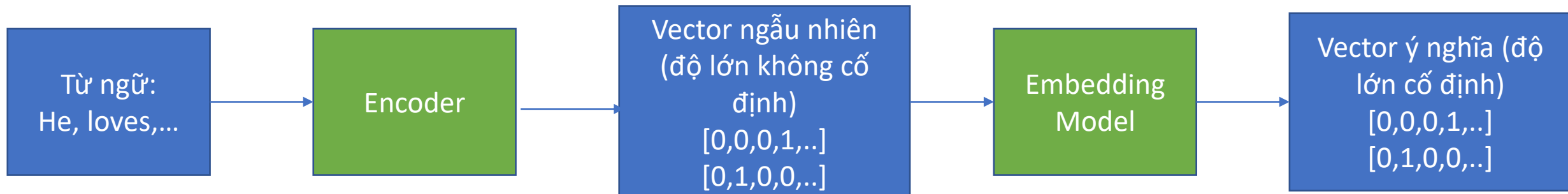
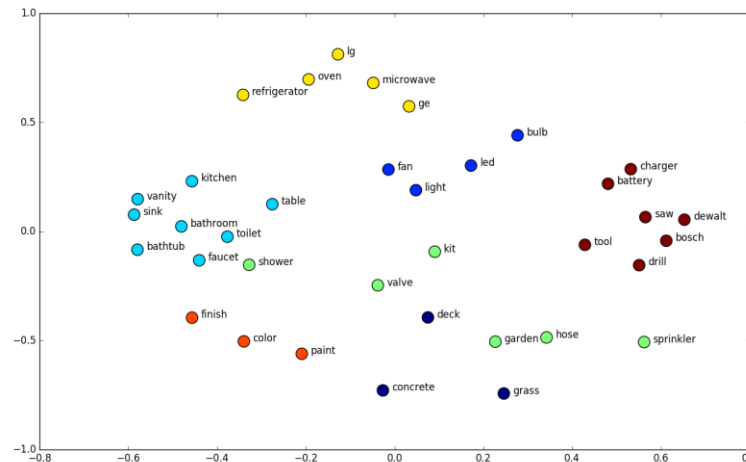


# Embedding Recap

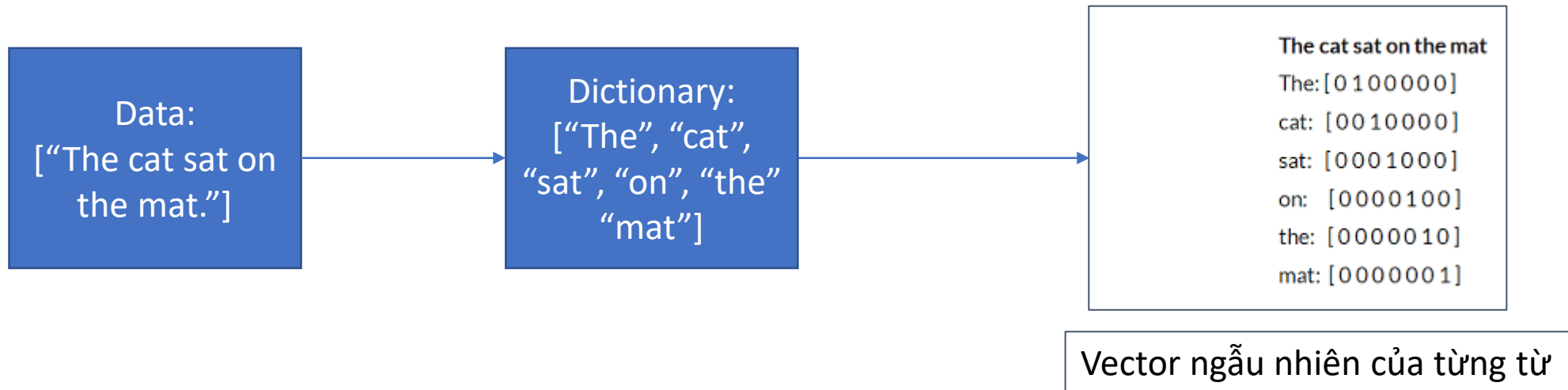
# 1.1 Model embedding là gì?

- Là một mô hình AI đưa vector ngẫu nhiên của từ về vector có ý nghĩa.



# Encoder (one-hot encoder)

- Là một thuật toán chuyển đổi giữa dạng từ (string) sang dạng vector.
- Dựa vào thứ tự từ đó trong bộ từ điển (dictionary).



- Độ lớn của vector ngẫu nhiên sẽ phụ thuộc vào số lượng từ ở dictionary

# Model Embedding được tạo ra như thế nào?

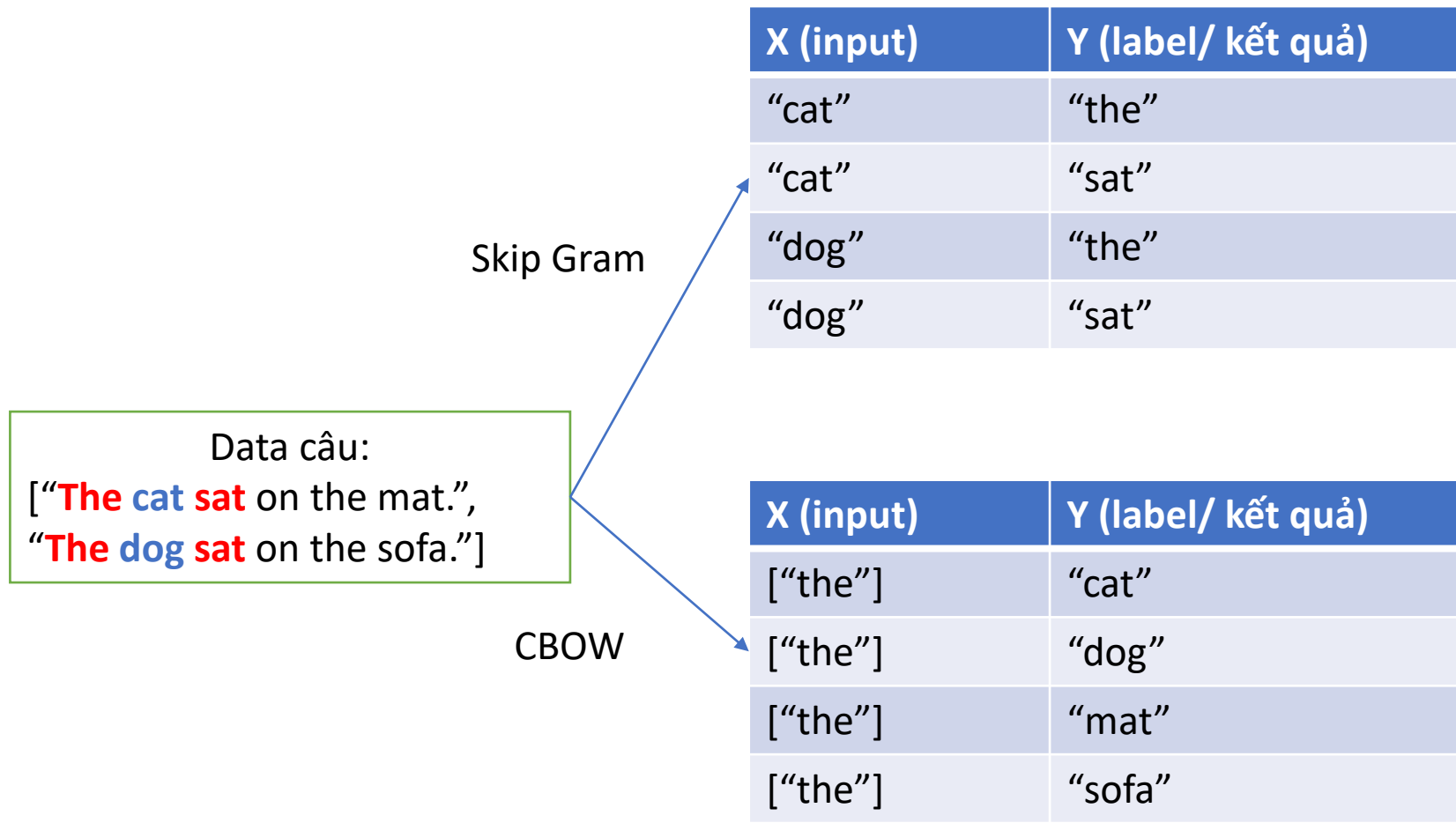
- Embedding là mô hình thể hiện được sự tương quan giữa những từ có ý nghĩa/trong ngữ cảnh tương tự.
- Để tìm ra sự tương quan giữa những từ, ta thấy rằng:
  - Các từ **tương tự** có những từ **xung quanh giống nhau**.

Loves  
I Likes you  
hates

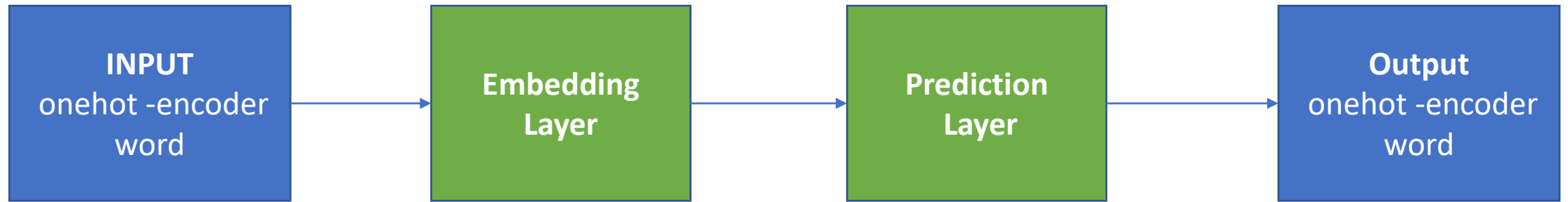
- Các từ **màu đỏ** là context words (từ ngữ cảnh)
- Các từ **màu xanh** là target words (những từ có embedding giống nhau)

# Model Embedding được tạo ra như thế nào?

- Cho nên, ta sẽ dựa theo những từ xung quanh để train ra 2 loại mô hình:
  - 1. Dự đoán những từ xung quanh (Skip Gram): **target** word: input, **context** word: label
  - 2. Dự đoán từ tiếp theo (CBOW): **target** word: label, **context** word: input



# Model Embedding được tạo ra như thế nào?

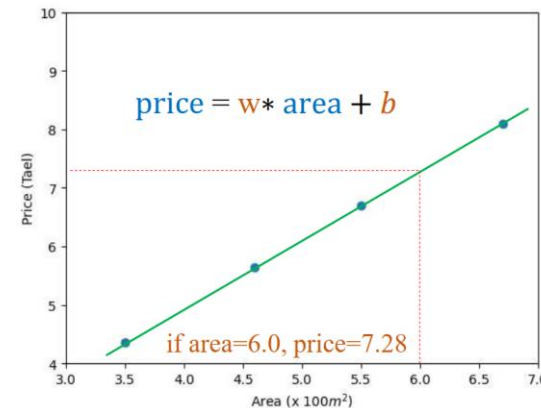


- Một model (Skip-Gram hoặc CBOW) sẽ có:
  - Input: One-hot encoder của một từ
  - Output: One-hot encoder của một từ
  - 2 layers:
    - Embedding Layer: layer được học để nhận biết những đặc điểm giữa các từ với nhau.
    - Prediction Layer: layer được học để dự đoán kết quả từ kết quả đặc điểm từ của embedding layer.
- -> Sau khi train xong model, ta sẽ lấy trọng số của embedding layer ra để làm embedding model.

# Mình đã học tới đâu rồi?

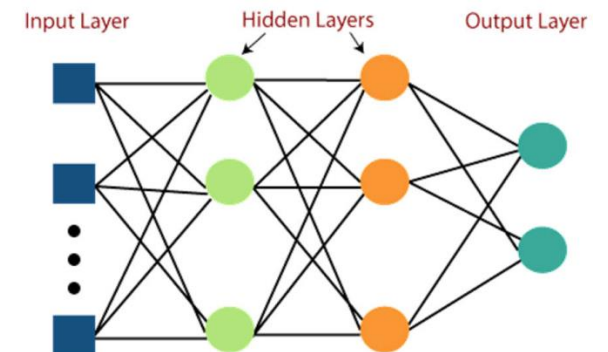
- 1. Linear Regression: là bài toán cơ bản của AI, để hiểu cách model AI dự đoán đưa ra kết quả như thế nào.

- Học về data training ( $x$ ,  $y_{gt}$ ,  $y_{pred}$ )
- Học về trọng số ( $W$ )
- Loss function (hàm mất mát)
- Quá trình cập nhật trọng số mỗi epoch



- 2. MLP (Multilayer-Perceptron): Là hệ thống kết hợp nhiều lớp lại để model giải quyết nhiều trường hợp hơn.

- Các loại layer (Input Layer, Hidden Layer, Output Layer)
- Trọng số trong từng layer
- Quá trình cập nhật trọng số qua từng layer trong mỗi epoch



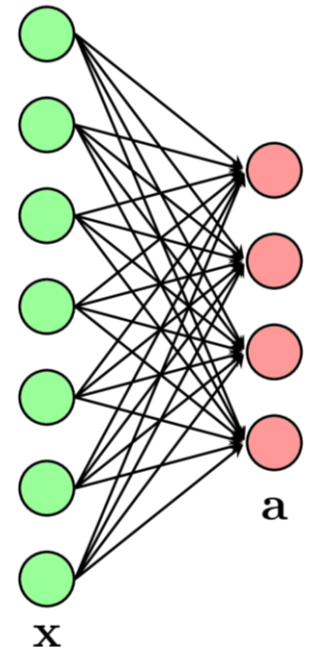
# Softmax Regression

## Bài toán phân loại.



# Softmax Regression – Bài toán phân loại

- Trong AI có những task mà yêu cầu đầu ra không phải là một giá trị (như Linear Regression) mà là nhiều giá trị.
- Task dự đoán của embedding (Skip Gram hoặc CBOW) cũng là dự đoán ra nhiều giá trị, vì một từ là một one hot encoder vector.

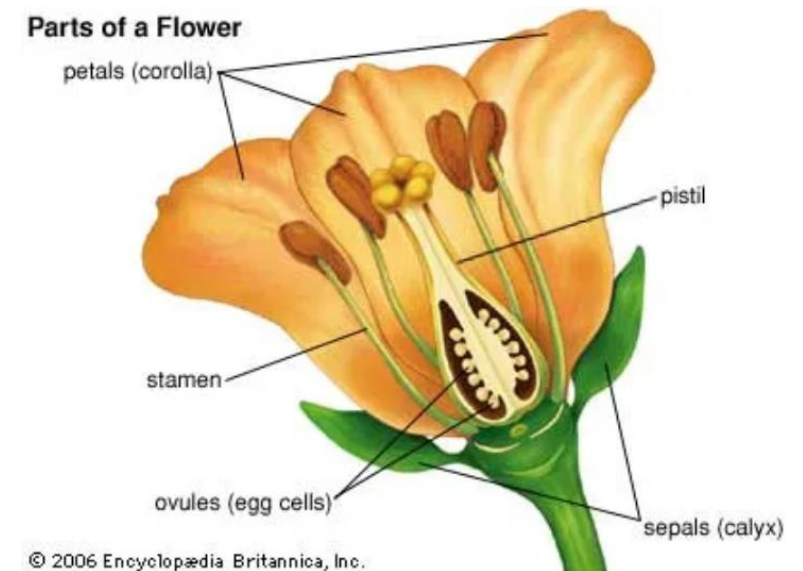


# Softmax Regression – bài toán phân loại hoa (IRIS Dataset)

- Iris Dataset là tập dataset dựa vào những đặc tính của hoa để dự đoán ra loại hoa. Dataset chứa:
  - 150 records
  - 4 features: sepal.length, sepal width, petal width, petal length
  - 3 class: Setosa, Virgnica, Versicolor

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	.2	Setosa
1	4.9	3	1.4	.2	Setosa
2	4.7	3.2	1.3	.2	Setosa
3	4.6	3.1	1.5	.2	Setosa
4	5	3.6	1.4	.2	Setosa
...	...	...	...	...	...
145	6.7	3	5.2	2.3	Virginica
146	6.3	2.5	5	1.9	Virginica
147	6.5	3	5.2	2	Virginica
148	6.2	3.4	5.4	2.3	Virginica
149	5.9	3	5.1	1.8	Virginica

150 rows × 5 columns



# Softmax Regression – bài toán phân loại hoa (IRIS Dataset)

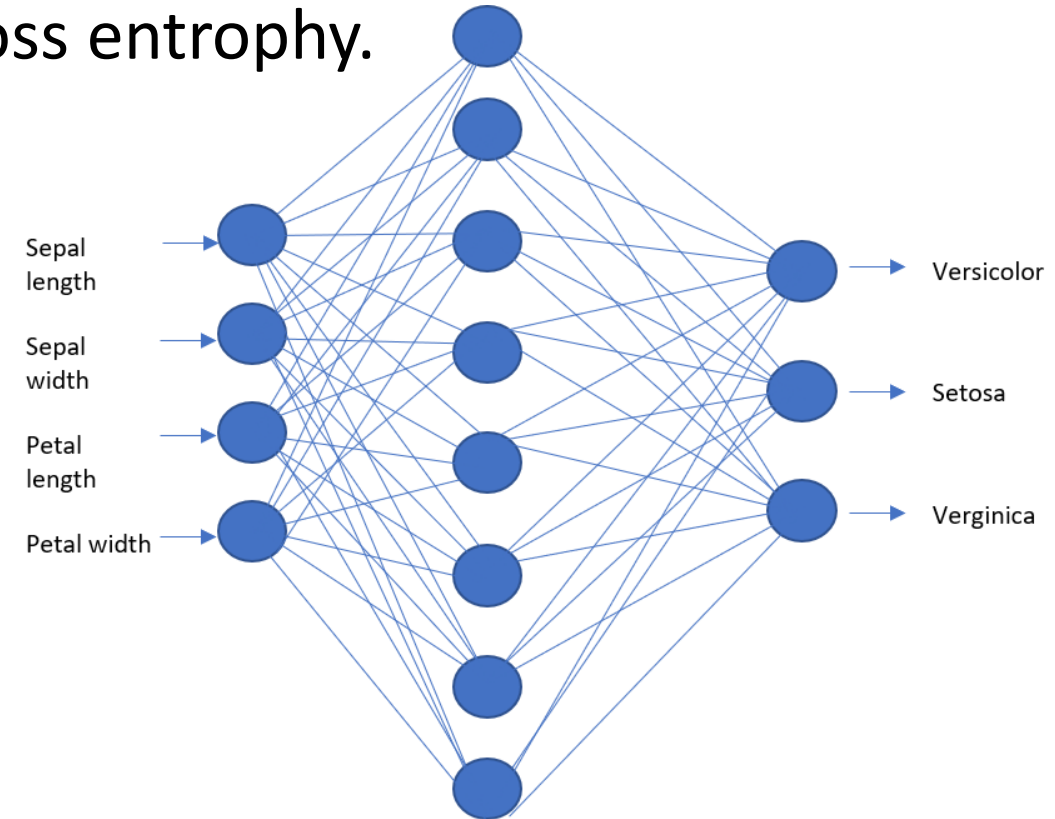
- Ta process lại dataset để thành data training.

sepal.length	sepal.width	petal.length	petal.width	variety_Setosa	variety_Versicolor	variety_Virginica
0	5.1	3.5	1.4	.2	1	0
1	4.9	3	1.4	.2	1	0
2	4.7	3.2	1.3	.2	1	0
3	4.6	3.1	1.5	.2	1	0
4	5	3.6	1.4	.2	1	0
...	...	...	...	...	...	...
145	6.7	3	5.2	2.3	0	0
146	6.3	2.5	5	1.9	0	0
147	6.5	3	5.2	2	0	0
148	6.2	3.4	5.4	2.3	0	0
149	5.9	3	5.1	1.8	0	0

150 rows x 7 columns

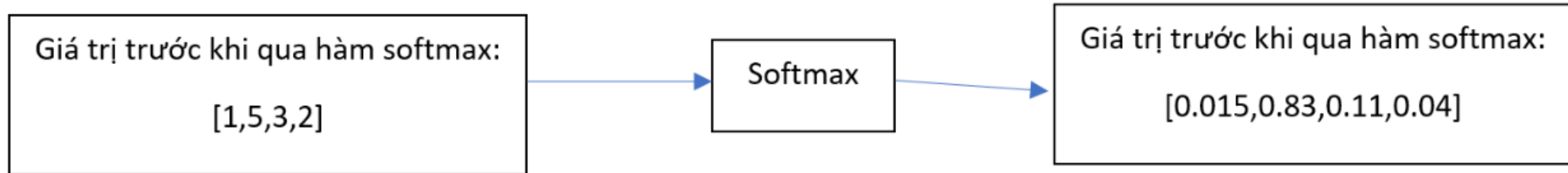
# Softmax function được áp dụng khi nào?

- Bài phân loại hoa, thì ta cần tạo ra model có 3 output (xác suất của từng loại hoa khi cho 4 feature vào).
- Ở output layer, để tổng các node bằng 1, ta dùng softmax function.
- Sau đó sẽ đem qua hàm loss function cross entropy.



# Softmax Function

- Hàm softmax biến vector k chiều có các giá trị thực bất kỳ thành vector k chiều có giá trị thực có tổng bằng 1.



- Công thức của Softmax function:


$$a_i = \frac{\exp(z_i)}{\sum_{j=1}^C \exp(z_j)}, \quad \forall i = 1, 2, \dots, C$$



# Hàm loss cross entropy.

- Là hàm tính mất mát khi:
  - Input và nhãn là vector
  - Vector input cộng lại bằng 1

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

where k is number of classes in the data

The lower the loss, the more accurate the model 

<b>DOG</b>  $\hat{y} = [0.4, 0.4, 0.2]$ $y = [0, 1, 0]$	$L(y, \hat{y}) = -0 \times \ln 0.4 - 1 \times \ln 0.4 - 0 \times \ln 0.2$ $= 0.92$
<b>HORSE</b>  $\hat{y} = [0.1, 0.2, 0.7]$ $y = [0, 0, 1]$	$L(y, \hat{y}) = -0 \times \ln 0.1 - 0 \times \ln 0.2 - 1 \times \ln 0.7$ $= 0.36$

365 DataScience

# Model Structure giải bài toán Iris

Bài:

- Cross Entropy Loss: hàm loss để tính loss value cho nhiều giá trị [vd:  $y = [0, 1, 0]$ ]. Thay vì một giá trị [vd:  $y = 5$ ] như Linear Regression
- Softmax Function: Quy đổi giá trị về dạng 0 1 0

