The Cascade-Correlation Learning Architecture S. Fahlman & C. Lebiere August 1991

Presentation by Jeremy Wurbs

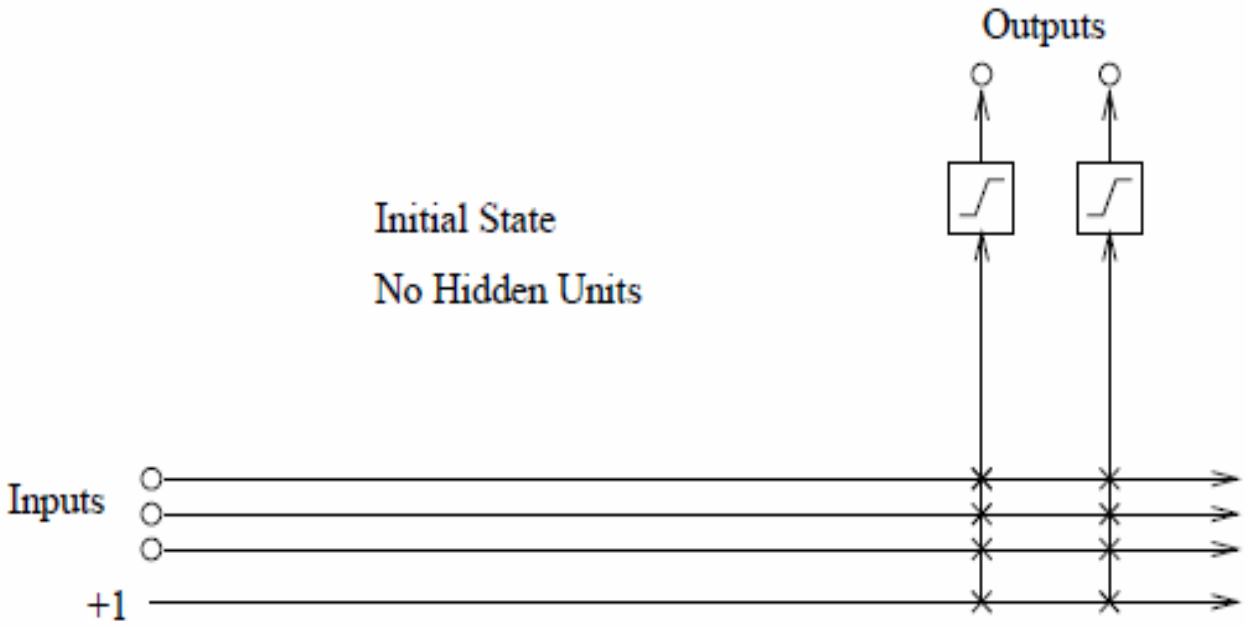
CSCE 636

2.22.2010

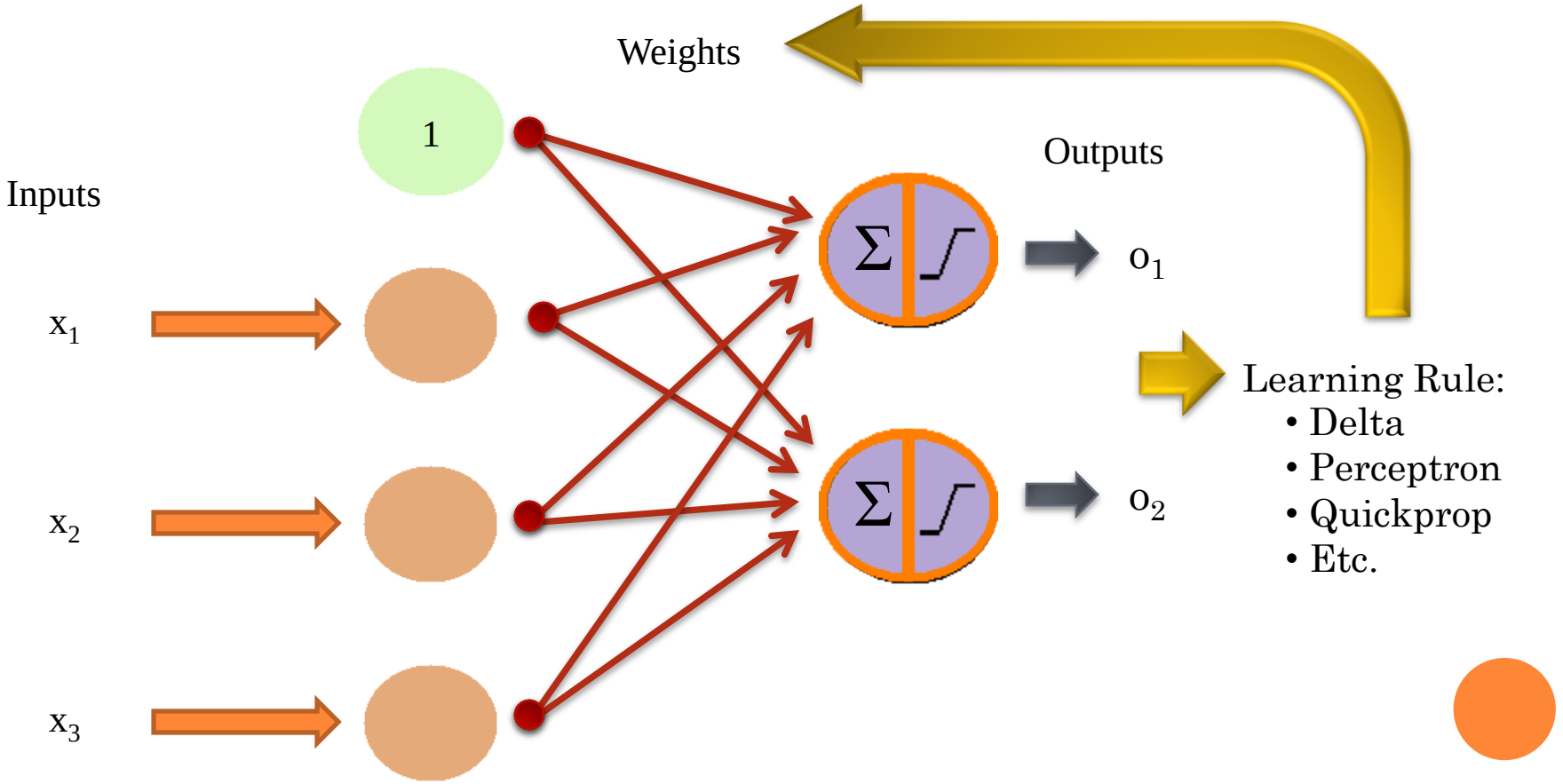
Presentation Overview

- CC Learning Architecture
  - Basic Architecture
  - Adding Hidden Units
- Advantages of CCLA
- Benchmark Tests
- Closing Remarks

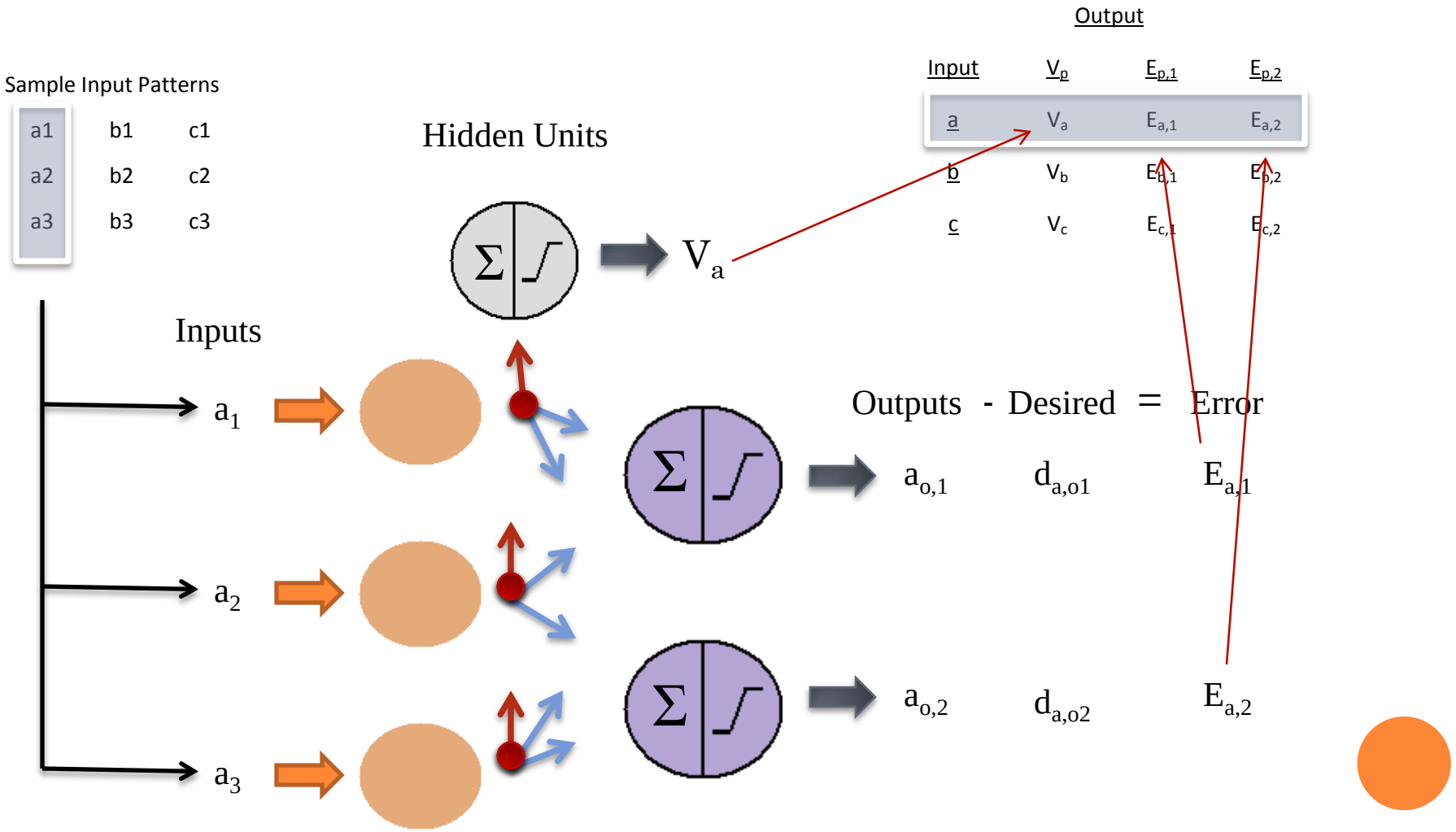
CCLA – Basic Architecture



CCLA – Basic Architecture



CCLA – Adding Hidden Units

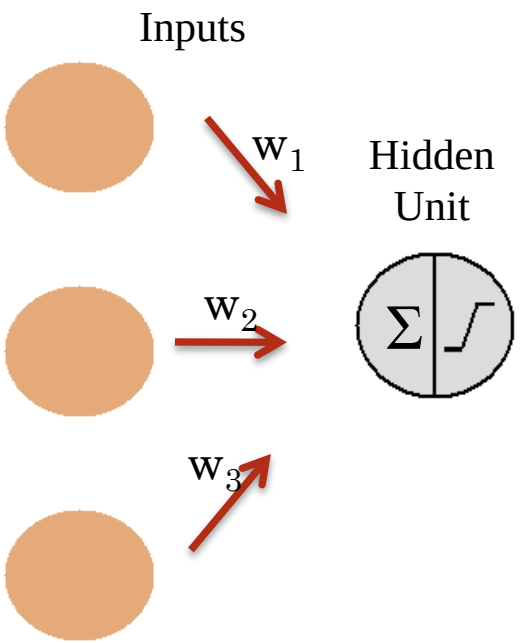


CCLA – Adding Hidden Units

	Output		
Input	V_p	$E_{p,1}$	$E_{p,2}$
a	V_a	$E_{a,1}$	$E_{a,2}$
b	V_b	$E_{b,1}$	$E_{b,2}$
c	V_c	$E_{c,1}$	$E_{c,2}$

$$S = \sum_o \left| \sum_p (V_p - \bar{V})(E_{p,o} - \overline{E_o}) \right|$$

$$\frac{\partial S}{\partial w_i} = \sum_{p,o} \sigma_o (E_{p,o} - \overline{E_o}) f_p' I_{i,p}$$

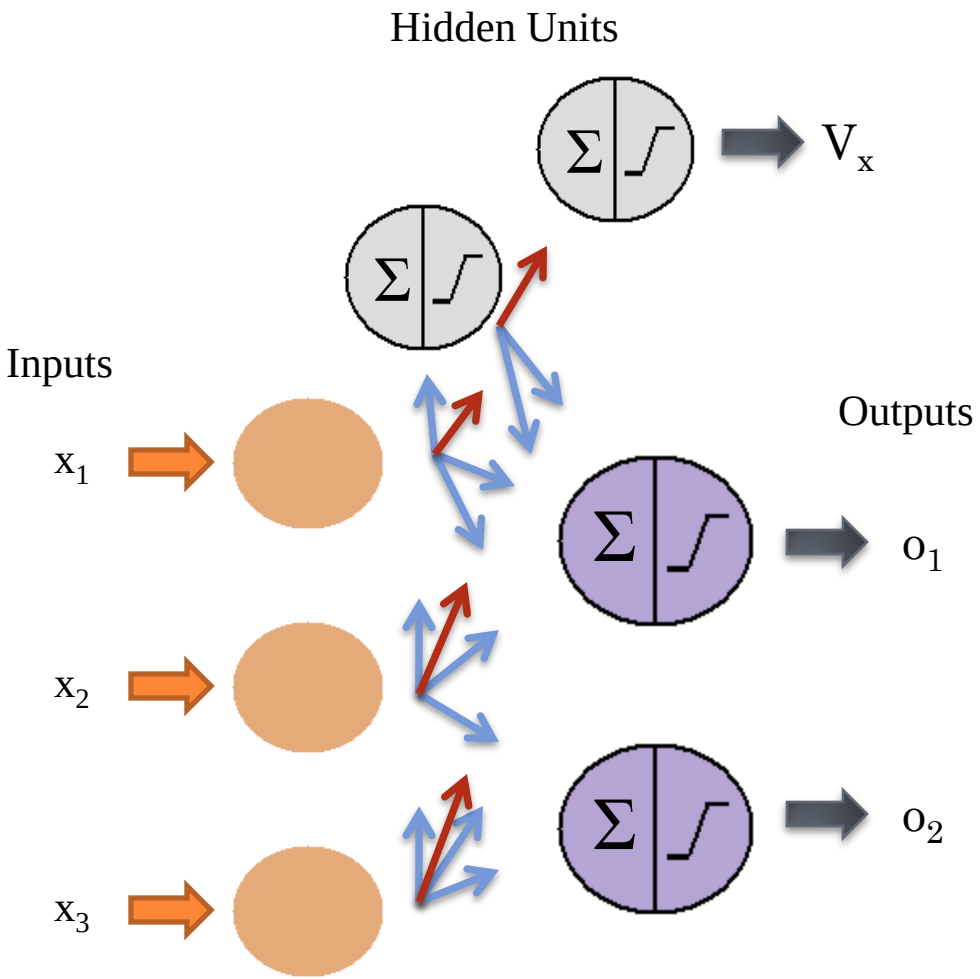


where

- $\sigma_o = \text{sign}(V_p - E_{p,o})$
- $I_{i,p}$ = input to the CU from unit i, pattern p
- f_p' = derivative of the CU's activation function wrt the sum of its inputs

*CU denotes 'candidate unit'

CCLA – Adding Hidden Units



Why Use Cascade-Correlation LA? Cited Problems

- The Step-Size Problem
  - How large should each gradient descent step be?
  - Momentum Terms
  - Quickprop
- The Moving Target Problem
  - Lack of communication b/w neurons
  - Herd Effect
  - Similar to adjusting spokes on a bicycle wheel

Why Use Cascade-Correlation LA? General Advantages

- Each hidden trained one at a time, limiting the moving target problem
- Network dimensions not needed in advance
- Easily builds higher order features
  - Complex learning structure that builds many layers quickly
  - Hidden units may use different activation functions
- Feature detectors aren't cannibalized
- Candidate pools can be used to assure unit utility

Benchmark Tests: 2-Spiral Problem

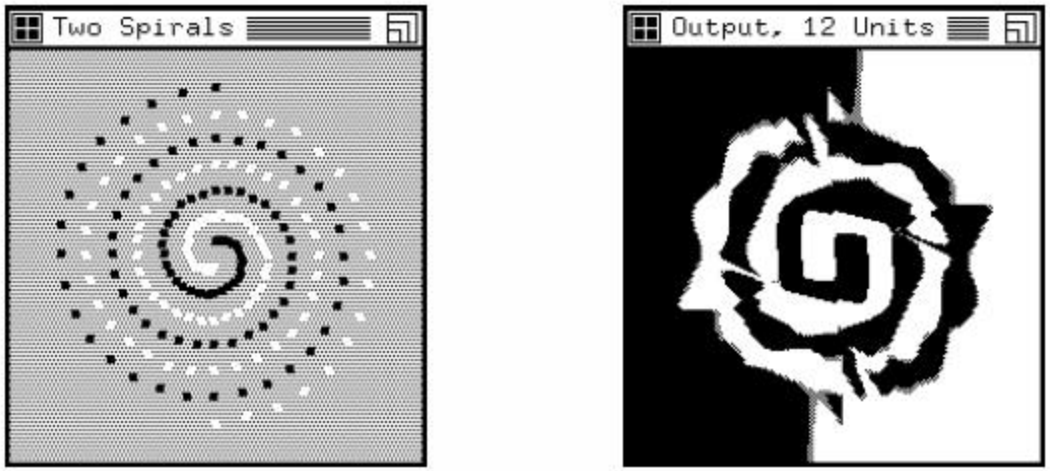
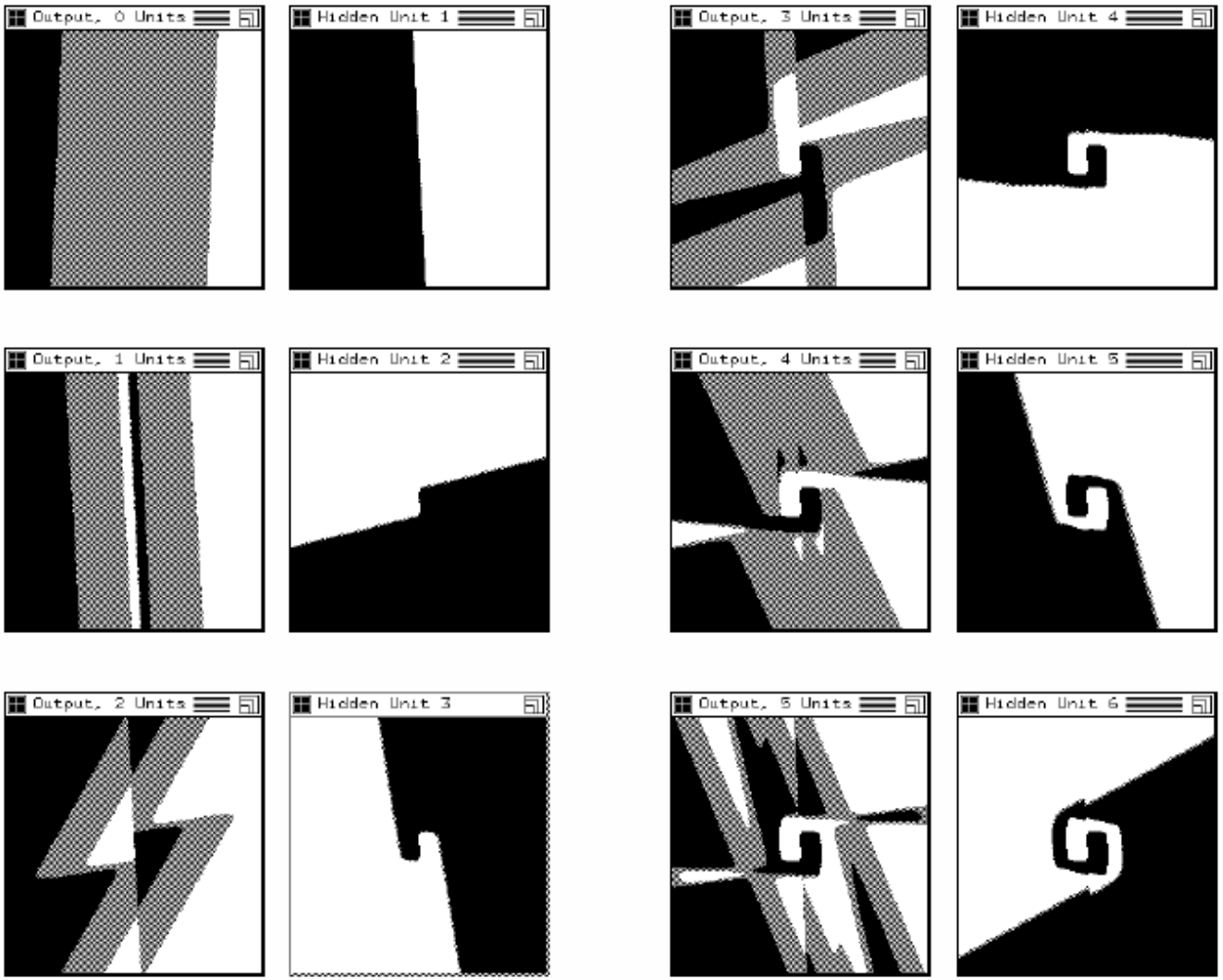
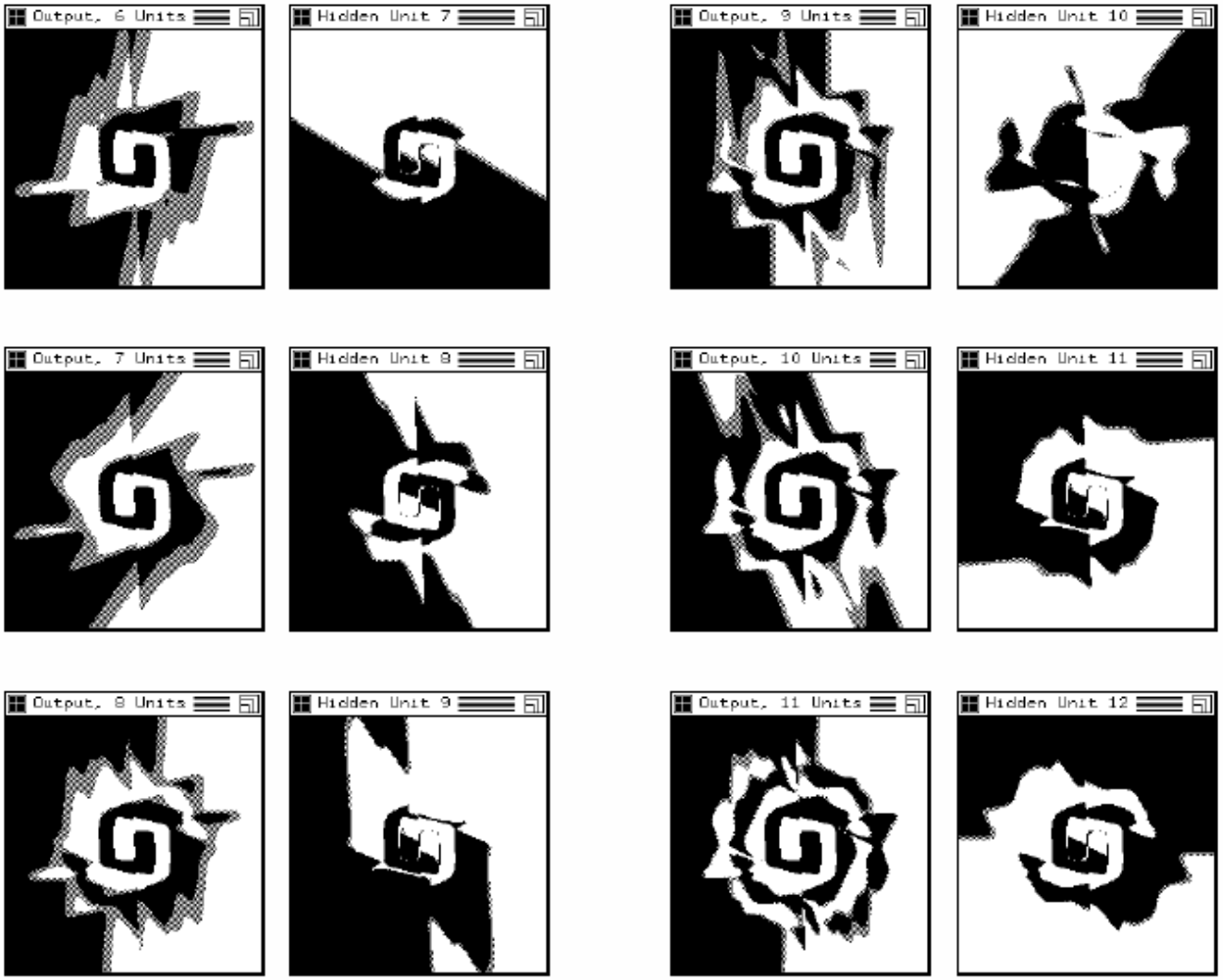


Figure 2: Training points for the two-spirals problem, and output pattern for one network trained with Cascade-Correlation.

Benchmark Tests: 2-Spiral Problem



Benchmark Tests: 2-Spiral Problem



Benchmark Tests: N-Parity Problem

- N-Parity Problem:

N=2:

	0	1
0	+	-
1	-	+

N=3:

$b_3 = 0$:

	0	1
0	+	-
1	-	+

$b_3 = 1$:

	0	1
0	-	+
1	+	-

N=4:

…

Benchmark Tests: N-Parity Problem

- Benchmark Results:

N	Cases	Hidden Units	Average Epochs
2	4	1	24
3	8	1	32
4	16	2	66
5	32	2–3	142
6	64	3	161
7	128	4–5	292
8	256	4–5	357

N = 10:

Train Cases	Test Cases	Hidden Units	Train Epochs	Test Errs	% Errs
512	512	4	282	9	1.8%
512	512	7	551	30	5.8%
512	512	7	491	32	6.2%
512	512	5	409	14	2.7%
256	768	4	382	111	14.4%
256	768	4	362	90	11.7%
256	768	4	276	55	7.2%
256	768	4	311	49	6.4%

Closing Remarks

- First ‘complex’ network architecture we’ve seen
- First network to dynamically add new hidden units & layers
- Paper was published nearly 2 decades ago; progress?