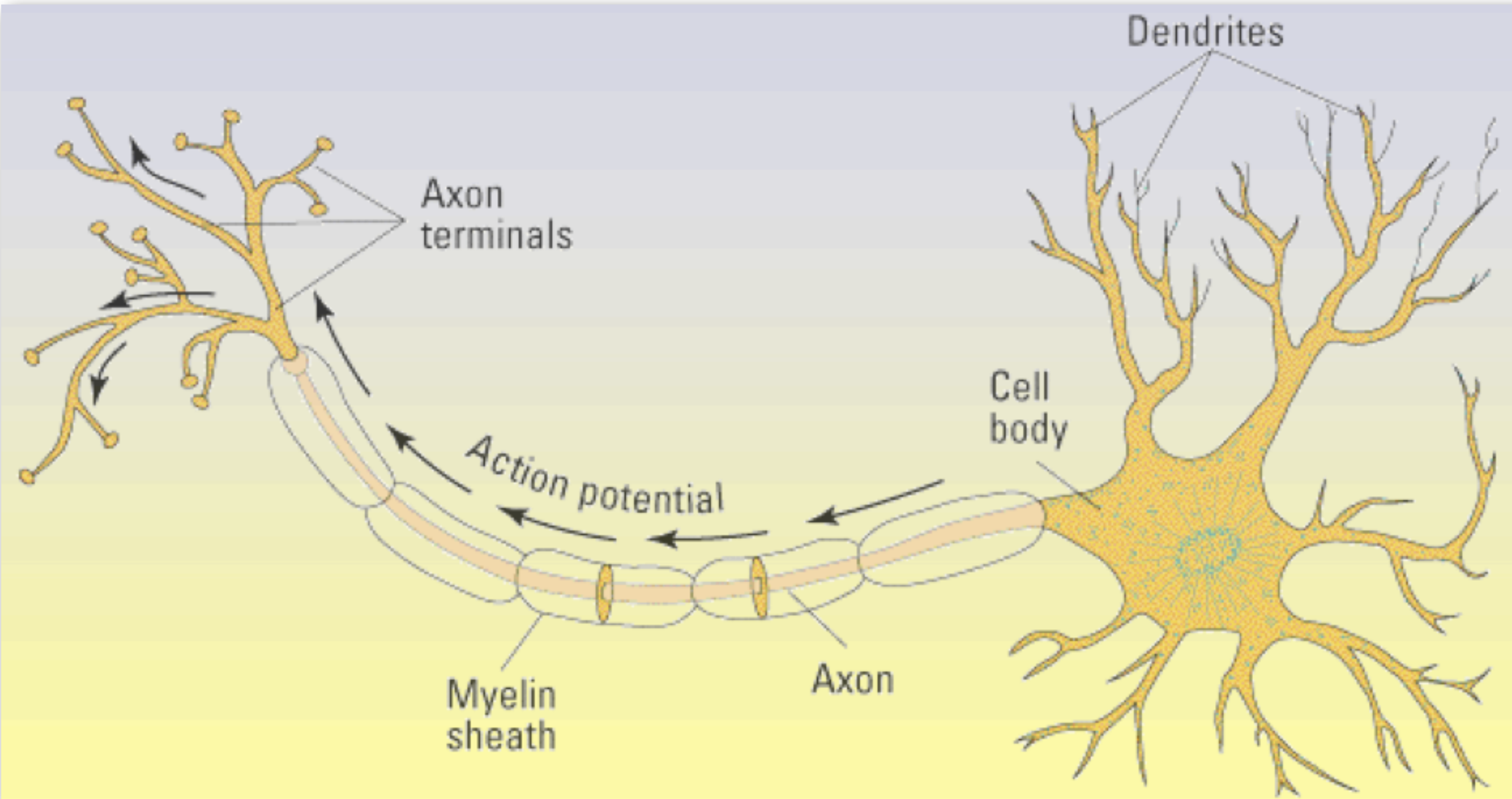
Neural Networks



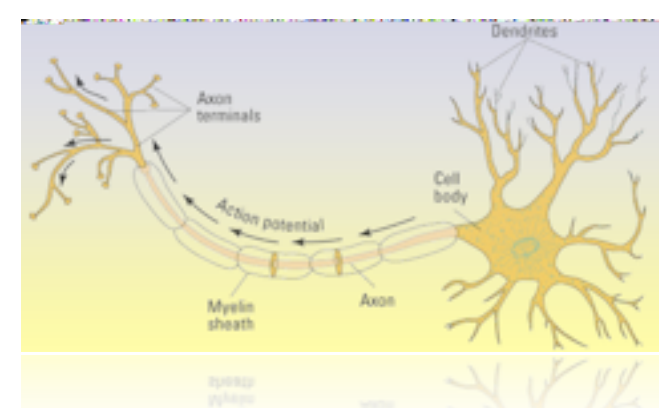
Nonparametric Estimation

Nonparametric estimation is a statistical method that allows the functional form of a fit to data to be obtained in the absence of any guidance or constraints from theory. As a result, the procedures of nonparametric estimation have ***no meaningful associated parameters***. Two types of nonparametric techniques are:

- Artificial neural networks
- Kernel estimation

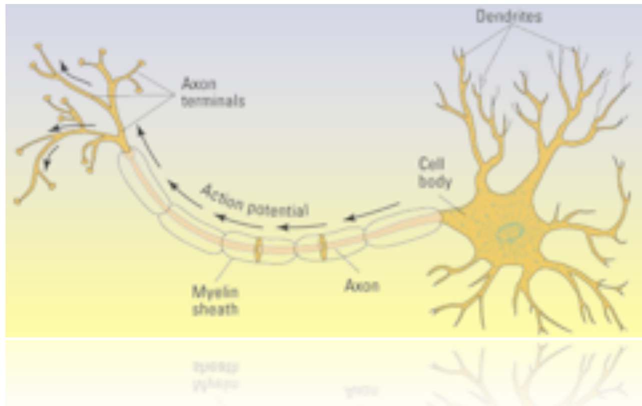
Neural Networks

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the connections between elements largely determine the network function. You can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements.



Neural Networks

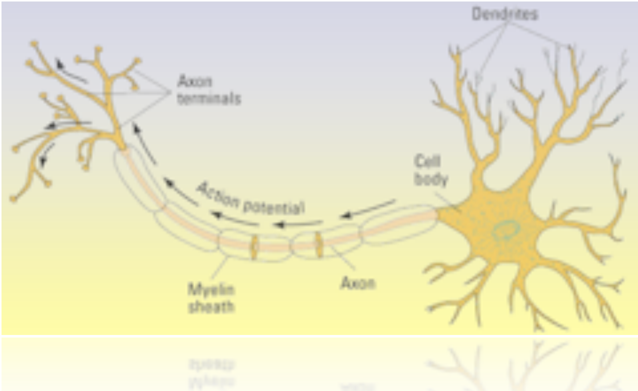
Typically, neural networks are adjusted, or trained, so that a particular input leads to a specific target output. The next figure illustrates such a situation. There, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically, many such input/target pairs are needed to train a network.



Neural Networks

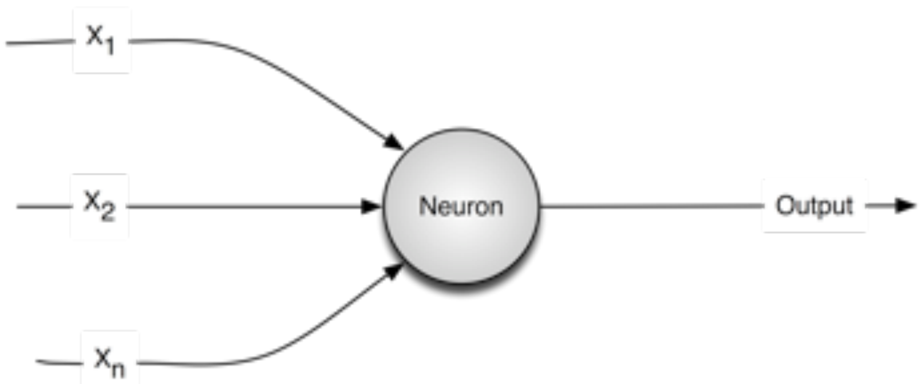
Neural networks have been trained to perform complex functions in various fields, including pattern recognition, identification, classification, speech, vision, and control systems.

Neural networks can also be trained to solve problems that are difficult for conventional computers or human beings.

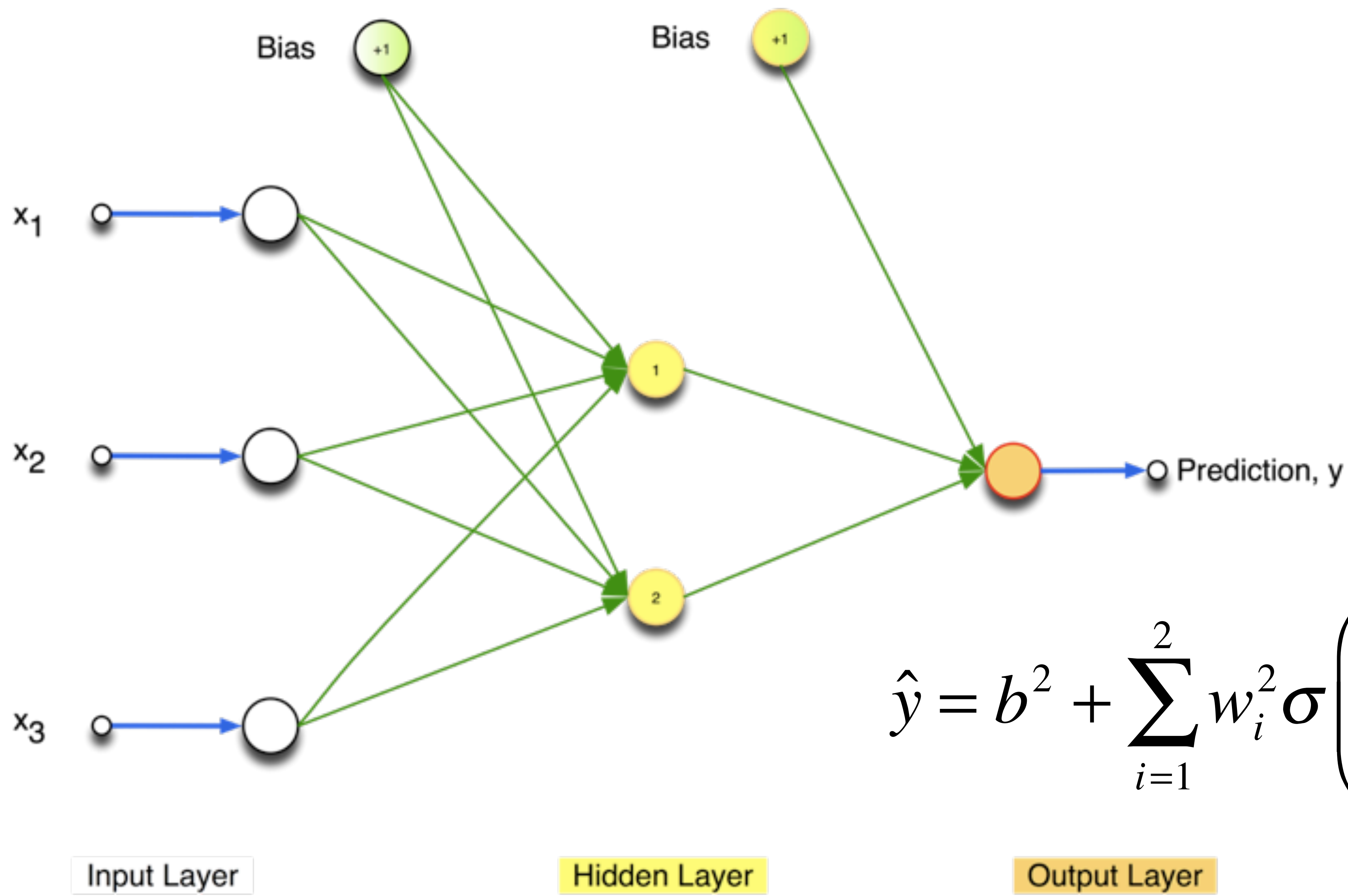


A simple neuron

An artificial neuron is a device with many inputs and one output. The neuron has two modes of operation; the training mode and the using mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not.



A feedforward neural network with three inputs, two hidden neurons, and one output neuron.



$$\hat{y} = b^2 + \sum_{i=1}^{2} w_i^2 \sigma\left(b_i^1 + \sum_{j=1}^{3} w_{i,j}^1 x_j \right)$$

Each arrow corresponds to a real-valued parameter, or a weight, of the network.
The values of these parameters are tuned in the network training.
b are the biases, w are the weights, σ is the activation function.

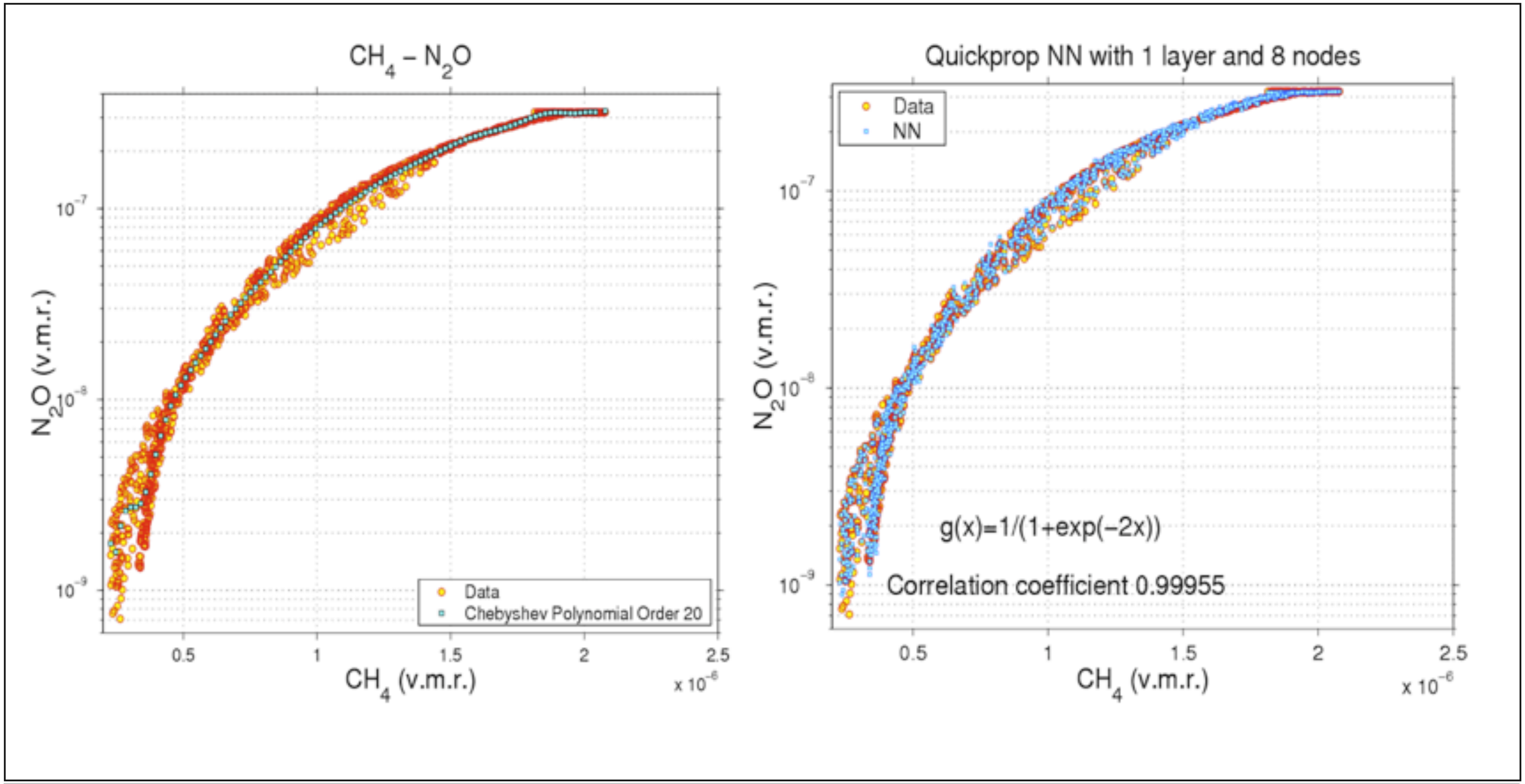
Over-learning and Generalization

Neural network weights are initially random numbers, and then a training algorithm iteratively improves the weights by minimizing the difference between the neural network output and training samples.

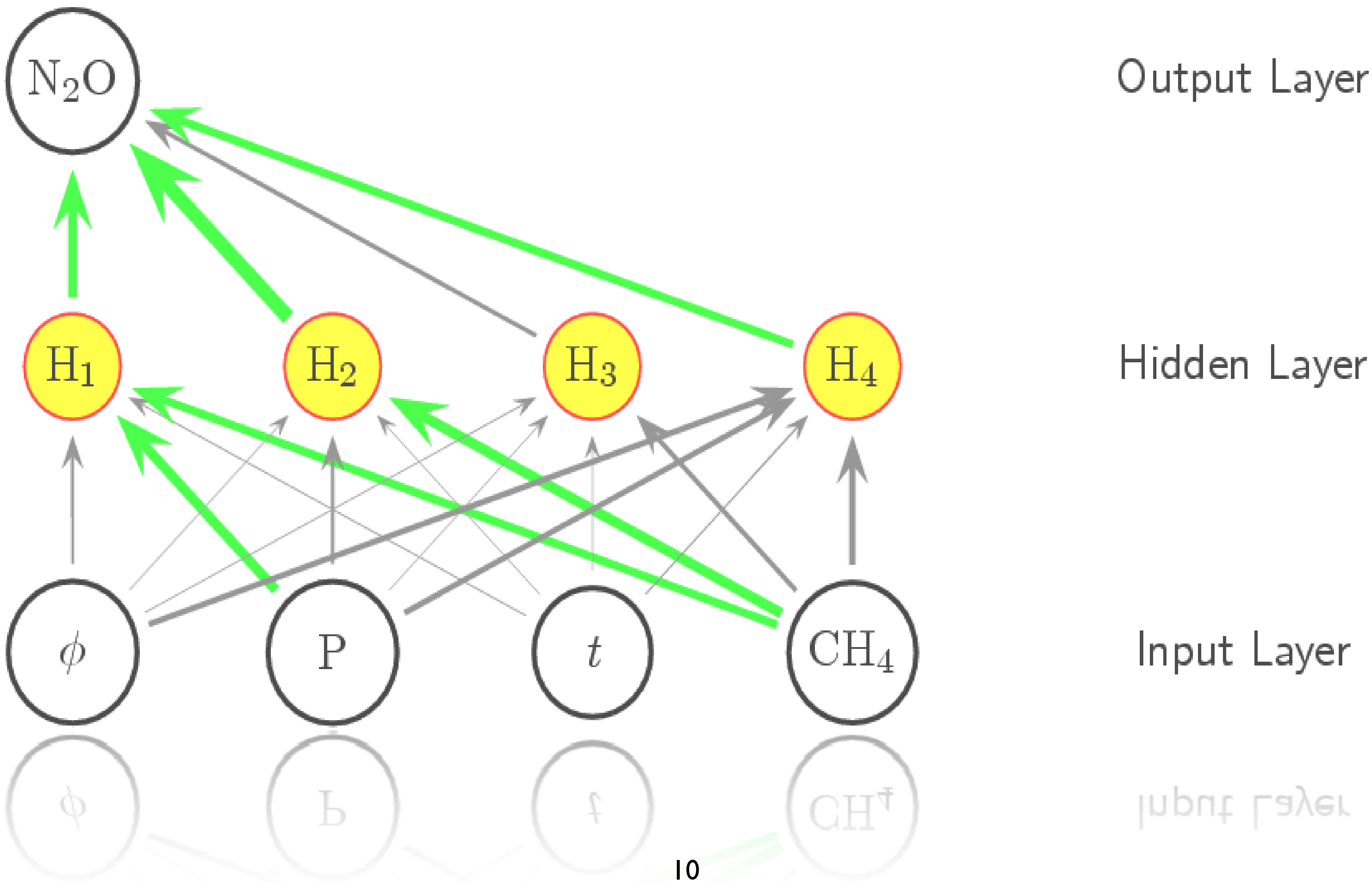
One problem can be that we don't actually minimize the error we are really interested in - which is the expected error the network will make when new cases are submitted to it. In other words, the most desirable property of a network is its ability to ***generalize*** to new cases. In reality, the network is trained to minimize the error on the training set, and short of having a perfect and infinitely large training set, this is not the same thing as minimizing the error on the real error surface - the error surface of the underlying and unknown model (see Bishop, 1995).

The most important manifestation of this distinction is the problem of over-learning, or over-fitting.

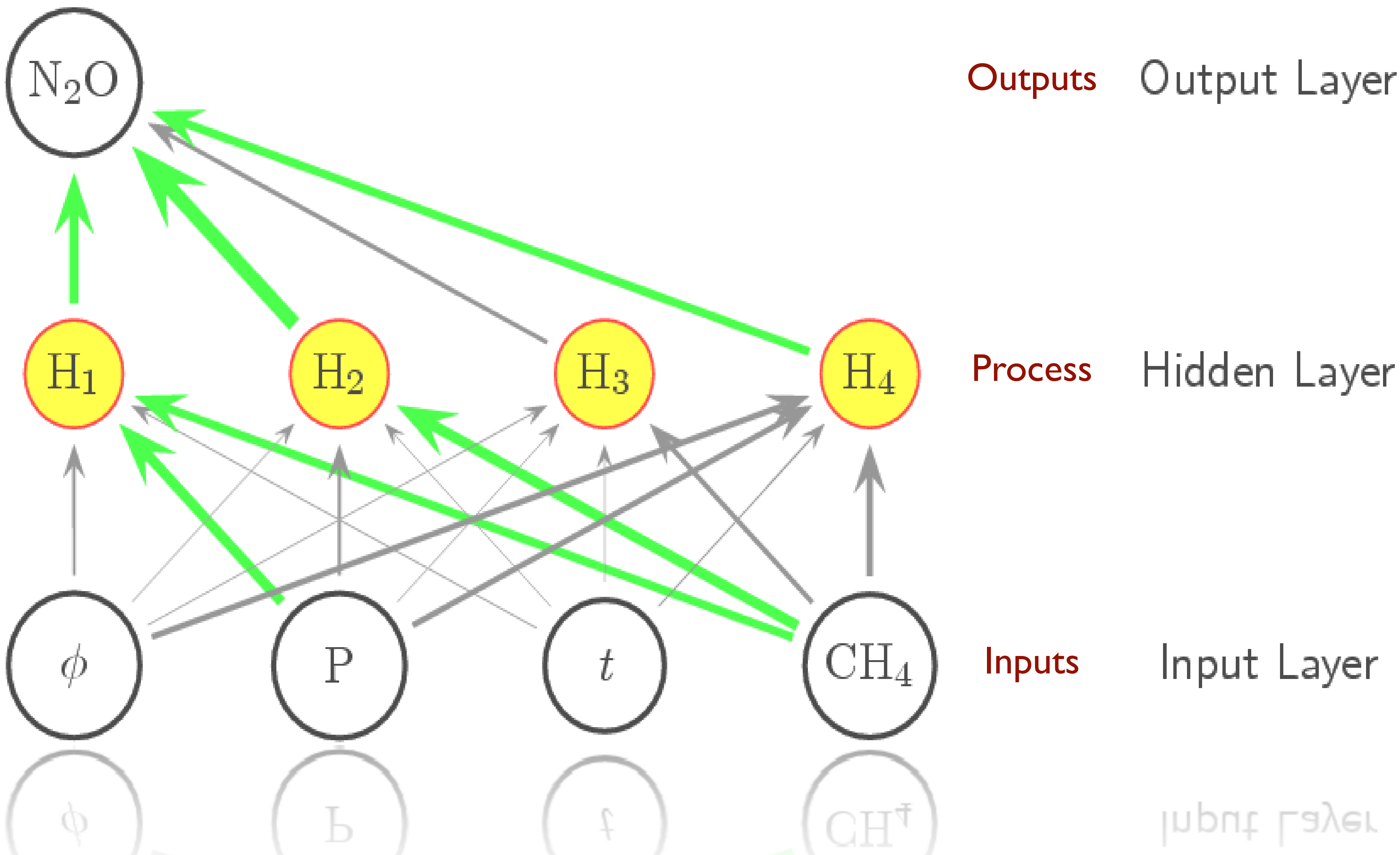
An example use of neural networks



An example neural network



An example neural network



Recommended Textbooks

- Bishop, C. (1995). Neural Networks for Pattern Recognition. Oxford: University Press. Extremely well-written. Requires a good mathematical background, but rewards careful reading, putting neural networks firmly into a statistical context.
- Carling, A. (1992). Introducing Neural Networks. Wilmslow, UK: Sigma Press. A relatively gentle introduction. Starting to show its age a little, but still a good starting point.
- Fausett, L. (1994). Fundamentals of Neural Networks. New York: Prentice Hall. A well-written book, with very detailed worked examples to explain how the algorithms function.
- Haykin, S. (1994). Neural Networks: A Comprehensive Foundation. New York: Macmillan Publishing. A comprehensive book, with an engineering perspective. Requires a good mathematical background, and contains a great deal of background theory.
- Patterson, D. (1996). Artificial Neural Networks. Singapore: Prentice Hall. Good wide-ranging coverage of topics, although less detailed than some other books.
- Ripley, B.D. (1996). Pattern Recognition and Neural Networks. Cambridge University Press. A very good advanced discussion of neural networks, firmly putting them in the wider context of statistical modeling.

Reading Assignment

- Read the Matlab Neural Network documentation sections:
  - Getting Started
  - Neuron Model and Network Architectures

NN Challenge

- Run the script `nntestdata.m` which will create sample test data for performing a NN fit
- In the `nftool` gui that opens set inputs to **p** and the targets to **t**
- Train the NN
- Click on the **performance** and **fit** buttons in the gui that opens
- Generate the associated m code
- Add a NN fit to the interpolation challenge we did in the last class