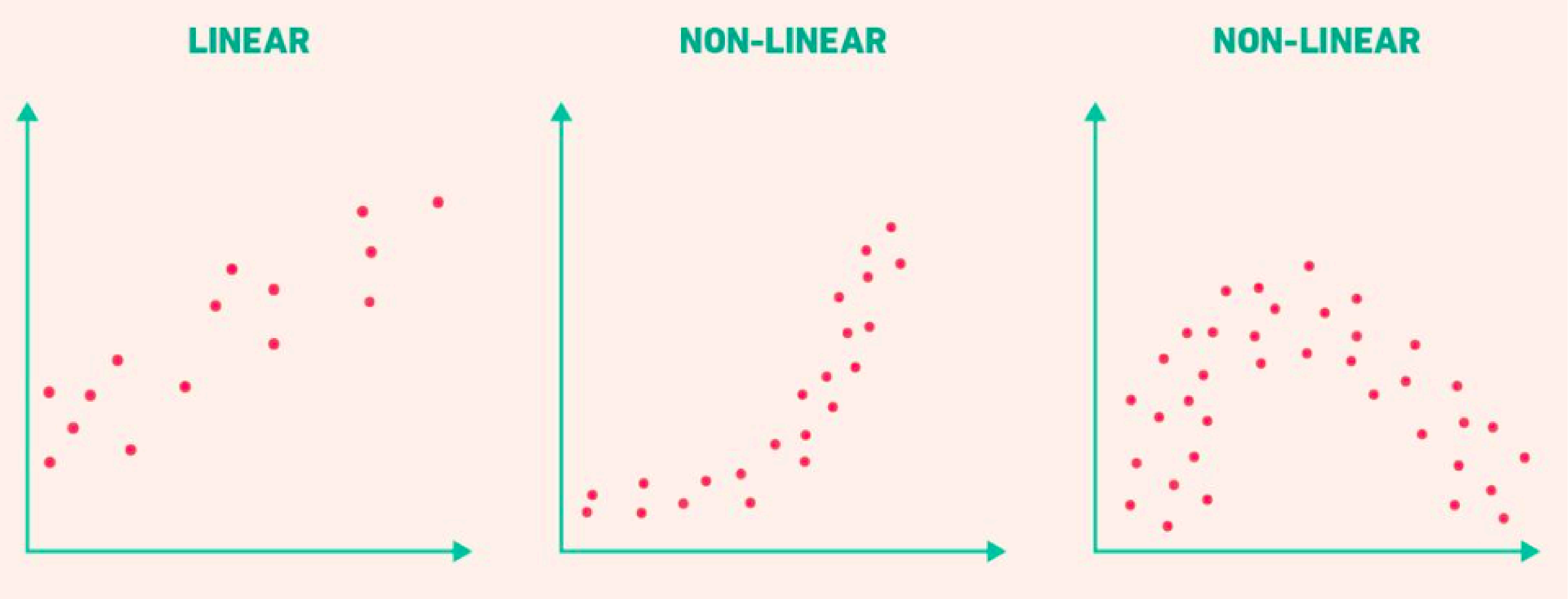
Module 1 - Session 3: Activation Functions

Session 3

The Problem: Linear Models Fail

Delivery company expanded: bike-only → city-wide with cars

Discovery: Delivery times stopped following a straight line



Why Linear Models Fail

Reality:

- First few miles: dense city traffic (slow)
- Further out: highways (faster)
- **The relationship curves**

Linear assumption: every mile needs the same time

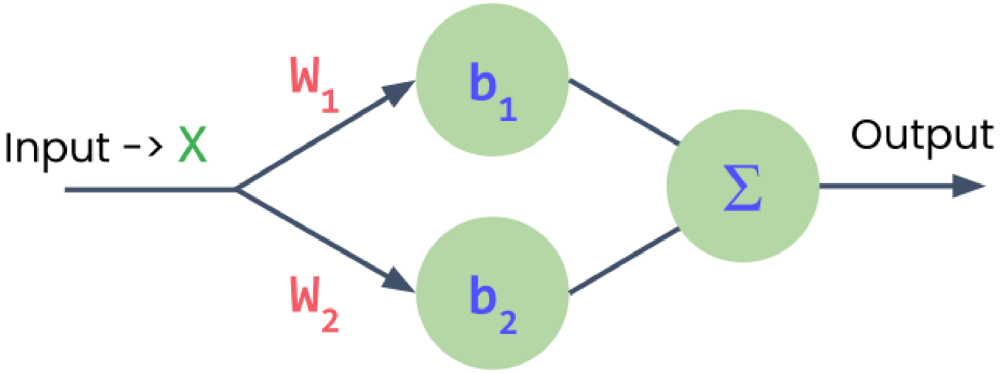
Adding More Neurons?

One neuron → one output

Two neurons → two outputs

Need: combine outputs into one prediction

Problem: Still just a linear equation!



$$\text{Output} = (W_1X + b_1) + (W_2X + b_2)$$

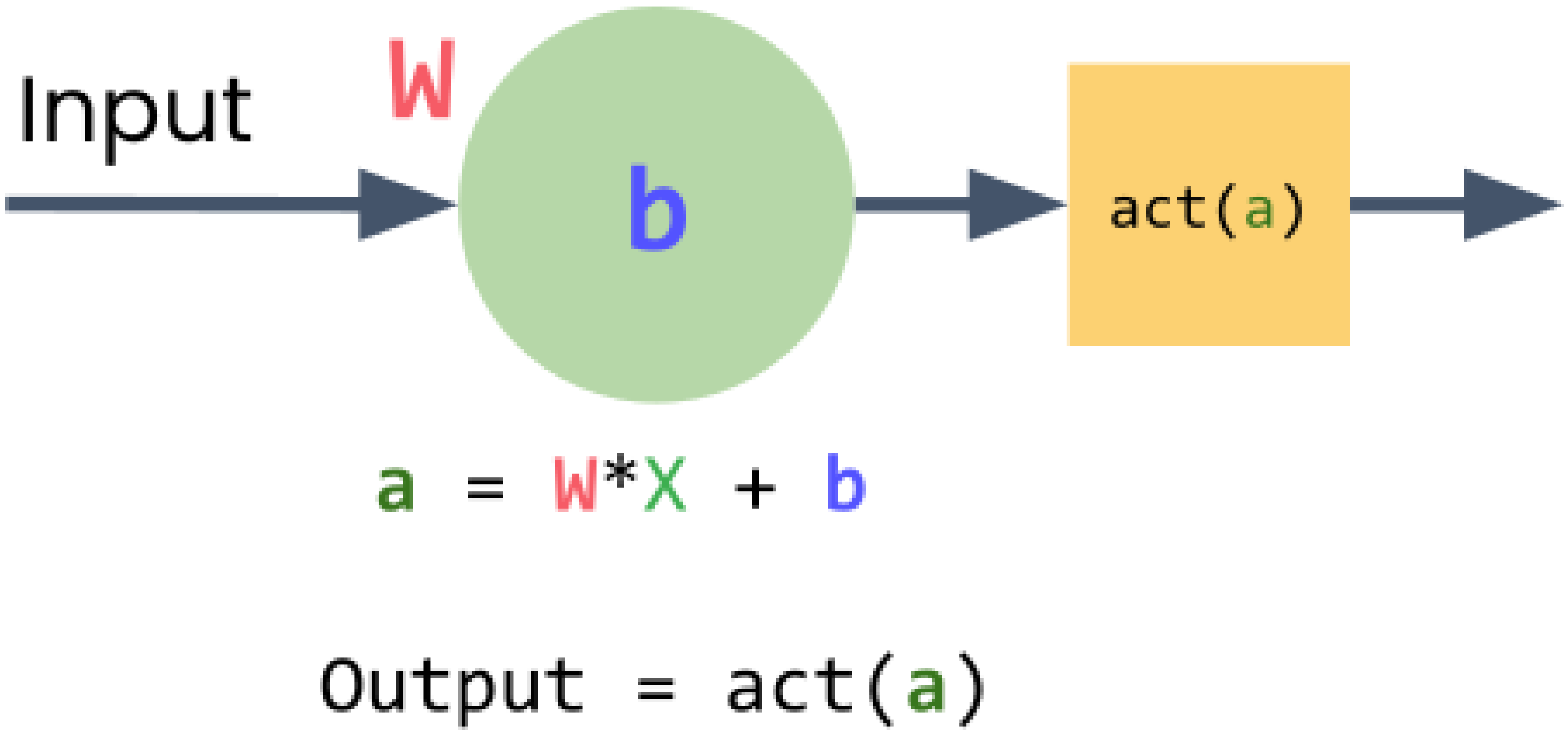
$$(W_1 + W_2)*X + (b_1 + b_2)$$

$$\boxed{W*X + b}$$ *Still a linear equation!*

The Solution: Activation Functions

Non-linear transformations applied to each neuron

Enables learning curves, not just straight lines

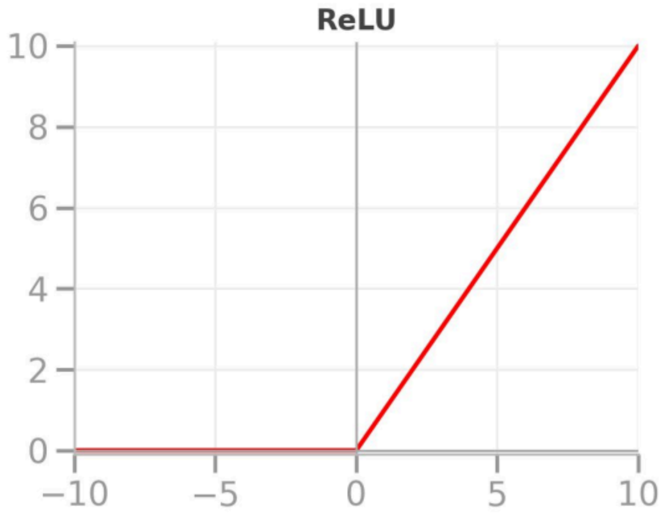


ReLU: Rectified Linear Unit

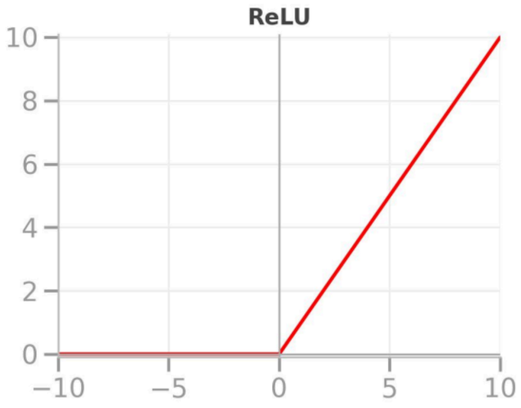
Simple rule:

- If input $< 0 \longrightarrow$ output $= 0$
- If input $\geq 0 \longrightarrow$ output $=$ input

Most popular activation function in deep learning



How ReLU Creates Non-Linearity



Without ReLU: straight line

With ReLU:

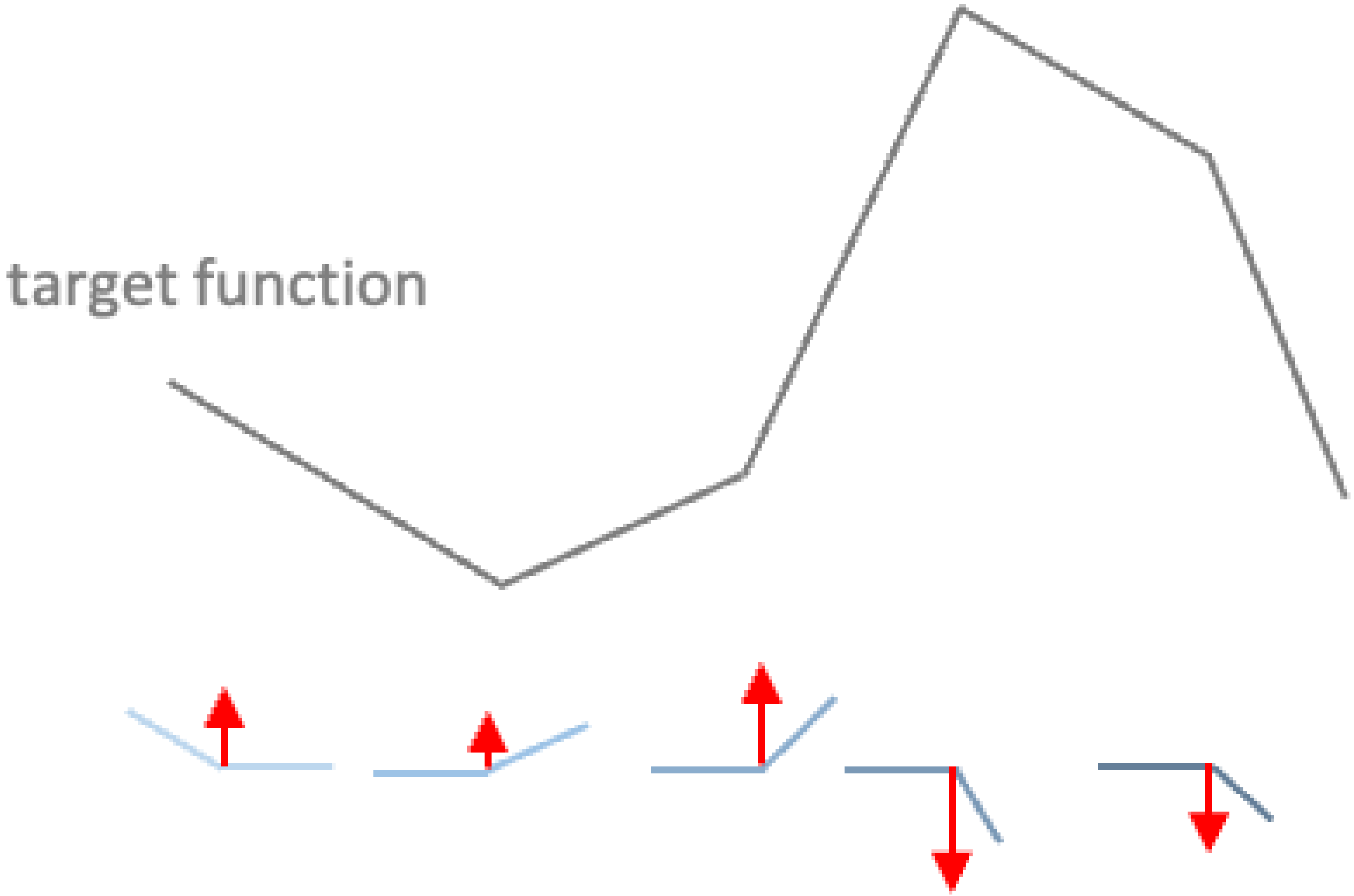
- When wx + b < 0 → flat line at 0
- When wx + b ≥ 0 → follows the line

Result: A bend, a corner where behavior changes

Multiple Neurons, Multiple Bends

One neuron → one bend

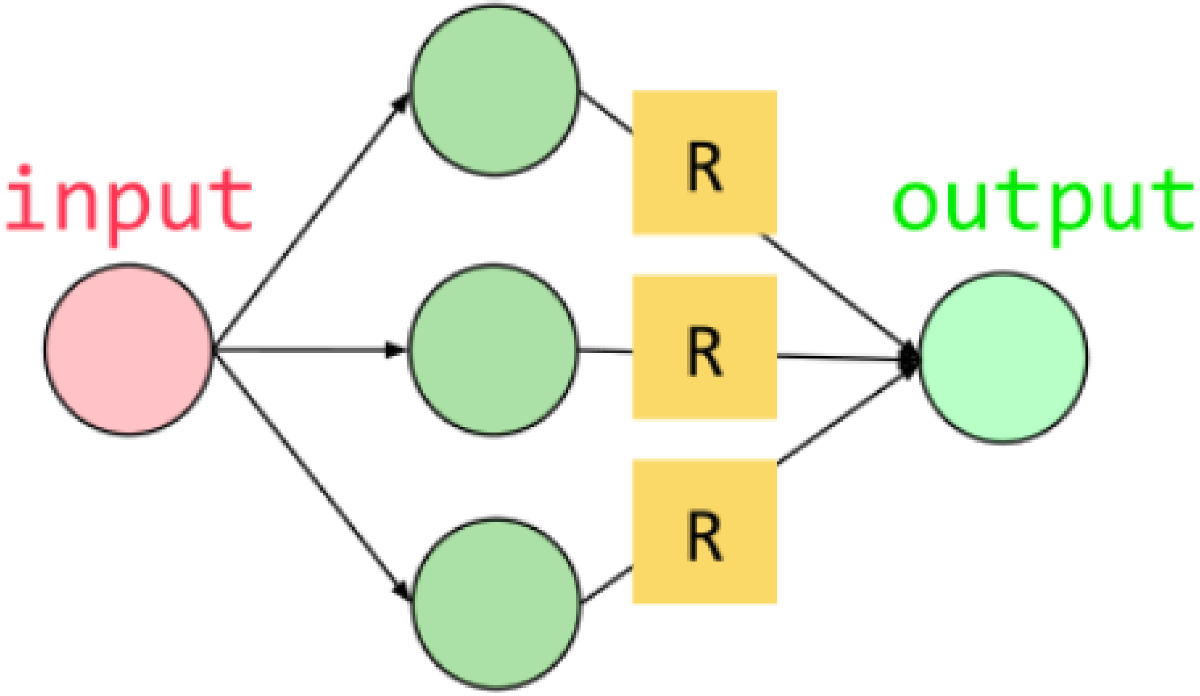
Multiple neurons → multiple bends → smooth curve



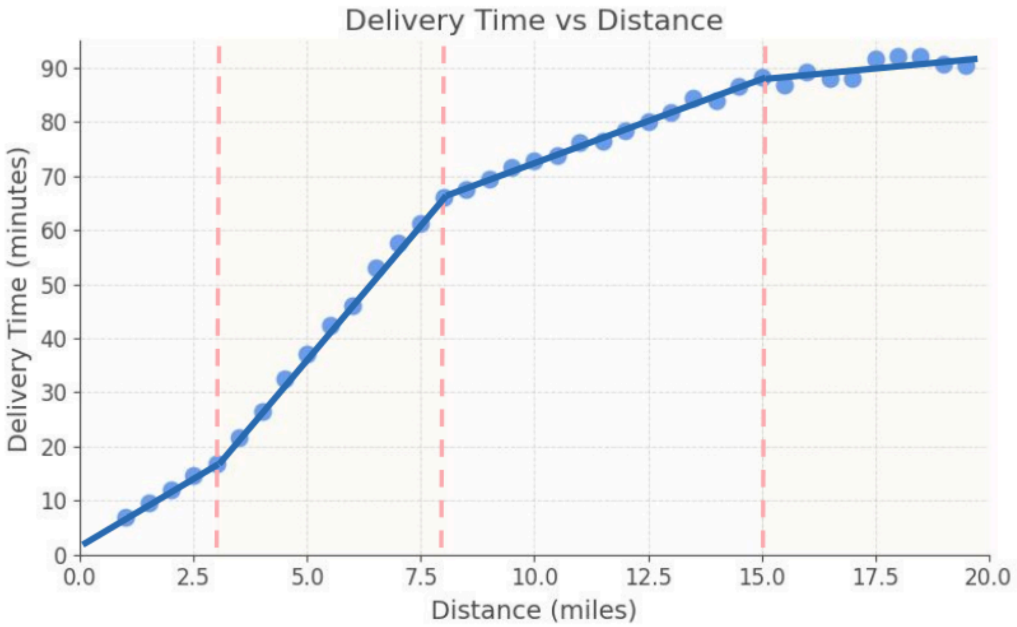
Building the Model in PyTorch

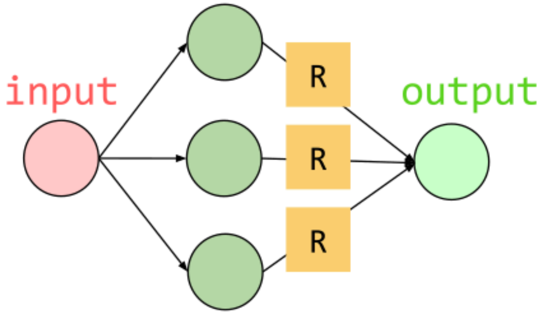
```
model = nn.Sequential(
    nn.Linear(1, 3),  # 1 input, 3 neurons
    nn.ReLU(),         # activation function
    nn.Linear(3, 1)   # 3 inputs, 1 output
)
```

Only two linear layers (ReLU is not a layer)



More Neurons = Smoother Curves



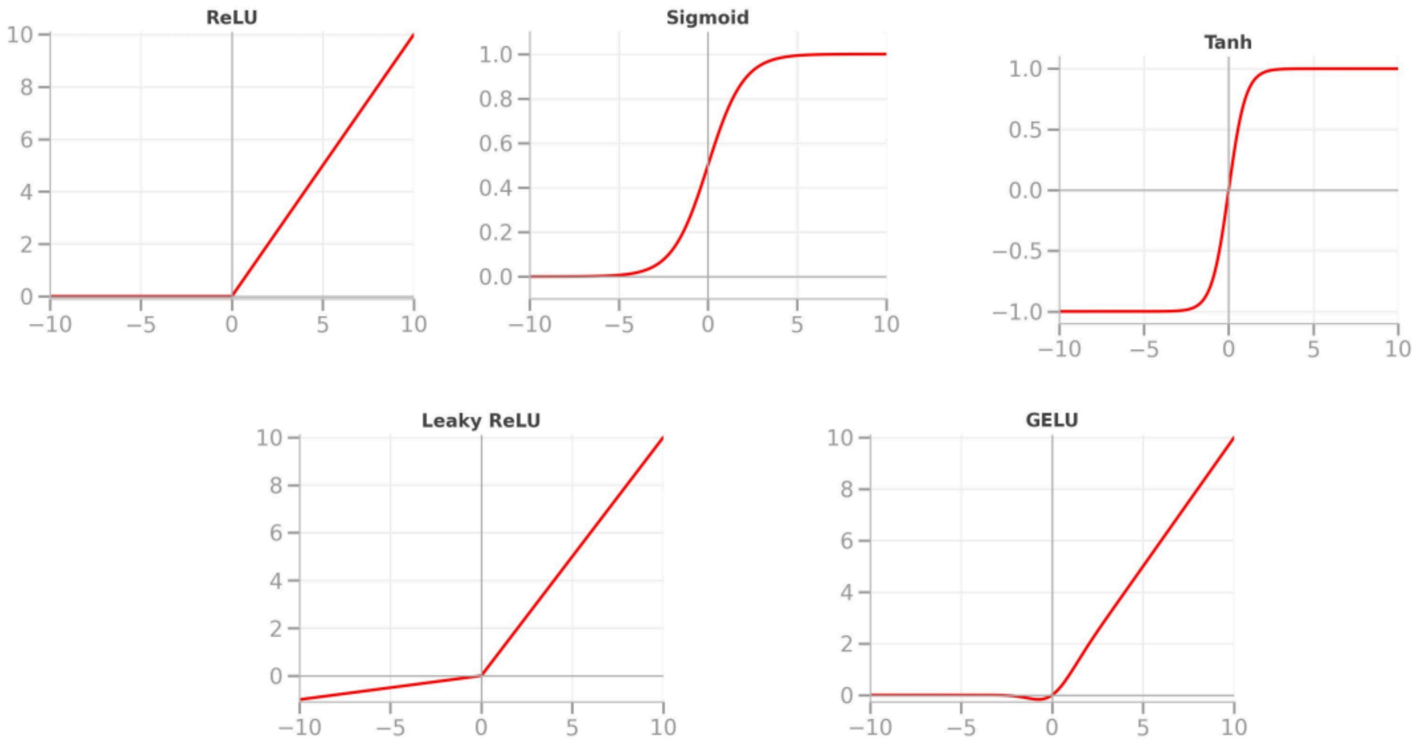


Each neuron learns: **Where to activate** and **how strongly to contribute**

Other Activation Functions

Sigmoid: outputs 0-1 (great for probabilities)

Tanh: outputs -1 to +1 (useful for many tasks)



ReLU: most common choice for hidden layers

Lab 2: Modeling Non-Linear Patterns with Activation Functions

> “In theory, there is no difference between theory and practice. In practice, there is.”

START WITH LAB 2

What’s Next?

In **Session 4: Working with Tensors** we learn:

- Understanding tensor shapes
- Data types in PyTorch
- Reshaping and indexing
- Element-wise operations and broadcasting