EENG582: Artificial Neural Networks

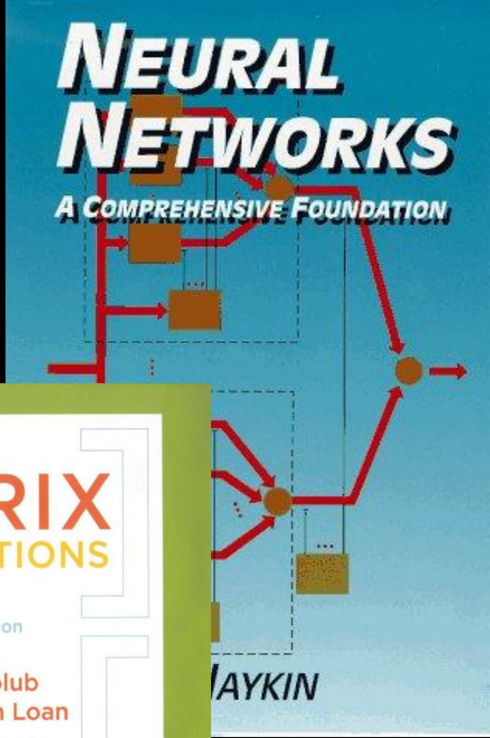
Reference 1: **Neural Networks**
A Comprehensive Foundation
by S. Haykin

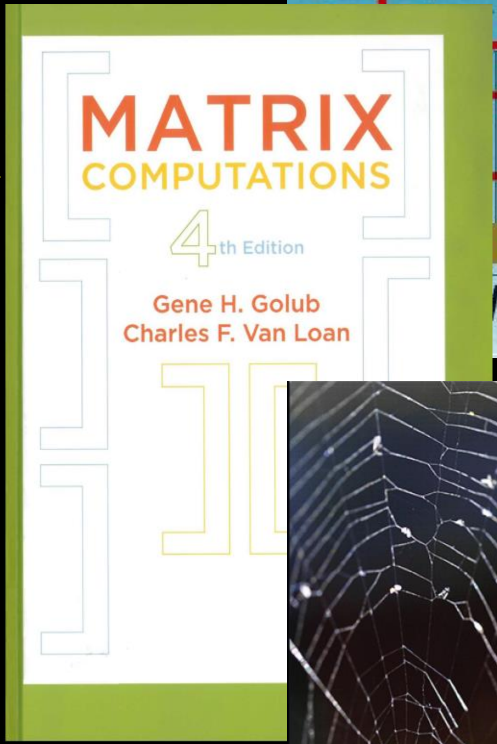
Sections 3.1 – 3.2 – Adaptive Filtering + Basic Optimization

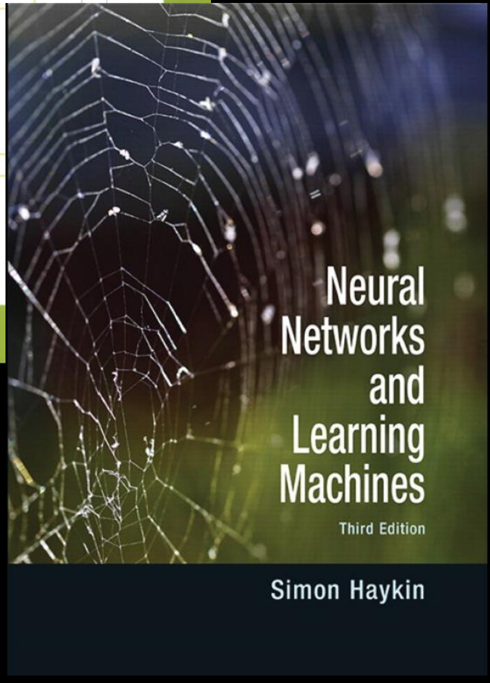
3. Single Layer Perceptrons
Adaptive Filtering + Optimization

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

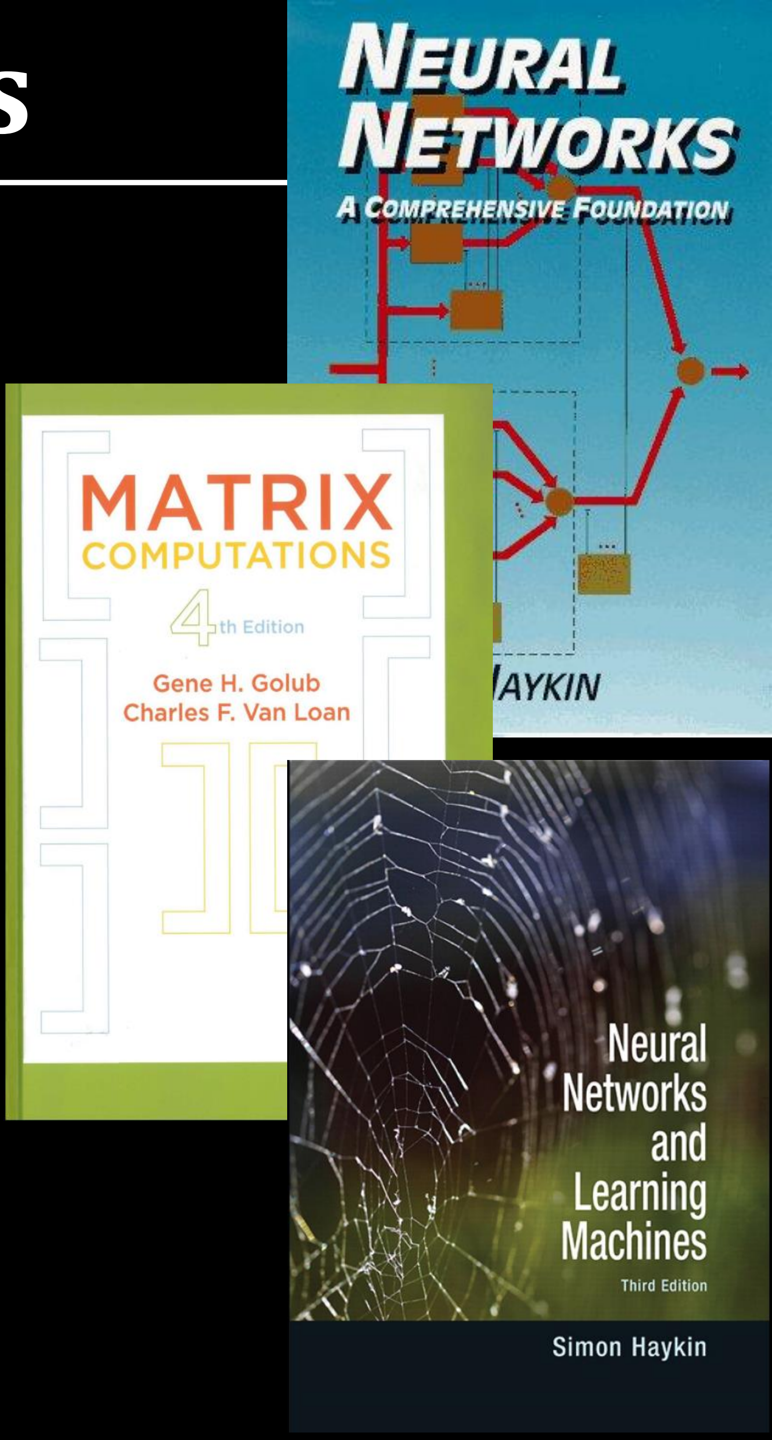
Reference 2: Matrix Computations, 4th Ed. By Golub and Van Loan - Chapter 11, Large Sparse Linear System Problems







3. Single Layer Perceptrons



Converting a Solution to an Optimization Problem
Adversarial Problem

- Neural networks have the reputation of providing bad probability estimates and they suffer from adversarial examples
- In short: neural networks are often highly confident even when they are wrong.
- This can be an issue when they are deployed in real-life scenarios (e.g. self-driving cars).
- A self-driving car should be certain when making decisions at 90 mph.

Converting a Solution to an Optimization Problem
Objective Function and Constraints

Objective Function for neural network

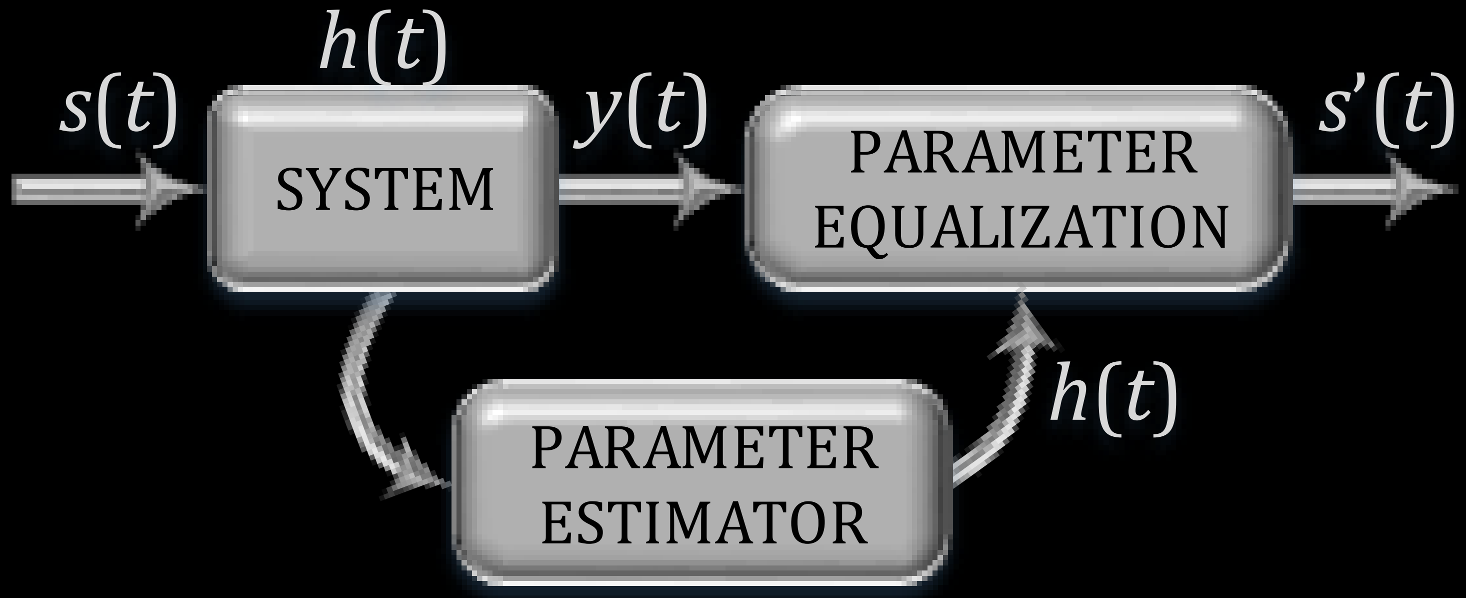
- Typically, with neural networks, we seek to minimize the error. As such, the objective function is often referred to as a cost function or a loss function and the value calculated by the loss function is referred to as simply "loss."

Constraint Function

- In mathematics, a constraint is a condition of an optimization problem that the solution must satisfy.
- There are several types of constraints
  - Primarily equality constraints,
  - inequality constraints, and
  - integer constraints.

Converting a Solution to an Optimization Problem

- Assume that we have a system as depicted in the following figure:
- In order to recover the estimate of the original signal $s'(t) = y(t)[h(t)]^{-1}$, the inverse of the system impulse response $[h(t)]^{-1}$ is needed.
- In matrix form, the received signal can be recovered as $\mathbf{S}' = \mathbf{Y}\mathbf{H}^{-1}$



Converting a Solution to an Optimization Problem

- The zero-forcing equalizer requires inverse of the system impulse response ($\mathbf{H}^{-1}$) in order to recover the message signal $\mathbf{S}$ from the system output $\mathbf{Y}$.
  - provided that system is identified through its impulse response.
- In systems with large number of parameters, obtaining inverse of the matrix $\mathbf{H}^{-1}$ can be computationally costly.
- Hence, the message signal can then be estimated non-iteratively as

$$\mathbf{S}' = \mathbf{Y}\mathbf{H}^{-1} \tag{1}$$

- The problem shown in (1) can be transformed into an optimization problem that can be solved iteratively without requiring a matrix inversion, such as

$$\mathbf{S}' = \mathbf{Y}\mathbf{H}^{-1} \quad \rightarrow \quad \mathbf{A}\mathbf{x} = \mathbf{b} \tag{2}$$

where $\mathbf{A}$ is a positive definite symmetric matrix defined as $\mathbf{A} = \mathbf{H}$ and vector $\mathbf{b}$ is the identity matrix $\mathbf{b} = \mathrm{vec}(\mathbf{I}_K)$.

Converting a Solution to an Optimization Problem

- The problem in (2) is same as solving the unconstraint optimization problem

$$\min_{\mathbf{x}\in\mathbb{C}^n} f(\mathbf{x}) \tag{3}$$

where the cost function $f(\mathbf{x})$, can be written as

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T(\mathbf{I}_K \otimes \mathbf{A})\mathbf{x} - \mathbf{b} \tag{4}$$

- The $f(\mathbf{x})$ is convex and the gradient of the cost function can be expressed as

$$\nabla f(x) = (\mathbf{I}_K \otimes \mathbf{A})\mathbf{x} - \mathbf{b} \tag{5}$$

where $\otimes$ is the Kronecker product. (Matrix Computations, 4th Ed. Pp.27)

- To find near optimal solution for the analog precoder, we propose the BB gradient algorithm for minimization for hybrid precoding.

Mathematical Optimization

- A mathematical optimization problem or just optimization problem has the form

$$\begin{aligned} &\text{minimize} && f_0(\mathbf{x}) \\ &\text{subject to} && f_i(\mathbf{x}) \le b_i,\ \ i = 1, 2, \dots, m. \end{aligned} \qquad (1)$$

- Here, the vector $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is the *optimization variable* of the problem, the function $f_0 : \mathbb{R}^n \to \mathbb{R}$ is the objective function, the functions $f_i : \mathbb{R}^n \to \mathbb{R}, i = 1, 2, \dots, m$, are the (inequality) *constraint functions*, and the constants $b_1, b_2, \dots, b_m$ are the *limits*, or *bounds*, for the constraints.

- A vector $\mathbf{x}^*$ is called *optimal*, or a *solution* of the problem (1), if it has the smallest objective value among all vectors that satisfy the constraints: for any z with $f_1(\mathbf{z}) \le b_1, \dots, f_m(\mathbf{z}) \le b_m$, we have $f_0(\mathbf{z}) \ge f_0(\mathbf{x}^*)$.

Source: [3] Stephen Boyd, Lieven Vandenberghe. (2004). Convex Optimization. Cambridge University Press, Cambridge 2004

Mathematical Optimization

- We generally consider families or classes of optimization problems, characterized by particular forms of the objective and constraint functions.
- As an important example, the optimization problem (1) is called a *linear program* if objective and constraint functions $f_0, \ldots, f_m$ are linear, i.e., satisfy

$$f_i(\alpha \mathbf{x} + \beta \mathbf{y}) = \alpha f_i(\mathbf{x}) + \beta f_i(\mathbf{y}) \quad (2)$$

- A convex optimization problem, however, is one in which the objective and constraint functions are convex, which means they satisfy the inequality:

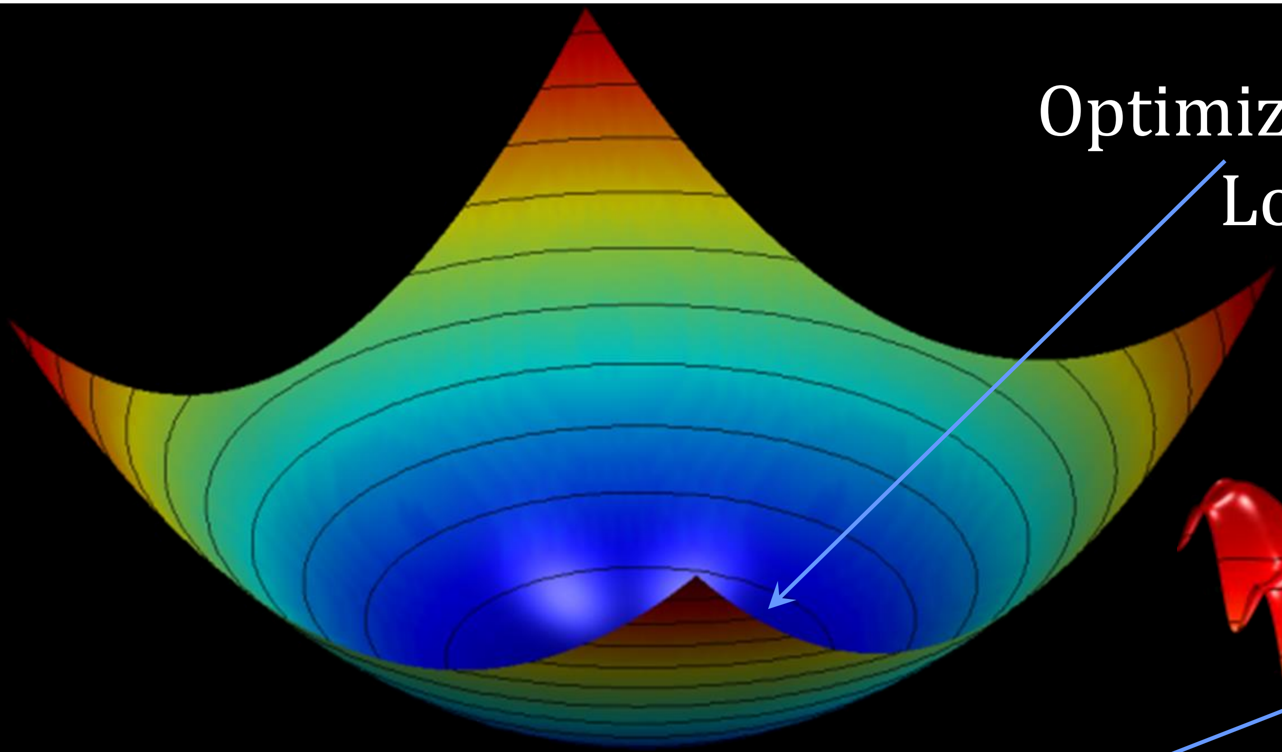
$$f_i(\alpha \mathbf{x} + \beta \mathbf{y}) \leq \alpha f_i(\mathbf{x}) + \beta f_i(\mathbf{y}) \quad (3)$$

For all $x, y \in \mathbb{R}^n$ and $\alpha, \beta \in \mathbb{R}$.

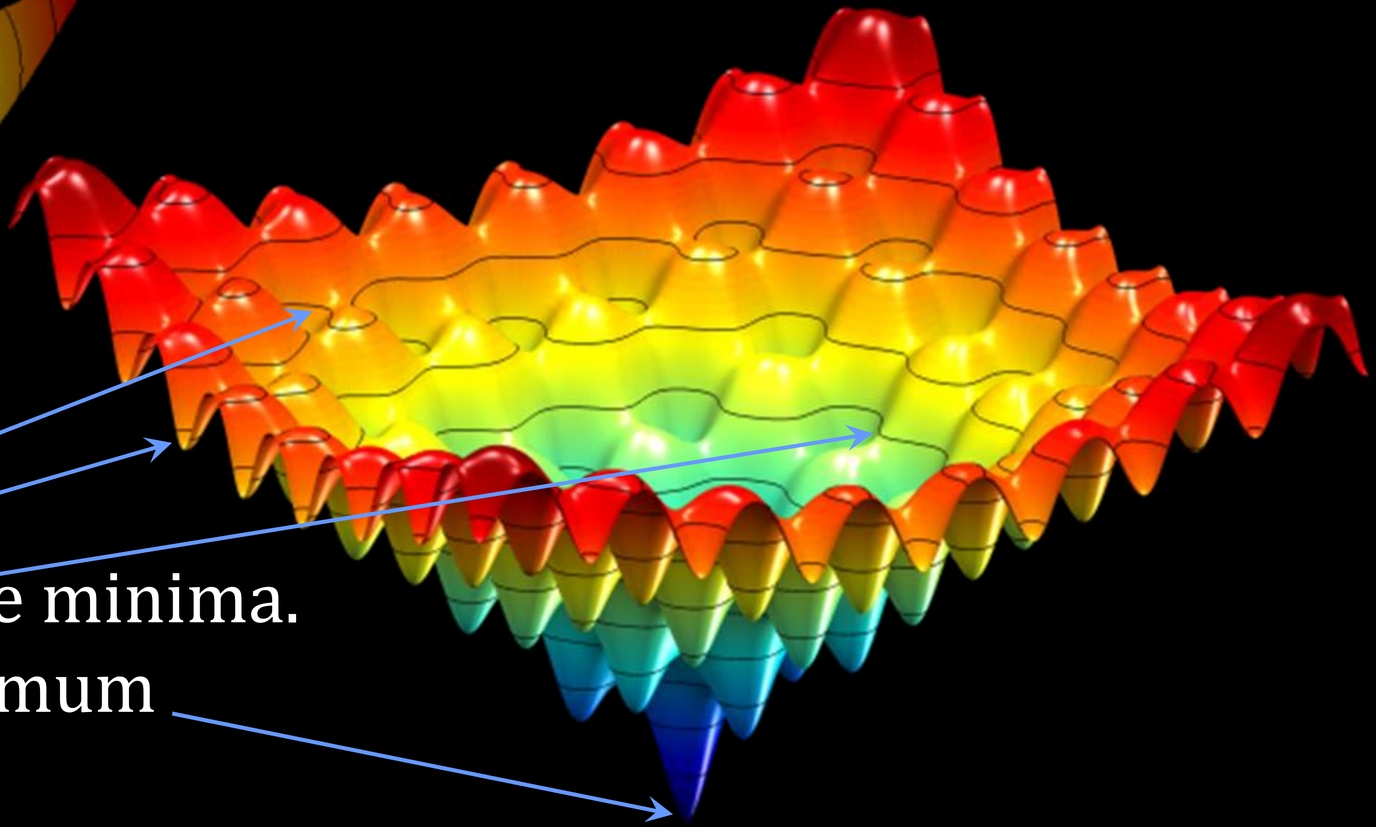
- If the optimization problem is not linear, it is called a *nonlinear program*.

Source: [3] Stephen Boyd, Lieven Vandenberghe. (2004). Convex Optimization. Cambridge University Press, Cambridge 2004

Mathematical Optimization



Optimization problem with a single minimum.
Local minimum = Global minimum



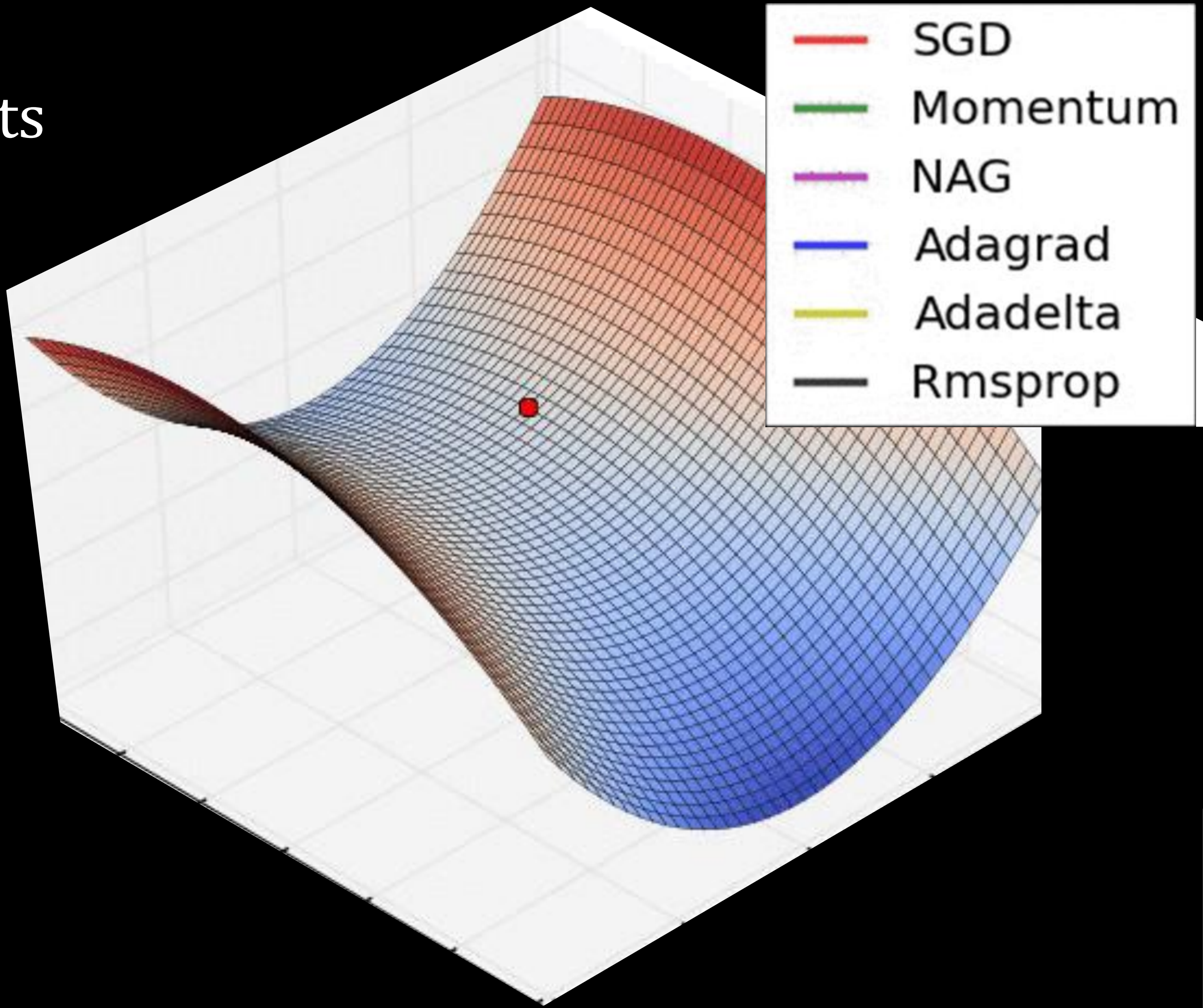
Optimization problem with multiple minima.
Local minimum ≠ Global minimum

https://www.cs.bham.ac.uk/research/projects/ecb/displayDataset.php?id=2

Mathematical Optimization
Types of Current Optimizers

Various optimizers are researched within the last few couples of years each having its advantages and disadvantages.

- Gradient Descent
- Stochastic Gradient Descent (SGD)
- Mini Batch Stochastic Gradient Descent (MB-SGD)
- SGD with momentum
- Nesterov Accelerated Gradient (NAG)
- Adaptive Gradient (AdaGrad)
- AdaDelta
- RMSprop
- Adam



https://www.kdnuggets.com/2020/12/optimization-algorithms-neural-networks.html

Mathematical Optimization
More Types of Current Optimizers

Name	Abb.	Cost function	**Sec** Downhill Simplex
Gradient Descent	GD	Once differentiable	2.2.2.3
Newton		Twice differentiable	2.2.2.4
Gauss-Newton	GN	Sum of squared functions non-linear least-squares (NLS)	2.2.2.5
Levenberg-Marquardt	LM	NLS	2.2.2.7
Cholesky Factorization		Linear system of equations (with positive definite) and, coincidently, sum of squared linear functions linear least-squares (LLS)	2.2.2.8
QR Factorization		LLS	2.2.2.9
Singular Value Decomposition	SVD	LLS	2.2.2.10
Iteratively Reweighted Least-Squares	IRLS	Weighted sum of squared functions with variable weights	2.2.2.11
Golden Section Search		Mono-variate, scalar-valued, uni-modal	2.2.2.12

More on Optimization Algorithms https://www.brnt.eu/phd/node10.html

Optimization in Economics

Economic optimization, including competitive production costs, is the ultimate goal of sound reservoir management.

It involves building multiple scenarios or alternative approaches in order to arrive at the optimum solution.

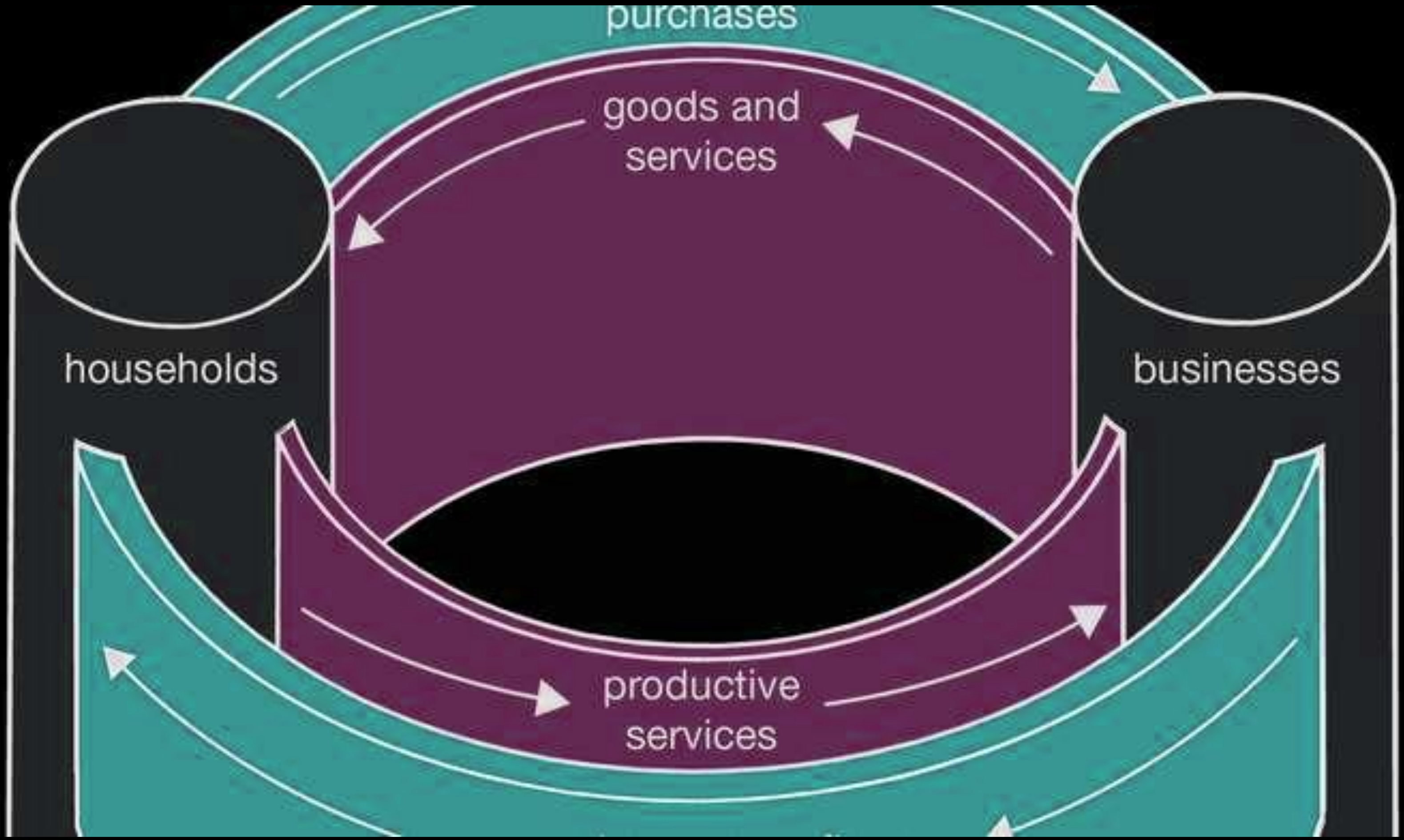


Diagram illustrating the flow of money, goods, and services in a modern industrial economy

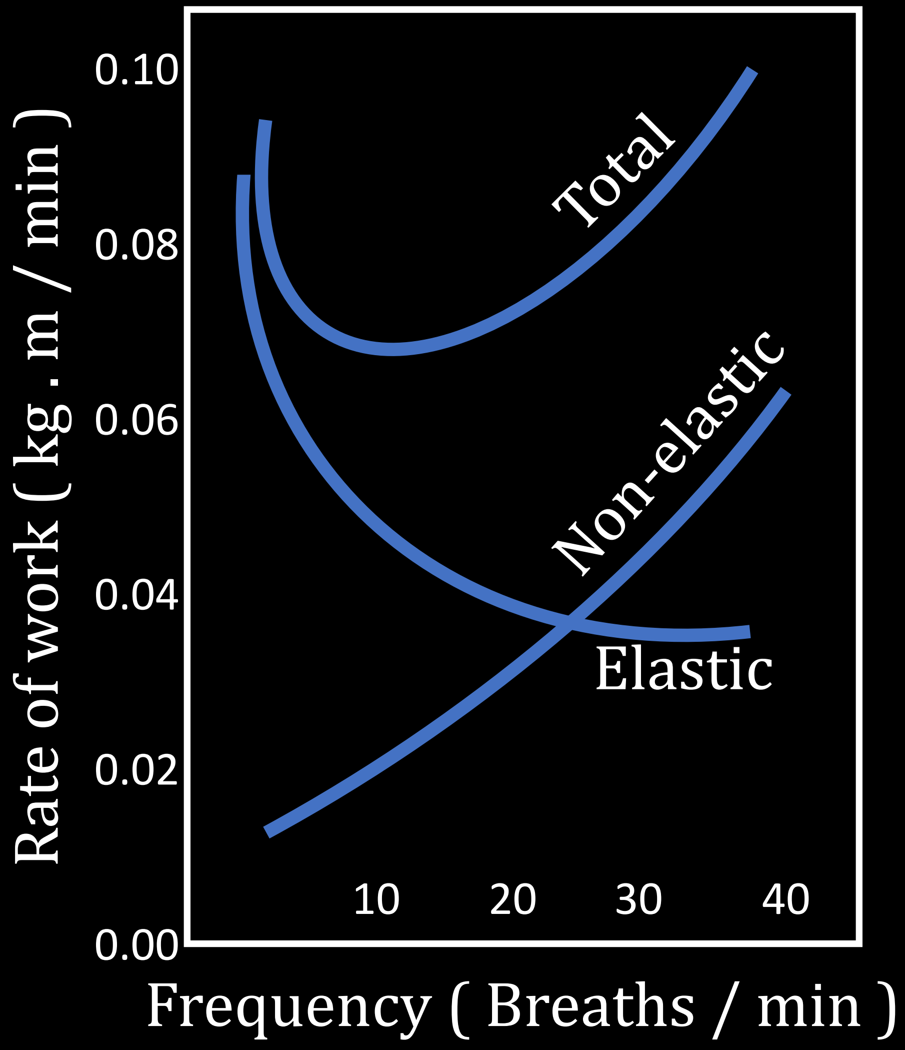
https://www.britannica.com/topic/economics

Optimization in Biology

Optimization aims to make a system or design as effective or functional as possible

The work rate of breathing is the sum of resistive (nonelastic) work and compliance (elastic) work.

The sum reaches a minimum at some particular frequency.



https://jbioleng.biomedcentral.com/articles/10.1186/1754-1611-7-6

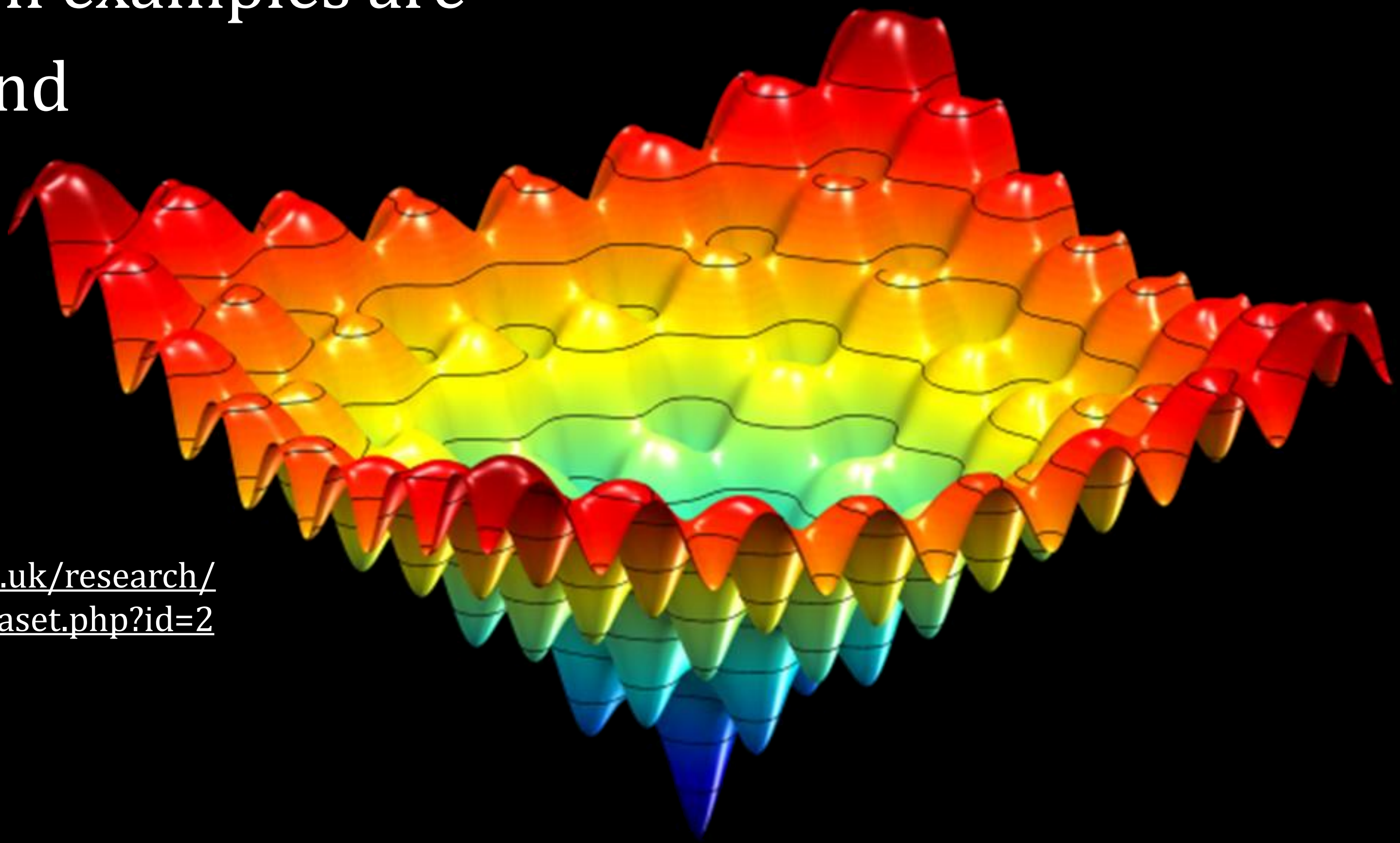
Solving Optimization Problems

- A *solution* for a class of optimization problems is an algorithm that computes a solution of the problem to some targeted accuracy.
- Effectiveness of these algorithms, i.e. ability to solve optimization problem in (1), varies considerably, and depends on factors such as form of objective and constraint functions, # variables and constraints and special structure such as sparsity.
  - A problem is sparse if each constraint function depends on only a small number of the variables.
- Optimization problems are usually difficult to solve.
  - Therefore, compromises such as very long computation time or possibility of not finding the solution are likely to happen.

Source: [3] Stephen Boyd, Lieven Vandenberghe. (2004). Convex Optimization. Cambridge University Press, Cambridge 2004

Solving Optimization Problems

- There are variety of effective algorithms that can reliably solve even large problems, with hundreds or thousands of variables and constraints.
- Two important and well-known examples are
    1. least-squares problems and
    2. linear programs.



Source
https://www.cs.bham.ac.uk/research/projects/ecb/displayDataset.php?id=2

Source: [3] Stephen Boyd, Lieven Vandenberghe. (2004). Convex Optimization. Cambridge University Press, Cambridge 2004

Solving Optimization Problems
The Least-Squares

- The least-Squares problem is an optimization problem with no constraints (i.e. $m = 0$) and an objective which is a sum of squares of terms of the form $a_i^T x - b_i$:

$$\text{Minimize} \;\; f_0(x) = \|Ax - b\|_2^2 = \sum_{i=1}^{k} \left(a_i^T x - b_i\right)^2 \quad (4)$$

Here, $A \in \mathbb{R}^{k \times n}$ (with $k \geq n$), a_i^T are the rows of A and the vector $x \in \mathbb{R}$ is the optimization variable.

- The solution of the Least-Squares problem in (4) can be reduced to solving a set of linear equations

$$(A^T A)x = A^T b$$

So, we have to solve for x analytically as $x = (A^T A)^{-1} A^T b$.

Source: [3] Stephen Boyd, Lieven Vandenberghe. (2004). Convex Optimization. Cambridge University Press, Cambridge 2004

Solving Optimization Problems
The Least-Squares

- Recognizing an optimization problem as a *least-squares problem* is straight-forward; we only need to verify that the objective is a quadratic function (and then test whether the associated quadratic form is positive semidefinite).

- In *weighted least-squares*, which is our focus in this course, the weighted least-squares cost is minimized as:

$$\sum_{i=1}^{k} w_i\left(a_i^T x - b_i\right)^2$$

where the weights $w_1, w_2, \dots, w_k$ are positive. In a statistical setting, *weighted least-squares* arises in estimation of a vector **x**, given linear measurements corrupted by errors with unequal variances.

Source: [3] Stephen Boyd, Lieven Vandenberghe. (2004). Convex Optimization. Cambridge University Press, Cambridge 2004

Solving Optimization Problems
Linear Programming

- In linear programming optimization problems, the objective and all constraint functions are linear:

$$\begin{aligned} &\text{minimize} \quad c^T x \\ &\text{subject to} \quad a_i^T x \leq b_i, \ \ i = 1, 2, \dots, m. \end{aligned} \tag{5}$$

Here the vectors **c**, $a_1, \dots, a_m \in \mathbb{R}^n$ and scalars $b_1, \dots, b_\mathrm{m} \in \mathbb{R}$ are problem parameters that specify objective and constraint functions.

- However, like least-squares, solving linear programs is a mature technology. Linear programming solvers are embedded in many tools and applications.

Source: [3] Stephen Boyd, Lieven Vandenberghe. (2004). Convex Optimization. Cambridge University Press, Cambridge 2004

Solving Optimization Problems
Linear Programming

- As a simple example to a linear programming problem, consider the Chebyshev approximation problem:

$$\text{minimize} \quad \max_{i=1,\dots,k} |a_i^T x - b_i| \tag{6}$$

Where $x \in \mathbb{R}^n$ is the variable and $a_1, \dots, a_k \in \mathbb{R}^n$, $b_1, \dots, b_k \in \mathbb{R}$ are parameters that specify the problem instance.

- The Chebyshev approximation problem (6) can be solved by solving the linear program

$$\begin{aligned} \text{minimize} \quad & t \\ \text{subject to} \quad & a_i^T x - t \leq +b_i, \; i = 1, 2, \dots, k \\ & -a_i^T x - t \leq -b_i, \; i = 1, 2, \dots, k \end{aligned} \tag{7}$$

Where $x \in \mathbb{R}^n$ and $t \in \mathbb{R}$.

Source: [3] Stephen Boyd, Lieven Vandenberghe. (2004). Convex Optimization. Cambridge University Press, Cambridge 2004