Health Monitoring and Disease Prediction using Deep learning Model

Group Member Information

- Sahithi Thulluru(700752037)
- Sahil Naidu Pagadala(700748694)
- Pavan Uppala(700754009)
- Priyanka Nayudu(700756568)

Role/Responsibilities and Contribution in project

Description/Task	Responsibility - Person
Data Read and Preprocessing	Sahithi Thulluru & Sahil Naidu Pagadala
Model Selection (simple feedforward Neural Network)	Sahil Naidu Pagadala & Pavan Uppala
Implementation(Data Splitting, Neural Network Training & Evaluation, Hyper parameter Tuning, K-fold cross validation, Decision Tree)	Sahithi Thulluru, Sahil Naidu Pagadala, Pavan Uppala & Priyanka Nayudu
Result Analysis	Priyanka Nayudu & Pavan Uppala

Motivation

- Improving Healthcare Decision-making: Enhancing medical facilities to ensure better patient diagnosis and treatment decisions.
- Utilizing Deep Learning: Harnessing the power of deep learning to process large and complex medical information.
- Early Disease Prediction: Implementing disease prediction for early prevention, aligning with the adage "Prevention is better than cure."

Objectives

- Utilize deep learning to analyze both structured and unstructured healthcare data, aiming to accurately assess disease risk with unprecedented accuracy and depth.
- Implement latent factor model approach within neural network architecture to address challenges posed by missing data in structured healthcare records, enhancing completeness and reliability of analyses.
- Employ deep learning techniques to automatically identify features from unstructured text data, enriching analyses and providing a more comprehensive understanding of underlying health dynamics.

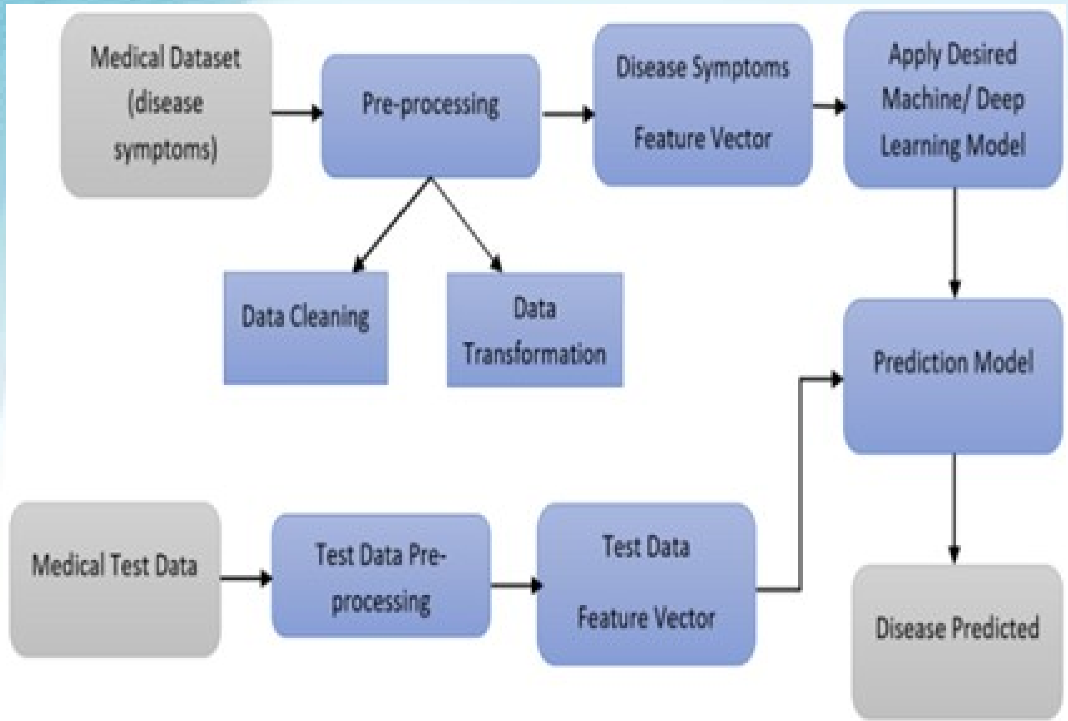
Related work

- Disease Phenol Type Similarity: Exploring miRNA prediction for human diseases using network-based techniques.
- Extreme Learning Machine for Multilayer Perceptron: The paper provides empirical evidence of the effectiveness of ELMs for training MLPs across various datasets, highlighting their potential as a powerful tool for machine learning tasks.
- Heart Disease Prediction: Utilizing Naive Bayesian classification and data mining tools for accurate risk factor prediction in heart disease.

Problem Statement

- Traditional healthcare predictive analytics methods struggle to accurately forecast diseases from patient treatment histories and health data, prompting a need for more advanced approaches.
- Deep learning neural networks offer a revolutionary solution to this challenge, leveraging their ability to learn complex patterns directly from raw medical data.
- Integrating deep learning into healthcare aims to enhance patient care by enabling earlier disease detection, personalized treatment approaches, and improved treatment efficacy, ultimately leading to better outcomes.

Proposed Solution



- Adoption of latent factor model within neural network architecture to address missing data.
- Utilization of deep learning techniques, such as the neural network (simple feedforward Neural Network) algorithm, for feature extraction from text data.
-

Neural Network Implementation

Data Preprocessing:

Split dataset into features (x) and target variable (y).

Normalize features to ensure consistent scaling.

Neural Network Definition:

Class **NeuralNetwork** inheriting from **nn.Module**.

Architecture: Two fully connected layers (fc1 and fc2) with ReLU activation and softmax output.

Dropout regularization to prevent overfitting.

Hyperparameter Tuning and Cross-Validation:

Define hyperparameters like learning rates and hidden layer sizes.

Perform K-Fold cross-validation to assess model performance.

Training loop iterates over different hyperparameter combinations and evaluates on validation data.

Training:

Train neural network model within each fold using Adam optimizer and cross-entropy loss.

Evaluation:

Select best model based on cross-validation accuracy.

Evaluate on full training set by comparing predicted labels with ground truth labels.

Testing:

Define test function to make predictions on new data (symptoms).

Convert user-provided symptoms into a binary input vector and pass through the trained model.

Result Analysis

- Accurate model evaluation requires careful separation of data into training and testing sets, with a common practice of allocating 67% for training and 33% for testing. This ensures that the model's performance is assessed on unseen data, crucial for gauging its real-world effectiveness.
- Monitoring the distribution of samples and labels in both the training and testing sets provides valuable insights into the representativeness of the data partitions. Maintaining a balanced distribution is essential for robust model assessment and generalization, facilitating accurate predictions on new, unseen data.
- PyTorch facilitates the creation of a robust neural network model for categorization tasks, with meticulous data preparation including label encoding and feature normalization.
- Leveraging two fully connected layers with ReLU activation and a softmax output layer, coupled with Adam optimizer and cross-entropy loss, the model achieves a commendable 90% accuracy on the training set, showcasing its efficacy in predicting target variables accurately.

```
Count of samples in X_train: 3296
Count of samples in X_test: 1624
Count of labels in y_train: 3296
Count of labels in y_test: 1624
```

Result

Data Sample

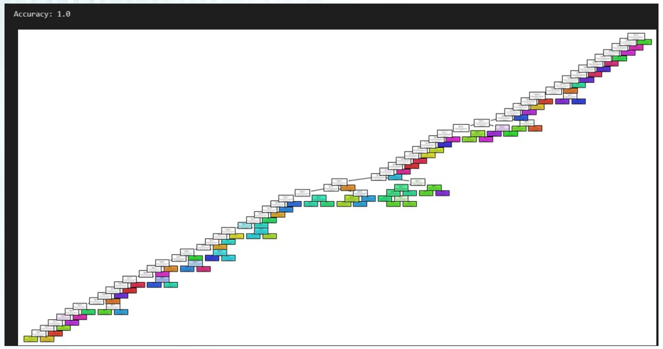
Sample Count

```
Count of samples in X_train: 3296
Count of samples in X_test: 1624
Count of labels in y_train: 3296
Count of labels in y_test: 1624
```

Simple feedforward Neural Network

```
Epoch [10/100], Loss: 3.6980
Epoch [20/100], Loss: 3.6542
Epoch [30/100], Loss: 3.5511
Epoch [40/100], Loss: 3.3771
Epoch [50/100], Loss: 3.1831
Epoch [60/100], Loss: 3.0495
Epoch [70/100], Loss: 2.9716
Epoch [80/100], Loss: 2.9258
Epoch [90/100], Loss: 2.8911
Epoch [100/100], Loss: 2.8688
Accuracy on train set: 0.90
```

Decision Tree (Reference for comparison)



Hyper parameter tuning - K-fold

```
Training with learning rate: 0.001, hidden size: 64
Avg. Accuracy for LR=0.001, Hidden Size=64: 0.27134146341463417
Training with learning rate: 0.001, hidden size: 128
Avg. Accuracy for LR=0.001, Hidden Size=128: 0.36097560975609755
Training with learning rate: 0.01, hidden size: 64
Early stopping in fold 1, epoch 73
Early stopping in fold 2, epoch 87
Early stopping in fold 3, epoch 71
Early stopping in fold 4, epoch 83
Avg. Accuracy for LR=0.01, Hidden Size=64: 0.49430894308943085
Training with learning rate: 0.01, hidden size: 128
Early stopping in fold 1, epoch 80
Early stopping in fold 2, epoch 63
Early stopping in fold 3, epoch 66
Early stopping in fold 4, epoch 53
Early stopping in fold 5, epoch 55
Avg. Accuracy for LR=0.01, Hidden Size=128: 0.5371951219512195
Accuracy on train set: 0.51
```

With Input Symptoms:

```
# Example usage of the check function
example_symptoms = ['headache', 'muscle_weakness', 'puffy_face_and_eyes',
prediction = check(example_symptoms)
print("Prediction:", prediction)
```

```
[[0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
Prediction: Chicken pox
```

References

- Kwon, K., Kim, D. & Park, H. A parallel MR imaging method using multilayer perceptron. Med. Phys. 44, 6209–6224 (2017).
- 2015 2nd National Foundation for Science and Technology Development Conference on Information and Computer Science" Duc-Hau Le, "Disease phenotypic similarity improves the prediction of novel disease associated microRNAs
- Tang, J., Deng, C. & Huang, G. B. Extreme learning machine for multilayer perceptron. IEEE Trans. Neural Netw. Learn. Syst. 27, 809–821 (2016).
- Anjan Nikhil Repaka, "Design and Implementing Heart Disease Prediction Using Naives Bayesian," in Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019); IEEE Xplore Part Number: CFP19J32-ART; ISBN: 978-1-5386-9439-8