UNIVERSITY OF MALAYA

# Leveraging Nonlinear Systems Equation for Efficient Spam Email Detection

Group A

January 8, 2025

# Group Members

| Matrix Number | Name |
|---|---|
| S2191553 | Yerong Liu |
| U2100875 | HAFIZ AIMAN BIN SHAMSUL BAHARI |
| S2148250 | SOO WEE LIM |
| U2101770 | MUHAMMAD AIDIL SHAZWAN BIN MARZUKHI |
| S2190329 | Yulun Deng |
| S2109049 | SUMAIYA MUNTAREEN BILLAH |

# Introduction

Email remains a cornerstone of communication in the digital age, but its widespread use has led to the proliferation of spam emails, ranging from simple advertisements to malicious phishing and malware attempts. Traditional detection methods often fall short in addressing the complexity and adaptability of modern spam strategies. To tackle this challenge, integrating nonlinear systems equations into the **Backpropagation** process of machine learning models offers a promising solution. Nonlinear systems excel at capturing intricate patterns within data, enabling neural networks to better differentiate legitimate emails from spam while adapting to evolving threats. This approach enhances both detection accuracy and computational efficiency, providing a robust framework for large-scale email filtering.

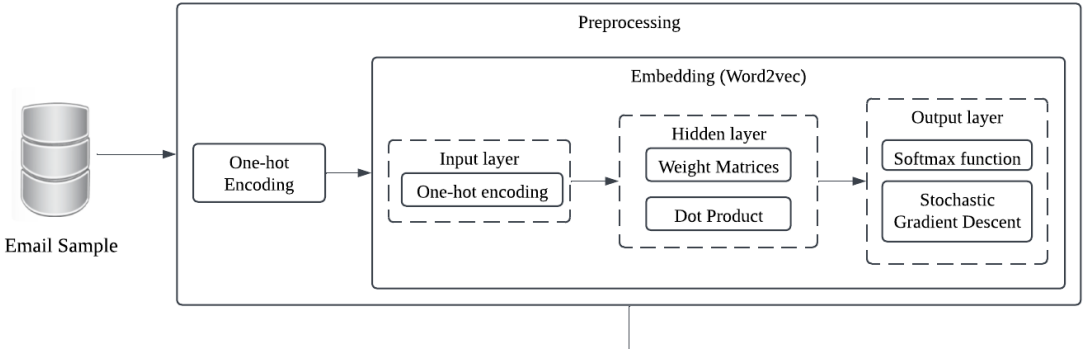
# Problem statement

With the rapid advancement of technology, the widespread adoption of large-scale AI models, and the increasing volume of spam, the prevalence of sophisticated spam has become a significant challenge. Traditional spam detection models often fail to effectively address the complexities of modern spam emails. As a result, there is a growing need for more complex and efficient spam processing models that can handle these advanced threats. To tackle this issue, leveraging nonlinear systems equations offers a promising approach to enhance the accuracy and efficiency of spam email detection.

# objectives

- **Reducing Data Noise:** By filtering out spam, we can ensure that important communications are not lost among irrelevant messages, improving the efficiency of email usage.
- **Enhance Spam Detection Efficiency:** Incorporate nonlinear system equations into the backpropagation process and loss function calculations of machine learning models to improve the model's ability to efficiently identify and filter spam emails, even as spam strategies evolve.
- **Improve User Experience and System Adaptability:** Optimize the model to maintain computational efficiency while adapting to changing spam patterns, ensuring a seamless, responsive email filtering system that enhances the overall user experience.

# Basic Flow Chart (Example)



# Basic Flow Chart (Example)

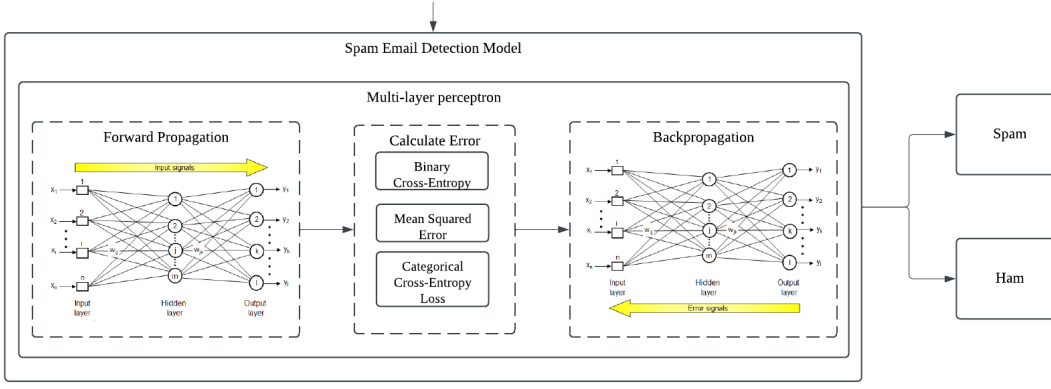



Figure: Detection Model

# Methodology

# Basic Forward

Calculate the result of each node and pass to the next layer, when we reaching the final node, we'll get the result. That's the process of forward.
The result $\mathbf{Z_1}$ is then passed through the ReLU activation function $\mathbf{f}$:
$\mathbf{Z_1} = \mathbf{XW_1} + \mathbf{b_1}$, where $\mathbf{f(x)} = \max(0, \mathbf{x})$ and $\mathbf{A_1} = \mathbf{f(Z_1)}$.
The output from the hidden layer $\mathbf{A_1}$ is passed to the next layer. It is multiplied by the weight matrix $\mathbf{W_2}$ and added to the bias $\mathbf{b_2}$, resulting in $\mathbf{Z_2} = \mathbf{A_1W_2} + \mathbf{b_2}$. The result $\mathbf{Z_2}$ is then passed through the sigmoid activation function $\mathbf{g(x)}$ to produce the final output $\mathbf{A_2}$, where

$$\mathbf{g(x)} = \frac{\mathbf{1}}{\mathbf{1} + \mathbf{e^{-x}}}, \quad \mathbf{A_2} = \mathbf{g(Z_2)}.$$

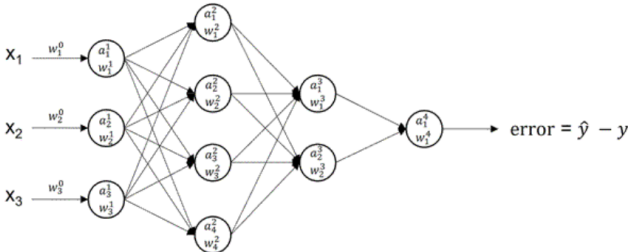
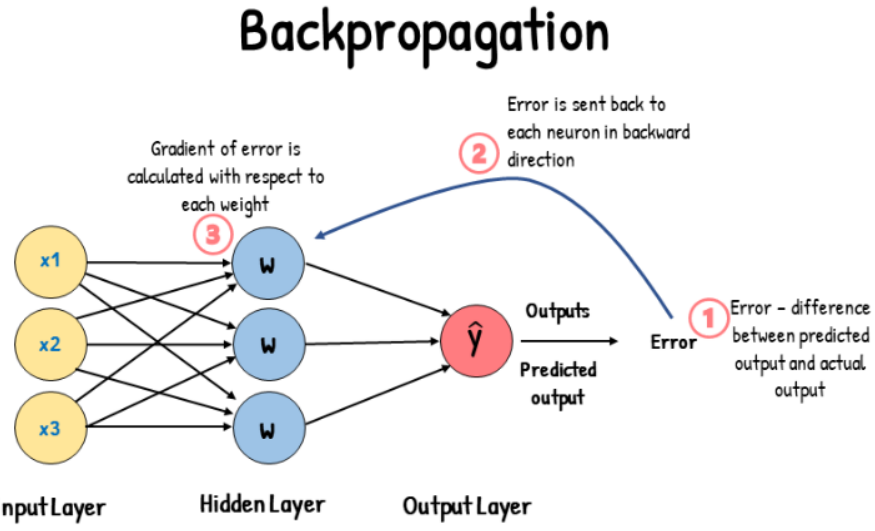



Figure: Input Layer to Hidden Layer: input vector $X$, matrix $W$, bias $b_1$.

# Training: What is Backpropagation

Short for **"backward propagation of error"**, is an algorithm for supervised learning of artificial neural networks using gradient descent.



# Training: Backpropagation

- **Calculate the output layer error:** Error of output layer:

$$E_2 = A_2 - Y$$

- **Calculate the gradient of the loss function with respect to $Z_2$:** The gradient of output layer:

$$\delta_2 = E_2 * g'(Z_2)$$

- **Calculate the gradient of hidden layer with loss function respect to $Z1$**The gradient of hidden layer:

$$\delta_1 = \delta_1 W_2^t \cdot f'(Z_1), \quad f'(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x \leq 0. \end{cases}$$

- **update the Weight and Biases:**

$$W_2 = W_2 - \eta A_1^T \delta_2, \quad b_2 = b_2 - \eta \sum \delta_2, \quad W_1 = W_1 - \eta X^T \delta_1, \quad b_1 = b_1 - \eta \sum \delta_1$$

# Evaluating: Loss Fuction-BCE 1

- Binary Cross-Entropy Loss: Binary Cross-Entropy Loss is commonly used for binary classification problems. It measures the performance of a classification model whose output is a probability value between 0 and 1.

$$E = -[\quad y\log(p) + (1-y)\log(1-p) \quad]$$

- Mean Squared Error (MSE) Loss : It measures the average squared difference between the actual and predicted values.

$$BCE_{\text{avg}} = -\frac{1}{N}\sum_{i=1}^{N}[\quad y_i\log(p_i) + (1-y_i)\log(1-p_i) \quad]$$

- Categorical Cross-Entropy Loss(CCE) :It measures the performance of a classification model whose output is a probability distribution over multiple classes.

$$\frac{\partial BCE}{\partial p} \quad = \quad \frac{p-y}{p(1-p)}$$

# Evaluating: Loss Fuction-BCE 2

Mean Squared Error (MSE) Loss: It measures the average squared difference between the actual and predicted values.

- he Mean Squared Error Loss for a single sample:

$$MSE = (y - y')^2$$

- he Mean Squared Error Loss for batch samples is:

$$MSE_{\text{avg}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - y_i')^2$$

- The gradient of MSE:

$$\frac{\partial MSE}{\partial y'} = 2(y - y')^2$$

# Evaluating: Loss Fuction-BCE 3

Categorical Cross-Entropy Loss(CCE): It measures the performance of a classification model whose output is a probability distribution over multiple classes.

- Categorical Cross-Entropy Loss for a single sample is

$$CCE = -\sum_{c=1}^{c} y_c * log(p_c)$$

- Categorical Cross-Entropy Loss for batch samples is

$$CCE_{avg} = -\frac{1}{N}\sum_{i=1}^{N\sum_{c=1}^{C}} y_{ic}\log(p_{ic})$$

- The gradient of CCE:

$$\frac{\partial MSE}{\partial y'} = 2(y - y')^2$$

# Conclusion

Our spam email detection model leverages cutting-edge techniques, combining Multi-Layer Perceptron (MLP) or Convolutional Neural Network (CNN) architectures with the integration of nonlinear systems equations into backpropagation and loss function calculations. This advanced approach enables the model to capture intricate patterns in email data, ensuring precise spam detection while adapting to evolving threats. Preprocessing steps such as tokenization, one-hot encoding, and token embedding streamline data handling, enhancing efficiency. By uniting these innovations, our system achieves exceptional accuracy, computational efficiency, and robustness, providing users with a cleaner, more secure inbox experience.

Thank You