stopping

• To select the number of epochs, stop training when validation set error increases (with some look ahead).



Conv Nets

- Convolutional networks leverage these ideas
  - Local connectivity
  - Parameter sharing
  - Convolution
  - Pooling / subsampling hidden units
  - Understanding Receptive Fields

- Local contrast normalization, rectification