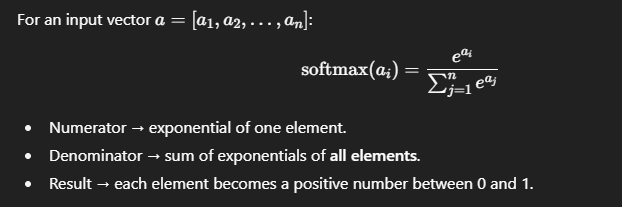


# The Softmax Activation Function

**1. Why Softmax is Different**

* Most activation functions (sigmoid, ReLU, tanh) take **one input number** and transform it independently.
* **Softmax is different**:
  + It takes the **entire vector** of values at once.
  + Each output depends on **all inputs together**.

2. The Formula



**3. Example Walkthrough**

Suppose we have a hidden layer with 3 units, producing:

1. Compute exponentials:
2. Denominator (sum):
3. Divide each exponential by 8:

Final softmax output:

**4. Why This Matters**

* Notice the properties:
  + All outputs are **between 0 and 1**.
  + They **sum to exactly 1**.
* That means the outputs form a **probability distribution**.

**5. Intuition in Practice**

* Let’s say we’re classifying an image among **3 classes**: Cat, Dog, Horse.
* The raw outputs (called **logits**) could be any numbers, like .
* After softmax → we get .
* This means:
  + 10% chance it’s a Cat
  + 20% chance it’s a Dog
  + 70% chance it’s a Horse

👉 That’s why softmax is almost always used in the **final layer of classification networks**.

**✅ Key Takeaway**

Softmax takes messy raw outputs (logits) and converts them into a **clean probability distribution** over possible classes, making the model’s predictions **interpretable and useful**.