Activation Functions in Neural Networks: A Mathematical Perspective

Nikolai Tristan Pazon
Bachelor of Science in Computer Science
University of San Carlos

October 14, 2024

Abstract

1 Introduction

This paper describes the process of my machine learning model that I created in Java. I chose java because of it is a lower level language compared to python- therefore it requires less processing power. I chose to create a machine learning model to perform what is called a churn analysis.

It is a simple problem that involves binary classification.

1.1 Churn Analysis

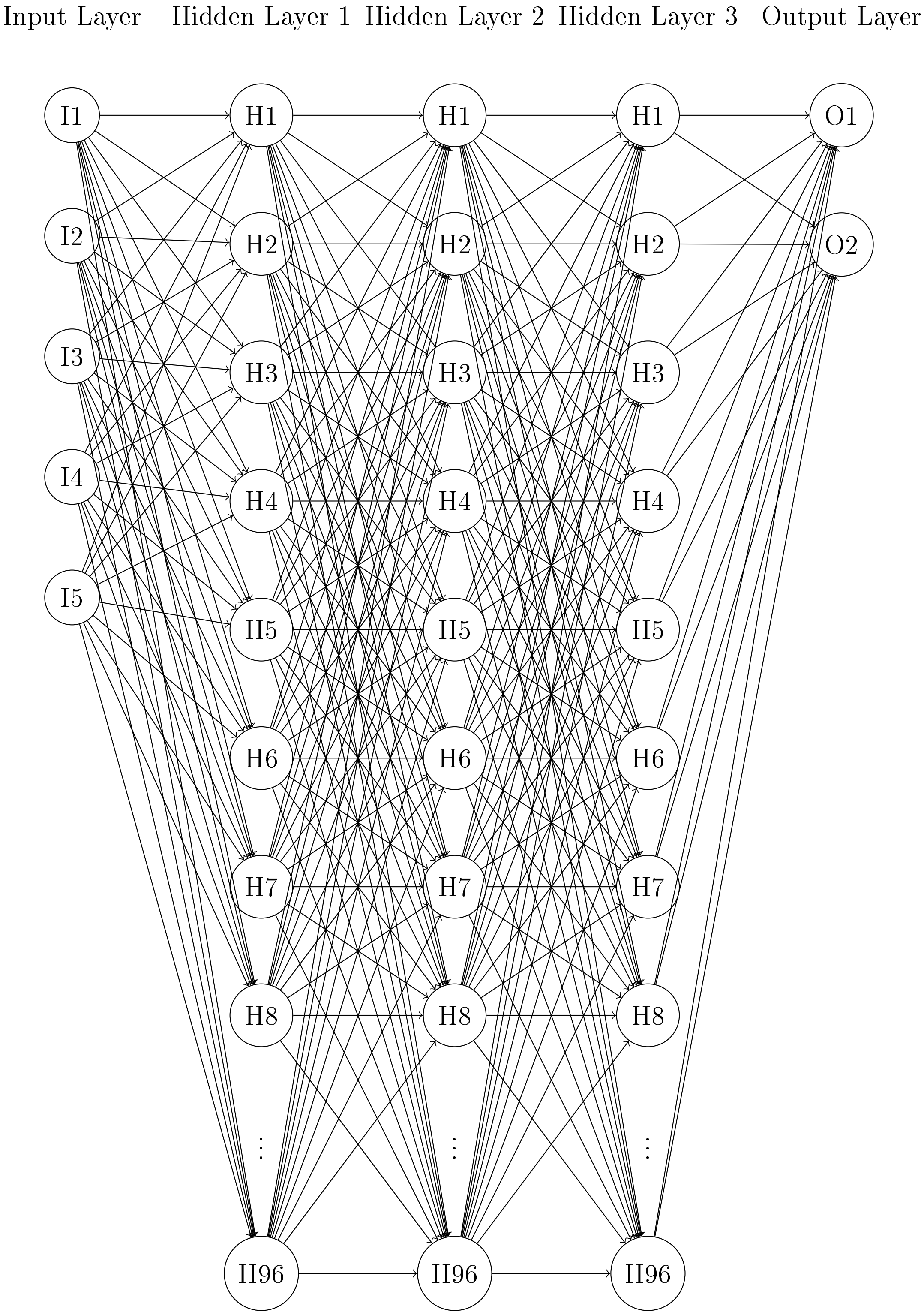
A Churn Analysis is a process in business where it refers to understanding why customers stop using their products or services. It also predicts which customers are likely to leave based on the data that a business has collected. In the customer relationship management, or CRM, it is especially important to retain customers because it is often more cost effective than finding new customers and clients. [2]

The process typically involves in identifying patters in the behavior of a customer, looking for indicators of dissatisfaction and indicators that a customer will leave. These indicators could be a decline in the engagement between the customer and the business, or it could be poor customer interaction or simply bad product/service quality. [3, 1, 6, 7]

The analysis are incredibly data-driven so methods like machine learning are a perfect use case. Machine learning is a prime tool for this as we need to analyze numerical factors that contribute to a customer's decision to stay or leave. Using this tool we can leverage the analysis to aid a business in designing customer retention strategies. [5, 4]

1.2 Model Architecture

Layer	Type	Number of Nodes	Activation Function
0	Input Layer	10	N/A
1	Hidden Layer	96	ReLU
2	Hidden Layer	96	Sigmoid
3	Hidden Layer	96	Sigmoid
4	Output Layer	2	Softmax


Input Layer
Hidden Layer 1
Hidden Layer 2
Hidden Layer 3
Output Layer
I1
I2
I3
I4
I5
H1
H2
H3
H4
H5
H6
H7
H8
⋮
H96
O1
O2

- **Input Layer:**
  - **Description:** This is the first layer of the neural network. It inputs the row data or features of the dataset into the model.
  - **Structure:** This consists of n Neurons where n is the number of features / columns in the dataset.
  - **Function:** This serves as a data entry point without affecting any of the data.

- **First Hidden Layer:**
  - **Activation Function:** Rectified Linear Unit (ReLU)
  - **Definition:** $f(x) = \max(0, x)$
  - **Purpose:** This is the first hidden layer of the model. In this layer we aim to introduce non-linearity into the model in the data. This is so that we mitigate the vanishing gradient problem and allow this model to learn complex patterns in the dataset.
  - **Structure:**96 neurons.

- **Second and Third Hidden Layers:**
  - **Activation Function:** Sigmoid
  - **Definition:** $f(x) = \frac{1}{1+e^{-x}}$
  - **Purpose:** Maps input values to a range between 0 and 1, useful for learning intermediate representations and refining features.
  - **Structure:**First: 96 Neurons, Second: 64 Neurons.

- **Final Hidden Layer:**
  - **Activation Function:** Softmax
  - **Definition:** $f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$ for each class i

  - **Purpose:** Converts raw output scores into probabilities, providing a probability distribution over possible classes, crucial for predicting customer churn.
  - **Structure:**2 neurons.

- **Output Layer:**
  - **Structure:** Consists of two neurons, each representing one class: churn and non-churn.
  - **Function:** Uses probabilities from the Softmax function to determine the class with the highest likelihood.
  - **Loss Function:** Cross-Entropy
    - **Definition:** The cross-entropy loss for a single instance is defined as $L = -\sum_i y_i \log(p_i)$, where y_i is the true label (1 for the correct class, 0 otherwise) and p_i is the predicted probability for class i.
    - **Purpose:** Measures the performance of the model by comparing predicted probabilities to actual class labels. It penalizes the model more when the predicted probability for the true class is low, encouraging the model to assign higher probabilities to the correct class.
    - **Advantages:** Cross-entropy is particularly effective for classification tasks as it provides a smooth gradient, which is beneficial for optimization algorithms like gradient descent.

1.3 Training

- **Loss Function:** Cross-Entropy
  - **Purpose:** Suitable for classification tasks, it measures model performance by comparing predicted probabilities to actual class labels, penalizing divergence from actual classes.

2 Activation Functions

2.1 Rectified Linear Unit (ReLU)

Definition: The Rectified Linear Unit (ReLU) is denoted as

$$f(x) = \max(0, x)$$

This function simply returns 0 if the input is negative and returns the input if the input is positive. In simple terms it just removes all negative values in the input.

With this, we introduce non-linearity into the model.

How does it introduce non-linearity?

- The function has a non-linear behavior and such has a non-linear shape where it has a bend in the line at x = 0.
- Breaking linear combinations of inputs. Without this function, the overall network will be linear no matter how many hidden layers there are.
- Activation Sparsity. Since the function returns 0 for all negative inputs, that means that in a layer it is probable that only some neurons will be activated for any input. This creates sparsity in the activations of the neurons.
- In this function, there is a binary nature of the function gradient. 0 for negative inputs and 1 for positive inputs. This binary gradient introduces non-linearity to the backpropagation process thus affecting how the network learns

Non-lineariy is important in neural networks as it allows it to learn and represent more complex relationships between features in the dataset. This is especially important in the case of the problem that we are trying to solve.

3 Mathematical Formulation

3.1 ReLU

Proof: Let $f(x) = \max(0, x)$

$$f(x) = max \begin{Bmatrix} 0, & x \leq 0 \\ x, & x \geq 0 \end{Bmatrix}$$

where

$$f'(x) = \begin{Bmatrix} 0 & for & x < 0 \\ 1 & for & x > 0 \\ undefined & for & x = 0 \end{Bmatrix}$$

let $x < 0$

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

$$f'(x) = \lim_{h \to 0} \frac{max(0, x+h) - max(0, x)}{h}$$

$$f'(x) = \lim_{h \to 0} \frac{0 - 0}{h} = 0$$

let $x > 0$

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

$$f'(x) = \lim_{h \to 0} \frac{max(0, x+h) - max(0, x)}{h}$$

$$f'(x) = \lim_{h \to 0} \frac{x + h - x}{h} = 1$$

let $x = 0$

left hand limit

$$f'(0) = \lim_{h \to 0^-} \frac{f(0+h) - f(0)}{h}$$

$$f'(0) = \lim_{h \to 0^-} \frac{max(0, 0+h) - max(0, 0)}{h}$$

$$f'(0) = \lim_{h \to 0^-} \frac{0}{h} = 0$$

right hand limit

$$f'(0) = \lim_{h \to 0^+} \frac{f(0+h) - f(0)}{h}$$

$$f'(0) = \lim_{h \to 0^+} \frac{max(0, 0+h) - max(0, 0)}{h}$$

$$f'(0) = \lim_{h \to 0^+} \frac{h}{h} = 1$$

3.2 Sigmoid

Proof:

Let:

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

$$\sigma'(x) = \frac{d}{dx}\left(\frac{1}{1+e^{-x}}\right)$$

$$or$$

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

$$f'(x) = \frac{(u'(x)v(x) - u(x)v'(x))}{(v(x))^2}$$

where

$$\begin{Bmatrix} u(x) = 1, & v(x) = 1 + e^{-x} \\ u'(x) = 0, & v'(x) = -e^{-x} \end{Bmatrix}$$

$$\sigma'(x) = \frac{0(1+e^{-x}) - 1(-e^{-x})}{(1+e^{-x})^2}$$

$$\sigma'(x) = \frac{e^{-x}}{(1+e^{-x})^2}$$

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

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

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

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

$$Therefore$$

$$\begin{Bmatrix} x^{+} & \rightarrow & \infty, & \sigma(x) & \rightarrow & 1 \\ x^{-} & \rightarrow & -\infty, & \sigma(x) & \rightarrow & 0 \end{Bmatrix}$$

$$Where$$

$$\left\{ x = 0, \quad \sigma(x) = \tfrac{1}{2} \right\}$$

3.3 Softmax

Let: k = number of classes

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}}$$

$$\frac{\partial f(x_i)}{\partial x_j} = \begin{cases} \dfrac{e^{x_i}\sum_{j=1}^{k} e^{x_j} - e^{x_i}e^{x_j}}{(\sum_{j=1}^{k} e^{x_j})^2} & \text{if } i = j \\ -\dfrac{e^{x_i}e^{x_j}}{(\sum_{j=1}^{k} e^{x_j})^2} & \text{if } i \neq j \end{cases}$$

Let: $S = \sum_{j=1}^{k} e^{x_j}$

$$let \quad i = j$$

$$\frac{\partial f(x_i)}{\partial x_i} = \frac{\partial}{\partial x_j}(\frac{e^{x_i}}{S})$$

$$= \frac{e^{x_i} \cdot S - e^{x_i} \cdot e^{x_j}}{(S)^2}$$

$$= \frac{e^{x_i}(S - e^{x_i})}{(S)^2}$$

$$= \frac{e^{x_i}}{S} - [\frac{e^{x_i}}{S}]^2$$

$$= f(x) - (f(x))^2$$

$$= f(x_i)(1 - f(x_i))$$

$$\frac{\partial f(x_i)}{\partial x_j} = \frac{e^{x_i} \cdot S - e^{x_i} \cdot e^{x_j}}{(S)^2} = \frac{e^{x_i}\sum_{j=1}^{k} e^{x_j} - e^{x_i} \cdot e^{x_j}}{(\sum_{j=1}^{k} e^{x_j})^2}$$

$$let \quad i \neq j$$

$$\frac{\partial f(x_i)}{\partial x_j} = \frac{\partial}{\partial x_j} \cdot \frac{e^{x_i}}{S}$$

$$= -\frac{e^{x_i} \cdot e^{x_j}}{(S)^2} = -\frac{e^{x_i}}{S} \cdot \frac{e^{x_j}}{S}$$

$$= -f(x_i) \cdot f(x_j)$$

4 Comparison of Activation Functions

5 References

References

[1] Adnan Amin, F. Al-Obeidat, B. Shah, A. Adnan, Jonathan Loo, and S. Anwar. Customer churn prediction in telecommunication industry using data certainty. *Journal of Business Research*, 2019.

[2] Jonathan Burez and D. V. Poel. Crm at a pay-tv company: Using analytical models to reduce customer attrition by targeted marketing for subscription services. *Expert Systems with Applications*, 2007.

[3] P. Lalwani, M. Mishra, Jasroop Singh Chadha, and Pratyush Sethi. Customer churn prediction system: a machine learning approach. *Computing*, 2021.

[4] N. Lu, Hua Lin, Jie Lu, and Guangquan Zhang. A customer churn prediction model in telecom industry using boosting. *IEEE Transactions on Industrial Informatics*, 2014.

[5] Jitendra Maan and Harsh Maan. Customer churn prediction model using explainable machine learning. *arXiv preprint arXiv:2303.00960*, 2023.

[6] M. Makhtar, S. Nafis, M. A. Mohamed, M. K. Awang, Mohd Nordin Abdul Rahman, and M. M. Deris. Churn classification model for local telecommunication company based on rough set theory. *Semantic Scholar*, 2018.

[7] V. Umayaparvathi and K. Iyakutti. A survey on customer churn prediction in telecom industry: Datasets, methods and metrics. *Semantic Scholar*, 2016.