Artificial Neural Networks (Neural Nets)

- **Nodes(neurons) are computational elements that are nonlinear and typically realized by analog circuits.**

- **Links are connections with weights representing the strength of connections.**

Key Factors for Specification

- **Net topology**
  - **feedforward type vs. feedback type**
  - **number of layers**
  - **number of nodes in each layer**

- **Node (neuron) type**
  - **different nonlinearities realized by analog circuits vs. complex mathematical relations realized by digital circuits**
  - **nodes operate continuously vs. discretely**

- **Weight specification**
  - **predetermined vs. adapted**

Artificial Neuron

$f(x) = \Phi(\sigma(x) - T)$ for $x \in C \subseteq R^n$,

where

$f(\cdot) : R^n \longrightarrow R$ neuron function

$\sigma(\cdot) : R^n \longrightarrow R$ accumulation function

$T \in R$ threshold

$\Phi(\cdot) : R \longrightarrow R$ (nonlinear) activation function

Examples:

- Sigmoidal neuron

$$y_j = \Phi_b(\sum_{i=1}^{n} w_{ij} x_i + T), \quad j = 1, \cdots, m$$

$$\Phi_b(u) = \tanh(\lambda u) = \frac{1 - e^{-2\lambda u}}{1 + e^{-2\lambda u}}$$

- **McCulloch-Pitts neuron**

$$
\begin{aligned}
y_j &= \mathrm{sgn}\left(\sum_{i=1}^{n} w_{ij} x_i - b\right) \quad j = 1, \cdots, m \\
&= \begin{cases} 1 & \text{if} \quad \sum_{i=1}^{n} w_{ij} x_i \geq b \\ -1 & \text{if} \quad \sum_{i=1}^{n} w_{ij} x_i < b \end{cases}
\end{aligned}
$$

- **Integrator neuron /Time-dependent neuron**

$$y(t) = \frac{x(t) - x(t - \Delta t)}{\Delta t}$$

as $\Delta t \longrightarrow 0$

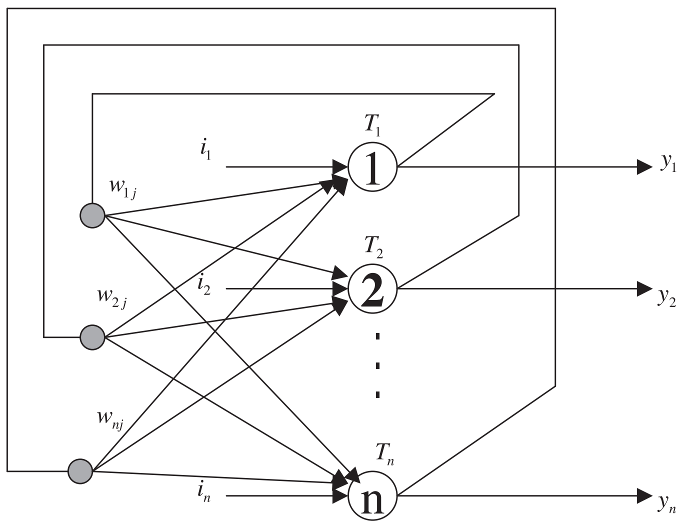
$$y(t) \longrightarrow \dot{x}(t)$$

Feedback Neural Networks

- **Single-layer feedback network**
  (Discrete Hopfield network)
- **Two-layer feedback network**
  (Continuous Grossberg model)

Single-layer Feedback Network

General architecture $N(W, T)$ of discrete Hopfield feedback network



$$\boldsymbol{y}(t+1) = \boldsymbol{f}(\boldsymbol{y}(t)) = \mathrm{sgn}(\boldsymbol{W}\boldsymbol{y}(t) - \boldsymbol{T})$$

$$= \mathrm{sgn}(\ \boldsymbol{H}(t)\ )$$

where

$$\mathrm{sgn}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$$

- **Finite number of states**
- **System dynamics**

$$\dot{\boldsymbol{y}}(t) = \boldsymbol{h}(\boldsymbol{y}(t))$$

Synchronous Hopfield Network

begin

Set $\quad t := 0$, **with a given initial vector** $y(0)$ **and a given stopping rule** $\mathcal{C}$**.**

step

$\quad y_i(t+1) = \mathrm{sgn}(W_i y(t) - T_i), \quad i = 1, \cdots, n$

$\quad$ **if** $y(t+1)$ **satisfies** $\mathcal{C}$, **stop!**

$\quad$ **else** $t := t+1$ **go to step**

end

Asynchronous Hopfield Network

begin

Set $t := 0$, with a given initial vector $y(0)$ and a given stopping rule $\mathcal{C}$.

step

Pick $i(t) \in \{1, \cdots, n\}$ according to some given rule.

$y_{i(t)}(t+1) = \mathrm{sgn}(W_{i(t)} y(t) - T_{i(t)}),$

$y_{i(t)}(t+1) = y_i(t), \quad \forall i \neq i(t).$

If $y(t+1)$ satisfies $\mathcal{C}$, stop!

else $t := t+1$ go to step.

end

Example

$N = (W, 0)$ with

$$W = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$$

(1) Synchronous mode

(2) Asynchronous

Initial		End
$\begin{pmatrix} -1 \\ 1 \end{pmatrix}$	$\longrightarrow$	
$\begin{pmatrix} 1 \\ -1 \end{pmatrix}$	$\longrightarrow$	
$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$\longrightarrow$	
$\begin{pmatrix} -1 \\ -1 \end{pmatrix}$	$\longrightarrow$	

Definition

1. N is convergent from a given initial state $\boldsymbol{y}(0)$ if $\exists$ integer $\boldsymbol{k} > 0$

$$\boldsymbol{s.t.} \quad \boldsymbol{y}(\boldsymbol{k}) = \operatorname{sgn}(\boldsymbol{W}\boldsymbol{y}(\boldsymbol{k}) - \boldsymbol{T})$$

2. $\overline{\boldsymbol{y}} = \boldsymbol{y}(\boldsymbol{k})$ is a stable state.

3. N is convergent if it converges from any given initial state.

4. N is convergent to a stable cycle of length $\boldsymbol{q}$ from a given initial state $\boldsymbol{y}(0)$ if $\exists$ integers $\boldsymbol{k}, \boldsymbol{q} > 0$

$$\boldsymbol{s.t} \quad \boldsymbol{y}(\boldsymbol{k}+\boldsymbol{q}) = \operatorname{sgn}(\boldsymbol{W}\boldsymbol{y}(\boldsymbol{k}) - \boldsymbol{T})$$

Known Result

Thm1. **If $\boldsymbol{W}$ is symmetric with non-negative diagonal elements, then $\boldsymbol{N(W, T)}$ always converges to a stable state in the asynchronous mode.**

<u>Proof of Thm1:</u>

Define Energy function

$$\boldsymbol{E(t) = -\frac{1}{2}y(t)^T W y(t) + y^T(t) T}$$

$$\boldsymbol{\Delta E(t) = E(t+1) - E(t)}$$

$$\boldsymbol{\Delta y_i(t) = y_i(t+1) - y_i(t)}$$

<u>Key idea:</u> Note that $\boldsymbol{N(W,T)}$ has a finite number of states.

$$\textbf{If} \quad \boldsymbol{\Delta E(t) \le 0 \ (\text{or } \Delta E(t) \ge 0) \qquad \forall t}$$

$$\textbf{then} \quad \boldsymbol{\Delta E(t) = 0 \text{ for } t \ge t_0 \ (\text{some } t_0 > 0)}$$

Since we run in the asynchronous mode,

$$\begin{aligned} \boldsymbol{\Delta E(t)} &= \boldsymbol{-\tfrac{1}{2}\Delta y_i(t) \left[\sum_{k=1}^{n} w_{ik} y_k(t) + \sum_{k=1}^{n} w_{ki} y_k(t) \right]} \\ &\quad \boldsymbol{-\tfrac{1}{2} w_{ii} \Delta y_i(t)^2 + \Delta y_i(t) T_i} \\ &= \boldsymbol{-\Delta y_i(t) H_i(t) - \tfrac{1}{2} w_{ii} \Delta y_i(t)^2} \end{aligned}$$

Moreover

$$\boldsymbol{\Delta y_i(t)} = \begin{cases} \mathbf{0} & \textbf{if} \;\; \boldsymbol{y_i(t) = \mathrm{sgn}(H_i(t))} \\ \mathbf{-2} & \textbf{if} \;\; \boldsymbol{y_i(t) = 1} \;\; \textbf{and} \\ & \quad \boldsymbol{y_i(t+1) = \mathrm{sgn}(H_i(t)) = -1} \\ \mathbf{2} & \textbf{if} \;\; \boldsymbol{y_i(t) = -1} \;\; \textbf{and} \\ & \quad \boldsymbol{y_i(t+1) = \mathrm{sgn}(H_i(t)) = 1} \end{cases}$$

$\Rightarrow \boldsymbol{\Delta y_i(t) H_i(t) \geq 0}$

$\Rightarrow \boldsymbol{\Delta E(t) \leq 0}$ **(because** $\boldsymbol{w_{ii} \geq 0}$**)**

$\Rightarrow \boldsymbol{\Delta E(t) = 0}$ **for** $\boldsymbol{t > t_0}$ **(some** $\boldsymbol{t_0 > 0}$**)**

$\Rightarrow \boldsymbol{\Delta y(t) = 0}$ **for** $\boldsymbol{t > t_0}$

$\Rightarrow \boldsymbol{N(W, T)}$ **converges to a stable state.**

NN Model for Discrete Optimization

Let W be a symmetric real matrix with non-negative diagonal elements and $T \in R^n$, consider the discrete quadratic optimization problem:

$$\begin{aligned} \text{Minimize} \quad & -\frac{1}{2} y^T W y + y^T T \\ s.t. \quad & y \in \{1, -1\}^n \end{aligned}$$

We can build an asynchronous Hopfield Network with system dynamics

$$\begin{aligned} y(t+1) &= f(y(t)) \\ &= \operatorname{sgn}(W y(t) - T) \end{aligned}$$

to solve the problem.

<u>Known Result</u>

Thm2. **If W is symmetric, then $N(W, T)$ converges in the synchronous mode to a stable state of a cycle of length 2. If W is also positive semi-definite, then $N(W, T)$ converges to a stable state.**

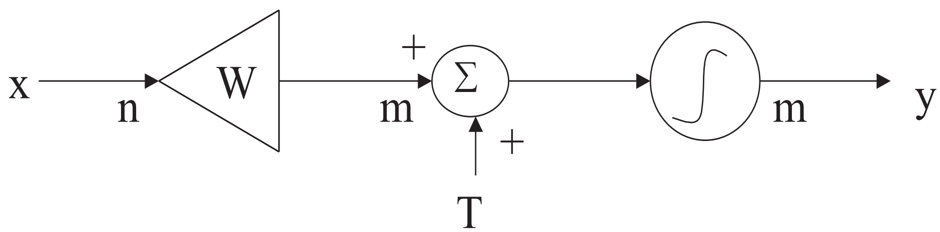
<u>Proof of Thm2</u>

$$E(t) = \frac{1}{2}y(t)^T W y(t-1) - \frac{1}{2}[y(t) + y(t-1)]^T T$$

<u>Question:</u> What kind of discrete optimization problems can be solved by the synchronous of Hopfield network?

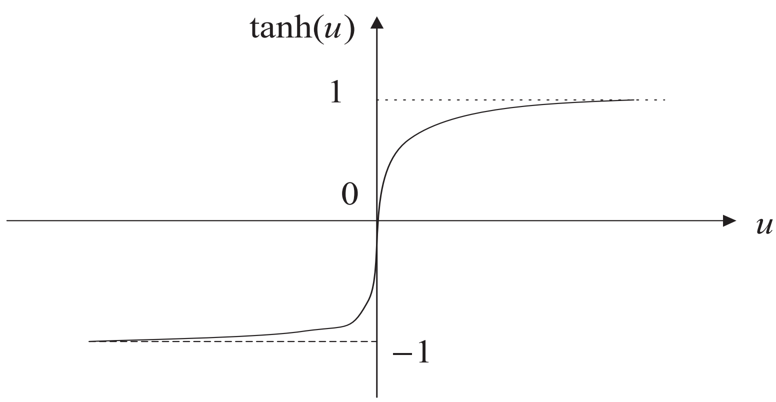
Neurons for Different Variables

(A) Sigmoidal neuron



$$y_j = \Phi(\sum_{i=1}^{n} w_{ij}x_i + T) \quad \forall\ j = 1, 2, \cdots, m$$

- $\tanh(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}} = \frac{1-e^{-2u}}{1+e^{-2u}}$



$$\tanh'(u) = 1 - \tanh^2(u)$$
$$\tanh''(u) = -2\tanh(u)\tanh'(u)$$

- $\Phi_b(u) \equiv \tanh(\lambda u) = \frac{1-e^{-2\lambda u}}{1+e^{-2\lambda u}}$

$$\Phi_b'(u) = \lambda(1-\tanh^2(\lambda u))$$
$$\Phi_b''(u) = -2\lambda\Phi_b(u)\Phi_b'(u)$$

as $\lambda \to \infty$

$$\Phi_b(u) \to \begin{cases} 1 & \text{for} \quad u > 0 \\ 0 & \text{for} \quad u = 0 \\ -1 & \text{for} \quad u < 0 \end{cases}$$

Hence

$$\Phi_b(u) \to \mathrm{sgn}(u) \quad \text{for} \;\; u \neq 0 \text{ as } \;\lambda \to \infty$$

- $\Phi_p(u) \equiv \frac{1}{2}(1 + \tanh(\lambda u)) = \frac{1}{1+e^{-2\lambda u}}$

$$\Phi_p'(u) = 2\lambda\Phi_b(u)(1 - \Phi_b(u))$$
$$\Phi_p''(u) = 2\lambda\Phi_b'(u)(1 - \Phi_b(u)) + 2\lambda\Phi_b(u)\Phi_b'(u)$$

as $\lambda \longrightarrow \infty$

$$\Phi_p(u) \longrightarrow \begin{cases} 1 & \text{for} \quad u > 0 \\ \frac{1}{2} & \text{for} \quad u = 0 \\ 0 & \text{for} \quad u < 0 \end{cases}$$

- **general sigmoidal function**

$$\Phi_g(u) \equiv \frac{\alpha + \beta e^{-\lambda u}}{\gamma + \zeta e^{-\lambda u}}$$

where

$$\alpha,\ \beta,\ \gamma,\ \zeta\ \in R$$

$\gamma,\ \zeta$ **not zero at the same time**

$$\lambda > 0$$

- **Serve as barrier function for constraints**

$$y \equiv \Phi_g(u) = \frac{\alpha + \beta e^{-\lambda u}}{\gamma + \zeta e^{-\lambda u}} \quad u \in R$$

For $\gamma = \zeta = 1,\ \beta \le \alpha$, **then**

$$\beta \le y \le \alpha$$

For $\gamma = \zeta = \alpha = 1,\ \beta = 0$, **then**

$$0 \le y \le 1$$

For $\gamma = \zeta = \alpha = 1,\ \beta = -1$, **then**

$$-1 \le y \le 1$$

For $\gamma = 0;\ \zeta = 1,\ \alpha > 0$, then

$$y \geq \beta$$

For $\alpha = \zeta = 1,\ \beta = \gamma = 0$, then

$$y \geq 0$$

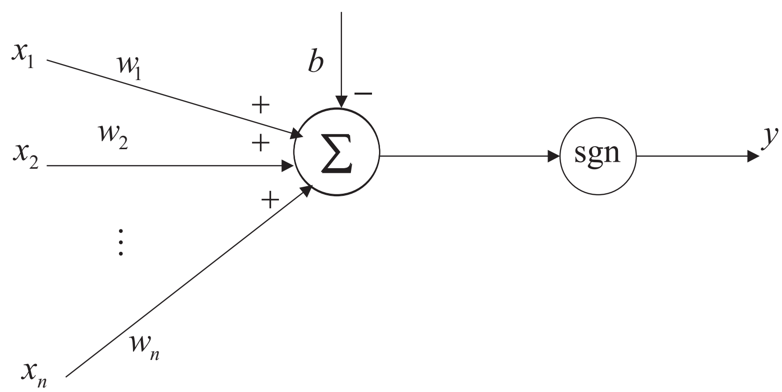
- **Applications:**

$$\begin{array}{rl} \text{minimize} & -\frac{1}{2}y^T W y + y^T T \\ s.t. & \alpha \leq y \leq \beta \end{array} \qquad (P)$$

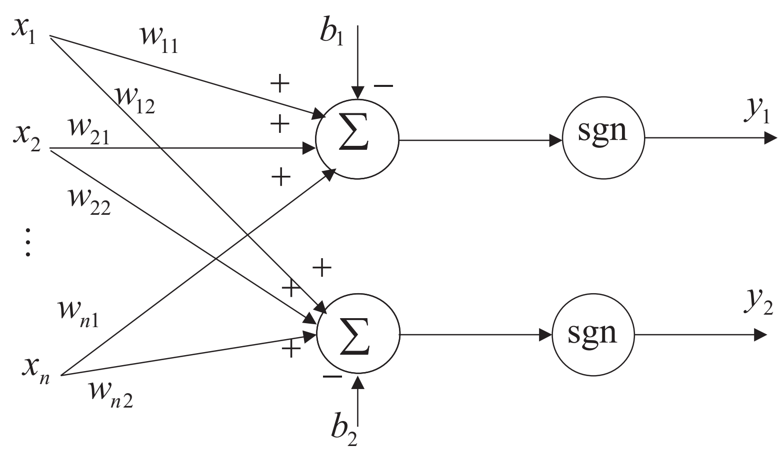
Asynchronous Hopfield network with dynamics

$$\begin{aligned} y(t+1) &= f(y(t)) \\ &= \Phi_g(Wy(t) - T) \end{aligned}$$

(B) McCulloch-Pitts neuron (Linear Classifier)



$$\boldsymbol{y = \mathrm{sgn}(\sum_{j=1}^{n} w_j x_j - b) = \begin{cases} 1 & \text{if } \sum_{j=1}^{n} w_j x_j > b \\ -1 & \text{if } \sum_{j=1}^{n} w_j x_j < b \end{cases}}$$



$$\boldsymbol{y_1 = \mathrm{sgn}(\sum_{j=1}^{n} w_{j1} x_j - b)}$$

$$\boldsymbol{y_2 = \mathrm{sgn}(\sum_{j=1}^{n} w_{j2} x_j - b)}$$

Hebb's Learning Rule (Donald Hebb 1949)

- based on Pavlov's conditional behaviors
- what do we want?

$$w^t x^p > b \text{ then } y^p = 1 \ \text{ for } x^p \in C_1$$

$$w^t x^q < b \text{ then } y^q = -1 \text{ for } x^q \in C_2$$

- what may go wrong?

$$\text{for } x^p \in C_1 \ (i.e., \ y^p = 1)$$

we have

$$w^t x^p < b$$

or

$$\text{for } x^q \in C_2 \ (i.e., \ y^q = -1)$$

we have

$$w^t x^q \geq b$$

- how to reduce the error ?

$$\Delta w(t) = \lambda y^i x^i \ (\lambda > 0)$$

$$w(t+1) = w(t) + \Delta w(t)$$

(C) Integrator neuron/Time-dependent neuron

- **accumulation function**

$$\boldsymbol{\sigma}(x(t), x(t-\Delta t)) = \frac{x(t)}{\Delta t} - \frac{x(t-\Delta t)}{\Delta t}$$

- **threshold** $T = 0$

- **activation function**

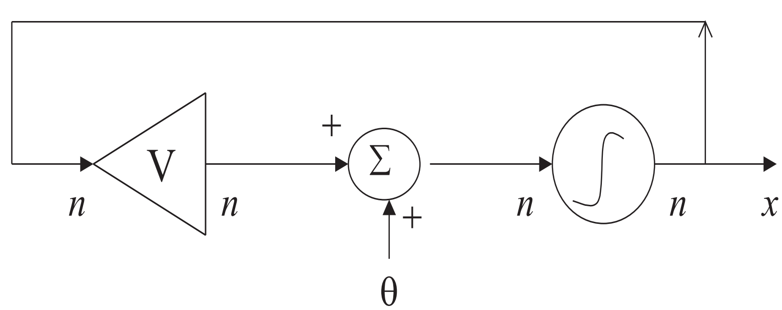
$$\Phi(x) = x$$

- **neuron function**

$$\begin{aligned} f(x(t)) &= \Phi(\boldsymbol{\sigma}(x(t), x(t-\Delta t))) \\ &= \frac{x(t) - x(t-\Delta t)}{\Delta t} \end{aligned}$$

as $\Delta t \longrightarrow 0, \ f(x(t)) \longrightarrow \frac{dx}{dt} = \dot{x}(t)$

<u>**Example**</u>



$$\dot{x}(t) = Vx(t) + \theta$$

Continuous Feedback Network

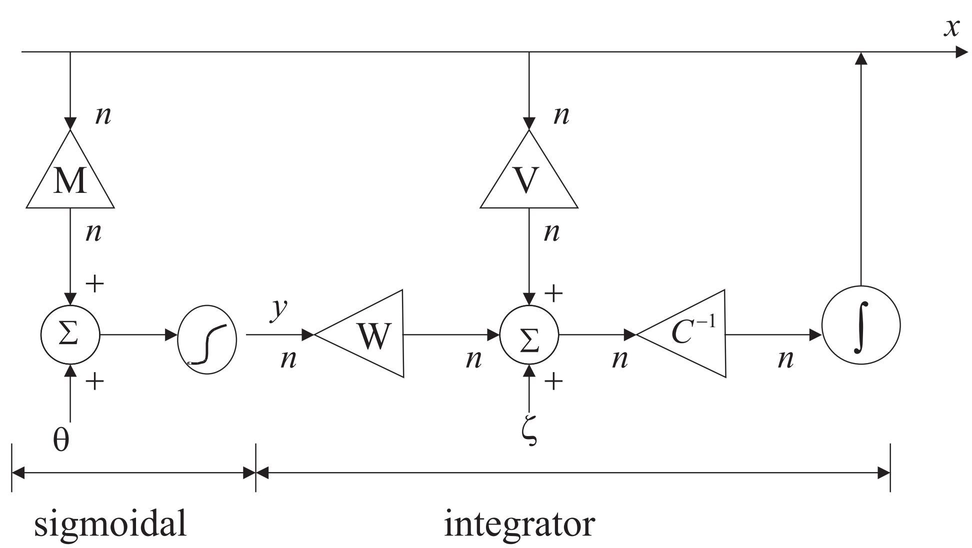
(two-layer feedback network)

- **General model**

  one layer: sigmoidal neuron

  one layer: integrator neuron

- **Network structure (Grossberg model)**



- **System Dynamics**

$$C\dot{x}(t) = Vx(t) + W\Phi(Mx(t) + \theta) + \zeta$$

or

$$\begin{cases} C\dot{x}(t) & = Vx(t) + Wy(t) + \zeta \\ y(t) & = \Phi(Mx(t) + \theta) \end{cases}$$

Componentwise

$$\begin{cases} c_i\dot{x}_i(t) & = -\frac{x_i(t)}{R_i} + \sum\limits_{j=1}^{n} w_{ij}y_j(t) + \sum\limits_{j=1}^{n} v_{ij}x_j(t) + \varsigma_i \\ y_j(t) & = \Phi(\sum\limits_{k=1}^{n} m_{jk}x_k(t) + \theta_j) \\ i,\ j & = 1,\ 2\ \cdots,\ n \end{cases}$$

$$\begin{aligned} C &= \mathrm{diag}(c_1,\ c_2,\ \cdots,\ c_n) \\ V &= \left(v_{ij} - \frac{\delta_{ij}}{R_i}\right)_{n\times n} \\ W &= (w_{ij})_{n\times n} \\ M &= (m_{ij})_{n\times n} \end{aligned}$$

- **Application 1**

$$\begin{array}{ll} \textbf{minimize} & \frac{1}{2}y^TQy + q^Ty \\ s.t. & -\alpha_j \le y_j \le \alpha_j,\ j = 1, \cdots, n \end{array}$$

Consider

$$\begin{cases} \frac{dx_i}{dt} = -\sum_{j=1}^{n} q_{ij}y_j - q_i \\ y_j = \Phi_g(x_j) \triangleq \alpha_j \frac{1-e^{-\lambda x_j}}{1+e^{-\lambda x_j}},\ j = 1, \cdots, n \end{cases}$$

or

$$\begin{cases} \dot{x}(t) = -Qy(t) - q \\ y(t) = \Phi_g(x(t)) \end{cases}$$

- **Application 2**

$$\begin{aligned} \text{minimize} \quad & f(y) \\ s.t. \quad & Dy = b \\ & \beta \le y \le \alpha \end{aligned}$$

where

$$y,\ \alpha,\ \beta\ \in R^n,\ D \in R^{m\times n},\ b \in R^m$$

Define

$$E(y) = f(y) + \|Dy - b\|^2$$

Consider

$$\begin{cases} \frac{dx_i}{dt} = -\frac{\partial E}{\partial y_i} \\ y_i = \Phi_g(x_i) = \frac{\alpha_i + \beta_i e^{-\lambda x_i}}{1 + e^{-\lambda x_i}} \end{cases}$$

or

$$\begin{cases} \dot{x}(t) = -\frac{\partial E(y(t))}{\partial y_i(t)} \\ y(t) = \Phi_g(x(t)) \end{cases}$$

Definition 1: For a given $t_0 \geq 0$ with $x(t_0) = x_0$, a vector x^* is an equilibrium point of the continuous feedback Grossberg network if

$$Vx^* + W\Phi(Mx^* + \theta) + \zeta = 0$$

Definition 2: An equilibrium point x^* is stable, if $\forall\, \epsilon > 0,\ \exists\, \delta > 0$

$$s.t.\ \|x(t_0) - x^*\| < \delta$$
$$\Rightarrow\ \|x(t) - x^*\| < \epsilon, \quad \text{for} \quad t \geq t_0$$

Definition 3: An equilibrium point x^* is asymptotically stable, if it is stable and

$$\lim_{t\to\infty} x(t) = x^*.$$

<u>Question:</u>

When do we have asymptotic stability?

<u>Theorem:</u>

A dynamic system is asymptotically stable, if there exists a Liapunov function associated with the system.

Liapunov Function

A Liapunov function or energy function is a function $E(x(t))$, which satisfies the following conditions.

1. $E(x(t))$ and each partial derivative $\frac{\partial E(x(t))}{\partial x_i(t)}$, $1 \leq i \leq n$, are continuous.

2. $E(x(t))$ is nonnegative, i.e., $E(x(t)) \geq 0$. Especially, $E(\tilde{x}) = 0$ and $E(x(t)) \geq 0$ for $x(t)$ in some neighborhood of the equilibrium point $\tilde{x}$.

3. The derivative of $E(x(t))$ with respective to time t is nonpositive, namely

$$\frac{dE(x(t))}{dt} = [\nabla_{x(t)} E(x(t))]^T \dot{x}(t) < 0$$

Example

- **Lagrange network for convex QP**

$$\begin{array}{ll} \text{minimize} & f(x) = \frac{1}{2}x^T Q x + q^T x \\ \text{s.t.} & Ax = b \end{array}$$

where $A \in R^{m\times n}, Q \in R^{n\times n}, q \in R^n, b \in R^m$.

$$\begin{aligned} L(x;\lambda) &= \frac{1}{2}x^T Q x + q^T x + \lambda^T(Ax - b) \\ \dot{x} &= -Qx - A^T\lambda - q \\ \dot{\lambda} &= Ax - b \end{aligned}$$

- **Diagram**