Deep Learning–Based Classification of Breast Cancer Using the Wisconsin Diagnostic Dataset

Kazi Abdul Jamil
University of Pavia

Abstract

This proposal outlines a plan to apply deep learning techniques to the Breast Cancer Wisconsin (Diagnostic) dataset. We will develop, train, and evaluate multiple neural network architectures for binary classification of malignant versus benign tumors based on the 30 real-valued features extracted from fine needle aspirate (FNA) images. Our goals include achieving state-of-the-art classification performance, exploring feature representation via autoencoders, and incorporating explainability methods to interpret model decisions.

1 Introduction

Breast cancer remains a leading cause of mortality among women worldwide. Early and accurate diagnosis greatly improves patient outcomes. The Breast Cancer Wisconsin (Diagnostic) dataset, comprising 569 instances with 30 features computed from digitized FNA images, provides a benchmark for machine learning–based diagnostic tools. Traditional methods (e.g., logistic regression, decision trees, support vector machines) have achieved high accuracy; however, deep learning offers potential advantages in automatic feature learning, robustness, and adaptability. This project will investigate deep neural network models tailored to tabular medical data and compare their performance with standard approaches.

2 Objectives

1. **Data Exploration and Preprocessing:**

- Perform descriptive statistics and visualizations to understand feature distributions and correlations.
- Handle missing values, if any, and apply normalization or standardization.

2. **Model Development:**
    - Design and implement multiple architectures:
        - Fully Connected Neural Networks (FCNNs) with varying depth and width.
        - Autoencoder-based feature extraction followed by classification head.
        - Hybrid models incorporating attention mechanisms or embedding layers.
    - Incorporate regularization techniques (dropout, weight decay) and batch normalization.

3. **Training and Optimization:**
    - Split data into training, validation, and test sets using stratified sampling.
    - Employ cross-validation to ensure robustness.
    - Use advanced optimizers (Adam, RMSprop) and learning rate schedules.
    - Conduct hyperparameter tuning via grid search or Bayesian optimization.

4. **Model Evaluation and Comparison:**
    - Evaluate models on key metrics: accuracy, precision, recall, F1-score, ROC-AUC.
    - Compare deep models against baseline classical algorithms (e.g., SVM, Random Forest).

5. **Explainability and Visualization:**
    - Apply model-agnostic methods such as SHAP and LIME to interpret feature importance.
    - Visualize learned representations via t-SNE or PCA on internal layer activations.

3 Data Description

The dataset comprises 569 samples (357 benign, 212 malignant) with 30 continuous features derived from FNA images of breast masses. Features include radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension (mean, standard error, and worst values for each).

4 Methodology

1. **Environment Setup:** Python, TensorFlow/Keras and PyTorch libraries.
2. **Preprocessing Pipeline:**
   - Impute or drop missing values.
   - Scale features to zero mean and unit variance.
   - Encode labels (benign = 0, malignant = 1).
3. **Model Implementation:**
   - Build FCNNs: experiment with 2–5 hidden layers, 32–256 units per layer.
   - Design an autoencoder to compress 30-dimensional input into a lower-dimensional latent space (e.g., 10 dimensions), then append a classifier.
   - Explore self-attention layers to weight features dynamically.
4. **Training Strategy:**
   - Use early stopping on validation loss.
   - Monitor overfitting; apply dropout (0.2–0.5).
   - Fine-tune learning rate and batch size (e.g., 16, 32, 64).
5. **Evaluation:**
   - Report confusion matrix, ROC curve, and precision-recall curve.
   - Perform statistical comparison (e.g., McNemar's test) between best deep model and baseline.

5 Timeline (10-Day Plan)

Day 1	Literature review; data acquisition; environment setup (install Python, TensorFlow/PyTorch).
Day 2	Exploratory data analysis: descriptive stats, visualizations, missing-value checks, initial preprocessing.
Day 3	Baseline classical models: implement and evaluate Logistic Regression, SVM, Random Forest.
Day 4	Design and implement initial FCNN architecture; begin training on stratified split.
Day 5	Refine FCNN: apply regularization (dropout, weight decay), batch normalization; tune learning rate.
Day 6	Develop autoencoder for feature extraction; pretrain encoder, inspect latent space.
Day 7	Append classification head to autoencoder; train end-to-end and compare with FCNN.
Day 8	Implement explainability: run SHAP/LIME on best models; visualize feature importance.
Day 9	Final hyperparameter sweep (batch size, optimizer choice, depth/width); generate ROC/PR curves.
Day 10	Consolidate results: confusion matrices, statistical tests; finalize report and prepare presentation.

6 Expected Outcomes

- A comparative analysis demonstrating whether deep learning architectures can surpass traditional models on the Wisconsin dataset.
- Insights into latent feature representations via autoencoders.
- An interpretable model with clear justifications for feature importance, aiding clinical transparency.
- A complete report and code repository suitable for academic submission.

7 References

1. Street, W. N., Wolberg, W. H., Mangasarian, O. L. (1993). Nuclear Feature Extraction for Breast Tumor Diagnosis. *IST/SPIE Conference on Digital Mammography.*

2. Chollet, F. (2017). *Deep Learning with Python.* Manning Publications.

3. Lundberg, S. M., Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, 30.