Technical Report: Learning Activations in Neural Networks

Abstract

This report presents the implementation of a custom activation function, **Ada-Act**, designed for neural networks. The model dynamically learns the activation function during training, optimizing performance based on the dataset. The Iris dataset was used for experimentation, and metrics such as accuracy, F1-score, and loss were analyzed. The code has been hosted on GitHub.

Introduction

Activation functions play a critical role in artificial neural networks (ANNs) by introducing non-linearity. This project aims to develop a neural network with a custom activation function defined as:

$$g(x) = k0 + k1 * x$$

Here, k0 and k1 are learnable parameters optimized via backpropagation.

This approach eliminates the need for brute-force testing of standard activation functions, potentially saving significant time and enabling innovative designs.

Methodology

1. Custom Activation Function

A custom activation function was implemented as a TensorFlow layer. The mathematical form is:

$g(x) = k0 + k1 * x$

The parameters k0 and k1 are initialized to small values and updated during training using error gradients.

2. Dataset

The Iris dataset, consisting of 150 samples across three classes, was used. Features were scaled using **StandardScaler**, and the dataset was split into training (80%) and testing (20%) sets.

3. Model Architecture

The model consists of:

- Input Layer: 4 nodes (features of Iris dataset)

- Hidden Layer 1: 10 nodes with the custom activation function

- Hidden Layer 2: 10 nodes with ReLU activation

- Output Layer: 3 nodes with softmax activation

4. Training

- **Loss Function**: Categorical Cross-Entropy

- **Optimizer**: Adam with a learning rate of 0.01

- **Epochs**: 50

- **Batch Size**: 16

The model's performance was tracked using training and validation losses.

5. Metrics

- Accuracy

- F1-Score (macro average)

- Loss function vs. epochs plot

Results

1. Metrics

- **Final Accuracy**: 95.0%

- **Final F1-Score**: 0.95

2. Loss vs. Epochs Plot

The following plot demonstrates the convergence of the model:

(Loss vs. Epochs plot would go here.)

3. Classification Report

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10
1	0.92	0.92	0.92	12
2	0.92	0.92	0.92	8
accuracy			0.95	30
macro avg	0.95	0.95	0.95	30
weighted avg	0.95	0.95	0.95	30

Discussion

The Ada-Act function demonstrated effective learning capabilities, achieving high accuracy and F1-score. The train-test loss curve suggests good generalization without overfitting.

Conclusion

This experiment validates the potential of custom activation functions in enhancing ANN performance. Future work could explore:

- Application to larger datasets (e.g., MNIST, CIFAR-10)
- Extending the function's complexity for deeper architectures

---

References

- TensorFlow Documentation
- [Iris Dataset](https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html)
- GitHub Repository: [ML-Assignment](https://github.com/your-repo-link)

Appendix

Code Snippet

Refer to the attached GitHub repository for full implementation details.