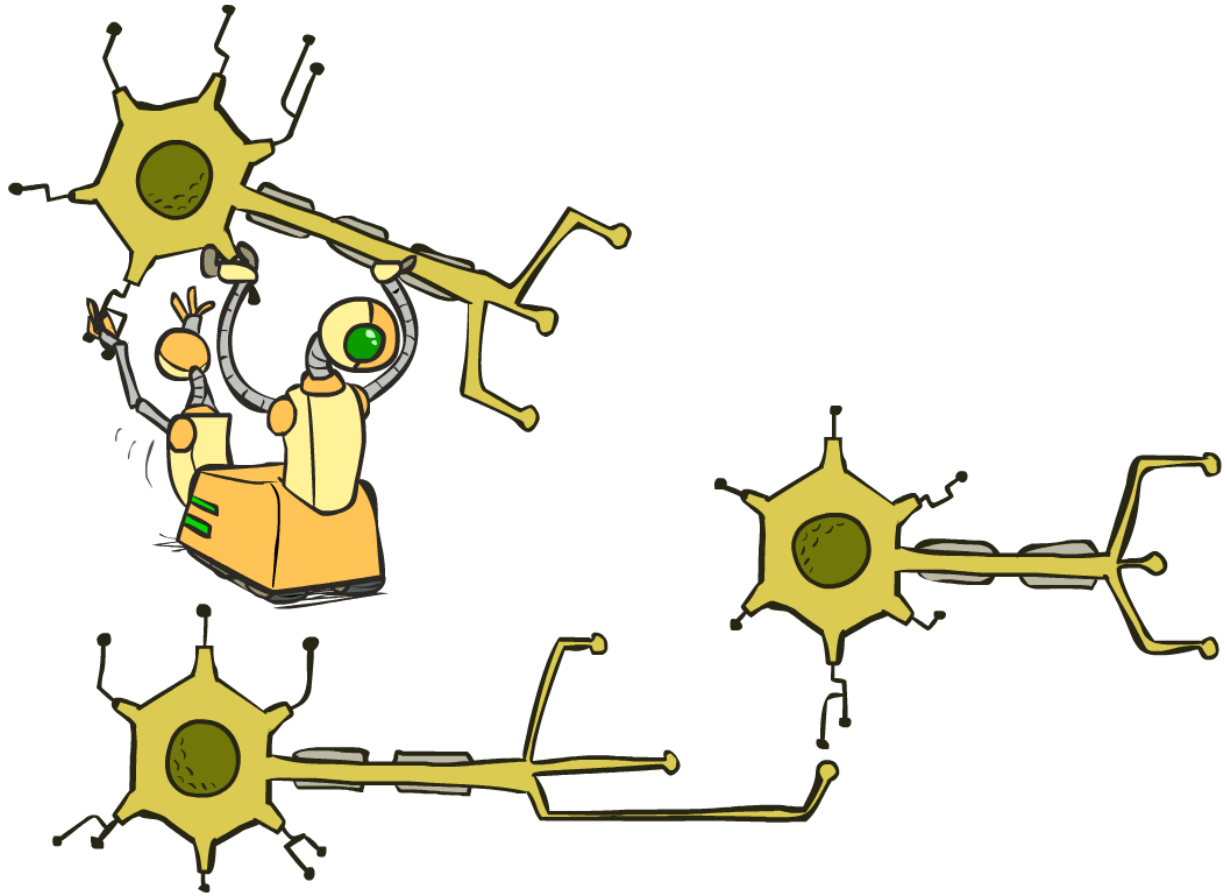
# CS 188: Artificial Intelligence

## Neural Nets

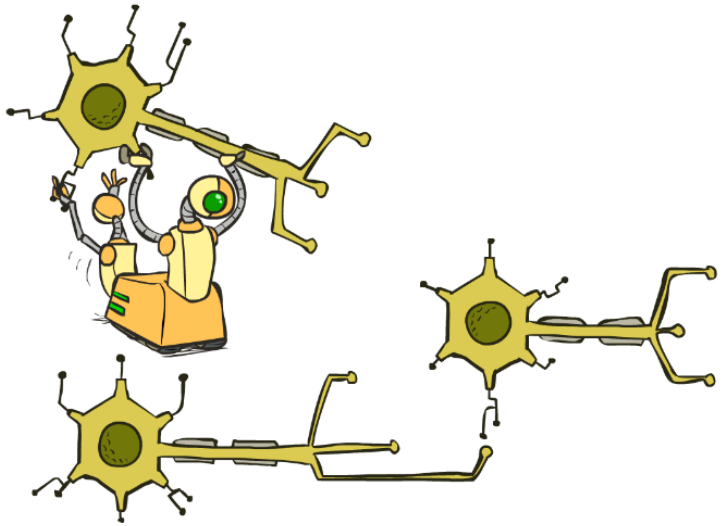


Instructors: Brijen Thananjeyan and Aditya Baradwaj--- University of California, Berkeley

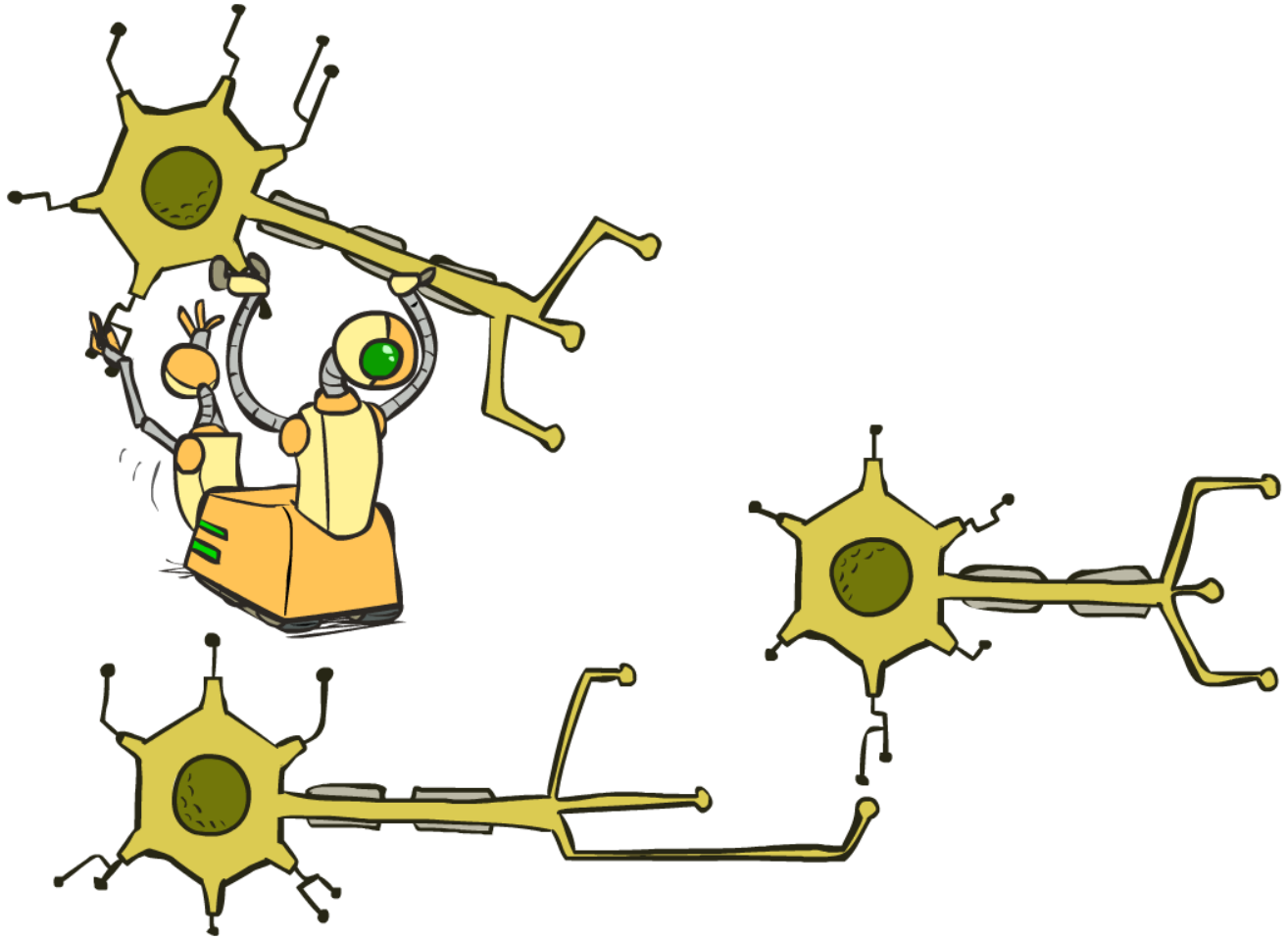
[These slides were created by Dan Klein, Pieter Abbeel, Sergey Levine. All CS188 materials are at http://ai.berkeley.edu.]

# Announcements

- MT2 Self Assessment: on Gradescope + due Sunday
- Q5:clustering and Q7:decision trees: now optional on HW6 written component
- Tomorrow: Guest lecture canceled, math/ML review

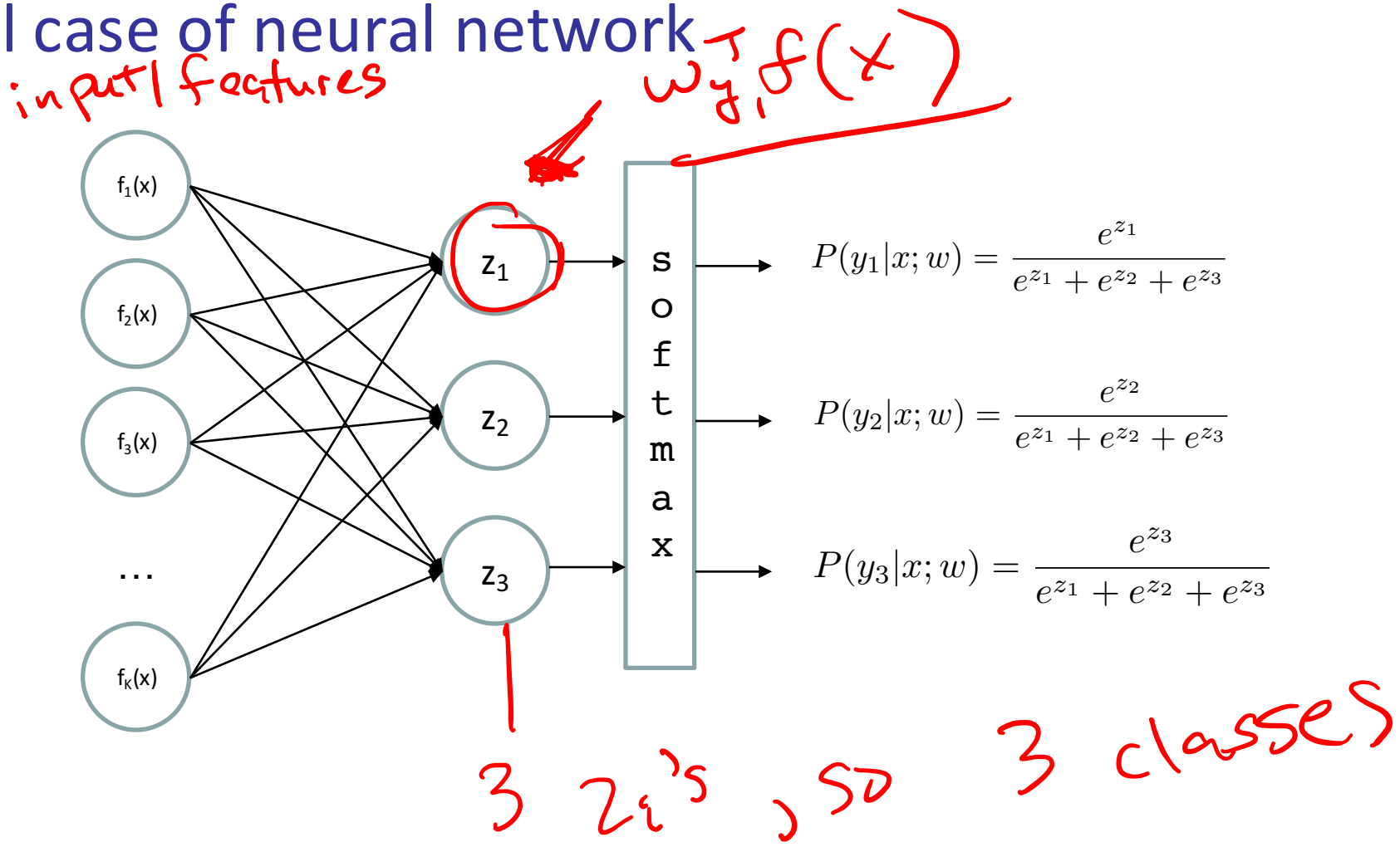


# Neural Networks

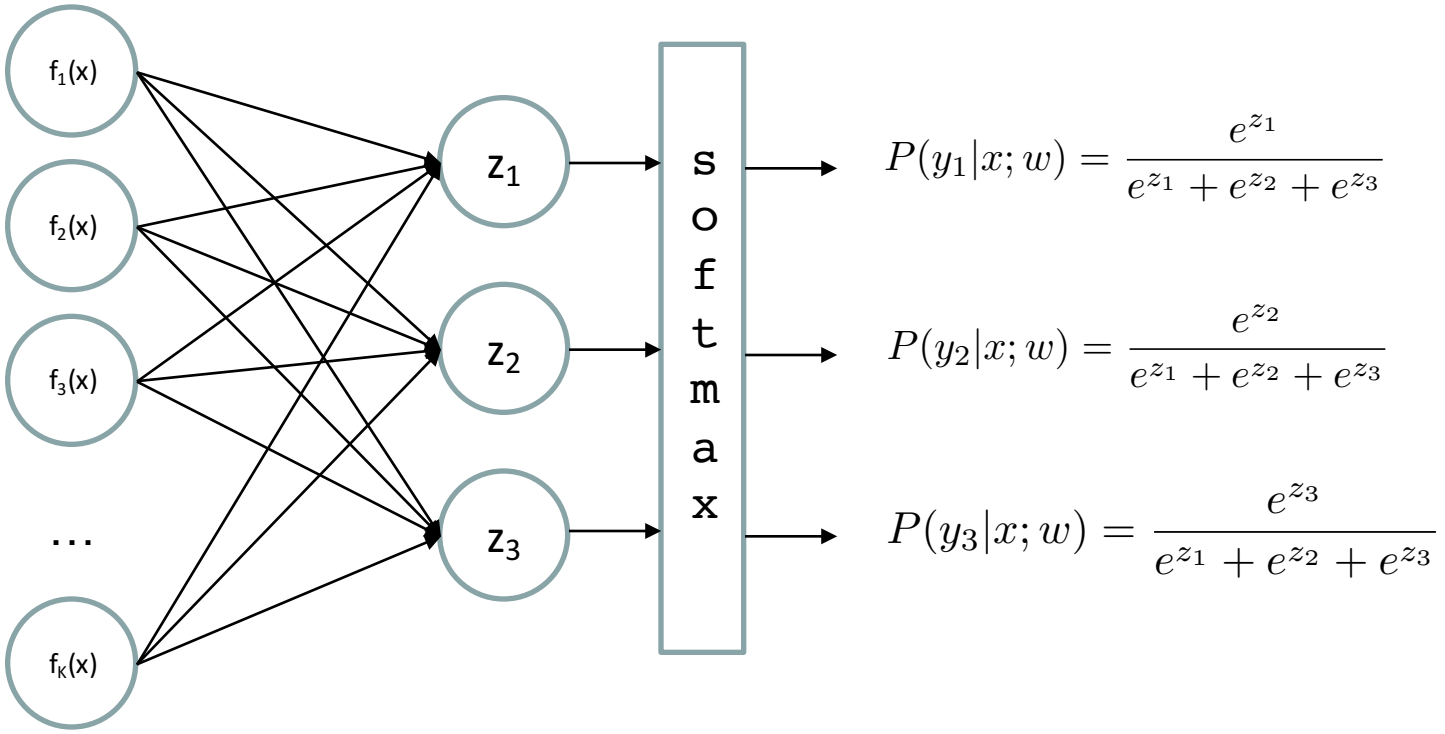


# Multi-class Logistic Regression

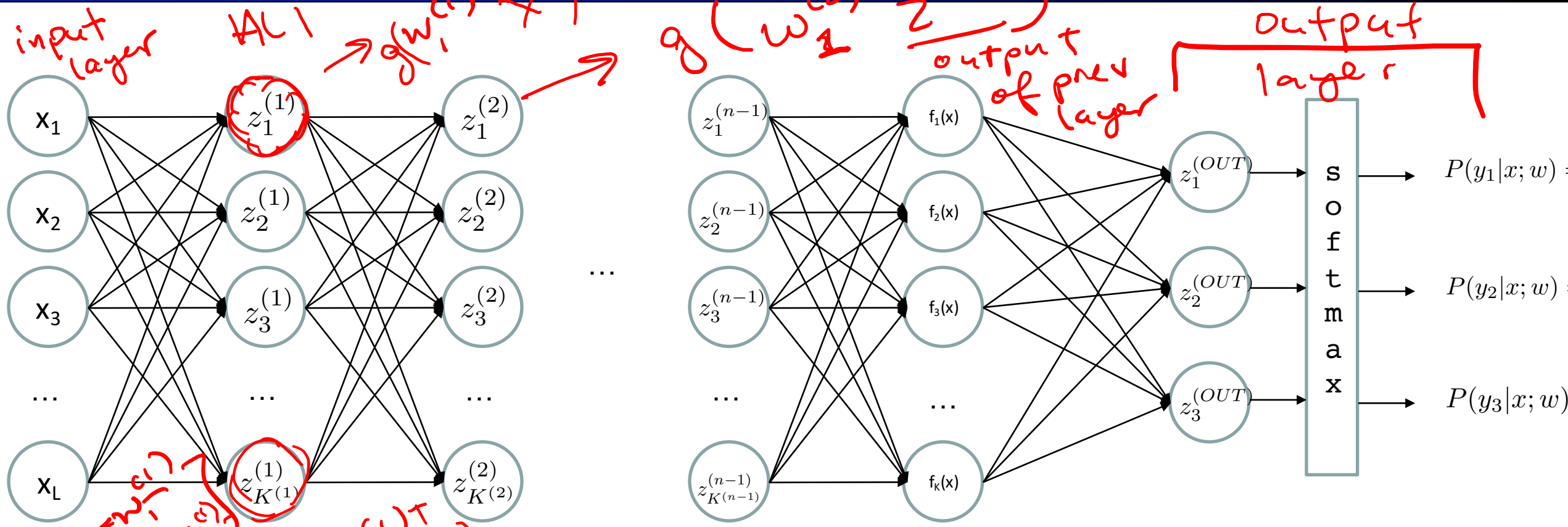
- = special case of neural network



# Deep Neural Network = Also learn the features!



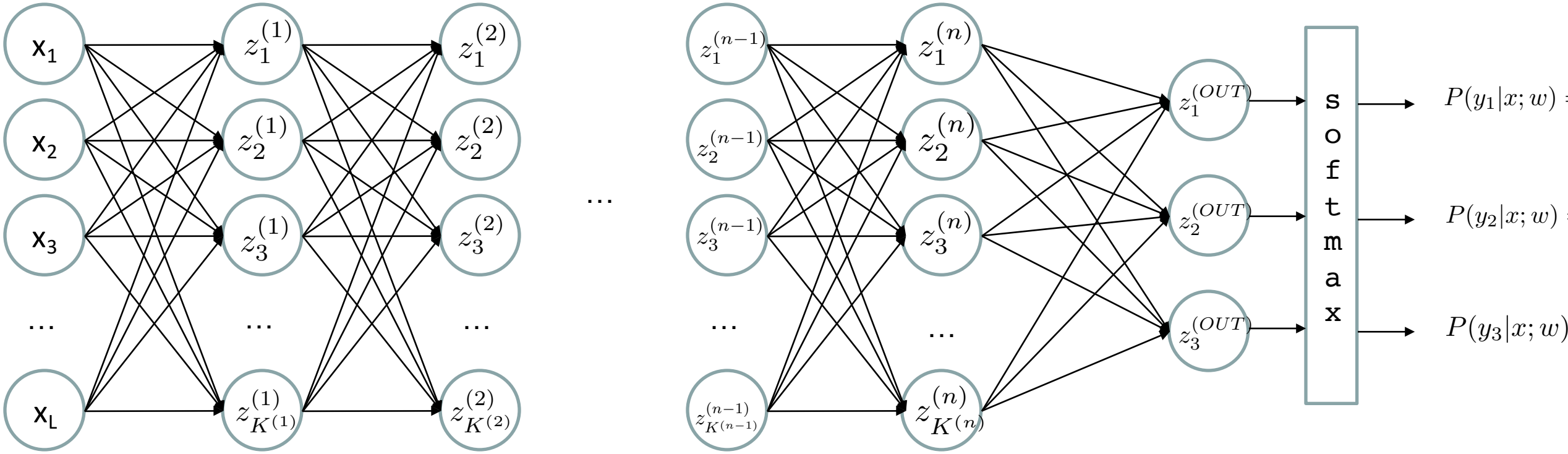
# Deep Neural Network = Also learn the features!




$$z_i^{(k)} = g\left(\sum_{i,j} W_{i,j}^{(k-1,k)} z_j^{(k-1)}\right)$$

**g = nonlinear activation function**

# Deep Neural Network = Also learn the features!

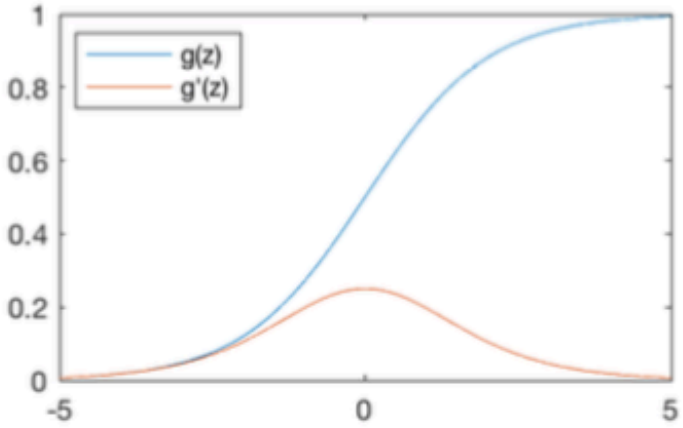



$$z_i^{(k)} = g(\sum_j W_{i,j}^{(k-1,k)} z_j^{(k-1)})$$

**g = nonlinear activation function**

# Common Activation Functions

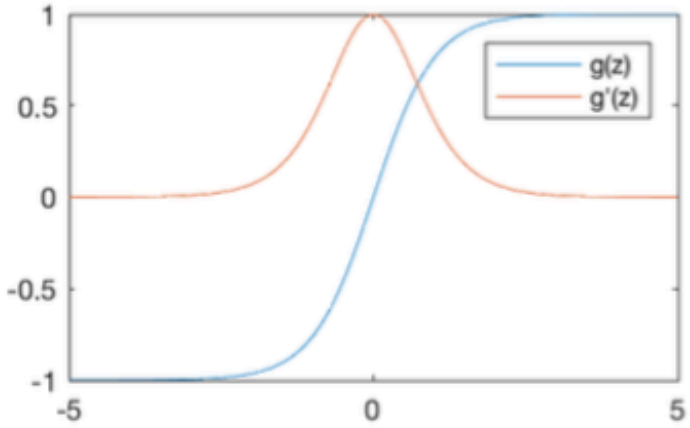
## Sigmoid Function




$$g(z) = \frac{1}{1 + e^{-z}}$$

$$g'(z) = g(z)(1 - g(z))$$

## Hyperbolic Tangent

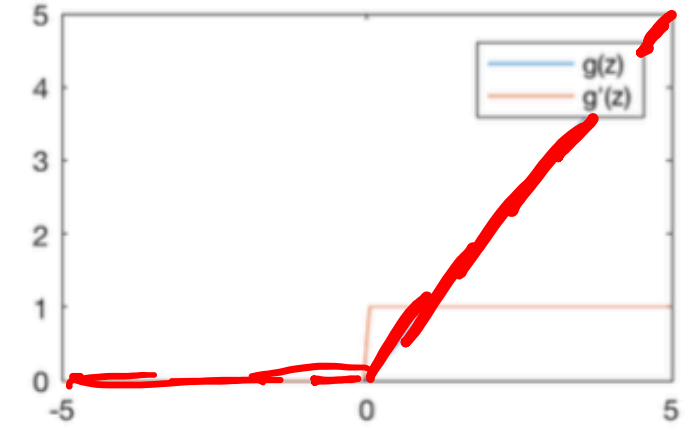



$$g(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

$$g'(z) = 1 - g(z)^2$$

tanh

## Rectified Linear Unit (ReLU)




$$g(z) = \max(0, z)$$

$$g'(z) = \begin{cases} 1, & z > 0 \\ 0, & \text{otherwise} \end{cases}$$

$g(z) = \begin{cases} z & z > 0 \\ 0 & \text{o/w} \end{cases}$

[source: MIT 6.S191 introtodeeplearning.com]

# Deep Neural Network: Also Learn the Features!

- Training the deep neural network is just like logistic regression:

→ model that best explains data

$$\max_w \quad ll(w) = \max_w \quad \sum_i \log P(y^{(i)}|x^{(i)}; w)$$

just w tends to be a much, much larger vector ☺

→just run gradient ascent

+ stop when log likelihood of hold-out data starts to decrease

# Neural Networks Properties

- Theorem (Universal Function Approximators). A two-layer neural network with a sufficient number of neurons can approximate any continuous function to any desired accuracy.

- Practical considerations
  - Can be seen as learning the features
  - Large number of neurons
    - Danger for overfitting
    - (hence early stopping!)

# How about computing all the derivatives?

- Derivatives tables:

$$\frac{d}{dx}(a) = 0$$

$$\frac{d}{dx}(x) = 1$$

$$\frac{d}{dx}(au) = a\frac{du}{dx}$$

$$\frac{d}{dx}(u+v-w) = \frac{du}{dx}+\frac{dv}{dx}-\frac{dw}{dx}$$

$$\frac{d}{dx}(uv) = u\frac{dv}{dx}+v\frac{du}{dx}$$

$$\frac{d}{dx}\left(\frac{u}{v}\right) = \frac{1}{v}\frac{du}{dx}-\frac{u}{v^2}\frac{dv}{dx}$$

$$\frac{d}{dx}(u^n) = nu^{n-1}\frac{du}{dx}$$

$$\frac{d}{dx}(\sqrt{u}) = \frac{1}{2\sqrt{u}}\frac{du}{dx}$$

$$\frac{d}{dx}\left(\frac{1}{u}\right) = -\frac{1}{u^2}\frac{du}{dx}$$

$$\frac{d}{dx}\left(\frac{1}{u^n}\right) = -\frac{n}{u^{n+1}}\frac{du}{dx}$$

$$\frac{d}{dx}[f(u)] = \frac{d}{du}[f(u)]\frac{du}{dx}$$

$$\frac{d}{dx}[\ln u] = \frac{d}{dx}[\log_e u] = \frac{1}{u}\frac{du}{dx}$$

$$\frac{d}{dx}\left[\log_a u\right] = \log_a e\frac{1}{u}\frac{du}{dx}$$

$$\frac{d}{dx}e^u = e^u\frac{du}{dx}$$

$$\frac{d}{dx}a^u = a^u \ln a\frac{du}{dx}$$

$$\frac{d}{dx}\left(u^v\right) = vu^{v-1}\frac{du}{dx}+\ln u\ \ u^v\frac{dv}{dx}$$

$$\frac{d}{dx}\sin u = \cos u\frac{du}{dx}$$

$$\frac{d}{dx}\cos u = -\sin u\frac{du}{dx}$$

$$\frac{d}{dx}\tan u = \sec^2 u\frac{du}{dx}$$

$$\frac{d}{dx}\cot u = -\csc^2 u\frac{du}{dx}$$

$$\frac{d}{dx}\sec u = \sec u\tan u\frac{du}{dx}$$

$$\frac{d}{dx}\csc u = -\csc u\cot u\frac{du}{dx}$$

[source: http://hyperphysics.phy-astr.gsu.edu/hbase/Math/derfunc.html

# How about computing all the derivatives?

- But neural net f is never one of those?
  - No problem: CHAIN RULE:

If $f(x) = g(h(x))$

Then $f'(x) = g'(h(x))h'(x)$

composition of differentiable functions

$x \xrightarrow{h} h(x) \xrightarrow{g} g(h(x))$

$\frac{\partial g(h(x))}{\partial x}$

Backpropagation alg: computes derivatives of compositions efficient

→ **Derivatives can be computed by following well-defined procedures**

# Automatic Differentiation

- Automatic differentiation software
  - e.g. Theano, TensorFlow, PyTorch, Chainer
  - Only need to program the function g(x,y,w)
  - Can automatically compute all derivatives w.r.t. all entries in w
  - This is typically done by caching info during forward computation pass of f, and then doing a backward pass = “backpropagation”
  - Autodiff / Backpropagation can often be done at computational cost comparable to the forward pass
- Need to know this exists
- How this is done? -- outside of scope of CS188

# Summary of Key Ideas

- Optimize probability of label given input

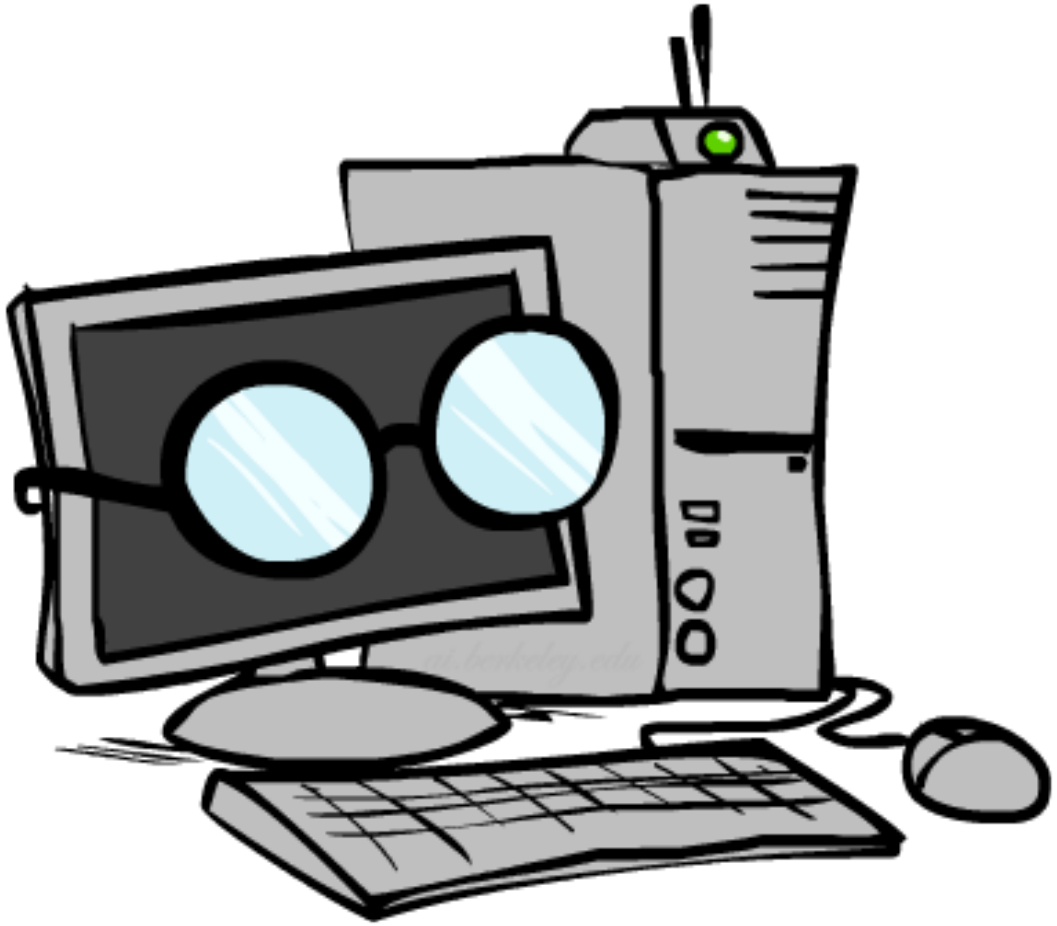
$$\max_w \; ll(w) = \max_w \; \sum_i \log P(y^{(i)}|x^{(i)};w)$$

plug in model

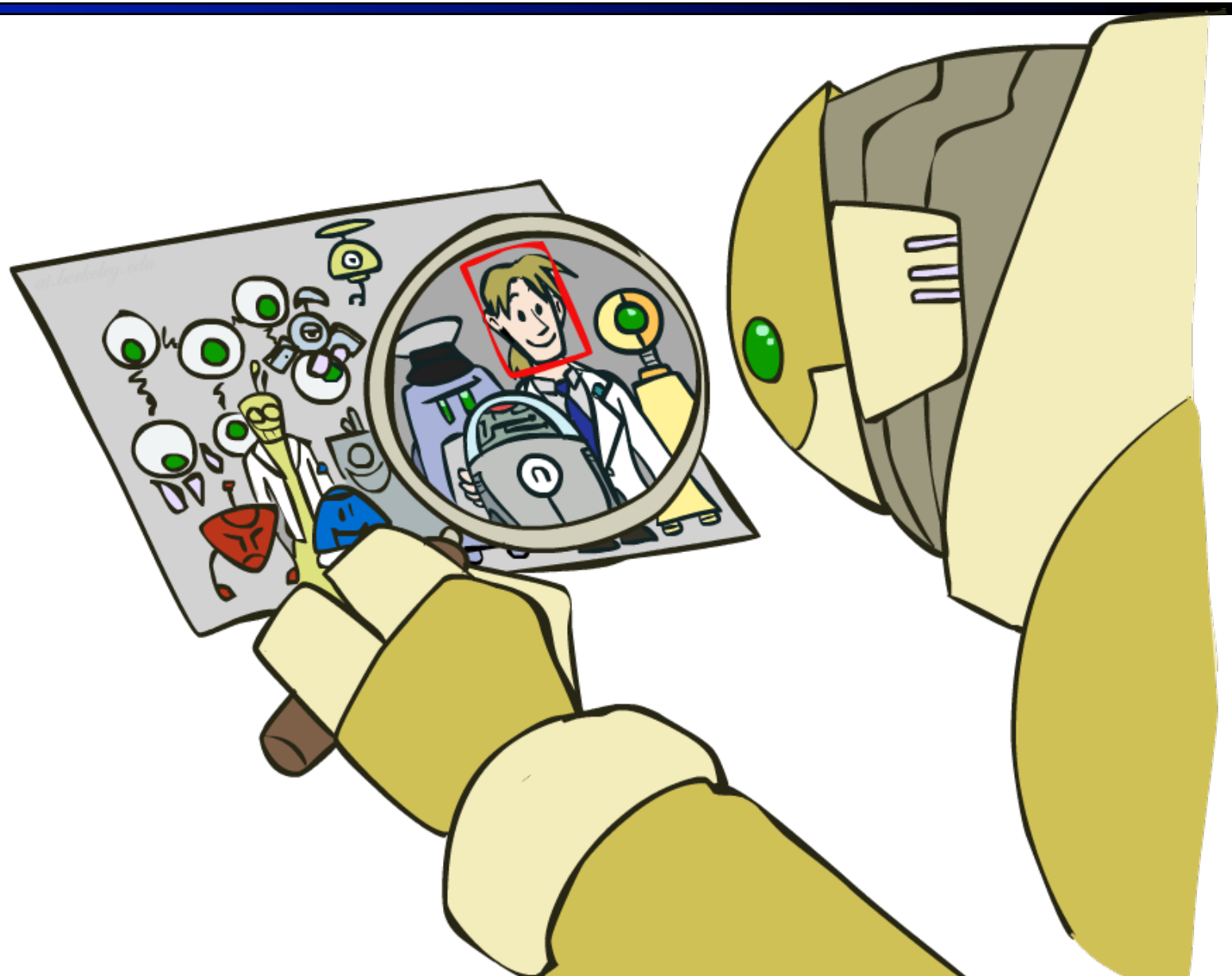
- Continuous optimization
  - Gradient ascent:
    - Compute steepest uphill direction = gradient (= just vector of partial derivatives)
    - Take step in the gradient direction
    - Repeat (until held-out data accuracy starts to drop = "early stopping")

- Deep neural nets
  - Last layer = still logistic regression
  - Now also many more layers before this last layer — learning features
    - = computing the features
    - → the features are learned rather than hand-designed
  - Universal function approximation theorem
    - `If` neural net is large enough
    - `Then` neural net can represent any continuous mapping from input to output with arbitrary accuracy
    - But remember: need to avoid overfitting / memorizing the training data → early stopping!
  - Automatic differentiation gives the derivatives efficiently (how? = outside of scope of 188)

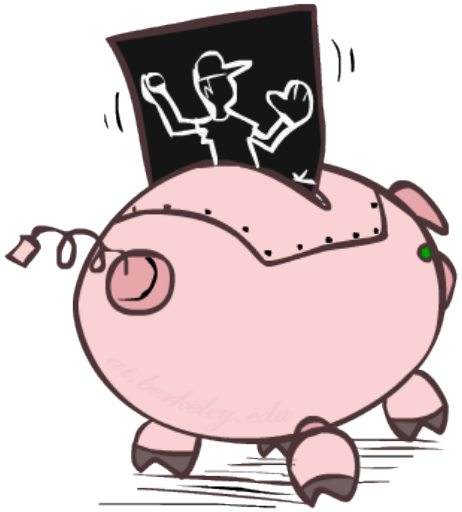
# Computer Vision

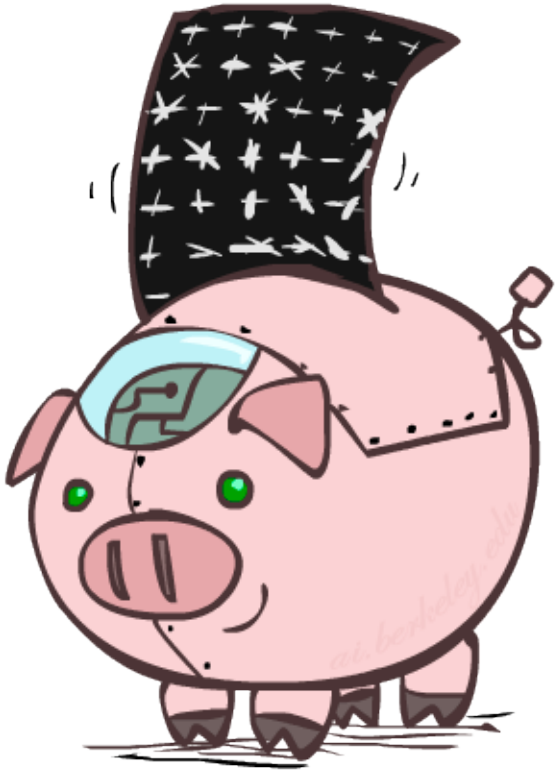


# Object Detection

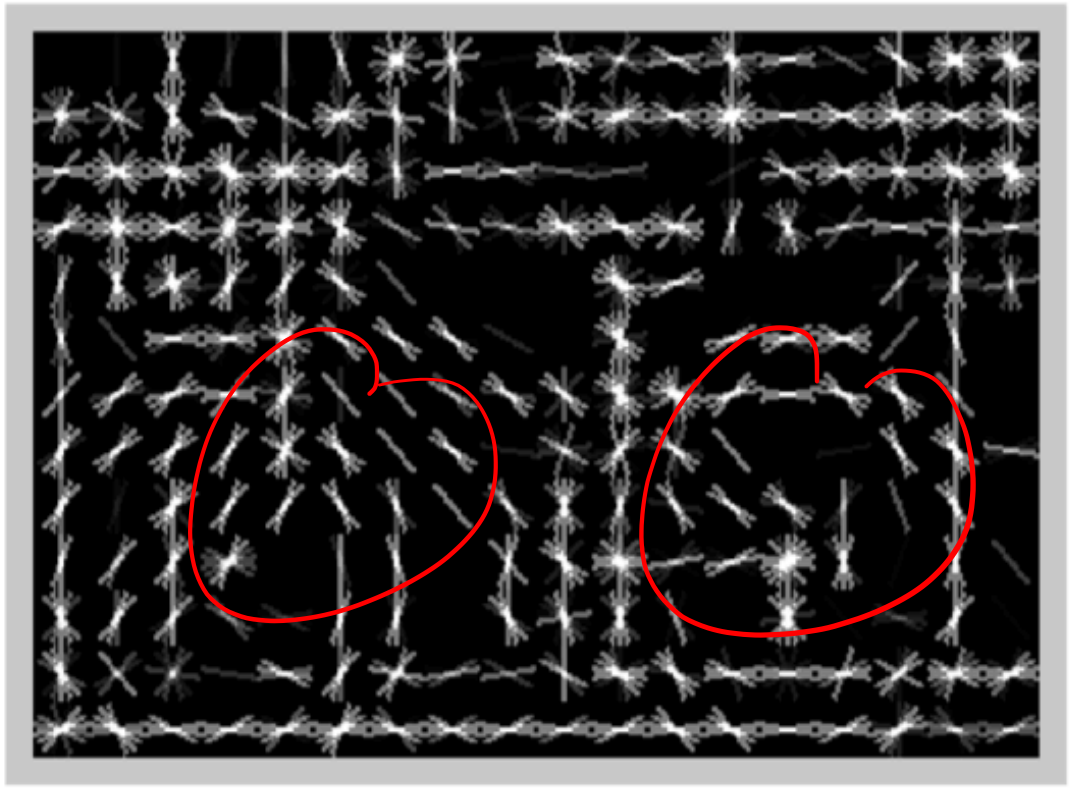


# Manual Feature Design





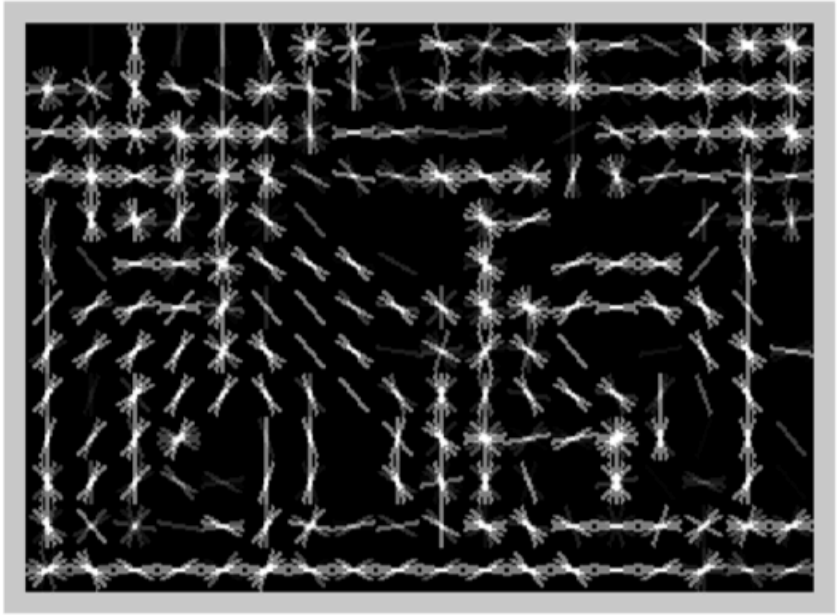
# Features and Generalization



[HoG: Dalal and Triggs, 2005]

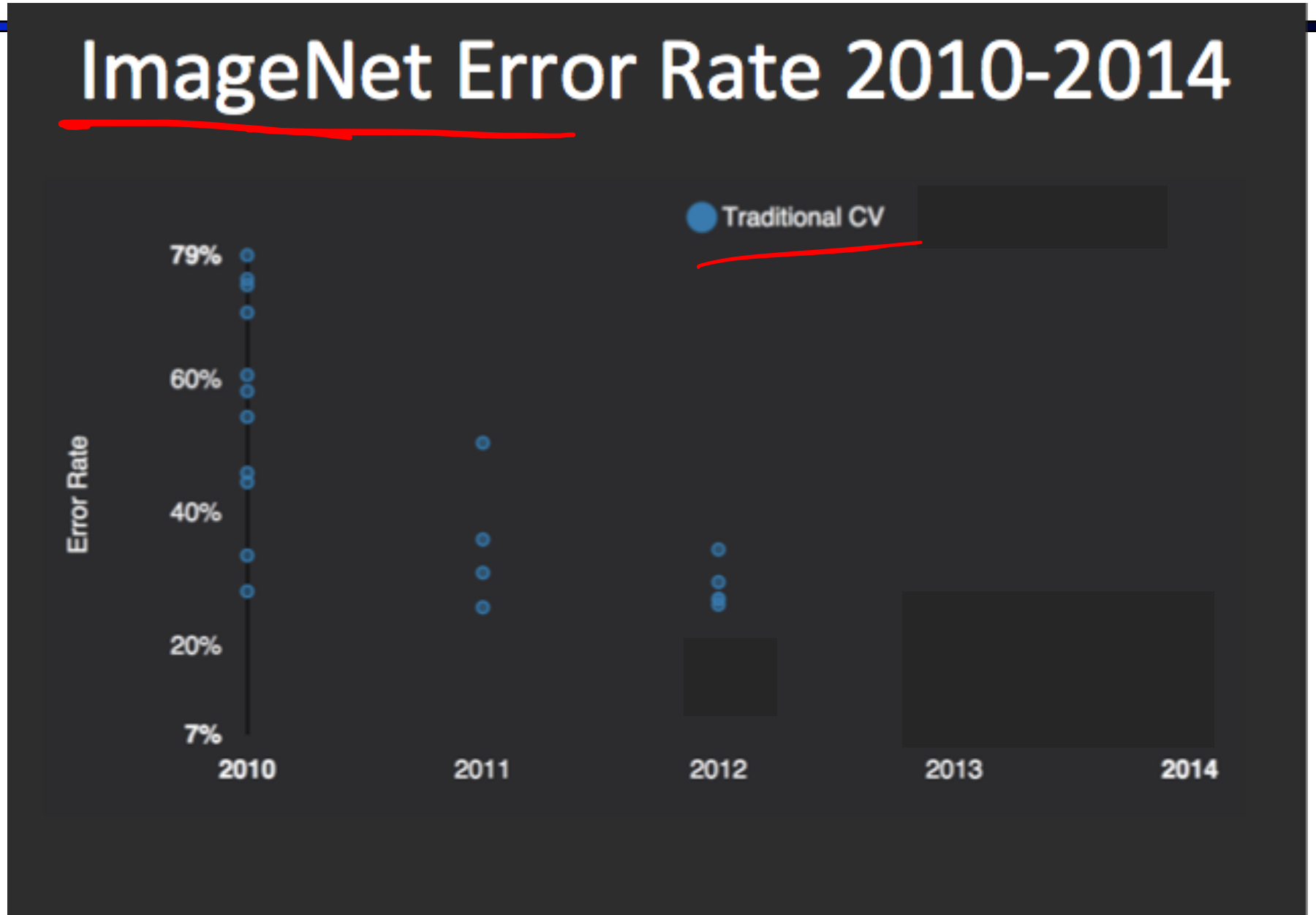
# Features and Generalization

Image



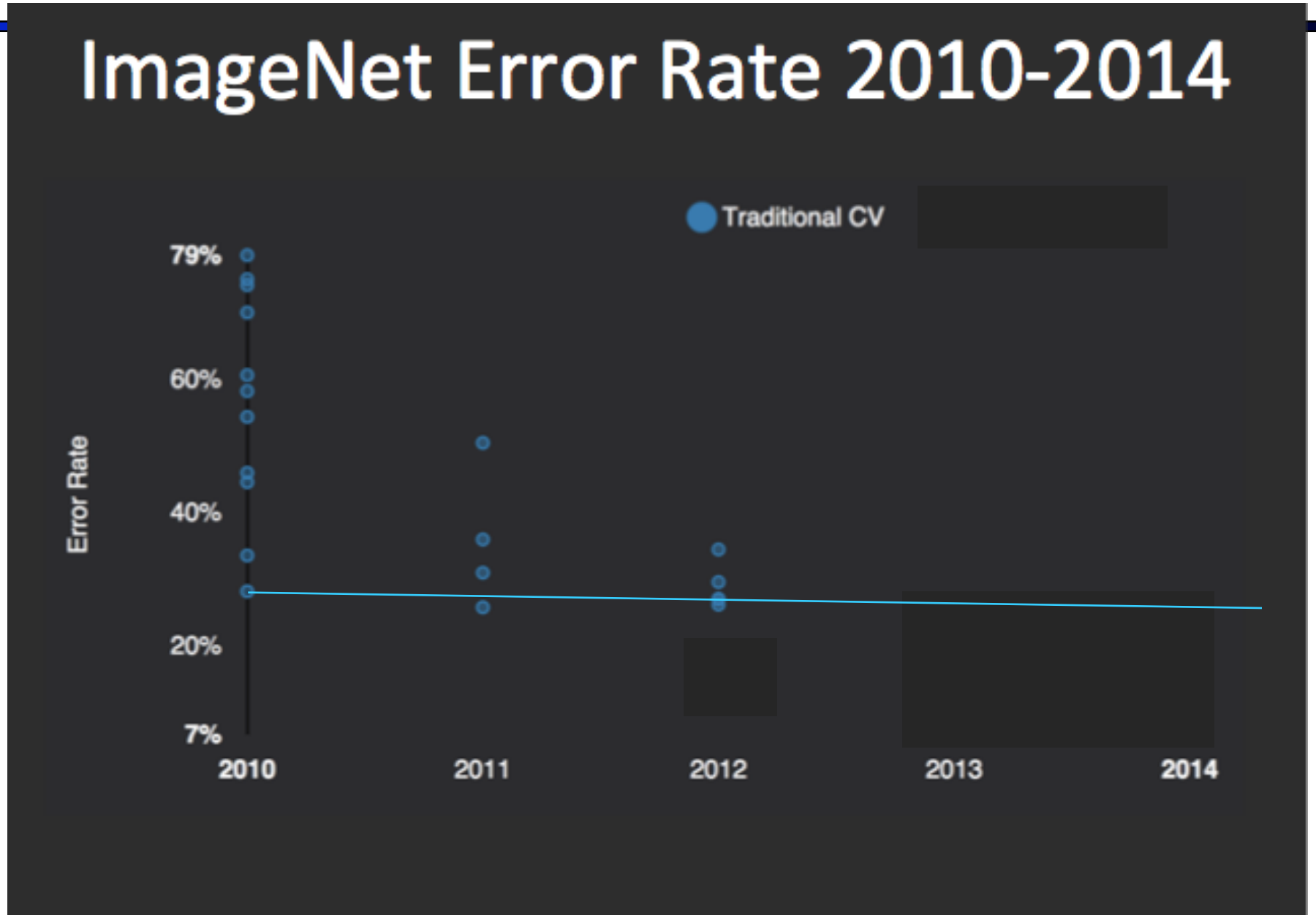
HoG

# Performance



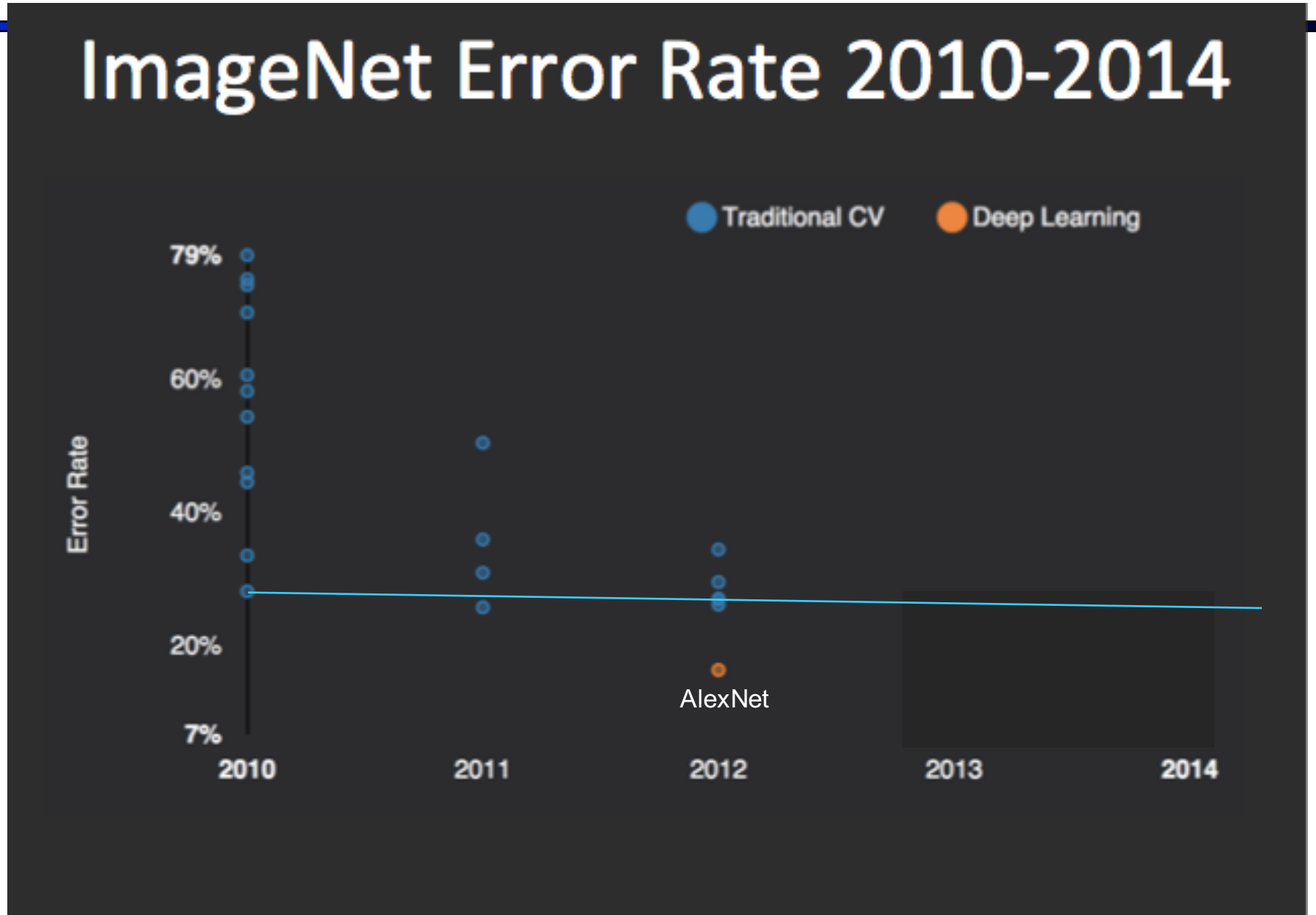

*graph credit Matt Zeiler, Clarifai*

# Performance



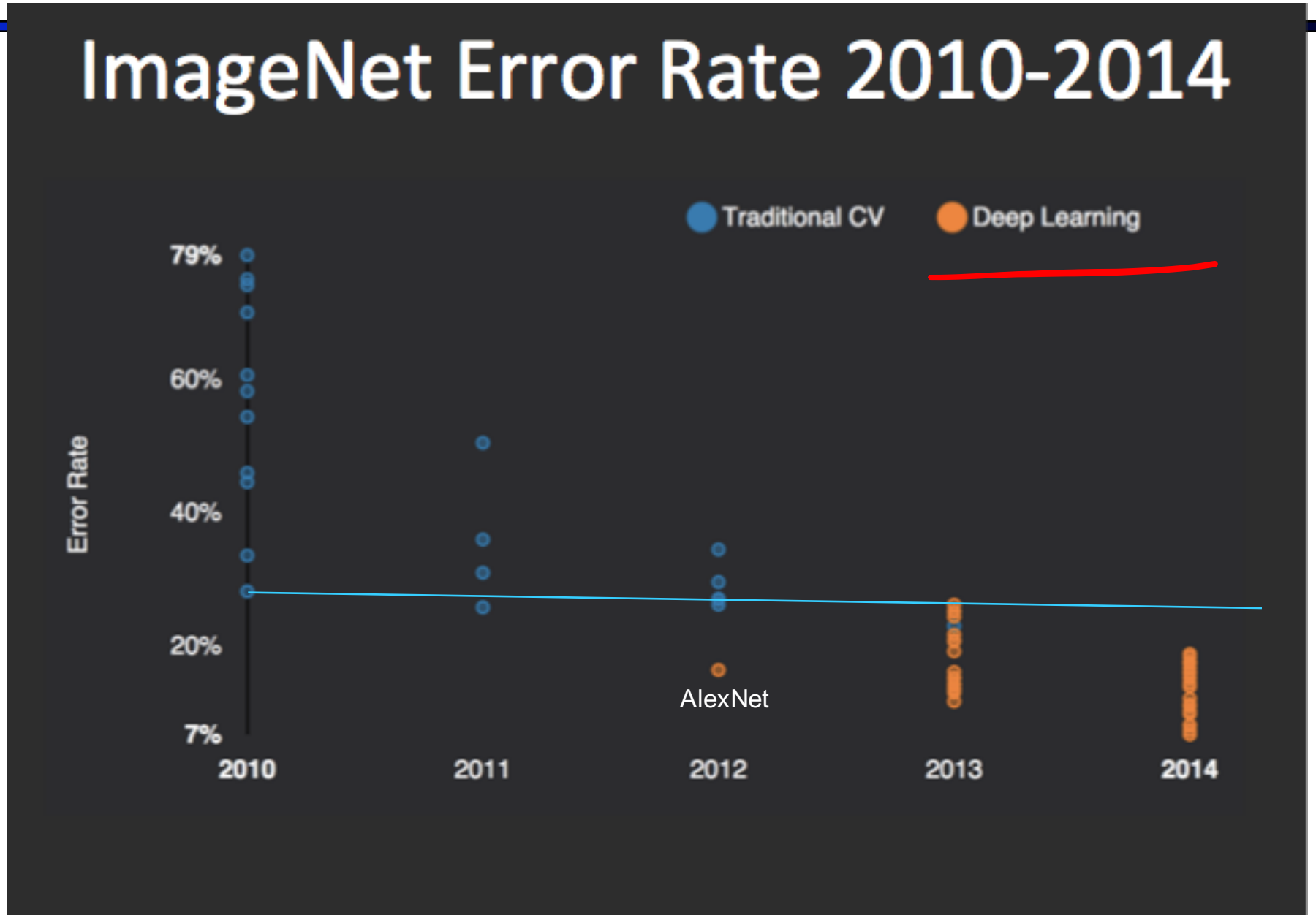

*graph credit Matt Zeiler, Clarifai*

# Performance



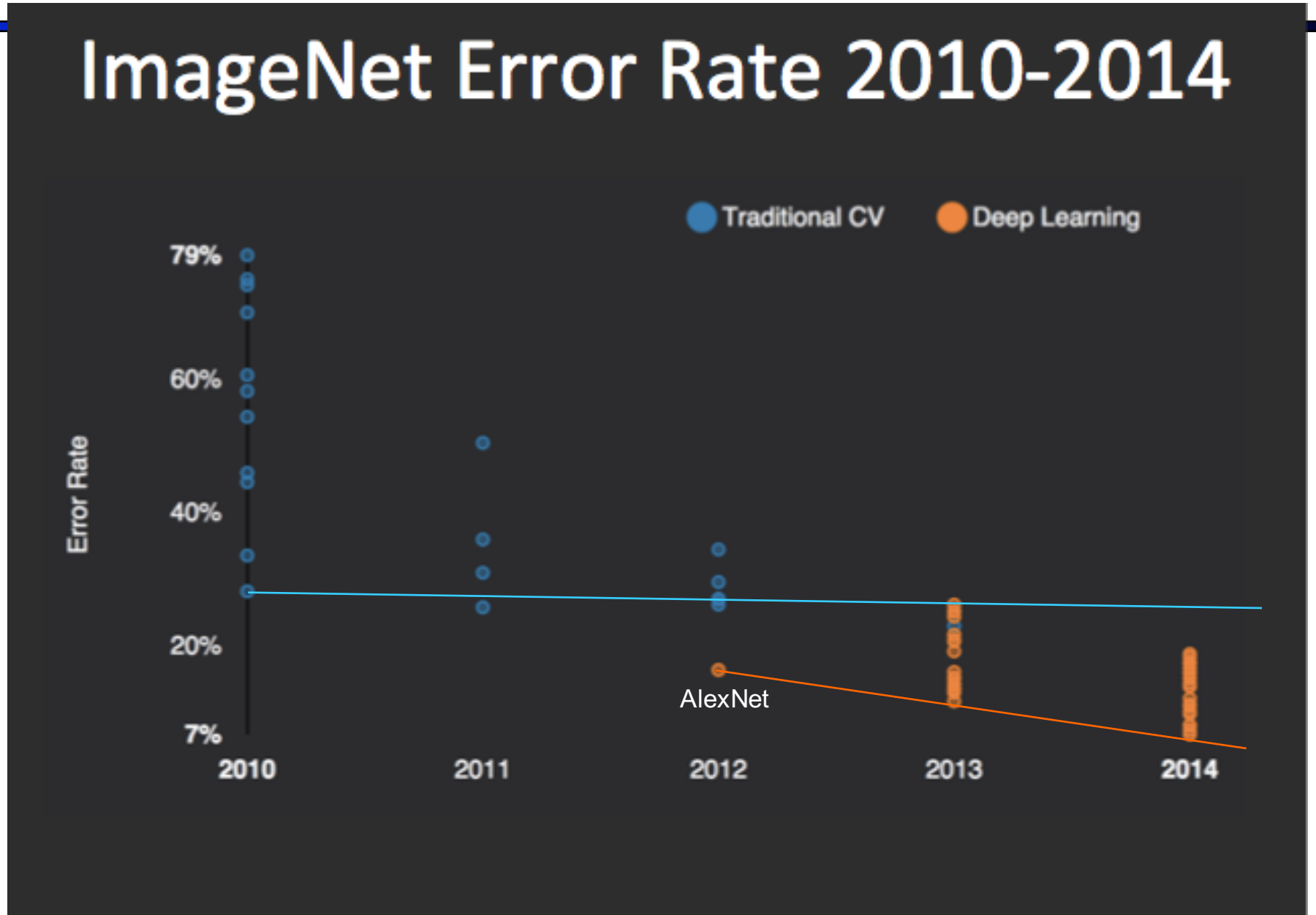

*graph credit Matt Zeiler, Clarifai*

# Performance




*graph credit Matt Zeiler, Clarifai*

# Performance




*graph credit Matt Zeiler, Clarifai*

# MS COCO Image Captioning Challenge

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

"boy is doing backflip on wakeboard."

"girl in pink dress is jumping in air."

"black and white dog jumps over bar."

"young girl in pink shirt is swinging on swing."

"man in blue wetsuit is surfing on wave."

Karpathy & Fei-Fei, 2015; Donahue et al., 2015; Xu et al, 2015; many more

# Visual QA Challenge

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh

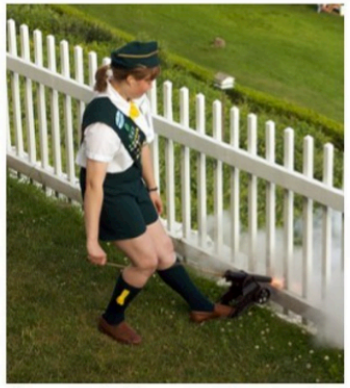
What vegetable is on the plate?
Neural Net: broccoli
Ground Truth: broccoli

What color are the shoes on the person's feet ?
Neural Net: brown
Ground Truth: brown

How many school busses are there?
Neural Net: 2
Ground Truth: 2

What sport is this?
Neural Net: baseball
Ground Truth: baseball

What is on top of the refrigerator?
Neural Net: magnets
Ground Truth: cereal

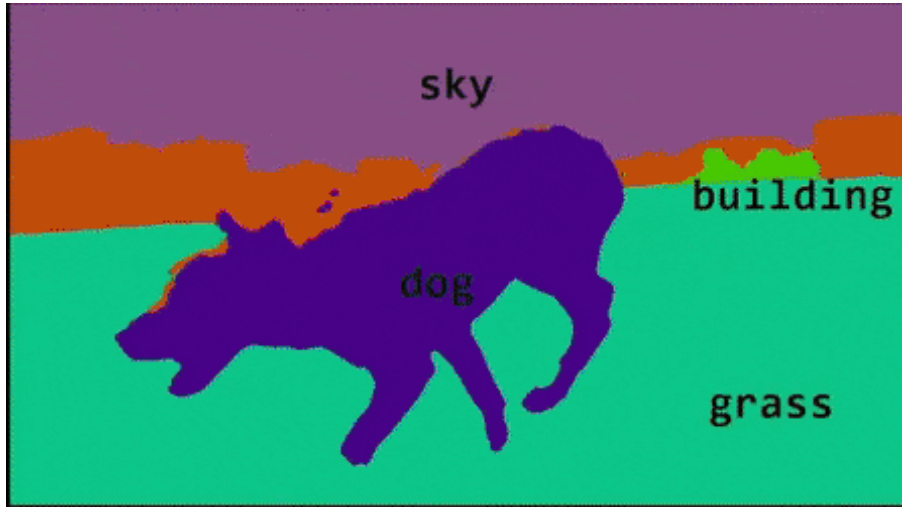


What uniform is she wearing?
Neural Net: shorts
Ground Truth: girl scout

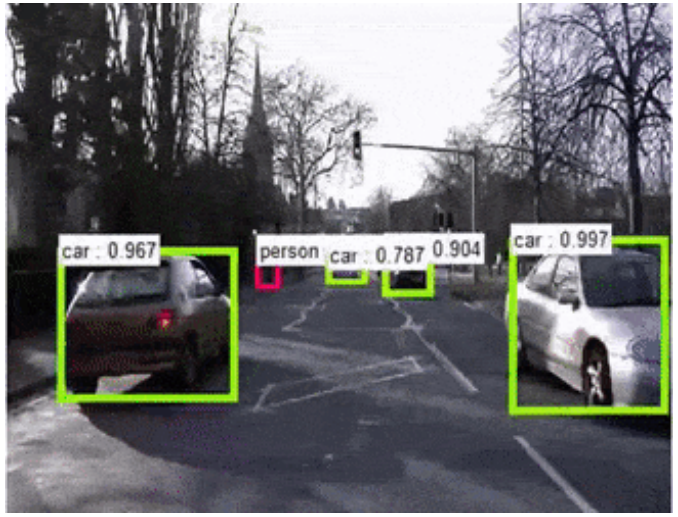
What is the table number?
Neural Net: 4
Ground Truth:40

What are people sitting under in the back?
Neural Net: bench
Ground Truth: tent

# Semantic Segmentation/Object Detection

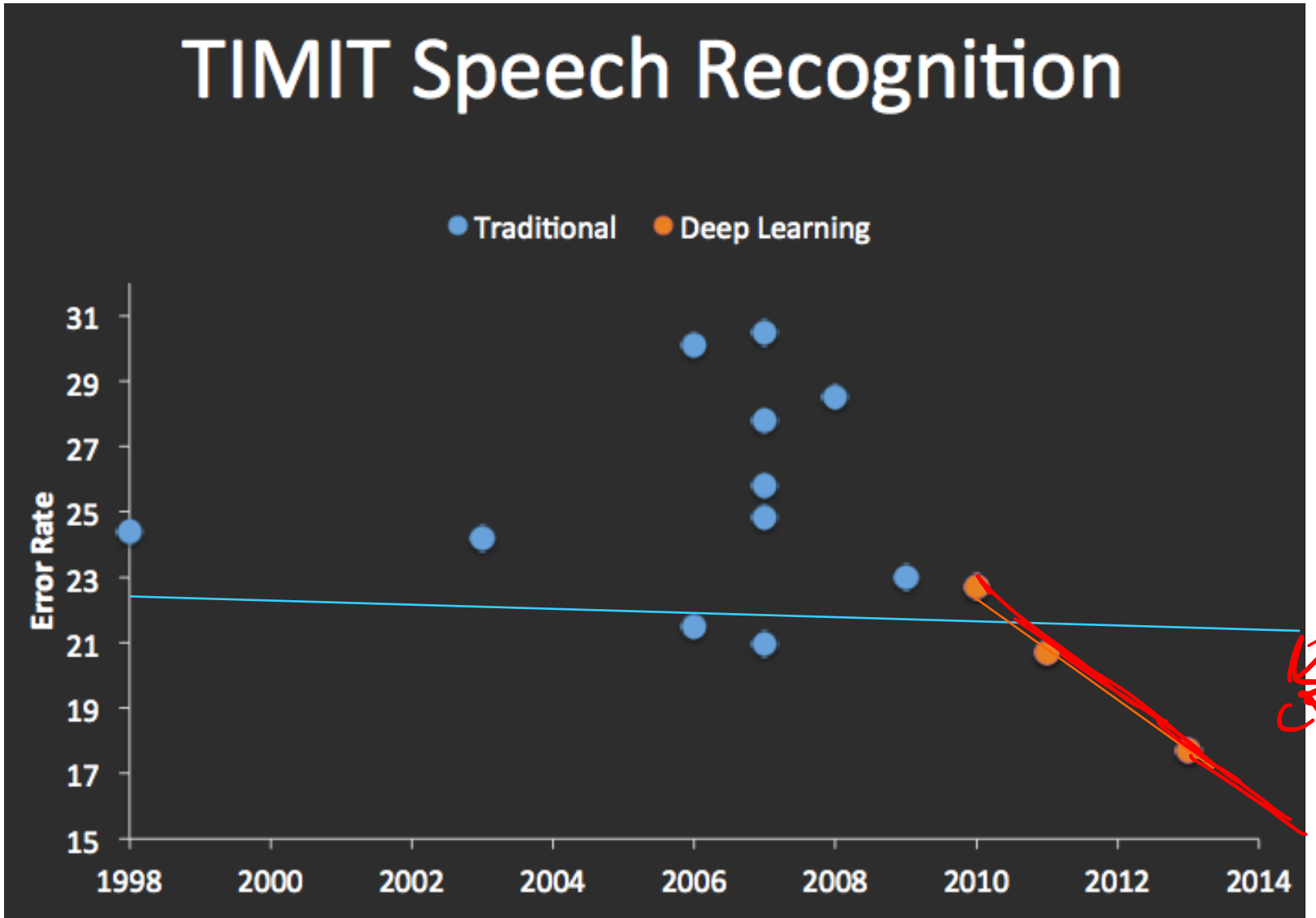


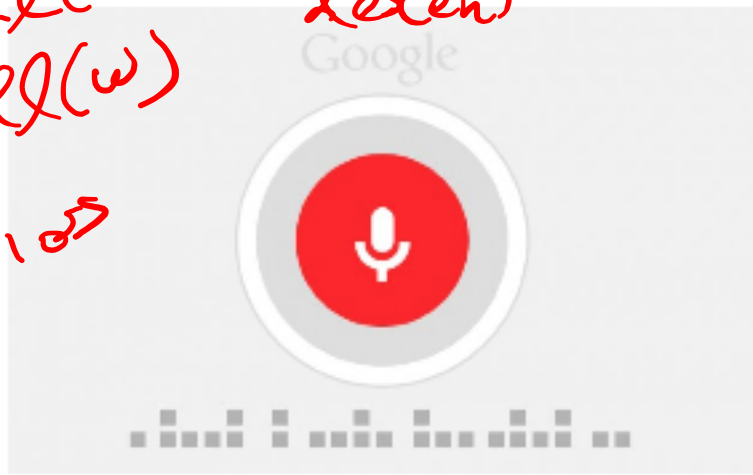
Semantic segmentation



object detection

# Speech Recognition






graph credit Matt Zeiler, Clarifai