Elemental Learning Theory
(Wrap-up!)

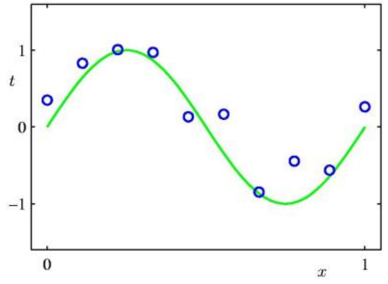
Alexander G. Ororbia II

Introduction to Machine Learning

CSCI-635

9/22/2023

Polynomial Curve Fitting with a Scalar



- With a <u>single</u> input variable x
- $y(x,\mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$

  M = order of polynomial,

  x^j denotes x raised to power j,

  Coefficients $w_0,\ldots,w_M$ are collectively denoted by vector $\mathbf{w}$

Training data set

N=10, Input x, target t

- **Task**: Learn $\mathbf{w}$ from training data $D=\{(x_i,t_i)\},\ i=1,..,N$
  - Can be done by minimizing an error function that minimizes misfit between $y(x,\mathbf{w})$ for any given $\mathbf{w}$ and training data
  - One simple choice of error function is sum of squares of error (*SSE*) between predictions $y(x_n,\mathrm{w})$ for each data point x_n and corresponding target values t_n so that we minimize: $$E(\mathrm{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathrm{w}) - t_n\}^2$$
  - It is zero when function $y(x,\mathrm{w})$ passes exactly through each training data point

On Basis Functions

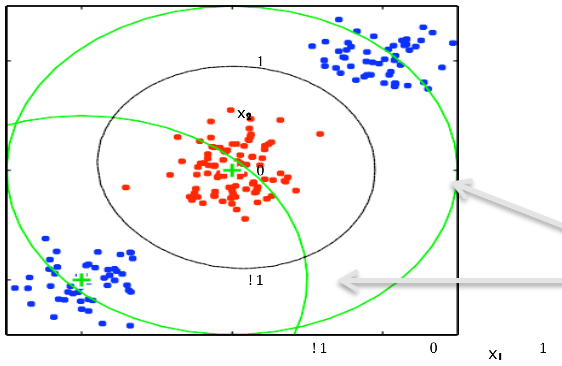
- In many applications, we apply some form of fixed-preprocessing, or feature extraction, to the original data variables
- If the original variables comprise the vector $\mathbf{x}$, then the features can be expressed in terms of basis functions $\{\phi_j(\mathbf{x})\}$
  - By using nonlinear basis functions we allow the function $y(\mathbf{x},\mathbf{w})$ to be a nonlinear function of the input vector $\mathbf{x}$
    - They are linear functions of parameters (gives them simple analytical properties), yet are nonlinear wrt input variables

Fixed Basis Functions

Although we use linear (classification) models

Linear-separability in ***feature*** space *does not* imply linear-separability in ***input*** space

Original Input Space (x_1, x_2)

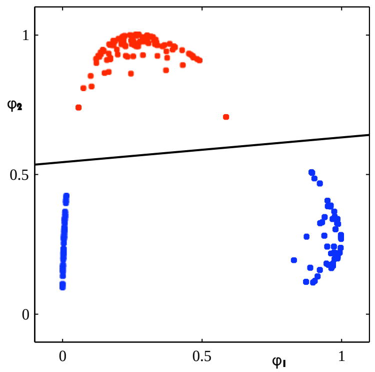


not linearly separable

Nonlinear transformation of inputs using vector of basis functions $\boldsymbol{\phi}(\boldsymbol{x})$

Basis functions are Gaussian with centers shown by crosses and green contours

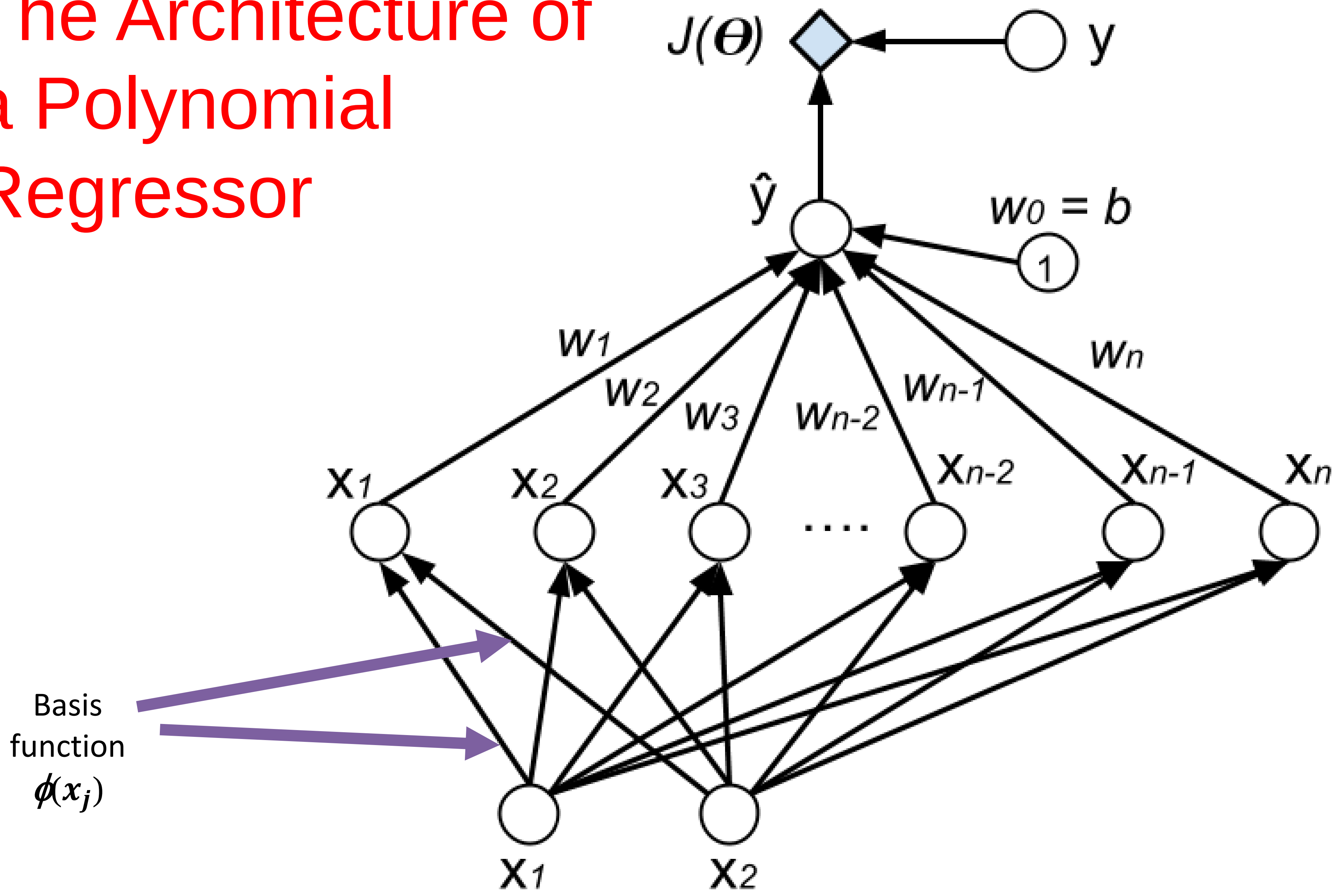
Feature Space (ϕ_1, ϕ_2)



linearly separable

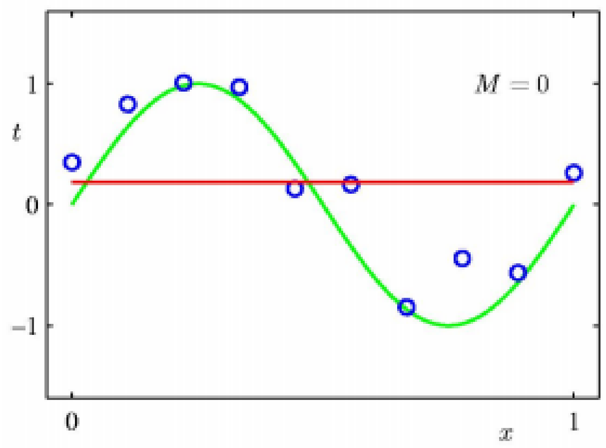
Basis functions with increased dimensionality often used

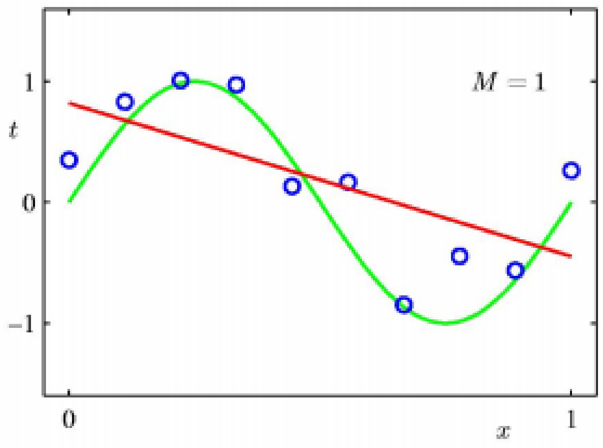
The Architecture of a Polynomial Regressor



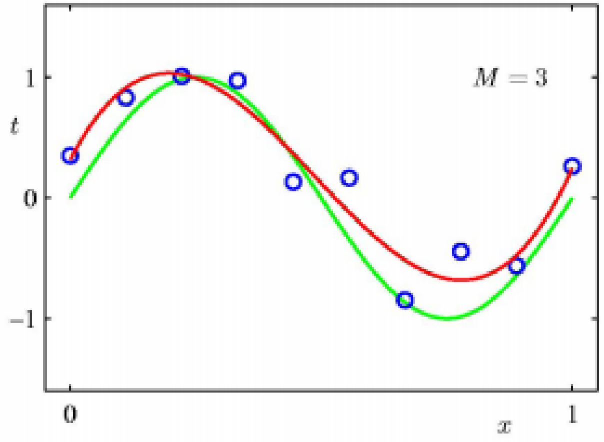
Choosing the Order of *M*

- Model Comparison or Model Selection
- Red lines are best fits with
  - $M = 0,1,3,9$ and $N=10$

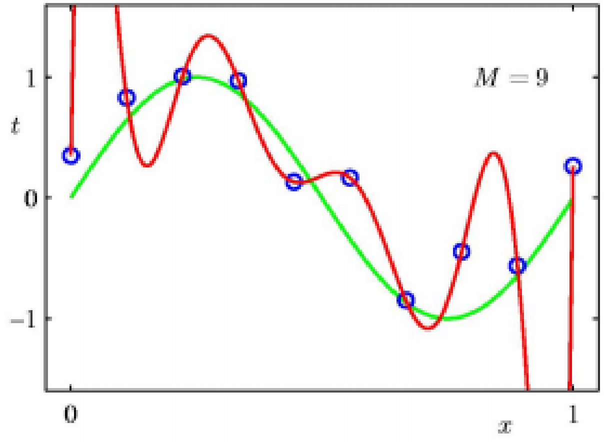




← Poor representations of $\sin(2\pi x)$



← Best Fit to $\sin(2\pi x)$

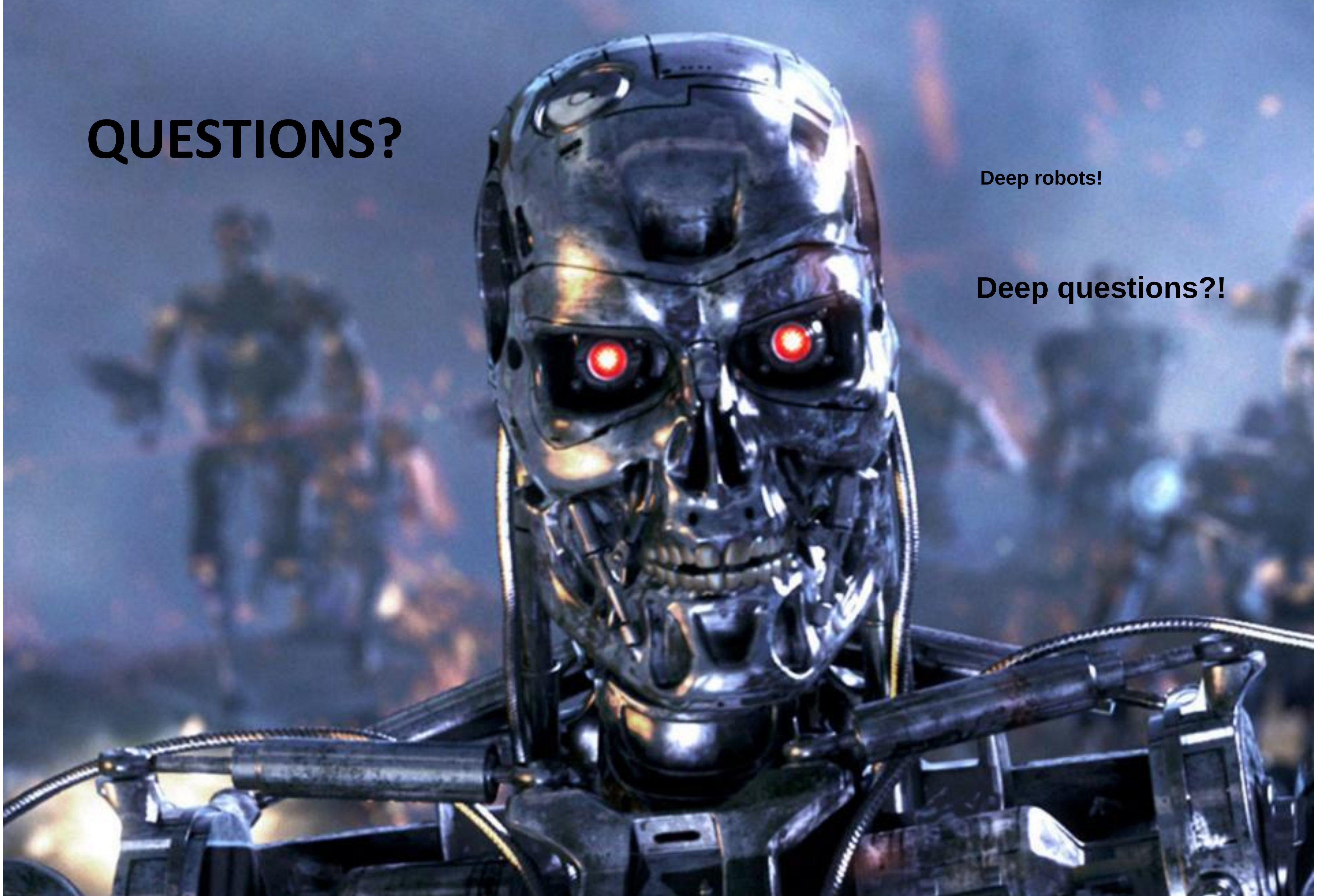


Over Fit
Poor representation of $\sin(2\pi x)$

Computational Bottleneck

- A recurring problem in machine learning:
  - Large training sets are necessary for good generalization
  - *BUT* large training sets are also computationally expensive to use
- Stochastic gradient descent (SGD) is an extension of gradient descent (GD) that offers a solution
  - Moreover, it is a vehicle for generalization beyond training set
  - Expectation may be approximated using small set of samples (we will also later refer to these sets as “mini-batches” → mini-batch GD)

QUESTIONS?



Deep robots!

Deep questions?!