Visualizing Activation Function

Chaitanya Katti

IIT, Bombay

August 3, 2023

Introduction - Neural Networks

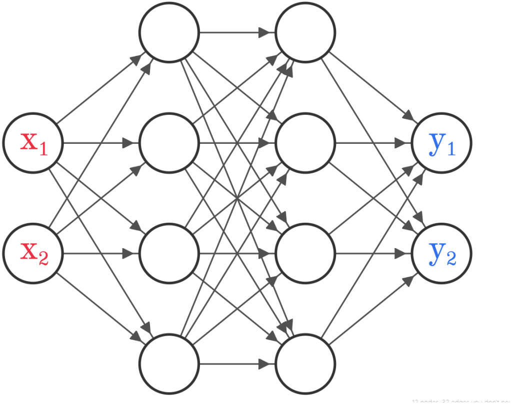


Figure: Architecture of a neural network

Introduction - Neural Networks

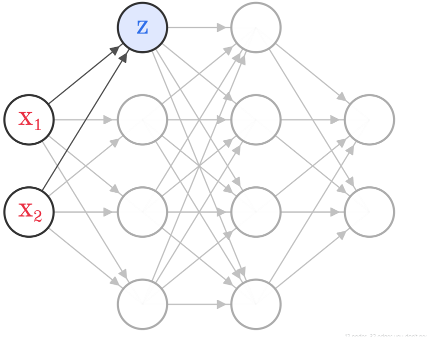


Figure: Activations of an artificial neuron

$$z = w_1 x_1 + w_2 x_2 + b$$

Introduction - Neural Networks

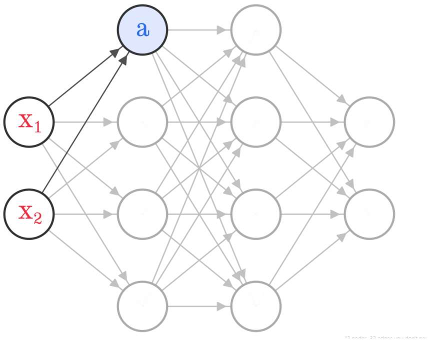


Figure: Activations of an artificial neuron

$$z = w_1 x_1 + w_2 x_2 + b \qquad a = f(z)$$

- where $f : \mathbb{R} \to \mathbb{R}$
- f is a smooth and differentiable function
- f must be non-linear

Introduction - Neural Networks

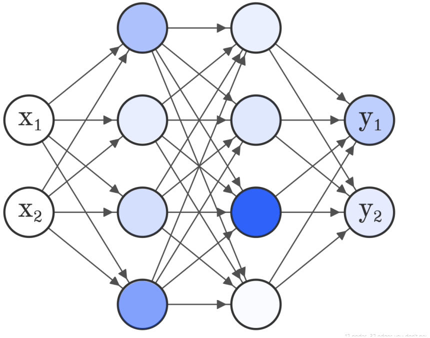


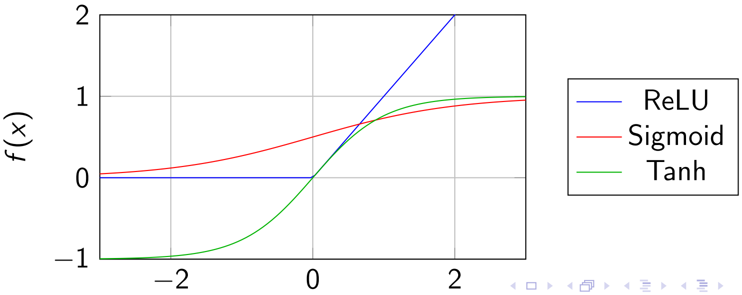
Figure: Activations of an artificial neuron

- Calculate all activations sequentially
- Final activation values of neurons are considered as the outputs

Activation Functions

Activation Function	Equation
ReLU	$f(x) = \max(0, x)$
Sigmoid	$f(x) = \dfrac{1}{1 + e^{-x}}$
Tanh	$f(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$

Table: Activation Functions and Their Equations



Approximating Non-Linear Functions With ReLU

We can approximate non-linear functions using ReLU activation function. Lets start with an example:

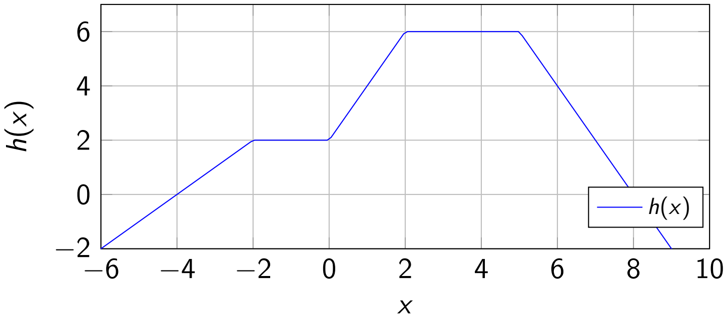
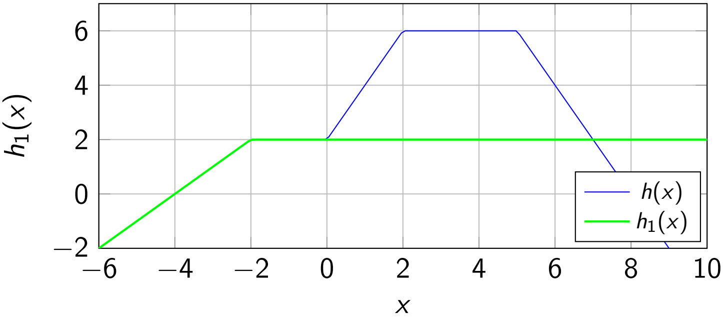


Figure: A non-linear function $h(x)$

Approximating Non-Linear Functions With ReLU

Try:

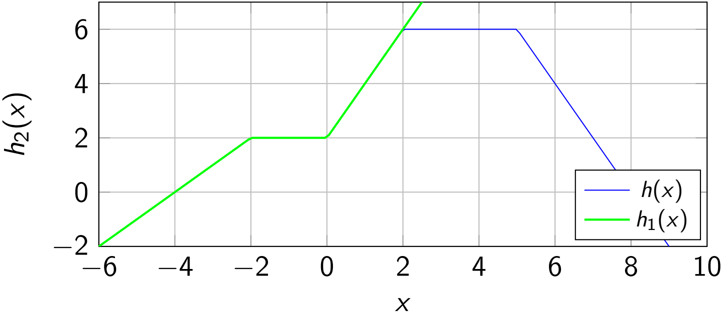
$$h_1(x) = 2 - f(-x - 4)$$



Approximating Non-Linear Functions With ReLU

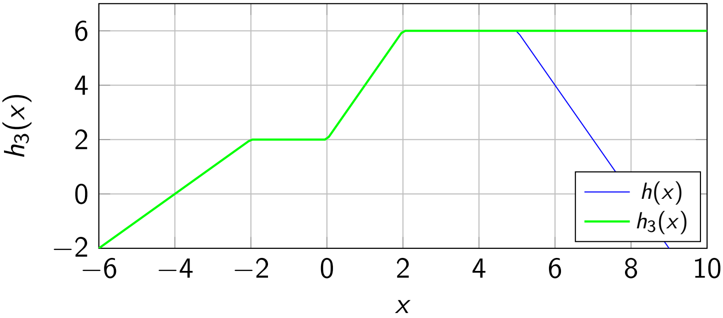
Try:

$$h_2(x) = 2 - f(-x-2) + 2f(x)$$



Try:

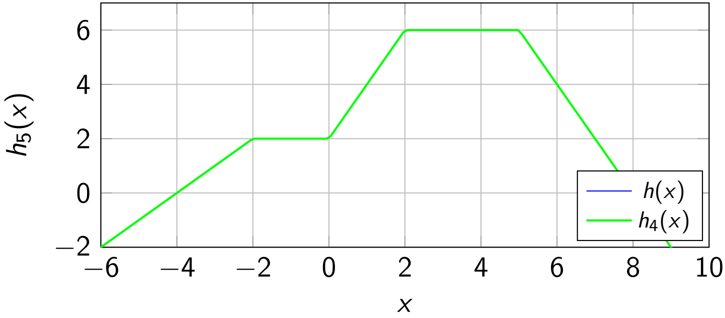
$$h_3(x) = 2 - f(-x-2) + 2f(x) - 2f(x-2)$$



Approximating Non-Linear Functions With ReLU

Finally Try:

$$h_4(x) = 2 - f(-x-2) + 2f(x) - 2f(x-2) - 2f(x-5)$$



Finally, we have approximated $h(x)$ using ReLU. Of course, this was a very simple example, but the same principle applies to more complex functions.

Approximating Functions With Tanh

- Tanh and sigmoid give smoother plots

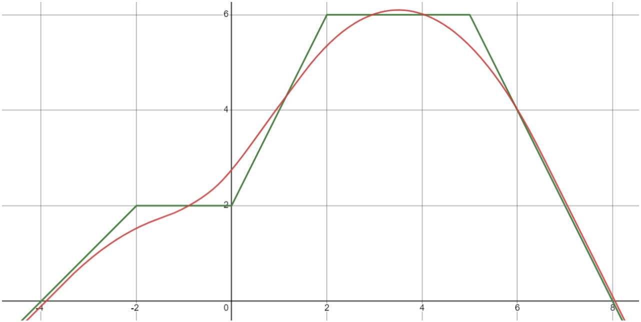


Figure: A rough fit using Tanh function

Multi-Class Classification

- **Multi-Class Classification** is the task of classifying an input into one of many classes.
- This is different from binary classification, where the input is classified into one of two classes.

Example: Given a cluster of points, classify each point into one of five classes.

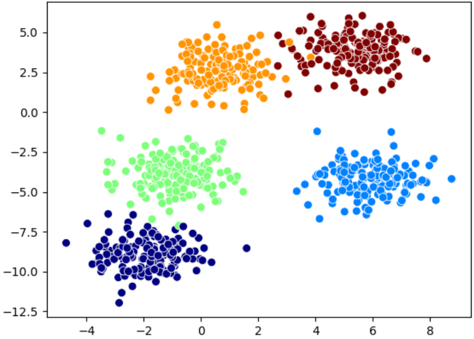
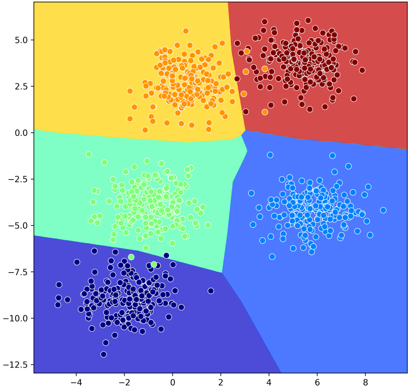


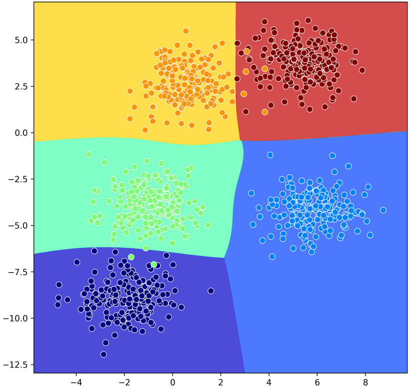
Figure: A cluster of points

Visualizing Desicion Boundaries

- Lets plot the prediction of each input point over a certain region.



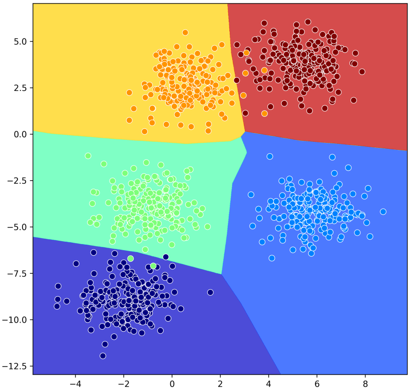
ReLU



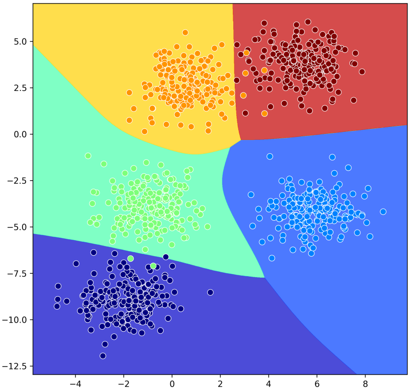
Sigmoid

Visualizing Desicion Boundaries

- Lets plot the prediction of each input point over a certain region.



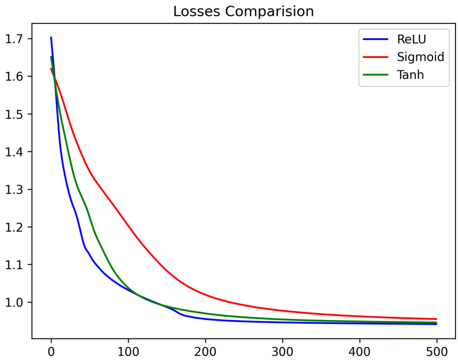
ReLU



Tanh

Loss Curves

- **Loss Curves** are plots of the loss over time.
- They are useful for visualizing how the loss changes over time.



Loss Curves

- Lets plot best, worst and mean loss curve over many training runs.

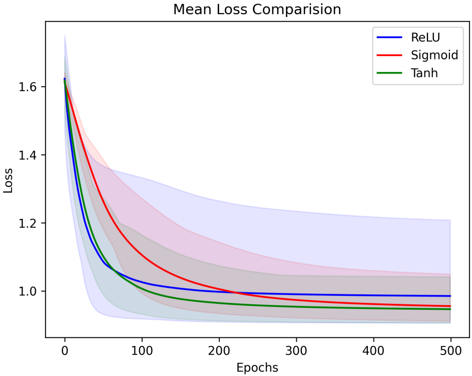


Figure: Mean loss curve

- ReLU converges the fastest.
- Sigmoid and Tanh have the bester worst case performance.

Conclusion

- ReLU (Rectified Linear Unit):
  - Simple and computationally efficient.
  - Solves the vanishing gradient problem.
  - Can suffer from the "dying ReLU" problem.
- Sigmoid:
  - Continuously differentiable.
  - Prone to vanishing gradient problem for deep networks.
  - Output is not zero-centered, leading to slower convergence.
- Tanh (Hyperbolic Tangent):
  - Output is zero-centered, aiding convergence.
  - Still suffers from the vanishing gradient problem.

Conclusion

Desicion boudaries are highly influenced by-

- Nature of classifier algorithm
- Choice of activation functions
- Number of total parameters

Activation functions also affect-

- Rate of convergence
- Final training loss
- Average accuracy