CytoAutoCluster: Enhancing Cytometry with Deep Learning

Project by: Abhay Raj Singh Internship at: Infosys Springboard Start Date: October 5th, 2024

Table of Contents

Introduction

The project titled CytoAutoCluster: Enhancing Cytometry with Deep Learning aims to implement a machine learning model that employs semi-supervised learning for deep clustering on cytometry data. The overarching goal is to improve the accuracy and interpretability of cell population identification. Cytometry data, specifically mass cytometry (CyTOF) data, are highly multidimensional, posing challenges for traditional clustering techniques. This project, therefore, aims to leverage deep learning methods to extract meaningful patterns, addressing the imbalanced nature of the data, and ultimately enhancing the classification of unlabelled cell populations.
Dataset Description

After exploring several resources, including Google Scholar and Kaggle, the dataset selected for this project is Levin 32, a well-known 32-dimensional mass cytometry (CyTOF) dataset. The dataset comprises the following characteristics:

- Cells: 265,627 individual cells
- Protein Markers (Dimensions): 32 protein markers
- Manually Gated Cell Populations (Clusters): 14 unique cell populations
- Cluster Labels: Available for 39% (104,184 cells) of the dataset
- Individuals Represented: 2 distinct individuals

Initial Observations:
Based on initial exploration, the dataset exhibited several key characteristics typical of high-dimensional cytometry data:
• High Dimensionality: 32 dimensions create opportunities for deep feature extraction but pose challenges for computational efficiency.
• Imbalanced Label Distribution: Only 39% of the data has manual labels, making this dataset a suitable candidate for semi-supervised learning.
• Variation Across Populations: The cluster distribution varies significantly across the 14 cell populations, with some populations underrepresented.

Data Cleaning:
Data cleaning is an essential step to ensure accurate model performance. The following steps were taken during the cleaning process:

· Handling Null Values: The dataset contains both labeled and unlabeled data. We identified null values in some columns, which represented unlabeled data. A key challenge was visualizing these null values accurately due to overlapping visualizations.
· Action Taken: We corrected this by creating non-overlapping histograms that accurately represented null and non-null values.
· Removing Unnecessary Columns: We identified and removed several columns that were redundant or irrelevant to the clustering task, including those without labels such as "Viability", "file_number", and "event_number".
Challenges Faced:

• Visualization Issues: While plotting null/non-null value graphs, there was significant overlap in the values, initially making it difficult to gain insights.
• Empty Columns: Some columns were found to have no names and caused errors during processing. This was resolved by explicitly removing these unnamed columns.

Exploratory Data Analysis (EDA)

EDA helped in understanding the underlying patterns and structure of the data. Several statistical techniques were applied, each offering unique insights:

Techniques Used:
• Histograms: These were used to visualize the distribution of values in each feature. The histograms were particularly useful for highlighting imbalances in cell populations and variations in feature values.
• Pair Plots: Pairwise relationships between features were visualized to understand potential correlations and interactions among protein markers.
• Skewness & Kurtosis: The distribution of each feature was assessed through skewness and kurtosis metrics. These statistics provided insights into the asymmetry and peakedness of the feature distributions, which are crucial when selecting appropriate algorithms for machine learning.
Visualizations Created:
• Null and Non-null Value Distribution: Histograms were created to display the null and non-null values across the dataset. These graphs clearly distinguished between the labeled and unlabeled parts of the dataset.
• Skewness and Kurtosis Graphs: Individual plots for features like 'DNA1', 'DNA2', 'CD45RA', and others were generated to represent the positive and negative skewness of the data, as well as the kurtosis.
Interpretations:
• Skewness: Features like 'DNA1' exhibited positive skewness, indicating that a significant number of observations had lower values than the median.
• Kurtosis: The kurtosis analysis revealed that certain features, such as 'CD45RA', exhibited a high kurtosis, suggesting a presence of outliers.
Key Findings

From the analysis, several key findings were made:
• High Dimensionality Complexity: The dataset's high dimensionality necessitates deep learning approaches to capture interactions across multiple dimensions effectively.
• Imbalance in Label Distribution: The large proportion of unlabeled data (approximately 61%) aligns with the goal of applying semi-supervised learning, allowing the model to leverage both labeled and unlabeled data.
• Feature Interactions: The pair plots indicated potential correlations between certain features, particularly those involved in immune response, such as 'CD11b' and 'CD45RA'.
Conclusion

The CytoAutoCluster project represents a promising application of deep learning to high-dimensional cytometry data. The use of semi-supervised learning allows us to maximize the utility of both labeled and unlabeled data. Key challenges, such as imbalanced data distribution and high dimensionality, were addressed through thoughtful data preprocessing and exploratory analysis. Future work will focus on refining the clustering algorithm and further improving model accuracy by incorporating more sophisticated feature selection techniques.

Dimensionality Reduction Techniques: PCA and t-SNE

Purpose: PCA and t-SNE were employed for reducing data dimensions, allowing better visualization of clusters while preserving key patterns. PCA captures linear variance, while t-SNE provides a nonlinear view, clustering similar data points.
Challenges: Selecting optimal components for PCA and tuning t-SNE's perplexity and iterations were complex, as misconfiguration could distort cluster formations.
Outcomes: PCA highlighted primary variance directions, and t-SNE visualized distinct clusters, setting the foundation for further dimensionality reduction with Autoencoders.

Autoencoders for Feature Extraction

Purpose: Autoencoders compressed the high-dimensional data into latent features, improving representation for clustering. The aim was to balance efficient encoding with minimal information loss.
Challenges: Hyperparameter tuning to prevent overfitting and deciding the bottleneck layer size were key challenges.
Outcomes: Autoencoders successfully captured core features, enhancing data representation and paving the way for semi-supervised learning on imbalanced data.

Semi-Supervised Learning (SSL)

Purpose: SSL was applied to leverage both labeled and unlabeled data, utilizing techniques like entropy minimization and consistency regularization to maximize the model's learning potential.
Challenges: Implementing regularization methods to generalize the model effectively was challenging, requiring extensive experimentation with SSL parameters.
Outcomes: SSL enhanced model performance by effectively learning from unlabeled data, supporting data corruption techniques in the next phase.

Data Corruption Techniques

Purpose: Binary masking and data shuffling introduced controlled noise, encouraging model robustness by simulating data variability.
Challenges: Balancing data masking levels to obscure features without distorting patterns was essential.
Outcomes: Data corruption techniques improved the model's adaptability, enabling it to identify key patterns in noisy or partial data. This prepared the data for initial model testing.

Model Development and Evaluation

Logistic Regression and XGBoost: These models provided a baseline (Logistic Regression) and high performance (XGBoost) with encoded data. XGBoost's ability to handle complex structures complemented Logistic Regression's simplicity.
Challenges: Overfitting prevention through feature scaling and loss value adjustments were necessary to optimize performance.
Outcomes: Both models leveraged encoded features effectively. XGBoost, in particular, showed strong accuracy on complex data, leading to Encoder model exploration.

Encoder Model Development

Purpose: The Encoder model refined latent features, generating high-quality representations for downstream training.
Challenges: Dimension balancing and training speed optimization required careful

adjustments.
Outcomes: Encoder-generated features improved model accuracy when used in supervised classifiers, building the foundation for further training on latent data.

Timeline
October 7, 2024
Agenda:
- Introduction to the CytoAutoCluster project and its objectives.
- Initial discussion of the dataset's characteristics.

Key Discussions:
- Understanding project goals and dataset complexity.

Action Items:
- Search for relevant datasets to use in the project.

October 8, 2024
Agenda:
- Dataset discussion and Python environment setup.

Key Discussions:
- Analysis of datasets available.
- Setting up GitHub and Python environments for collaborative work.

Action Items:
- Finalize dataset selection and ensure coding environment is properly configured.

October 9, 2024
Agenda:
- Finalization of the dataset.
- Explanation of key dataset features.

Key Discussions:
- Final selection of the Levin 32 dataset.
- Introduction to dataset dimensions and features.

October 10, 2024
Agenda:
- Data Exploration.

Key Discussions:
- Review of tools such as Pandas and NumPy for data exploration.

Action Items:
- Perform preliminary EDA on the dataset.

October 11, 2024
Agenda:
- Complete Data Exploration.

Key Discussions:
- Implementation of EDA techniques.
- Correcting output visualizations for clarity.

October 14-16, 2024
Agenda:
- Finalize EDA and correct visualizations.
- Read research papers relevant to deep clustering and semi-supervised learning for cytometry.

October 17, 2024
Agenda: Complete Data Exploration, implement PCA and t-SNE
Action Items: Gain deeper understanding of PCA and t-SNE

October 18, 2024
Agenda: Review PCA and t-SNE outputs
Action Items: Finalize t-SNE and compare with other results
October 22, 2024
Agenda: Begin work on Autoencoders
Action Items: Basic understanding of Autoencoders
October 23, 2024
Agenda: Identify Autoencoder models for tabular data
Action Items: Understand Autoencoders for feature extraction
October 25, 2024
Agenda: Learn Semi-Supervised Learning basics
Action Items: Study SSL concepts
October 28, 2024
Agenda: Discuss entropy minimization and consistency regularization
Action Items: Implement SSL techniques
October 29, 2024
Agenda: Review binary mask implementation
Action Items: Evaluate binary mask outputs
October 30, 2024
Agenda: Data corruption methods
Action Items: Evaluate corrupted data for training
November 1, 2024
Agenda: Data splitting and train-test setup
Action Items: Complete model preparation
November 5, 2024
Agenda: Implement Logistic Regression
Action Items: Finalize model parameters
November 7, 2024
Agenda: Develop Encoder model
Action Items: Initiate latent feature training
November 8, 2024
Agenda: Train on Logistic Regression and XGBoost
Action Items: Troubleshoot Encoder-based training
November 12, 2024
Agenda: Begin Semi-Supervised Learning model
Action Items: Implement SSL on encoded data