CSC 455: Machine Learning

Case Study 3

Optimizing Neural Networks for CIFAR-10 Image Classification

Objective

In this case study, you will explore, implement, and optimize various neural network architectures using the **CIFAR-10 dataset**, containing **60,000 32x32 color images** divided into 10 classes. The primary goal is to develop an effective neural network model for image classification. You will delve into Deep Neural Networks (DNNs), Deep and Wide Networks, and Convolutional Neural Networks (CNNs) using Keras' Sequential API to understand the trade-offs and performance differences between these architectures. You will also experiment with optimization and regularization techniques to improve the accuracy and generalization of your models.

Dataset Overview

The CIFAR-10 dataset includes images classified into 10 categories: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The dataset is split into:

- **Training Set**: 50,000 images
- **Testing Set**: 10,000 images

Each image is 32x32 pixels with RGB color channels.

Problem Statement

Build and optimize neural networks using the CIFAR-10 dataset for image classification. Explore and evaluate different neural network architectures to determine which is most suitable for this task, experimenting with deep, wide, and convolutional networks to compare their performances on image data.

Step 1: Build a Deep Neural Network (DNN)

- **Configuration**: Build a model with 20 hidden layers, each containing 100 neurons. Use He initialization and Swish activation.
- **Optimization**: Apply the Nadam optimizer with early stopping to train the model, monitoring validation accuracy.
- **Learning Rate Tuning**: Experiment with different learning rates to identify the optimal value for this architecture.
- **Evaluation**: Assess the model's performance on the test set and document observations on how well it classifies images without convolutional layers.

Step 2: Explore a Deep and Wide Neural Network

- **Configuration**: Implement a model with two branches:
    - **Wide Branch**: A few dense layers with a higher number of neurons.
    - **Deep Branch**: Multiple hidden layers for capturing complex patterns.
- **Combining Layers**: Merge the outputs of both branches and add final layers leading to a softmax output.
- **Optimization**: Use Nadam optimizer, batch normalization, and early stopping.
- **Evaluation**: Compare the Deep and Wide Network's performance with the DNN and evaluate if the combination of wide and deep paths improves classification results.

Step 3: Construct a Convolutional Neural Network (CNN)

- **Configuration**:
    - Start with a few convolutional layers (e.g., Conv2D) using ReLU activations and MaxPooling layers.
    - Continue with convolutional layers for feature extraction, followed by dense layers for classification.
    - Use He initialization for the dense layers and experiment with different kernel sizes in convolutional layers.
- **Optimization**: Add batch normalization, dropout, and early stopping.
- **Evaluation**: Measure the CNN's performance on CIFAR-10 and compare it with the DNN and Deep and Wide Networks, documenting how CNN layers process and classify images effectively.

Step 4: Experiment with Regularization Techniques

- **Batch Normalization**: Apply batch normalization layers to accelerate training and stabilize learning across DNN, Deep and Wide, and CNN models. Record its impact on training speed and final accuracy.
- **Dropout and Alpha Dropout**: Use dropout (standard and alpha dropout) to prevent overfitting, especially within the DNN and Deep and Wide models. Observe the effect of dropout on training time and accuracy.
- **Early Stopping and MC Dropout**: Employ early stopping to avoid overfitting and test Monte Carlo dropout (MC dropout) for uncertainty estimation, particularly with CNNs.

Report Structure

Your submission should include a comprehensive technical report with the following structure:

1. **Introduction**
2. **Dataset Overview**
3. **Problem Statement**
4. **Objectives**
5. **Methodology**
6. **Results**
7. **Discussion and Analysis**
8. **Conclusion**
9. **References**

The Word and latex templates are provided on Canvas.

Deliverables

- **PDF report** with findings and analysis, following the structured format above.
- **Google Colab Notebook** containing code for data preprocessing, model training, and evaluation.