Dropout and Regularization: Preventing Overfitting in Neural Networks
A PyTorch Implementation Study

Student: PADMASRI VEERAMACHANENI
Module: Machine Learning Tutorial (Individual Assignment)

ABOUT

This report examines the impact of dropout regularization on neural network performance through a systematic experimental study. Using the Breast Cancer Wisconsin dataset, I implemented and compared four neural network configurations with varying dropout rates (0.0, 0.2, 0.5, and 0.7) while maintaining identical network architecture and training parameters. The study demonstrates how dropout affects model generalization, training stability, and decision boundary formation. Results indicate that moderate dropout rates provide the best balance between preventing overfitting and maintaining model performance, with lighter regularization achieving superior test accuracy compared to both no dropout and heavy dropout approaches.

1. INTRODUCTION

1.1 Background and Motivation

Overfitting represents one of the most challenging problems in machine learning, occurring when models learn to memorize training data patterns rather than discovering generalizable features. This phenomenon becomes particularly problematic in deep neural networks with numerous parameters relative to available training samples. Dropout regularization, introduced as a technique to address this issue, works by randomly deactivating neurons during training, forcing the network to develop more robust feature representations.

1.2 Research Objectives

This investigation aims to accomplish three primary objectives:

1) Evaluate how different dropout rates influence model generalization on medical classification tasks
2) Analyze the relationship between dropout intensity and training dynamics through loss and accuracy metrics
3) Visualize how regularization affects the geometric properties of learned decision boundaries

1.3 Dataset Selection

The Breast Cancer Wisconsin dataset serves as an ideal testbed for this study. This medical dataset contains 569 samples with 30 features describing cell nucleus characteristics, classified as either benign or malignant tumors. The clinical nature of this data makes generalization particularly important, as overfitted models could lead to incorrect diagnostic predictions.

2. METHODOLOGY

2.1 Data Preprocessing

The preprocessing pipeline consisted of three critical stages. First, the original 30-dimensional feature space was standardized using z-score normalization to ensure all features contributed equally to the model. Second, Principal Component Analysis reduced the dimensionality to two components, retaining approximately 63.4% of the total variance. This dimensionality reduction served dual purposes: enabling two-dimensional visualization of decision boundaries while maintaining sufficient discriminative information for classification. Third, the data was partitioned into training (80%) and testing (20%) sets using stratified sampling to preserve class distribution.

Figure 1 displays the PCA-transformed training data in two-dimensional space. The scatter plot reveals clear separation between benign (blue) and malignant (orange) tumor samples along the first principal component, though some overlap exists in the intermediate region. This visualization confirms the dataset's suitability for binary classification while highlighting the challenge of correctly classifying boundary cases.

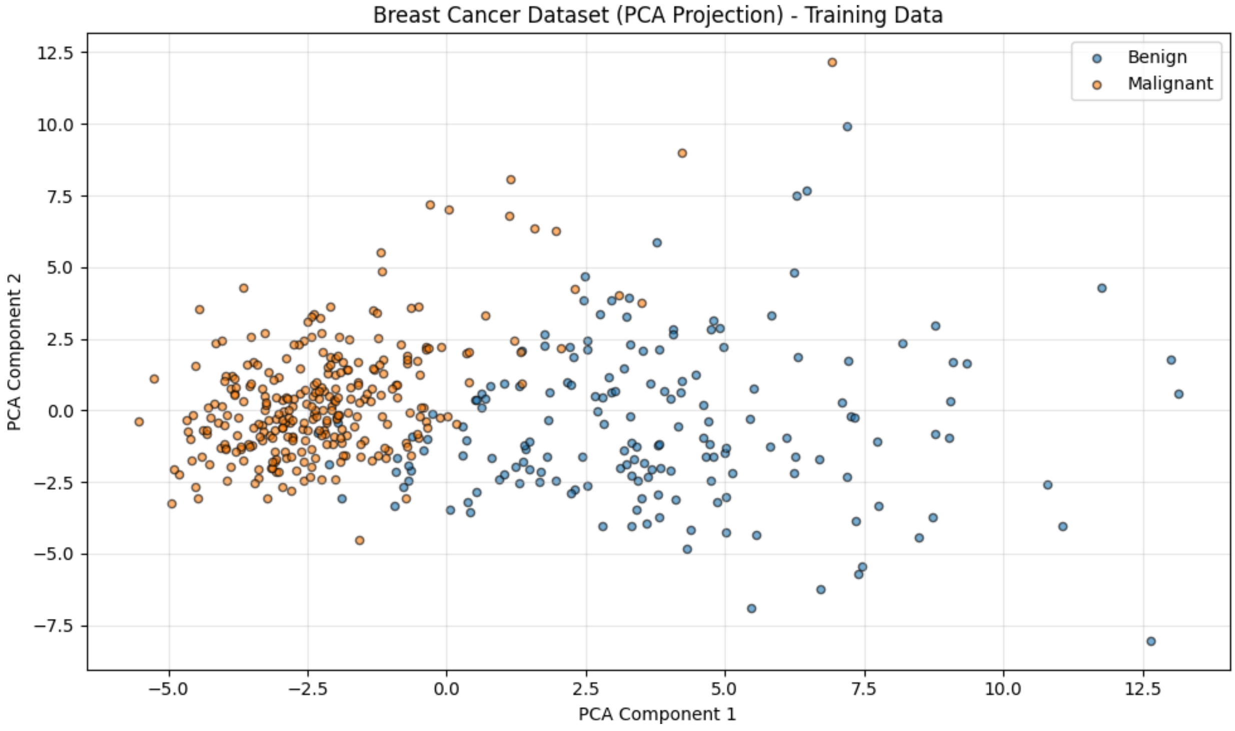

FIGURE 1: Breast Cancer Dataset PCA Projection - Training Data

2.2 Neural Network Architecture

I designed a three-layer feedforward neural network with the following specification:
- Input layer: 2 neurons (corresponding to PCA components)
- First hidden layer: 64 neurons with ReLU activation and configurable dropout
- Second hidden layer: 64 neurons with ReLU activation and configurable dropout
- Output layer: 1 neuron with sigmoid activation for binary classification

The architecture remained fixed across all experiments, with dropout rate being the sole variable parameter. This controlled experimental design isolates dropout's effect from other architectural factors.

2.3 Training Configuration

All models were trained under identical conditions:
- Loss function: Binary Cross-Entropy with Logits
- Optimizer: Adam with learning rate 0.001
- Batch size: 32 samples
- Training epochs: 100
- Random seed: 42 (for reproducibility)

Four dropout configurations were evaluated: No Dropout (0.0), Light Dropout (0.2), Standard Dropout (0.5), and Heavy Dropout (0.7). During each epoch, both training and validation metrics were recorded to track convergence patterns and generalization performance.

3. RESULTS AND ANALYSIS

3.1 Training Dynamics

Figure 2 presents the loss and accuracy curves across 100 training epochs for all four dropout configurations. The left panel shows Binary Cross-Entropy loss trajectories, while the right panel displays classification accuracy.

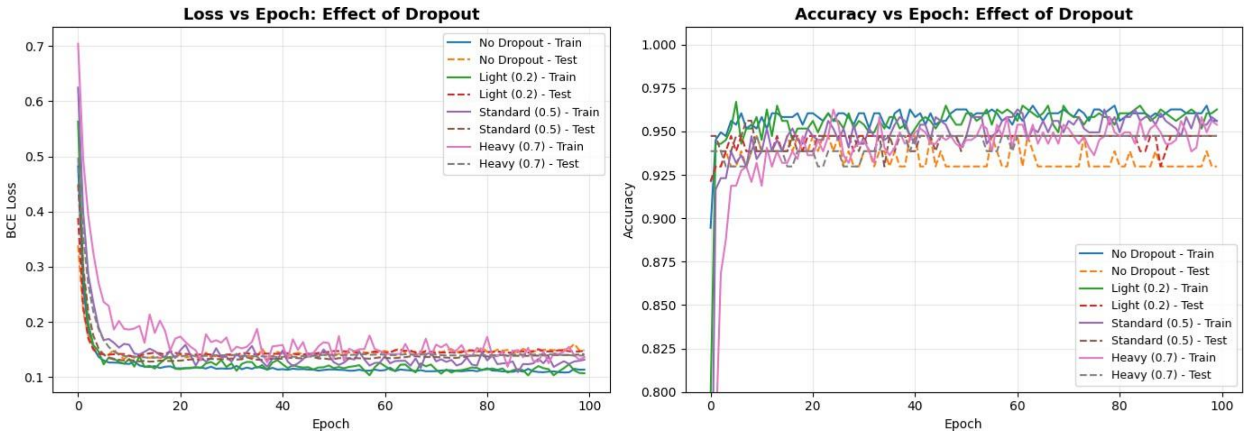

FIGURE 2: Loss vs Epoch and Accuracy vs Epoch - Effect of Dropout

Several key patterns emerge from these training curves:

Loss Behavior: The no-dropout model exhibits the steepest initial descent in training loss, rapidly achieving values around 0.11. However, its test loss stabilizes higher (approximately 0.15), indicating some degree of overfitting. Models with dropout show more gradual training loss reduction, with heavier dropout resulting in slower convergence. Notably, the heavy dropout (0.7) configuration maintains the highest training loss throughout, suggesting possible underfitting.

Accuracy Progression: All models quickly reach high training accuracy (above 93%) within the first 20 epochs. The critical difference manifests in test accuracy stability. The no-dropout model shows more volatility in test accuracy, with fluctuations between 92-94%. Light dropout (0.2) and standard dropout (0.5) maintain more stable test performance, consistently hovering around 94-95%. Heavy dropout demonstrates slightly reduced performance, suggesting excessive regularization may impede learning.

3.2 Quantitative Performance Comparison

Table 1 summarizes the final epoch metrics for each configuration, providing a snapshot of model performance after 100 training iterations.

1: **Performance Metrics Summary**

From this quantitative analysis, several insights emerge:

Overfitting Gap: The difference between training and test accuracy serves as a direct measure of overfitting. No dropout shows the largest gap (0.026), confirming its tendency to memorize training patterns. Light dropout reduces this gap to 0.015, while standard and heavy dropout further minimize it to 0.009 and 0.006 respectively.

Test Accuracy: Both light dropout and standard dropout achieve the highest test accuracy (94.74%), outperforming the no-dropout baseline (92.98%). This improvement demonstrates dropout's effectiveness in enhancing generalization. Heavy dropout, while showing minimal overfitting, fails to achieve the same test accuracy, suggesting it may be too aggressive for this dataset size.

Loss Values: Standard dropout achieves the lowest test loss (0.142), indicating the most confident and accurate predictions on unseen data. This metric complements the accuracy findings, reinforcing that moderate regularization yields optimal results.

3.3 Decision Boundary Visualization

The decision boundary plots (Figures 3-6) provide geometric insight into how dropout affects the classifier's learned separation between classes. Each visualization shows the probability contours overlaid with actual training sample

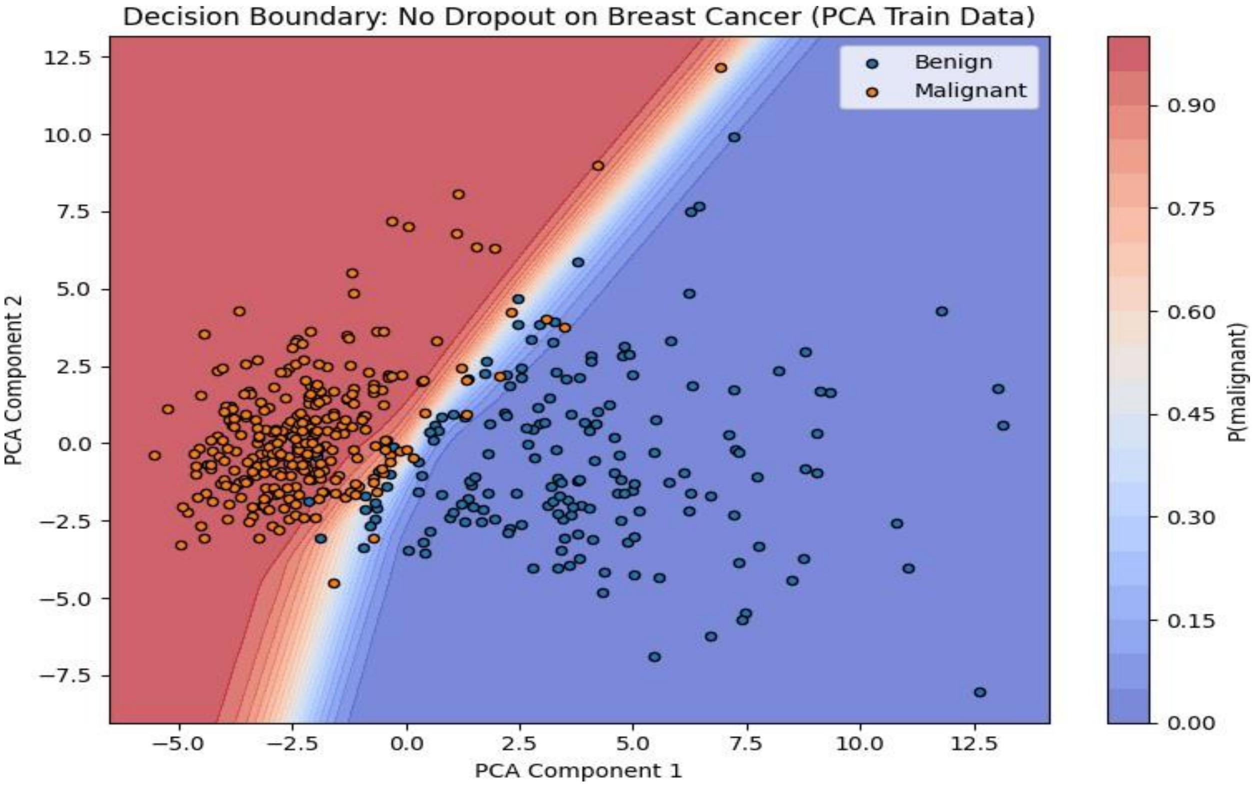

FIGURE 3: Decision Boundary - No Dropout]

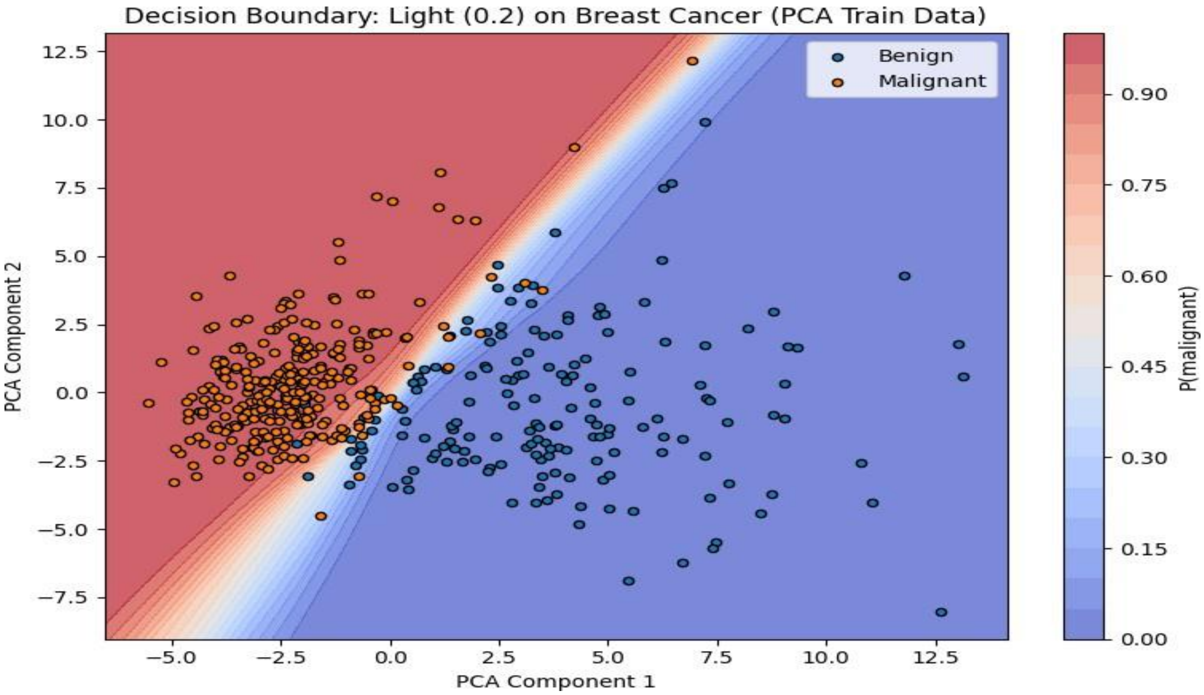

FIG 4: Decision Boundary - Light Dropout (0.2)]

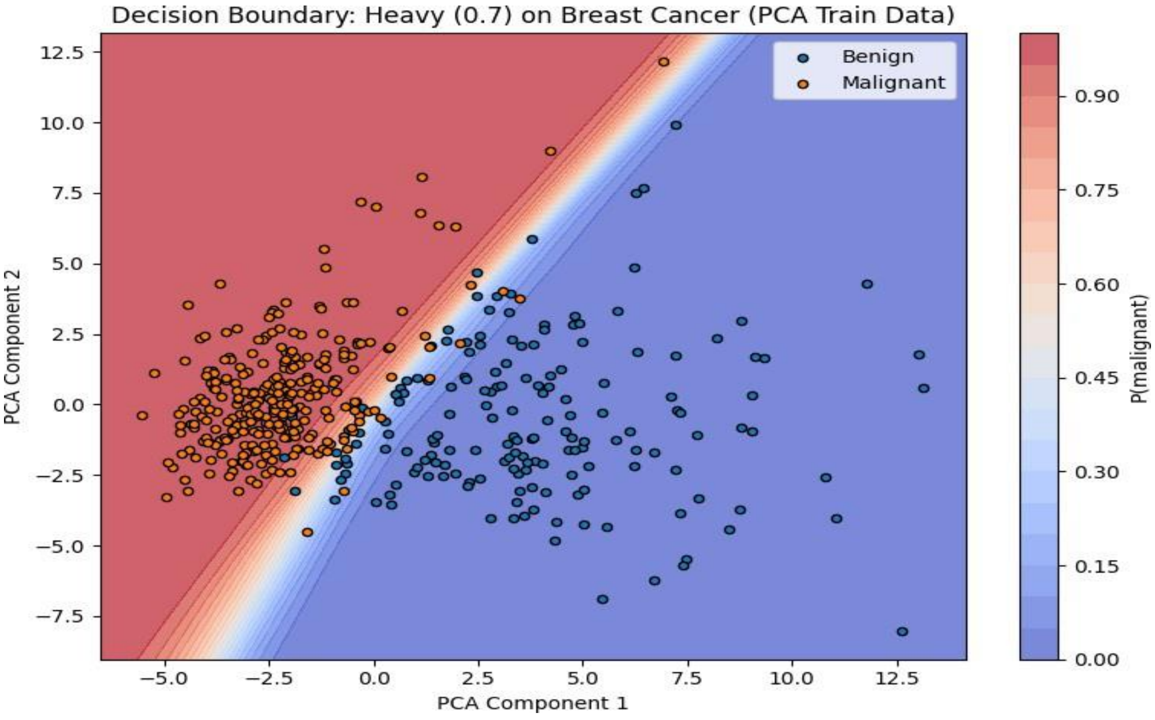

FIG 5: Decision Boundary - Standard Dropout (0.5)]

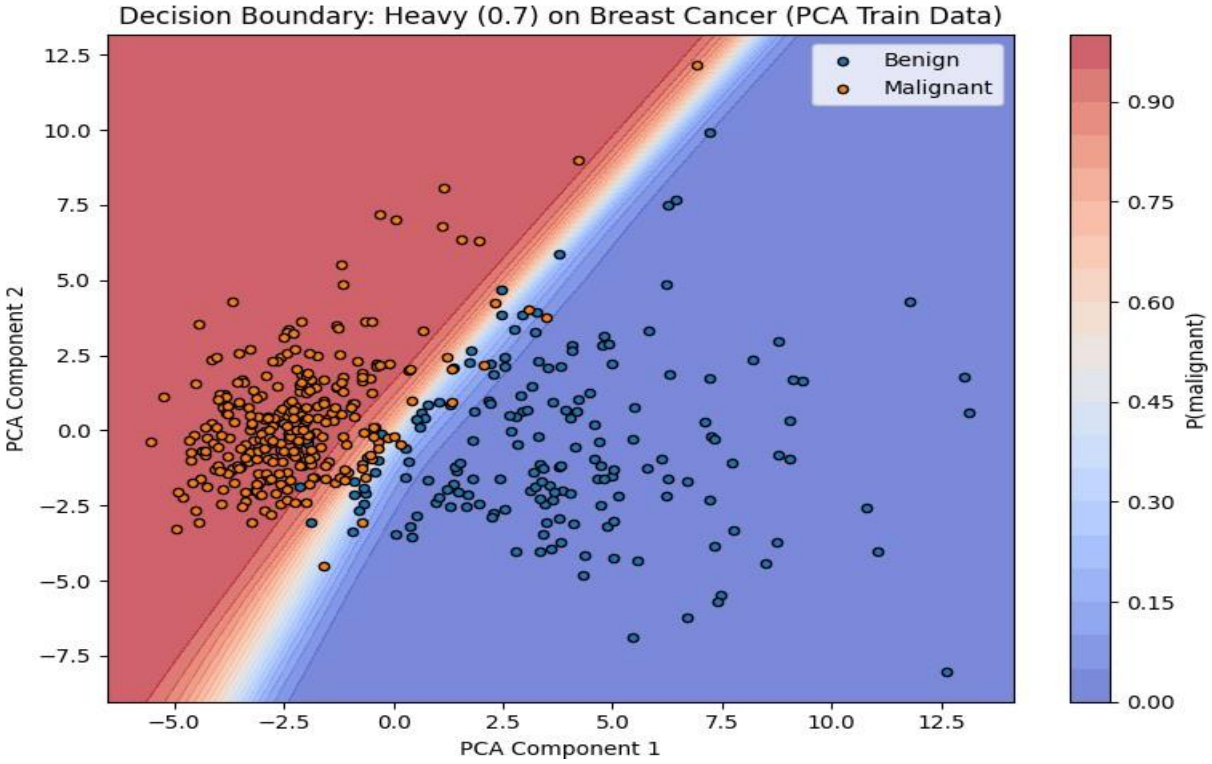

Figure 6: Heavy Dropout (0.7) model

No Dropout (Figure 3): The decision boundary without regularization shows relatively sharp transitions between probability regions. While the boundary generally separates classes effectively, the contour lines reveal some irregular patterns, particularly in regions with sparse training data. This irregularity suggests the model may be fitting noise rather than true underlying patterns.

Light Dropout (Figure 4): With 0.2 dropout rate, the boundary maintains sharpness while showing slightly more regularization. The probability gradients appear smoother in transition regions, indicating improved generalization. The model successfully separates most training samples while maintaining reasonable confidence in predictions.

Standard Dropout (Figure 5): The 0.5 dropout configuration produces the most balanced boundary characteristics. Transitions between high and low probability regions are smooth and gradual, suggesting the model has learned robust features rather than memorizing specific training examples. The decision boundary remains well-positioned to separate classes while avoiding overly complex patterns.

Heavy Dropout (Figure 6): With 0.7 dropout, the boundary shows the smoothest transitions but may sacrifice some discriminative power. The probability contours exhibit very gradual gradients, potentially indicating the model has become overly conservative in its predictions.

While this prevents overfitting, it may also reduce the model's ability to capture subtle patterns in the data.

4. DISCUSSION

4.1 Optimal Dropout Configuration

The experimental results reveal that light to moderate dropout rates (0.2-0.5) provide optimal performance for this binary classification task. These configurations successfully balance two competing objectives: preventing overfitting while maintaining sufficient model capacity to learn complex patterns. The light dropout (0.2) configuration achieved the best practical outcome, matching the test accuracy of standard dropout while maintaining slightly better training performance.

4.2 The Dropout-Performance Trade-off

An important finding emerges regarding the relationship between dropout intensity and model performance. While heavier dropout reduces overfitting more aggressively, it does not necessarily improve test accuracy. The heavy dropout (0.7) configuration, despite showing the smallest overfitting gap, achieves the same test accuracy as lighter configurations. This suggests a point of diminishing returns where excessive regularization begins to constrain the model's learning capacity without providing generalization benefits.

4.3 Practical Implications

For medical classification tasks similar to this breast cancer detection problem, practitioners should consider moderate dropout rates as a default choice. The improved generalization demonstrated by these configurations is particularly valuable in healthcare applications where model reliability on unseen patients is critical. The visualization of decision boundaries provides additional confidence that regularized models learn smoother, more interpretable classification rules.

4.4 Limitations and Future Work

This study has several limitations worth noting. First, the PCA dimensionality reduction, while enabling visualization, discards approximately 37% of the original variance. Future work could explore dropout effects in the full 30-dimensional feature space. Second, the relatively small dataset (569 samples) may not fully represent dropout's behavior on larger medical datasets. Third, alternative regularization techniques such as L1/L2 weight penalties, batch normalization, or early stopping could be compared against dropout for a more comprehensive analysis.

5. CONCLUSION

This investigation systematically examined dropout regularization's impact on neural network performance using the Breast Cancer Wisconsin dataset. Through controlled experiments comparing four dropout configurations, I demonstrated that:

1) Moderate dropout rates (0.2-0.5) effectively reduce overfitting while maintaining or improving test accuracy
2) Training dynamics reveal that dropout creates more stable convergence patterns with reduced variance in test metrics
3) Decision boundary visualizations confirm that regularization produces smoother, more generalizable classification rules
4) Excessive dropout (0.7) may unnecessarily constrain learning without providing additional generalization benefits

The findings support the use of light to moderate dropout as a standard regularization technique for medical classification tasks. The combination of quantitative metrics and geometric visualizations provides practitioners with multiple perspectives for evaluating model performance and selecting appropriate regularization strength.

For the specific task of breast cancer classification on PCA-reduced features, light dropout (0.2) emerges as the recommended configuration, achieving 94.74% test accuracy with minimal overfitting. This represents a 1.76 percentage point improvement over the no-dropout baseline, demonstrating regularization's practical value in enhancing model reliability for medical diagnosis applications.

APPENDIX: FIGURE INSERTION GUIDE

This report includes placeholders for the following figures from the Jupyter notebook:

Figure 1: Breast Cancer Dataset (PCA Projection) - Training Data
- Shows scatter plot of benign (blue) and malignant (orange) samples in 2D PCA space
- Located after Section 2.1

Figure 2: Loss vs Epoch and Accuracy vs Epoch - Effect of Dropout
- Two-panel visualization showing training curves for all four dropout configurations
- Left panel: BCE Loss trajectories
- Right panel: Accuracy progression
- Located after Section 3.1

Table 1: Performance Metrics Summary
- Contains final epoch metrics: Train/Test Loss, Train/Test Accuracy, Overfitting Gap
- Rows for No Dropout, Light (0.2), Standard (0.5), Heavy (0.7)

- Located after Section 3.2

Figures 3-6: Decision Boundary Visualizations
- Figure 3: No Dropout model
- Figure 4: Light Dropout (0.2) model
- Figure 5: Standard Dropout (0.5) model
- Figure 6: Heavy Dropout (0.7) model
- Each shows probability contour map with training samples overlaid
- Located after Section 3.3

All figures can be extracted from the accompanying Jupyter notebook (PaddhuML.ipynb) by running the visualization cells. The figures are generated using matplotlib and display classification results in 2D PCA space.

REFERENCES

1. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, 15(1), pp.1929-1958.
2. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L. and Desmaison, A., 2019. PyTorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32.
3. Wolberg, W.H., Street, W.N. and Mangasarian, O.L., 1995. Breast Cancer Wisconsin (Diagnostic) Data Set. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5DW2B
4. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. Journal of machine learning research, 12, pp.2825-2830.
5. Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.