Artificial Neural Networks

Chapter 3: Supervised Learning Neural Networks
Neural Network Types

Prof. Dr. BARSEKH-ONJI Aboud

Facultad de Ingeniería
Universidad Anáhuac México

January 16, 2026

Agenda

Neural Network Architectures

Architectural Classification

Neural Networks are characterized by:

- **Architecture**: The arrangement of neurons and their connections.
- **Learning Algorithm**: The method used to adjust weights.
- **Activation Functions**: The equations controlling the output.

This chapter focuses on the main architecture types for Supervised Learning:

- Feedforward Neural Networks (FNN)
- Recurrent Neural Networks (RNN)

Agenda

Feedforward Classification

Characteristics

In Feedforward Neural Networks (FNN), the signal flows in only one direction: from input to output. There are no cycles or loops.

- Single-Layer Perceptron
- Multi-Layer Perceptron (MLP)
- Functional Link Neural Networks (FLNN)
- Product Unit Neural Networks (PUNN)

Single-Layer Perceptron

Structure

The simplest form of ANN.

- Only has Input and Output layers.
- Can only solve linearly separable problems.
- Uses step activation function (originally).

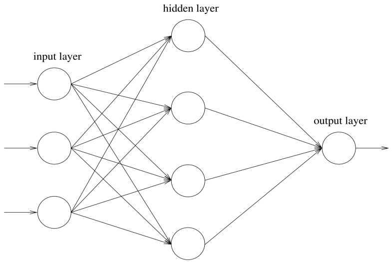


Figure 1: Single-Layer Network Structure.

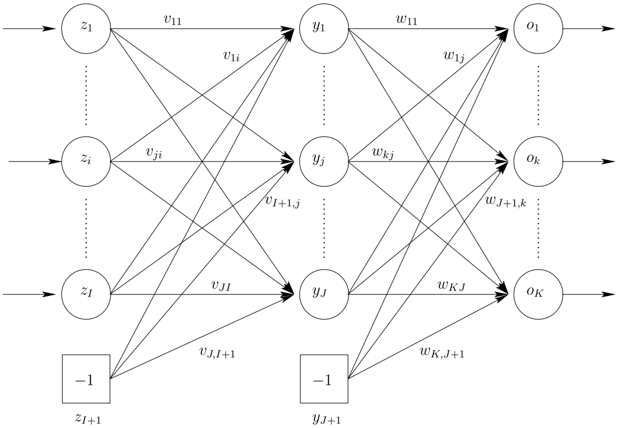
Multi-Layer Perceptron (MLP)

Adding Hidden Layers

Includes one or more "hidden" layers between input and output.

- Can approximate any continuous function (Universal Approximation Theorem).
- Solves non-linearly separable problems (e.g., XOR).
- Typically trained with Backpropagation.

Multi-Layer Perceptron (MLP)



Functional Link Neural Networks (FLNN)

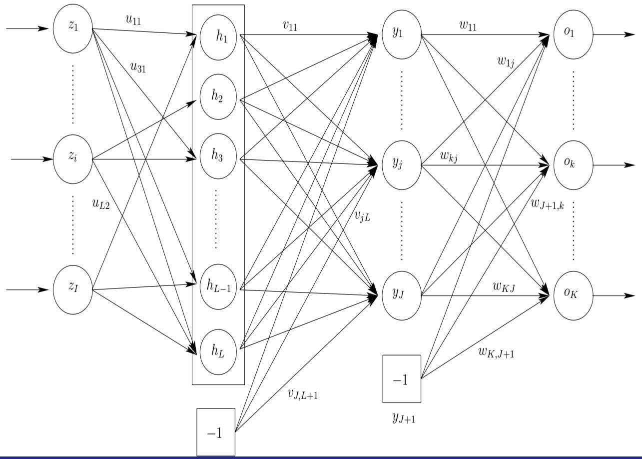
Higher-Order Combinations

Instead of adding hidden layers, inputs are expanded using functional links.

- **Tensor Model**: Multiplies inputs ($z_1 z_2, z_1^2$, etc.)
- **Functional Expansion**: Passing inputs through functions ($\sin(z_1), \cos(z_1)$).

Advantages: Faster training (no hidden layers to train), effectively a single-layer network with expanded input space.

Functional Link Neural Networks (FLNN)



Product Unit Neural Networks

Multiplicative Inputs

Instead of summing weighted inputs ($\sum z_i v_i$), Product Units calculate the weighted product:

$$net_j = \prod_{i=1}^{I} z_i^{v_{ji}} \tag{1}$$

- Can implement higher-order functions more efficiently.
- Harder to train (complex error surfaces).

Agenda

Recurrent Features

Feedback Loops

Recurrent Neural Networks (RNNs) have feedback connections where outputs are fed back into their own inputs or previous layers.

- **Memory**: They have a temporal state/memory of previous inputs.
- Suitable for time-series prediction and sequence processing.

Elman Recurrent Network

Context Layer

A simple recurrent network (SRN).

- Hidden layer outputs are copied to "Context Units".
- Context units feed back into the hidden layer in the next time step.
- Allows learning of temporal patterns.

Jordan Recurrent Network

Output Feedback

Similar to Elman, but the feedback comes from the **Output Layer** instead of the hidden layer.

- Context units store the previous output.
- Often used for control systems where the next state depends on the previous action (output).

Elman and Jordan Recurrent Network

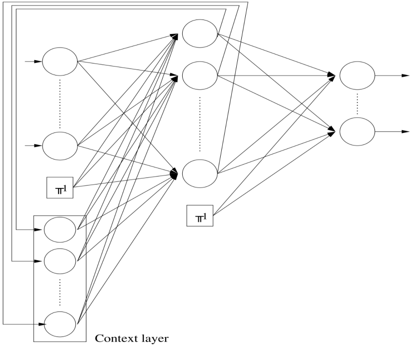


Figure 4: Elman Recurrent Network Structure.

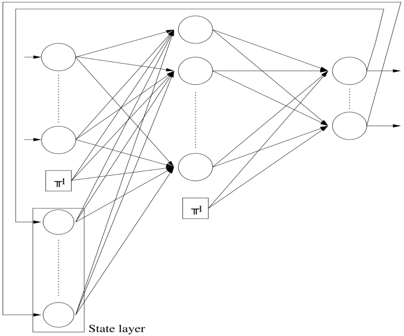


Figure 5: Jordan Recurrent Network Structure.

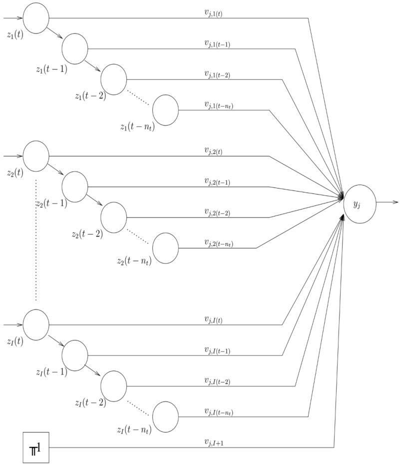
Agenda

Time-Delay Neural Networks

The output of a TDNN is calculated as:

$$o_{k,p} = f_{o_k}\left(\sum_{i=1}^{I+1} u_{ki} z_i + \sum_{j=1}^{J} w_{kj} f_{y_j}\left(\sum_{i=1}^{I+1} v_{ji} z_i + \sum_{l=1}^{j-1} s_{jl} y_l\right)\right) \quad (2)$$

Time-Delay Neural Networks



Agenda

Cascade Neural Networks

The output of a Cascade Neural Network is calculated as:

$$o_{k,p} = f_{o_k}\left(\sum_{j=1}^{J+1} w_{kj} f_{y_j}\left(\sum_{i=1}^{I}\sum_{t=0}^{n_t} v_{j,i(t)} z_{i,p}(t) + z_{I+1} v_{j,I+1}\right)\right) \tag{3}$$

Cascade Neural Networks

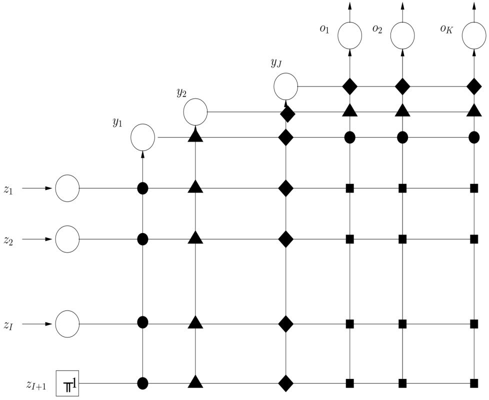


Figure 7: Cascade Neural Network Structure.

Agenda

Summary

- **Feedforward Networks** (MLP) are the standard for static pattern classification and regression.
- **Functional Link** and **Product Units** offer alternative ways to handle non-linearity without deep layers.
- **Recurrent Networks** (Elman, Jordan) introduce memory, making them essential for temporal and sequential data problems.
- **Time-Delay Neural Networks** are used for time series prediction and sequence processing.
- **Cascade Neural Networks** are used for function approximation and regression.