Brain Tumor Classification using Particle Swarm Optimization

Technical Report

Prepared by: Farouk Mohamed Farouk
Supervised by: Prof. Ahmed Magdy
Dr. Iman Mostafa

1. Introduction

Medical image classification, particularly for brain tumors, is a critical application of deep learning in healthcare. Accurate classification of brain tumors from MRI images can assist radiologists and neurosurgeons in diagnosis and treatment planning. However, developing effective convolutional neural networks (CNNs) for this task requires careful tuning of hyperparameters, which significantly impacts model performance. Manual hyperparameter tuning is time-consuming and often suboptimal, creating a need for automated optimization techniques.

2. Problem Objectives

The primary objectives of this project are:

- Develop a CNN model for classifying brain tumor MRI images into four categories: Glioma tumor, Meningioma tumor, No tumor, and Pituitary tumor

- Implement Particle Swarm Optimization (PSO) to automatically tune the hyperparameters of the CNN model

- Evaluate the performance of the PSO-optimized model against baseline approaches

- Identify the optimal hyperparameter configuration for maximizing classification accuracy

3. Mathematical Modeling

3.1. Objective Function

The objective function for our optimization problem is the validation accuracy of the CNN model, which we aim to maximize:

max $f(\theta)$ = Validation Accuracy(θ)

Where θ represents the set of hyperparameters being optimized.

3.2. Constraints

The hyperparameters are constrained within specific bounds to ensure practical implementation:

- Learning rate: $0.00001 \leq lr \leq 0.01$

- Weight decay: $0 \leq wd \leq 0.001$

- Dropout rates (layers 1-3): $0.1 \leq dropout\_i \leq 0.5$

- Dropout rate (fully connected): $0.3 \leq dropout\_fc \leq 0.7$

- Filters in first layer: $16 \leq filters\_1 \leq 64$

- Filters in second layer: $32 \leq filters\_2 \leq 128$

- Filters in third layer: $64 \leq filters\_3 \leq 256$

- Batch size: $8 \leq batch\_size \leq 64$

4. Implementation Details

4.1. PSO Algorithm Implementation

The PSO algorithm was implemented with the following components:

- **Particle Class**: Represents a single solution in the search space

  - Position: Hyperparameter values

  - Velocity: Rate of position change

  - Personal best: Best solution found by the particle

  - Fitness: Validation accuracy achieved

- **PSO Parameters**:

  - Number of particles: 10

  - Maximum iterations: 20

  - Inertia weight (w): 0.5

 - Cognitive coefficient (c1): 1.5   - Social coefficient (c2): 1.5

- **Position Update**:

  $x_i^{(t+1)} = x_i^t + v_i^{(t+1)}$

- **Velocity Update**:

  $v_i^{(t+1)} = w \cdot v_i^t + c1 \cdot r1 \cdot (p_i - x_i^t) + c2 \cdot r2 \cdot (g - x_i^t)$

4.2. CNN Architecture

The CNN model architecture was optimized using PSO with the following components:

- **Convolutional Layers**:
  - Number of layers: 1-3
  - Filters per layer: 16-512
  - Kernel sizes: 2-5

- **Dense Layers**:
  - Number of layers: 1-3
  - Neurons: 32-256 (first layer)
  - Dropout: Optional (0.1-0.5)

- **Training Parameters**:
  - Batch size: 16-128
  - Learning rate: 0.0001-0.01
  - Activation functions: swish, relu, tanh

5. Results and Analysis

5.1. Optimal Hyperparameters

The PSO algorithm found the following optimal configuration:

- **Network Architecture**:
  - Number of convolutional layers: 1
  - Filters in first layer: 16
  - Kernel size: 4
  - Dense layer neurons: 201
  - Activation function: swish

- **Training Parameters**:
  - Learning rate: 0.00261
  - Batch size: 26
  - Dropout: Disabled
  - Dropout rate: 0.135 (not used)

5.2. Performance Metrics

The CNN model trained with the optimal hyperparameters achieved:

- Best validation accuracy: 77.35%
- Final test accuracy: 73.52%
- Training time: 6,613.25 seconds

5.3. Convergence Analysis

The PSO optimization showed consistent improvement across iterations:

- Initial best fitness: ~0.65

- Final best fitness: 0.77

- Convergence achieved around iteration 15

6. Discussion and Future Improvements

6.1. Strengths of the Approach

- **Efficiency**: PSO explored the hyperparameter space more systematically than manual tuning

- **Automation**: Eliminated need for manual hyperparameter selection

- **Performance**: Achieved competitive accuracy with minimal human intervention

6.2. Limitations and Future Work

- **Computational Cost**: Each fitness evaluation required training a CNN

  - Potential solution: Implement parallel particle evaluation

  - Consider using early stopping strategies for unpromising configurations

- **Search Space**: Fixed bounds for hyperparameters

  - Future work: Implement adaptive bounds

  - Consider hierarchical optimization approaches

- **Model Architecture**: Only hyperparameters were optimized

  - Future direction: Include architectural choices in optimization

  - Explore neural architecture search integration

7. Conclusion

The project successfully demonstrated the effectiveness of PSO for hyperparameter optimization in CNN-based brain tumor classification. The optimized model achieved a validation accuracy of 77.35%, showing the potential of automated hyperparameter tuning. The implementation provides a framework that can be extended to other medical image classification tasks and serves as a foundation for further research in combining evolutionary algorithms with deep learning for healthcare applications.

8. References

1. Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. Proceedings of ICNN'95

2. Cheng, S., et al. (2018). A quarter century of particle swarm optimization. Complex & Intelligent Systems

3. Pereira, S., et al. (2016). Brain tumor segmentation using CNNs in MRI images. IEEE TMI

4. Lorenzo, P. R., et al. (2017). PSO for hyper-parameter selection in deep neural networks. GECCO