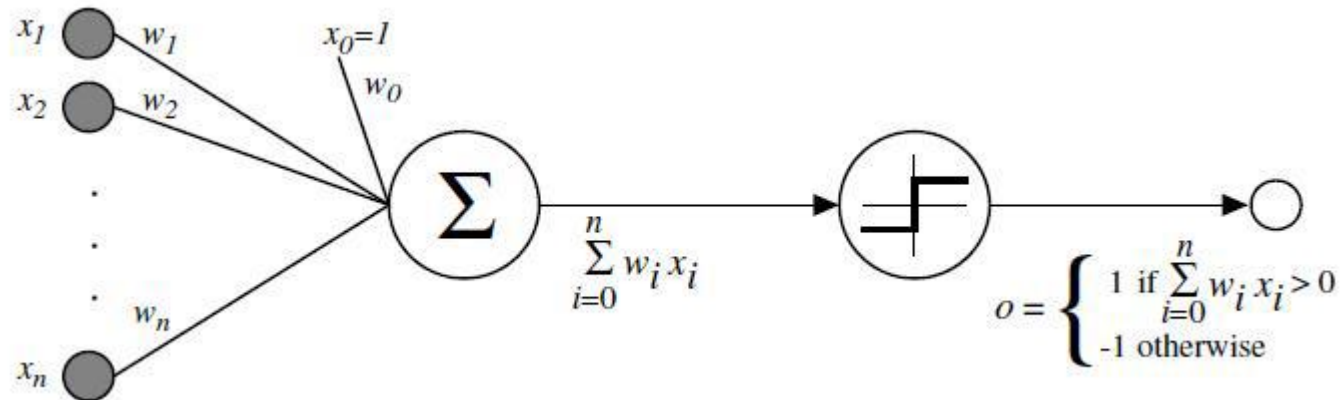


Perceptron to Neural Networks

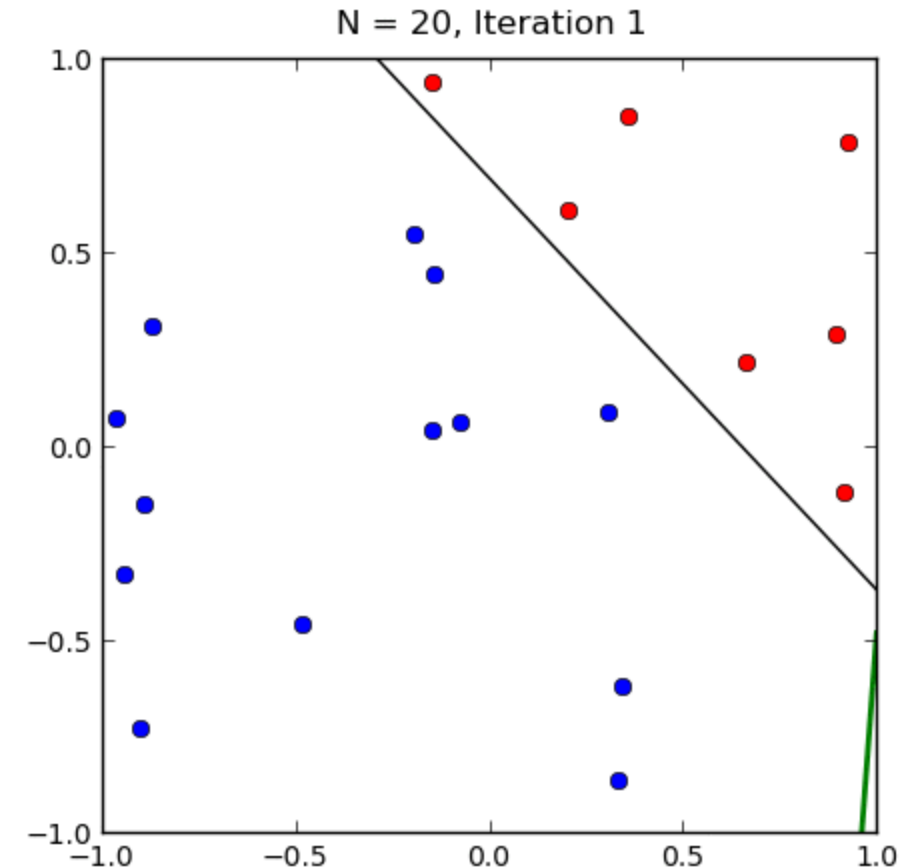
Perceptrons

- Rosenblatt proposed a machine for binary classifications
- Main idea
 - One weight w_i per input x_i
 - Multiply weights with respective inputs and add bias $x_0 = +1$
 - If result larger than threshold return 1, otherwise 0



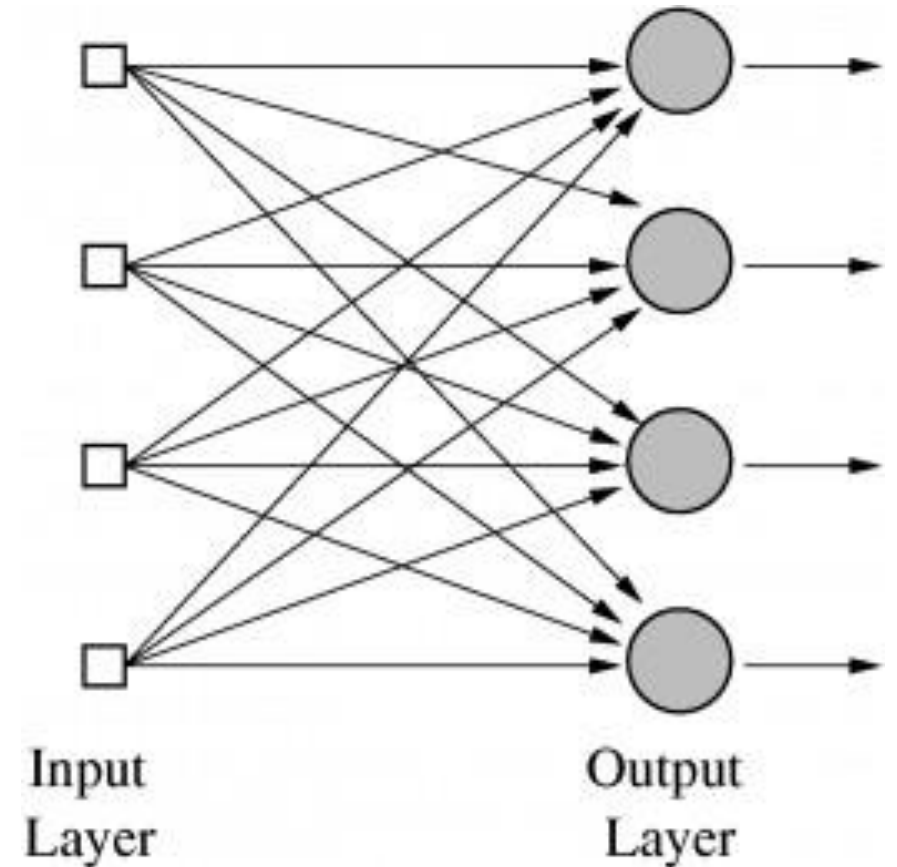
Training a perceptron

- Rosenblatt's innovation was mainly the learning algorithm for perceptrons
- Learning algorithm
 - Initialize weights randomly
 - Take one sample x_i and predict y_i
 - For erroneous predictions update weights
 - If the output was $\hat{y}_i = 0$ and $y_i = 1$, increase weights
 - If the output was $\hat{y}_i = 1$ and $y_i = 0$, decrease weights
 - Repeat until no errors are made

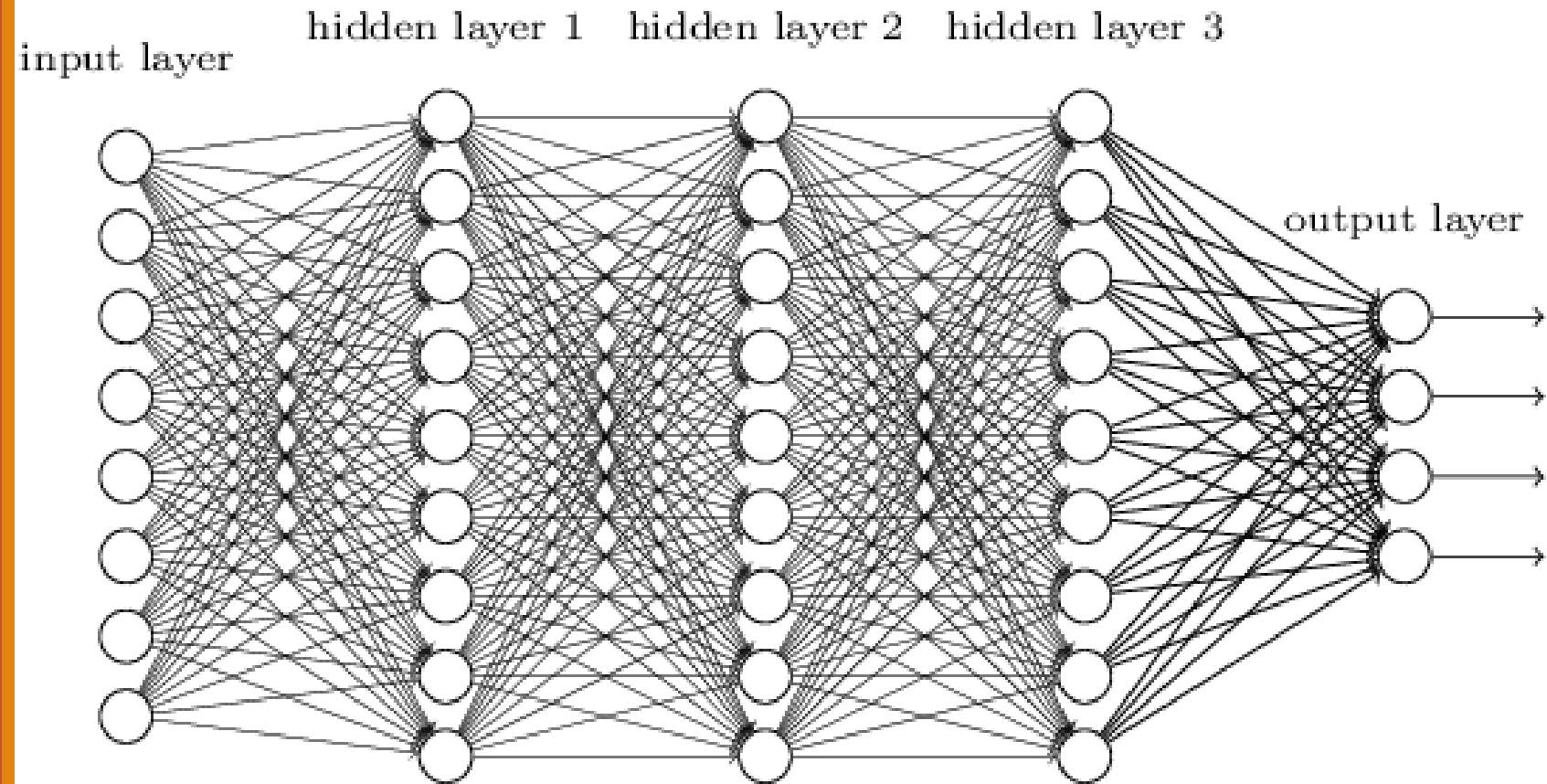


From a perceptron to a neural network

- One perceptron = one decision
- What about multiple decisions?
 - E.g. digit classification
- Stack as many outputs as the possible outcomes into a layer
 - Neural network
- Use one layer as input to the next layer
 - Multi-layer perceptron (MLP)

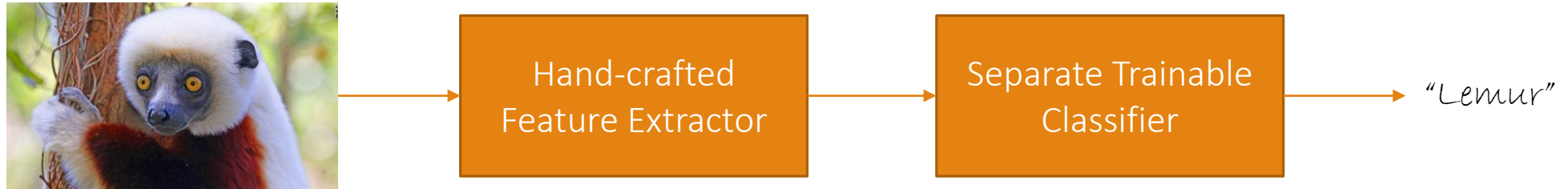


Deep Learning: The *What* and *Why*



Learning Representations & Features

- Traditional pattern recognition



- End-to-end learning → Features are also learned from data

