EENG582: Artificial Neural Networks

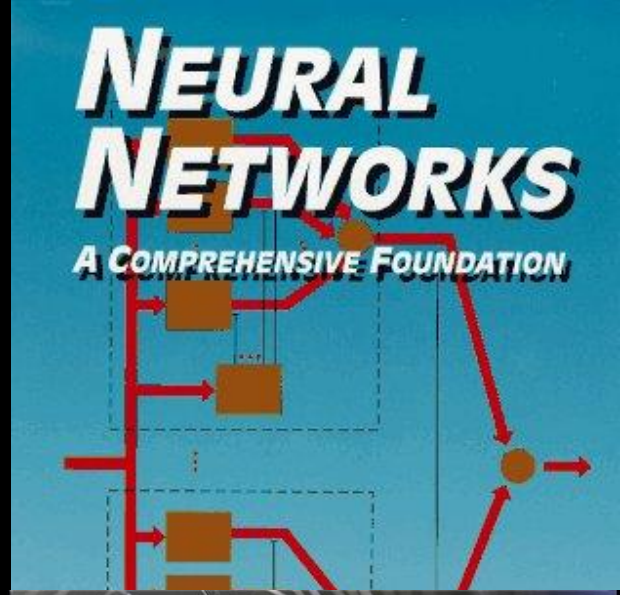
Neural Networks, A Comprehensive Foundation
by S. Haykin
Neural Networks and Deep Learning, by Michael Nielsen
http://neuralnetworksanddeeplearning.com
Introduction to Convolutional Neural Networks, by Vicky Kalogeiton

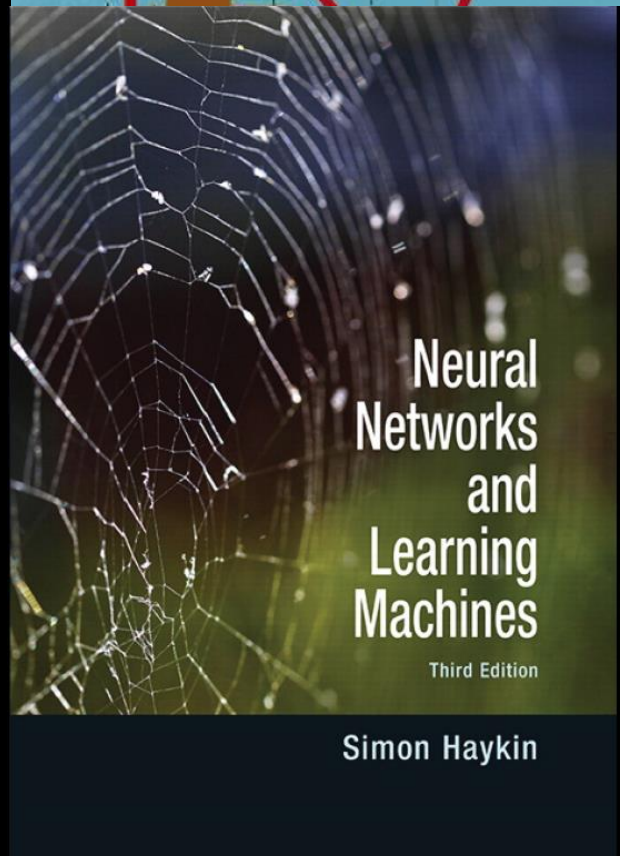
Sections 4.19 – 4.20

4 Multilayer Perceptrons

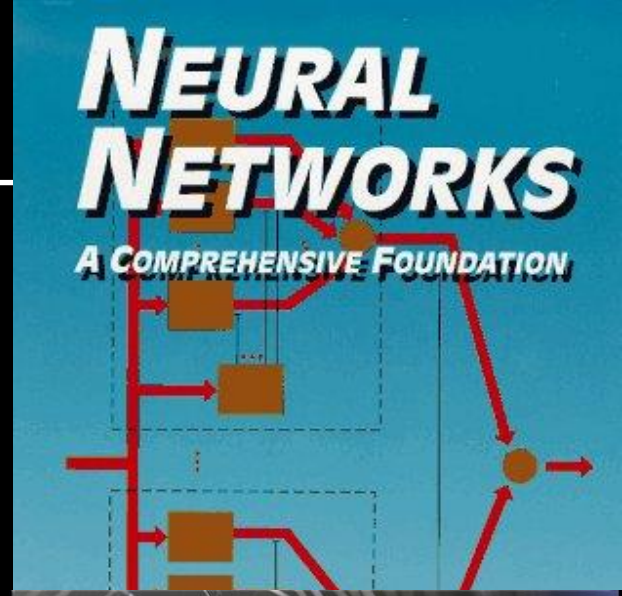
Prof. Dr. Hasan AMCA
Electrical and Electronic Engineering Department
(ee.emu.edu.tr)

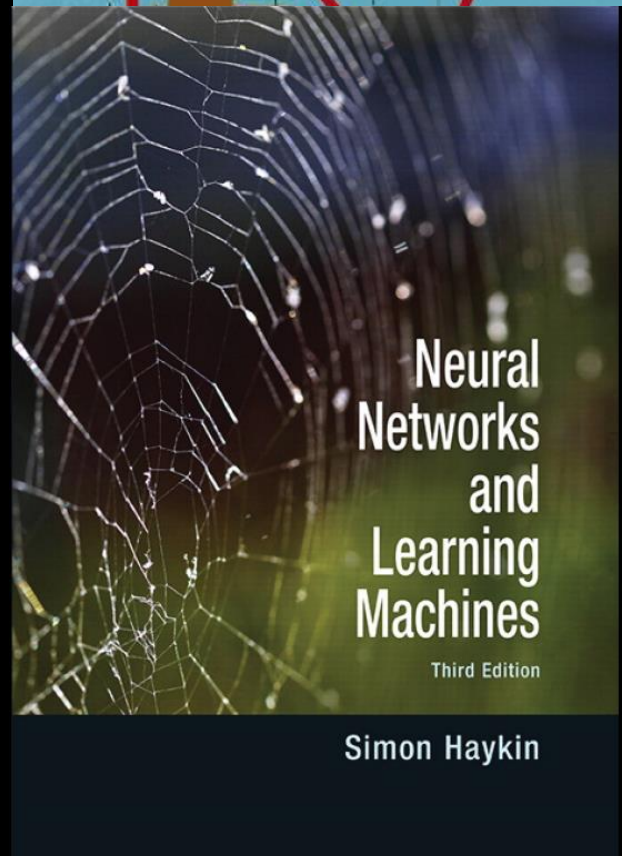
Eastern Mediterranean University
(emu.edu.tr)





4 Multi Layer Perceptrons





4.19 Small-scale Versus Large-scale Learning Problems

- In this section, we shall focus on distinguishing the small-scale and large-scale learning problems as two kinds of supervised learning.
- We begin the discussion with structural risk minimization (SRM), which is entirely statistical in nature;
- SRM is adequate for dealing with small-scale learning problems.
- Then we broaden the discussion by considering computational issues that assume a prominent role in dealing with large-scale learning problems.

4.19 Small-scale Versus Large-scale Learning Problems
Structural Risk Minimization

- The feasibility of supervised learning depends on the following key question:

Does a training sample consisting of N independent and identically distributed examples $(\boldsymbol{x}_1, d_1), (\boldsymbol{x}_2, d_2), \ldots, (\boldsymbol{x}_N, d_N)$ contain sufficient information to construct a learning machine capable of good generalization performance?

- The answer to this fundamental question lies in the method of structural risk minimization, described by Vapnik (1982, 1998).
- Let the natural environment responsible for generating the training sample be represented by the nonlinear regression model

$$d = f(\mathbf{x}) + \epsilon \qquad (4.158)$$

where $\mathbf{x}$ is the regressor, d is the response and ϵ is the modelling error.

4.19 Small-scale Versus Large-scale Learning Problems
Structural Risk Minimization

- To estimate $f(\,)$, define the expected risk (i.e. the average cost function) as

$$J_{\text{actual}}(f) = \mathbb{E}_{\mathbf{x},d}\left[\frac{1}{2}\left(d - f(\mathbf{x})\right)^2\right] \tag{4.159}$$

- Defining the conditional mean estimator as the minimum value of the cost function as

$$\hat{f}^* = \mathbb{E}[d|\mathbf{x}] \tag{4.160}$$

- Choosing a single-layer multilayer perceptron to do the machine learning, let the function $F(\mathbf{x}; \mathbf{w})$ denote the input–output relationship of the neural network, then we make our first approximation by setting

$$f(\mathbf{x}) = F(\mathbf{x}; \mathbf{w}) \tag{4.161}$$

- Then the model cost function

$$J(\mathbf{w}) = \mathbb{E}_{\mathbf{x},d}\left[\frac{1}{2}\left(d - F(\mathbf{x}; \mathbf{w})\right)^2\right] \tag{4.162}$$

4.19 Small-scale Versus Large-scale Learning Problems
Structural Risk Minimization

- Let $$\hat{\mathbf{w}}^* = \arg\min_{\mathbf{w}} J(\mathbf{w}) \quad (4.163)$$
- Even if we can find the minimizer $\hat{\mathbf{w}}^*$, it is likely that the resulting cost function will be worse than the minimized cost function $J_{\text{actual}}(\hat{f}^*)$, so,

$$J(\hat{\mathbf{w}}^*) > J_{\text{actual}}(\hat{f}^*) \quad (4.164)$$

- To ease the difficulty of not knowing the joint probability distribution of the pair $(\mathbf{x}, d)$, a second approximation is made by using the empirical risk.
- hence, the time-averaged energy function

$$\mathcal{E}_{av}(N;\mathbf{w}) = \frac{1}{2\mathrm{N}}\sum_{n=1}^{N}\big(d(n) - F(\mathbf{x}(n);\mathbf{w})\big)^2 \quad (4.165)$$

- whose minimization is defined by

$$\hat{w}_N = \arg\min_{\mathbf{w}} \mathcal{E}_{av}(N;\mathbf{w}) \quad (4.166)$$

4.19 Small-scale Versus Large-scale Learning Problems
Structural Risk Minimization

- The minimized cost function $J(\hat{\mathbf{w}}_N)$ is likely to be

$$J(\hat{\mathbf{w}}_N) > J(\hat{\mathbf{w}}^*) > J_{\text{actual}}(\hat{f}^*) \tag{4.167}$$

- By enlarging the size of the hidden layer, the approximate function $J(\hat{\mathbf{w}}^*)$ will be closer to the absolute optimum $J_{\text{actual}}(\hat{f}^*)$ by compromising the generalization capability of the multilayer perceptron.
  - The excess error $\{J(\hat{\mathbf{w}}^*) - J_{\text{actual}}(\hat{f}^*)\}$ will also increase unless training sample size N is also increased.
- Decomposing this excess error into two terms as

$$\underbrace{J(\hat{\mathbf{w}}_N) - J_{\text{actual}}(\hat{f}^*)}_{\text{Excess error}} = \underbrace{J(\hat{\mathbf{w}}_N) - J(\hat{\mathbf{w}}^*)}_{\text{Estimation error}} + \underbrace{J(\hat{\mathbf{w}}^*) - J_{\text{actual}}(\hat{f}^*)}_{\text{Approximation error}} \tag{4.168}$$

4.19 Small-scale Versus Large-scale Learning Problems
Structural Risk Minimization

- In this decomposition of errors, the following points are noteworthy:

  i) The *estimation error* provides a measure of how much performance is lost as a result of using a training sample of some prescribed size N.

  The approximation error is relevant in the assessment of network training.

  i) The *approximation error* provides a measure of how much performance is lost by choosing a model characterized by approximating function $F(\mathbf{x}, \mathbf{w})$. The estimation error is relevant in the assessment of network testing.

4.19 Small-scale Versus Large-scale Learning Problems
Structural Risk Minimization

- For the example of a single-layer multilayer perceptron, the larger the size of the hidden layer, the larger the *VC dimension* (i.e. h) will be.
- Considering a family of nested approximating network functions denoted by

$$\mathcal{F}_k = \{F(\mathbf{x};\mathbf{w})(\mathbf{w} \in W_k)\}, \qquad k = 1, 2, \ldots, K \tag{4.169}$$

- Such that $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \ \ldots \ \subset \mathcal{F}_K$, where $\subset$ means is contained in. The VC dimensions should also satisfy $h_1 < h_2 < \ \ldots < h_K$.
- Figure 4.29 plots variations of the approximation and estimation errors versus the size K of the family of approximating network functions $\mathcal{F}_k$.

4.19 Small-scale Versus Large-scale Learning Problems

Structural Risk Minimization

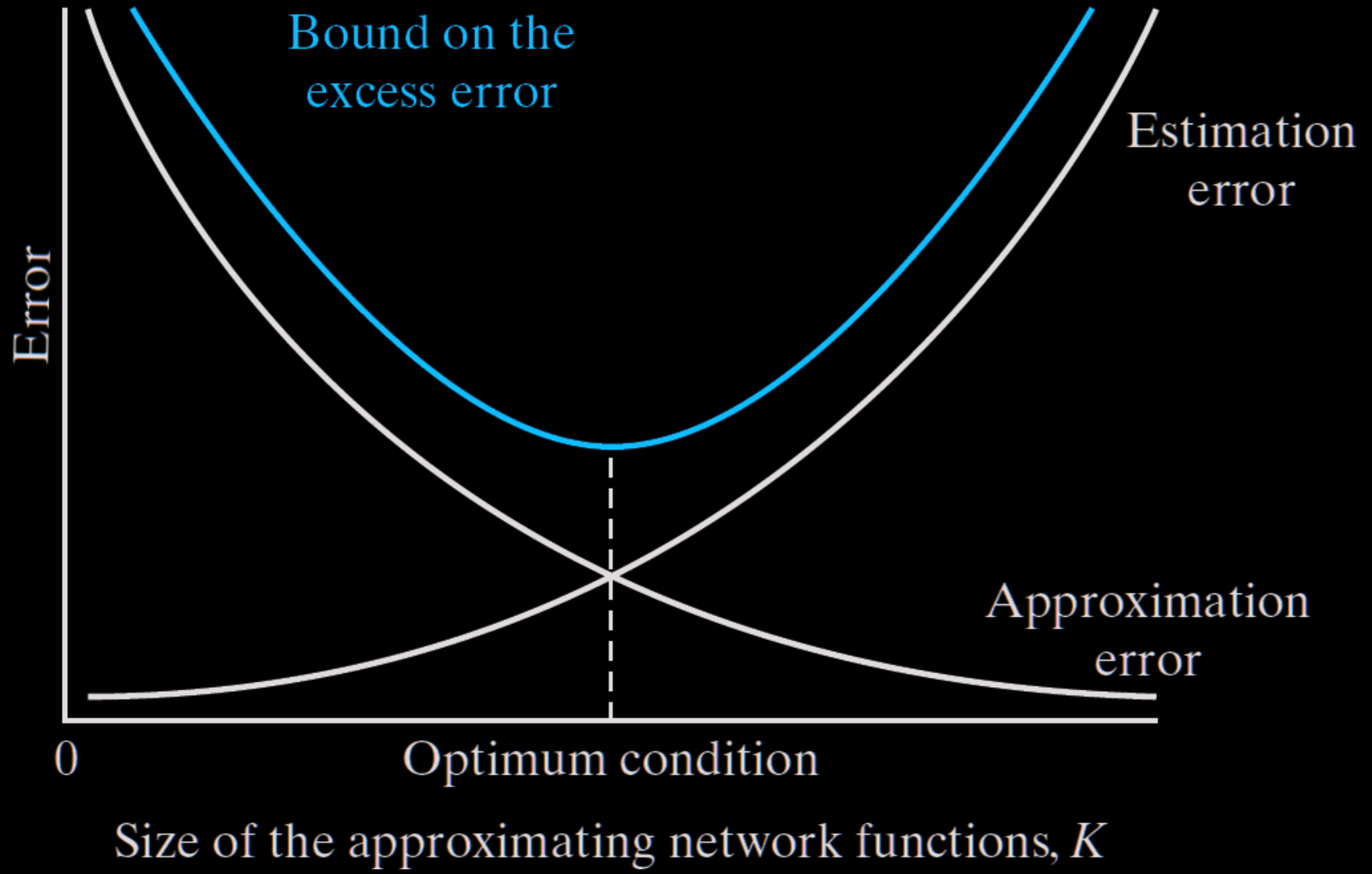
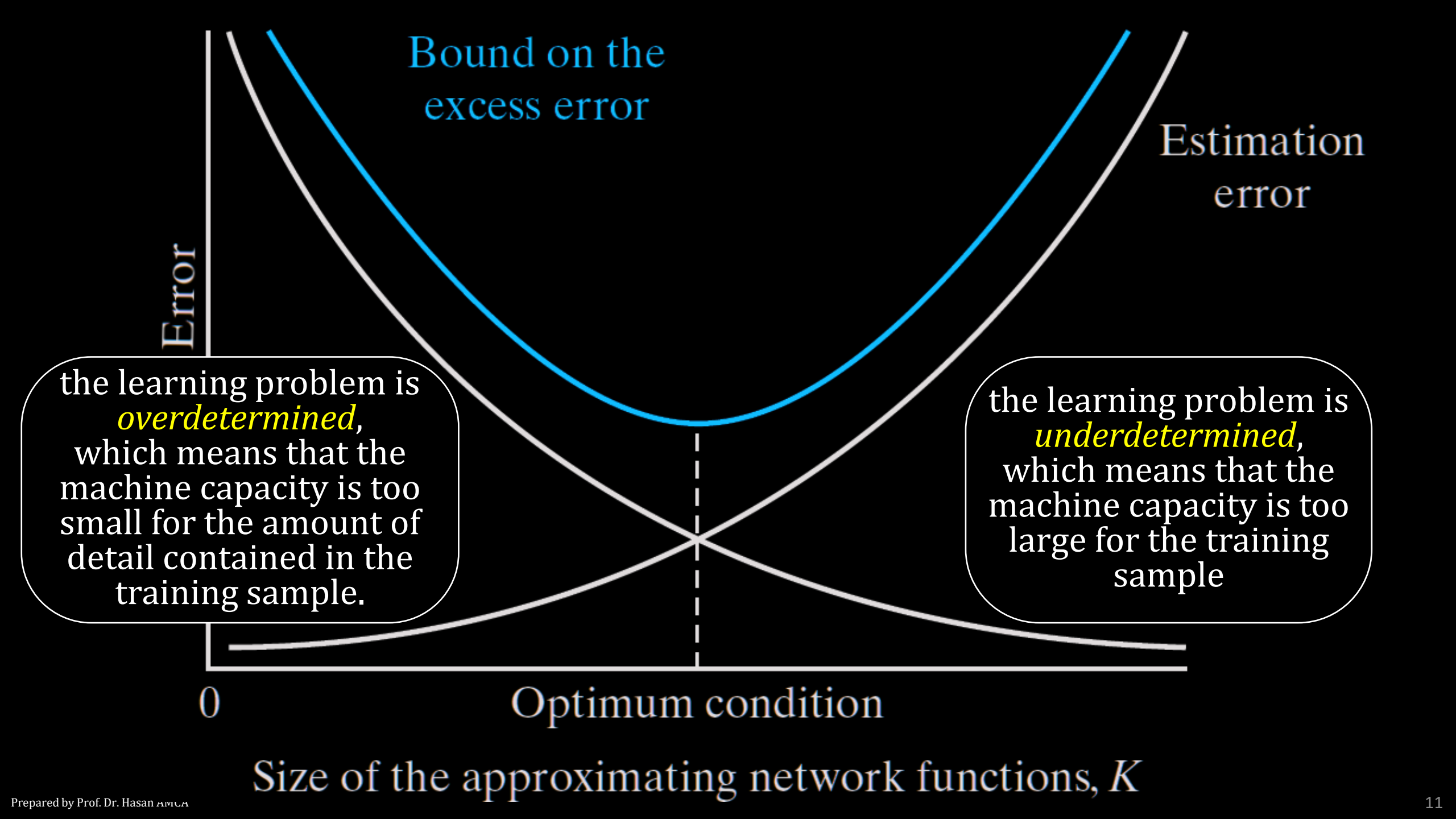


Fig. 4.29 Variations of the approximation and estimation errors with the size K.


Bound on the excess error
Estimation error
Error
the learning problem is *overdetermined*, which means that the machine capacity is too small for the amount of detail contained in the training sample.
the learning problem is *underdetermined*, which means that the machine capacity is too large for the training sample
0
Optimum condition
Size of the approximating network functions, K

4.19 Small-scale Versus Large-scale Learning Problems
Computational Considerations

- Suppose that we settle on a network model characterized by the weight vector $\tilde{\mathbf{w}}_N$ which is different from $\hat{\mathbf{w}}_N$.
- For example, the on-line learning algorithm could be stopped long before convergence has been reached, due to limited computing time.
- In any event, $\tilde{\mathbf{w}}_N$ is a suboptimal solution that satisfies the condition

$$\mathcal{E}_{av}(N;\tilde{\mathbf{w}}_N) \leq \mathcal{E}_{av}(N;\hat{\mathbf{w}}_N) + \rho \qquad (4.170)$$

- Where ρ constitute a new controlled variable; it provides a measure of computational accuracy.

4.19 Small-scale Versus Large-scale Learning Problems
Computational Considerations

- We must now adjust three controlled variables:
  1. the network model (i.e. number of hidden neurons in a multilayer perceptron),
  2. the number of training examples, and
  3. the optimization accuracy (i.e. by prematurely terminating computation of the minimizer $\hat{\mathbf{w}}_N$ and settling on the suboptimal solution $\tilde{\mathbf{w}}_N$).
- In order to hit the best test performance, we have to satisfy *budget constraints*, which define the maximum number of training examples that we can use and the maximum computing time that we can afford.

4.19 Small-scale Versus Large-scale Learning Problems
Definitions (Small-Scale and Large-Scale Learning)

- According to the *active budget constraint* that distinguishes one learning problem from the other (Bottou, 2007), the small-scale and large-scale learning problems are respectively defined as follows:

Definition I. Small-scale learning

- A supervised-learning problem is said to be small-scale when the *size of the training sample* (i.e., the number of examples) is the active budget constraint imposed on the learning process.

Definition II. Large-scale learning

- A supervised-learning problem is said to be large-scale when the *computing time* is the active budget constraint imposed on the learning process.

4.19 Small-scale Versus Large-scale Learning Problems
Definitions (Small-Scale and Large-Scale Learning)

- As an example of a *small-scale learning problem*, consider the design of an adaptive equalizer to compensate for the distortion of information-bearing data transmitted over a communication channel.

  The LMS algorithm, rooted in stochastic gradient descent is widely used for solving this on-line learning problem (Haykin, 2002).

- As an example of a *large-scale learning problem*, consider the design of a check reader where the training examples consist of joint pairs, each of which describes a particular {*image*, *amount*} pair, where "*image*" is associated with a check and "*amount*" pertains to the amount of money inscribed in the check (Bottou, 2007).

4.19 Small-scale Versus Large-scale Learning Problems

Small-Scale Learning Problems

- In small-scale learning problems, there are three variables available to the designer of a learning machine:

  1. the number of training examples, N;
  2. the permissible size K of the family of approximating network functions $\mathcal{F}$;
  3. the computational error ρ introduced in Eq. (4.170).

- With the active budget constraint being the number of examples, the design options in learning problems of the first kind are as follows (Bottou, 2007):
  - Reduce the estimation error by making N as large as the budget permits.
  - Reduce the optimization error by setting the computational error $\rho = 0$, which means setting $\tilde{\mathbf{w}}_N = \hat{\mathbf{w}}_N$.
  - Adjust the size of $\mathcal{F}$ to the extent deemed to be reasonable.

4.19 Small-scale Versus Large-scale Learning Problems

Small-Scale Learning Problems

- With $\rho = 0$, the method of structural risk minimization, involving the approximation–estimation tradeoff illustrated in Fig. 4.29, is adequate for dealing with small-scale learning problems.

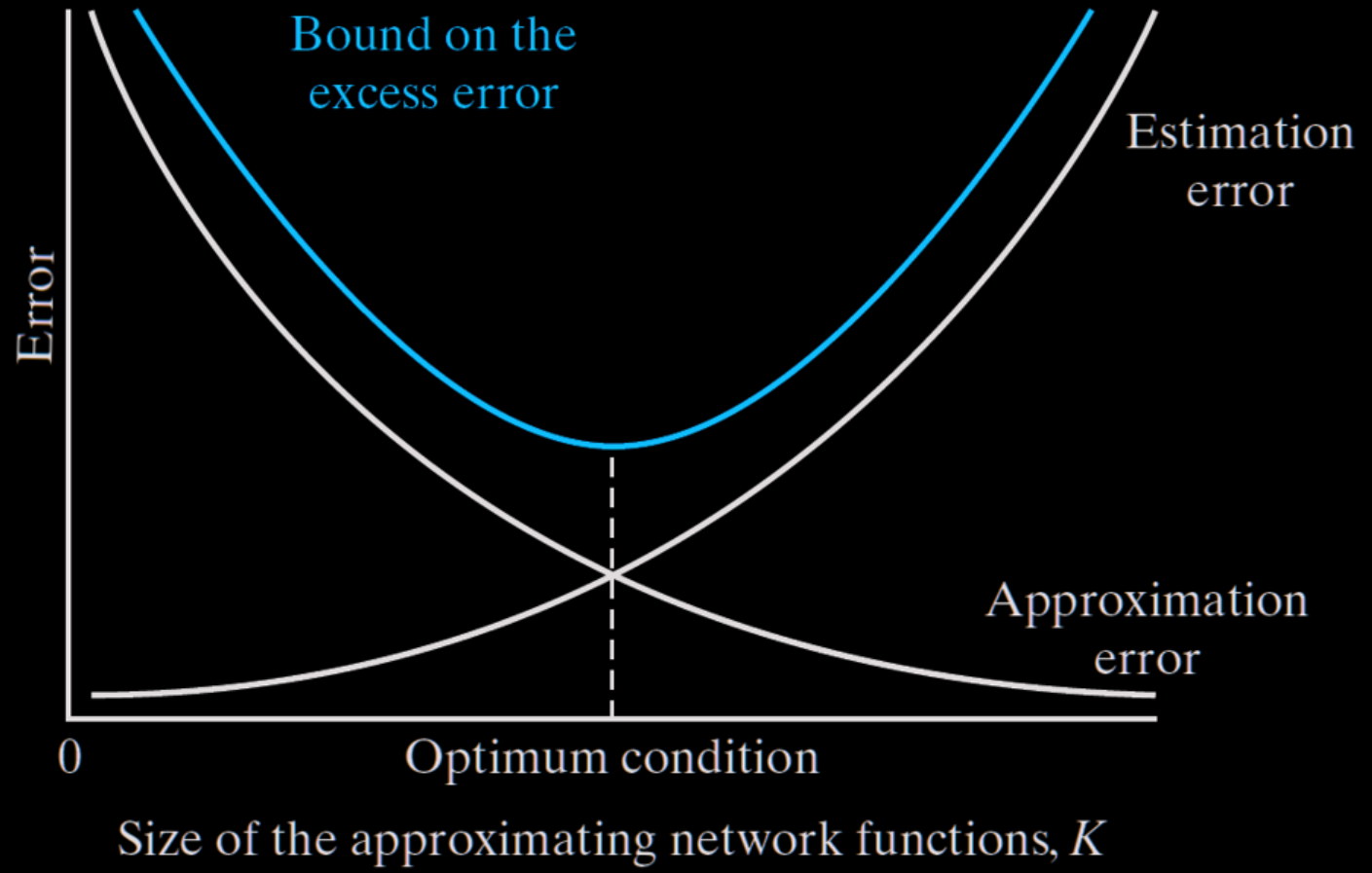


Fig. 4.29 Variations of the approximation and estimation errors with the size K.

4.19 Small-scale Versus Large-scale Learning Problems

Large-Scale Learning Problems

- The active budget constraint in large-scale learning problems is the computing time.
- In large-scale learning problems, the excess error is defined by the difference $J(\tilde{\mathbf{w}}_N) - J_{\text{actual}}(\hat{f}^*)$, which is decomposed into three terms, as shown by the following (Bottou, 2007):

$$\underbrace{J(\tilde{\mathbf{w}}_N) - J_{\text{actual}}(\hat{f}^*)}_{\text{Excess error}} = \underbrace{J(\tilde{\mathbf{w}}_N) - J(\hat{\mathbf{w}}_N)}_{\text{Optimization error}} + \underbrace{J(\hat{\mathbf{w}}_N) - J(\hat{\mathbf{w}}^*)}_{\text{Estimation error}} + \underbrace{J(\hat{\mathbf{w}}^*) - J_{\text{actual}}(\hat{f}^*)}_{\text{Approximation error}} \quad (4.171)$$

- The only difference between small-scale and large-scale learning problems is the first term in Eq. (4.171). This new term, called the optimization error, is obviously related to the computational error ρ.

4.19 Small-scale Versus Large-scale Learning Problems

Large-Scale Learning Problems

- We analyze Eq. (4.171) in terms of convergence rates rather than bounds. Then, the requirement is to minimize the sum of the three terms by adjusting the available variables:
  1) the number of examples, N;
  2) the permissible size K of approximating network functions, $\mathcal{F}_K$;
  3) the computational error ρ, which is no longer zero.
- The elements for an approximate solution to trade-offs in tackling large-scale learning problems.
- Most importantly, the trade-offs depend on the choice of the optimization algorithm.

4.19 Small-scale Versus Large-scale Learning Problems
Large-Scale Learning Problems

- Figure 4.30 illustrates how a plot of log ρ versus log T is affected by the type of optimization algorithm used to solve a large-scale learning problem.
- Three categories of optimization algorithms are identified in this Fig. 4.30,
  - namely, *bad*, *mediocre*, and *good*
  - examples of which respectively include stochastic gradient descent (i.e., on-line learning), gradient descent (i.e., batch learning), and second-order gradient descent (e.g., quasi-Newton optimization algorithm of the BFGS kind or its extension). Table 4.4 summarizes the distinguishing features of these three categories of optimization algorithms.
- Table 4.4 summarizes the distinguishing features of these three categories of optimization algorithms.

4.19 Small-scale Versus Large-scale Learning Problems

Large-Scale Learning Problems

TABLE 4.4 Summary of Statistical Characteristics of Three Optimization Algorithms		
Algorithm	**Cost per iteration**	**Time to Reach ρ**
1. Stochastic gradient descent (on-line learning)	$O(m)$	$O(1/p)$
2. Gradient descent (batch learning)	$O(nm)$	$O(\log(1/p))$
3. Second-order radient descent (on-line learning)	$O(m(m+n))$	$O(\log(\log(1/p)))$

m : dimension of input vector **x**
N : number of examples used in training
ρ: computational error
*This table is compiled from Bottou (2007).

4.19 Small-scale Versus Large-scale Learning Problems

Large-Scale Learning Problems

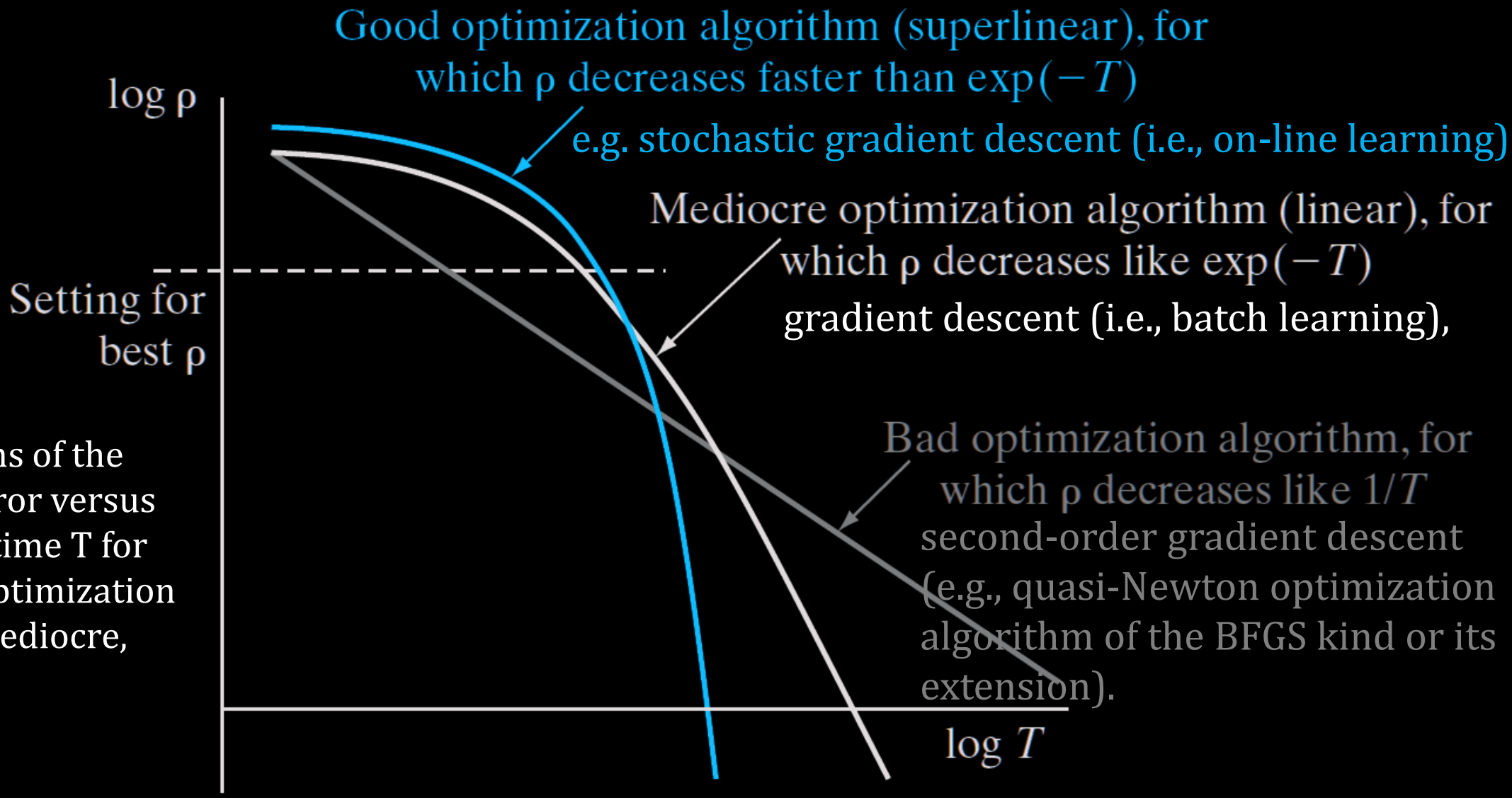
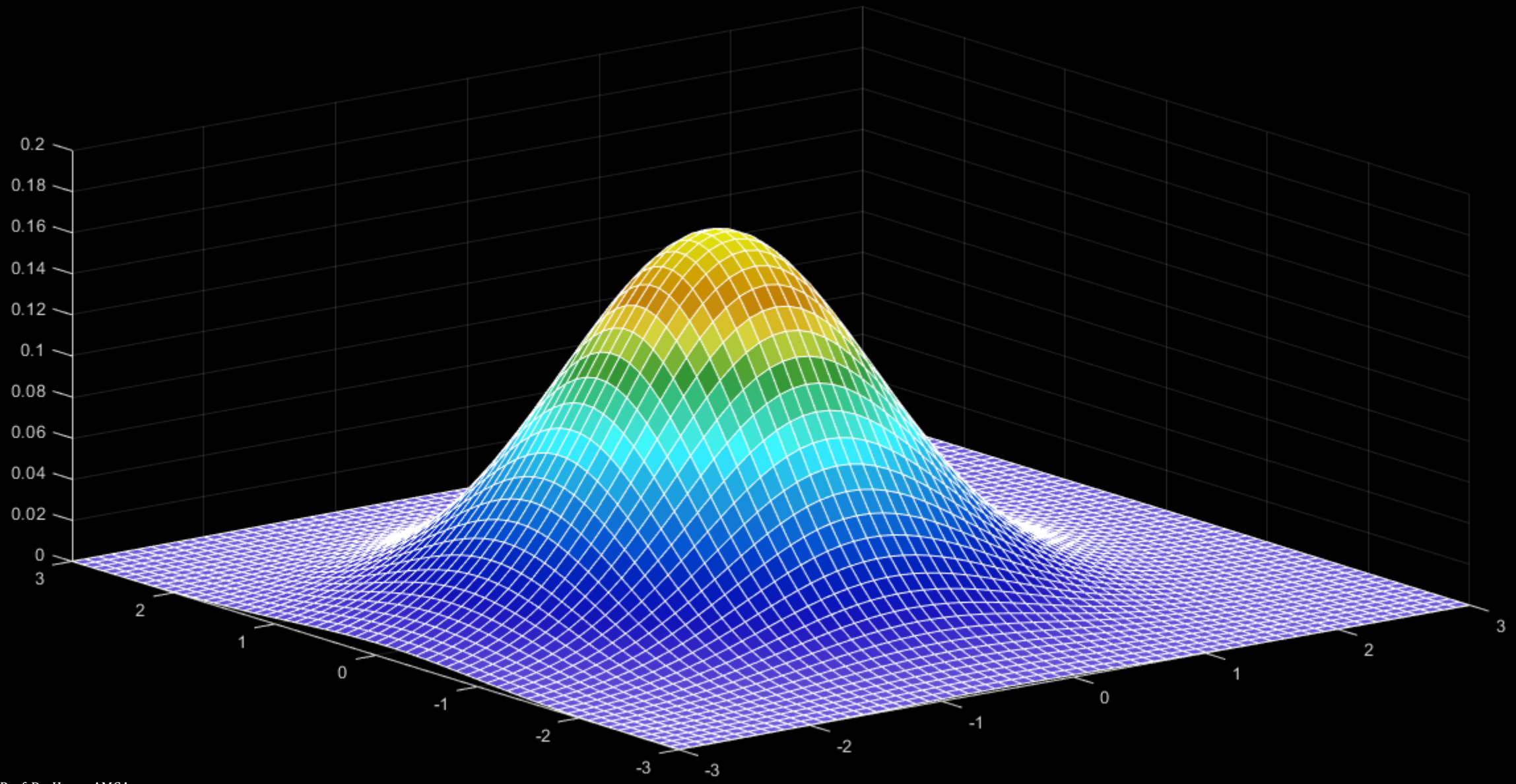


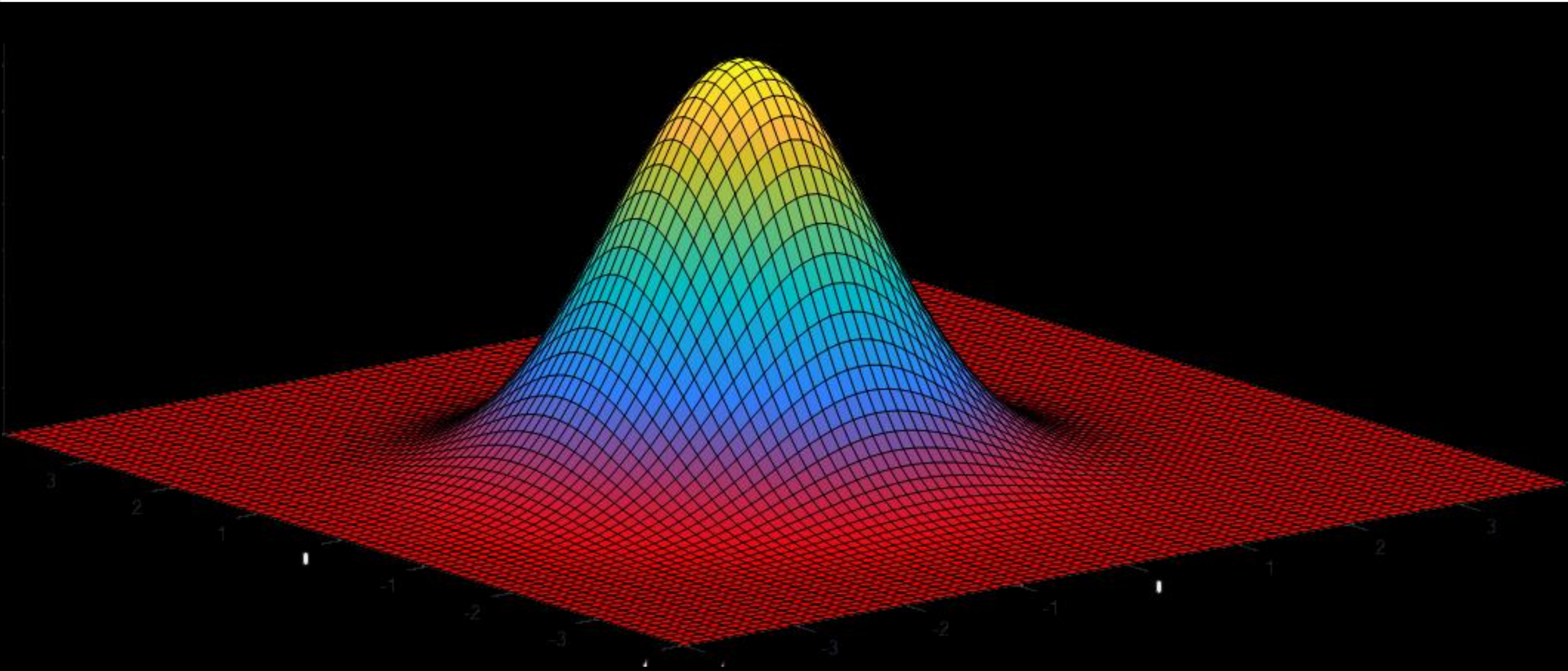
Fig. 4.30 Variations of the computational error versus the computation time T for three classes of optimization algorithm: bad, mediocre, and good.

End of Section 4.19

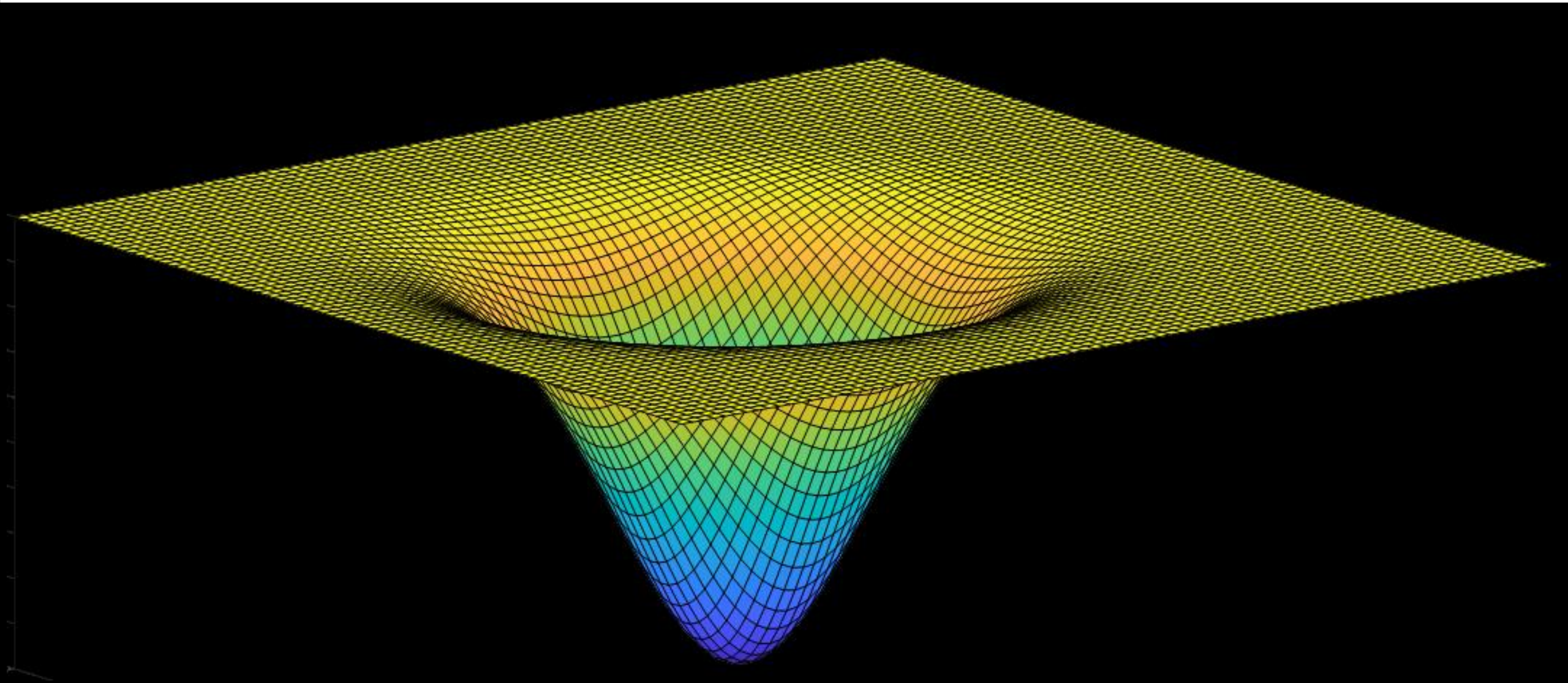
Multivariate Gaussian PDF Plot



Multivariate Gaussian PDF Plot



Multivariate Gaussian PDF Plot



Tips and Tricks:
Getting Started Using Optimization with MATLAB

https://www.mathworks.com/videos/tips-and-tricks-getting-started-using-optimization-with-matlab-81594.html?s_iid=doc_rw_GD_footer