CSCE 790: Neural Networks and Their Applications

AIISC and Dept. Computer Science and Engineering
Email: vignar@sc.edu

October 10, 2023

UNIVERSITY OF
South Carolina

Hopfield Network

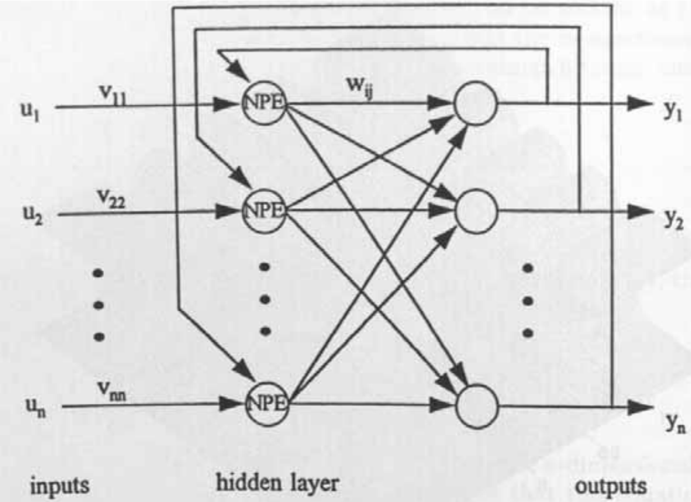

Figure: Hopfield network (Lewis, '99)

Neural Processing Elements (NPE)

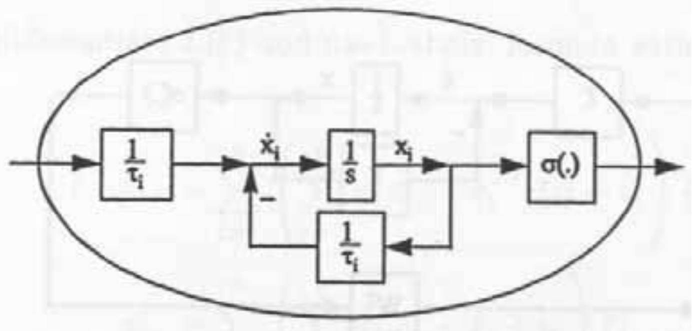

Figure: NPE - Continuous time units (Lewis, '99)

$$y_i(t) = \sum_{j=1}^{n} w_{ij}\sigma_j(x_j)(t), \quad \tau_i\dot{x}_i(t) = -x_i(t) + \sum_{j=1}^{n} w_{ij}\sigma_j(x_j)(t) + v_{ii}u_i$$

DT - Neural Processing Elements (NPE)

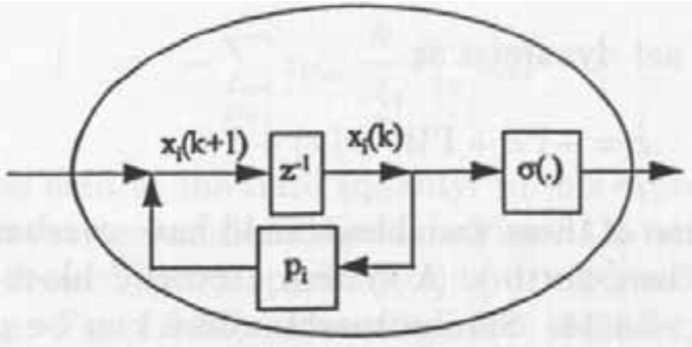

Figure: NPE - Discrete time units (Lewis, '99)

$$y_i(k) = \sum_{j=1}^{n} w_{ij}\sigma_j(x_j)(k), \quad x_i(k+1) = p_i x_i(k) + \sum_{j=1}^{n} w_{ij}\sigma_j(x_j)(k) + v_{ii}u_i$$

Hopfield Networks

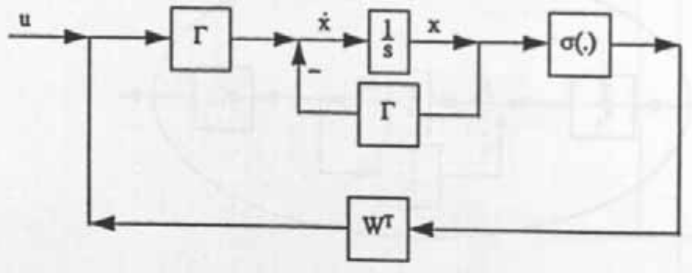

Figure: Hopfield network with NPE (Lewis, '99)

Generalized Recurrent Neural Network

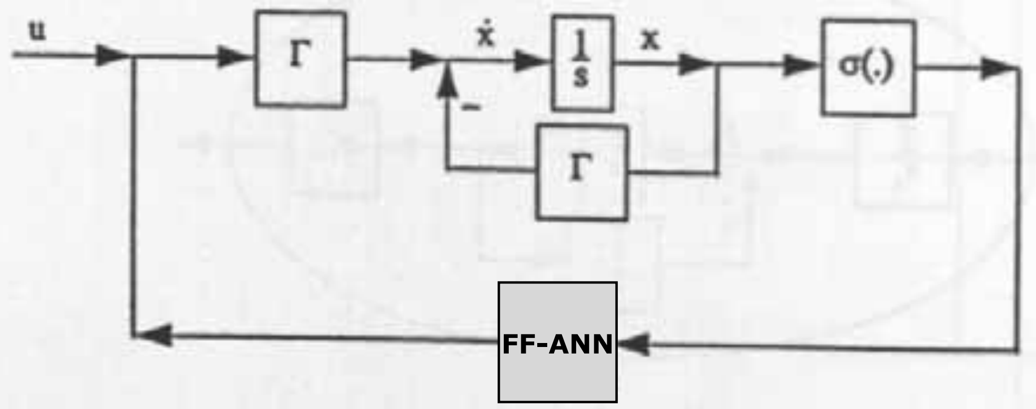

Figure: Generalized recurrent neural networks (Lewis, '99)

Direct Computation of Weights for Hopfield Network

- In the Hopfield net, the weights can be initialized by direct computation of outer products between desired outputs
- Suppose we would like to design a Hopfield network that can classify or discriminate between P given bipolar pattern $\{X^1, X^2, \ldots, X^P\}$ each having n entries of either $+1$ or -1
- Given $x(0)$ as initial condition (input), the Hopfield network should perform association and match the input with one of the P patterns

Hopfield Weight Selection

- Hopfield showed that weights to solve this problem may be selected by using the Hebbian philosophy of learning as the outer product of X^P

$$W = \frac{1}{n}\sum_{p=1}^{P} X^P (X^P)' - \frac{1}{n} PI,$$

- I is the identity matrix
- The purpose of the term PI is to zero out the diagonal
- Note that this weight matrix W is symmetric
- This formula effectively encodes the exemplar patterns in the weights of the NN
- Though there is no weight tuning, technically this formula is an example of supervised learning, as the desired outputs are used to compute the weights

Example

Example

Consider a Hopfield network

$$\dot{x}(t) = -\frac{1}{2}x(t) + \frac{1}{2}W'\sigma(x(t)) + \frac{1}{2}u,$$

with $x(t) \in \mathbb{R}^2$ and a symmetric sigmoid function

$$\sigma(x_i) = \frac{1 - e^{-100x_i}}{1 + e^{-100x_i}}.$$

Suppose the prescribed exemplar patterns are $X^1 = (1, 1)'$ and $X^2 = (-1, -1)'$. $(u = 0)$

Example

Then, according to the 'training' equation, the weight matrix is $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

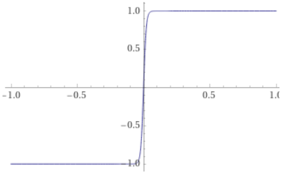

Figure: Symmetric Sigmoid

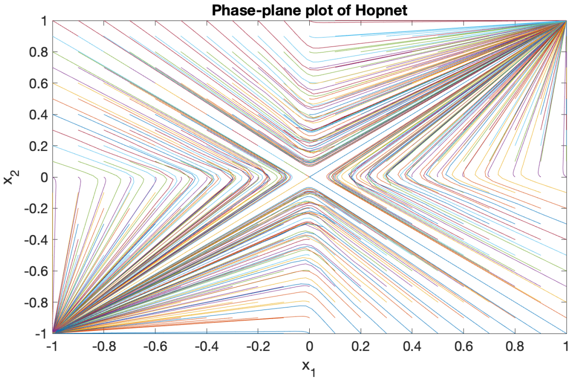

Figure: Trajectories of the Hopfield networks

Backpropagation Through Time (wiki)

- Backpropagation through time (BPTT) for training certain types of recurrent neural networks is an analogue to Backpropagation algorithm for training feedforward neural networks

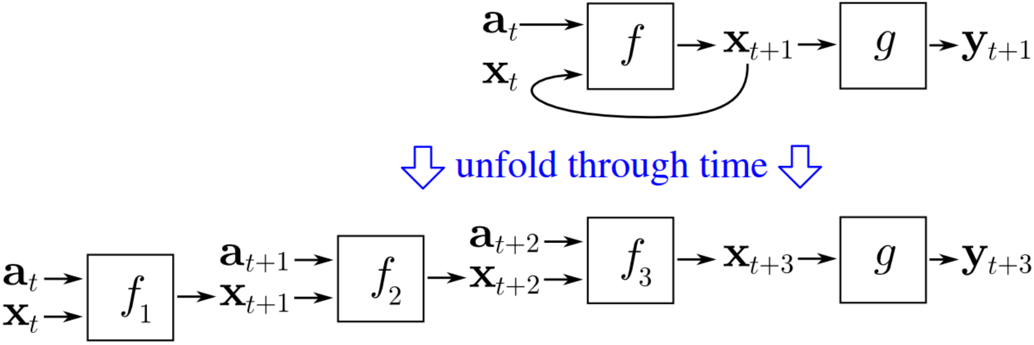

Figure: Backpropagation through time (wiki)

BPTT

- In the example the neural network contains a recurrent layer f and a feedforward layer g
- Training cost can be defined in various way
- Example: Aggregated cost - average of the costs of each time steps
- In the figure the cost at time $t + 3$ is show by unfolding the recurrent layer f for three time steps and adding the feedforward layer g
- Each instance of f in the unfolded network shares the same parameters
- Thus the weight updates in each instance f_1, f_2, f_3 are summed together

NN Application for Control - Learning Paradigm - ”Reinforcement Learning”

Control Application: Lyapunov Techniques for Controller Design

- Recall the problem of Control design:
- Example: Cruise control problem for a toy car model

$$\dot{x}(t) = -\frac{c}{m}u(t), \quad x(0) \in \mathbb{R}^+, \tag{1}$$

  where $x(t)$ is the velocity of the car at time t.
- What happens to this system when a proportional control input $u(t) = Kx(t)$ is selected $(K > 0)$?

Reference Tracking Problem

- Given a reference/desired velocity $r(t)$, what should be the control input so that the car moves with the given velocity?
- Define the error, i.e., the difference between the reference velocity and the actual velocity as

$$e(t) = r(t) - x(t)$$

- Compute how this error changes with time,

$$\dot{e}(t) = \dot{r}(t) - \dot{x}(t) = \dot{r}(t) + \frac{c}{m}u(t),$$

- How to design control input for this case?

Lyapunov Techniques for Controller Design

- Recall the problem of Control design:
- Example: Cruise control problem for a toy car model

$$\dot{x}(t) = -\frac{c}{m}u(t), \quad x(0) \in \mathbb{R}^+, \tag{2}$$

where $x(t)$ is the velocity of the car at time t.

- What happens to this system when a proportional control input $u(t) = Kx(t)$ is selected $(K > 0)$?

Feedback Control Problem

- Given a reference/desired velocity $r(t)$, what should be the control input so that the car moves with the given velocity?
- Define the error, i.e., the difference between the reference velocity and the actual velocity as

$$e(t) = r(t) - x(t)$$

- Compute how this error changes with time,

$$\dot{e}(t) = \dot{r}(t) - \dot{x}(t) = \dot{r}(t) + \frac{c}{m}u(t),$$

- How to design control input for this case?

Regulation Control Problem

- Given a constant or fixed reference/desired velocity, i.e., $r(t) = R$ for all $t > 0$, what should be the control input so that the car moves with the given velocity?
- Define the error, i.e., the difference between the reference velocity and the actual velocity as

$$e(t) = R - x(t)$$

- Compute how this error changes with time,

$$\dot{e}(t) = \dot{R} - \dot{x}(t) = 0 + \frac{c}{m}u(t),$$

- How to design control input for this case?

Example: Robotic System

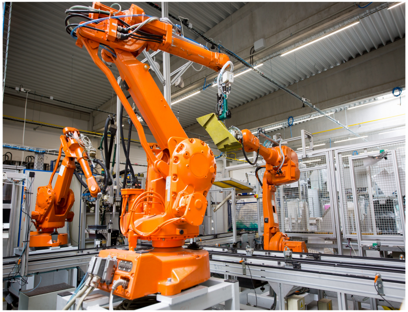

Figure: Robotic Systems (wiki). SKYWASH, DaVinci AEG, Dornier, Fraunhofer Institute, Putzmeister - Germany Using 2 Skywash robots for cleaning a Boeing 747-400 jumbo jet, its grounding time is reduced from 9 to 3.5 hours. The world´s largest cleaning brush travels a distance of approximately 3.8 kilometers and covers a surface of around 2,400 m^2 which is about 85% of the entire plane´s surface area. The kinematics consist of 5 main joints for the robot´s arm, and an additional one for the turning circle of the rotating washing brush.The Skywash includes database that contains the aircraft-specific geometrical data. A 3-D distance camera accurately positions the mobile robot next to the aircraft. The 3-D camera and the computer determine the aircraft´s ideal positioning, and thus the cleaning process begins.

Example

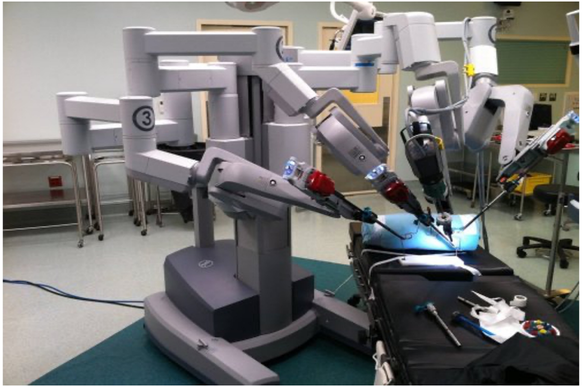

Figure: Medical Robotics

Feedback Control System

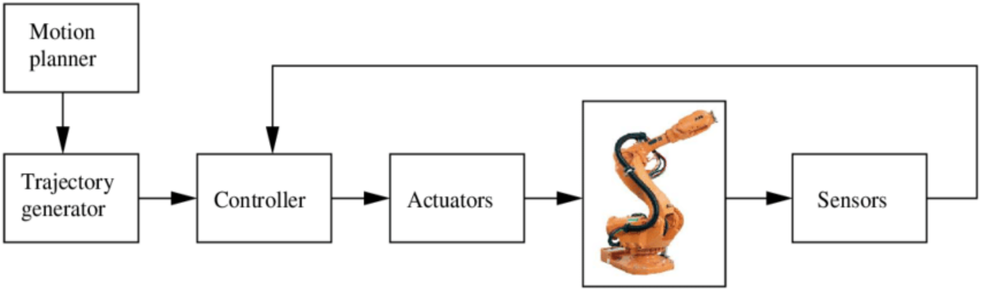

Figure: Block diagram of a feedback control system

Example Control Problem

- Given
  - The desired or the reference trajectory for the robotic system to track
  - Measurements from the sensor informing the actual path/trajectory of the robotic system
- To Do
  - Design control inputs or policies that steers the actual path traced by the robotic system is close to the reference trajectory

Physics-based Model

- Robotic arm

$$M(q)\ddot{q}(t) + V_m(q, \dot{q}) + G(q) + F(q, \dot{q}) = \tau(t) + \tau_d(t)$$

- Dynamic Equations - Newton-Euler method or Lagrangian Dynamics
- $q(t)$ Joint variable
- $M(q)$ Models of inertial mass
- $V_m(q, \dot{q})$ Models of coriolis/centripetal force
- $F(q, \dot{q})$ Models of friction
- $G(q)$ models of Gravity
- $\tau(t)$ Control torque
- $\tau_d(t)$ models of disturbance

Tracking Control Problem

- Let the desired trajectory for the robot manipulator be $q_d(t)$
- Now, we can define the tracking error as

$$e(t) = q_d(t) - q(t)$$

- Define the filtered tracking error as

$$r(t) = \dot{e}(t) + \lambda e(t)$$

- Filtered tracking error dynamics

$$\dot{r}(t) = \ddot{e}(t) + \lambda \dot{e}(t)$$

Tracking Control Problem

- Filtered tracking error dynamics are: $\dot{r}(t) = \ddot{e}(t) + \lambda \dot{e}(t)$
- Recall the robot dynamics: $M(q)\ddot{q}(t) + V_m(q,\dot{q}) + G(q) + F(q,\dot{q}) = \tau(t) + \tau_d(t)$

$$M\dot{r}(t) = -V_m r(t) - \tau(t) + h + \tau_d(t)$$
$$h = M(q)(\ddot{q}_d + \lambda\dot{e}) + V_m(q,\dot{q})(\dot{q}_d + \lambda e) + F(\dot{q}) + G(q)$$

Control Torque

$$\tau(t) = \hat{h} + K_v r(t)$$

with λ, K_v being a positive design parameter

- The closed-loop dynamics is obtained as

$$M\dot{r}(t) = -V_m r(t) - \hat{h} - K_v r(t) + h + \tau_d(t)$$

NN Control - Function Approximator

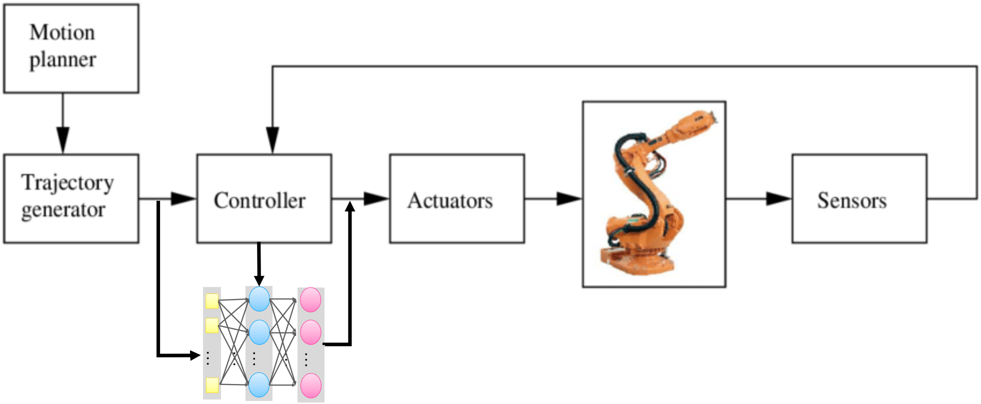

Figure: Feedback NN control

Steady-State Analysis of Feedback Control System

- Filtered tracking error dynamics

$$\dot{r}(t) = -\frac{V_m - K_v}{M} r(t) + \frac{h - \hat{h}}{M} + \frac{\tau_d(t)}{M}$$

$$\downarrow$$

$$\dot{r}(t) = -Kr(t) + N_\varepsilon + d(t)$$

- What does the Lyapunov approach reveal?