Neural Networks

Overview

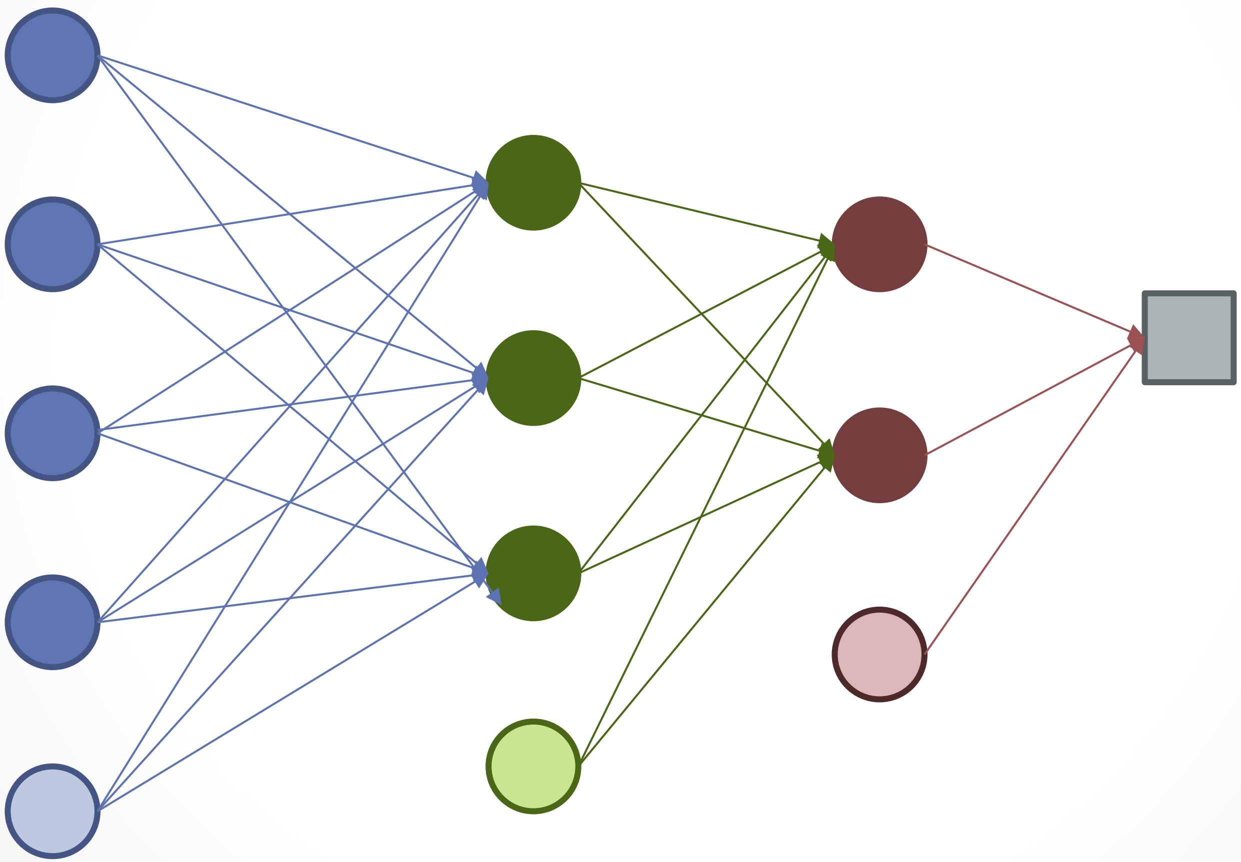
- Neural Networks are considered black-box models
- They are complex and do not provide much insight into variable relationships

- They have the potential to model very complicated patterns (“universal approximators”)
- Can be used for both classification and continuous prediction tasks.

The History

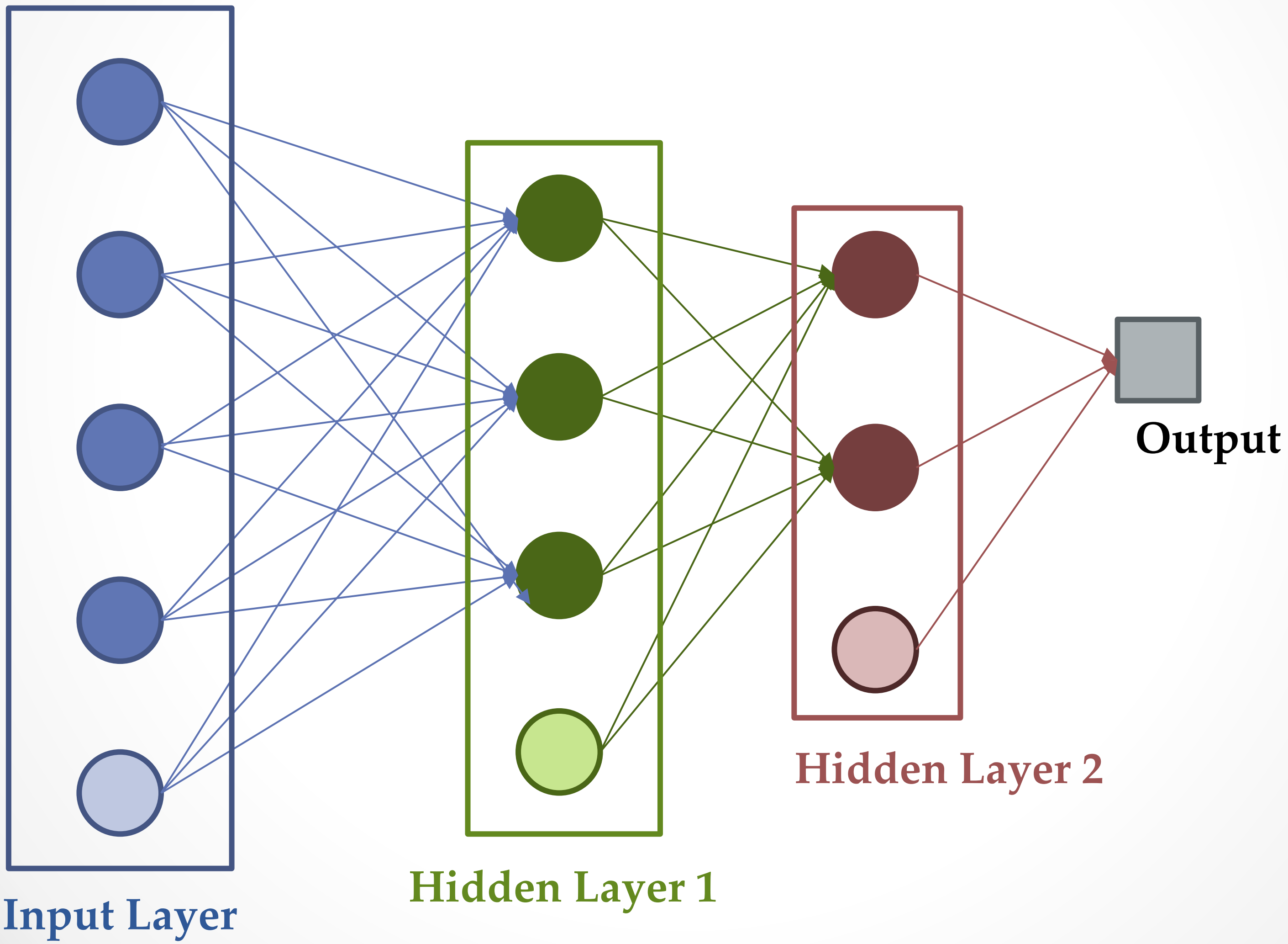
- Concept was welcomed with enthusiasm in 80’s
- Didn’t live up to expectations then
  - Too much hype, perhaps
- Overtaken by other black box techniques like Support Vector Machines with Kernels in 2000’s
- Now in the age of image and visual recognition problems, neural networks have made comeback
  - Area of rapid development
  - Rebranded as “Deep Learning”
    - Recurrent Neural Networks
    - Convolutional Neural Networks
    - Feedforward Neural Networks

The Structure of a Neural Network

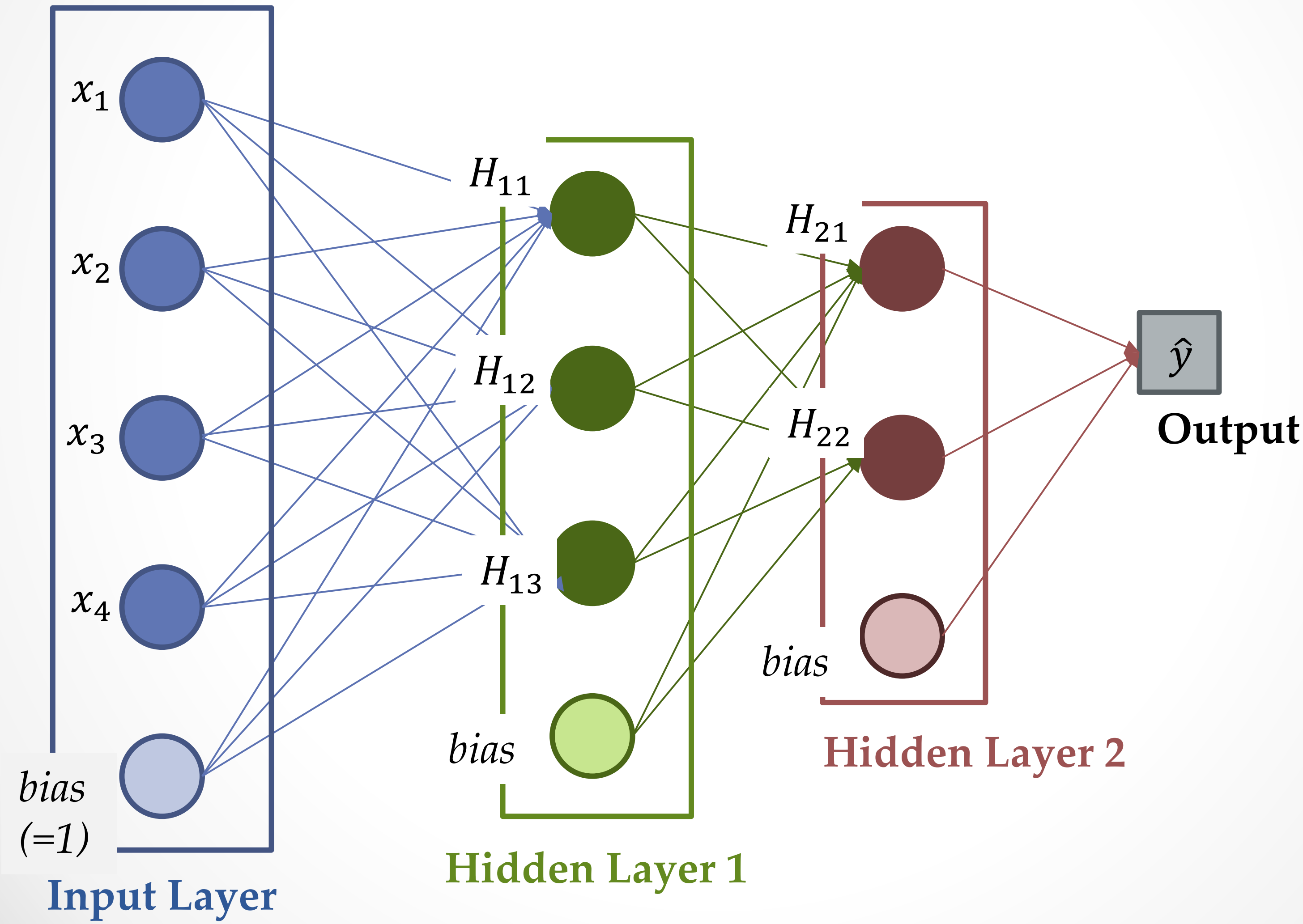


These Neural Networks are often called **Multilayer Perceptrons** (MLPs)

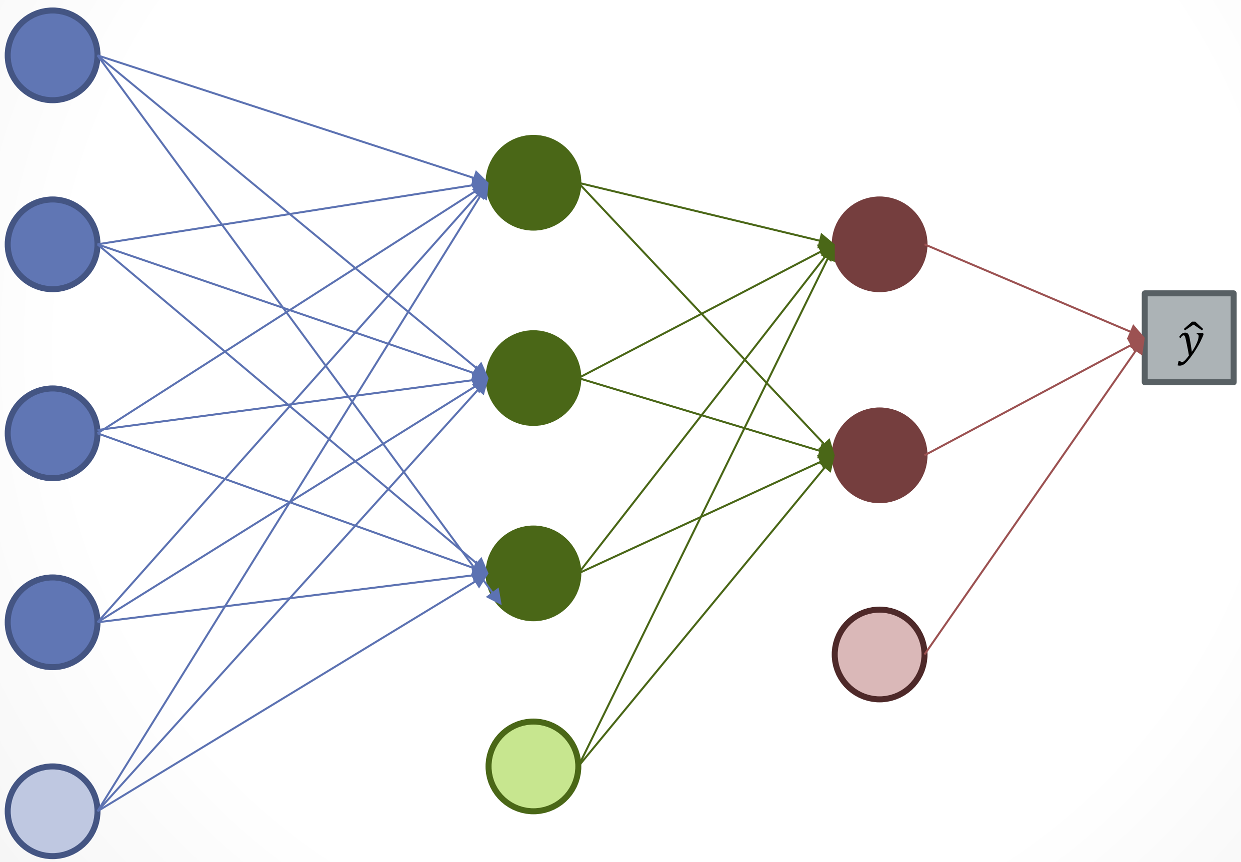
The Structure of a Neural Network



The Structure of a Neural Network

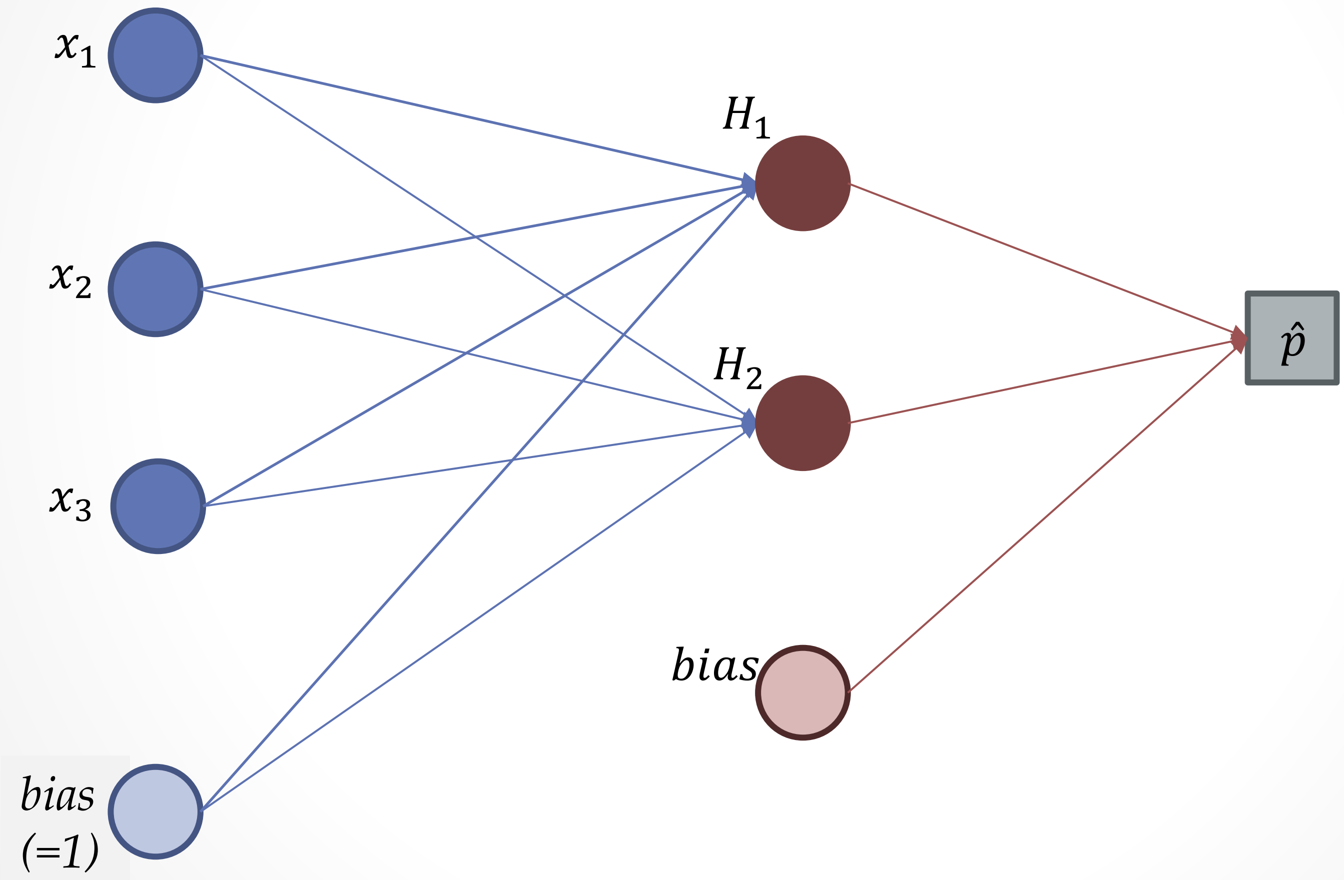


The Structure of a Neural Network



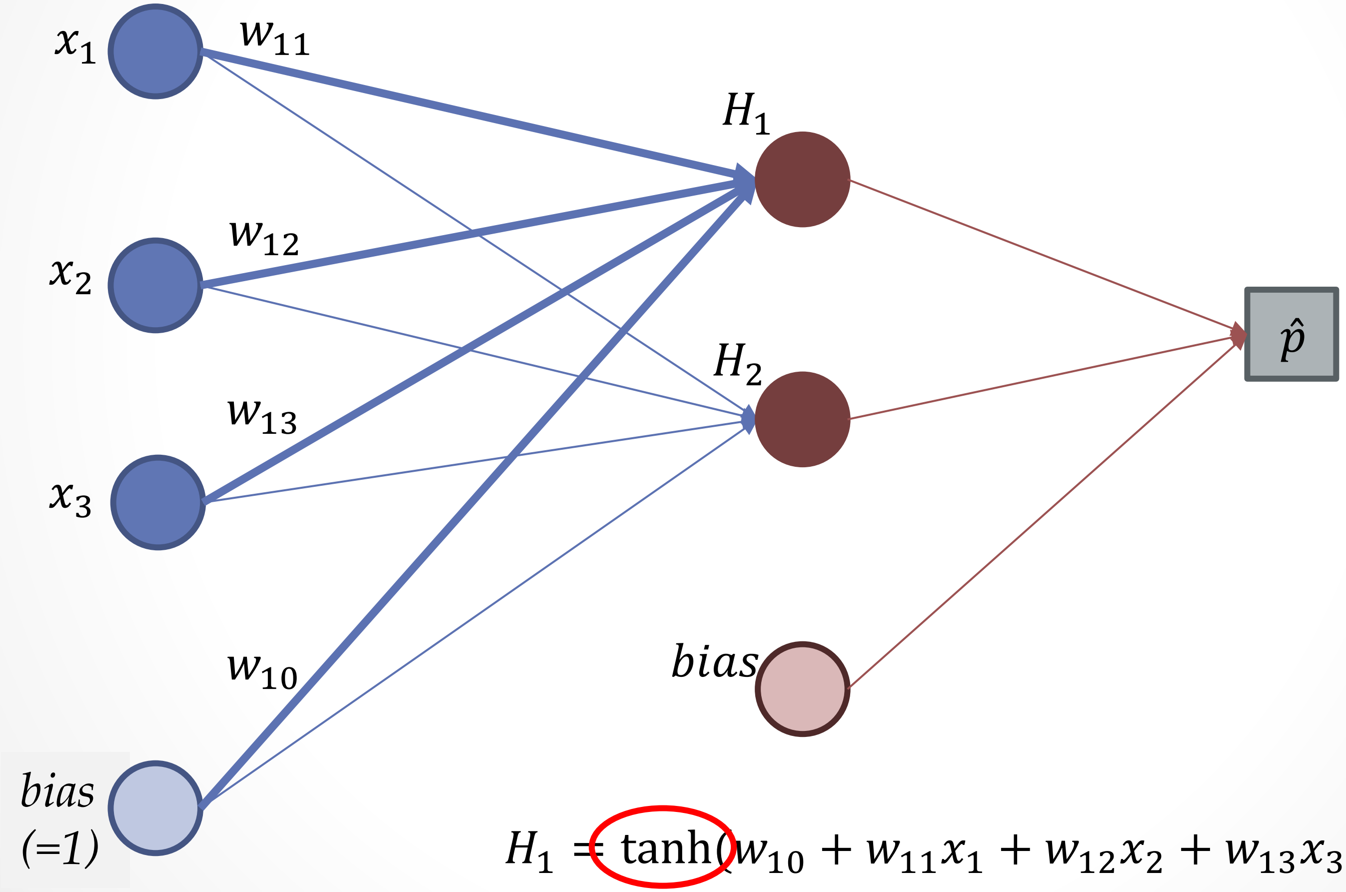
Associated with each line in this diagram is a parameter to be solved for!

A Simpler Neural Network



To avoid triple subscripts, let's simplify our network to 1 hidden layer and just 3 input variables. We'll assume a binary target

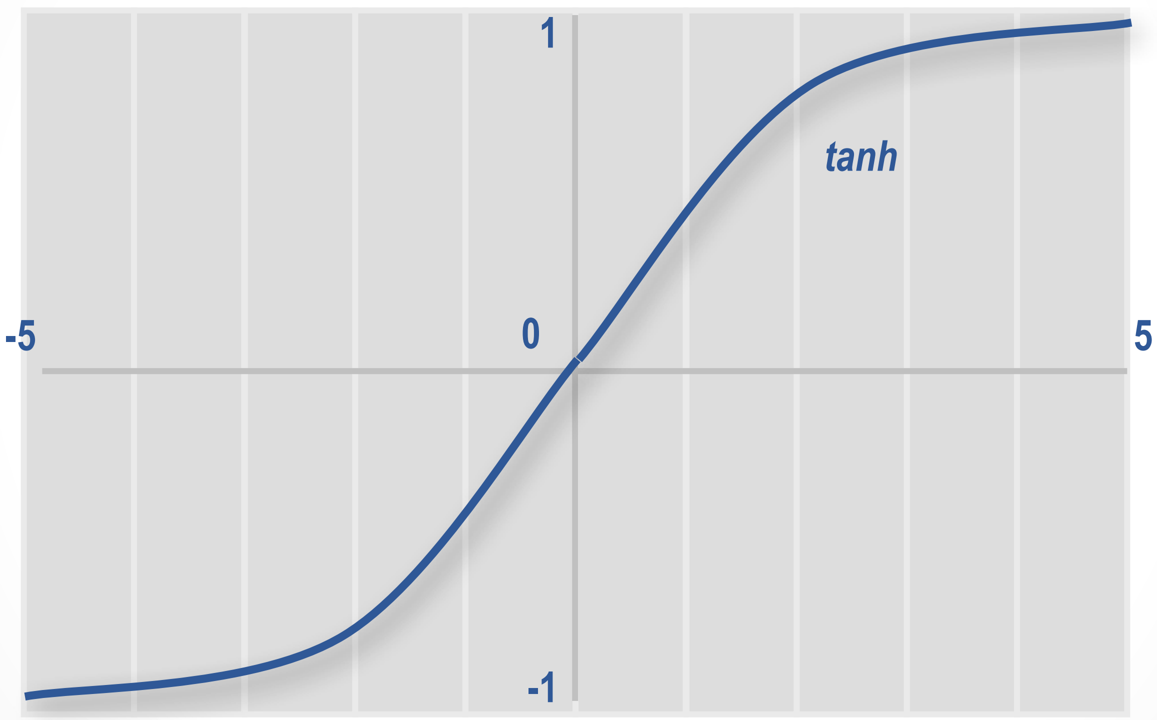
Math Structure of a Neural Network



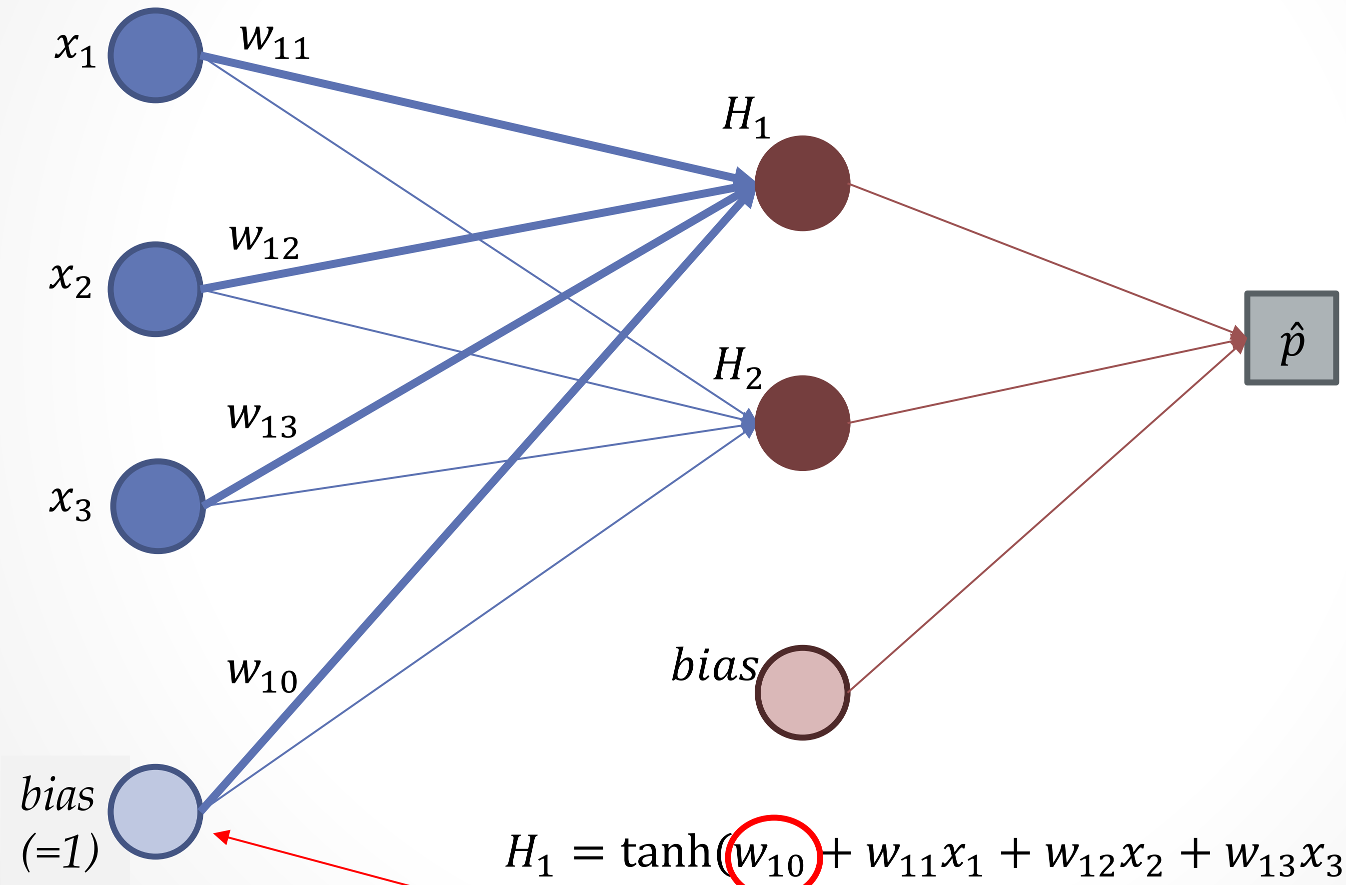
$$H_1 = \tanh(w_{10} + w_{11}x_1 + w_{12}x_2 + w_{13}x_3)$$

Hyperbolic tangent. One of many possible "sigmoid" functions. Range is -1 to 1. Related to logistic function.

Sigmoid Function



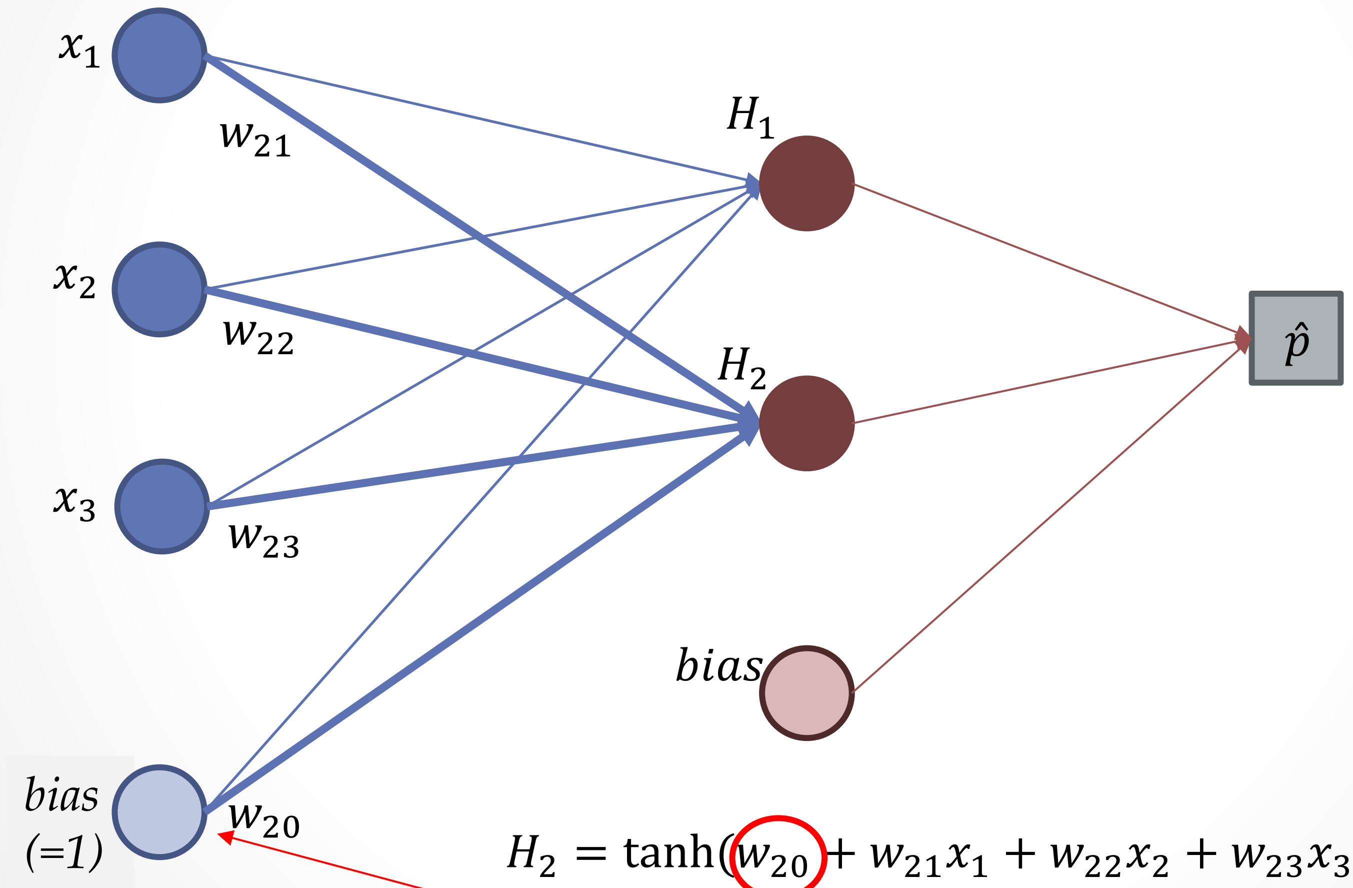
Math Structure of a Neural Network



$$H_1 = \tanh(w_{10} + w_{11}x_1 + w_{12}x_2 + w_{13}x_3)$$

The intercept of each equation called "bias" term

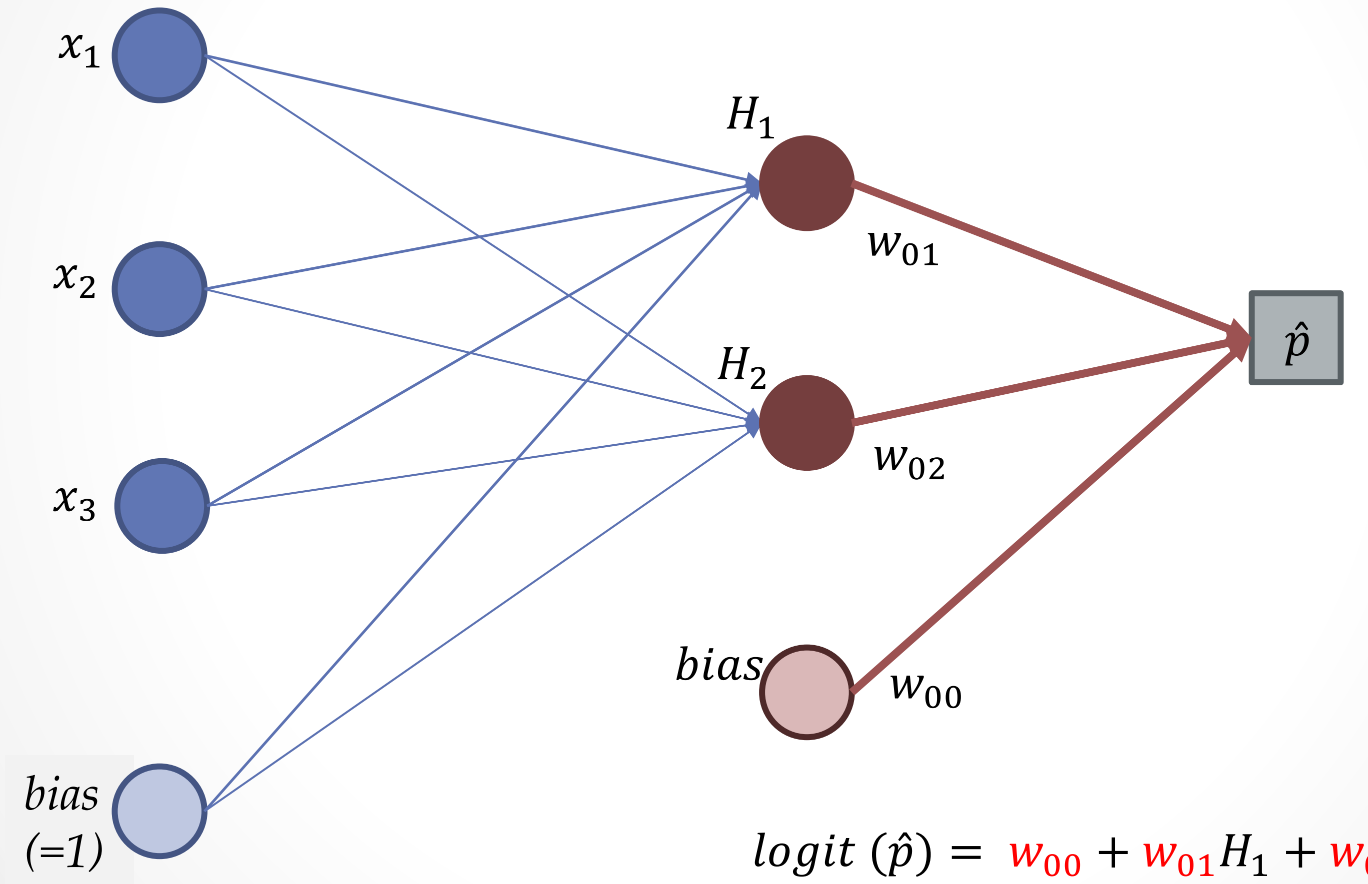
Math Structure of a Neural Network



$$H_2 = \tanh(w_{20} + w_{21}x_1 + w_{22}x_2 + w_{23}x_3)$$

The intercept of each equation called "bias" term

Math Structure of a Neural Network



$$logit\ (\hat{p}) = w_{00} + w_{01}H_1 + w_{02}H_2$$

Math Structure of a Neural Network

- With just 3 input variables and 1 hidden layer containing 2 hidden units, we have to estimate **11** parameters!

$$H_1 = \tanh(w_{10} + w_{11}x_1 + w_{12}x_2 + w_{13}x_3)$$
$$H_2 = \tanh(w_{20} + w_{21}x_1 + w_{22}x_2 + w_{23}x_3)$$
$$logit\ (\hat{p}) = w_{00} + w_{01}H_1 + w_{02}H_2$$

- Weight estimates found by maximizing the log-likelihood function for a class target
- The process involves an algorithm called backpropagation

Math Structure of a Neural Network

- With just 3 input variables and 1 hidden layer containing 2 hidden units, we have to estimate **11** parameters!

$$H_1 = \tanh(w_{10} + w_{11}x_1 + w_{12}x_2 + w_{13}x_3)$$
$$H_2 = \tanh(w_{20} + w_{21}x_1 + w_{22}x_2 + w_{23}x_3)$$
$$logit\ (\hat{p}) = w_{00} + w_{01}H_1 + w_{02}H_2$$

- Probability estimates are obtained by solving the logit equation for *p* for each (x_1, x_2):

$$\hat{p} = \frac{1}{1 + e^{-logit(\hat{p})}}$$

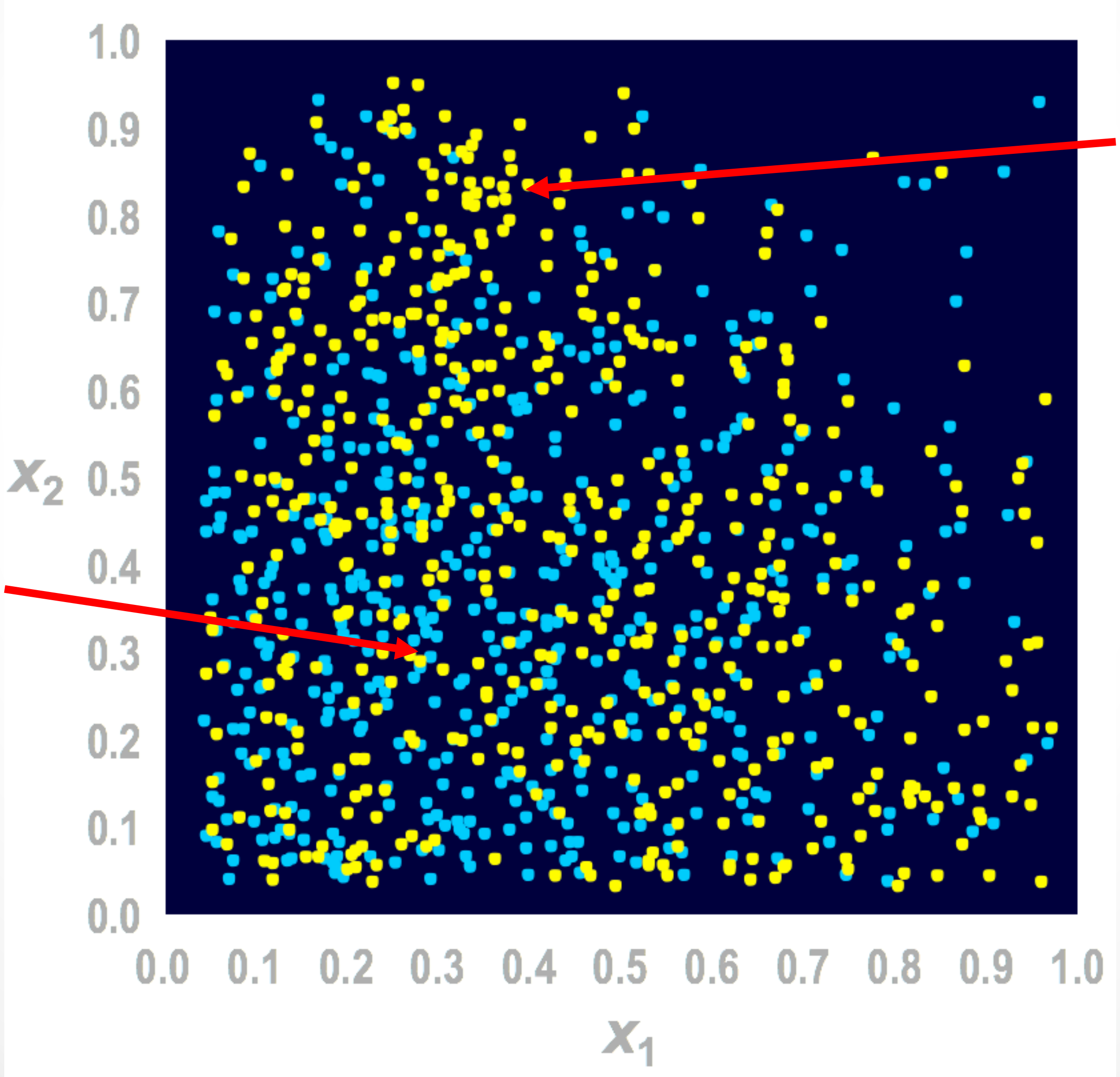
Training a Neural Net (Backpropagation Algorithm)

- **Forward phase**: Starting with some initial weights (often random), the calculations are passed through the network to the output layer where a predicted value is computed.
- **Backward phase**: The predicted value is compared to the actual value and the error is propagated backwards in the network to modify the connection weights.
- Repeat until something like convergence.

Standardization

- Neural Networks work best when input data are scaled to a narrow range around 0
  - For bell shaped data, statistical z-score standardization appropriate
  - For severely non-normal data, range standardization more appropriate.

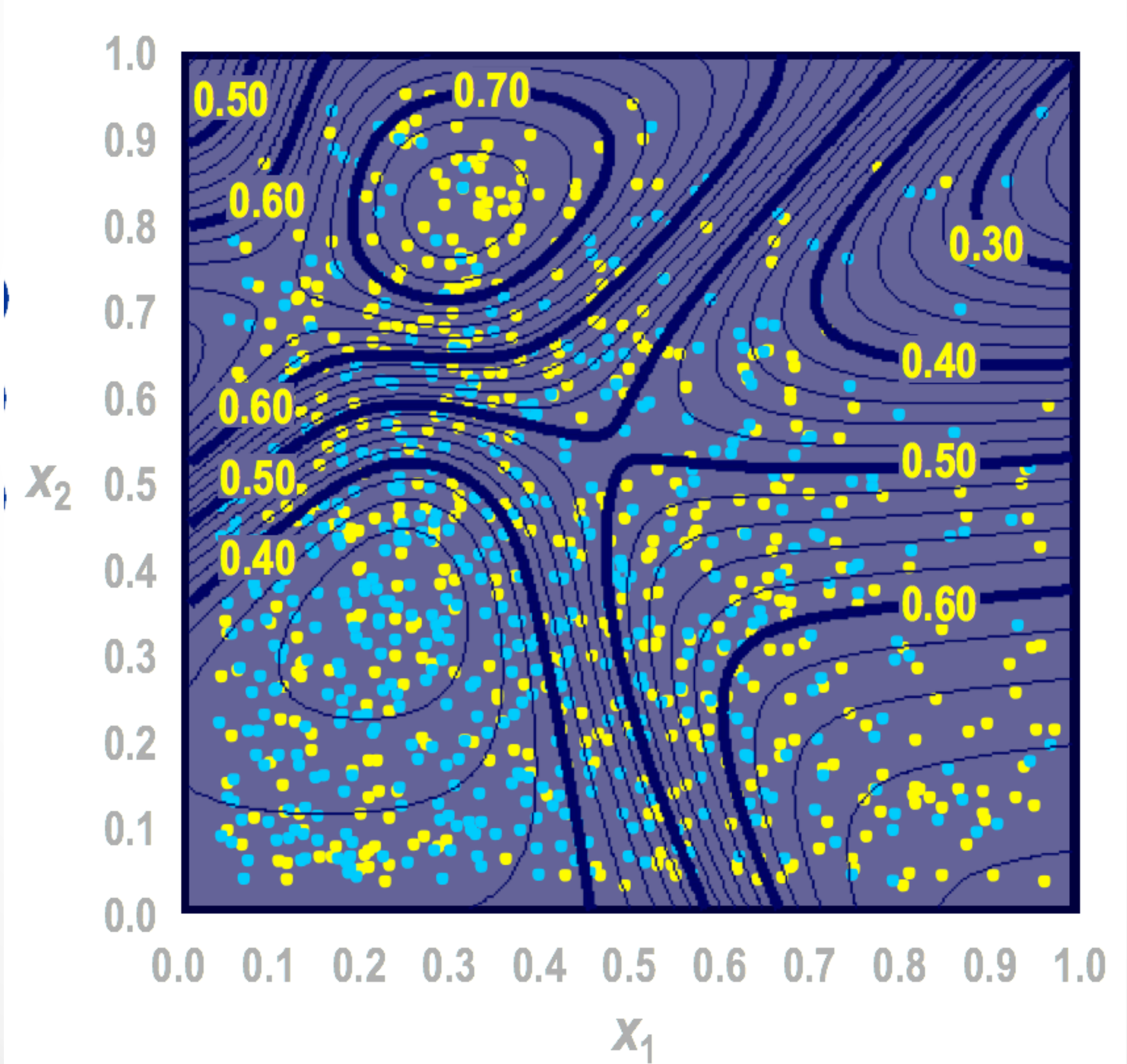
Probability Surface of a Neural Network



High probability of yellow class

Lower probability of yellow class

Probability Surface of a Neural Network



Advantages of a Neural Network

- Can be adapted to classification or numerical prediction problems
- Capable of modelling complex nonlinear patterns
- Makes few assumptions about the data's underlying relationships.

Disadvantages of a Neural Network

- Neural Networks have no mechanism for variable selection. You provide inputs. All inputs are used.
- Very difficult to see the relationships underlying the data.
  - Signs of weights can cancel each other out through the networks
  - Each input gets weight for each hidden unit which then get combined
- Extremely computationally intensive
  - Slow to train
  - Particularly if network structure is complex or number of variables is large
- Prone to overfitting training data

The Big IDEA