ADAM: A Method For Stochastic Optimization

Introduction

Team name: Entropy Death

- **Vishal Reddy Mandadi** (2019101119)
- **Pranshul Chawla** (2019101057)
- **Kadari Ruthwik Reddy** (2019101121)
- **Naman Verma** (2019101066)
- TA advisor: **Abhinaba Bala**

GitHub Link: https://github.com/vishal-2000/SMAI-project (Restricted Access)

Overview and Problem Statement

Optimizers play a very important role in the world of AI. They are the backbone of the learning process, being as important as the loss function itself. In the language of commons, they say the model what to learn from a given experience (experience being the loss). Gradient descent was the first-ever optimizer, introduced or rather suggested by Cauchy in 1847. Since the advent of backpropagation, AI picked up speed, and scientists have made attempts on a regular basis to solve and find the best optimizers for fast and robust training. This lead to the development of various famous optimizers such as:

1. SGD
2. Momentum
3. Nesterov Accelerated GD
4. ADAGRAD
5. ADADELTA
6. RMSprop

7. ADAM

and many more... Of all these the most famous, popular, and effective optimizer is ADAM, which is also the latest among the ones mentioned above, introduced in 2015. ADAM combines the advantages of RMSprop and Adagrad and is claimed to be very effective and robust with many advantages.

Our objective in brief is to test the performance of ADAM by replicating the results shown in the paper. We will further strive to understand what makes ADAM so great during this process of replication.

Goals and objectives

The main objective of this project is to replicate the results of ADAM on the datasets and models shown in the paper. In order to do so, we are going to construct our own implementation of ADAM and compare its performance with other pre-built optimizers namely - SGD, Momentum, Nesterov Accelerated GD, Adagrad, Adadelta, and RMSprop. The comparison mainly includes:

1. Training graphs, which plot loss vs iterations for each of these optimizers
2. Accuracy and F1 score comparisons between the models trained on the above optimizers

If time permits, we shall also look towards implementing simple visualization of these optimizers on 3D data in MatPlotLib. Further, we would also look into the implementation of ADAMAX and temporal averaging extensions of ADAM, if time permits.

Datasets and models

We plan to invest our time mainly on two datasets - MNIST and CIFAR10. For MNIST data, we will be training an MLP with various optimizers and comparing their performances with each other. And for CIFAR10, we will be building a Convolutional Neural Network with various optimizers. If time permits, we will try out logistic regression on the IMDB rating dataset.

List of deliverables

1. Making data loaders and data pre-processor for MNIST dataset.

2. Implementing the Adaptive Moment Estimation Optimizer on our own from its fundamentals.
3. Implementing Neural Network model for MNIST.
4. Train and test results with the previous model.
5. Making data loaders and data pre-processor for CIFAR-10 dataset.
6. Implementing Convolutional Neural Network model for CIFAR-10.
7. Train and test results with the previous model.
8. Making observations and analysis on the results obtained.
9. Making an optimizer Visualizer (if time permits)

Milestones and Timeline

Aa Week	Tasks
Week-1	data loader and pre-processor for MNIST, coding ADAM optimizer, buliding neural network for MNIST, prelimnary training and test results
Week-2	data loader and pre-processor for CIFAR-10, coding ADAM optimizer, buliding neural network for CIFAR, prelimnary training and test results
Week-3	Training and Testing both models, testing the efficiency of ADAM, analysing the results

Work Distribution

▼ MNIST data loaders and pre-processors for MNIST - Ruthwik and Vishal

▼ Implementing ADAM from basics - Vishal

▼ Multi-Layer Perceptron model for MNIST - Naman and Pranshul

▼ Training and testing MNIST model - Naman and Ruthwik

▼ data loaders and pre-processors for CIFAR10 - Naman and Pranshul

- Build CNN model for CIFAR10 - Vishal and Ruthwik
- Training and testing for CIFAR-10 model- Pranshul and Vishal
- Results and analysis - Full team