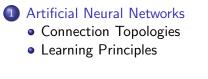
## Linear Layered Networks



2 Linear Feed-Forward Networks

# What can they Compute?Hebbs Learning Hypothesis

Örjan Ekeberg	Brain Modeling and Machine Learning	Örjan Ekeberg	Brain Modeling and Machine Learning
Artificial Neural Networks Linear Feed-Forward Networks What can they Compute?	Connection Topologies Learning Principles	Artificial Neural Networks Linear Feed-Forward Networks What can they Compute?	Connection Topologies Learning Principles
		Connection Topologies	

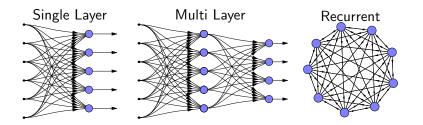
#### Artificial Neural Networks

- Connection Topologies
- Learning Principles

2 Linear Feed-Forward Networks

#### 3 What can they Compute?

• Hebbs Learning Hypothesis



Artificial Neural Networks Linear Feed-Forward Networks What can they Compute? Learning Principles	Artificial Neural Networks Linear Feed-Forward Networks What can they Compute?
<ul> <li>Coincidence Detection</li> <li>Error Correction</li> <li>Competitive Learning</li> </ul>	<ul> <li>Artificial Neural Networks         <ul> <li>Connection Topologies</li> <li>Learning Principles</li> </ul> </li> <li>2 Linear Feed-Forward Networks</li> <li>3 What can they Compute?         <ul> <li>Hebbs Learning Hypothesis</li> </ul> </li> </ul>
Örjan Ekeberg       Brain Modeling and Machine Learning         Artificial Neural Networks       Hebbs Learning Hypothesis         Linear Feed-Forward Networks       Hebbs Learning Hypothesis         Linear Feed-Forward Networks       Hebbs Learning Hypothesis	Örjan Ekeberg       Brain Modeling and Machine Learning         Artificial Neural Networks       Hebbs Learning Hypothesis         Linear Feed-Forward Networks       Hebbs Learning Hypothesis         Linear Feed-Forward Networks       Ketworks         Vhat can they Compute?       Hebbs Learning Hypothesis
	Cascaded Linear Networks
What can be computed by a linear network? $x_{1} \leftarrow w_{1}$ $x_{2} \leftarrow w_{2}$ $x_{3} \leftarrow w_{4}$ $x_{5} \leftarrow w_{5}$ $w_{5}$ $w_{5}$ $w_{5}$ $w_{1} \leftarrow w_{1}$ $x_{1} \leftarrow w_{1}$ $x_{2} \leftarrow w_{1}$ $x_{2} \leftarrow w_{1}$ $x_{3} \leftarrow w_{1}$ $x_{4} \leftarrow w_{4}$ $x_{5} \leftarrow w_{5}$ $w_{5}$	
$y = \vec{w}^T \cdot \vec{x} \qquad \qquad \vec{y} = W \vec{x}$	$\vec{y} = W_3(W_2(W_1\vec{x})) = (W_3W_2W_1)\vec{x}$

 $\vec{w}$  — Weight Vector W — Weight Matrix

Still a linear mapping

Let  $W = W_3 W_2 W_1 \quad \Rightarrow \quad \vec{y} = W \vec{x}$ 

## Storing Mappings

The "program" resides in the weights How do we find the right weights? Learning  $\approx$  Change the weights to achieve better performance

#### Hebbs Learning Hypothesis

Simultaneous activation of two neurons strengthens the synaptic connection between them

Common interpretation:

$$\Delta w_{ij} = x_j y_i$$

Note! Outer Product

Brain Modeling and Machine Learning

### Storing Mappings

Storing a mapping using Hebbs rule

$$\vec{x}_1 \rightarrow \vec{y}_1 \qquad \vec{x}_2 \rightarrow \vec{y}_2 \qquad \vec{x}_3 \rightarrow \vec{y}_3 \quad \cdots$$

Hebbs rule

 $\Delta w_{ij} = x_j y_i$ 

Result

$$W = \sum_{p} \vec{y}_{p} \vec{x}_{p}$$

Örjan Ekeberg

Correlation Memory

Örjan Ekeberg Brain Modeling and Machine Learning

Artificial Neural Networks Linear Feed-Forward Networks What can they Compute?

#### Storing Mappings

Retrieving a Memory Trace

$$W = \sum_{p} \vec{y}_{p} \vec{x}_{p}^{T}$$
$$\vec{x}_{k} \to ?$$

$$\vec{y}_{\text{out}} = W \vec{x}_k = \sum_p (\vec{y}_p \vec{x}_p^T) \vec{x}_k = \sum_p \vec{y}_p (\vec{x}_p^T \vec{x}_k) =$$
$$= \vec{y}_k (\vec{x}_k^T \vec{x}_k) + \sum_{p \neq k} \vec{y}_p (\vec{x}_p^T \vec{x}_k) \approx \alpha \vec{y}_k$$

• Perfect memory if the patterns  $\vec{x}_p$  are orthogonal