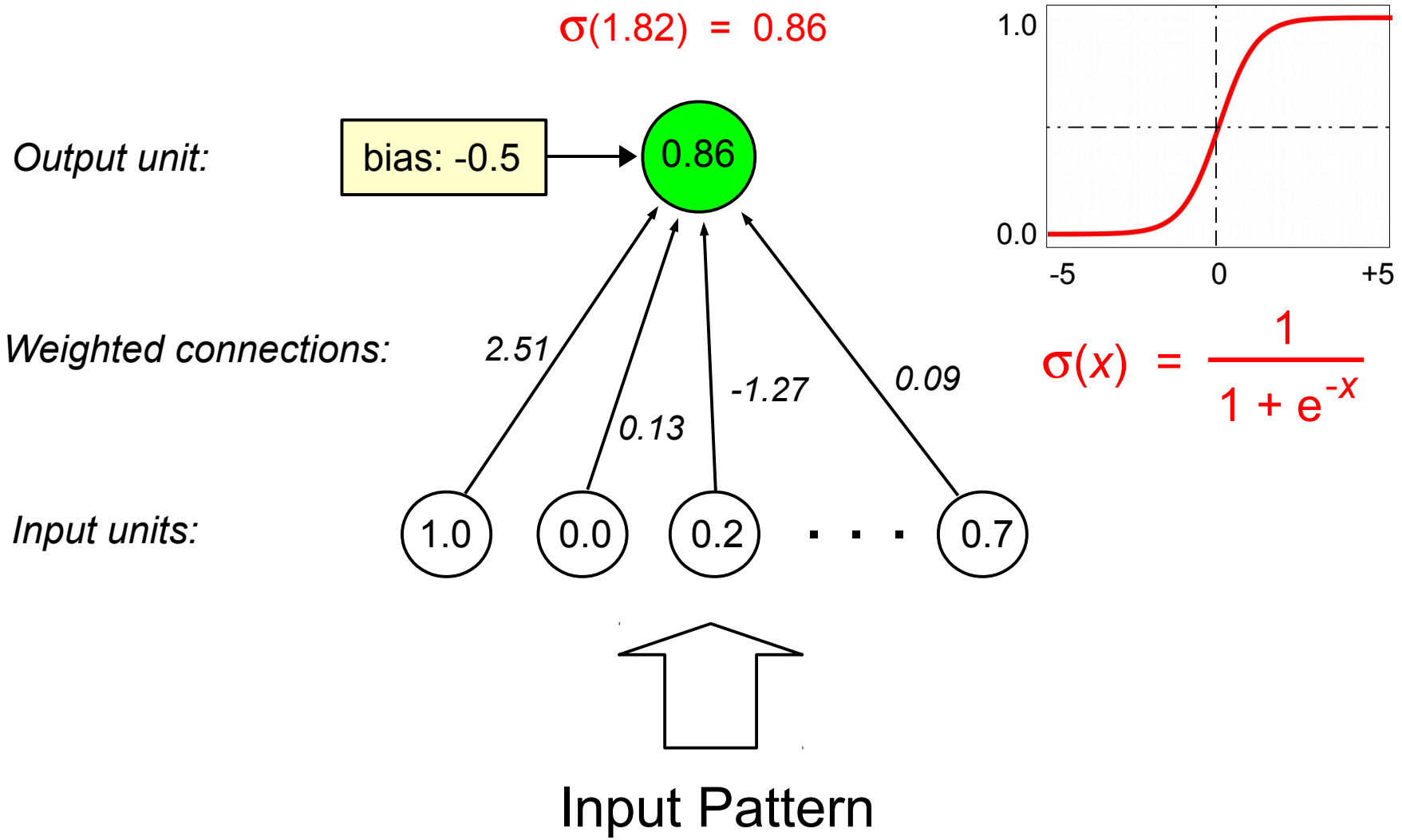
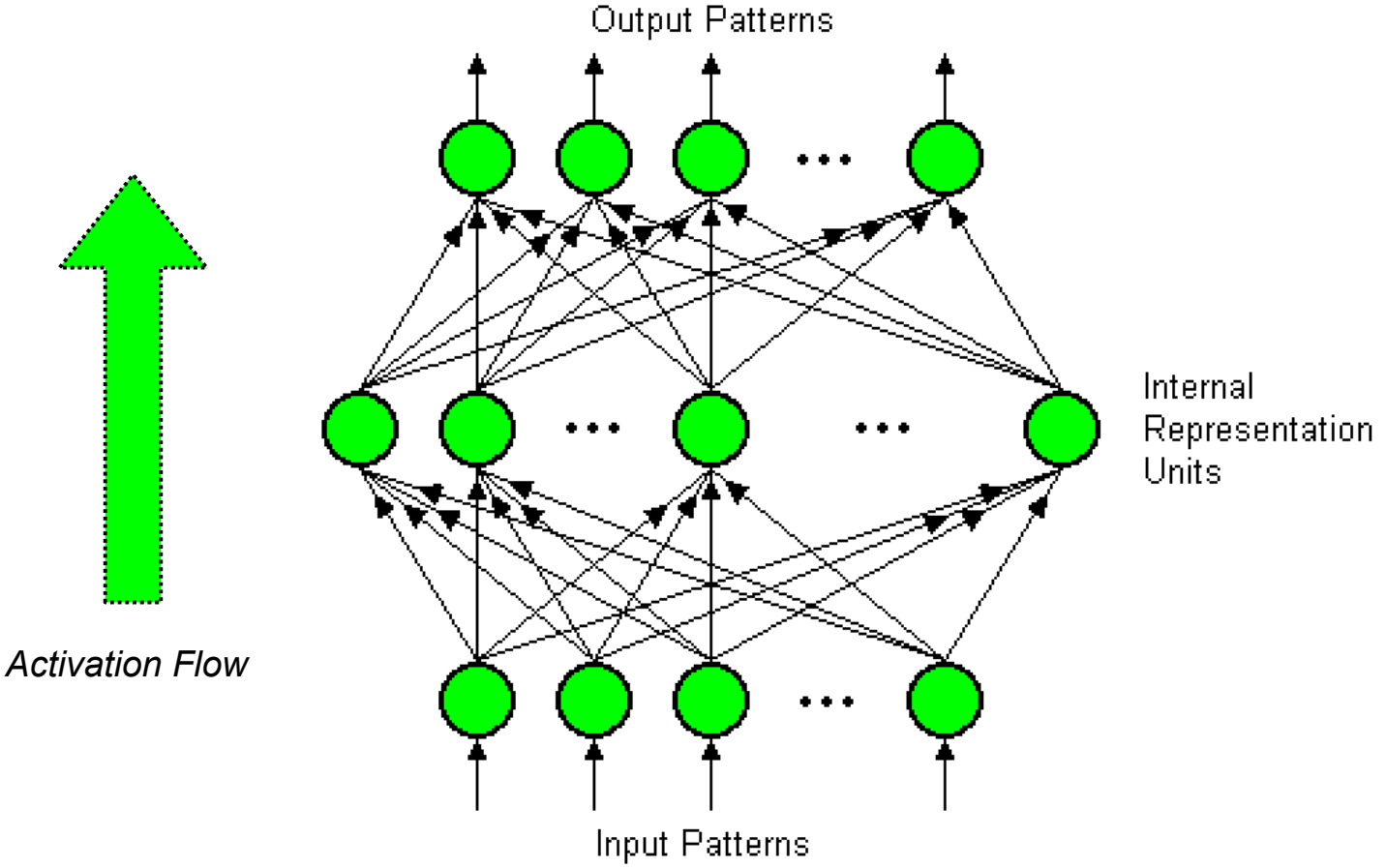
Artificial Neurons: Continuous Version

1.0 × 2.51 + **0.0** × 0.13 + **0.2** × -1.27 + . . . + **0.7** × 0.09 + **-0.5** = 1.82

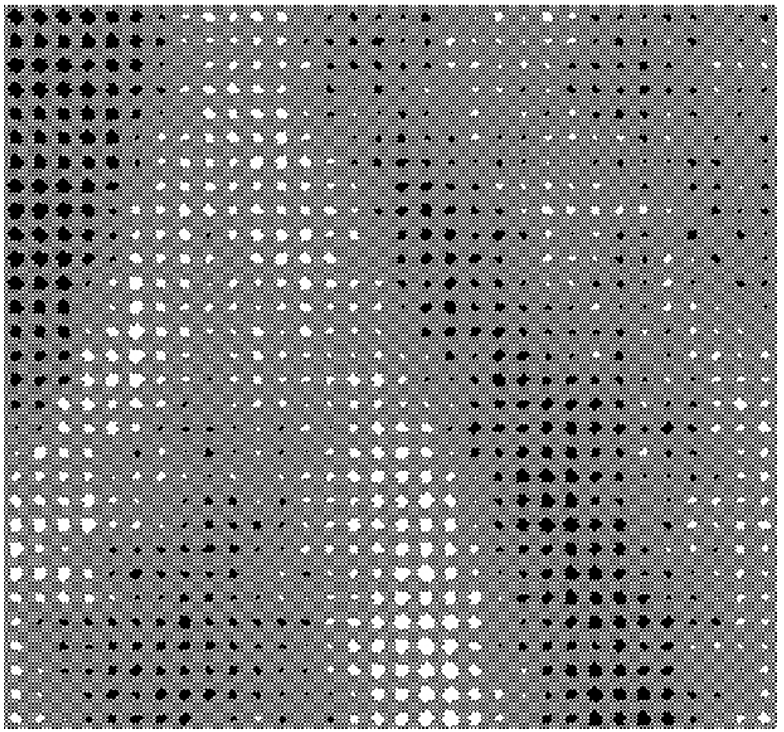


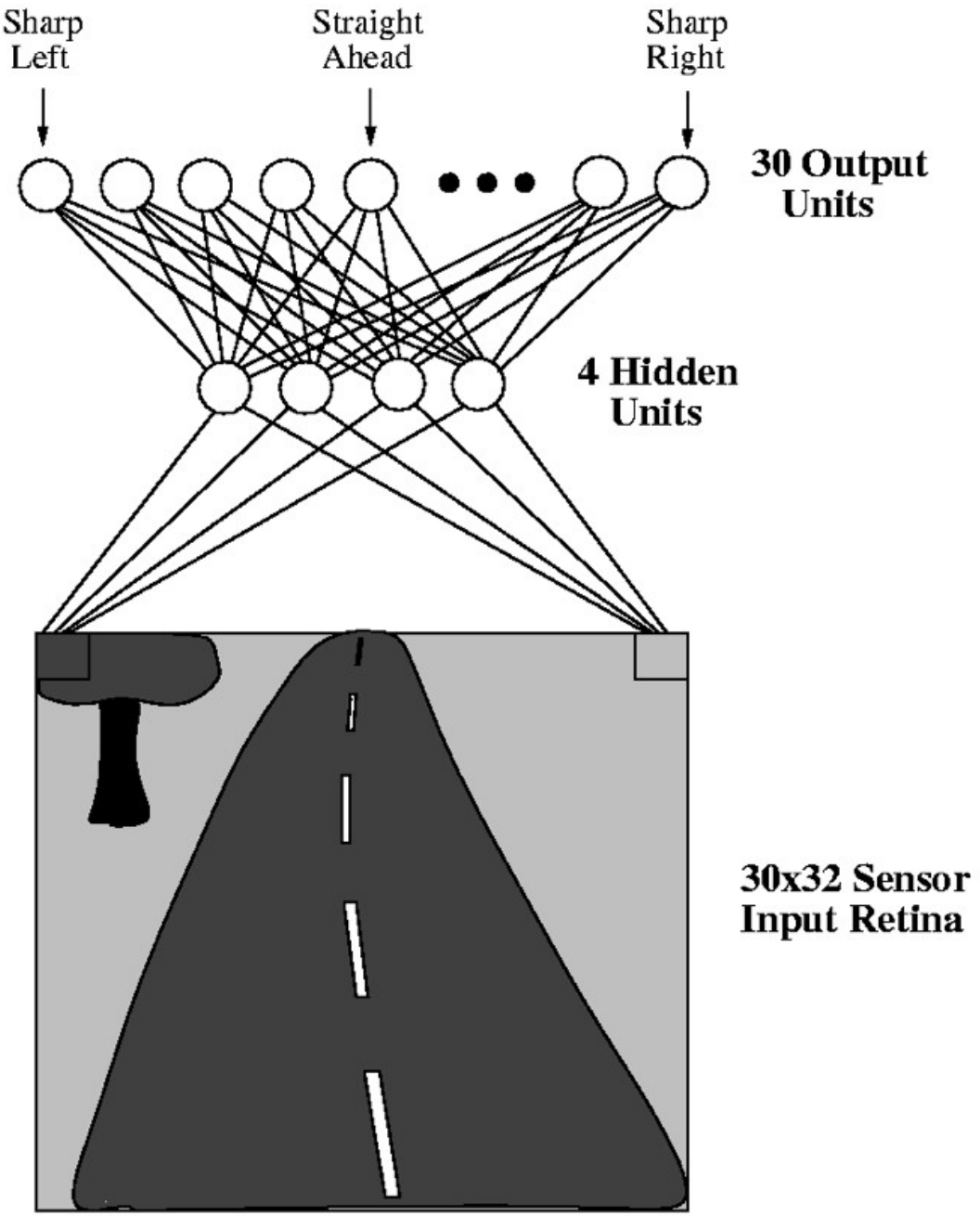
Pattern Associator Networks

- Units are arranged into successive layers
- Feed-forward connections only
- Layer activations represent stimulus/response associations

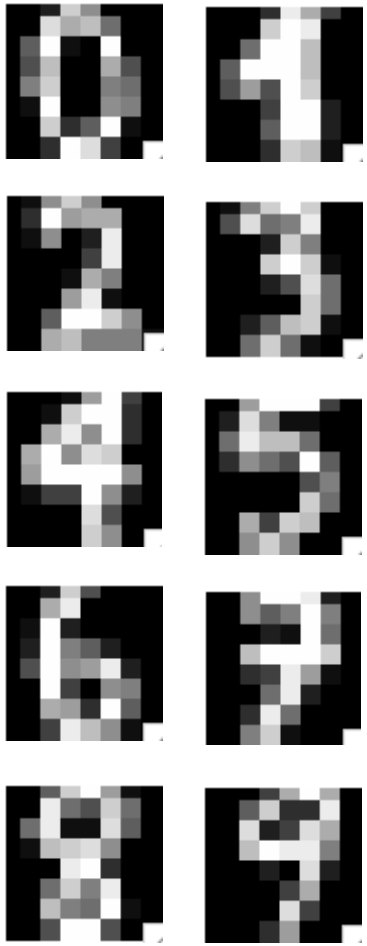


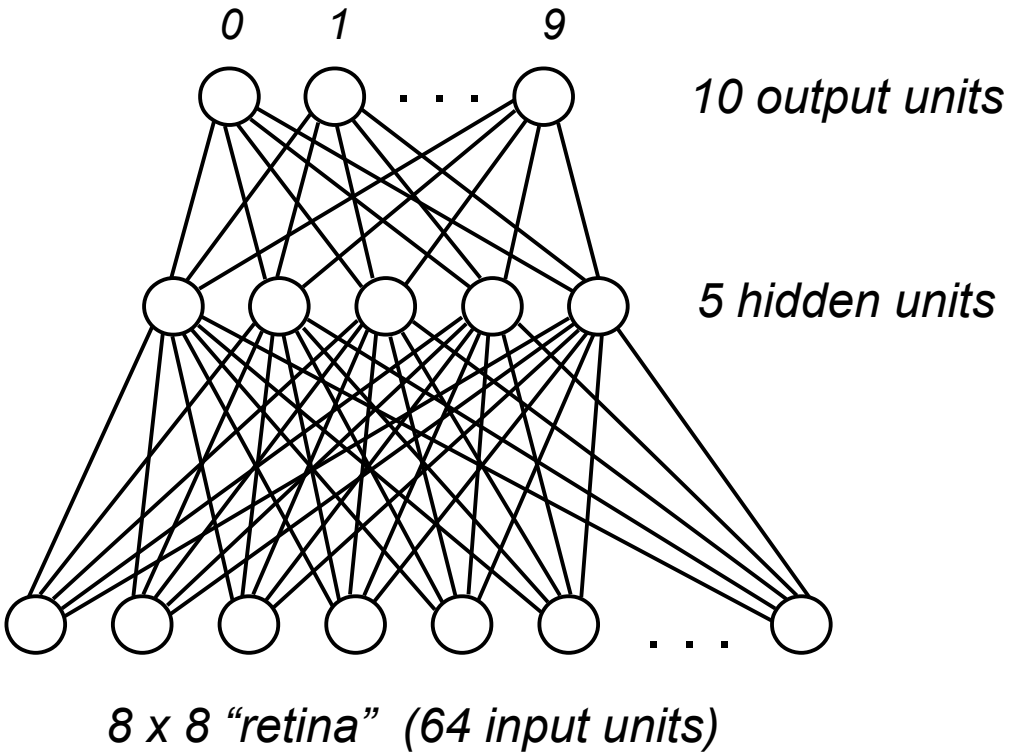
ALVINN



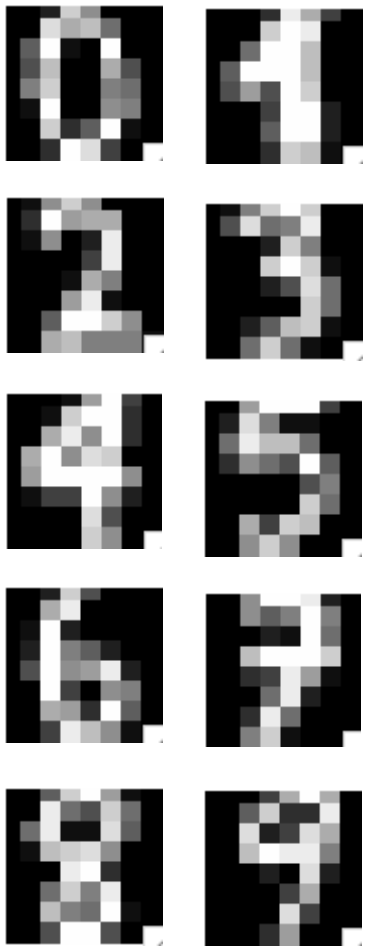

Sharp Left
Straight Ahead
Sharp Right
30 Output Units
4 Hidden Units
30x32 Sensor Input Retina

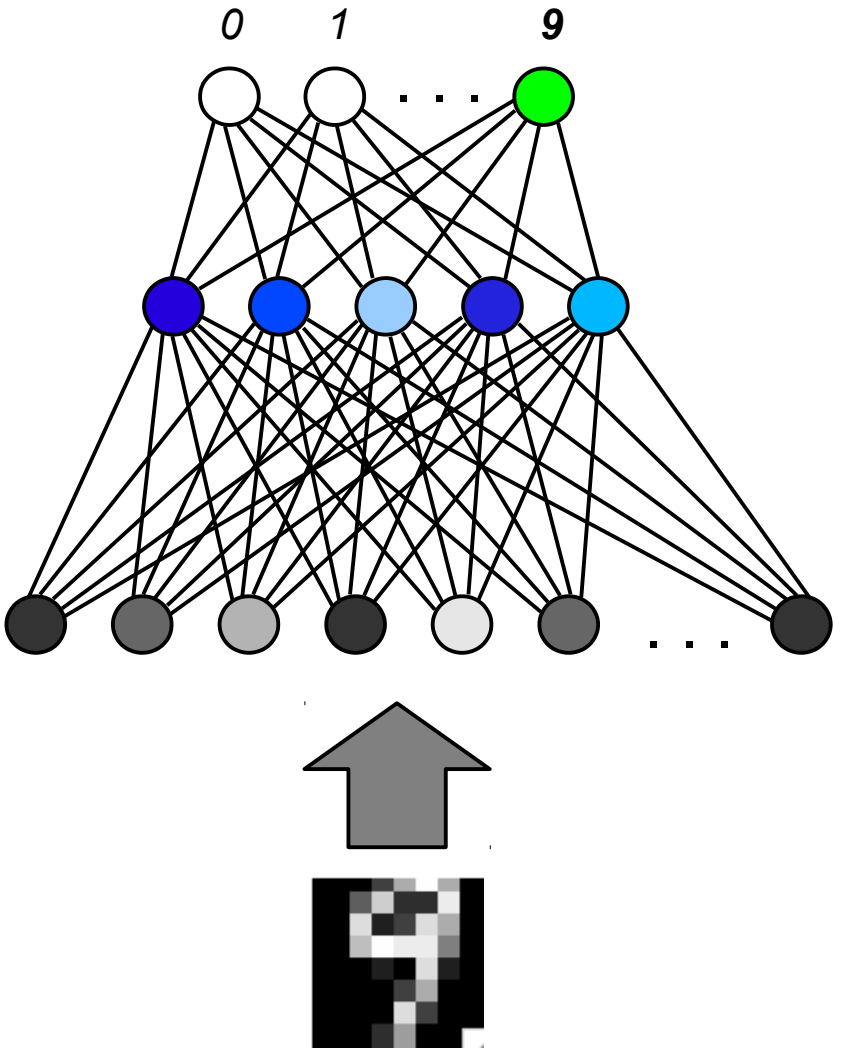
Recognizing Handwritten Digits



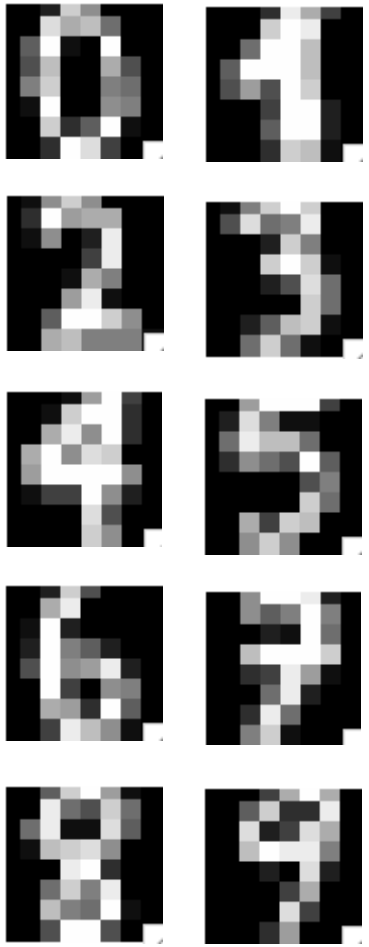


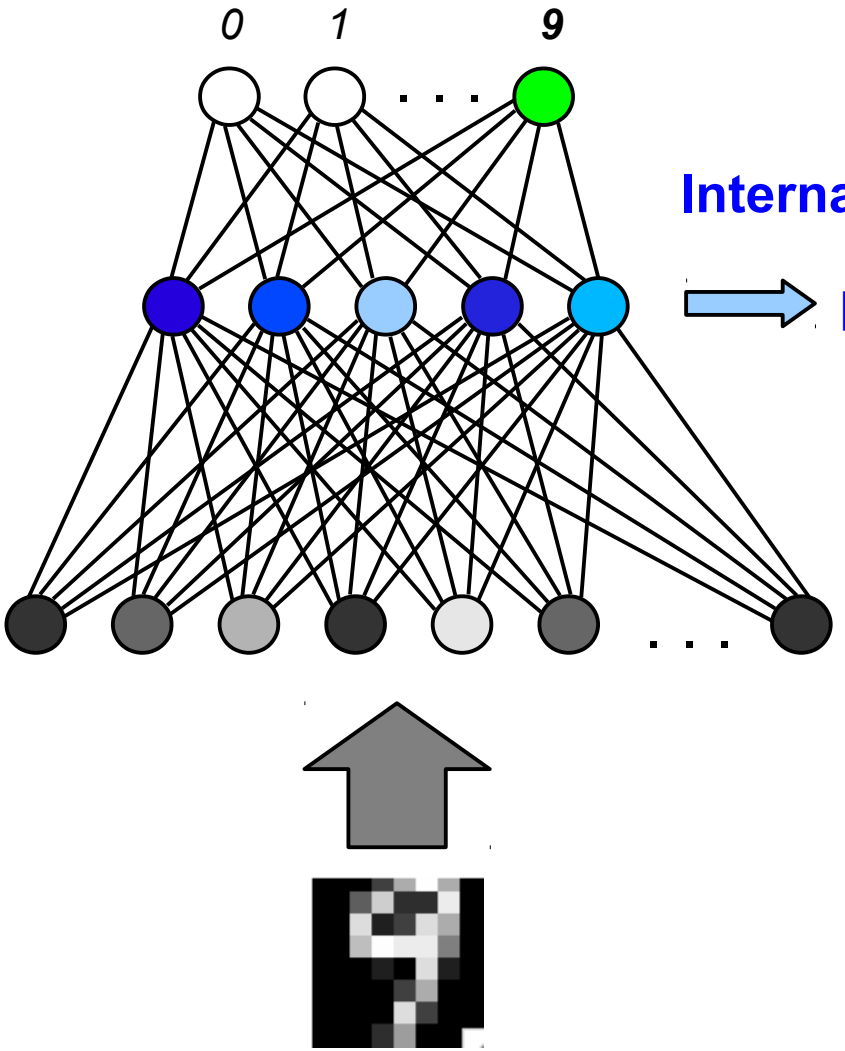
Recognizing Handwritten Digits





Recognizing Handwritten Digits



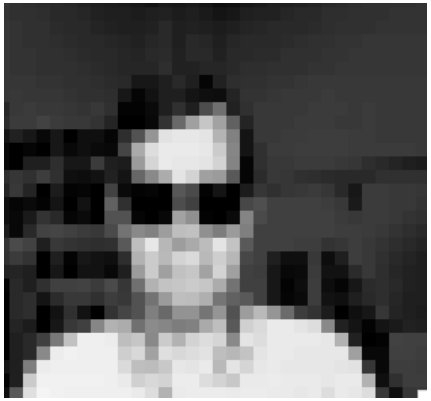


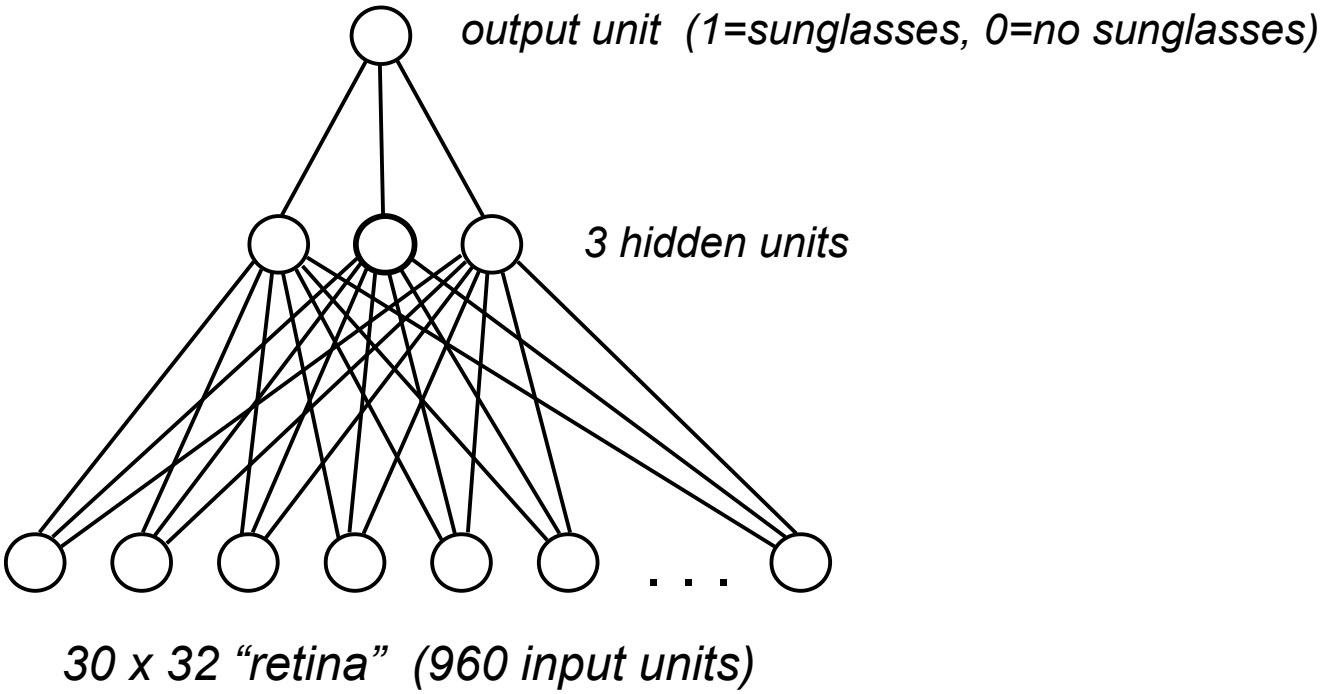
Internal representation of

[0.54, 0.16, 0.77, 0.29, 0.85]

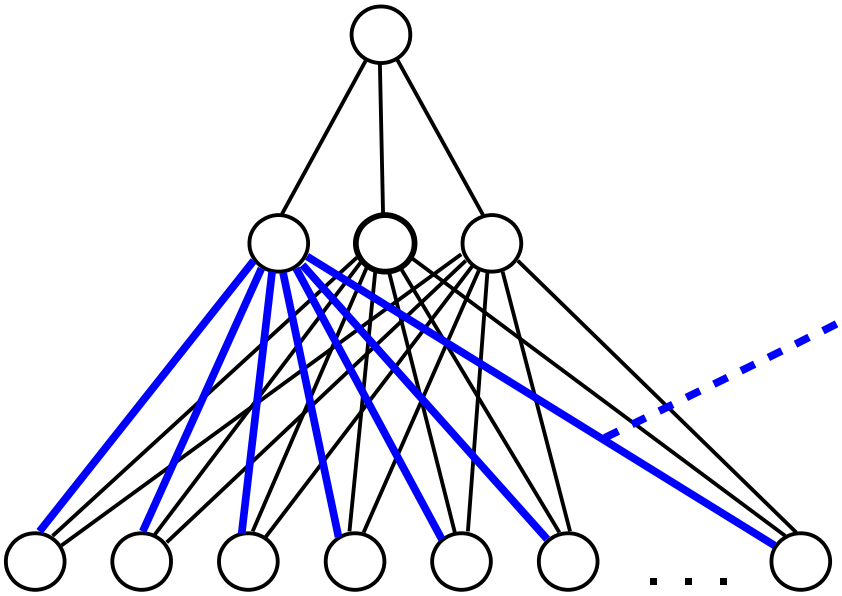
Handwritten Digits Demo

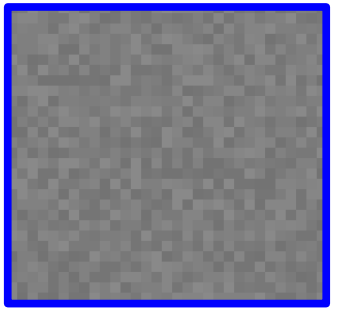
Recognizing Sunglasses





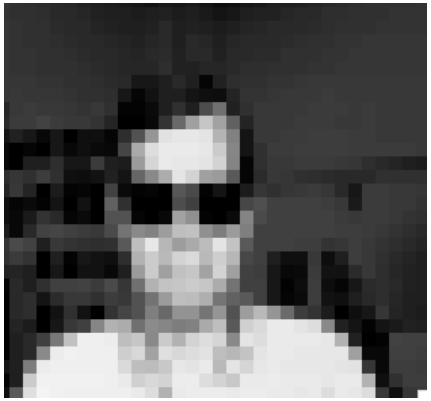
Recognizing Sunglasses

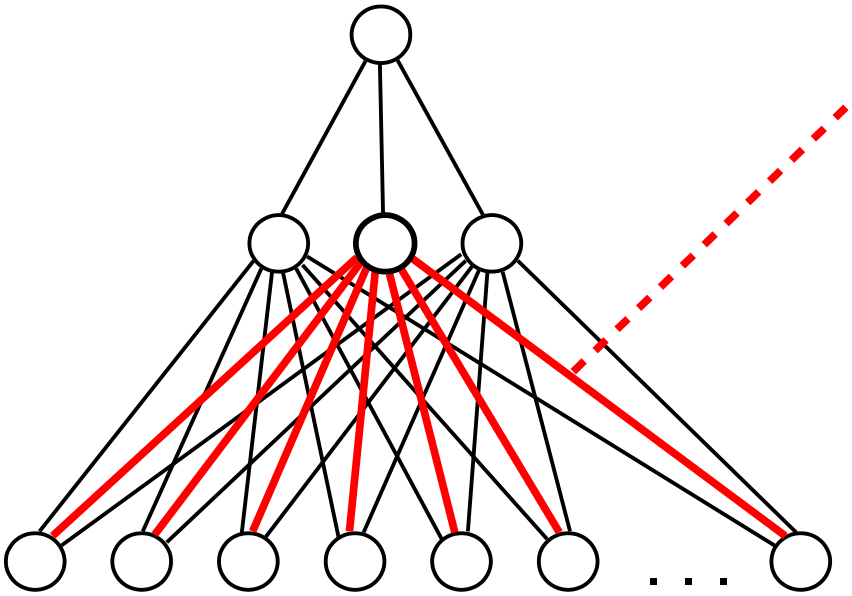


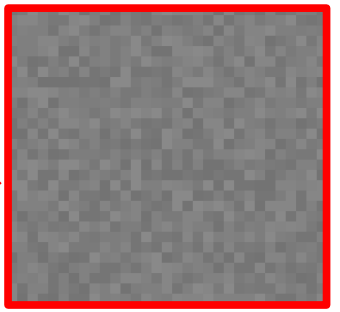


Weights from “retina” to first hidden unit

Recognizing Sunglasses

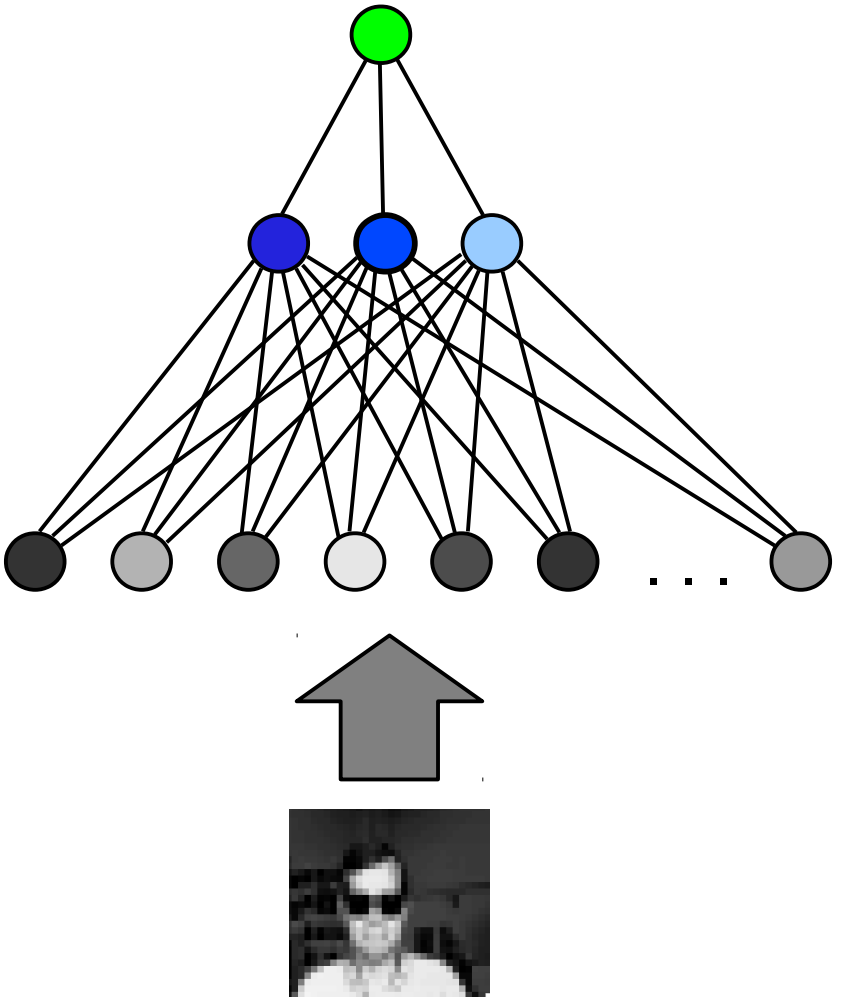






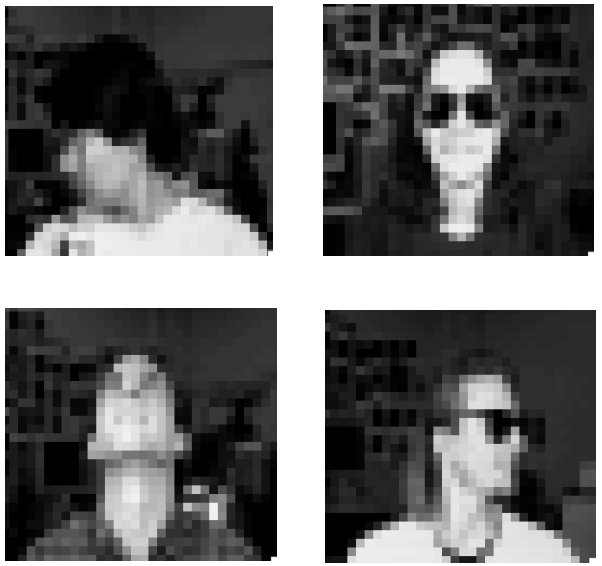
Weights from "retina" to middle hidden unit

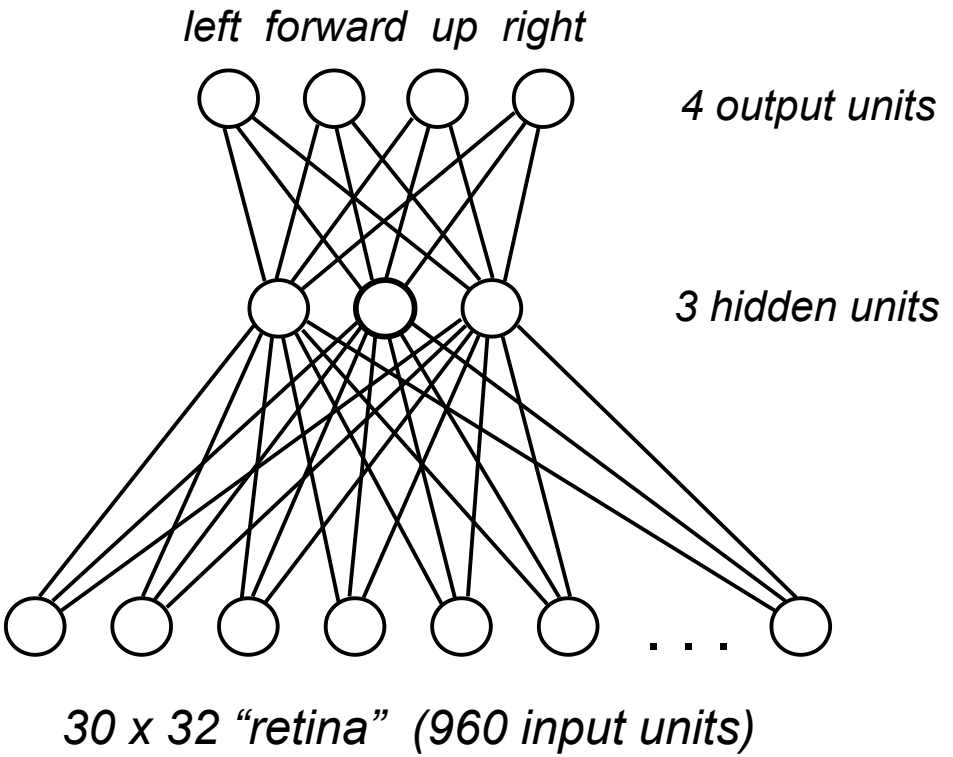
Recognizing Sunglasses



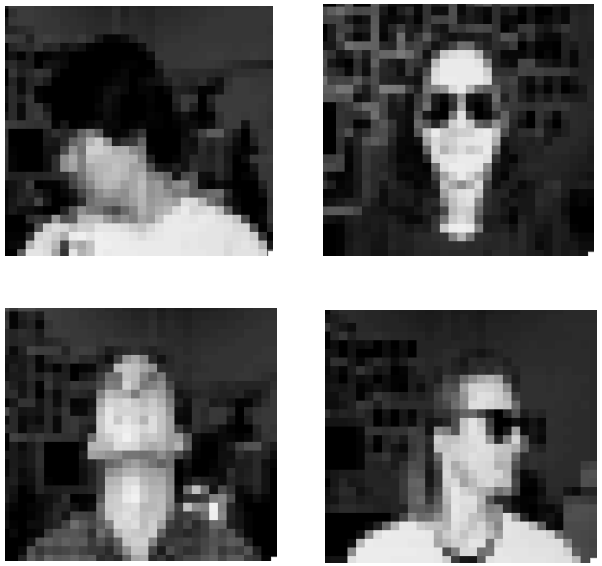
Sunglasses Recognizer Demo

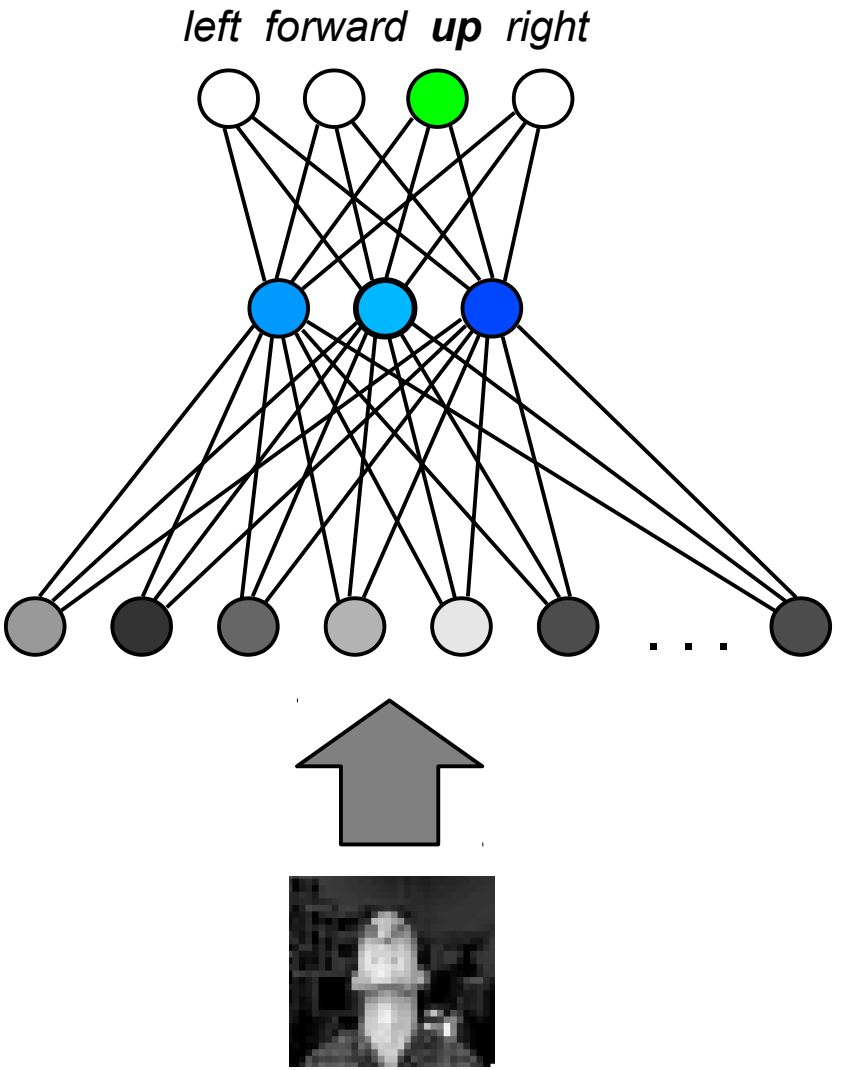
Recognizing Poses



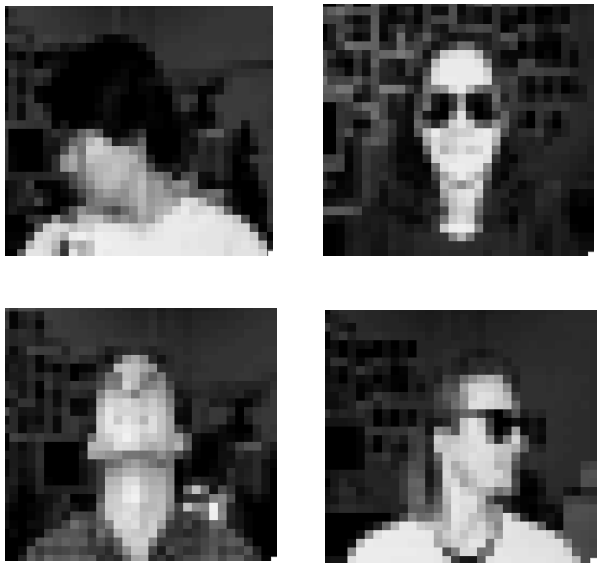


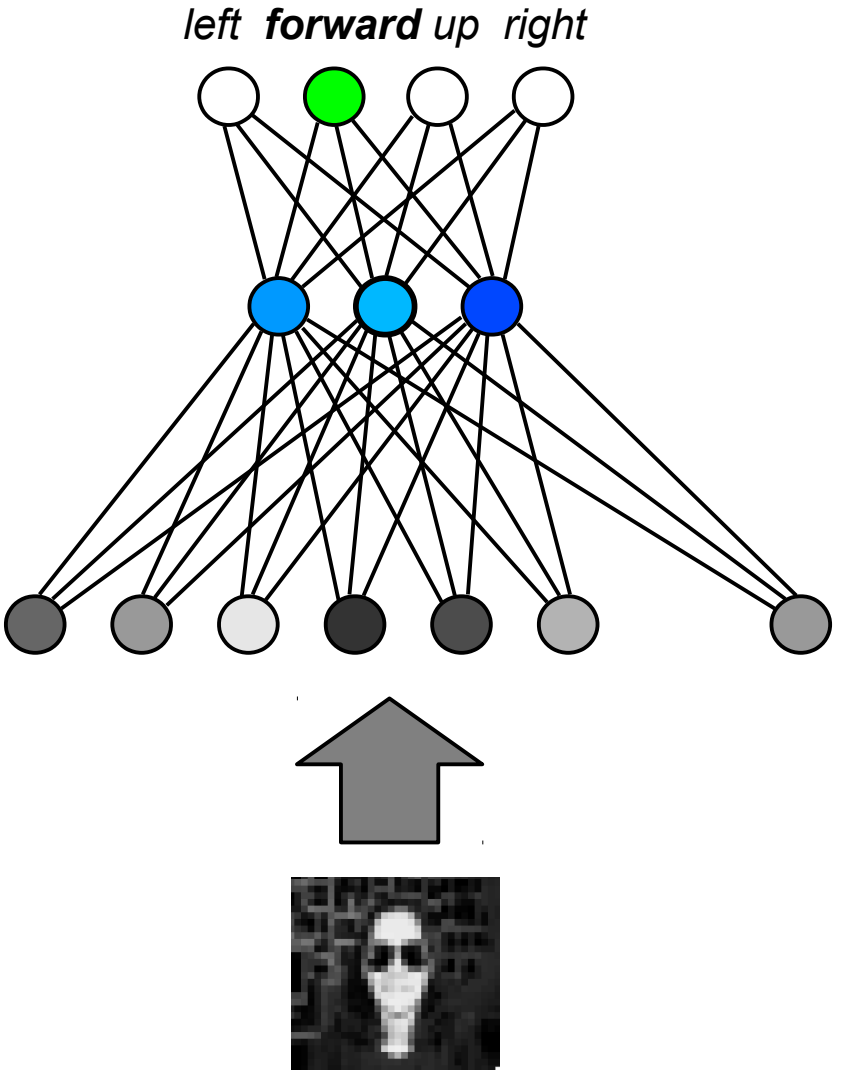
Recognizing Poses





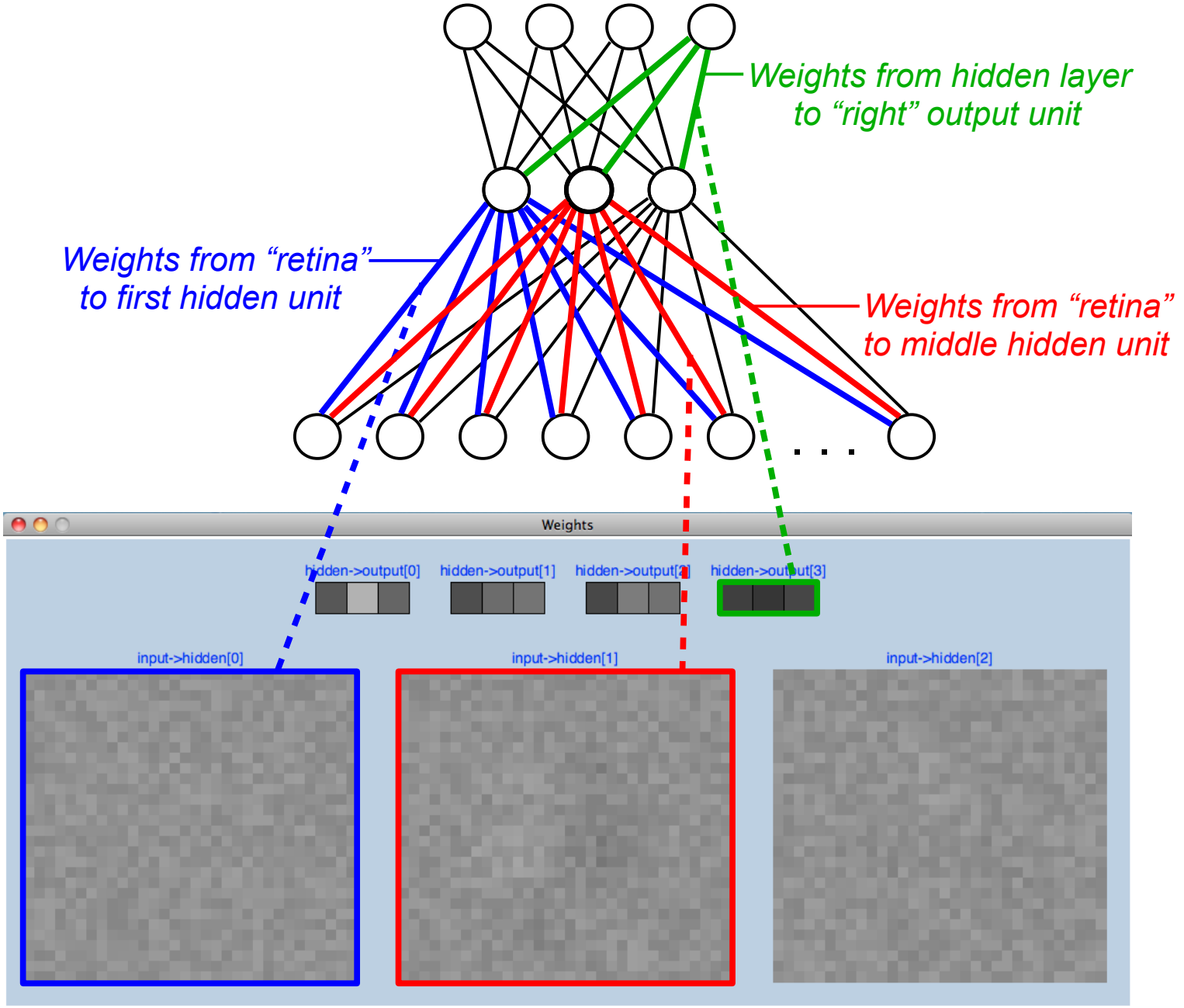
Recognizing Poses





Pose Recognizer Demo

The Knowledge is in the Connection Weights

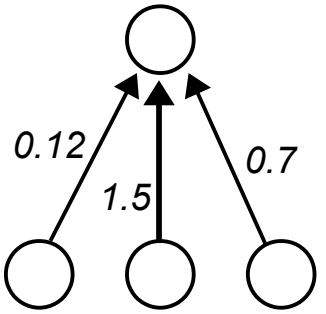


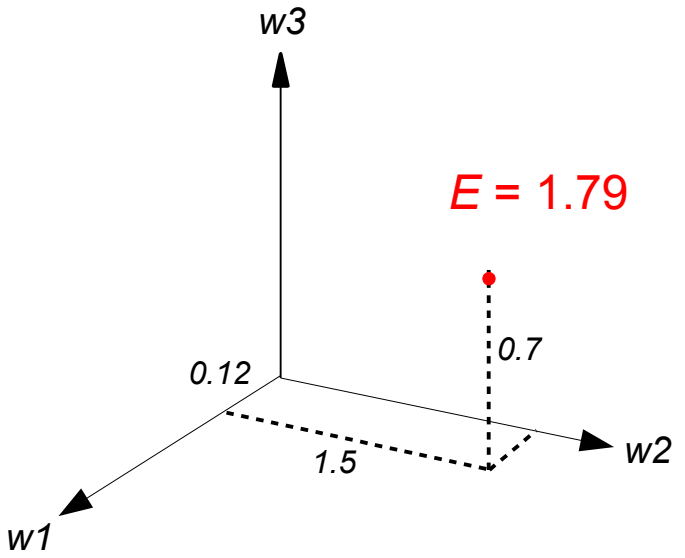
Neural Network Learning

- Connection weights determine network behavior
- Behavior could be “good” or “bad”
- **Error function** quantifies this measure

$$E = (target_1 - output_1)^2 + (target_2 - output_2)^2 + \ldots$$

Input	Target	Output
0 0 0	0	0.50
0 0 1	0	0.67
0 1 0	0	0.82
0 1 1	1	0.90
1 0 0	0	0.53
1 0 1	1	0.69
1 1 0	1	0.83
1 1 1	1	0.91





“Weight space”

Neural Network Learning

- How to change the weights so that *E* goes down?
- **Backpropagation learning algorithm** modifies the weights
- On each time step, the overall error of the network moves "downhill" in the direction of the **gradient**