Deep into CNN

MID-EVAL REPORT

Week 1

- **Plan/Goals**
  - **Numerical data** : Multi layer Perceptron (MLP) :
    - Regression python Implementation
    - Gradient Descent,relu layer, MSE loss
    - Binary Classification,sigmoid layer,BCE loss
    - Multiclass Classification,softmax layer
  - **NLL loss MLP + PyTorch** :
    - Linear Algebra, Single Layer NN, Training, - Inference and Validation : Illustrated Through Pytorch
    - Implement 1-hidden layer NN using PyTorch but train in python

- **Tasks**
  - **Content reading on Regression And Shallow NN Using Python**.
    - **Things Learnt:**
      - **Cost/Loss function:** Structure and the basic use of cost/loss functions were taught
      - **(Stochastic) Gradient descent:** It includes the process of linear descent and how it works. It also included the limit and use of parameters like learnrate while implementing gradient descent
      - **Python Implementation from scratch**: All the above things were taught in python without using any big library. It was done to give a clear picture of how different things/functions work and implemented directly.

- **Completing the programming exercises shared and updating github repo with practice code and completed exercises.**
    - **Things learnt:**
        - **Basic data handling with numpy and pandas**: Mentees were taught how to load data from csv and clean the data with the help of numpy and pandas. Other than the basic functions, mentees were also taught One-hot encoding the data, normalisations and its importance,and some implementations of matplotlib to view data.
        - **Implementing sigmoid and error calculation functions**: sigmoid and error calculation functions were taught and implemented with numpy.
        - **Training and implementing shallow NN from scratch in python:** After all the functions were made, mentees implemented the them along with error term calculation and back-propagation to make a shallow NN and there accuracy and loss were observed after every epochs

Weeks 2

- **Plans/Goals**
    - **Intro to CNN :**
        - Simple Feed-forward Network :
            - Flatten images first and then treat them as numerical data.
        - Convolutional Neural Networks :
            - Use Spatial Information
        - Compare results with MLP on MNIST data
        - Start Using PyTorch.
- **Tasks**
    - **Content reading on Neural Networks in pytorch**
        - **Things learnt -**

- **Backprop:** Though it was already implemented in week 1, but it was used only in shallow NN and therefore it was included again in week 2 to give a complete sense of its implementation on deep NN. Also, this time it was implemented using pytorch.

- **Softmax**: Mentees were taught about the use, importance and implementation of softmax functions from scratch and in pytorch.

- **Basic of Pytorch**: Assignments were given to teach mentees the implementation of different functions (related to NN) in pytorch.

- **Complete the programming exercises shared and update github repo with practice code and solved assignments:** Total of 8 assignments were given. Mentees were taught how to -

    - Load and handle data using pytorch
    - Uses and implementation of dataloaders and importance of parameters like batch-size
    - Loading already available data using torch vision and the process of normalisation
    - Basic structure and documentation about the NN in pytorch
    - Criterions like CrossEntropyLoss, NLLLoss,
    - Optimizer like Adam, SGD
    - Use, importance and complete implementation of pytorch autograd and its use with loss functions.
    - Training of neural networks and setting of hyperparameters and learning rate to adjust data along with RELu and LogSoftmax functions
    - Importance of Validating the data during training and its implementation using the Dropout function.
    - Saving and loading back the already trained models and its importance
    - An optional exercise to make a Cat-Dog identifier was also given along with the basic framework required to make it.

- **Hackathon-1 starts**: The Hackathon 1 data set was very noisy and was given to give an idea of how Kaggle and Hackathon works. It also signified an important fact that simple fully connected NN can sometimes be inefficient to train and predict the data accurately.

Weeks 3

- **Plans/Goals**
    - **LeNet:**
        - Convolution
        - Pooling
        - Fully connected layers
    - Competition of Hackathon 1
    - Practice assignments on CNN

- **Tasks**
    - **Hackathon-1 submission**: The hackathon was an open one. Hosted on kaggle and included a total of over 800 teams (Tabular Playground Series - Jun 2021). Following were top 5 scores of mentees along with their scores:

Rank	TeamName	Score
67	BTL	1.7449
98	Vyjayanthi Reddy	1.74601
443	Cosmos	1.7591
495	Riya Bhalla	1.76537
504	WarriorZ	1.76777

    - **Hackathon-2 starts:** It is based on training NN and making predictions for RGB images.
    - **Read a famous shared paper and update github repo with practice code and solved implementation of that paper**
        - **Things learnt:** Mentees were asked to choose one of the following SOTA models on ImageNET classification papers and implement it.
            - AlexNet
            - VGG
            - Inception
            - Xception