Barium: Reinventing Human-Machine Interaction through Advanced Gesture Recognition

Daniel R. Alvarenga*
danielralvs@proton.me

Alvaro Richard*
alvarorichard@proton.me

Vitor Eduardo S. de Carvalho*
vitorcarvalho@proton.me

Robson Cardoso*
robsoncardoso@proton.me

Abstract

The "Barium" project emerges as a pioneering endeavor in the realm of Human-Machine Interfaces (HMI), harnessing the prowess of neural networks, machine learning, and deep learning to track and interpret human body movements, notably hand gestures. This paper delineates the development and application of a 4D neural network training approach, where time is regarded as a crucial dimension, heralding groundbreaking prospects in diverse technological domains. Developed in Python, Barium activates through webcam-captured hand gestures, facilitating user interactions with operating systems via predefined actions and a virtual mouse.

1 Neural Network Models

For the development of this project, two advanced neural network models were elaborated. The first model employs three-dimensional convolution layers (Conv3D), which are exceptionally suitable for processing video data, owing to their ability to capture both spatial and temporal characteristics. The second model, of a sequential nature, is focused on optimizing performance. This model incorporates the scientific core of the project, demonstrating an innovative and efficient approach in the processing and analysis of video data.

1.1 Conv3D Model (Conventional Model)

The Conv3D Model, specifically designed for the processing of three-dimensional objects, stands out for its effectiveness in videos, resembling Recurrent Neural Network (RNN) models in terms of performance. However, its applicability is limited due to the requirement of intensive computational processing. Next, we present a detailed overview of this model, including the structure of its layers, the activation functions used, and the number of neurons in each segment:

1.2 Flatten Model (Efficient Model)

The Flatten Model represents a revolutionary milestone in the field of video processing. This sequential model, notable for its efficiency and speed, stands out for the innovative way it collects and processes data. The architecture of its neural network layers is designed to maximize efficiency, allowing users to reconfigure and retrain the network in a matter of seconds to incorporate new movements. This ability for rapid adaptation is a significant advancement, making the Flatten Model a powerful tool for dynamic and real-time applications.

2 Dataset Construction

The construction of a robust and representative dataset played a crucial role in the development of "Barium". This dataset is not only the backbone of the system but also the key to its authenticity and operational freedom. The efficacy of the project's neural network, which is responsible for the precise processing and interpretation of gestures, immensely depends on the quality and variety of the collected data.

2.1 Creation of the Collector

Gesture Capture and Processing: We developed a collector using a Python script to record movements and translate them into a numerical format. This allows for consistent representations of gestures to be encoded accurately.

Video Feature Extraction: By decomposing each video into individual frames, we extract relevant data to build a diverse dataset, reflecting natural variations in the execution of gestures.

Conversion to Numerical Data: Converting visual information from frames into a set of numerical coordinates (x, y), as illustrated in our graphs, represents hand movements over time.

Normalization and Standardization: We apply normalization and standardization techniques to reduce variability between samples, which is essential for the machine learning model to generalize from the collected data. The normalized coordinates (x', y') are calculated as:

$$x' = \frac{x - \mu_x}{\sigma_x}, \quad y' = \frac{y - \mu_y}{\sigma_y} \tag{1}$$

where μ_x and μ_y are the means of the x and y coordinates across the dataset, and σ_x and σ_y are their respective standard deviations.

2.2 Labeling and Data Annotation

Labeling and annotating our dataset was carried out with the following methodology: each gesture is numerically represented as a line in a CSV file, where "X" symbolizes the gesture data and "Y" a unique numerical identifier for each type

of movement. The structure can be described by a regular expression to facilitate understanding:

```
(([")[(][0-9]+[,][ ][0-9]+[)]["][,]){22}[0-9]+[.][0-9]+[,]){20}
```

This approach allows the entire dataset to be submitted for training and testing of the neural network in a single file, ensuring flexibility and ease in data handling. The simplicity of data ingestion contrasts with the abstract complexity of the dataset's structure and its labeling. A single file containing a complete dataset for computational video vision, in numerical format, represents a significant advancement and provides an efficient alternative for neural network training.

2.3 Neural Network Input

The input to our neural networks is designed to capture the complexity and dynamics of human gestures. The dataset features are detailed in Table 1, which represents the 20-dimensional input vector for our neural network.

Feature	Description
$p0\_0$	Normalized x-coordinate of point 0
$p1\_1$	Normalized y-coordinate of point 1
$p2\_4$	Normalized x-coordinate of point 2
$p18\_20$	Normalized x-coordinate of point 18
$p19\_18$	Normalized y-coordinate of point 19

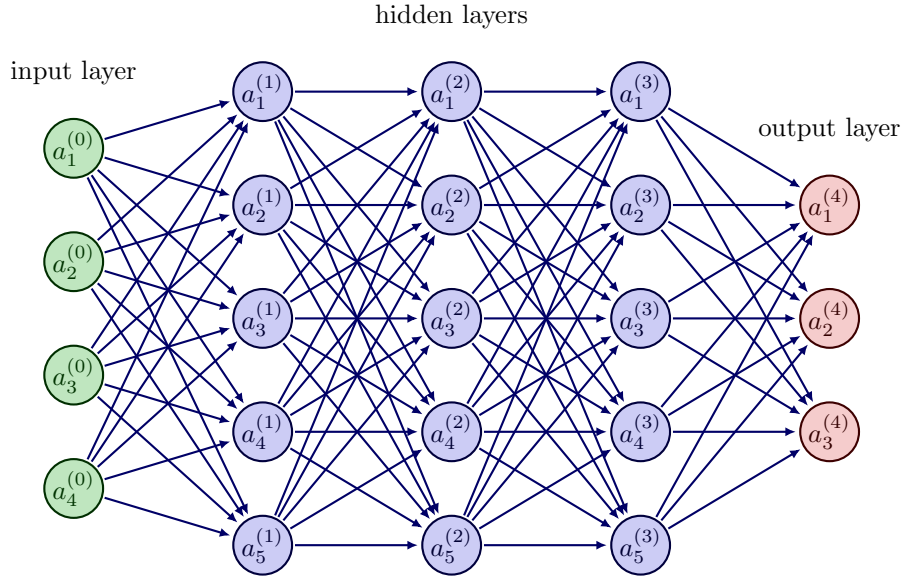
Table 1: The 20-dimensional input vector representing hand gesture features and time represent the last number.

Each row in the table corresponds to a feature extracted from the video frames, encapsulating the spatial positioning of hand landmarks. These features are subsequently flattened into a vector form for input into the neural network.

2.4 Optimization and Model Validation

To ensure the efficiency and efficacy of the neural network, we adopted a rigorous approach to optimization and validation. We used techniques such as cross-validation and hyperparameter tuning to find the ideal configuration that provides the best accuracy without falling into the trap of overfitting. Moreover, we employed regularization and dropout techniques to better generalize our model. Validation is performed with an independent test dataset, ensuring that the model can generalize well to data not seen during training.

2.5 Neural Network Diagram



2.6 Activation Functions

Activation functions introduce non-linearity to the neural network, enabling it to learn complex patterns. In our models, we use the following activation functions:

ReLU (Rectified Linear Unit): Applied to hidden layers, ReLU introduces non-linearity, enhancing the network's ability to capture a wide range of phenomena:

$$f(x) = \max(0, x) \tag{2}$$

Sigmoid: Often used in the output layer for binary classification, it squashes the input to a range between 0 and 1, suitable for binary outcome predictions:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

Softmax: Utilized in the output layer for multi-class classification, Softmax provides a probability distribution over various classes:

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}} \tag{4}$$

for $i = 1, \ldots, K$, where K is the number of classes.

3 Conclusion

"Barium" stands as a significant milestone in human-computer interaction. The successful integration of neural networks and hand-tracking technologies not

only demonstrates the technical feasibility of such innovations but also opens new avenues for intuitive and accessible interfaces. The project extends its impact beyond technology, influencing fields like education, healthcare, and entertainment.