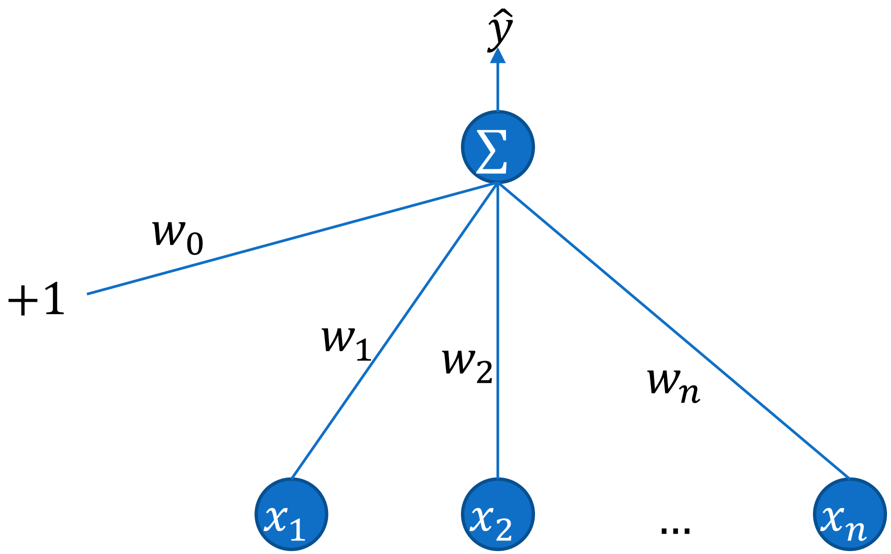
Review

LINEAR AND LOGISTIC REGRESSION

A Visual Perspective

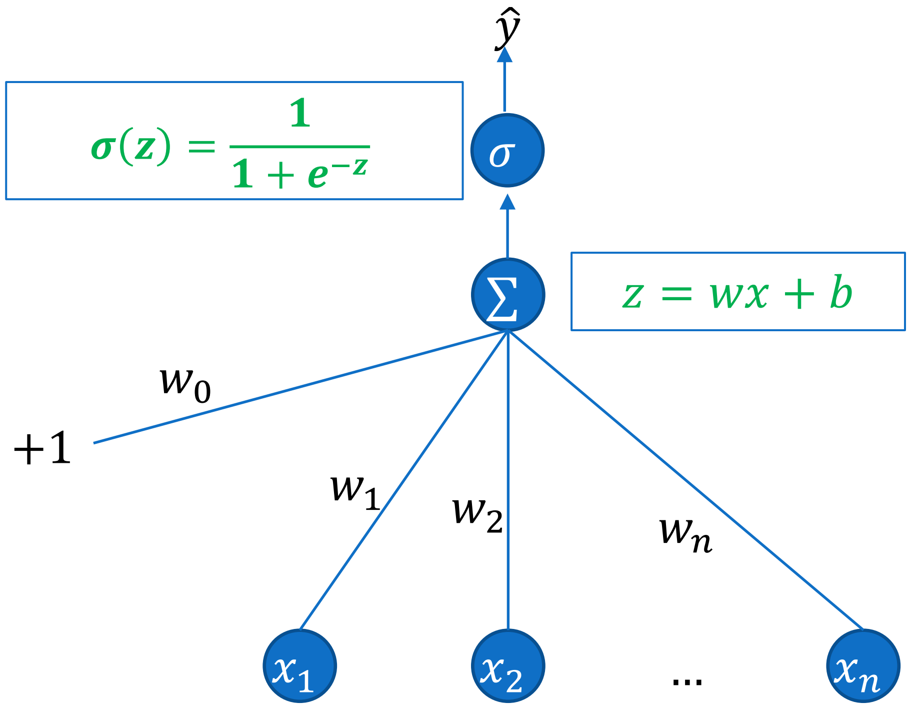
Linear Regression

Compute Error: $y - \hat{y}$



Logistic Regression

Compute Error: $y - \hat{y}$



Softmax

- Recall that sigmoid produces a number between 0 and 1
  - The sum of probabilities of an email being either spam or not spam is 1.0

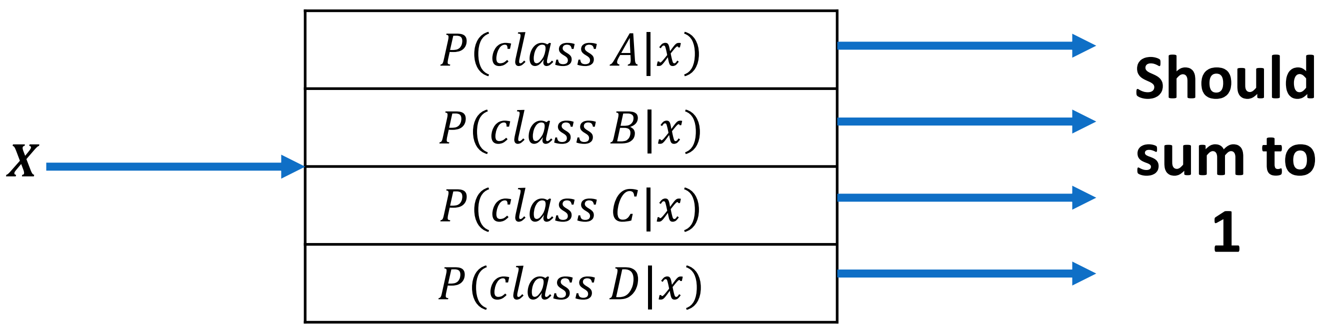
- Softmax extends this idea into a multi-class world.
  - It assigns a probability to each class in a multi-class problem and the probabilities across classes add up to 1.0
  - Works only when each example is a member of only one class.
    - If this is not true, rely on multiple logistic regression (oVr, oVo) i.e., multilabel classification

One VS All/Rest:

cat, not_cat	***white, not_white***
$\uparrow$	$\uparrow$
Classifier 1	Classifier 2

Generalizing Logistic Regression

- For multiple classes ($\#classes > 2$), we can generalize logistic regression to **Softmax regression**
  - One-of-c classes
    - E.g., 4 classes (0, 1, 2, 4)



- Softmax converts these C raw scores to probabilities that sum to 1.0

Softmax Activation Function

$$z = wx + b$$

$$S(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}$$

$$S(z)_i = \frac{e^{z_i}}{e^{z_1}, +e^{z_2}, \dots, +e^{z_C}}$$

Note: For 2 classes, the softmax is identical to sigmoid.

$$S(z)_1 = \frac{e^{z_1}}{e^{z_1}, e^{z_2}} = \sigma(z)$$

$$S(z)_2 = \frac{e^{z_2}}{e^{z_1}, e^{z_2}} = 1 - \sigma(z)$$

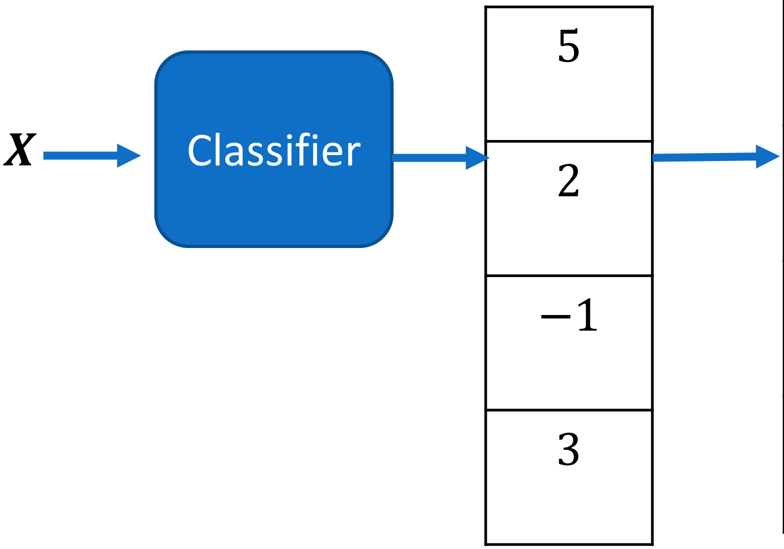
Softmax Activation Function

$$z = wx + b$$

$$S(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}$$

$$S(z)_i = \frac{e^{z_i}}{e^{z_1}, + e^{z_2}, \ldots, + e^{z_C}}$$

How softmax helps in computing loss?



Softmax

$$\frac{e^5}{e^5 + e^2 + e^{-1} + e^3} = 0.842$$

$$\frac{e^2}{e^5 + e^2 + e^{-1} + e^3} = 0.042$$

$$\frac{e^{-1}}{e^5 + e^2 + e^{-1} + e^3} = 0.002$$

$$\frac{e^3}{e^5 + e^2 + e^{-1} + e^3} = 0.114$$

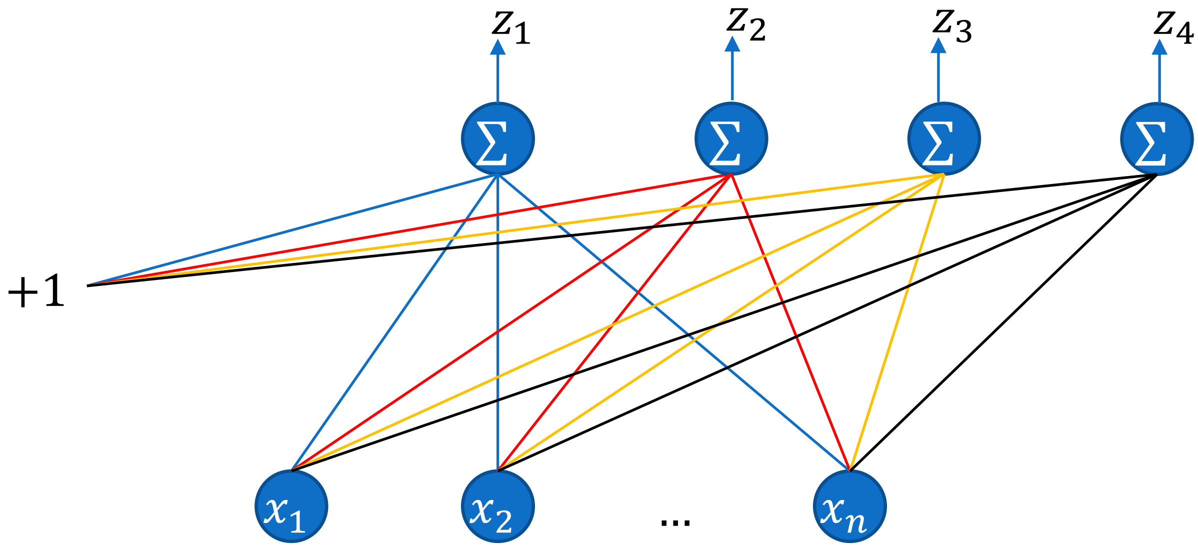
Hardmax

1
0
0
0

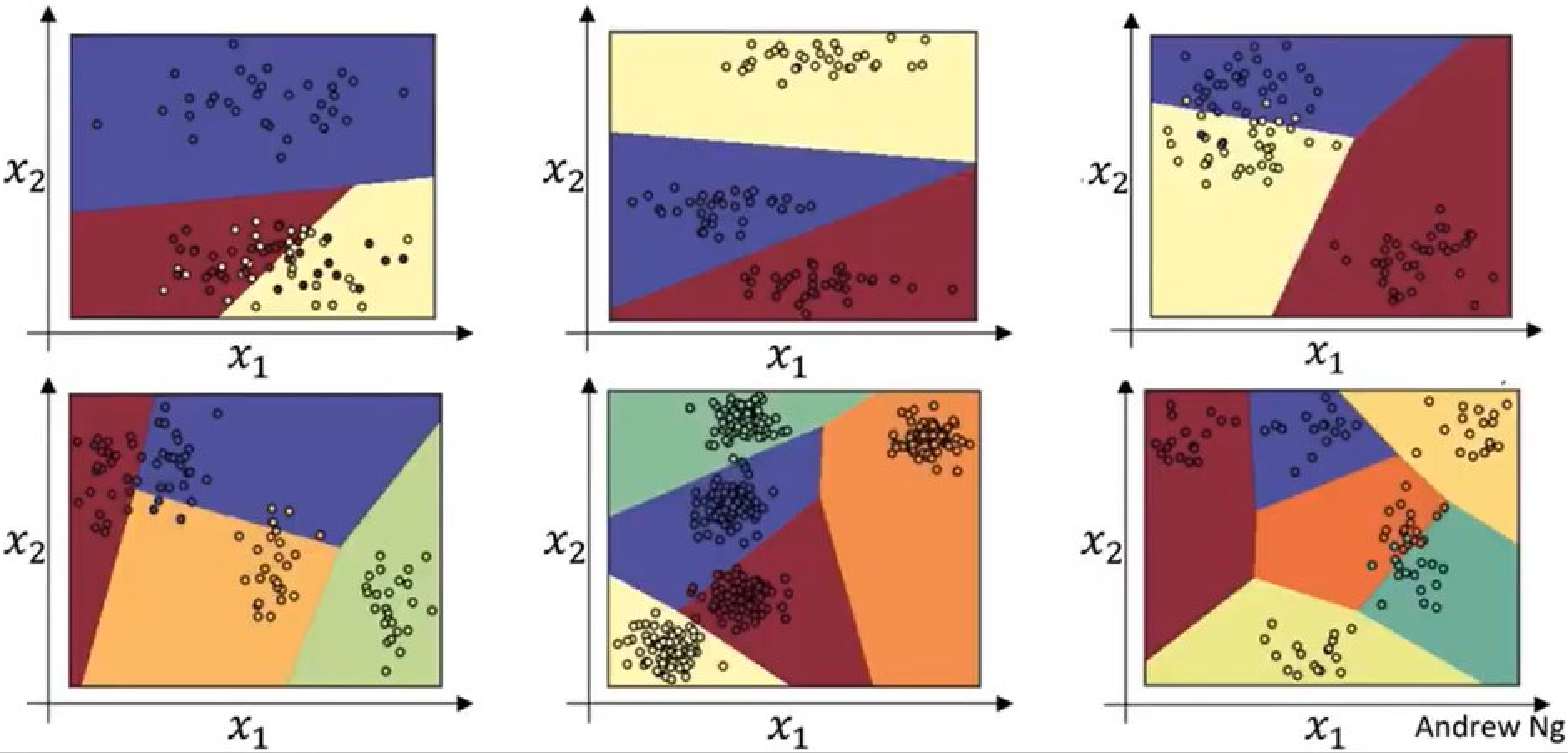
Softmax: A Visual Perspective

Compute Error

$$S(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}$$



Softmax Decision Boundaries



Loss Function

Actual labels are one-hot-encoded. $\rightarrow$ $y = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \hat{y} = \begin{bmatrix} 0.3 \\ 0.2 \\ 0.1 \\ 0.4 \end{bmatrix}$

$$\boxed{Loss(y, \hat{y}) = -\sum_{j=1}^{C} y_j log\hat{y}_j}$$

$$Loss(y, \hat{y}) = -(0.\, log\hat{y}_1 + 1.\, log\hat{y}_2 + 0.\, log\hat{y}_3 + 0.\, log\hat{y}_4)$$

$$Loss(y, \hat{y}) = -(1.\, log\hat{y}_2) = -log\hat{y}_2$$

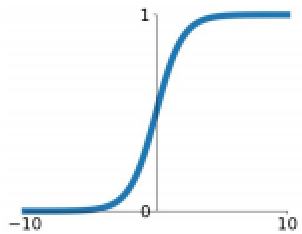
$$\boxed{J(y, \hat{y}) = \frac{1}{m}\sum_{i=1}^{m} Loss(y, \hat{y})}$$

Use gradient descent to adjust weights once you have cost.

Other Activation Functions

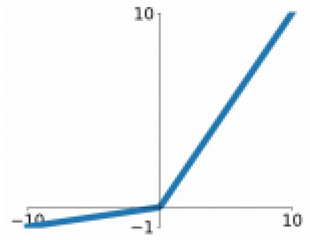
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



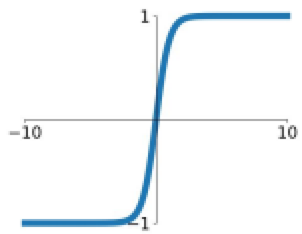
Leaky ReLU

$\max(0.1x, x)$



tanh

$\tanh(x)$

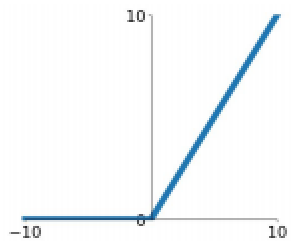


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

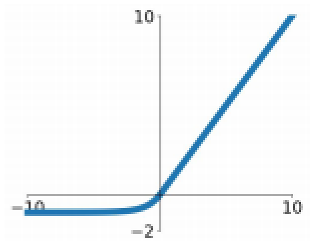
ReLU

$\max(0, x)$



ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



One Hot Encoding (aka categorical encoding)

x_1	x_2	y
5	9	0
6	8	0
1	2	1
11	12	2

y *One Hot Encoded*
$[1, 0, 0]$
$[1, 0, 0]$
$[0, 1, 0]$
$[0, 0, 1]$

$\hat{y}$ *After Softmax*
$[0.9, 0.1, 0]$
$[0.8, 0.2, 0]$
$[0.1, 0.75, 0.15]$
$[0, 0.05, 0.95]$

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

$$J(y, \hat{y}) = \frac{1}{m} \sum_{i=1}^{m} Loss(y, \hat{y})$$

Assignment 3 – Task 3

- Implement Linear regression for your age prediction dataset using TensorFlow and train on your training split.
- Implement Multi-class classification using softmax logistic regression for emotion recognition labels in your dataset using TensorFlow. Train on your training split.
- Use appropriate loss functions and activation functions.
- Train both models for 50 epochs.
- Use checkpointing to save the model on the epoch where your model has minimum loss based on the test split.
- Compute performance metrics for both above trained models on test split.

Book Reading

- Murphy – Chapter 8
- Jurafsky – Chapter 5, Chapter 4