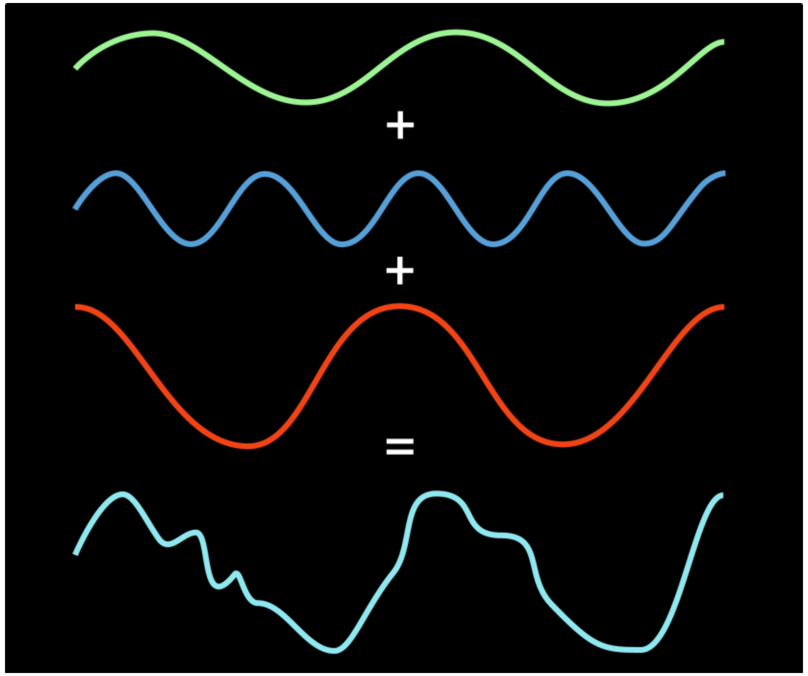
Deep Learning Course Math

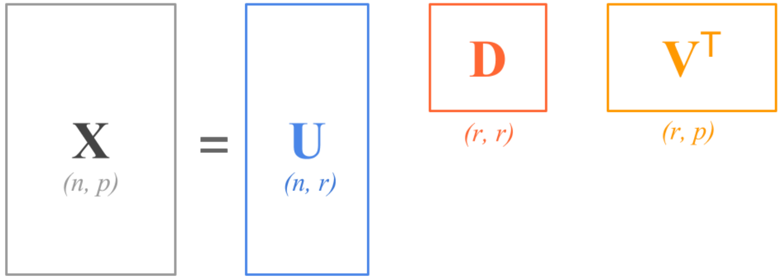
Farid Afzali, Ph.D., P.Eng.

Spectral Theories in Mathematics

- **Spectral theories** in mathematics share a common goal: breaking down complex entities into simpler components.
- This reductionist perspective emphasizes understanding intricate systems by comprehending their elementary parts.
- Analogous to visible light's spectrum, where diverse frequencies and waveforms collectively create light.
- Practical applications include the **Fourier transform** and **eigendecomposition** (singular value decomposition) that decompose complex signals and matrices into simpler constituents.

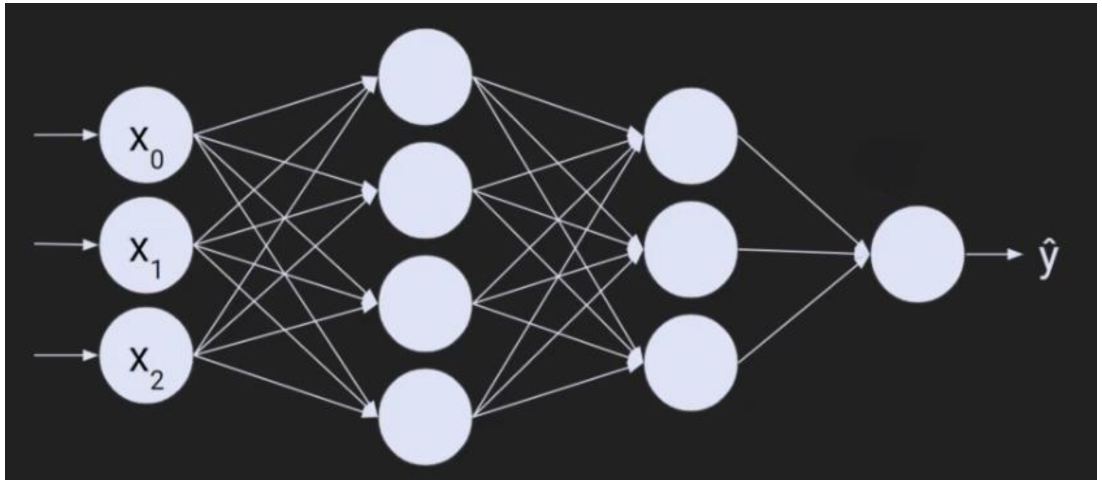
Spectral Theories in Mathematics

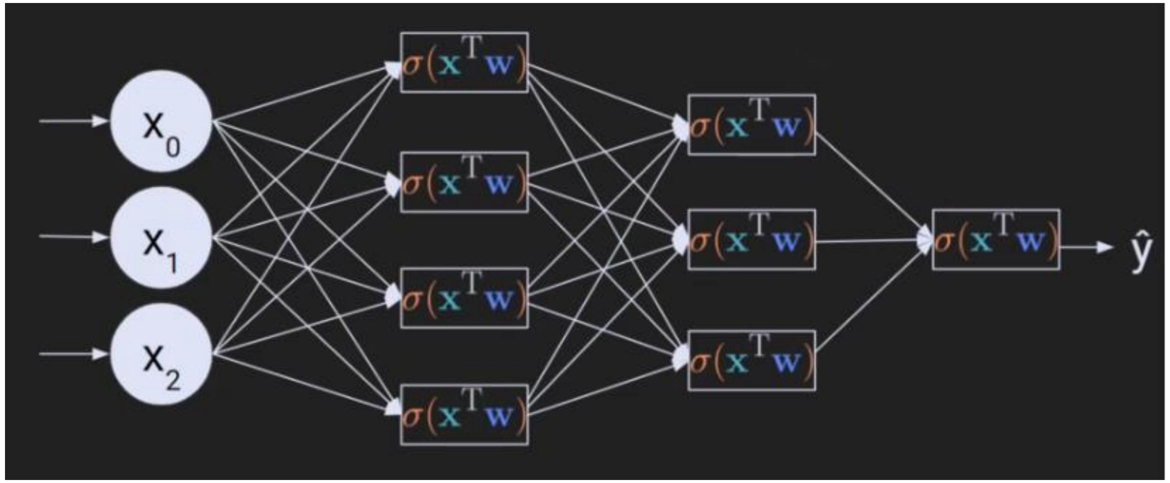




Deep Learning as Simple Components

- Despite its apparent intricacy, deep learning can be deconstructed into elementary mathematical operations.
- Neural network nodes execute fundamental computations, primarily involving the dot product and simple nonlinear transformations.
- The mathematics underlying deep learning comprises these basic, intuitive components.

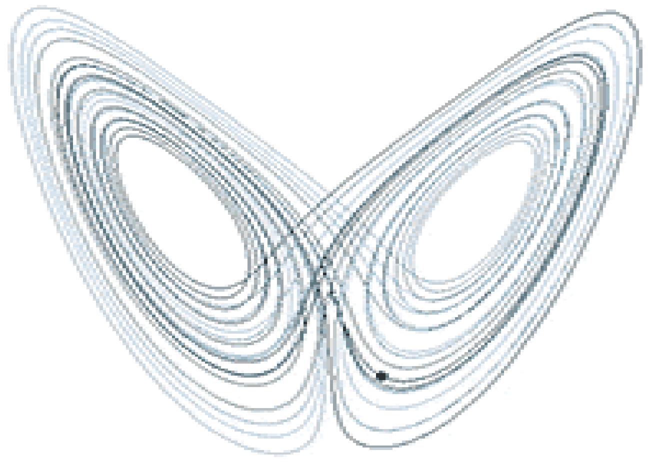




Complicated vs. Complex Systems

- It's crucial to differentiate between "complicated" and "complex" systems.
- Complicated systems involve numerous components, potentially linear and a few nonlinear parts, remaining reasonably understandable (e.g., a car, phone).
- Complex systems may lack many components but contain multiple nonlinearities, making them unintuitive and challenging to predict (e.g., Conway's Game of biology).

$$
\begin{aligned}
\frac{\mathrm{d}x}{\mathrm{d}t} &= \sigma(y - x), \\
\frac{\mathrm{d}y}{\mathrm{d}t} &= x(\rho - z) - y, \\
\frac{\mathrm{d}z}{\mathrm{d}t} &= xy - \beta z.
\end{aligned}
$$

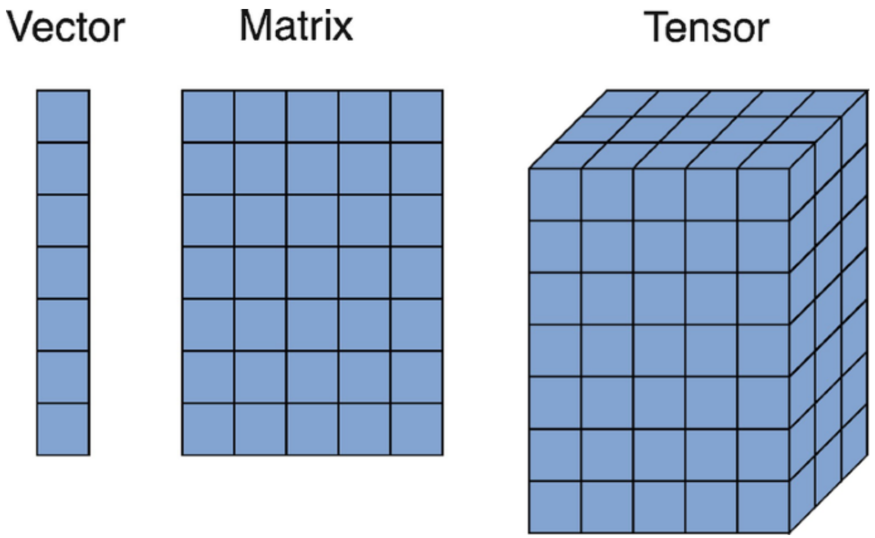


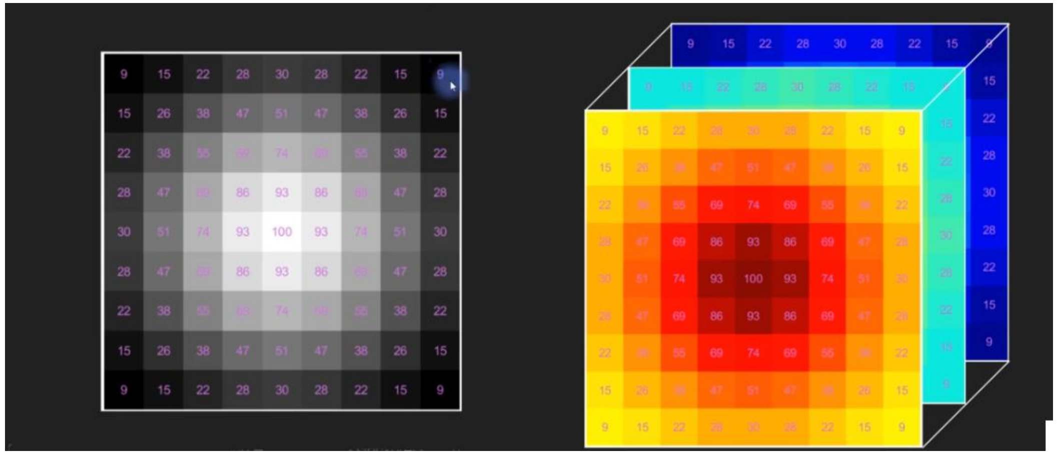
Deep Learning: Simple, Complicated, and Complex

- Deep learning combines **simplicity**, **complexity**, and **complication**.
- Its core mathematics is straightforward, but complexity emerges from interconnected components and nonlinearities.
- Deep learning's behavior can be unintuitive, prompting debates about its understandability.
- This blend of simplicity, complication, and complexity shapes the landscape of deep learning.

Important Terminology in Linear Algebra and Computer Science

- **Key Terms in Linear Algebra**
  - **Scalar**: A single, standalone number, often used in linear algebra to scale or stretch vectors.
  - **Vector**: A collection of numbers, arranged either in a column or a row. It represents one physical dimension.
  - **Matrix**: A two-dimensional spreadsheet of numbers, resembling an Excel spreadsheet.
  - **Tensor**: A higher-dimensional object, like a cube of numbers. Deep learning often works with tensors, which can have more than three dimensions.





Data Types

- Computer Science Data Type: Refers to the format of data storage in computing. Examples include **floating-point numbers, Booleans, and strings.**

- Statistics Data Type: Describes the category of data, influencing which statistical procedures can be applied. Examples include **categorical, numerical, and ratio data**.

Computer Science Data Type

- **Definition**: In computer science, a data type refers to the format of data storage and the operations that can be performed on that data.
- **Examples**: Computer science data types include:
  - **Floating-Point Numbers**: Used to represent real numbers with decimal points, allowing for both integer and fractional parts.
  - **Booleans**: Represent two values, typically "true" or "false," used for logical operations.
  - **Strings:** Used to store sequences of characters or text.
- **Purpose**: Computer science data types determine how data is stored in computer memory and how operations are performed on that data. They help ensure that the computer processes the data correctly and efficiently.

Statistics Data Type

- **Definition**: In statistics, data type categorizes data based on its nature, which influences the statistical procedures and analyses that can be applied.
- **Examples**: Statistics data types include:
  - **Categorical Data**: Represents distinct categories or labels, such as colors or types of fruits.
  - **Numerical Data**: Consists of numerical values and can further be divided into:
    - **Interval Data**: Represents numerical values where the intervals between values have meaning (e.g., temperature in Celsius).
    - **Ratio Data**: Similar to interval data but includes a true zero point (e.g., height in centimeters, weight in kilograms).
- **Purpose**: Statistics data types help statisticians choose appropriate methods for analyzing and summarizing data. Different types of data require different statistical techniques and tools. For example, you wouldn't calculate the mean (average) of categories like colors; you'd use it for numerical data like heights.

Handling Data Types in Deep Learning

- In this course, we primarily use the **computer science interpretation of data type**, focusing on how data is stored and processed in computing.
- Different operations may require specific data types, so we may need to convert between them.

Transpose in Linear Algebra

- **Transpose Operation**: In linear algebra, the transpose operation converts rows into columns and columns into rows within matrices or vectors.
- **Visual Interpretation**: Transposing does not change the values within a vector or matrix, only their orientation.
- **Double Transpose**: Applying the transpose operation twice (double transpose) reverses it, returning to the original vector or matrix.

Transpose Operation in Python (Numpy)

- **Numpy:** A widely used Python library for numerical operations.

Practice In Python

Transpose Operation in Python (PyTorch)

- **PyTorch:** A popular Python library for deep learning and tensor computation.

Practice In Python

Dot Product

- **Notations for Dot Product**
  - Dot Product Notations: The dot product is represented in various ways, including as "A.B," "A ,B" within angle brackets, or "A^T. B" (most common).

- **Dot Product in Mathematics**

Feature selections

  - **Dot Product Definition**: Mathematically, the dot product between two vectors A and B involves multiplying corresponding elements and then summing them:

$$A \cdot B = \sum_{i=1}^{n} A_i \cdot B_i$$

  - Dot Product for Vectors: Dot product is defined for vectors that have the same number of elements (dimensionality).

Dot Product Example and Constraints

- Example Calculation: Compute the dot product between vector A and vector B:

$$\mathbf{A} = [2.0, 3.0, 4.0]$$

$$\mathbf{B} = [3.0, 4.0, 6.0]$$

- Result: The dot product of A and B is

$$\mathbf{A} \cdot \mathbf{B} = 2.0 \cdot 3.0 + 3.0 \cdot 4.0 + 4.0 \cdot 6.0 = 42.0$$

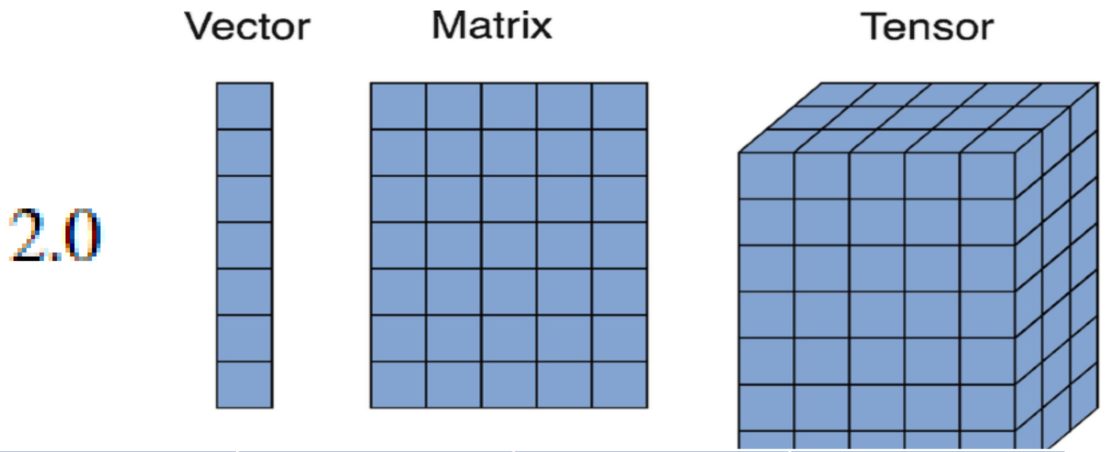
- **Dot Product Constraints**: The dot product is only defined between vectors of the same dimensionality (having the same number of elements).
- **Example**: Attempting to compute the dot product between a 5-element vector and a 3-element vector will result in an error.

Interpretation of Dot Product

- **Interpretation**: The dot product is a single number that reflects the similarity or commonality between two mathematical objects, such as vectors or matrices.
- **Connection to Statistics**: The dot product is related to statistics concepts like correlation and covariance, which are essentially normalized versions of the dot product.
- **Applications**: The dot product serves as the computational backbone for various operations, including convolution, matrix multiplication, and style transfer.

Numpy and PyTorch Terminology

- In Numpy, vectors are called arrays, while matrices and tensors are referred to as n-dimensional arrays.
- In PyTorch, all data structures, including scalars, are called tensors, and they have associated data types.



Math	Scaler	Vector	Matrix	Tensor
Numpy	Array	Array	ND Array	ND Array
PyTorch	Tensor	Tensor	Tensor	Tensor

Numpy and PyTorch

- **NumPy**:
  - **Vectors**: In NumPy, vectors are typically referred to as arrays. These arrays can be one-dimensional and represent sequences of data.
  - **Matrices and Tensors**: In NumPy, matrices and tensors with more than two dimensions are collectively referred to as n-dimensional arrays or n-d arrays. They are the fundamental data structures used for handling multi-dimensional data in numerical computations.
- **PyTorch**:
  - **Scalars, Vectors, Matrices, and Tensors**: In PyTorch, all data structures are called tensors. This includes scalars (single values), vectors (one-dimensional arrays), matrices (two-dimensional arrays), and tensors with more than two dimensions. The term "tensor" is used uniformly to represent these different types of data structures.
  - **Data Types:** PyTorch tensors have associated data types that define the type of data they can hold, such as float32, int64, etc. Specifying data types is important for ensuring proper memory allocation and precision in deep learning computations.

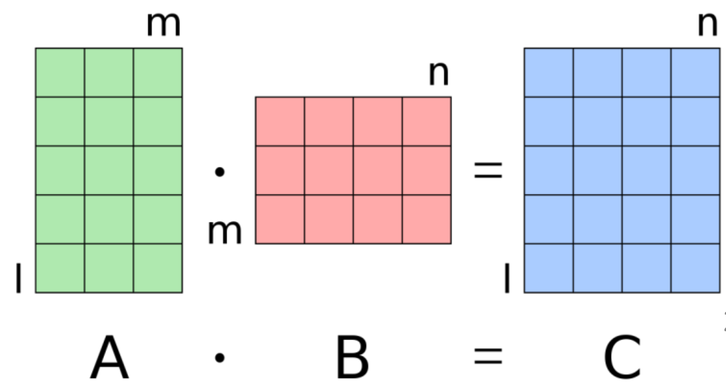
Matrix Multiplication

- ## Understanding Matrix Sizes

- Before exploring matrix multiplication further, it's crucial to understand how matrices are described in terms of their sizes, typically denoted as M rows by N columns.

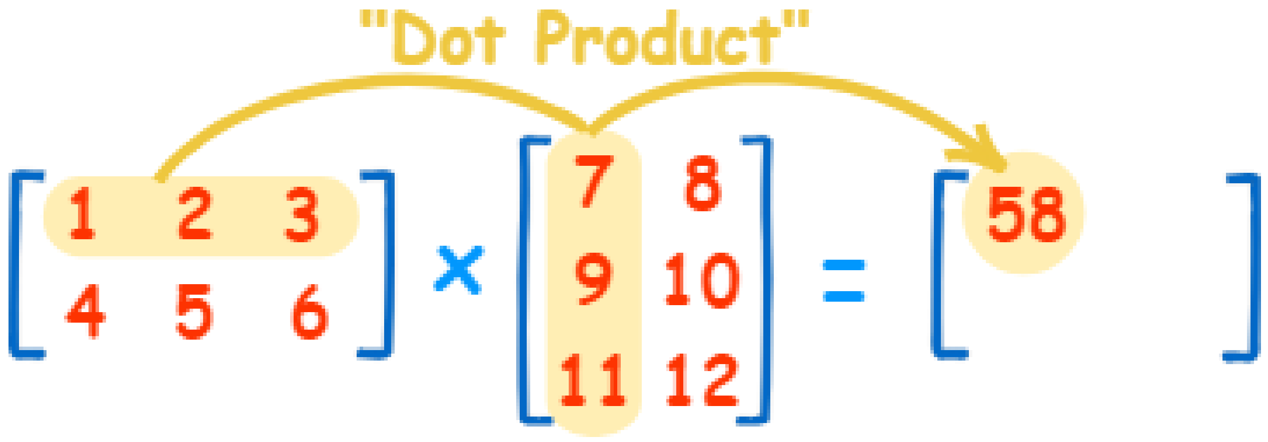
- ## Rules for Matrix Multiplication Validity

- Matrix multiplication is not always valid, and understanding when it is possible is crucial. To determine if matrix multiplication is valid, you must ensure that the inner dimensions match (number of columns in the left matrix equals the number of rows in the right matrix). The resulting matrix's size is determined by the outer dimensions.



Matrix Multiplication Example

- Given matrices A and B, an example demonstrates how to compute their product. Each element of the product matrix is obtained by taking the **dot product** of the corresponding row in the left matrix and the corresponding column in the right matrix.

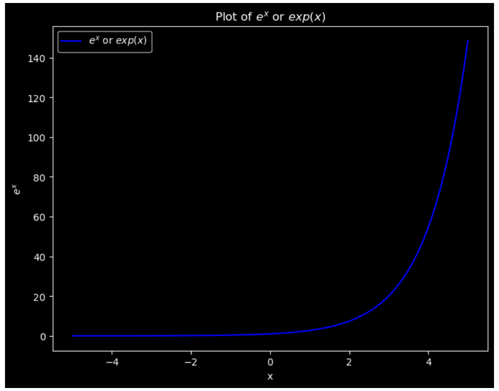


Softmax Function

- Softmax is a mathematical function often used in deep learning for various tasks, particularly in the output layer of neural networks, such as in **multi-class classification** problems.

$$\text{softmax}(z)_j = \frac{e_j^z}{\sum_{j=1}^{K} e_j^z}$$

- It takes a vector of real numbers as input and transforms it into a probability distribution over multiple classes.



- The softmax function is typically applied to the raw output scores or logits produced by the last layer of a neural network.
  - ***softmax(z)_i*** *represents the i-th element of the output probability distribution.*
  - ***z_i*** *is the i-th element of the input vector.*

- The softmax function exponentiates each element of the input vector and then normalizes them by dividing by the sum of all exponentiated values. This normalization ensures that the output values are between 0 and 1 and that they sum up to 1, making them suitable as probabilities.

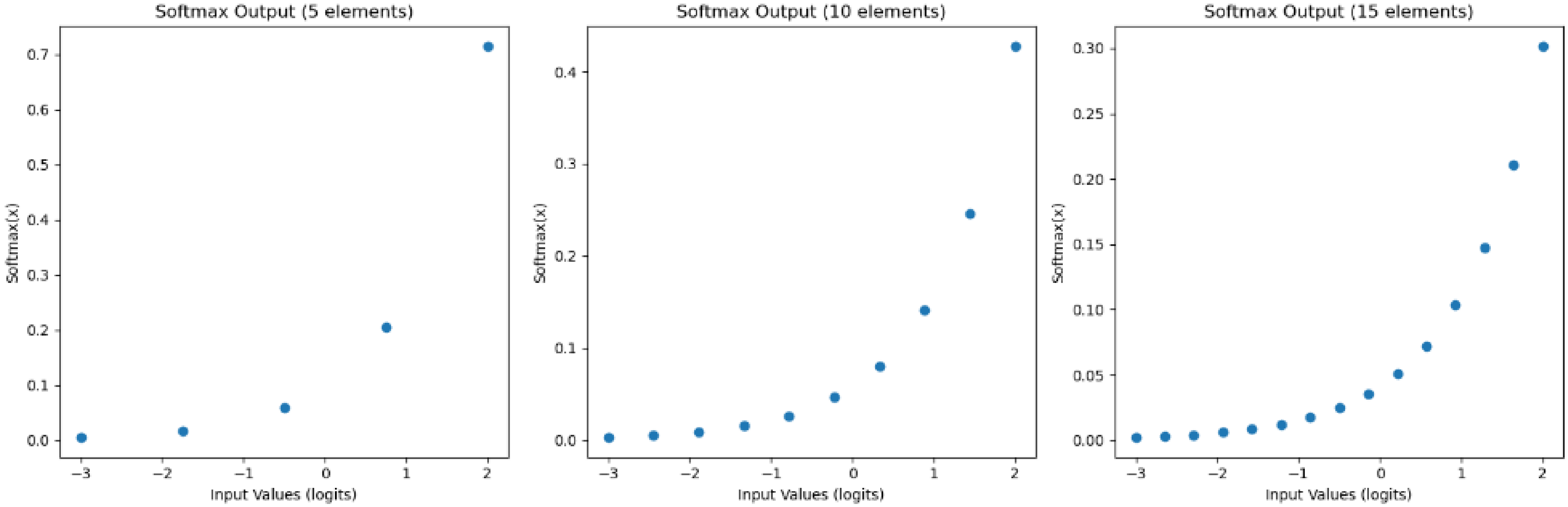
Properties of Softmax

1. The output of the Softmax function is a probability distribution, as the values are all between 0 and 1, and they sum up to 1.
2. It can be applied to any real-valued vector, making it suitable for **multiclass classification** problems.
3. The Softmax function has a **non-linear** nature; **it amplifies large input values and suppresses small ones**, making it sensitive to differences in input values.

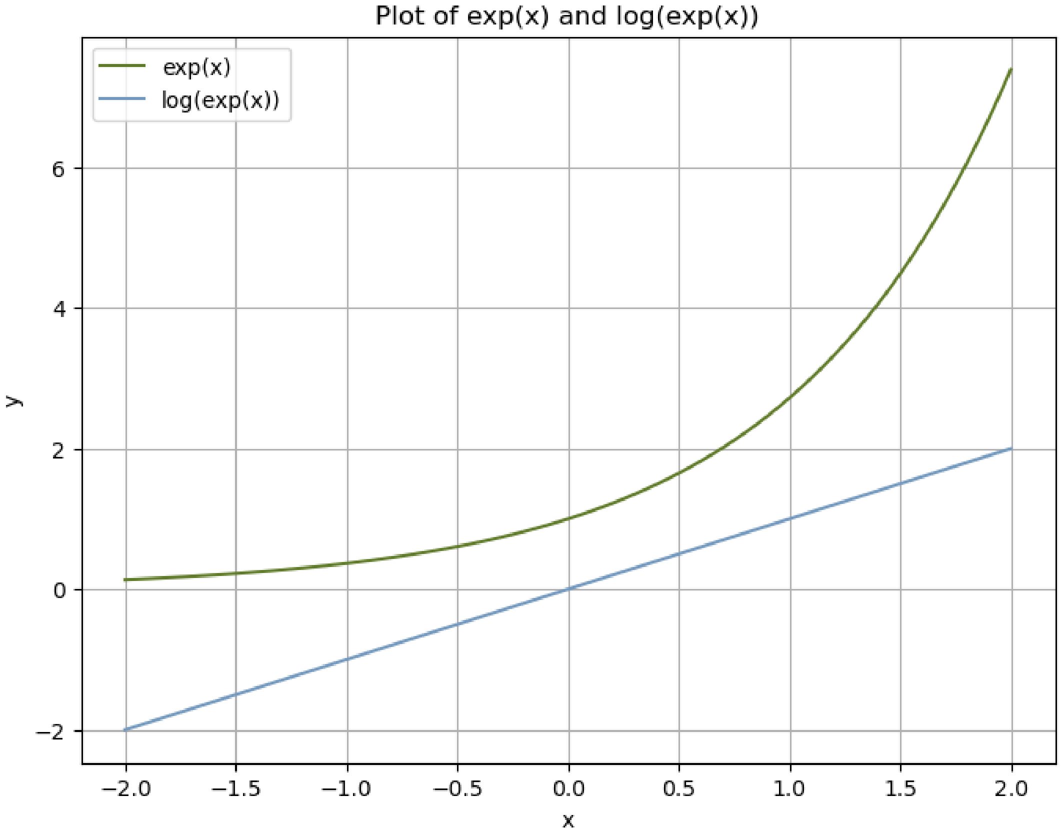
In **deep learning**, especially in neural networks for classification tasks, the Softmax function is commonly used in the final layer to produce class probabilities. The network's output is a vector of raw scores or logits, and the Softmax function is applied to convert these scores into probabilities.

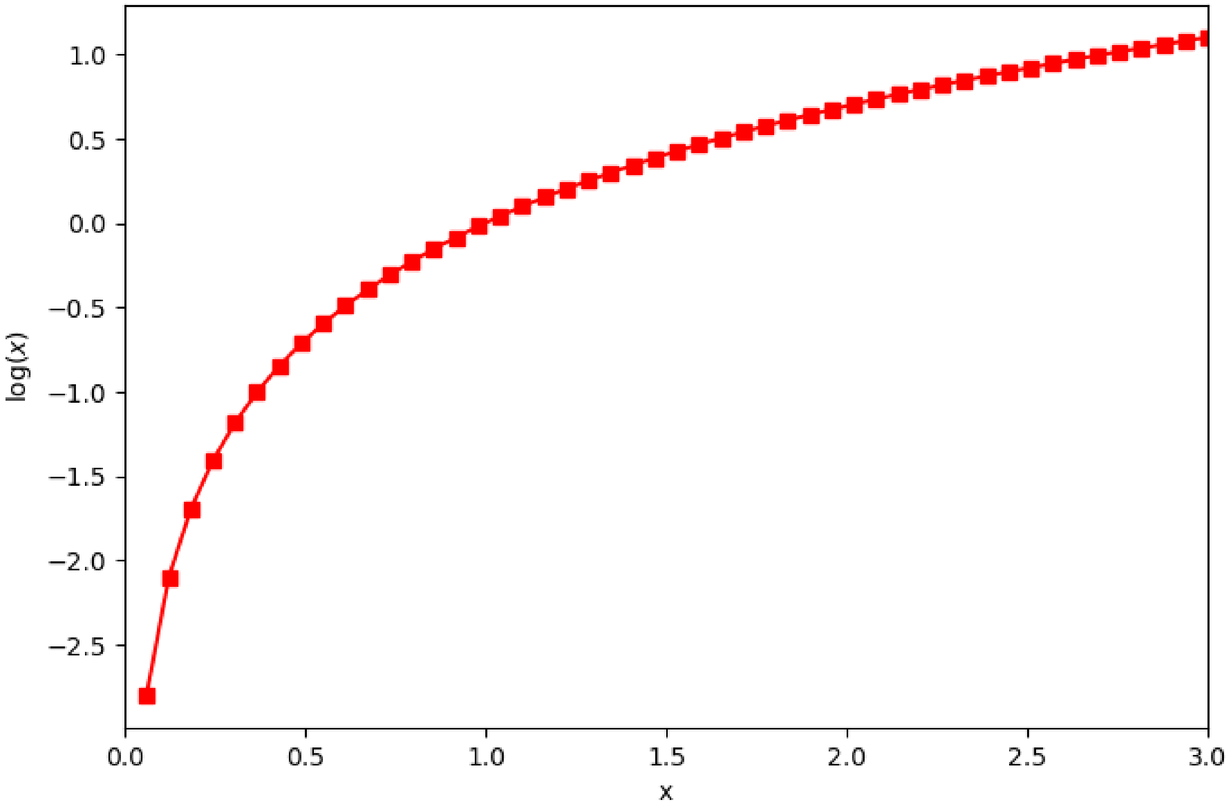
Softmax Output Features

how can we use softmax to optimize performance?

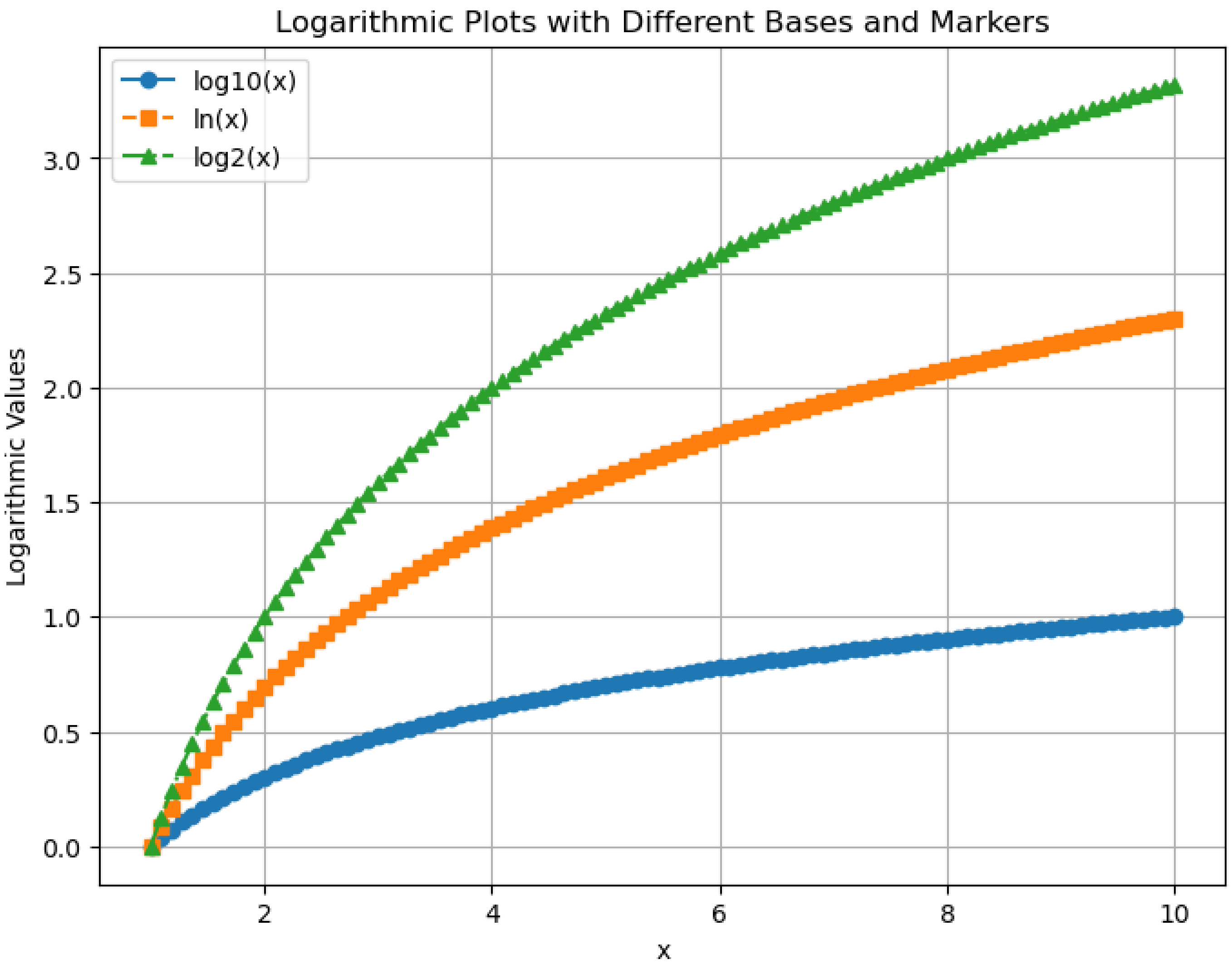


Logarithms





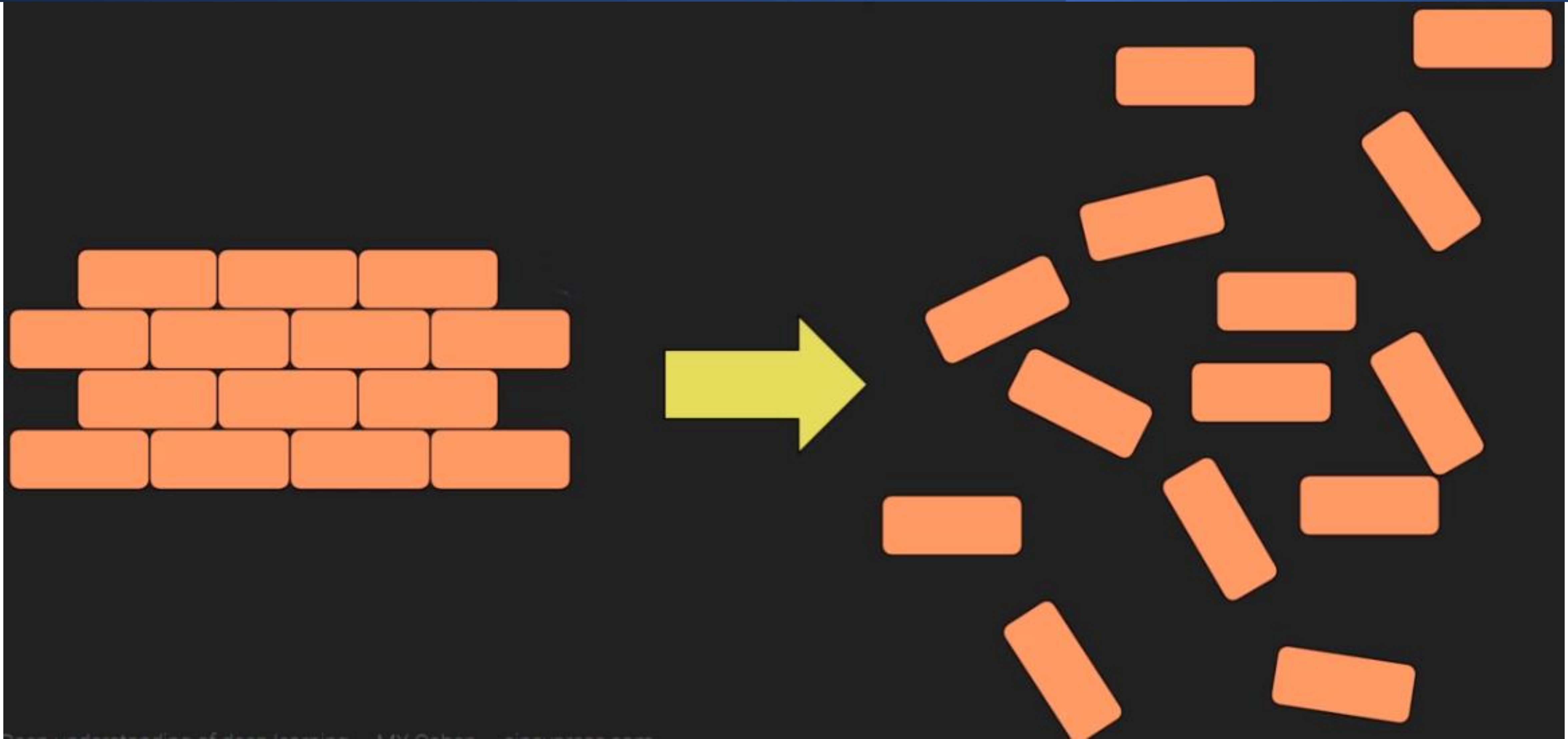
Logarithms in different Bases



Properties of the Log Function

- **Monotonic:** The log function is monotonic, which means it increases as its input increases and decreases as its input decreases.
- **Stretching Small Values:** The log function stretches out small values, making them more distinguishable when they are close to zero.
- **Numerical Precision:** In machine learning and deep learning, the log function is often used to work with probabilities and loss values.
  - Minimizing the log of small probabilities is computationally more stable and avoids precision issues.

Entropy in Physics



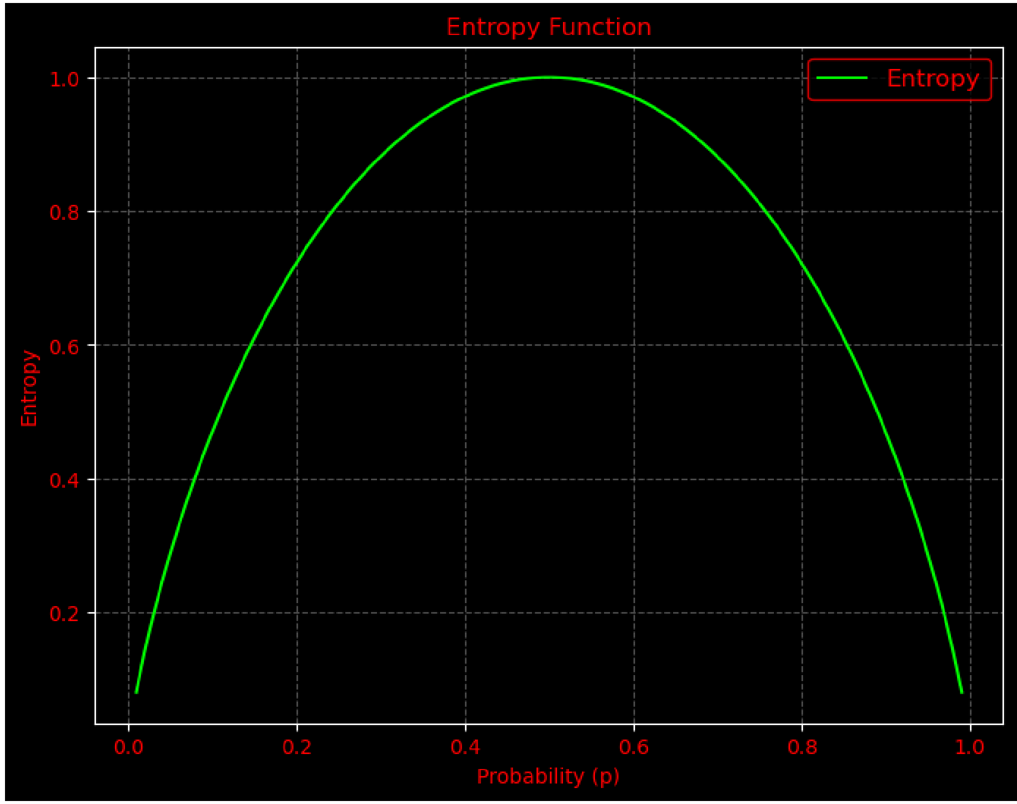
Entropy Concept

Cross Entropy

- In information theory, entropy is a fundamental concept that measures the uncertainty or randomness associated with a random variable or a probability distribution.

- It quantifies the amount of information or surprise contained in an event or a set of events.

- Entropy is often denoted by the symbol "H" and is expressed in units called "bits" when using base-2 logarithms (binary entropy) or in "nats" when using natural logarithms.



Entropy Concept ++

- **Entropy in Information Theory:** Entropy, as introduced by Claude Shannon, is a measure of the amount of surprise or uncertainty associated with a specific variable or probability distribution. It quantifies how unpredictable an event or dataset is.

- **High Entropy:** A dataset with high entropy implies high variability and unpredictability. Most values in the dataset are different, and there is more surprise.

- **Low Entropy:** Conversely, a dataset with low entropy means that most values repeat, and there is less surprise or uncertainty. The dataset is more predictable.

Entropy Formula

- ***H*(*X*)** is the entropy of the random variable X.
- ***p*(*x*)** represents the probability of observing a particular outcome *x*.
- The summation is taken over all possible outcomes *x* of the random variable.

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log(p(x_i))$$

$$H_2(X) = -\sum_{i=1}^{n} p(x_i)\log_2(p(x_i))$$

Entropy Properties

- **Uncertainty:** Entropy is a measure of uncertainty. A random variable with a high entropy is associated with a high degree of uncertainty or randomness, while a low entropy indicates a more predictable or certain random variable.
- **Maximum Entropy:** The entropy is maximized when all possible outcomes are equally likely. In this case, the system has the highest degree of uncertainty.
- **Minimum Entropy:** The entropy is minimized when one outcome is certain (probability 1) and all other outcomes have a probability of 0. In this case, the system is completely certain, and entropy is 0.
- **Units:** The choice of logarithm base (binary or natural) determines the units of entropy (bits or nats). In practice, binary entropy (bits) is commonly used in information theory and computer science.
- **Information Gain:** In the context of decision trees and machine learning, the reduction in entropy between the parent node and child nodes measures the information gain achieved by splitting the data based on a particular attribute.

Cross Entropy

- **Cross-entropy**, also known as **log loss**, is a concept used in information theory and machine learning to measure the **dissimilarity** between two probability distributions.

- It is often used in the context of **classification** problems, particularly in evaluating the performance of classification models like logistic regression and neural networks.

- Cross-entropy quantifies how well the predicted probability distribution (predicted by a model) matches the true probability distribution (**ground truth**).

Cross Entropy Formula

- ***H*(*P*,*Q*)** is the cross-entropy between the true distribution P and the predicted distribution Q.
- ***i*** represents the individual classes or labels.
- ***P*(*i*)** is the probability of class *i* according to the true distribution.
- ***Q*(*i*)** is the probability of class *i* according to the predicted distribution.

$$H(P,Q) = -\sum_{i=1}^{n} P_i \log(Q_i)$$

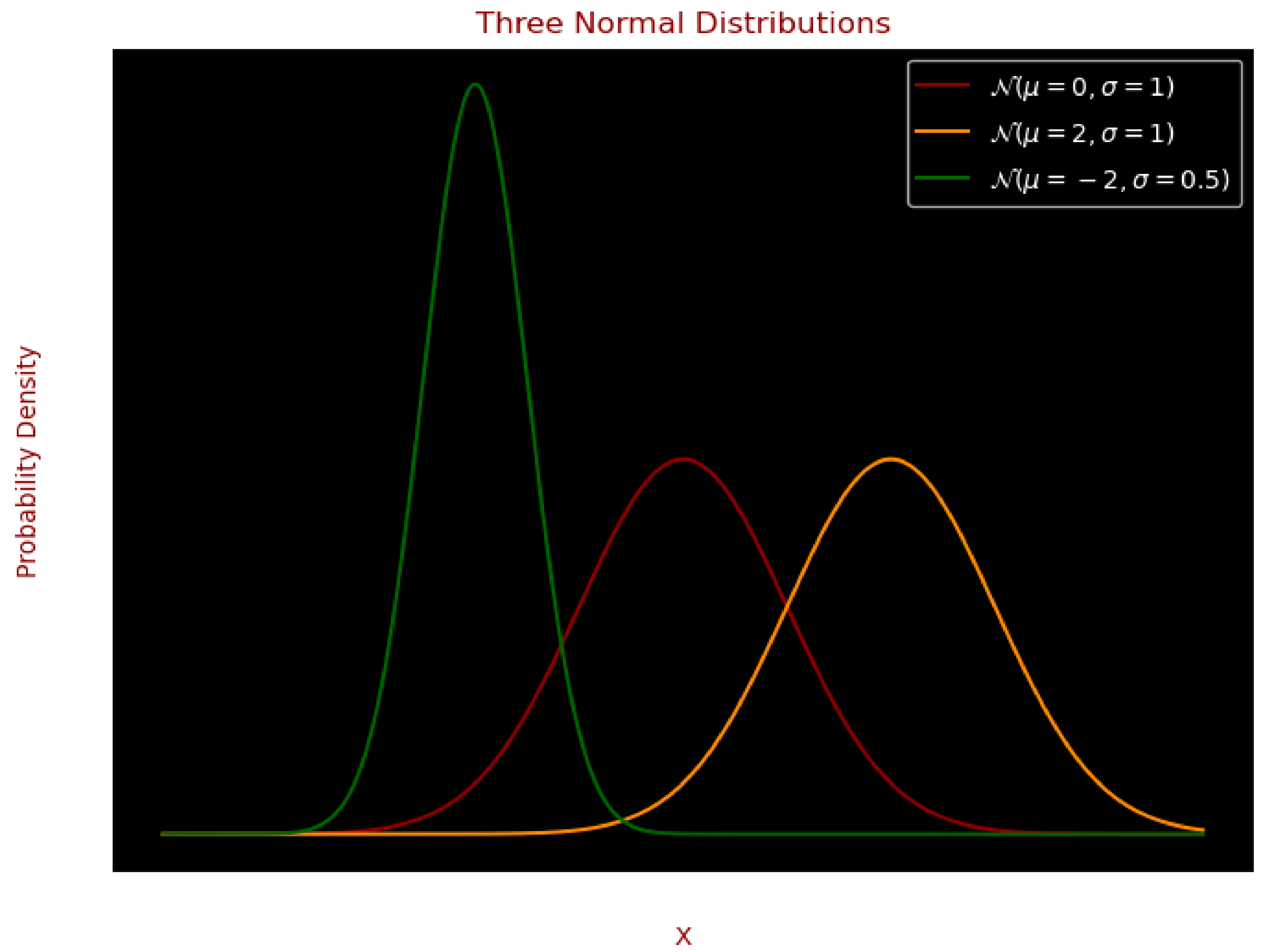
Cross Entropy in Deep Learning

- **Minimization:** In classification tasks, the goal is to minimize the cross-entropy between the true and predicted distributions. This is because a lower cross-entropy indicates that the predicted probabilities are closer to the true probabilities.
- **Perfect Match:** When the predicted probabilities exactly match the true probabilities, the cross-entropy is minimized and becomes zero.
- **Information Gain:** Cross-entropy can be seen as a measure of information gain or surprise. It quantifies how surprising the predicted distribution is given the true distribution.
- **Incorporates Uncertainty:** Cross-entropy penalizes models more heavily for making highly confident incorrect predictions. It reflects not only whether the model's predictions are correct but also how uncertain or confident those predictions are.
- **Relation to Entropy:** Cross-entropy is related to entropy (Shannon entropy). In fact, the entropy of the true distribution is the lower bound for cross-entropy. When the true distribution is known, the cross-entropy can be thought of as the extra amount of information needed to represent the true distribution using the predicted distribution.

argmin, argmax

- **argmax** is a mathematical function that returns the index of the maximum value in a given array or sequence. In other words, it finds the position of the element with the highest value. The term "argmax" stands for "argument of the maximum."
- In deep learning and machine learning, the argmax function is commonly used for various purposes:
  - **Classification:** In classification tasks, especially in multi-class classification, the argmax function is often used to determine the predicted class. For example, in a neural network's output layer, each neuron might represent a class, and the argmax function helps identify the class with the highest predicted probability.
  - **Evaluation Metrics:** In evaluating machine learning models, metrics like accuracy, precision, recall, and F1-score often require comparing the predicted class (using argmax) with the true class labels to determine whether a prediction is correct or incorrect.
  - **Reinforcement Learning:** In reinforcement learning, the argmax function is used to select actions. For example, in the Q-learning algorithm, the action with the maximum Q-value is chosen using argmax.
  - **Image Segmentation:** In image segmentation tasks, argmax is employed to assign each pixel to a specific class or category based on the predicted probabilities for each class.
  - **Natural Language Processing (NLP):** In NLP, argmax can be used to determine the most likely next word in a sequence when generating text using models like recurrent neural networks (RNNs) or transformers.

Variance and Mean



Mean (Average)

- The mean, often referred to as the average, is a measure of central tendency in a dataset.
- It is calculated as the sum of all data points divided by the number of data points.
- The mean represents the "typical" value in the dataset and provides insight into the dataset's center.

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$$

Variance

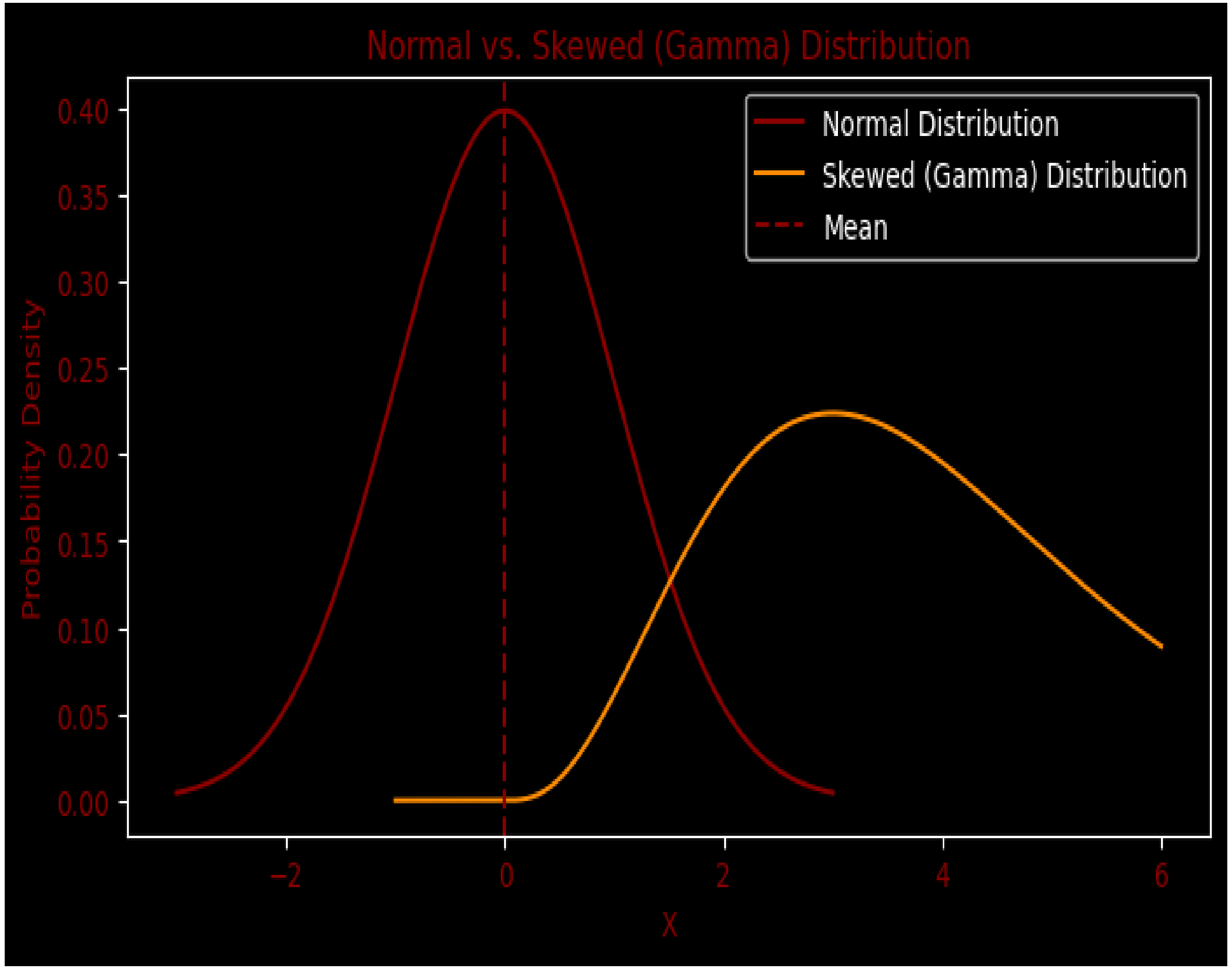
- **Variance** is a measure of the spread or **dispersion** of a dataset.
- It quantifies how much individual data points deviate from the mean.
- A higher variance indicates that data points are more spread out from the mean, while a lower variance means data points are closer to the mean.
- Variance is calculated by taking the average of the squared differences between each data point and the mean.

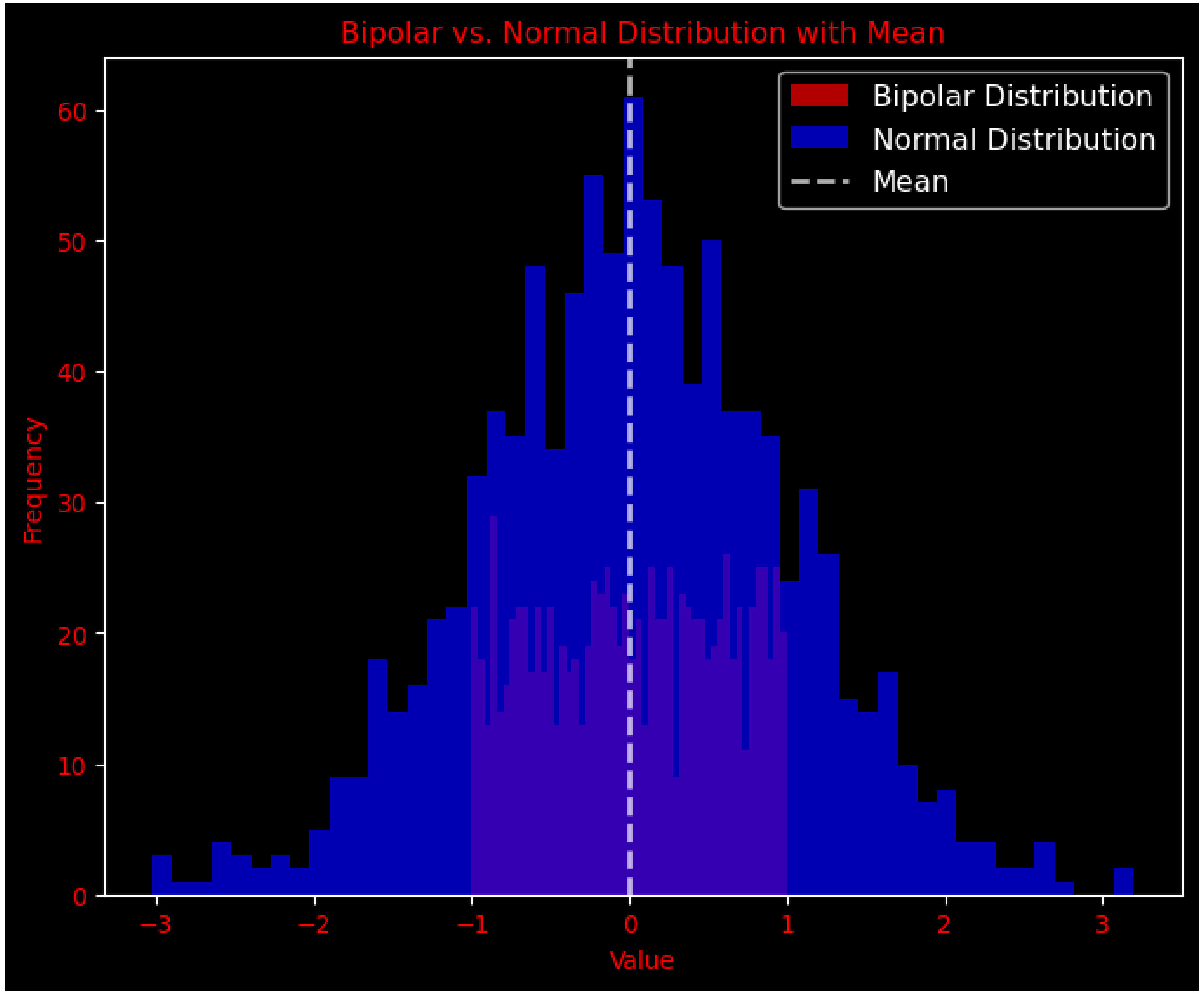
$$\mathrm{Var}(X) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$$

Mean & Variance

- **Mean:**
  - Provides a measure of central tendency.
  - Sensitive to extreme values (outliers) in the dataset.
  - Affected by every data point.
  - Often used to describe the "average" value in the data.
- **Variance:**
  - Provides a measure of dispersion or spread.
  - Measures how much data points deviate from the mean.
  - Squares the differences from the mean, so it gives more weight to larger deviations.
  - Often used to quantify the amount of variation or uncertainty in the data.

Sampling Variability





Sampling Variability

- **Sampling variability** refers to the natural variation that occurs when different samples are drawn from the same population.
- In statistics, when you collect data or observations from a population, you typically collect a sample because it's often impractical or impossible to collect data from the entire population.
- However, because each sample is a random subset of the population, the statistics computed from different samples can vary.
- **Random Sampling**: When you take random samples from a population, you introduce randomness into the process. Random sampling ensures that each observation in the population has an equal chance of being included in the sample.

Sampling Variability Moreeee

- **Variability Between Samples**: Because each sample is a random selection, different samples may contain different sets of observations. This leads to variability in the statistics calculated from each sample.
- **Statistics and Estimates**: Common statistics, such as the mean, variance, standard deviation, and confidence intervals, are subject to sampling variability. When you calculate these statistics from one sample, you get one estimate of the population parameters. Another sample from the same population may yield a different estimate due to sampling variability.
- **Understanding Uncertainty:** Sampling variability is a source of uncertainty in statistics. It means that your sample-based estimates are not fixed values but rather have a range of possible values due to random chance.
- **Importance of Large Samples:** Larger sample sizes tend to reduce sampling variability. As the sample size increases, the sample statistics become more stable and approach the true population parameters.
- **Inference:** In statistical inference, which includes hypothesis testing and confidence interval estimation, understanding sampling variability is crucial. It helps statisticians make statements about the population based on sample data while acknowledging the uncertainty inherent in the sampling process.
- **Bootstrapping**: Bootstrapping is a statistical technique used to estimate the sampling distribution of a statistic by resampling from the observed data. It provides a way to quantify sampling variability and compute confidence intervals without relying on theoretical distributions.

Sampling variability in deep learning

- **Data Collection:** In many deep learning applications, collecting data from the entire population is impractical or impossible. Instead, we work with samples of data. Understanding sampling variability helps us interpret the results and predictions made using these samples.

- **Training and Validation Sets:** When training machine learning models, we typically split the data into training and validation sets. The model learns from the training data and is evaluated on the validation set. Sampling variability in the training and validation sets can affect model performance, and understanding this variability helps us assess model robustness.

- **Generalization:** The goal of machine learning is often to build models that generalize well to unseen data. Since the training data is a sample of the population, the model's performance on new, unseen data can vary due to sampling variability. It's important to account for this variability when assessing a model's generalization capabilities.

- **Cross-Validation:** Cross-validation is a technique used to assess model performance by repeatedly splitting the data into different training and validation sets. Sampling variability in the cross-validation process can affect performance metrics, and it's essential to consider this when comparing models or selecting hyperparameters.

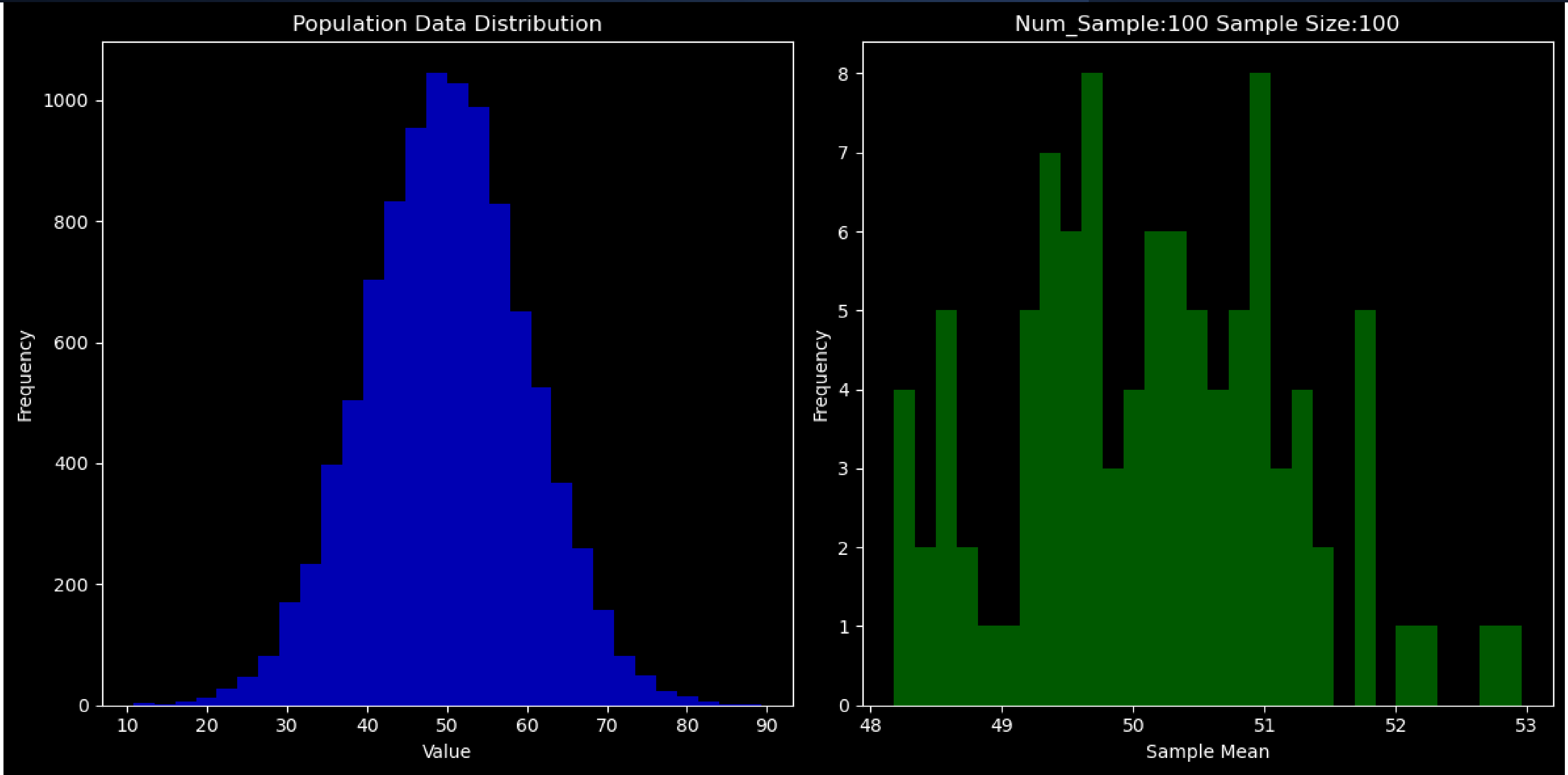
- **Uncertainty Estimation:** In some applications, it's important to quantify uncertainty in model predictions. Sampling variability is one source of uncertainty. Techniques like bootstrapping or Monte Carlo dropout leverage sampling variability to estimate prediction uncertainty.

- **Bias and Fairness:** Biases in data collection or sampling can introduce sampling variability. Understanding this variability is crucial when addressing issues related to bias and fairness in machine learning, especially in sensitive applications like healthcare or finance.

- **Ensemble Methods:** Ensemble methods combine multiple models to improve predictions. Each model in an ensemble may be trained on a different sample of data. Sampling variability among ensemble members can lead to more robust and accurate predictions.

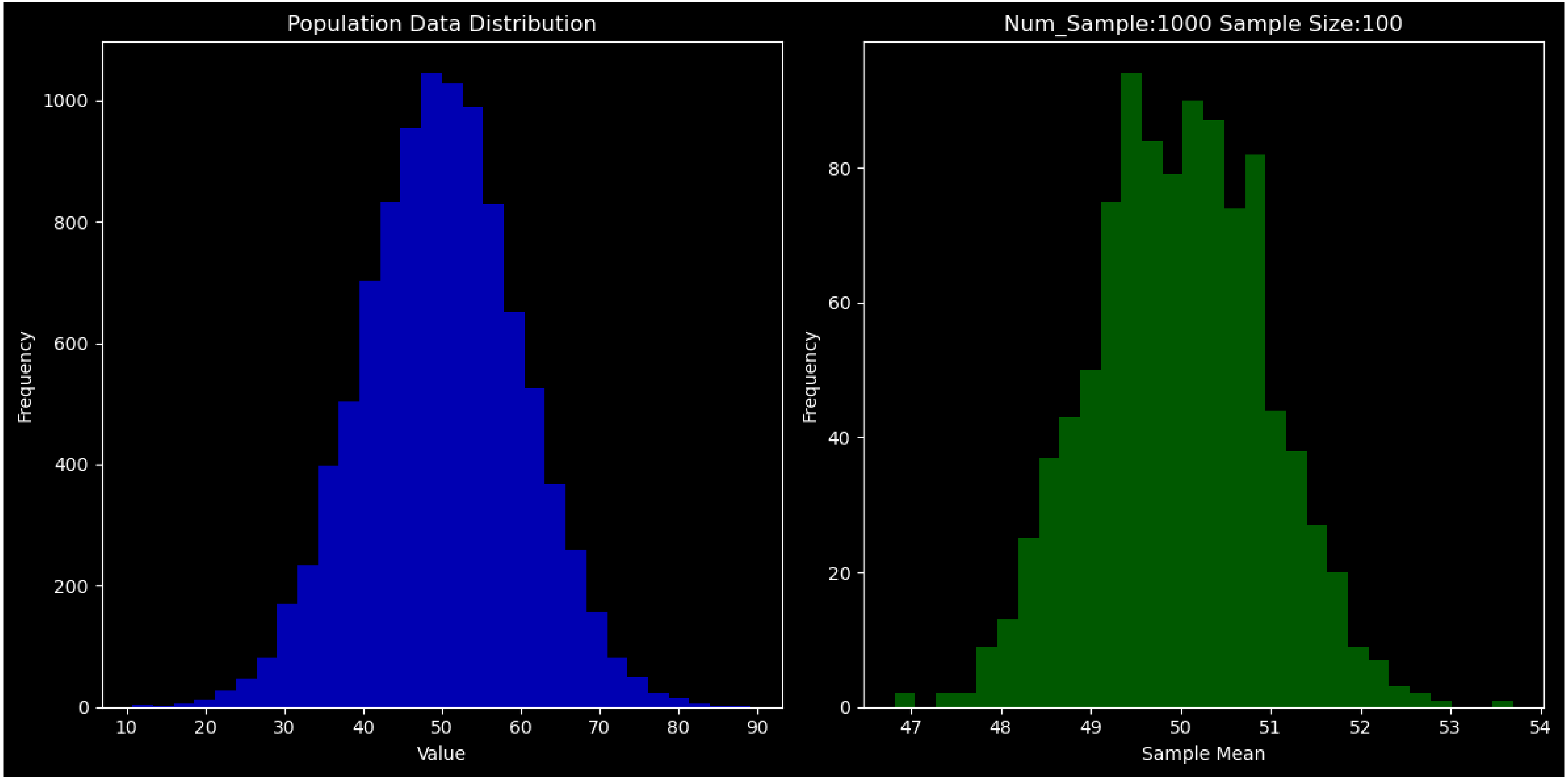
- **Statistical Significance:** When testing hypotheses or making statistical inferences using machine learning, it's important to consider the statistical significance of results. Sampling variability plays a role in hypothesis testing and confidence interval estimation.

Sampling

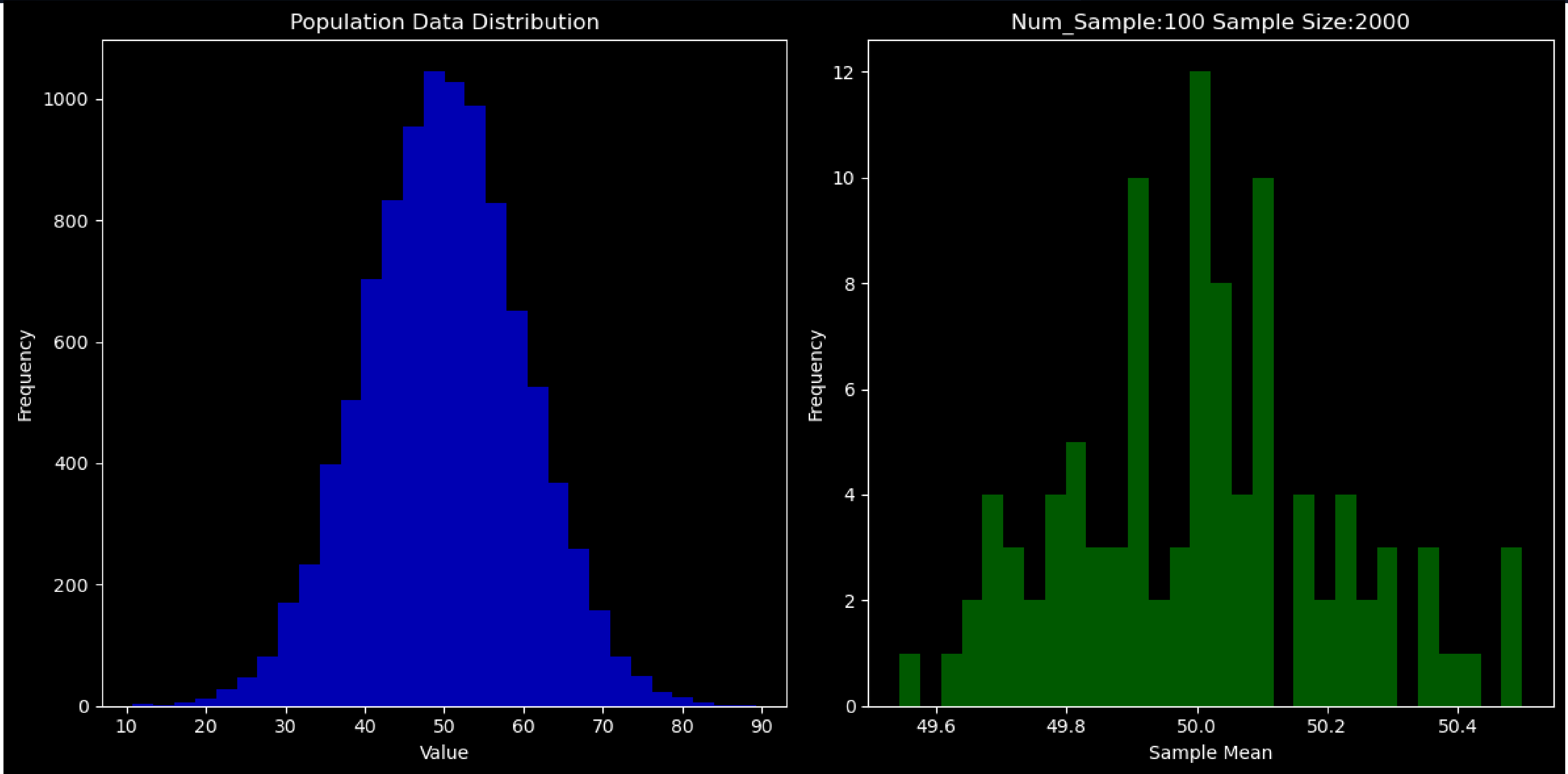


Population Data Distribution
Frequency
Value
0
200
400
600
800
1000
10
20
30
40
50
60
70
80
90
Num_Sample:100 Sample Size:100
Frequency
Sample Mean
0
1
2
3
4
5
6
7
8
48
49
50
51
52
53

Increasing the number of samples



Increasing the sample size



Reproducing randomness via seeding

- "Reproducing randomness via seeding" refers to the practice of using a random seed when generating random numbers in a program or experiment.

- The seed is a starting point for the random number generator (RNG), and by using the same seed, you can recreate the same sequence of random numbers each time you run the program or experiment. This allows for the reproducibility of random processes and results.

1. **Reproducibility:** In many scientific and computational applications, it's essential to obtain consistent and reproducible results. When dealing with random processes or algorithms that involve randomness (such as machine learning, simulations, or statistical experiments), you want to ensure that your results can be recreated and verified by others.

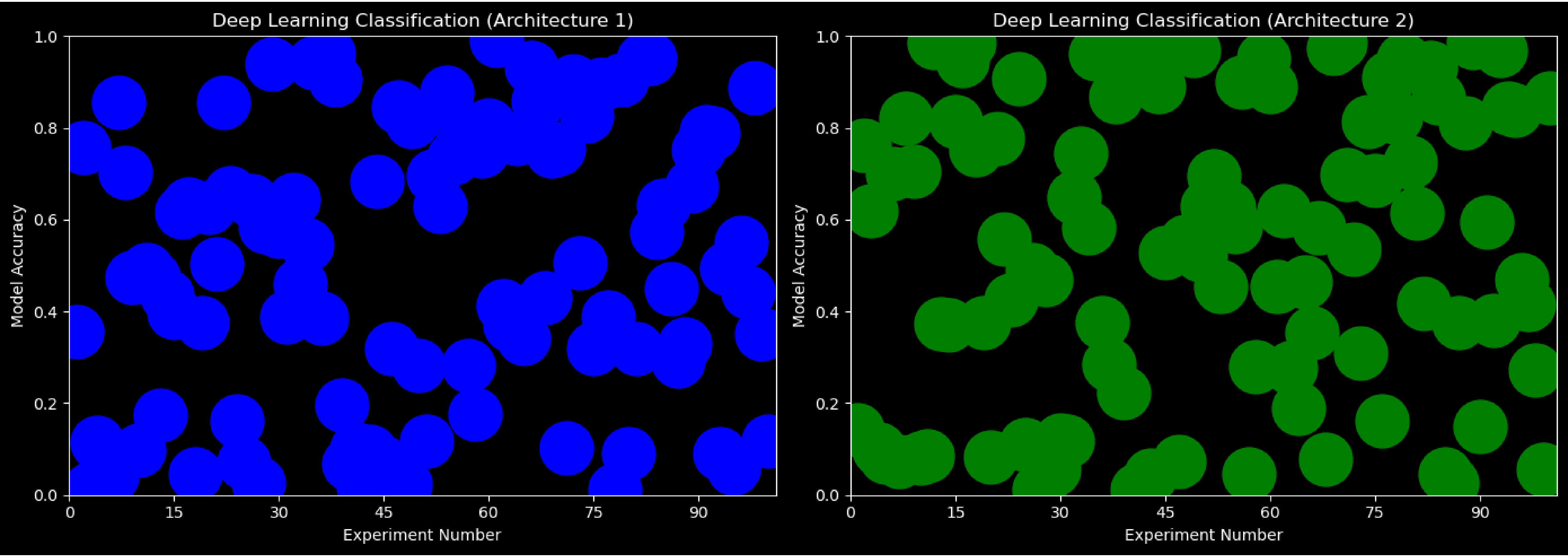
2. **Debugging:** When you encounter unexpected behavior or issues in your code or experiments, being able to reproduce the randomness allows you to isolate and debug the problem more effectively. If randomness were entirely unpredictable, it would be challenging to identify the source of issues.

3. **Comparability:** Researchers often need to compare results from different studies or experiments. Reproducible randomness ensures that the same starting conditions (seed) lead to the same results, making it easier to compare and validate findings.

4. **Testing and Validation:** Reproducible randomness is crucial for testing and validating algorithms, models, or simulations. By using a fixed seed, you can ensure that your tests are consistent across runs and that any improvements or changes to your code can be assessed systematically.

5. **Documentation and Reporting:** In scientific research and data analysis, it's common to document and report the methods used, including the random processes involved. Providing the seed value used in your experiments allows others to replicate your work and verify your findings.

t-Test


Deep Learning Classification (Architecture 1)
Deep Learning Classification (Architecture 2)
Model Accuracy
Experiment Number
1.0
0.8
0.6
0.4
0.2
0.0
0
15
30
45
60
75
90

t-Test

it measures how different between two groups of data by comparing their means

- A t-test, or **Student's t-test**, is a statistical hypothesis test used to determine if there is a significant difference between the means of two groups or populations.

- It is commonly used when you have two sets of data and want to assess whether the means of those sets are statistically different from each other.

- The t-test is widely used in various fields, including science, medicine, social sciences, and business, for comparing means and assessing the significance of observed differences.

t-test Conducting

1. **Formulate Hypotheses:** Define a null hypothesis (H0) and an alternative hypothesis (Ha). The null hypothesis typically states that there is no significant difference between the means, while the alternative hypothesis suggests there is a significant difference.

2. **Collect Data:** Obtain data from the two groups you want to compare.

3. **Calculate Test Statistic:** Compute the t-statistic using the sample data and appropriate formulas. The t-statistic measures how many standard errors the sample means are apart.

4. **Determine Degrees of Freedom:** The degrees of freedom (df) depend on the type of t-test being used and the sample sizes. It affects the shape of the t-distribution.

5. **Find Critical Value or p-Value:** Depending on your chosen significance level (alpha), look up the critical value from a t-distribution table or calculate the p-value associated with the t-statistic.

6. **Make a Decision:** Compare the calculated t-statistic or p-value to the critical value or significance level:

    1. If the t-statistic is beyond the critical value or if the p-value is less than alpha, reject the null hypothesis in favor of the alternative hypothesis.

    2. If not, fail to reject the null hypothesis.

7. **Draw a Conclusion:** Based on your decision, draw a conclusion regarding the significance of the observed differences.

Mean (Average)

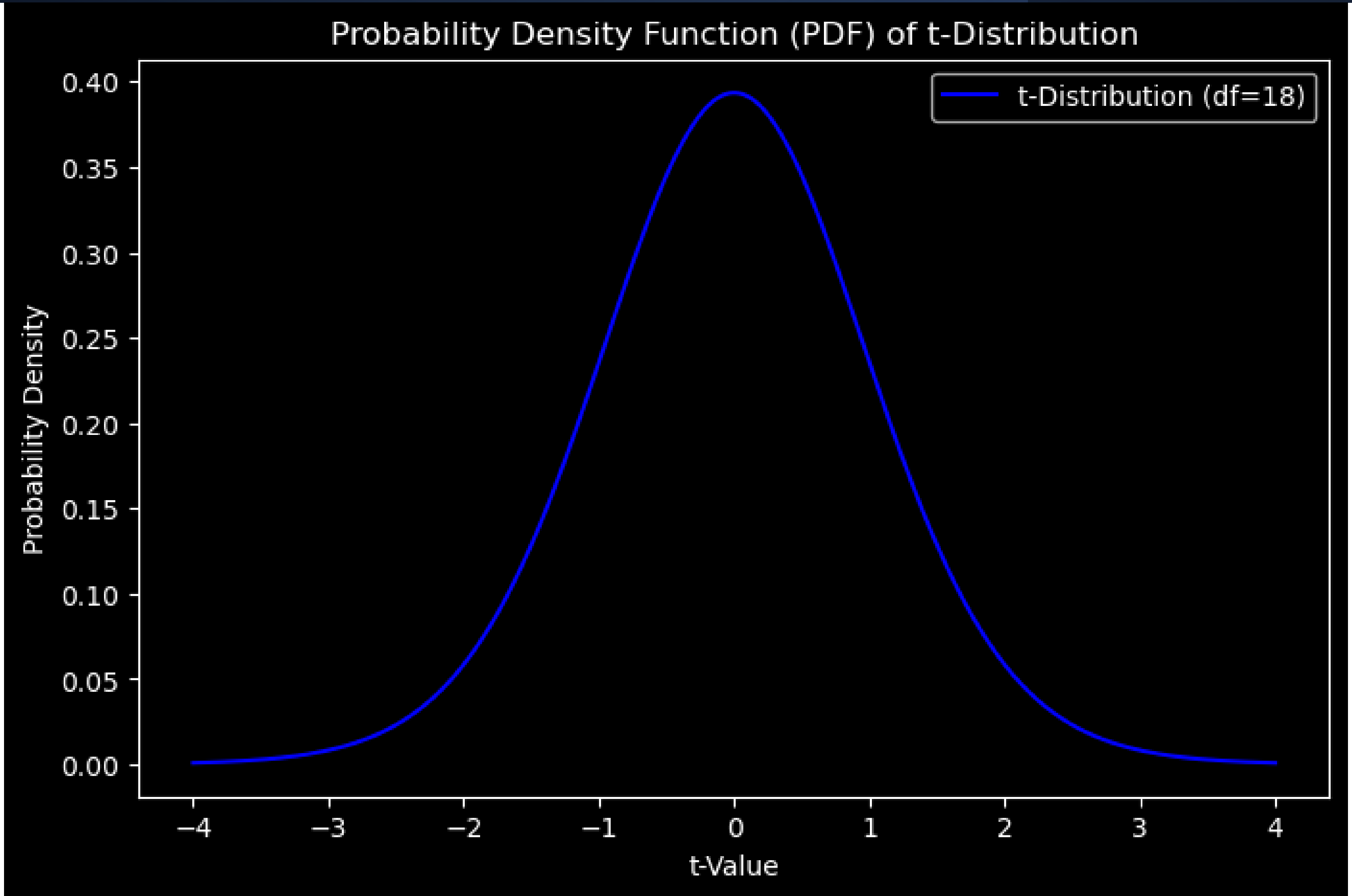
- *t* is the t-statistic.
- *X*_1 and *X*_2 are the sample means of the two groups.
- *s*1 and *s*2 are the sample standard deviations of the two groups.
- *n*1 and *n*2 are the sample sizes of the two groups

$$t = \frac{\bar{x} - \mu}{s / \sqrt{n}}$$

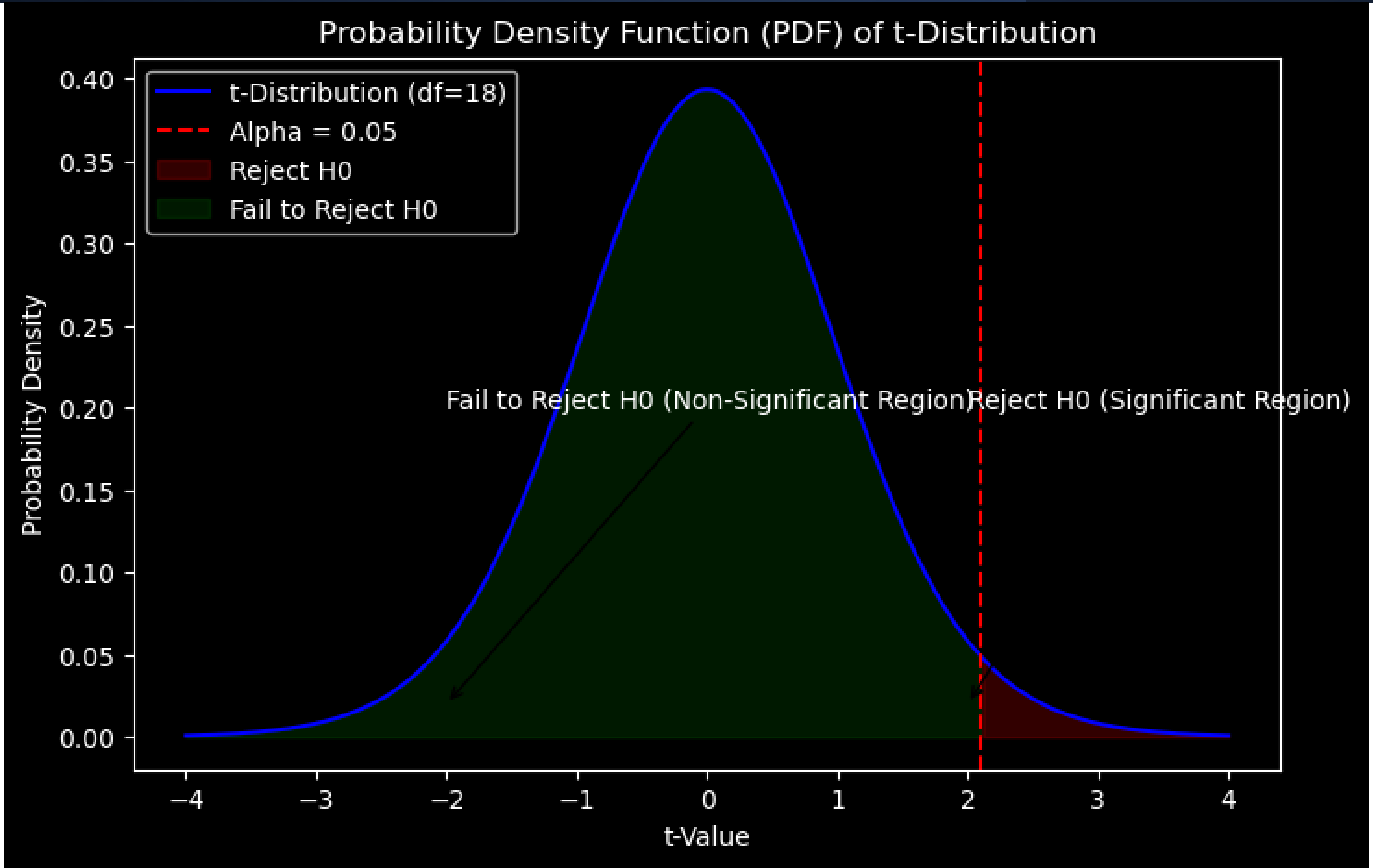
- *f*(*t*) is the PDF.
- *df* is the degrees of freedom.
- Γ(*x*) is the gamma function.

$$p(t \mid H_0) = \frac{\Gamma\left(\frac{df+1}{2}\right)}{\sqrt{df\pi}\,\Gamma\left(\frac{df}{2}\right)}\left(1 + \frac{t^2}{df}\right)^{-\frac{df+1}{2}}$$

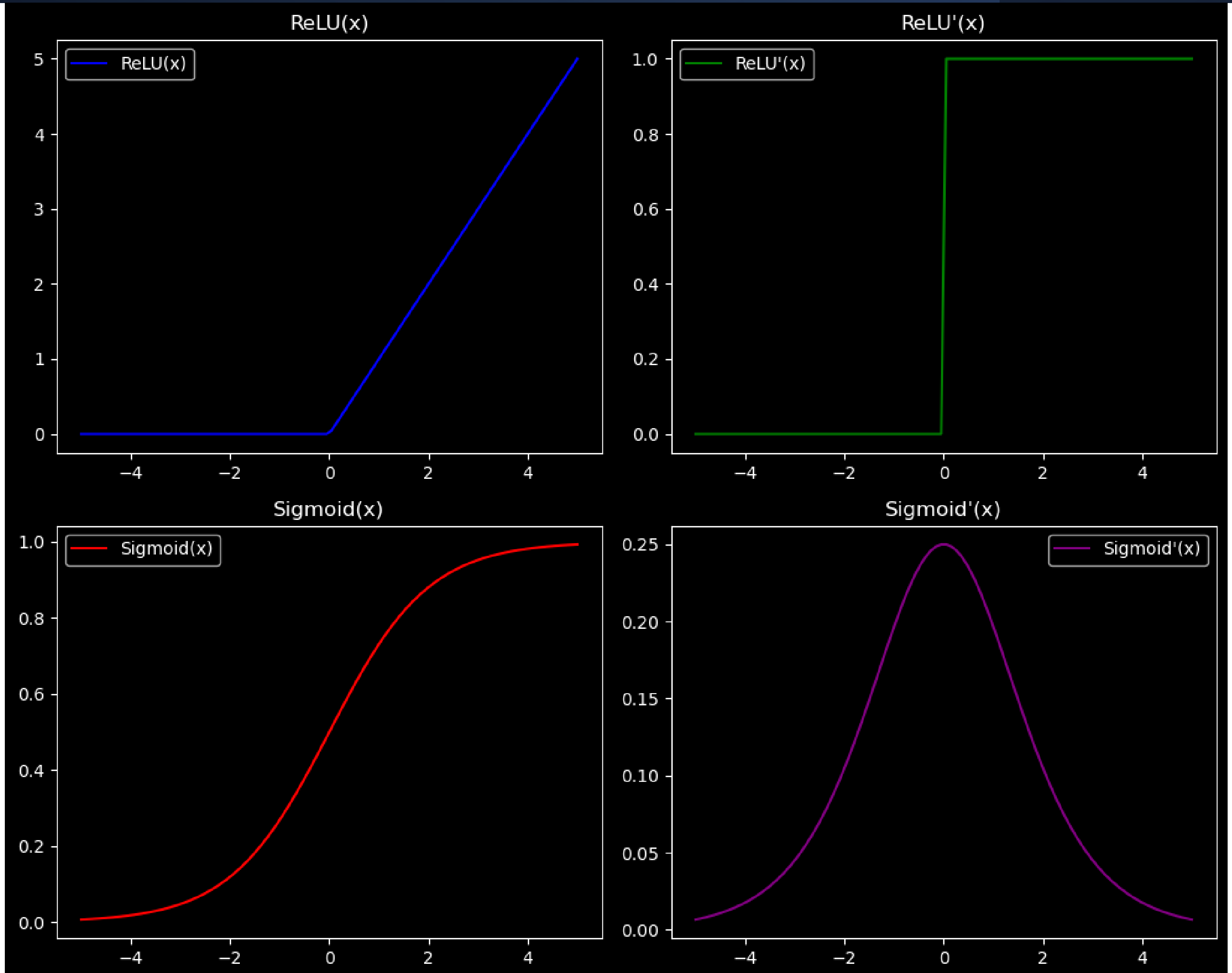
t_test PDF


Probability Density Function (PDF) of t-Distribution
t-Distribution (df=18)
Probability Density
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−4
−3
−2
−1
0
1
2
3
4
t-Value

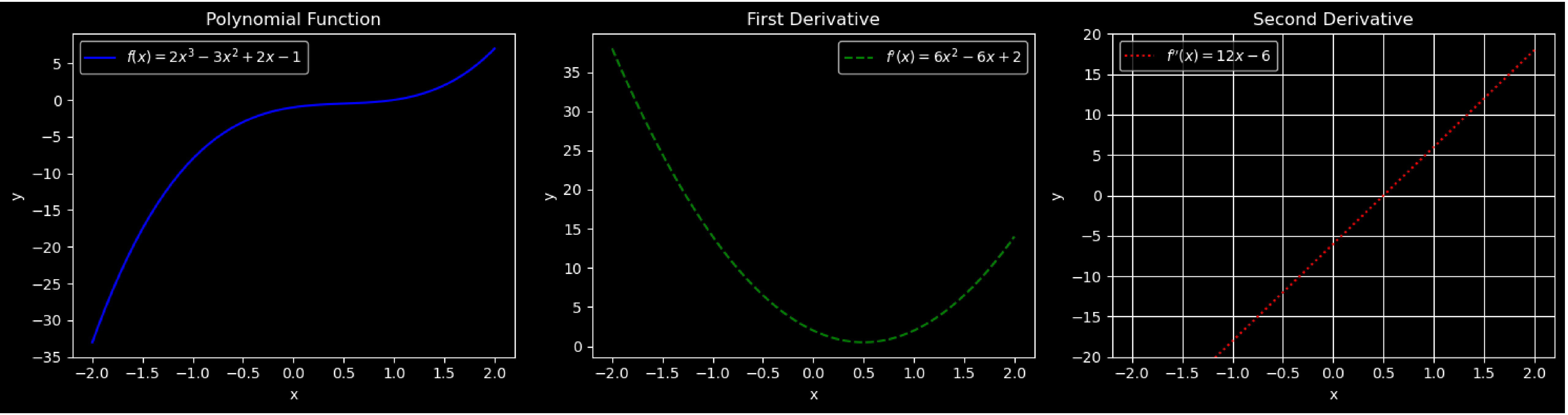
t-test Conduction



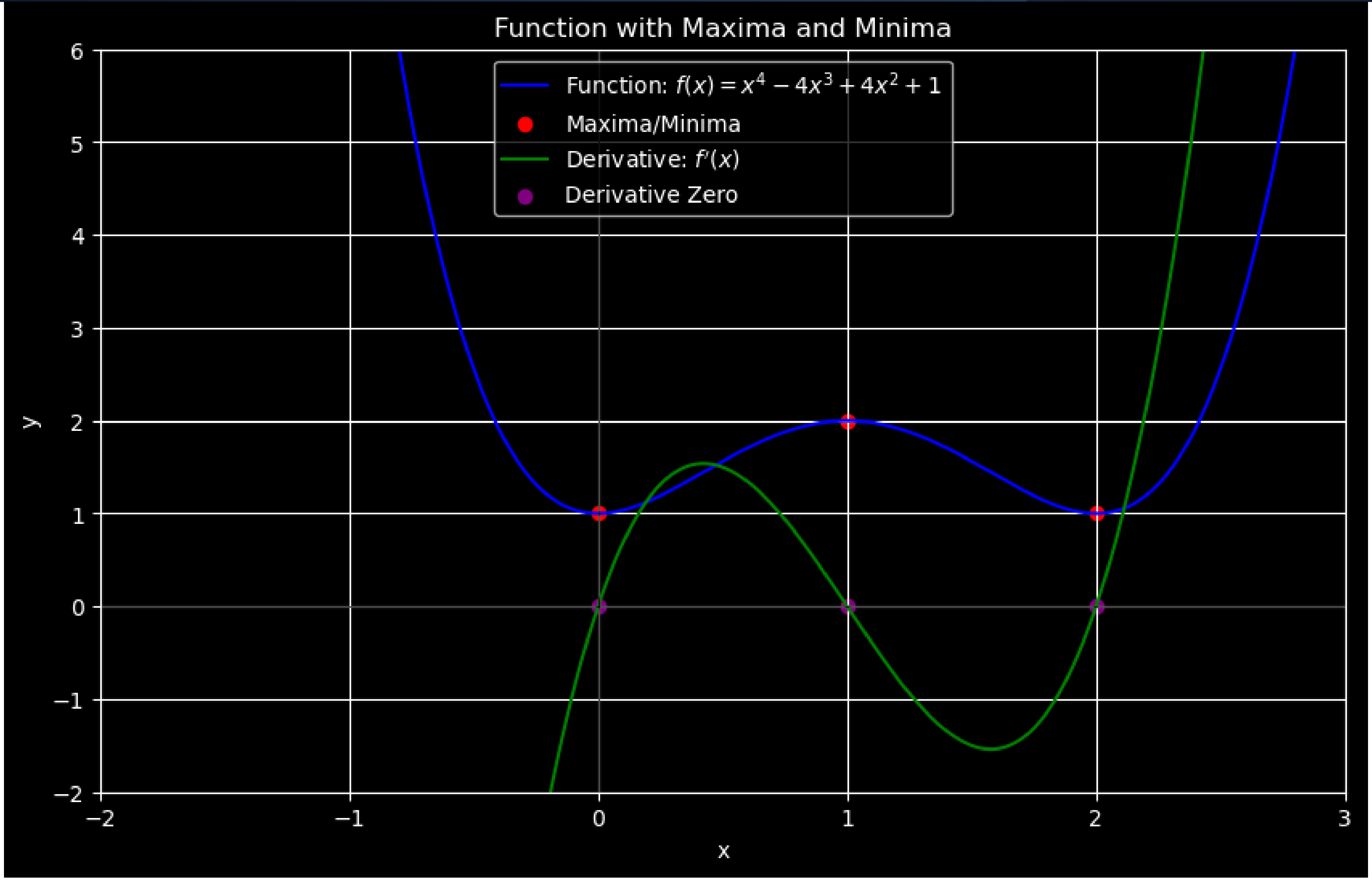
Derivatives



Derivative Examples



Minima and Maxima



Chain Rule and Product Rule

$$u(x) = 2x^2$$

$$v(x) = \sin(x)$$

$$u'(x) = 4x$$

$$v'(x) = \cos(x)$$

$$w(x) = u(x) \cdot v(x)$$

$$w'(x) = u'(x) \cdot v(x) + u(x) \cdot v'(x)$$

$$w'(x) = (4x) \cdot (\sin(x)) + (2x^2) \cdot (\cos(x))$$

$$w'(x) = (4x) \cdot (\sin(x)) + (2x^2) \cdot (\cos(x)) = 2x\,(x\cos(x) + 2\sin(x))$$

$$f(g(x)) = u^3 + 1$$

$$g(x) = 3x^2 - 2$$

$$f'(u) = 3u^2$$

$$g'(x) = 6x$$

$$f'(g(x)) = f'(u) * g'(x)$$

$$f'(g(x)) = 3u^2 * 6x = 18x(u^2)$$

$$f'(g(x)) = 18x(3x^2 - 2)^2$$

$$f'(g(x)) = 18x(3x^2 - 2)^2 = 18x\left(3x^2 - 2\right)^2$$