ECE6255
Digital Processing of Speech Signals

Lecture 2-2: Optimization Theory Essentials

Chin-Hui Lee
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA
chl@ece.gatech.edu

Optimization Tools: An Overview

- The nature of optimization
  - Determine system parameters from data based on some prescribed objectives which are functions of observed data and parameters
  - Define objective functions which can be linear or nonlinear with single or multiple objectives
  - Design optimization algorithms which can be deterministic or stochastic in nature
  - Solve with either global or local optimality
- Why optimization?
  - Real-world data do not always follow assumptions
  - Solutions need to observe some optimal properties

Example 1: Engineering Design Problem

- Consider lighting a large area with a number of lamps:
- Each lamp has a total power limit
- Several points in the room have a 'desired illumination level'
- *How much power should be applied to each lamp to get the room as close as possible to desired level?*

Example 2: Inventory Levels

- A wholesale Bicycle Distributor:
  - Purchases bikes from manufacturer and supplies to many shops
  - Demand to each shop is uncertain
  - *How many bikes should the distributor order from the manufacturer?*
- Costs:
  - Ordering cost to manufacturer
  - Holding cost in factory
  - Shortage cost due to lack of sales

Example 3: Network Flow

- A telecom service provider:
  - Routing calls through existing networks
  - Demanding on call distribution is uncertain
  - Desiring good overall network performance
  - *How many calls should be distributes to which part of the network?*
- Costs:
  - Minimum time for each call or groups of calls
  - Maximal flow for each call or groups of calls
  - Overall capacity and QoS are two major constraints

Optimization Topics

- Root finding
- Curve fitting and regression
- Linear programming
- Nonlinear programming
- Heuristic methods
- Integer programming
- Dynamic programming
- Inventory theory

Types of Optimization Problems

- Linear: linear functions for objective and constraints
- Nonlinear: nonlinear functions…
- Convex
- Integer
- Mixed-Integer
- Combinatorial
- Unconstrained: no constraints
- Dynamic: solved in stages

Modeling and Optimization Stages

- Define problem and gather data
  - Feasibility check
- Formulate mathematical model
- Develop computer-based method for finding optimal solution
  - Design and Software implementation
- Test and refine model
  - Validation
- Prepare for ongoing model utilization
  - Training, installation
- Implement
  - Maintenance, updates, reviews, documentation, dissemination

Root Solving: Nonlinear Equations

Given $g(V)=I$

It can be expressed as: $f(V)=g(V)-I$

$\Rightarrow$ Solve $g(V)=I$ equivalent to solve $f(V)=0$

Hard to find analytical solution for $f(x)=0$

Solve iteratively

Root Solving: Iterative Method

- Start from an initial value x^0
- Generate a sequence of iterate x^{n-1}, x^n, x^{n+1} which hopefully converges to the solution x^*
- Iterates are generated according to an iteration function F: $x^{n+1}=F(x^n)$

Ask

- When does it converge to correct solution ?
  - What is the convergence rate ?

Newton-Raphson (NR) Method

- Consisting of linearizing the system
  - Want to solve *f(x)=0* → Replace *f*(*x*) with its linearized version and solve

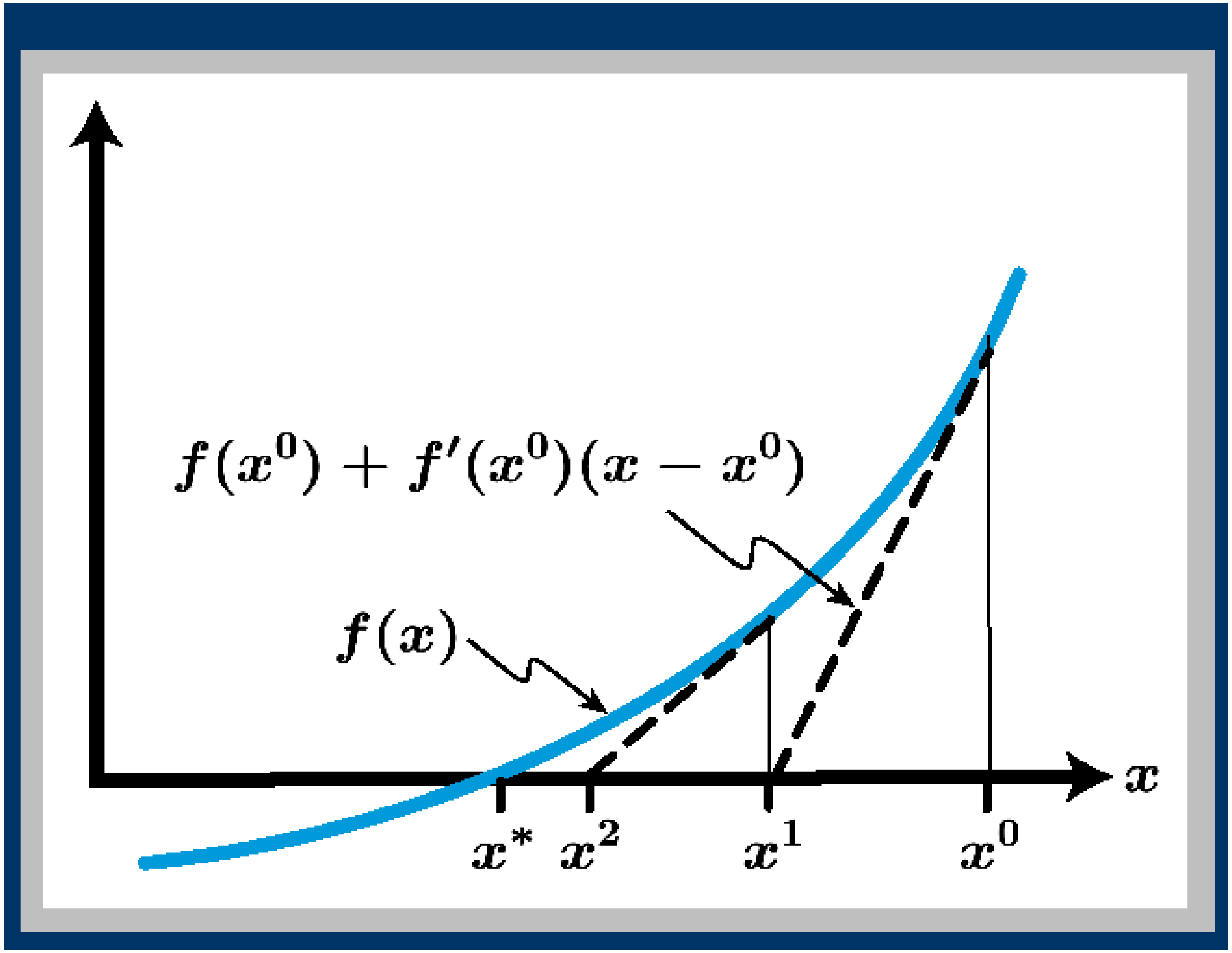
$$f(x) = f(x^*) + \frac{df}{dx}(x^*)(x - x^*) \qquad \textit{Taylor Series}$$

$$f(x^{k+1}) = f(x^k) + \frac{df}{dx}(x^k)(x^{k+1} - x^k)$$

$$\Rightarrow x^{k+1} = x^k - \left[\frac{df}{dx}(x^k)\right]^{-1} f(x^k) \qquad \textit{Iteration function}$$

- Note: at each step need to evaluate *f* and *f'*

Newton-Raphson Method - Graphics



Curve Fitting

- Fit *y=r(x)* to a set of pairs of random samples:

  $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$

  – We will have curve fitting errors: d_i

  – *r(.)* is a regression function

  – Goodness of fit: minimizing least squared errors

$$d_i = y_i - r(x_i) \qquad r(x) = \sum_{k=0}^{p} a_k x^k \qquad D = \sum_{t=1}^{n} d_i^2$$

- Polynomial fitting (MATLAB example):
- Linear fitting: *y=r(x)=a+bx*
- Spline (cubic) fitting
  – Local and global optimization
  – Various optimization criteria

Linear Regression

- Least squares: minimizing sum of squared error

$$D = \sum_{t=1}^{n} d_i^2 = \sum_{t=1}^{n} [y_i - (a + bx_i)]^2 = \text{minimum}$$

- We obtain the following matrix normal equation

$$\frac{\partial D}{\partial a} = 0 \Rightarrow \sum_{t=1}^{n} y_i = an + b\sum_{t=1}^{n} x_i, \quad \frac{\partial D}{\partial b} = 0 \Rightarrow \sum_{t=1}^{n} x_i y_i = a\sum_{t=1}^{n} x_i + b\sum_{t=1}^{n} x_i^2$$

- Solving for intercept *a* and slope *b* : y=polyfit(y,x,n)

$$\hat{b} = \frac{n\sum_{t=1}^{n} x_i y_i - (\sum_{t=1}^{n} x_i)(\sum_{t=1}^{n} y_i)}{n\sum_{t=1}^{n} x_i^2 - (\sum_{t=1}^{n} x_i)^2}, \hat{a} = \frac{(\sum_{t=1}^{n} y_i)(\sum_{t=1}^{n} x_i^2) - (\sum_{t=1}^{n} x_i)(\sum_{t=1}^{n} x_i y_i)}{n\sum_{t=1}^{n} x_i^2 - (\sum_{t=1}^{n} x_i)^2} = \frac{\sum_{t=1}^{n} y_i - \hat{b}\sum_{t=1}^{n} x_i}{n} = \hat{\bar{Y}} - \hat{b}\hat{\bar{X}}$$

- Extend to more than one regressor (econometrics)

Optimization Overview

- Variables: $x = (x_1, x_2, \ldots, x_N)$
- Objective: $\min f(x)$
- Subject to Constraints: $\begin{cases} c_i(x) = 0, i \in \mathrm{E} \\ c_i(x) \geq 0, i \in \mathrm{I} \end{cases}$
- Sometimes additional constraints:
  - Binary
  - Integer
- Sometimes *uncertainty* in parameters (stochastic optimization)

Center of Signal and Image Processing
Georgia Institute of Technology
CSIP

Transforming the Objective Function

- In many instances it is easier to work with transformations of a function – i.e., logarithmic transformation of Cobb-Douglas

- Under what conditions do solutions to original and transformed optimization problems correspond?

Theorem: Let $\varphi: R \rightarrow R$ be a strictly increasing function, that is, a function such that

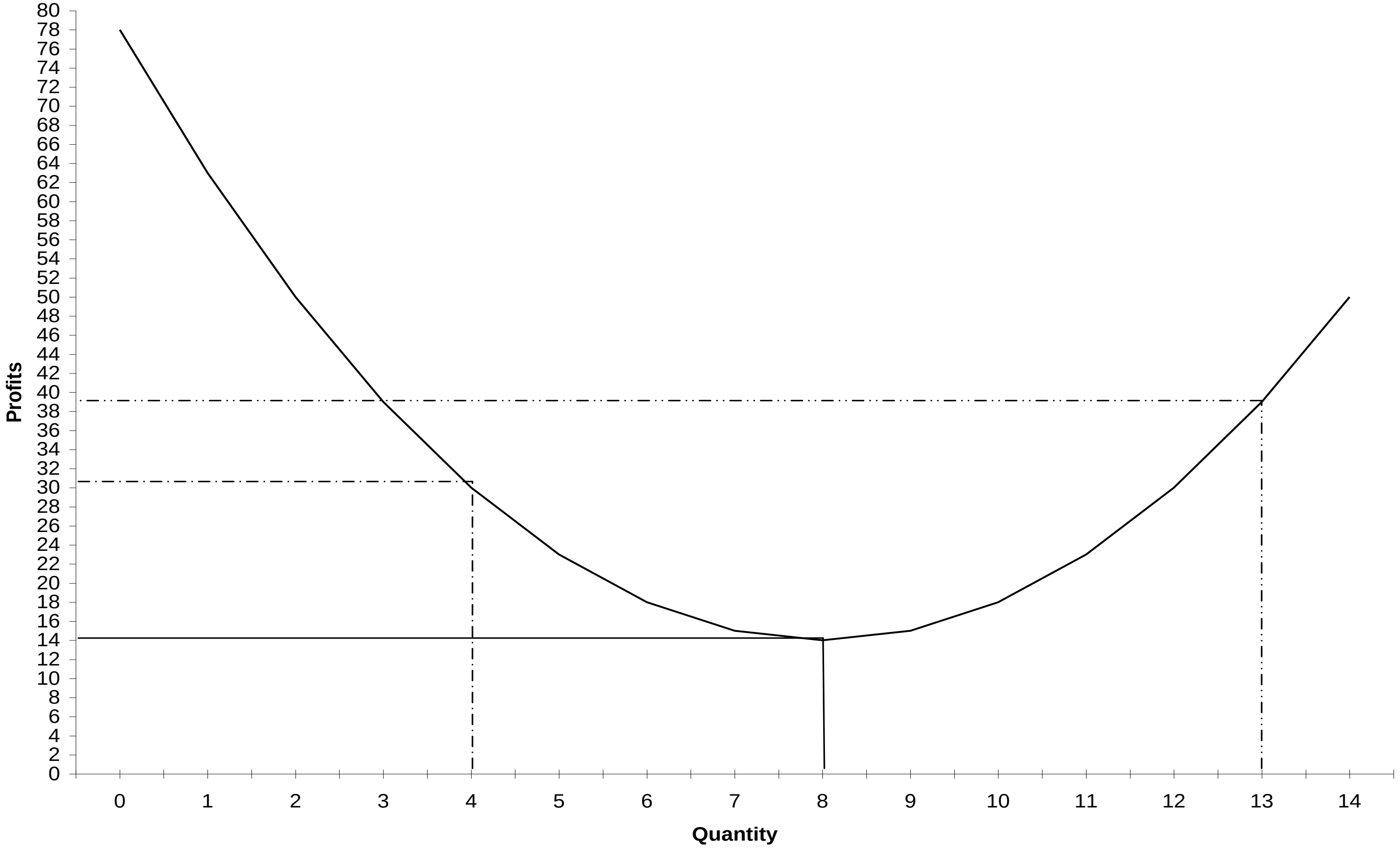
$$x > y \text{ implies that } \varphi(x) > \varphi(y)$$

Then x is a maximum of f on $\boldsymbol{S}$ if and only if x is also a maximum of the composition $\varphi \circ f$ on $\boldsymbol{S}$.

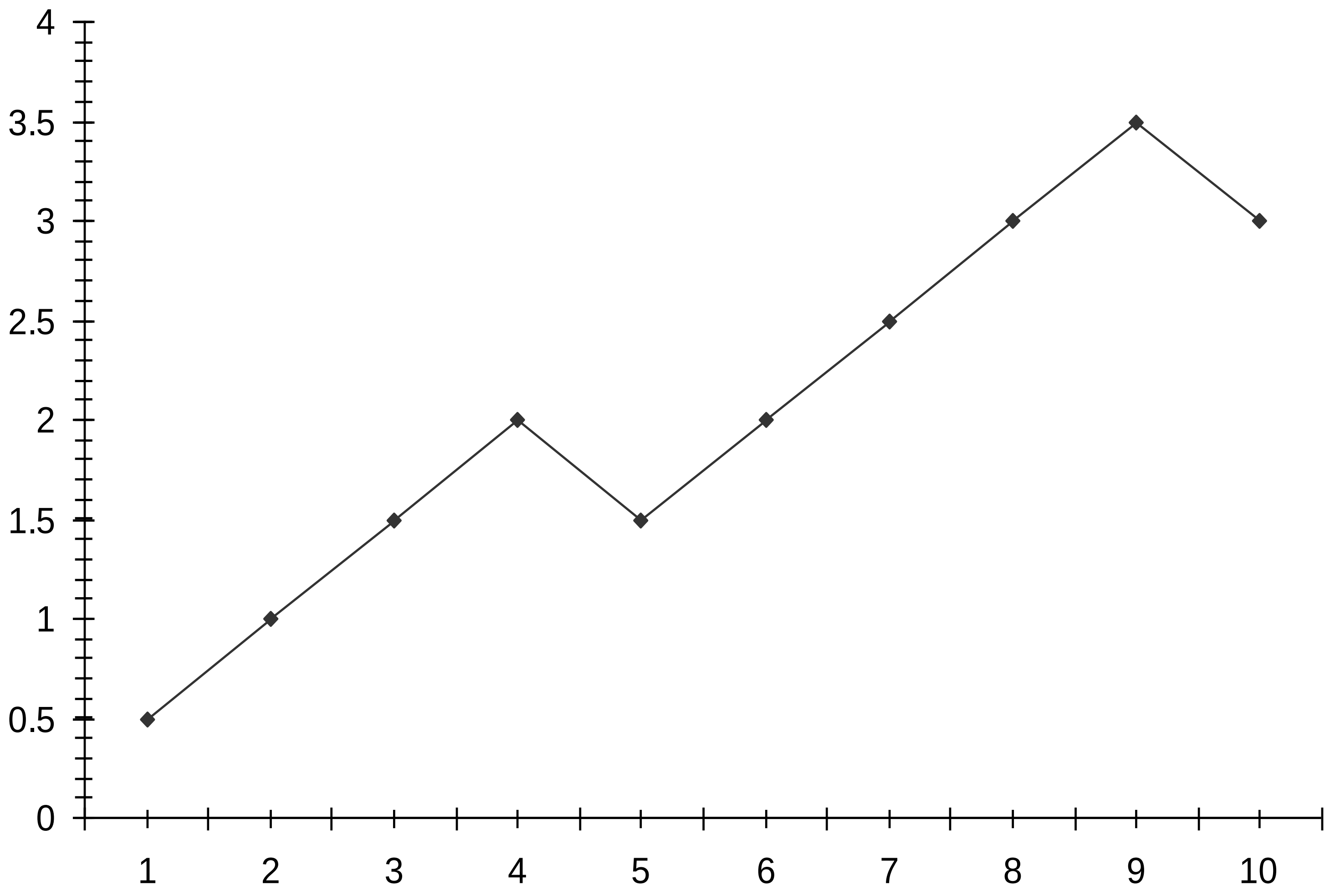
Existence of an Optimum

- Under what conditions on the objective function and the constraint set are we *guaranteed* that solutions will always exist ?
- Trivial conditions can always be introduced that guarantee existence – i.e., a finite constraint set – but we want general conditions
- Weierstrass Theorem describes such a set of conditions
  – Constraint set is compact
  – Objective function is continuous on the constraint set
- Conditions of Weierstrass Theorem are sufficient so there are situations where conditions are violated but optima exist

Global Minimum – Convex Function



Local Extrema – Higher Order Polynomial



Unconstrained Optima: First-Order Conditions

Theorem: Suppose $x^* \in \text{int}\, S \subset \Re^n$ is a local maximum of a differentiable function f on $\boldsymbol{S}$. Then $Df(x^*) = 0$.

- Intuition – Single Variable Case

  Unable to increase the value of the objective function by moving a small amount from x* in either direction

Note: Optima correspond to stationary points of the objective function. However not all stationary points are in fact optima.

Unconstrained Optima: Second-Order Conditions

Proposition (Second-order conditions for optimum of a function)

Let f be a function of n variables with continuous partial derivatives of first and second order, defined on the set S. Suppose that x^* is a stationary point of f in the interior of S (so that $f_i'(x^*) = 0$ for all i).

- If $H(x^*)$ is negative definite then x^* is a local maximizer.
- If x^* is a local maximizer then $H(x^*)$ is negative semidefinite.
- If $H(x^*)$ is positive definite then x^* is a local minimizer.
- If x^* is a local minimizer then $H(x^*)$ is positive semidefinite.

Second Order Conditions – Curvature of the Objective Function

- Strict concavity of the objective function is sufficient to ensure that any x^* yielding maximum value

- Conversely, strict convexity of the objective function is sufficient to ensure that any x^* yielding minimum value

- How do we test the curvature properties of a function?
  - Second-derivatives and second order conditions
  - Hessian matrix vs. gradient vector

Local versus Global Optima

- Important distinction in the second-order conditions for global and local optima
  - Definiteness of Hessian is evaluated at a given point for local optima
  - Definiteness of Hessian must hold for all values of x for a global optima

To state briefly the results for maximizers together:

Sufficient conditions for local maximizer: if x^* is a stationary point of f and the Hessian of f is **negative definite at x^*** then x^* is a **local** maximizer of f

Sufficient conditions for global maximizer: if x^* is a stationary point of f and the Hessian of f is **negative semidefinite for all values of x** then x^* is a **global** maximizer of f.

Constrained Optimization

- So far we have examined case where set of feasible choices is unlimited
  - Agents have unlimited income
  - No scarcity of resources
  - No regulatory constraints on actions

- In many real world applications, the set of feasible choices is constrained
  - Agents have a finite budget set to spend on purchases
  - Factors of production are finite and scarce
  - Regulatory agencies limit the use of certain inputs

Intuition: The Theorem of Lagrange

- Consider the following maximization problem

$$L(x;\lambda) = f(x) + \sum \lambda_i g_i(x)$$

- Intuitively we want to find a stationary point of this objective function
  - Unable to increase the value of the objective function by changing any *x* by a small amount without violating one of the constraints

- However, stationarity at a point is only a necessary condition for a local optima

Optimization – Functions of Multiple Variables

- Consider a function $g(x_1, x_2)$ that depends upon two variables – x_1 and x_2
- How do we solve for a vector (x_1^*, x_2^*) that maximizes this objective function?
- Tools of optimization
  - Extend analysis to consider system of first-order conditions
  - System of equations obtained by taking partial derivatives of $g(\cdot)$ with respect to its arguments – x_1 and x_2
  - Simultaneously solve this system of equations to derive optimal choice

Optimization – Functions of Multiple Variables

- Consider the following optimization problem:

$$\max_{x_1, x_2} g(x_1, x_2)$$

- To solve such a problem, take the partial derivative of $g(x_1, x_2)$ with respect to both x_1 and x_2 and set these partials equal to zero

- Want to find a point where *ceteris paribus* an incremental change in x_1 does not alter the value of the objective function

- Generates a system of two equations in two unknowns – x_1 and x_2

Linear Programming: Problem Definition

Maximize: $c_1x_1 + c_2x_2 + \ldots + c_dx_d$

Subject to the conditions:

$$a_{1,1}x_1 + \ldots\ a_{1,d}x_d \leq b_1$$
$$a_{2,1}x_1 + \ldots\ a_{2,d}x_d \leq b_2$$
$$\vdots \qquad \vdots \qquad \vdots$$
$$a_{n,1}x_1 + \ldots\ a_{n,d}x_d \leq b_n$$

Linear program of dimension d:

$$\vec{c} = (c_1, c_2, \ldots, c_d)$$
$$h_i = \{(x_1, \ldots, x_d)\ ;\ a_{i,1}x_1 + \ldots + a_{i,d}x_d \leq b_i\}$$

l_i = hyperplane that bounds h_i (straight lines, if d=2)

$$H = \{h_1, \ldots, h_n\}$$

Convex Programming

$$\text{Min } f(x_1,\ldots,x_n)$$

$$\text{s.t. } g_i(x_1,\ldots,x_n) \le b_i \quad i = 1,\ldots,m$$

$$x_1 \ge 0,\ldots,x_n \ge 0$$

is a convex program if f is convex and each g_i is convex

$$\text{Max } f(x_1,\ldots,x_n)$$

$$\text{s.t. } g_i(x_1,\ldots,x_n) \le b_i \quad i = 1,\ldots,m$$

$$x_1 \ge 0,\ldots,x_n \ge 0$$

is a convex program if f is concave and each g_i is convex

Dynamic Programming

- Break the main problem in sub-problems
- Express the optimum solution of the main problem in terms of those of the sub-problems
- Solve the sub-problems recursively
- Combine the solutions of the subproblems to solve the main problem

Bellman's Principle of Optimality

- The global problem is solved optimally only if all sub-problems are solved optimally
- Holds for shortest path problem
  – Any segment of a shortest path is a shortest path between the corresponding source and destination
- May not always hold

Multi-layer Feed-forward Neural Networks

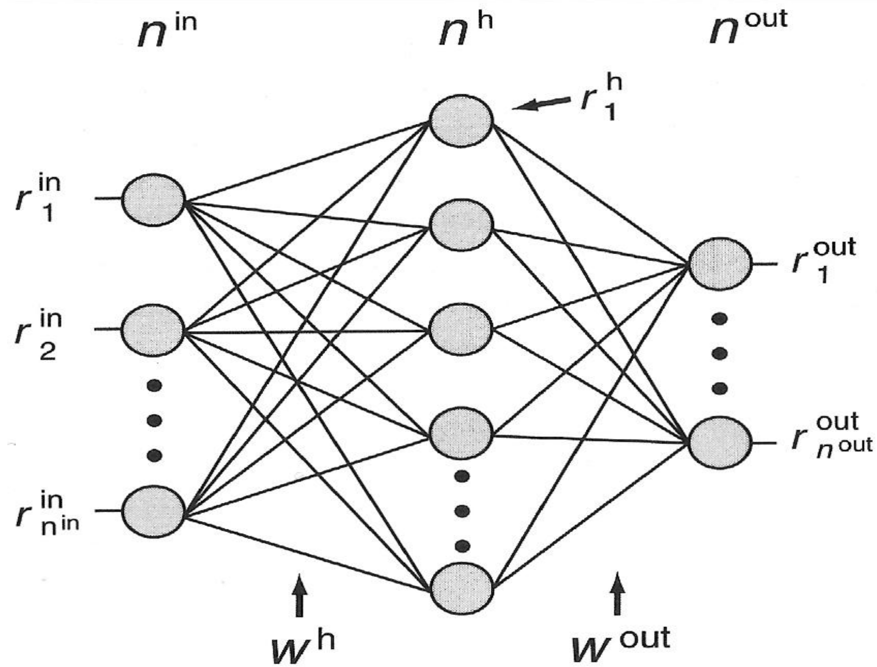


Fig. 6.6 The standard architecture of a feed-forward multilayer network with one hidden layer, in which input values are distributed to all hidden nodes with weighting factors summarized in the weight matrix $\mathbf{w}^{\mathrm{h}}$. The output values of the nodes of the hidden layer are passed to the output layer, again scaled by the values of the connection strength as specified by the elements in the weight matrix $\mathbf{w}^{\mathrm{out}}$. The parameters shown at the top, n^{in}, n^{h}, and n^{out}, specify the number of nodes in each layer, respectively.

Summary

- Today's Class
  - Optimization essentials (for self-study only)
  - Web: http://www.ece.gatech.edu/~chl/ECE6255.sp10
- Next Classes
  - DSP Fundamentals on Jan. 15-20
  - Speech Acoustics on Jan. 22-29
- Reading Assignments
  - Quatieri, Chapters 1 & 2
  - Rabiner and Schafer, Chapters 1 & 2