Exploration of Hyperparameters in Neural Network Design

Name: Meet Viral Shah
USC Id: 5581-2770-75
USC Email: meetvira@usc.edu

This report documents the systematic exploration of hyperparameters to optimize my neural network used for the classification task. The journey focuses on evaluating different network architectures, activation functions, learning rates, decay rates, regularization, and other key hyperparameters to improve the model's performance across multiple data splits. The final model configuration demonstrated a notable improvement in accuracy across five distinct data splits after extensive testing and iteration.

1. **Activation Functions:**
   I began my exploration by comparing two popular activation functions: ReLU (Rectified Linear Unit) and Sigmoid. The initial model setup included three layers with sizes [256, 128, 64], a learning rate of 0.01, and no regularization.

   - ReLU Activation:
     - Dataset 1: 41.7%
     - Dataset 2: 41.8%
     - Dataset 3: 44.0%
     - Dataset 4: 44.1%
     - Dataset 5: 43.2%

   - Sigmoid Activation:
     - Dataset 1: 39.5%
     - Dataset 2: 39.4%
     - Dataset 3: 42.6%
     - Dataset 4: 42.5%
     - Dataset 5: 41.7%

   Observations:
   ReLU outperformed Sigmoid consistently across all datasets. This could be due to the vanishing gradient problem associated with Sigmoid in deeper networks.

2. **Learning Rate and Decay Rate:**
   Next, I experimented with different learning rates and decay rates while keeping the activation function as ReLU and maintaining the same network architecture.

   - Learning Rate = 0.01, Decay Rate = 1.0 (No Decay)
     - Dataset 1: 41.7%
     - Dataset 2: 41.8%
     - Dataset 3: 44.0%
     - Dataset 4: 44.1%
     - Dataset 5: 43.2%

   - Learning Rate = 0.02, Decay Rate = 0.99
     - Dataset 1: 42.9%
     - Dataset 2: 42.8%
     - Dataset 3: 45.5%
     - Dataset 4: 45.6%
     - Dataset 5: 44.7%

- ❖ Learning Rate = 0.04, Decay Rate = 0.99 (Final Model)
    - ➢ Dataset 1: 43.8%
    - ➢ Dataset 2: 43.9%
    - ➢ Dataset 3: 46.8%
    - ➢ Dataset 4: 46.9%
    - ➢ Dataset 5: 45.9%

Observations:
Implementing a decay rate with an optimized learning rate significantly improved performance, showing that gradual reduction in learning rate helps in achieving better convergence.

3. **Regularization (L2 Regularization):**
The impact of L2 regularization was also tested to control overfitting and improve the model's generalization.

- ❖ L2 Regularization = 0.0 (No Regularization):
    - ➢ Dataset 1: 42.1%
    - ➢ Dataset 2: 42.0%
    - ➢ Dataset 3: 45.2%
    - ➢ Dataset 4: 45.3%
    - ➢ Dataset 5: 44.5%

- ❖ L2 Regularization = 0.001 (Final Model):
    - ➢ Dataset 1: 43.8%
    - ➢ Dataset 2: 43.9%
    - ➢ Dataset 3: 46.8%
    - ➢ Dataset 4: 46.9%
    - ➢ Dataset 5: 45.9%

- ❖ L2 Regularization = 0.01:
    - ➢ Dataset 1: 42.8%
    - ➢ Dataset 2: 42.9%
    - ➢ Dataset 3: 45.7%
    - ➢ Dataset 4: 45.8%
    - ➢ Dataset 5: 45.0%

Observations:
A moderate level of L2 regularization (0.001) provided the best compromise between bias and variance, improving model accuracy without overly penalizing the weight size.

4. **Network Architecture:**
I varied the number of layers and the number of nodes per layer to assess the impact on the model's accuracy.

- ❖ Two Layers [128, 64]:
    - ➢ Dataset 1: 39.9%
    - ➢ Dataset 2: 40.1%
    - ➢ Dataset 3: 43.1%
    - ➢ Dataset 4: 43.0%
    - ➢ Dataset 5: 42.3%

- ❖ Three Layers [256, 128, 64] (Final Model):
    - ➢ Dataset 1: 43.8%

    - Dataset 2: 43.9%
    - Dataset 3: 46.8%
    - Dataset 4: 46.9%
    - Dataset 5: 45.9%

- Four Layers [512, 256, 128, 64]:
    - Dataset 1: 43.6%
    - Dataset 2: 43.5%
    - Dataset 3: 46.2%
    - Dataset 4: 46.1%
    - Dataset 5: 45.5%

Observations:
Adding more layers did not necessarily improve the performance significantly and tended to increase the computational cost. The three-layer architecture provided the best balance between accuracy and complexity.

Conclusion:
The exploration of hyperparameters proved crucial in enhancing the model's performance. The final model configuration with ReLU activation, three hidden layers, an optimized learning rate of 0.04, a decay rate of 0.99, and L2 regularization of 0.001 achieved the most satisfactory results across all datasets. Regular updates to learning rates, layer configurations, and appropriate regularization, guided by validation performance, were key strategies in optimizing the neural network.