Activation Functions in Deep Learning :
Sigmoid vs Tanh vs ReLU vs Leaky ReLU

Author: *Jaya krishna katta*
student id:24100682

Github Repository: https://github.com/jayakrishnakatta/AFDL

Learning Outcomes:

By working through this tutorial, you will be able to:

- Understand why activation functions are essential in deep learning.
- Compare how Sigmoid, Tanh, ReLU, and Leaky ReLU behave in practice.
- Visualise differences in activation patterns and training behaviour.
- Choose appropriate activation functions for different model settings.
- Interpret accuracy curves, loss curves, and activation histograms.

Introduction:

- Activation functions play a crucial role in deep learning because they introduce non-linearity, enabling neural networks to model complex relationships. Even though they appear simple, the choice of activation function can significantly influence training speed, gradient flow, and final model performance.

- In this tutorial, we compare four widely used activations—Sigmoid, Tanh, ReLU, and Leaky ReLU—using the same neural network trained on the Fashion-MNIST dataset. Through experiments and visualisations, we show how each activation affects learning stability, convergence, and accuracy.

Technique and Focus

Technique: Activation functions in deep neural networks
Focus: Their practical impact on model performance
Although activation functions are introduced early in many courses, the deeper effects—like saturation, vanishing gradients, and dead neurons—are often not examined experimentally.

- This tutorial fills that gap using:
- Side-by-side model training comparisons
- Activation distribution plots
- Accuracy and loss graphs

Literature Review and Theory:

Research in deep learning has extensively analysed how activation functions behave Some key findings include:

Glorot & Bengio (2010): Sigmoid and Tanh activations tend to saturate, which leads to the vanishing gradient problem.

Nair & Hinton (2010): Introduced ReLU and showed that it speeds up training by avoiding saturation for positive inputs.

Maas et al. (2013): Proposed Leaky ReLU to reduce the "dead neuron" issue caused when ReLU outputs zero too often.

Course material (Week 6): provides mathematical background on activation functions and their effect on gradient-based optimisation.

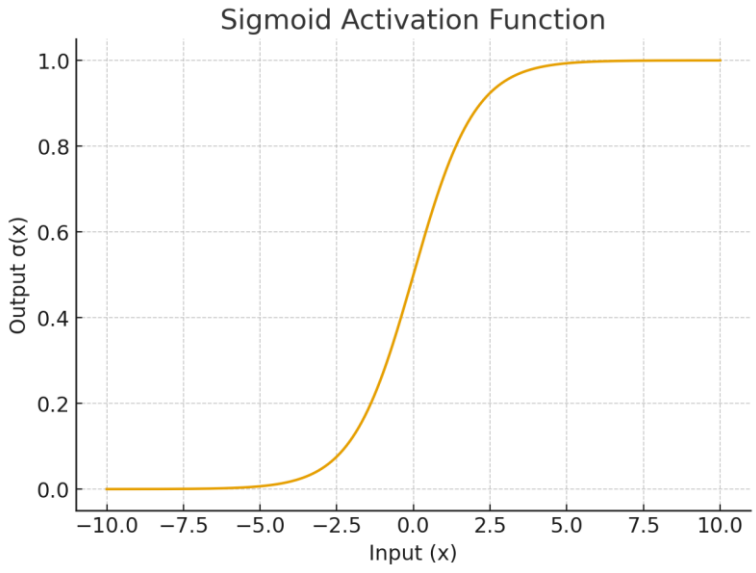
Overview of Key Activation Functions:

- **Sigmoid:** The Sigmoid function squashes any real number into the range (0, 1).

  Formula:

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

Sigmoid activation function graph look like this:



Characteristics

Range: (0, 1)

Strength: Good for output layers in binary classification

Weakness: Rarely used in hidden layers today

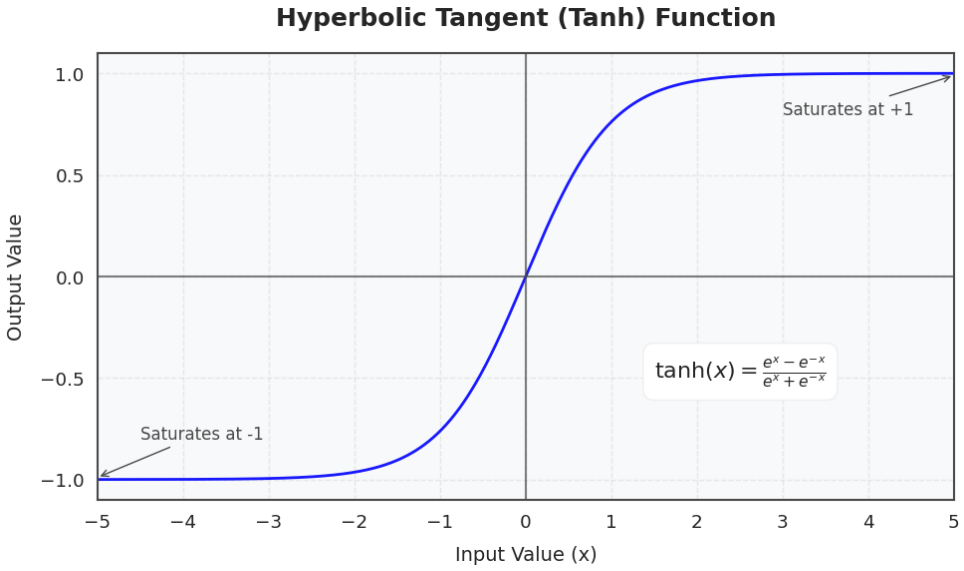
Use Case: Saturates quickly → vanishing gradient

Tanh : Tanh outputs values between –1 and 1, making it zero-centred and generally more balanced than Sigmoid.

Formula

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

- *Tanh activation function graph looks like this:*



Characteristics:

- ***Range****: (–1, 1)*
- ***Strength:*** *Zero-centred → smoother optimisation*
- ***Weakness****: Still suffers from saturation and vanishing gradients*

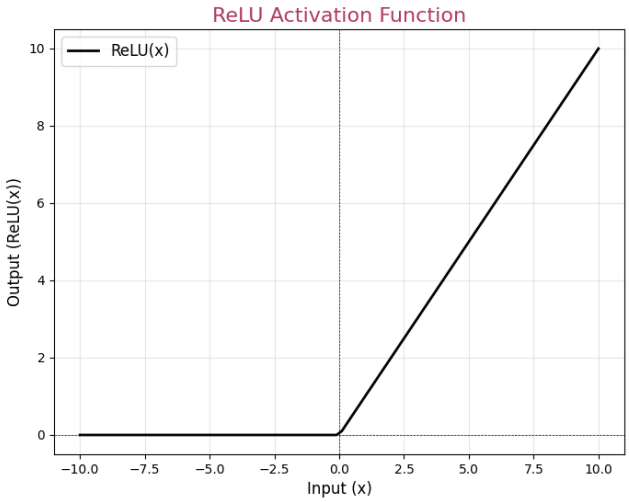
ReLU :

ReLU is one of the most commonly used activation functions in deep networks. It outputs the input itself if it's positive; otherwise, it outputs zero.

Formula:

$$f(x)=ma(x)$$

- *ReLU activation function graph looks like this:*



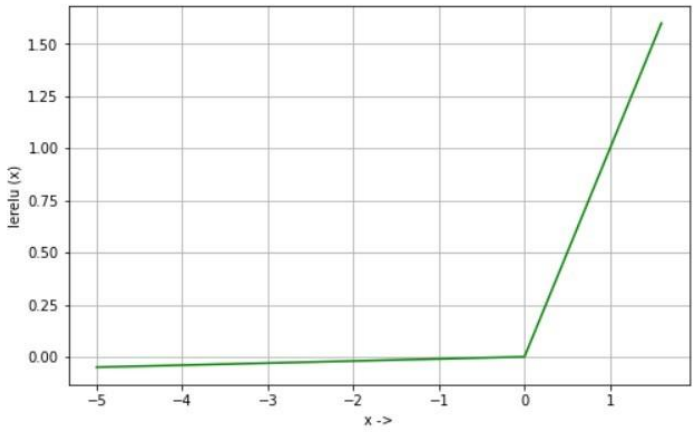
Characteristics:

- **Range:** [0, ∞)
- **Strengths:** Fast training, minimal saturation for positive inputs
- **Weakness:** Can produce dead neurons

Leaky ReLU: Leaky ReLU is a modified version of ReLU intended to fix the dead neuron problem. Negative inputs produce a small, non-zero output.

Formula:

$$\text{Leaky ReLU}(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.01 \cdot x, & \text{if } x \leq 0 \end{cases}$$



Characteristics:

- **Range:** Negative slope for x < 0 (e.g., 0.01x)
- **Strength:** Reduces dead neurons and maintains non-zero gradients
- **Use Case:** Often the most stable option in practice

Dataset:

Fashion-MNIST is chosen because:

- It is lightweight and easy to train on.
- It is more challenging and realistic than classic MNIST digits.
- It clearly highlights differences in activation-function behaviour.

class	category
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

validation Accuracy :

- ReLU and Leaky ReLU reach higher accuracy much faster.
- Sigmoid and Tanh train slowly because of vanishing gradients.
- Leaky ReLU shows the most stable improvements over time.

Validation Loss Comparison:

- sigmoid and Tanh reduce loss very slowly.
- ReLU and Leaky ReLU converge significantly faster.
- ReLU occasionally spikes due to dead neurons.
- Leaky ReLU has the smoothest, most stable loss curve.

Activation Distributions:

Activation	Distribution Behaviour	Implication
Sigmoid	clustered at 0 or 1	severe saturation → tiny gradients
Tanh	clustered near –1 or 1	saturates → vanishing gradients
ReLU	spike at 0 + positive tail	parse activations → efficient learning
Leaky ReLU	small negative tail	fewer dead neurons → more stable

Final Accuracy Table:

Activation	Final Validation Accuracy
Sigmoid	~80–83%
Tanh	~84–86%
ReLU	~88–90%
Leaky ReLU	~89–91% (best)

Summary of Findings:

- Sigmoid performs the worst due to strong saturation and vanishing gradients.
- Tanh is better but still suffers from the same fundamental issues.
- ReLU trains quickly and performs well but sometimes causes dead neurons.
- Leaky ReLU delivers the best stability and accuracy thanks to its non-zero negative slope.

Accessibility Considerations:

- Graphs use colourblind-friendly palettes.
- Axes and labels are clear and readable.
- Text uses high-contrast formatting for better visibility.
- Figures include descriptive captions and can be saved with alt-text for screen readers.

Final Reflection:

This tutorial highlights that activation functions are not minor details—they shape how neural networks learn, how quickly they converge, and how accurately they perform. By combining theoretical insights with practical experiments on the Fashion-MNIST dataset, we observe that:

> ReLU and Leaky ReLU outperform classical activations such as Sigmoid and Tanh. Leaky ReLU is generally the most consistent and stable option.

These findings provide a strong foundation for choosing activation functions in real-world deep learning projects.

References:

- **Glorot, X., & Bengio, Y. (2010).** Understanding the difficulty of training deep feedforward neural networks.
- **Nair, V., & Hinton, G. E. (2010).** Rectified linear units improve restricted Boltzmann machines.
- **Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013)**. Rectifier nonlinearities improve neural network acoustic models.
- **Clevert, D., Unterthiner, T., & Hochreiter, S. (2016). ELUs**: Fast and accurate deep network learning.
- **Course Slides** — Machine Learning and Neural Networks (Week 6).