

# Computational Intelligence

Samaneh Hosseini

Isfahan University of Technology

# Outline

- Multi-class classification
  - Softmax Regression
  - Training a Softmax Classifier
  - Loss Function
  - Gradient Descent with Softmax

# Multi-class classification: Softmax Regression

# Recognizing cats, dogs, and baby chicks



3

1

2

0

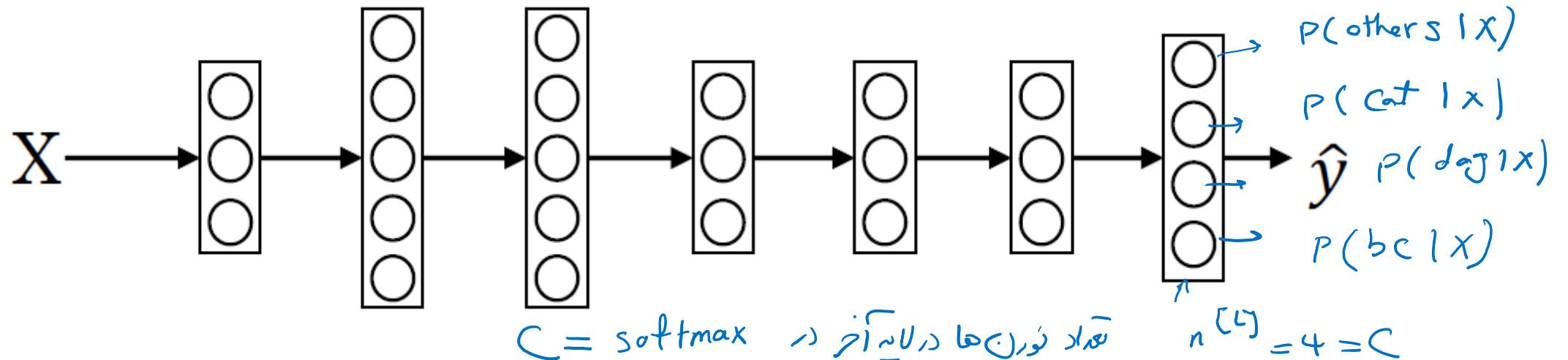
3

2

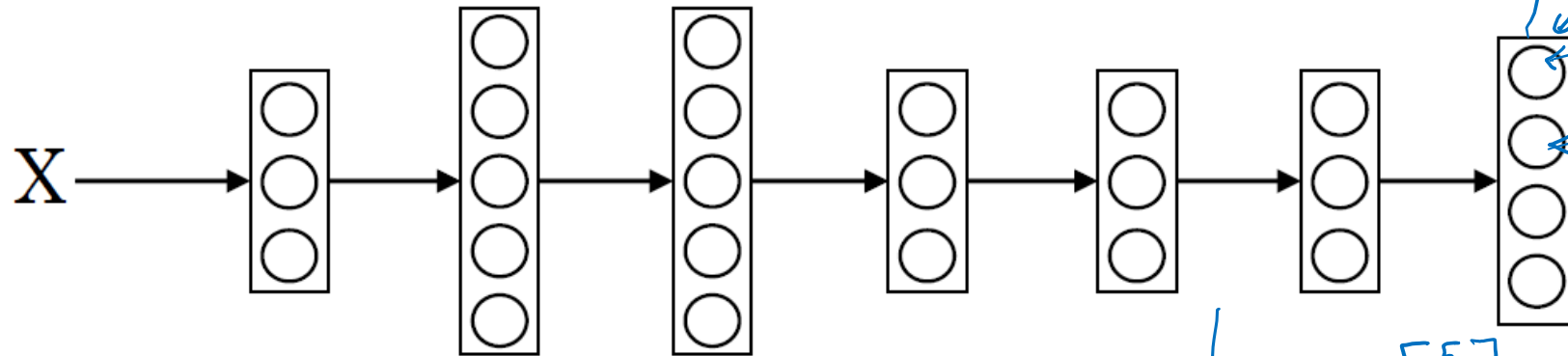
0

1

# classes = 4 = C



# Softmax layer



softmax layer

$$z^{[L]} = w^{[L]} a^{[L-1]} + b^{[L]} \quad (4 \times 1)$$

softmax activation function:

$$t = e^{z^{[L]}} \rightarrow (4 \times 1)$$

$$a^{[L]} = \frac{e^{z_i^{[L]}}}{\sum_{j=1}^4 t_j}, \quad a_i^{[L]} = \frac{t_i}{\sum_{j=1}^4 t_j}$$

softmax activation function

$$z^{[L]} = \begin{bmatrix} 5 \\ 2 \\ -1 \\ 3 \end{bmatrix}$$

$$t = \begin{bmatrix} e^5 \\ e^2 \\ e^{-1} \\ e^3 \end{bmatrix}$$

$$= \begin{bmatrix} 148.4 \\ 7.4 \\ 0.4 \\ 20.1 \end{bmatrix}$$

$$\sum_{j=1}^4 t_j = 176.3$$

$$a^{[L]} = \frac{t}{176.3}$$

$$\frac{e^5}{176.3} = 0.842$$

$$\frac{e^2}{176.3} = 0.042$$

$$\frac{e^{-1}}{176.3} = 0.002$$

$$\frac{e^3}{176.3} = 0.114$$

# Can You Use Argmax Activations Instead?

- The argmax function returns the index of the maximum value in the input array.
- Example:  $z = [5, 2, -1, 3]$  → image is predicted to be as others →  $a = [1, 0, 0, 0]$
- Limitation:
  - gradients with respect to the raw outputs of the neural networks are always zero.
  - No learning during backpropagation as gradients are zero.

# Can You Use Argmax Activations Instead?

- From a probabilistic viewpoint:
- notice how the argmax function puts all the mass on index 1:
- the predicted class and 0 elsewhere.
- it's straightforward to infer the predicted class label from the argmax output.
- However, we would like to know how likely the image is to be that of a cat, a dog,  
or a baby chicks,
- The softmax scores help us with just that!

# Tips on Softmax

- Applying softmax preserves the relative ordering of scores.
- All entries in the softmax output vector are between 0 and 1.
- the classes are mutually exclusive
  - One input can not belong to two class
  - entries of the softmax output sum up to 1

# Softmax Implementation

```
1 import numpy as np
2
3 def softmax(z):
4     '''Return the softmax output of a vector.'''
5     exp_z = np.exp(z)
6     sum = exp_z.sum()
7     softmax_z = np.round(exp_z/sum, 3)
8     return softmax_z
```

# Why Won't Normalization by the Sum Suffice

- Why use something math-heavy as the softmax activation? Can we not just divide each of the output values by the sum of all outputs?

```
1 | def div_by_sum(z):  
2 |     sum_z = np.sum(z)  
3 |     out_z = np.round(z/sum_z, 3)  
4 |     return out_z
```

# Why Won't Normalization by the Sum Suffice

1 Consider  $\mathbf{z1} = [0.25, 1.23, -0.8]$

```
1  z1 = [0.25, 1.23, -0.8]
2  div_by_sum(z1)
3
4  # Output
5  array([ 0.368,  1.809, -1.176])
```

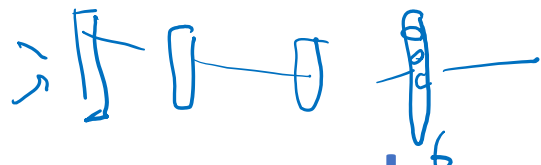
We still aren't able to interpret the entries as probability scores.

# Why Won't Normalization by the Sum Suffice

2 Let z2 = [-0.25, 1, -0.75].

```
1  z2 = [-0.25, 1, -0.75]
2  div_by_sum(z2)
3
4  # Output
5  RuntimeWarning: divide by zero encountered in true_divide
6  array([-inf, inf, -inf])
```

When you divide by the sum to normalize, you'll face runtime warnings, as division by zero is not defined.

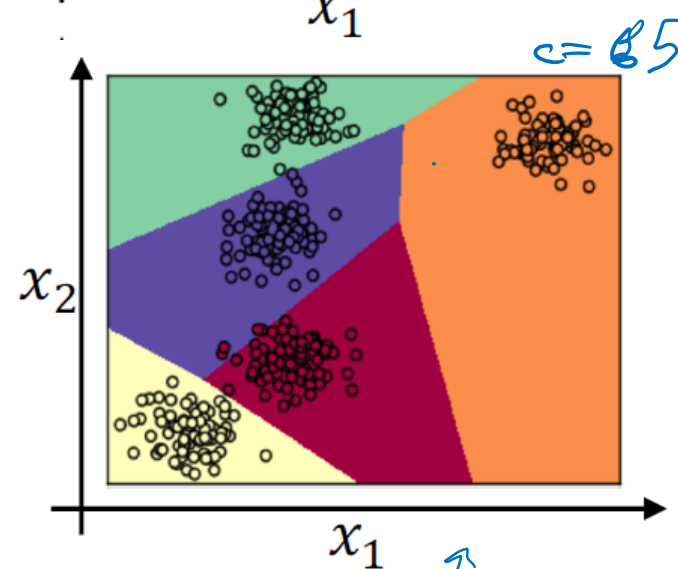
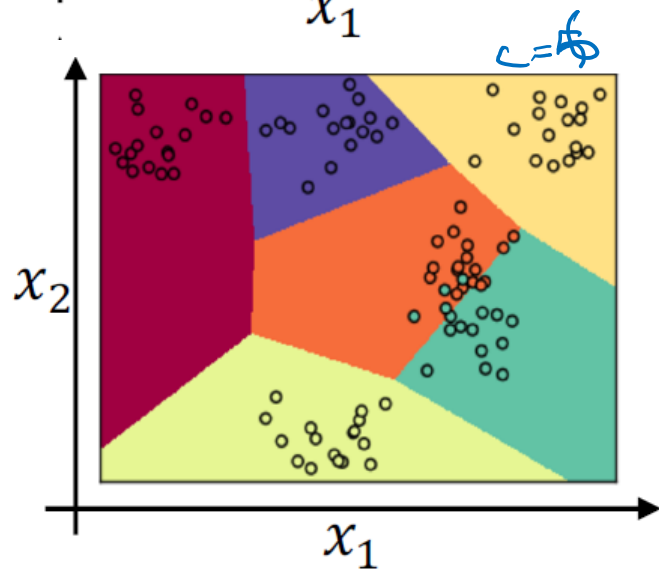
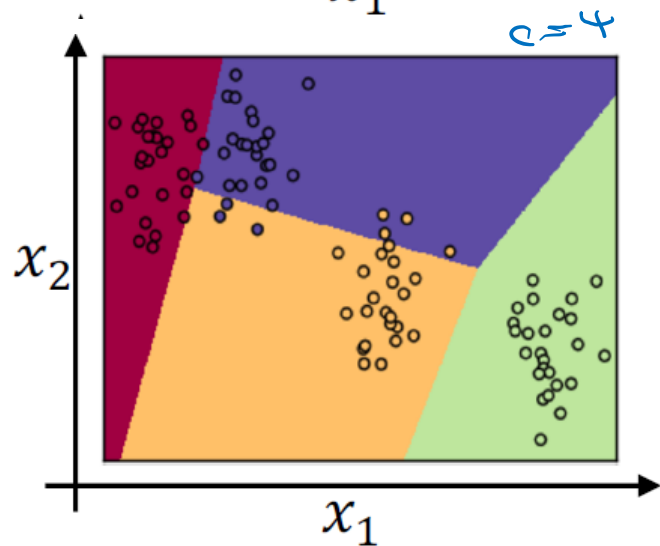
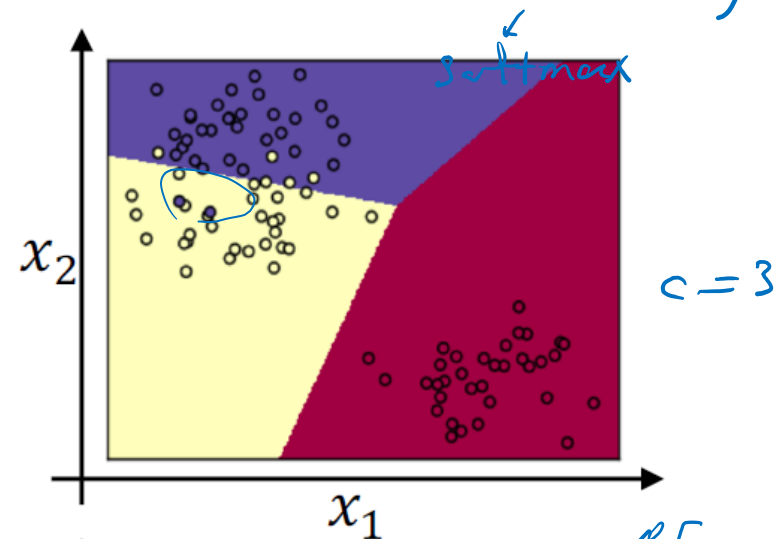
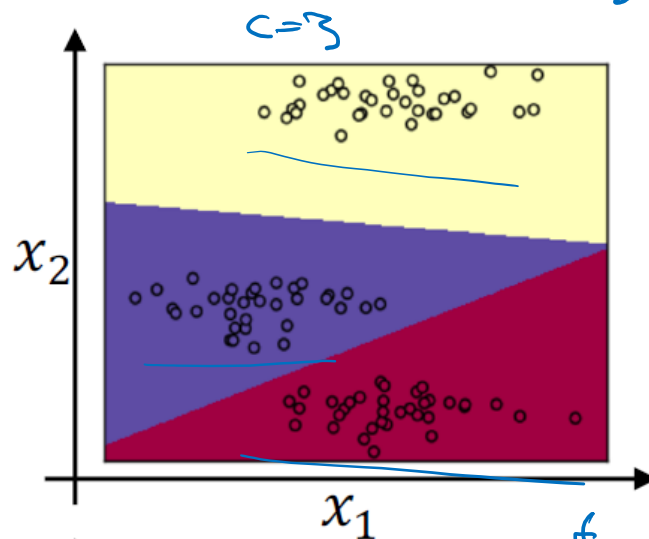
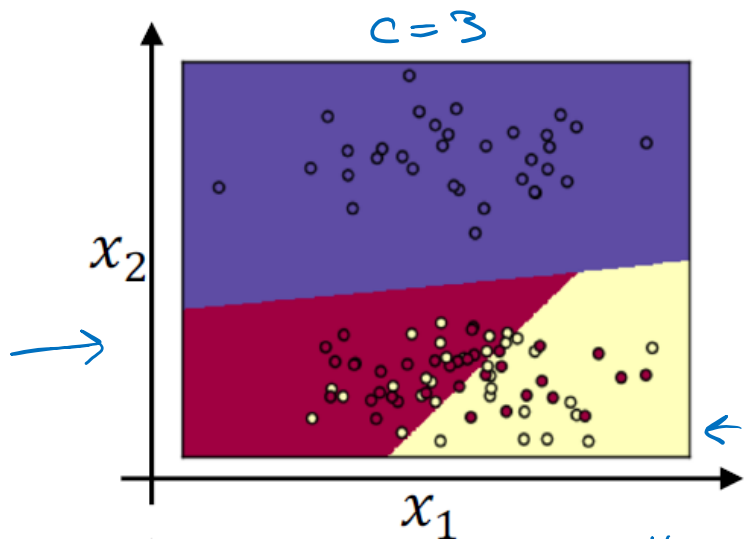


# Softmax examples

$$\begin{matrix} x_1 \\ x_2 \end{matrix} \rightarrow \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \rightarrow \hat{j}$$

$$z^{[1]} = w^{[1]}x + b^{[1]}$$

$$a^{[1]} = \hat{j} = g(z^{[1]})$$



# Multi-class classification: Training a Softmax Classifier

# Understanding Softmax

$$z^{[L]} = \begin{bmatrix} 5 \\ 2 \\ -1 \\ 3 \end{bmatrix}$$

$$t = \begin{bmatrix} e^5 \\ e^2 \\ e^{-1} \\ e^3 \end{bmatrix}$$

$$C=4$$

$$g^{[L]}(\cdot) = \frac{e^{z^{[L]}}}{\sum_{j=1}^C e^{z_j^{[L]}}}$$

$$a^{[L]} = g^{[L]}(z^{[L]}) = \begin{bmatrix} \frac{e^5}{e^5 + e^2 + e^{-1} + e^3} \\ \frac{e^2}{e^5 + e^2 + e^{-1} + e^3} \\ \frac{e^{-1}}{e^5 + e^2 + e^{-1} + e^3} \\ \frac{e^3}{e^5 + e^2 + e^{-1} + e^3} \end{bmatrix} = \begin{bmatrix} 0.842 \\ 0.042 \\ 0.002 \\ 0.114 \end{bmatrix}$$

hard max

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Softmax regression generalized form of logistic regression

$$\text{If } C=2$$

$$\begin{bmatrix} \frac{e^1}{e^1 + e^2} \\ \frac{e^2}{e^1 + e^2} \end{bmatrix} \neq \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 1-0 \end{bmatrix}$$

# Understanding Softmax

There's only one node as the other is not given any weight at all. The raw output vector now becomes  $\mathbf{z} = [z, 0]$ .

$$\begin{aligned} \text{softmax}(\mathbf{z})_0 &= \frac{e^z}{e^0 + e^z} = \frac{e^z}{1 + e^z} = \sigma(z) \\ \text{softmax}(\mathbf{z})_1 &= \frac{e^0}{e^0 + e^z} = \frac{1}{1 + e^z} = 1 - \sigma(z) \end{aligned}$$

# Loss Function

$$y^{(1)} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \rightarrow \text{cat}$$

$$a^{[1]} = \hat{y}^{(1)} = \begin{bmatrix} 0.3 \\ 0.2 \\ 0.1 \\ 0.4 \end{bmatrix}$$

$$L(\hat{y}, y) = - \sum_{j=1}^4 y_j \log \hat{y}_j$$

Small

$$-y_2 \log \hat{y}_2 = -\log \hat{y}_2$$

Small

$\Rightarrow \hat{y}_2$  big

$$Y = \begin{bmatrix} y^{(1)} & y^{(2)} & \dots & y^{(m)} \end{bmatrix}_{4 \times m}$$

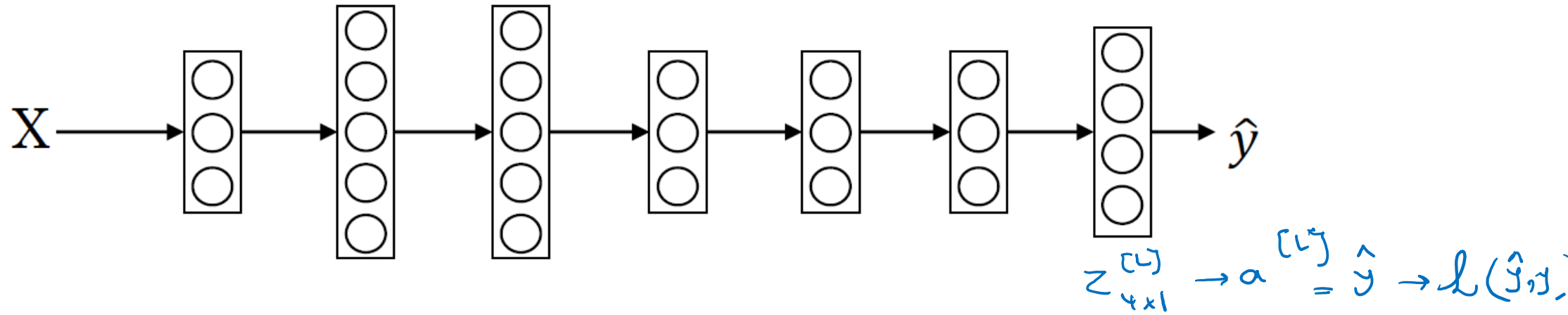
$$= \begin{bmatrix} 0 & 0 & 1 & \dots \\ 1 & 0 & 0 & \dots \\ 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & \dots \end{bmatrix}$$

$$J(\omega^{[1]}, b^{[1]}, \dots) = \frac{1}{m} \sum_{i=1}^m L(\hat{y}^{(i)}, y^{(i)})$$

$$\hat{Y} = \begin{bmatrix} \hat{y}^{(1)} & \dots & \hat{y}^{(m)} \end{bmatrix}$$

$$= \begin{bmatrix} 0.3 & \dots & \dots \\ 0.2 & \dots & \dots \\ 0.1 & \dots & \dots \\ 0.4 & \dots & \dots \end{bmatrix}_{4 \times m}$$

# Gradient Descent with Softmax



Backprop :  $\frac{\partial \mathcal{L}}{\partial z^{[L]}} = \hat{y} - y$

# Core foundation Review

- Multi-class classification
  - Softmax Regression
  - Training a Softmax Classifier

# Core Foundation Review

- Multi-class classification
  - Softmax Regression
  - Training a Softmax Classifier
  - Loss Function
  - Gradient Descent with Softmax