

Data Mining & Machine Learning

CS37300
Purdue University

October 20, 2017

Deep Learning (Introduction)

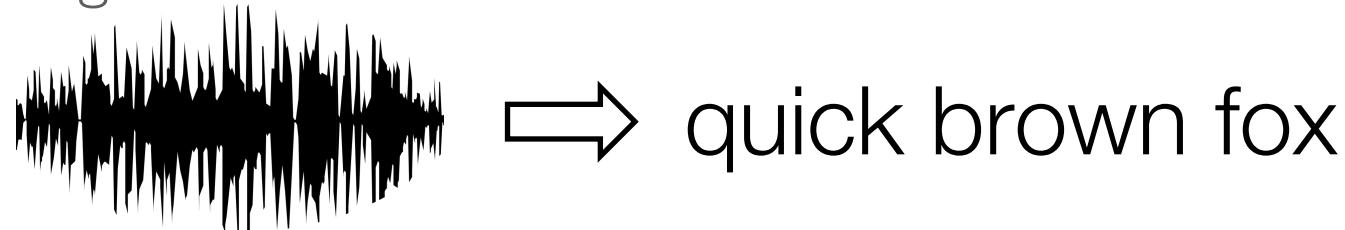
Tasks Deep Learning Excels

- ▶ Image/Video Recognition



Image by Neurala

- ▶ Text Recognition / Prediction
 - ▶ The quick brown fox jumps over the lazy dog
- ▶ Speech Recognition



(CVPR 2016) Face2Face: Real-time Face Capture and Reenactment of RGB Videos, Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, Matthias Nießner

Source Actor



Real-time Reenactment



Reenactment Result



Target Actor

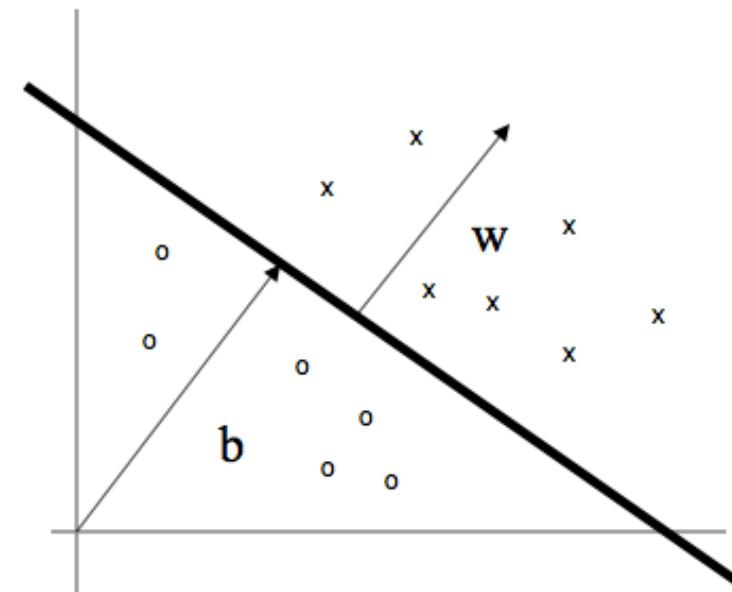
Perceptron Review

Perceptron

Model:

$$f(x) = \sum_{i=1}^m w_i x_i + b$$

$$y = \text{sign}[f(x)]$$



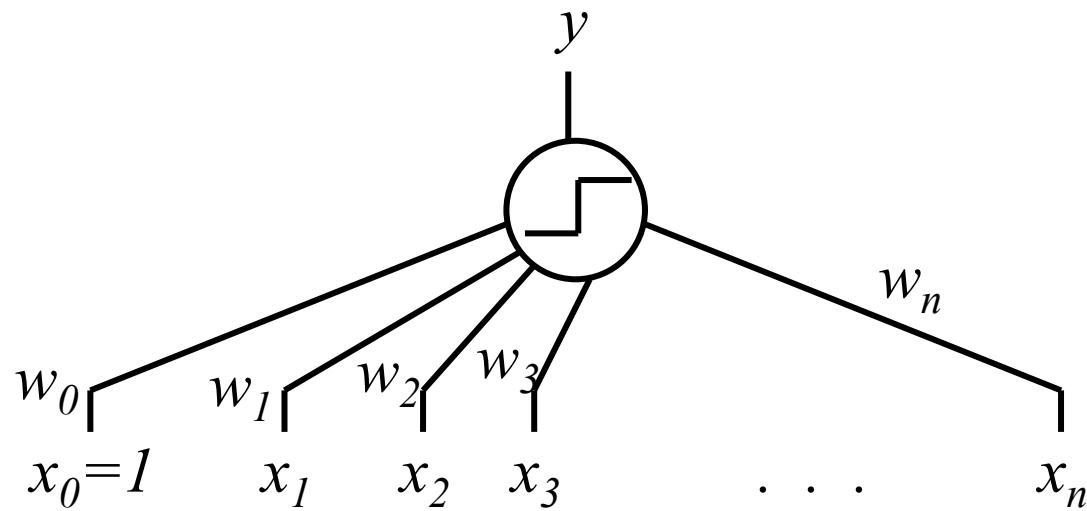
Dot product is product of:

(i) projection of x onto w , and (ii) the length of w

Dot product is 0 if x is perpendicular to w

Add b , if >0 then positive class, if <0 then negative class

Perceptron Structure

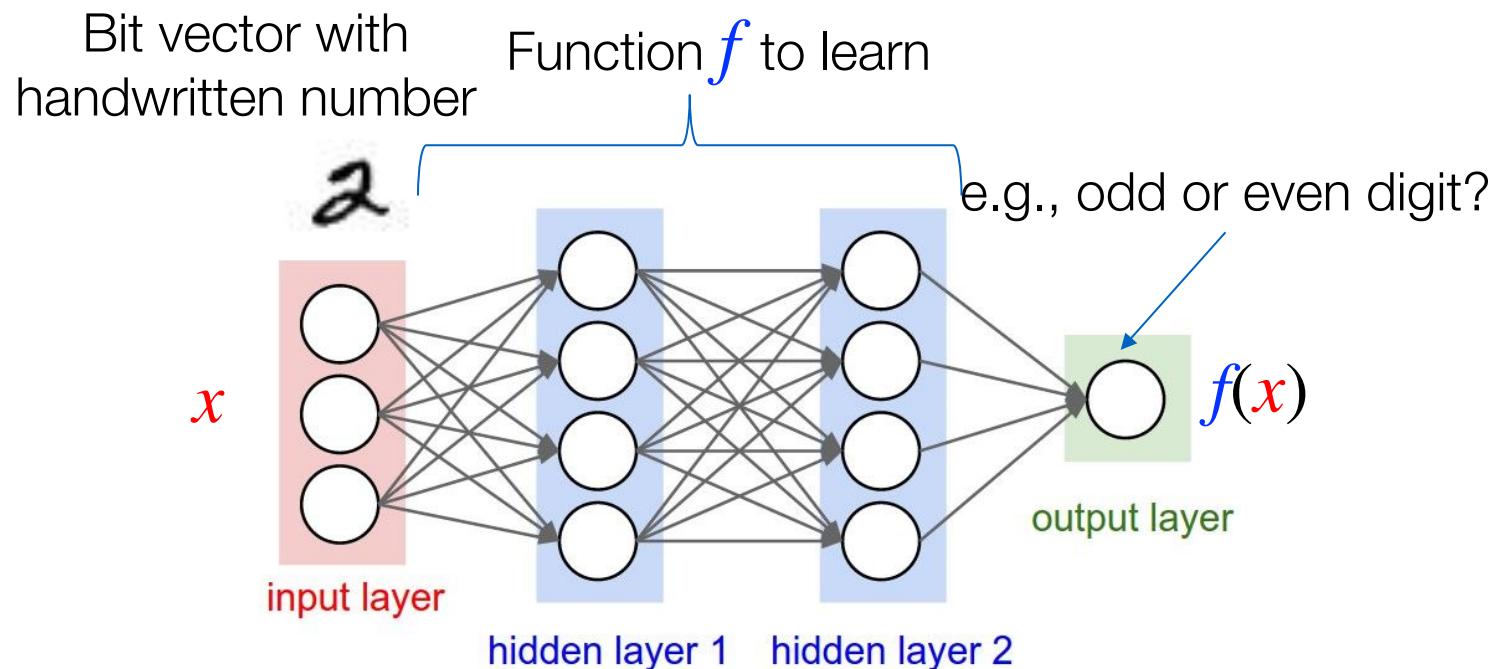


$$y = \begin{cases} 1 & \text{if } \sum_{i=0}^n w_i x_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

$b = x_0 w_0$ compensates for fixed threshold "> 0"

Neural Network

- More general than perceptron
- *Input layer* is a integer or real-valued vector, representing the data
- *Hidden layers* represent a latent (hidden) variables (hard to interpret)
- *Output layer* represents prediction we want to make

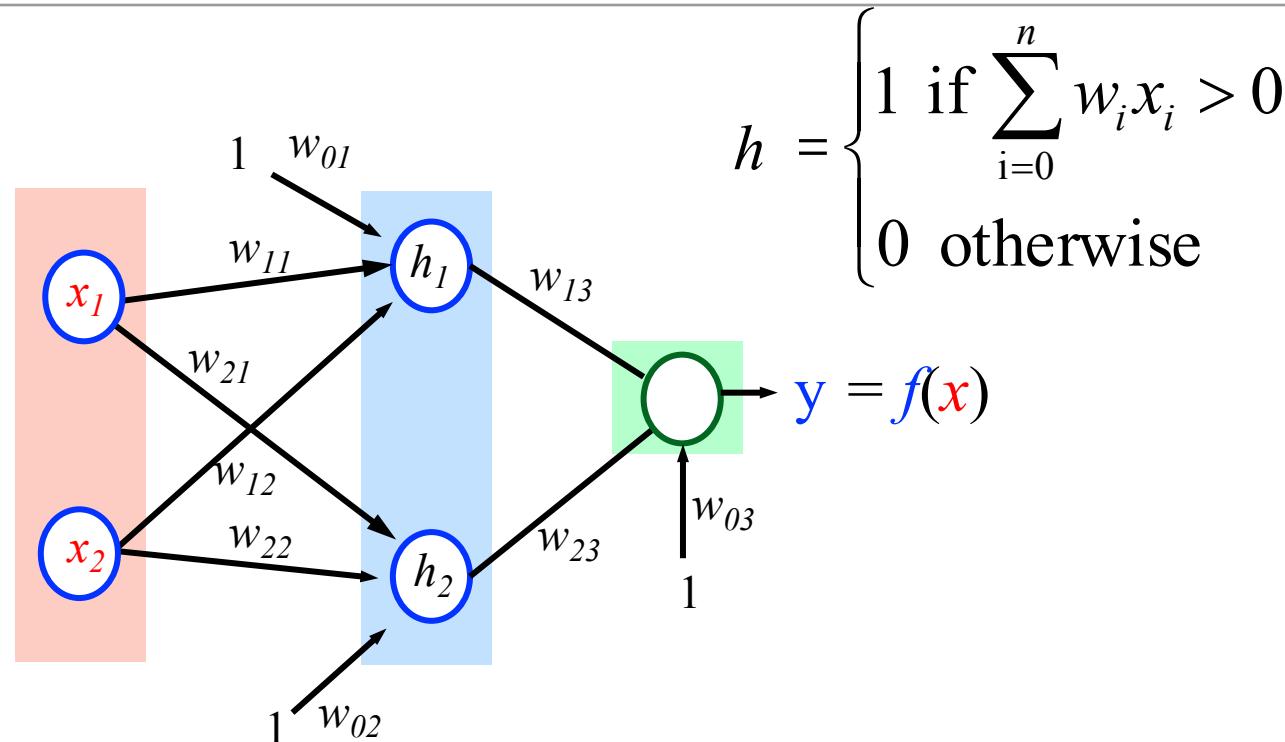


Neural Nets

- Pro: More general than perceptrons
 - Not restricted to linear discriminants
 - Multiple outputs: one classification each
- Con: No simple, guaranteed training procedure
 - Use greedy, hill-climbing procedure to train
 - “Gradient descent”, “Backpropagation”

Example: Solving the XOR Problem

Network
Topology:
2 hidden nodes
1 output



Desired behavior:

x_1	x_2	y
0	0	0
1	0	1
0	1	1
1	1	0

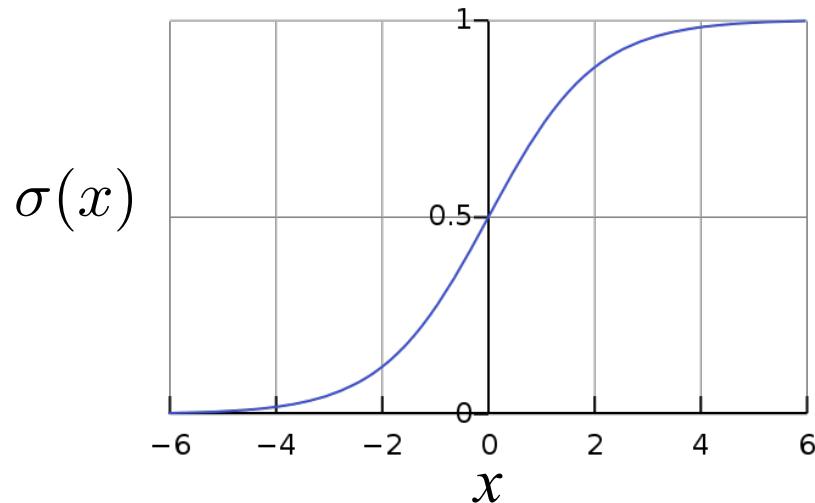
Weights:

- $w_{11} = w_{21} = 1$
- $w_{12} = w_{22} = 1$
- $w_{01} = -3/2; w_{02} = -1/2; w_{03} = 1/2$
- $w_{13} = -1; w_{23} = 1$

More complex examples

Logistic (neuron) Activation (non-linear filter)

- If input is $x = \mathbf{w}^T \mathbf{x}$, the output will look like a probability $\sigma(\mathbf{w}^T \mathbf{x}) \in [0, 1]$
- $p(y = 1 | \mathbf{x}; \mathbf{w}) = \sigma(\mathbf{w}^T \mathbf{x}) \in [0, 1]$

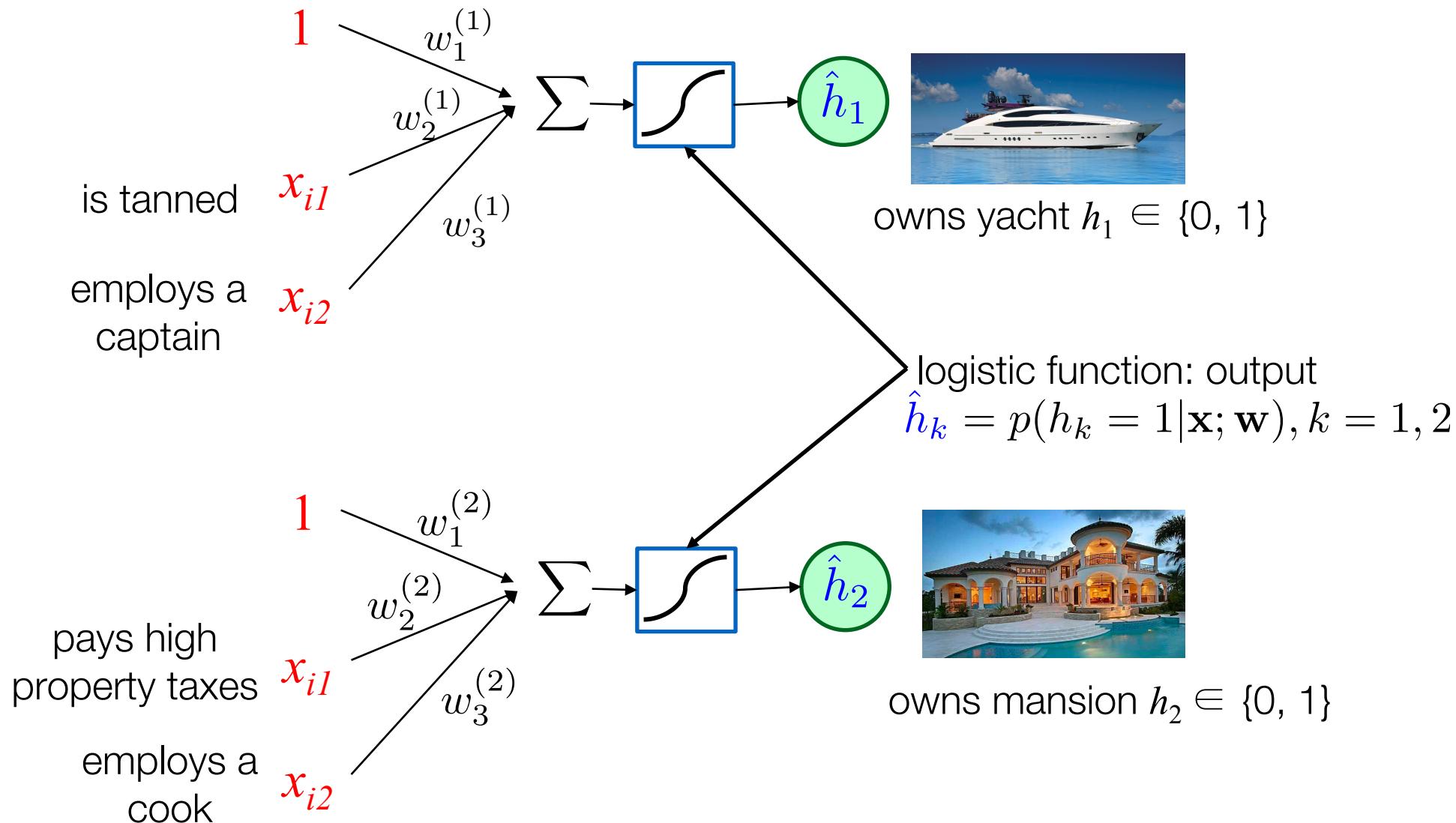


$$\sigma(x) = \frac{\exp(x)}{1 + \exp(x)}$$

- We will represent the logistic function with the symbol:

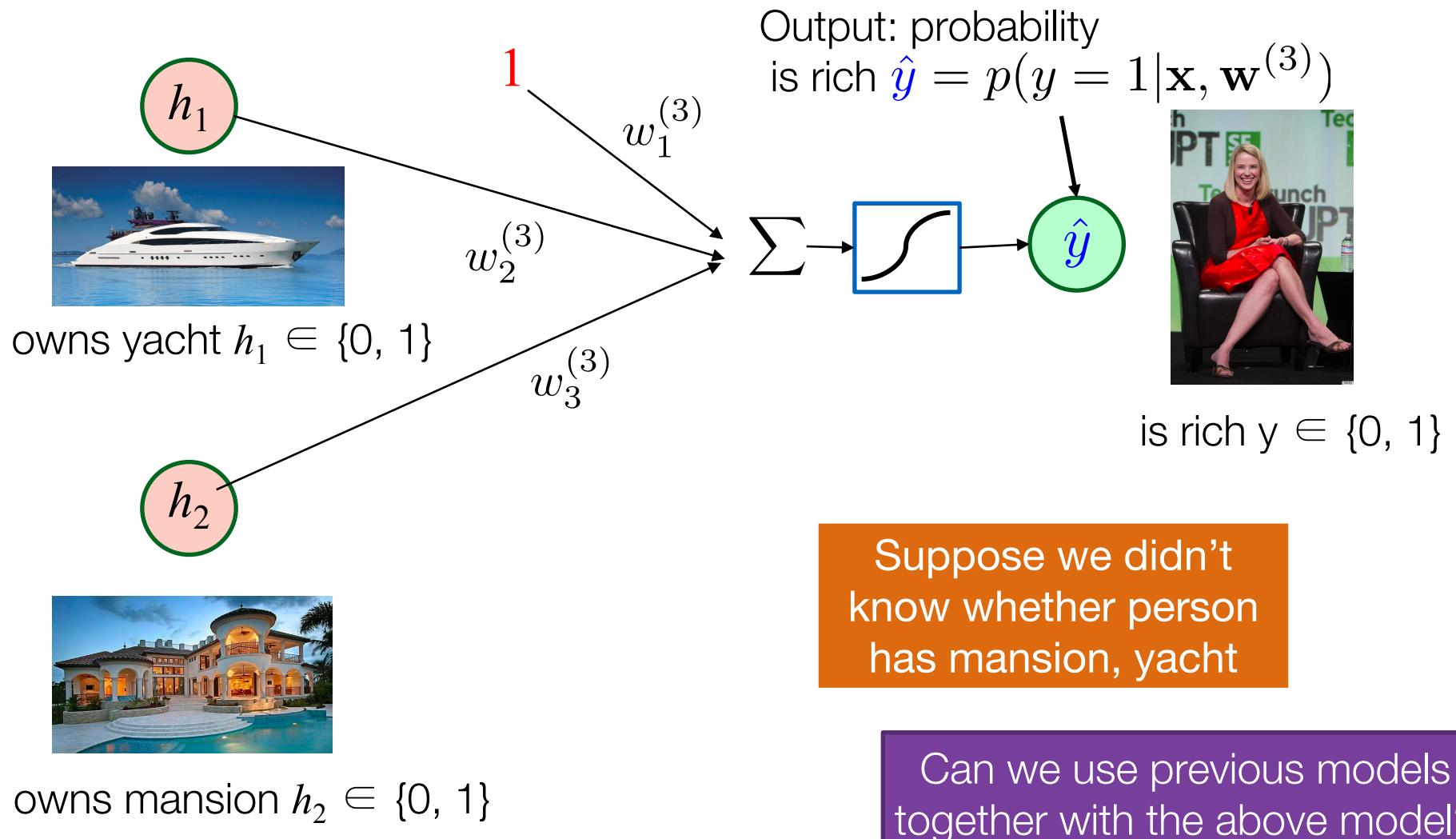


Single Neurons (Layer 1) : Logistic Regression



Top Neuron in Isolation (Layer 2)

- Assume we know these facts: **yacht and mansion ownership**

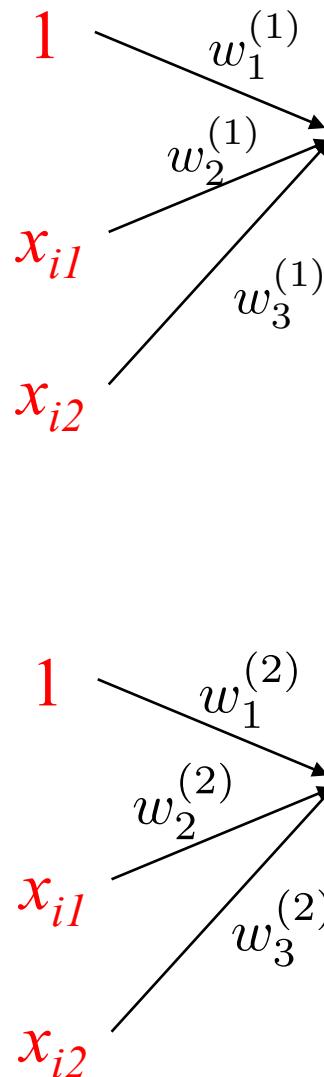


Assume unobserved Intermediate Values (Hidden Layers)

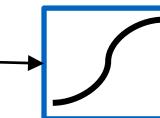
Model M1

is tanned
employs a captain

pays high property taxes
employs a cook



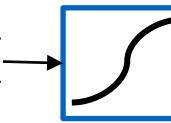
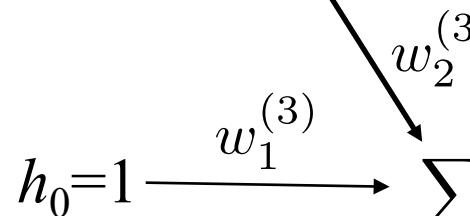
\sum



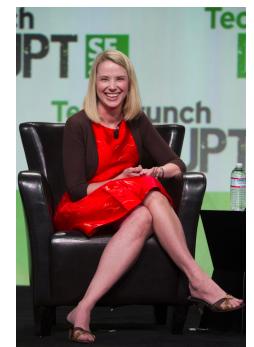
\hat{h}_1



(hidden) owns yacht:
probability $\in [0, 1]$



\hat{h}_2



is rich $y \in \{0, 1\}$

Mansion and yacht just aids to determine if person is rich

(hidden) owns mansion:
probability $\in [0, 1]$

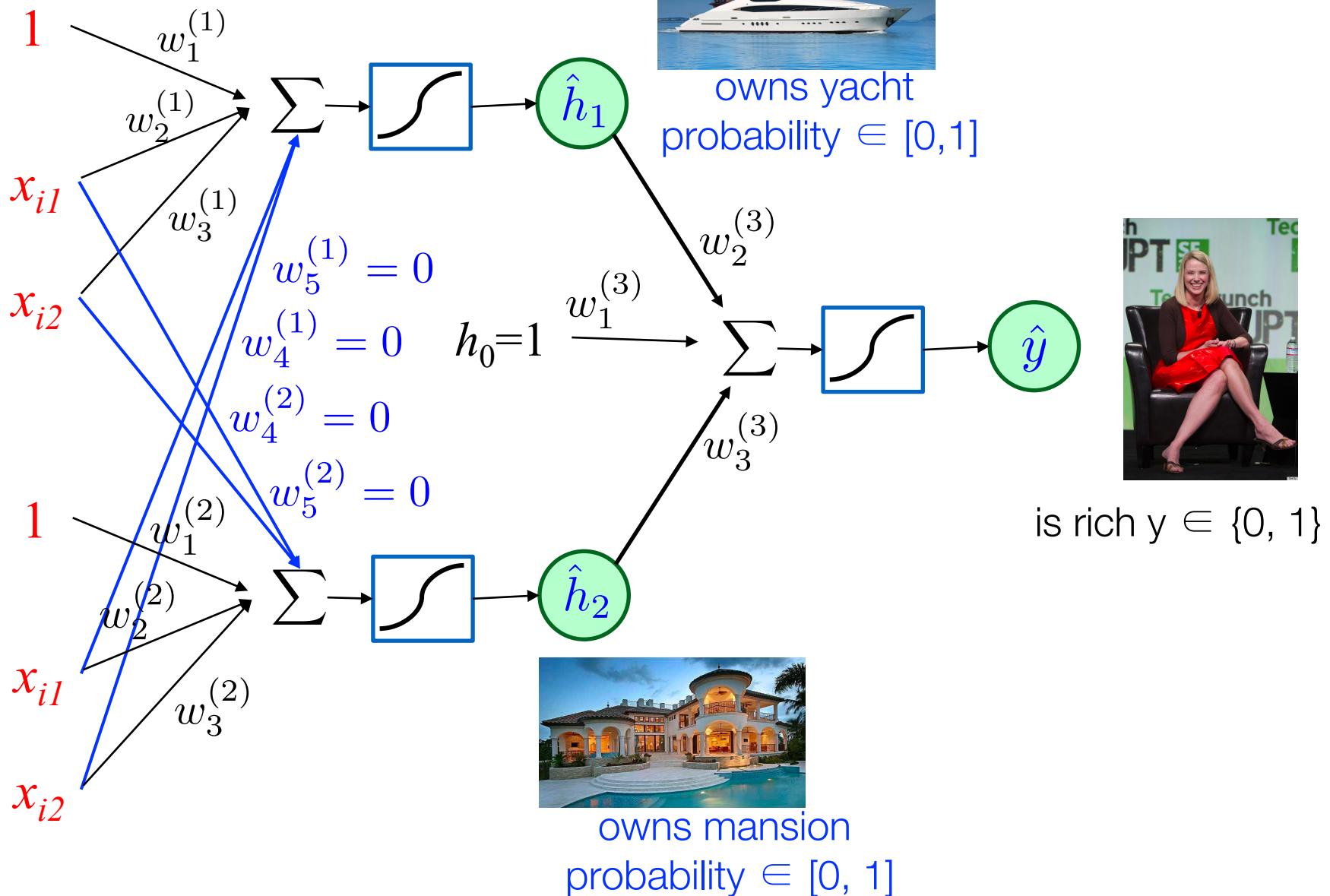


We don't know / don't care if person owns these items

Equivalent model: Fully connected with zero weights between yacht and mansion parts of the model

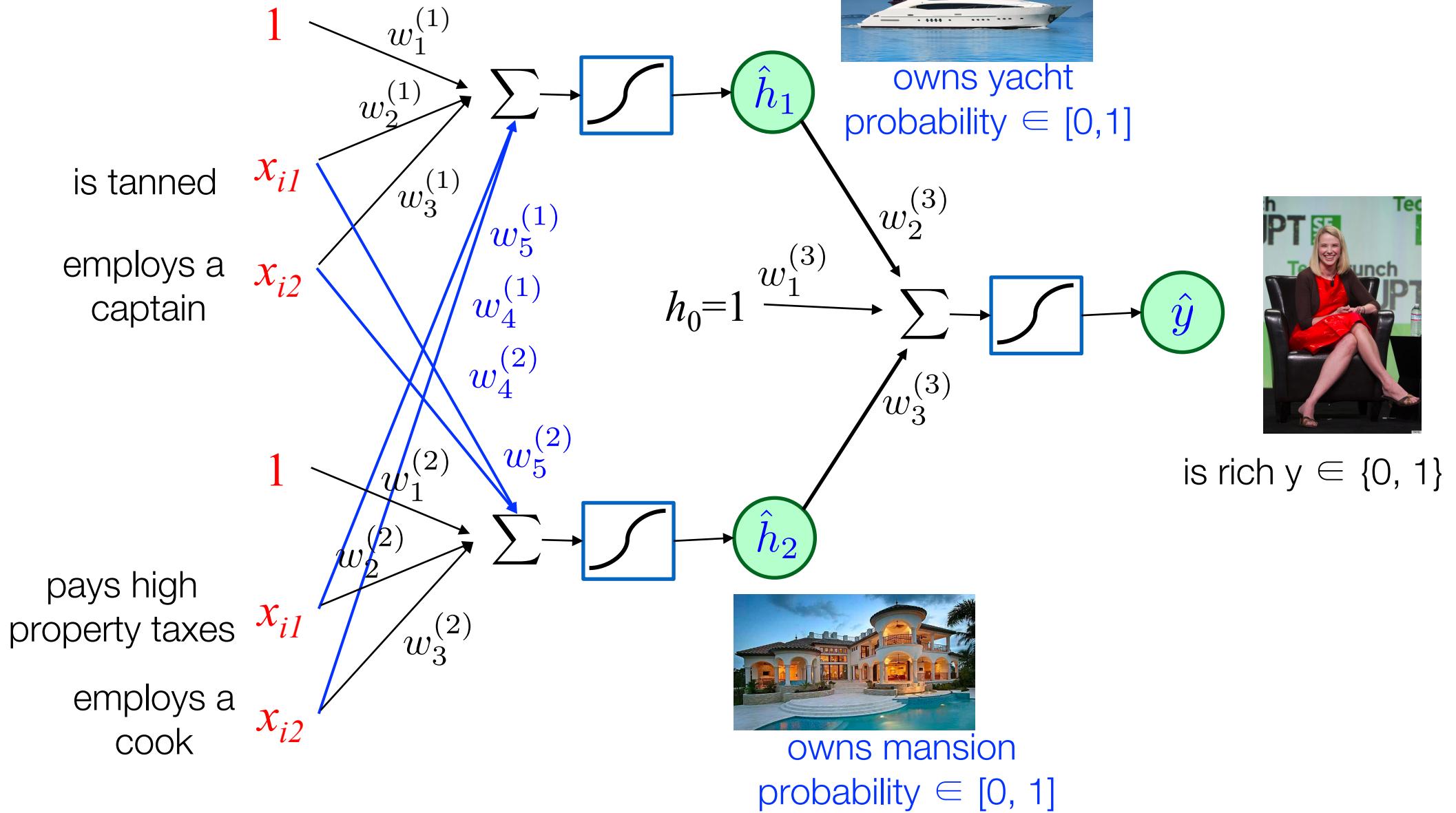
Model M2

is tanned
employs a captain
pays high property taxes
employs a cook



More flexible model: Fully connected allowing interconnected weights to have any value

Model M3



Model search for our example

- Training data: $\{(x_1, y_1), \dots, (x_n, y_n)\}$
- $\mathbf{x} = (1, x_{11}, x_{12}, x_{21}, x_{22})$
- $\mathbf{w}^{(k)} = (b, w_2^{(k)}, w_3^{(k)}, w_4^{(k)}, w_5^{(k)})$, $k = 1, 2$, where $b = w_1^{(1)} + w_1^{(2)}$
- $\mathbf{w}^{(3)} = (w_1^{(3)}, w_2^{(3)}, w_3^{(3)})$

Optimize using maximum likelihood estimation

$$\begin{aligned} & \arg \max_{\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \mathbf{w}^{(3)}} \frac{1}{n} \sum_{i=1}^n \log p(y = y_i | \mathbf{x}_i; \mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \mathbf{w}^{(3)}) \\ &= \frac{1}{n} \sum_{i=1}^n y_i \log \sigma((\mathbf{w}^{(3)})^T \mathbf{h}(\mathbf{x}_i)) + (1 - y_i) \log(1 - \sigma((\mathbf{w}^{(3)})^T \mathbf{h}(\mathbf{x}_i))), \end{aligned}$$

Characteristics of user i (*tan, captain, cook, ...*)

User i in training data: is rich $y_i \in \{0, 1\}$

where $\sigma(x) = \frac{\exp(x)}{1+\exp(x)}$, and $\mathbf{h}(\mathbf{x}) = (1, \hat{h}_1(\mathbf{x}), \hat{h}_2(\mathbf{x}))$,

$$\hat{h}_1(\mathbf{x}) = p(h_1 = 1 | \mathbf{x}) = \sigma((\mathbf{w}^{(1)})^T \mathbf{x})$$

and

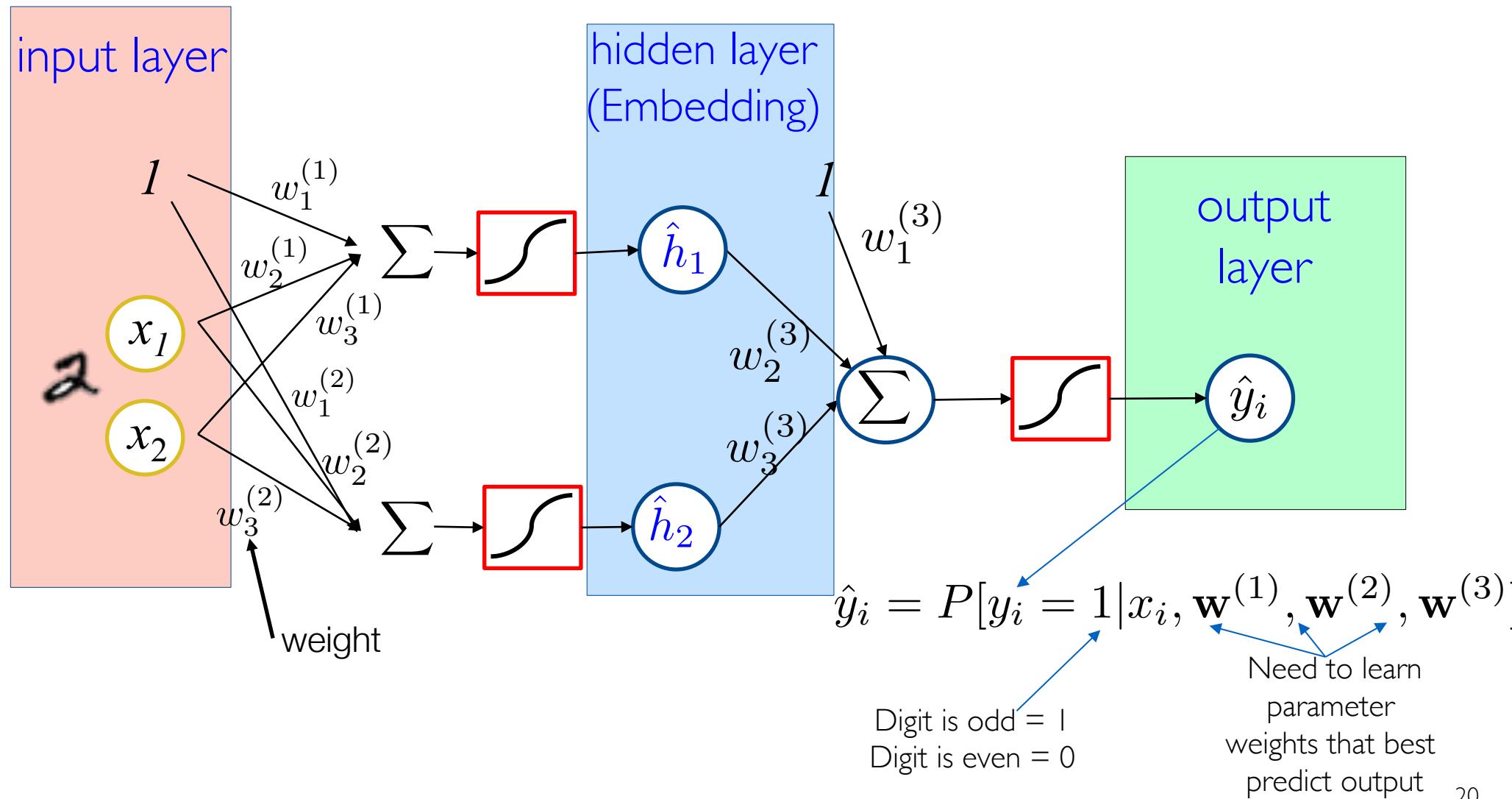
$$\hat{h}_2(\mathbf{x}) = p(h_2 = 1 | \mathbf{x}) = \sigma((\mathbf{w}^{(2)})^T \mathbf{x})$$

Questions

- What is the training data?
- What is the model?
- What is the score function?
- What is the search procedure?
- Which model is more likely to overfit the data? M1, M2, or M3?
 - Why are M1 and M2 equivalent?

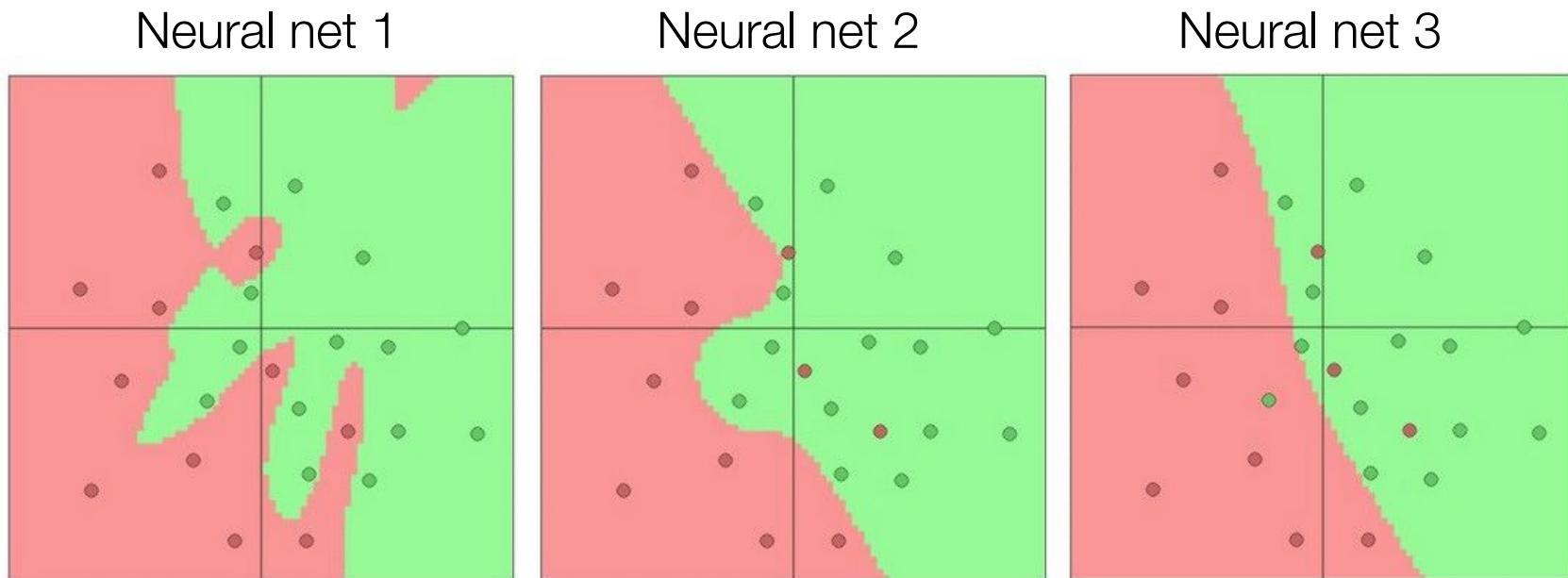
Neural Network Definitions

Neural networks are a combination of linear and non-linear functions



Neural Network = Ensemble of “Simple” Models

- Just like with ensemble classifiers, more complex models (e.g. logistic) requires care not to overfit the training data



Next Class: Searching for good neural network parameters (optimization)
