Machine Learning 10-601

Tom M. Mitchell
Machine Learning Department
Carnegie Mellon University

September 25, 2017

This section:
- Artificial neural networks
- Backpropagation
- Representation learning

Reading:
- Goodfellow: Chapter 6
- optional: Mitchell: Chapter 4

Artificial Neural Networks

We like logistic regression, but

$$P(Y = 1|X =< X_1, ...X_n >) = \frac{1}{1 + exp(w_0 + \sum_i w_i X_i)}$$

- how would it perform when trying to learn
  P(image contains Hillary Clinton | pixel values X_1, X_2 ... $X_{10,000}$)

We like logistic regression, but

$$P(Y = 1|X =< X_1, ...X_n >) = \frac{1}{1 + exp(w_0 + \sum_i w_i X_i)}$$

what X_i image features to use? edges? color blotches? generic face? subwindows? lighting invariant properties? position independent? SIFT features?

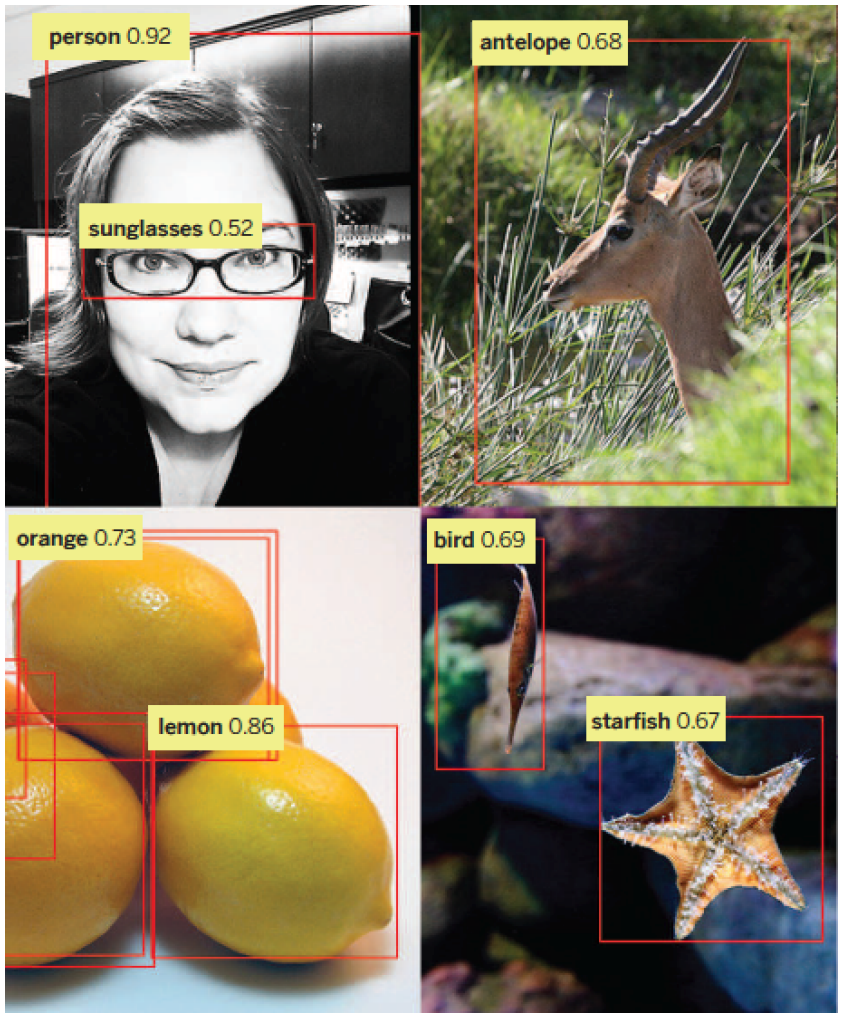
We like logistic regression, but

$$P(Y = 1|X =< X_1, ...X_n >) = \frac{1}{1 + exp(w_0 + \sum_i w_i X_i)}$$

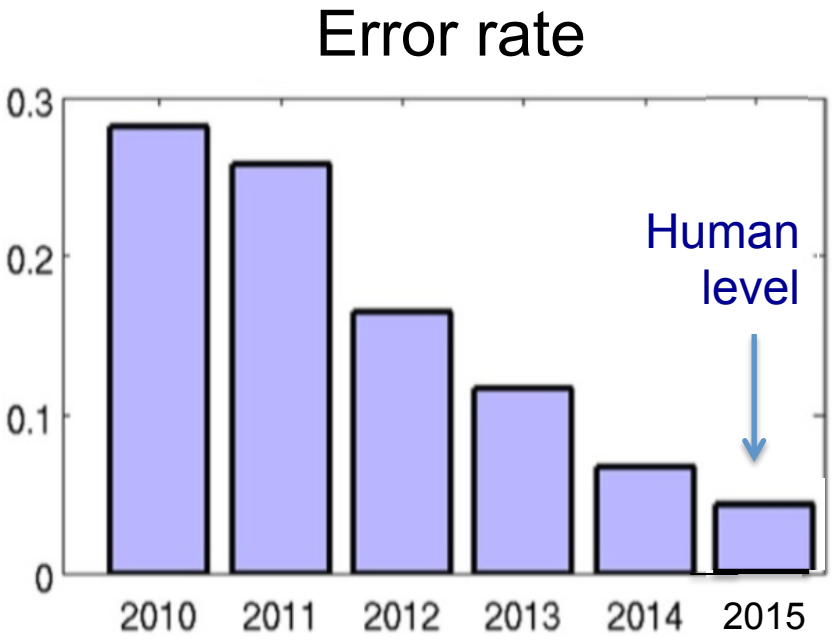
what X_i image features to use? edges? color blotches? generic face? subwindows? lighting invariant properties? position independent?

Deep nets : learn the features automatically !

Computer Vision



Imagenet Visual Recognition Challenge



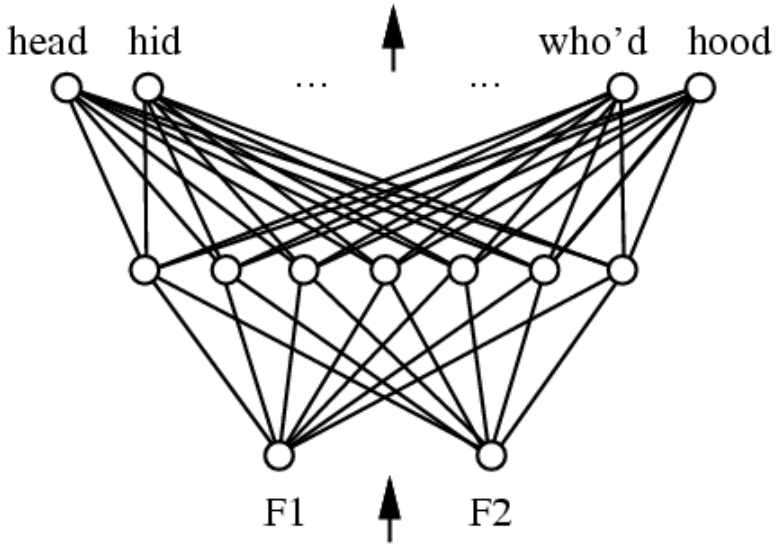
Artificial Neural Networks to learn f: X → Y

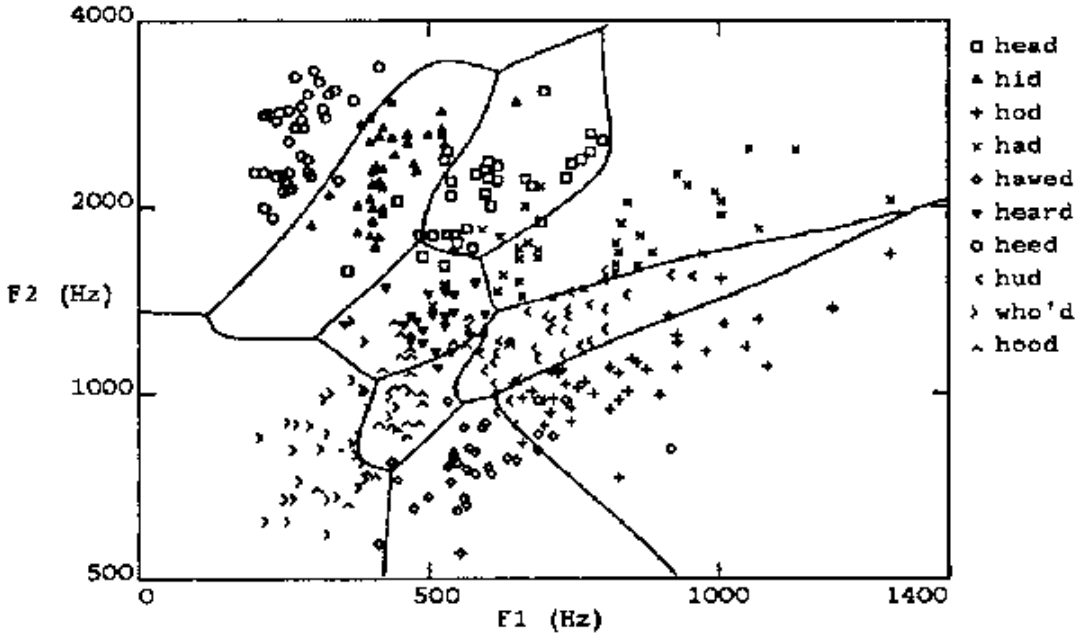
- f might be complex, non-linear, real or discrete-valued
- X (vector of) continuous and/or discrete vars
- Y (vector of) continuous and/or discrete vars

- we can also train network to learn P(Y|X)

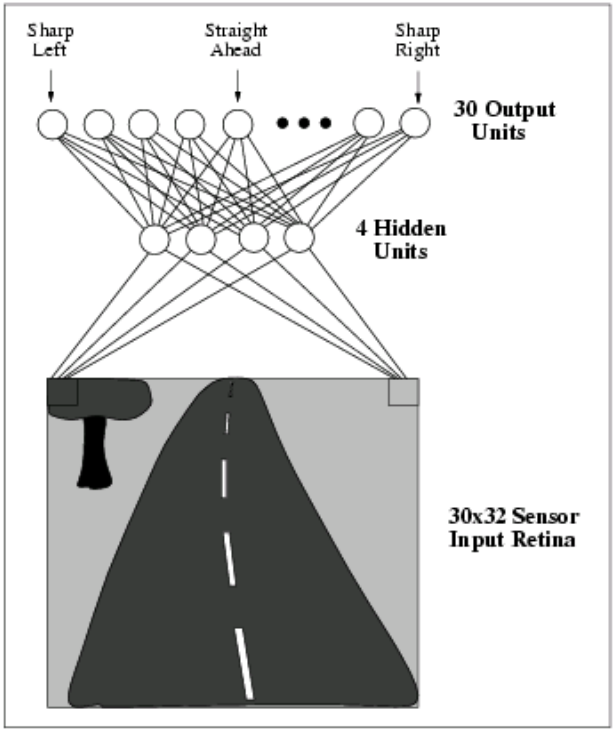
- Represent f by *network* of computational units
  which may contain millions of trained parameters

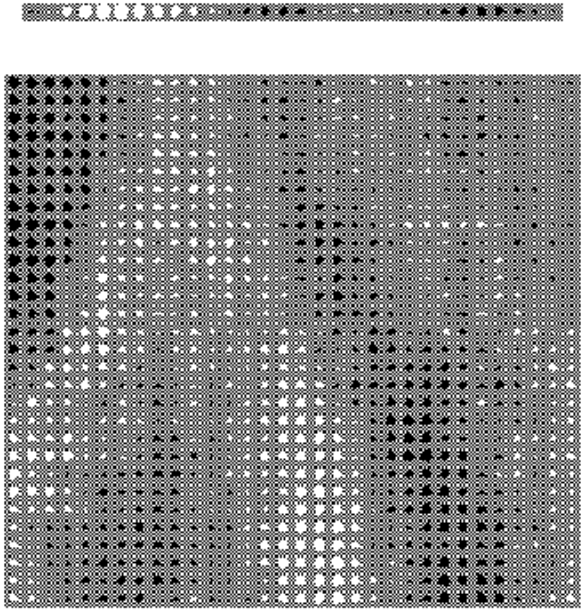
Multilayer Networks of Sigmoid Units





ALVINN
[Pomerleau 1993]





Speech Recognition

October 2016: Microsoft reports reaching human-level accuracy of 94.1% at standard switchboard task.

March 2017: IBM reports 94.5% accuracy

Text Generated from Images

Given	Generated
	dog, cat, pet, kitten, puppy, ginger, tongue, kitty, dogs, furry
	sea, france, boat, mer, beach, river, bretagne, plage, brittany
	portrait, child, kid, ritratto, kids, children, boy, cute, boys, italy

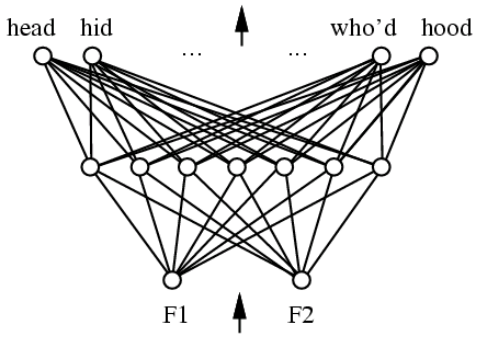
Given	Generated
	insect, butterfly, insects, bug, butterflies, lepidoptera
	graffiti, streetart, stencil, sticker, urbanart, graff, sanfrancisco
	canada, nature, sunrise, ontario, fog, mist, bc, morning

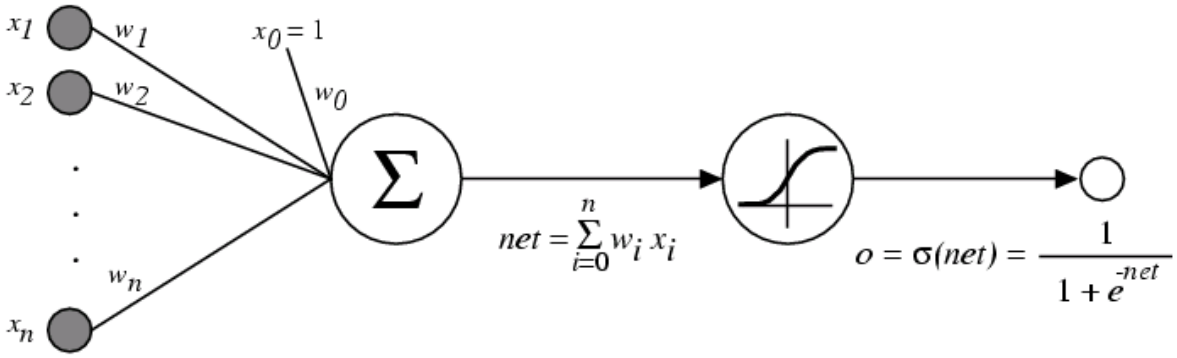
[Courtesy, R. Salakhutdinov]

Text Generated from Images

Given	Generated
	portrait, women, army, soldier, mother, postcard, soldiers
	obama, barackobama, election, politics, president, hope, change, sanfrancisco, convention, rally
	water, glass, beer, bottle, drink, wine, bubbles, splash, drops, drop

Sigmoid Unit





$\sigma(x)$ is the sigmoid function

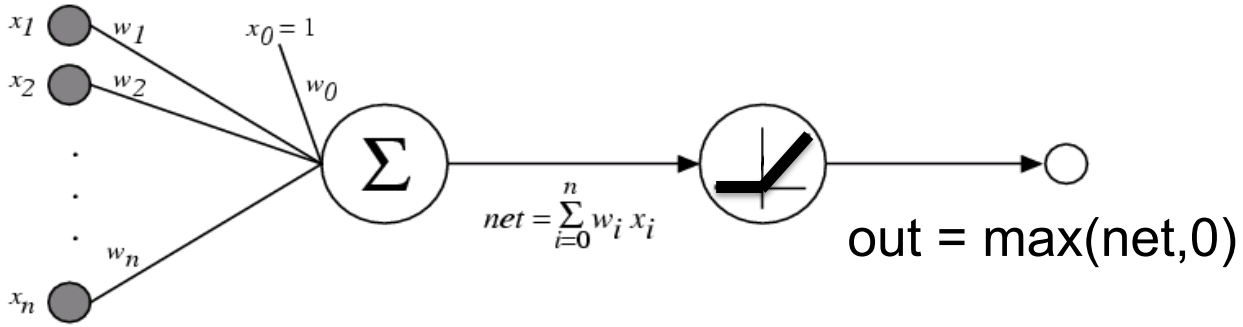
$$\frac{1}{1 + e^{-x}}$$

Nice property: $\frac{d\sigma(x)}{dx} = \sigma(x)(1 - \sigma(x))$

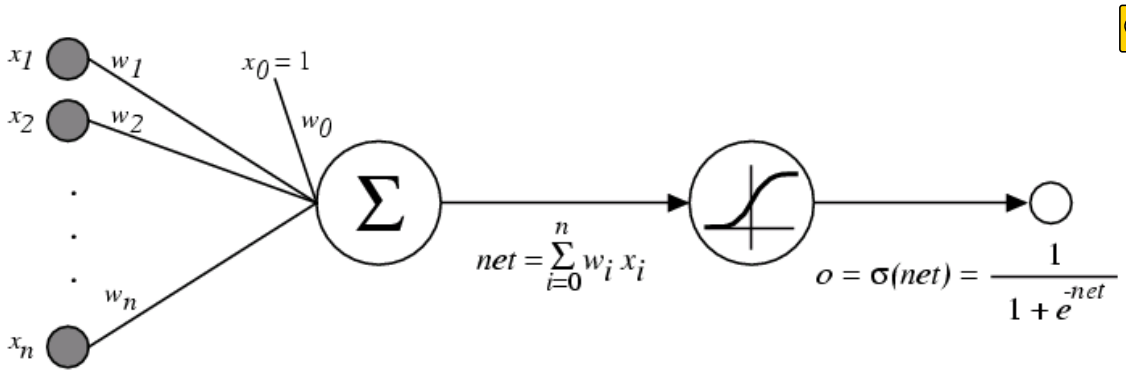
Sigmoid units are exactly the form we use for logistic regression

Rectified Linear Unit (ReLU)

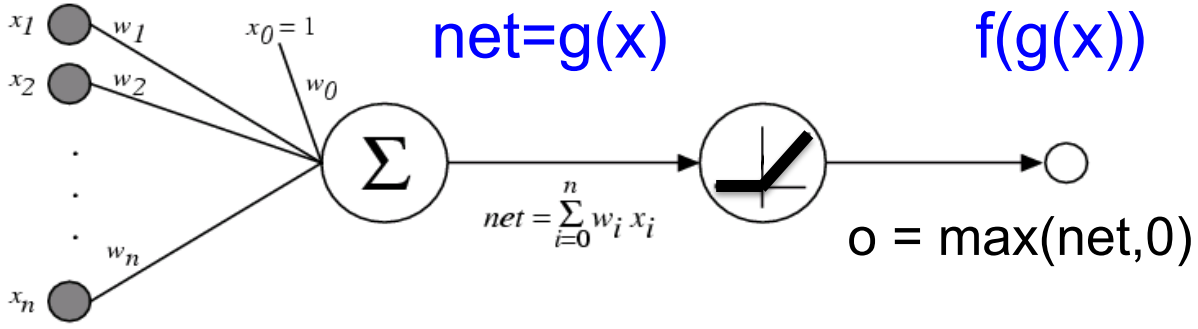
- Rectified linear unit : linear with thresholded output

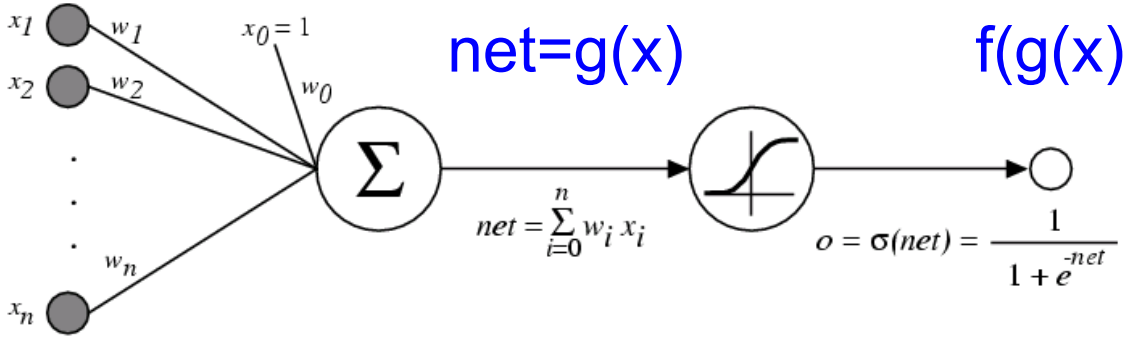


- Sigmoid unit:



Units often composed so output o = f(g(x))
f is called the “activation function”





Many types of parameterized units

- Sigmoid units
- ReLU
- Leaky ReLU (fixed non-zero slope for input<0)
- Parametric ReLU (trainable slope)
- Max Pool
- Inner Product
- GRU's
- LSTM's
- Matrix multiply
- …. no end in sight

Any unit h(X; W) that is differentiable w.r.t. X and W

Training Deep Nets

1. Choose loss function J(θ) to optimize
   - sum of squared errors for y continuous: $\Sigma (y - h(x; \theta))^2$
   - maximize conditional likelihood: $\Sigma \log P(y|x; \theta)$
   - MAP estimate: $\Sigma \log P(y|x; \theta) P(\theta)$
   - ~~0/1 loss. Sum of classification errors: $\Sigma \delta(y = h(x; \theta)$~~
   - …
2. Design network architecture
   - Network of layers (ReLU's, sigmoid, convolutions, ...)
   - Widths of layers
   - Fully or partly interconnected
   - ...
3. Training algorithm
   - Derive gradient formulas Ê
   - Choose gradient descent method, including stopping condition
   - Experiment with alternative architectures

Example

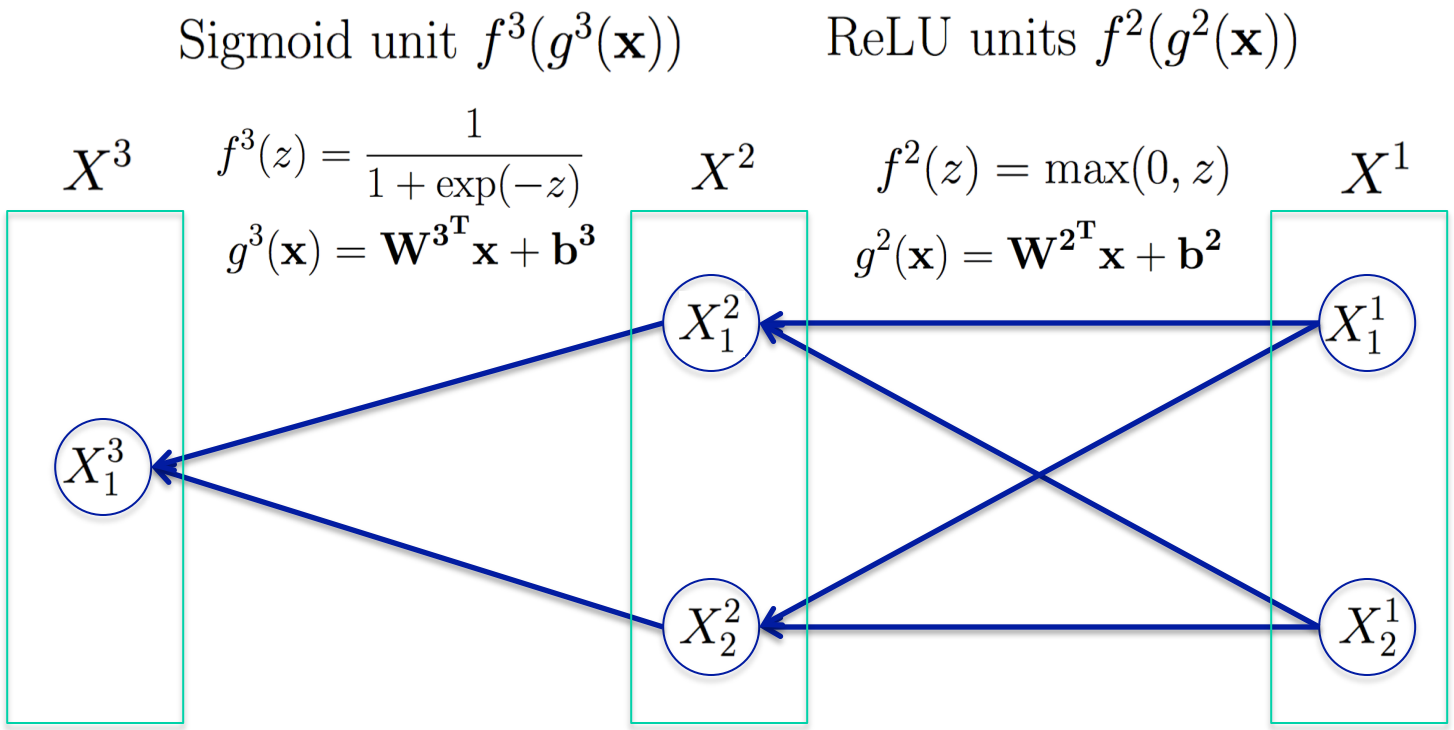
Example: Learn probabilistic XOR

- Given boolean Y, X_1, X_2 learn $P(Y|X_1,X_2)$, where

$$P(Y = 0|X_1 = X_2) = 0.9$$

$$P(Y = 1|X_1 \neq X_2) = 0.9$$

- Can we learn this with logistic regression?



Loss function $J(\theta)$ to be minimized: negative log likelihood

$$J(\theta) = \sum_{\langle \mathbf{x}, y \rangle \in D} -\log P(Y = y | X = \mathbf{x})$$

where $X_1^3 = P(Y = 1 | X = \mathbf{X^1})$, $\theta = \{\mathbf{W^3}, \mathbf{b^3}, \mathbf{W^2}, \mathbf{b^2}\}$

Example: Learn probabilistic XOR

- Given boolean Y, X_1, X_2 learn $P(Y|X_1,X_2)$, where

$$P(Y = 0|X_1 = X_2) = 0.9$$

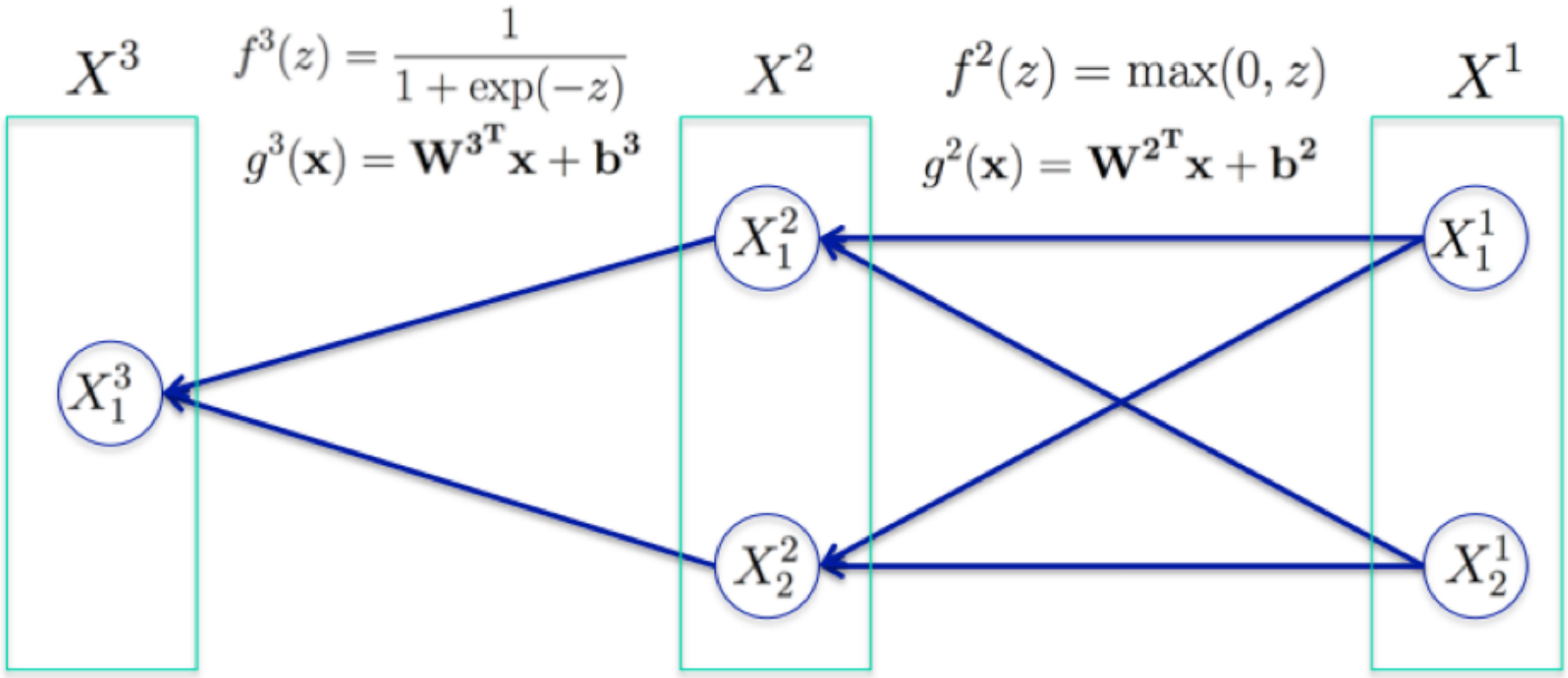
$$P(Y = 1|X_1 \neq X_2) = 0.9$$

- Choose max conditional likelihood,
  equally, minimize negative log conditional likelihood

$$\begin{aligned} J(\theta) &= -\sum_k \log P(Y = y_k|X = \mathbf{x}_k) \\ &= -\sum_k y_k \log P(Y = 1|X = x_k) + (1 - y_k)(1 - P(Y = 1|X = x_k)) \end{aligned}$$

Feed Forward

Sigmoid unit $f^3(g^3(\mathbf{x}))$ ReLU units $f^2(g^2(\mathbf{x}))$



X^3	$g^3(X^2)$	W^{3T}		b^3
0.53	0.12	0.10	-0.09	0.10

X^2
0.20
0.00
1

$g^2(X^1)$	W^{2T}		b^2
0.20	0.10	-0.10	0.10
-0.25	-0.20	0.10	-0.05

X^1
1
0
1

Derive the gradient we need

$$
\begin{aligned}
J(\theta) &= -\sum_k \log P(Y = y_k | X = \mathbf{x}_k) \\
&= -\sum_k y_k \log P(Y = 1 | X = x_k) + (1 - y_k)(1 - P(Y = 1 | X = x_k))
\end{aligned}
$$

simplify notation by considering just one training example

$$
\begin{aligned}
\frac{\partial J(\theta)}{\partial X_1^3} &= \frac{\partial \left(-Y \log P(Y = 1|X) - (1 - Y)(1 - P(Y = 1|X))\right)}{\partial X_1^3} \\
&= \frac{\partial \left(-Y \log X_1^3 - (1 - Y) \log(1 - X_1^3)\right)}{\partial X_1^3} \\
&= \frac{-Y}{X_1^3} - (1 - Y)\frac{1}{1 - X_1^3}(-1) \\
&= \frac{-Y}{X_1^3} + \frac{(1 - Y)}{1 - X_1^3}
\end{aligned}
$$

$$\frac{\partial J(\theta)}{\partial g_1^3} = \frac{\partial J(\theta)}{\partial X_1^3}\frac{\partial X_1^3}{\partial g^3} = \frac{\partial J(\theta)}{\partial X_1^3} X_1^3(1-X_1^3)$$

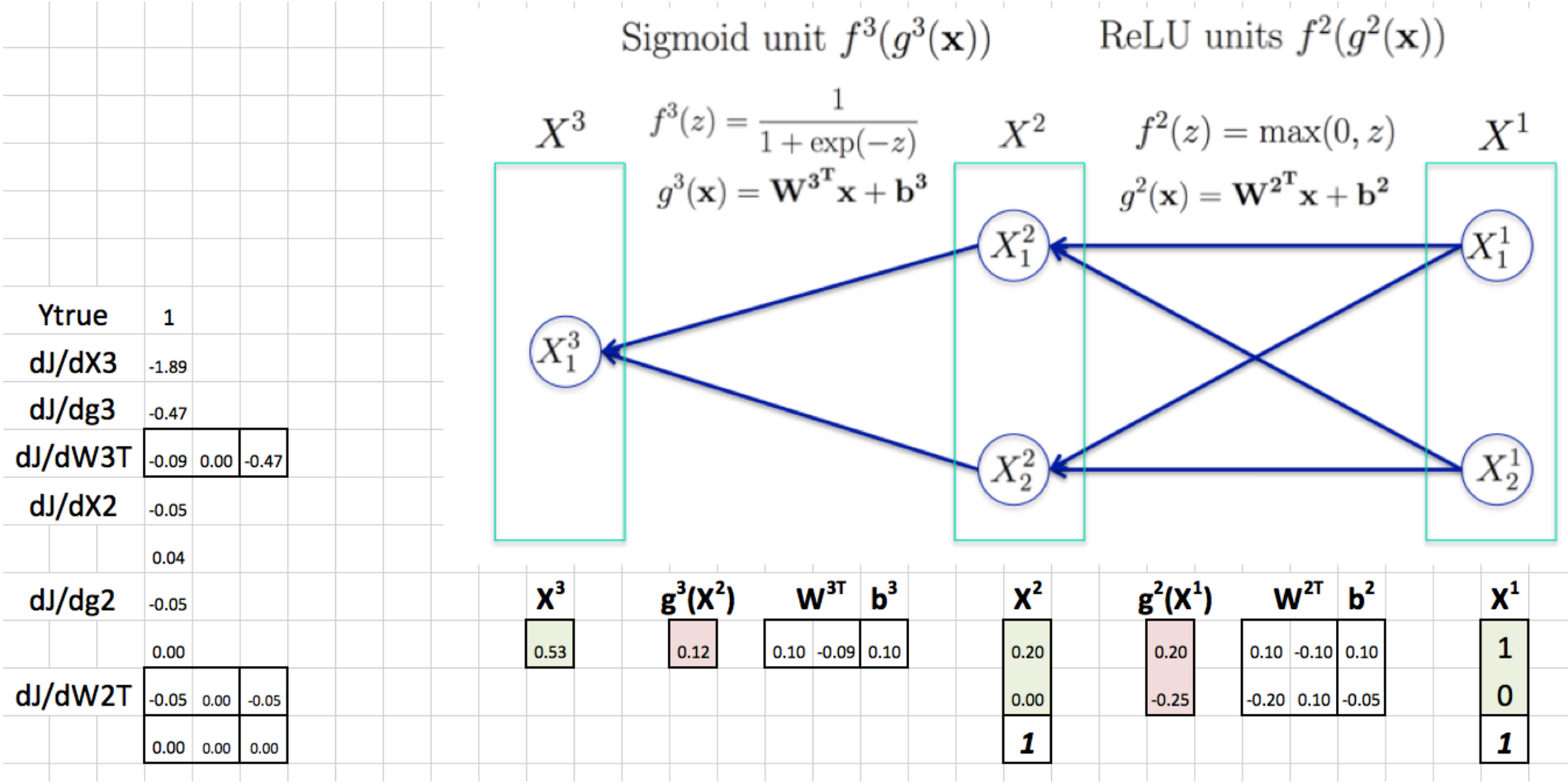
$$\frac{\partial J(\theta)}{\partial w_i^3} = \frac{\partial J(\theta)}{\partial g^3}\frac{\partial g^3}{\partial w_i^3} = \frac{\partial J(\theta)}{\partial g^3} X_i^2$$

$$\frac{\partial J(\theta)}{\partial X_i^2} = \frac{\partial J(\theta)}{\partial g^3}\frac{\partial g^3}{\partial X_i^2} = \frac{\partial J(\theta)}{\partial g^3} w_i^3$$

$$\frac{\partial J(\theta)}{\partial g_i^2} = \frac{\partial J(\theta)}{\partial X_i^2}\frac{\partial X_i^2}{\partial g_i^2} = \frac{\partial J(\theta)}{\partial X_i^2} \times [\text{if } g_i^2 > 0 \text{ then 1 else 0}]$$

$$\frac{\partial J(\theta)}{\partial w_{ik}^2} = \frac{\partial J(\theta)}{\partial g_i^2}\frac{\partial g_i^2}{\partial X_k^1} = \frac{\partial J(\theta)}{\partial g^3} X_k^1$$

Back propagation



Ytrue	1		
dJ/dX3	-1.89		
dJ/dg3	-0.47		
dJ/dW3T	-0.09	0.00	-0.47
dJ/dX2	-0.05		
	0.04		
dJ/dg2	-0.05		
	0.00		
dJ/dW2T	-0.05	0.00	-0.05
	0.00	0.00	0.00

X^3	$g^3(X^2)$	W^{3T}		b^3	X^2	$g^2(X^1)$	W^{2T}		b^2	X^1
0.53	0.12	0.10	-0.09	0.10	0.20	0.20	0.10	-0.10	0.10	1
					0.00	-0.25	-0.20	0.10	-0.05	0
					1					*1*