CS598 Deeplearning For HealthCare Group Project

Reproduction of DeepMicro Study

Deep Representation Learning for Disease Prediction based on Microbiome Data

Team Members

- **Joaquin Ugarte (jugarte2@illinois.edu)**
- **Prasad Gole (gole2@illinois.edu)**
- **Ehit Agarwal (ehitda2@illinois.edu)**

Project Repo: https://github.com/Prasad-py/DLH_Project

Original Project: https://github.com/minoh0201/DeepMicro

Project Scope and Hypotheses

Scope: Replicate the DeepMicro study and test its hypotheses

Hypothesis 1:

Training classifiers using a lower dimensional representation will result in more accurate predictions

Hypothesis 2:

Training classifiers using a lower dimensional representation will speed up the model training and hyperparameter tuning process

ORIGINAL DEEPMICRO STUDY

- DeepMicro is a deep representation learning framework for disease prediction using microbiome data

- Transforms high-dimensional microbiome data into a robust low-dimensional representation using autoencoders

- Applies machine learning classification on the learned representation

- Outperforms current best approaches in five out of six datasets

Methodology Overview

Data Preprocessing:

- Obtained publicly available human gut metagenomic samples from six disease cohorts
- Preprocessed abundance and marker datasets, extracting features and labels
- Converted data into image format for Convolutional AutoEncoder (CAE)
- Split data into train (80%) and test (20%) sets

AutoEncoder Models:

- Implemented four types of autoencoders: Shallow AutoEncoder (SAE), Deep AutoEncoder (DAE), Variational AutoEncoder (VAE), and Convolutional AutoEncoder (CAE)
- Trained autoencoders with various hyperparameter combinations to learn low-dimensional representations
- Evaluated autoencoders using Mean Squared Error (MSE) loss on the test set
- Selected the best model for each autoencoder type based on the lowest MSE loss

Methodology

Classifier Models:

- Trained three classifiers: Multi-layer Perceptron (MLP), Support Vector Machine (SVM), and Random Forest (RF)
- Trained classifiers on both the original dataset and the encoded representations from the best autoencoders
- Evaluated classifiers using Area Under the ROC Curve (AUC) with 5-fold cross-validation
- All train/validation/test splits were stratified to maintain class balance

Implementation Details:

- Models implemented using PyTorch
- Utilized early stopping with checkpoint saving to prevent overfitting and ensure reasonable training times
- Initialized neural network weights using the Glorot Uniform initializer
- Conducted experiments on a machine with GTX 4080 16 GB GPU, AMD 5800X CPU, and 64 GB of RAM
- Total training time was about 20 hours

Discussion

Possible explanations for results:

- Error in the training process of autoencoders (unlikely, as training loss graphs have expected shape)

- Differences between PyTorch and Keras/TensorFlow implementations

- Need for better tuning of training hyperparameters

Future plans:

- Implement missing models and proper hyperparameter search

- Add new datasets from recent studies (phylaGAN and MV-CVIB)

Results Summary

1. Classifiers trained on original data consistently performed better than those trained on encoded representations
2. This result is different from Hypothesis 1 and the DeepMicro paper's findings
3. Classifiers trained faster on the learned representation, supporting Hypothesis 2
4. However, total time taken for training both the encoder and classifier is higher than just training the classifier

Conclusion

1. Successfully replicated the DeepMicro study using PyTorch
2. Results did not support Hypothesis 1, but supported Hypothesis 2
3. Identified areas for improvement and future work
4. Gained valuable experience in implementing deep learning models for microbiome data analysis

Challenges and Learnings

Challenges Faced:

- Hyperparameter tuning complexities.
- Differences in framework-specific behaviors affecting model performance.

Key Learnings:

- Importance of rigorous validation schemes.
- Insights into handling high-dimensional biological data.

References

1. Cho, I. & Blaser, M. J. The human microbiome: at the interface of health and disease. Nature Reviews Genetics 13, 260 (2012).

2. Eloe-Fadrosh, E. A. & Rasko, D. A. The human microbiome: from symbiosis to pathogenesis. Annual review of medicine 64, 145-163 (2013).

3. Hamady, M. & Knight, R. Microbial community profiling for human microbiome projects: tools, techniques, and challenges. Genome research 19, 1141-1152 (2009).

4. Scholz, M. et al. Strain-level microbial epidemiology and population genomics from shotgun metagenomics. Nature methods 13, 435 (2016).

5. Divya Sharma, Wendy Lou, Wei Xu, phylaGAN: Data augmentation through conditional GANs and autoencoders for improving disease prediction accuracy using microbiome data, Bioinformatics, 2024;, btae161, https://doi.org/10.1093/bioinformatics/btae161

6. Cui Z, Wu Y, Zhang Q-H, Wang S-G, He Y and Huang D-S (2023) MV-CVIB: a microbiome-based multi-view convolutional variational information bottleneck for predicting metastatic colorectal cancer. Front. Microbiol. 14:1238199. doi: 10.3389/fmicb.2023.1238199

7. U. Gülfem Elgün Çiftcioğlu, O. Ufuk Nalbanoglu, DeepGum: Deep feature transfer for gut microbiome analysis using bottleneck models, Biomedical Signal Processing and Control, Volume 91, 2024, 105984, ISSN 1746-8094, https://doi.org/10.1016/j.bspc.2024.105984.

8. Oh, Min, and Liqing Zhang. "DeepMicro: deep representation learning for disease prediction based on microbiome data." *Scientific Reports* 10.1 (2020): 1-9. https://doi.org/10.1038/s41598-020-63159-5.

9. Pasolli, E., Truong, D. T., Malik, F., Waldron, L. & Segata, N. Machine learning meta-analysis of large metagenomic datasets: tools and biological insights. PLoS computational biology 12, e1004977 (2016).

10. https://github.com/minoh0201/DeepMicro

Thank You

- **Joaquin Ugarte (jugarte2@illinois.edu)**
- **Prasad Gole (gole2@illinois.edu)**
- **Ehit Agarwal (ehitda2@illinois.edu)**

Project Repo: https://github.com/Prasad-py/DLH_Project

Original DeepMicro Study:

https://github.com/minoh0201/DeepMicro