Code implementation of the paper:
A New Framework for Multi-Hop ABS-Assisted 5G- Networks with Users' Mobility Prediction

A Report Submitted in Fulfilment

of

Bachelor of Technology

in

Computer Science & Engineering

By

Kartik Deepak Dange (20204093)
Priyanshu Upman (20204155)
Manish Kumar Singh (20204107)
Lokesh Kumar (2020410)

under the guidance of

Dr. Shailendra Shukla

to the

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY, ALLAHABAD

November, 2023

UNDERTAKING

I declare that the work presented in this report titled Code implementation of the paper: A New Framework for Multi-Hop ABS-Assisted 5G- Networks with Users' Mobility Prediction, submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology, Allahabad, for the completion of the Major Project is my original work. I have not plagiarized or submitted the same work anywhere else before. In case this undertaking is found incorrect, I accept that my degree may be unconditionally withdrawn.

November, 2023

Allahabad

(Kartik Deepak Dange 20204093)

(Priyanshu Upman 20204155)

(Manish Kumar Singh 20204107)

(Lokesh Kumar 20204104)

CERTIFICATE

Certified that the work contained in the report titled "Code implementation of the paper: A New Framework for Multi-Hop ABS-Assisted 5G- Networks with User's Mobility Prediction", by Kartik Deepak Dange, Priyanshu Upman, Manish Kumar Singh, Lokesh Kumar, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Dr. Shailendra Shukla)
Computer Science and Engineering Dept.
M.N.N.I.T, Allahabad

November, 2023

Preface

Human mobility machine learning refers to the utilization of deep learning techniques for analysing and modelling human mobility patterns by utilizing large-scale mobility datasets and neural network architectures. The objective of deep move is to identify fundamental patterns and predict future movements of individuals and groups.

Deep move human mobility machine learning has the potential to benefit society in many ways, such as improving transportation efficiency, emergency response planning, and public health measures. However, ethical considerations such as privacy protection, bias mitigation, and transparency must be thoroughly addressed to ensure responsible and fair development and deployment of deep move models.

Acknowledgements

We must mention several individuals who were of enormous help in developing this work. Dr. Shailendra Shukla, our mentor, encouraged us to carry out the work. His continuous and invaluable guidance helped us complete the work, and we hope to continue further. We want to express our sincere gratitude to Prof. R. S Verma, Director, MNNIT Allahabad, Allahabad and Prof. D. K. Yadav, Head, Computer Science and Engineering Department, for providing us with all the facilities required for the completion of this work. We would also like to thank all of our friends for their constant motivation, advice and support.

Contents

Chapter 1

Introduction

Human mobility prediction is of great importance for a wide spectrum of location-based applications. Human mobility is crucial due to its impact on several aspects of our society, such as disease spreading, urban planning, well-being, pollution, and more. Predicting human mobility leads to higher complexities.

1.1 Motivation

Unmanned Aerial Vehicles (UAVs) are increasingly recognized as a promising solu-tion in 5G networks. Utilizing UAVs as Aerial Base Stations (ABS) alongside tradi-tional cellular infrastructure offers a flexible and cost-effective approach to cater to on-demand communications in these advanced networks. However, deploying UAVs poses challenges, including optimal 3D placement, coverage enhancement, and back-haul limitations.

To address these challenges, we propose a novel framework for the 3D positioning of an ABS swarm to extend the coverage of ground base stations (GBS). This scheme allows ABSs to stay connected to the core network through GBSs or other deployed ABSs. The objective is to maximize the number of served User Equipments (UEs) by optimizing ABS placement.

Our main objective is to integrates machine learning techniques to predict user mobility. This enables ABSs to dynamically adjust their positions based on the spatial distribution of users, contributing to an adaptive and efficient deployment. In this project we are basically tended towards human mobility that can be more easy by using Transformer encoder-decoder architecture.

In contrast to ground-based stations, employing UAVs as aerial base stations offers the following benefits:

- Elevated altitudes of Aerial Base Stations (ABS) result in increased line-of-sight (LOS) connections with ground base stations and users. This enables ABSs to efficiently supplement existing cellular systems, enhancing capacity, coverage, and rate performances [3].

- Due to their high mobility and agility, Aerial Base Stations (ABSs) can navigate around obstacles, optimizing their position to increase line-of-sight (LoS) links. This capability enables ABSs to offer rapid, flexible, and on-demand wireless communication.deployment.

- Furthermore, Aerial Base Stations (ABSs) are a cost-effective alternative to traditional communication infrastructures. This affordability makes them a practical choice for swift service recovery in the aftermath of natural disasters like hurricanes, floods, and earthquakes [4].

1.2 Human Mobility

Transformers, a breakthrough in machine learning, have demonstrated remarkable capabilities in various applications. In the realm of human mobility prediction, transformers excel in capturing complex spatial and temporal patterns. Leveraging their attention mechanisms, transformers efficiently process sequences of mobility data, considering the dynamic interactions and dependencies over time. However, predicting mobility is not trivial because of three challenges:

- the complex sequential transition regularities exhibited with time-dependent and high-order nature;

- the multi-level periodicity of human mobility; and

- the heterogeneity and sparsity of the collected trajectory data

1.3 Challenges :

- The regularity of human mobility keeps on varying and changes on daily basis.
- The patterns of sequential transitions are highly complex, time-dependent, and exhibit a high-order nature.
- The collected human mobility data exhibits heterogeneity and sparsity.

1.3.1 Examples of Challenges :

- Ex-1 : The probability of a commuter moving from home to office is higher on workday mornings than on weekends. However, this transition pattern can be irregular and high-order due to the possibility of people visiting other places on their commute routes.
- Ex-2 : Human mobility periodicity is complex and multi-level, involving daily routines, leisure activities, festivals, and personal periodic events. These activities are interwoven in intricate ways and are difficult to capture.
- Ex-3 : The majority of human mobility data is sparsely sampled, with location information recorded only when the user accesses the location service.

Chapter 2

Related Work

While the deployment of Aerial Base Stations (ABSs) presents promising opportunities, a substantial number of technical challenges must be addressed to fully harness their potential in the next generation of wireless networks. Main challenges could be :

2.1 Optimal ABS Placement

The flexible positioning of Aerial Base Stations (ABSs) is vital for swift on-demand wireless connections to ground users. However, deploying ABSs in a continuous 3D space presents challenges, impacting channel quality and network performance [14]. Achieving optimal ABS deployment for coverage extension, capacity maximization, and applications like public safety or IoT has garnered significant interest, with two main scenarios to consider.

1) Single ABS Placement: In [14] and [15], the 3D placement of a single Aerial Base Station (ABS) serving a set of users is explored. The goal is to maximize the coverage of users. In [14], the problem is formulated as a quadratically constrained mixed-integer nonlinear optimization problem, with a proposed numerical solution. Meanwhile, [15] models it as a circle placement and smallest enclosing circle problem, constrained by the minimization of transmit power. Additionally, [18] investigates the optimal positioning of an ABS serving as a relay between a Ground Base Station

(GBS) and a fixed-position user in a dense urban area.In [21], the research focuses on a wireless network featuring an Aerial Base Station (ABS) deployed as a relay between a transmitter and a receiver. The primary goal is to identify the optimal ABS location that maximizes the average rate.

2) Multiple ABSs Placement: Deploying multiple Unmanned Aerial Vehicles (UAVs) simultaneously poses challenges, particularly due to the impact of inter-cell interference. In [9] and [10], the authors explore scenarios where a set of Aerial Base Stations (ABSs) hover at the same altitude, focusing on the placement problem in the horizontal plane. In [9], they present an optimization model to strategically position ABSs, aiming to maximize covered users while minimizing communication costs between ABSs. Meanwhile, [10] addresses a fleet of ABSs covering mobile sensors, optimizing their placement to report information to the ground. The op-timal placement of UAVs is achieved through an optimization model that reduces variables and utilizes column generation.

In [11], the study encompasses both drone base stations and drone users. The proposed method involves a truncated octahedron shapes-based approach for the 3D placement of Aerial Base Stations (ABSs), with optimal cell association defined using optimal transport theory to minimize drone users' latency. On the other hand,

[22] explores the use of multiple ABSs as relays to enhance the connectivity of a ground wireless network. The focus is on optimizing the deployment of ABSs to ensure the delivery of sensors' messages to their destinations.

2.2 Path Planning

Leveraging the high mobility of Aerial Base Stations (ABS) enables cellular networks to fully unlock the potential of ABS-assisted networks. However, designing and optimizing ABS trajectories pose challenges, as they involve exploring an infi-nite number of potential ABS locations [30].

Trajectory optimization in ABS-assisted wireless networks is considered more challenging. Indeed, it has to simultaneously process both mobility and QoS metrics constraints. In [8], the study focuses on a wireless communication system where a group of Aerial Base Stations (ABSs) serves ground users. The authors optimize ABS trajectories to maximize the average minimum throughput of ground users, formulating the problem as a mixed-integer non-convex optimization problem. Similarly, [31] introduces a path planning algorithm for ABSs with multiple antennas in a comparable scenario, aiming to maximize overall throughput in uplink communication.

2.3 Network Planning

Network planning entails many issues such as backhaul management, user-cell as-sociation, frequency allocation, and interference mitigation, etc. It becomes more challenging in the context of ABS-assisted wireless networks due to ABS mobility, limited wireless backhaul connectivity, and LoS interference.

The association between users and Aerial Base Stations (ABSs) in ABS-based cellular networks is explored in [36], [37], and [2]. The studies in [36], [37] seek to determine the minimum necessary number of ABSs and their association with ground users to maximize network coverage and capacity, respectively. In [2], the focus is on meeting user-required rates using the minimum transmit power from ABSs. Additionally, [38] delves into optimal cell partitioning between ABSs and ground base stations, employing optimal transport theory tools to address the association problem.Unlike ground base stations, which have a strong reliable wired/wireless connection.Authors in [13] consider the 3D placement of ABSs to maximize coverage and the average rate of users link under a predefined fixed limited backhaul rate and bandwidth constraints with the core network, untethered flying ABSs require a wireless backhaul link to connect with the core network.

TABLE I
ABS-ASSISTED WIRELESS NETWORKS: CHALLENGES, REFERENCES, COVERED ISSUES AND USED TOOLS

Challenges	Key references	Covered issues	Tools and techniques
3D placement	[15] [16], [17], [18] [20] [8] [9] [10] [2], [11] [19], [21] [22]	- Deployment in the presence of terrestrial networks - Effect of ABS altitude on the network performance - Joint 3D placement optimization and QoS maximization - Joint 3D placement and user-ABS association	- Centralized optimization theory - Optimal transport theory - Stochastic geometry
Channel modelling	[23] [6] [24], [25] [26], [27] [28].	- Air-to-ground channel modeling - Air-to-ground path loss modeling - LoS and NLoS probabilities - Small scale fading modeling	- Extensive measurement - Ray-tracing - Probabilistic models
Trajectory optimization	[29], [7], [30] [31] [32] [33]	- Energy-efficient trajectory optimization - Joint trajectory and QoS optimization	- Centralized optimization theory - Machine learning
Network planning	[36] [2]. [35], [37] [38]. [12] [39]. [40]	- Joint trajectory optimization and user-ABS association. - Joint trajectory and transmit power optimization. - ABS backhauling	- Centralized optimization theory - Optimal transport theory - Facility location theory

Table 1. Related course, References, Issues, Tools

Chapter 3

Proposed Work

3.1 Objective –

Optimize Aerial Base Station (ABS) deployment by strategically predicting users' mobility for improved connectivity and seamless handovers, focusing on elevating overall user experience in dynamic environments. The framework aims to contribute to efficient and adaptive 5G networks, emphasizing accurate mobility prediction.

Section 1. The Challenges of using UAVs as Aerial Base Stations (ABS)

Deploying Aerial Base Stations (ABSs) in 5G networks presents challenges in optimal placement, trajectory optimization, channel modeling, and network planning. The three-dimensional deployment of ABSs requires addressing complexities in maximizing coverage, capacity, and ensuring public safety or IOT applications. Addressing these challenges is crucial for realizing the potential benefits of ABS-assisted 5G networks.

Section 2. System Model and Description

The system aims to extend coverage and provide access to mobile users by deploying Unmanned Aerial Vehicles (UAVs) as dynamic Aerial Base Stations (ABSs) alongside Ground Base Stations (GBSs) in a multi-hop architecture. The system's adaptability relies on machine-learning algorithms predicting users' mobility, enabling real-time adjustments to ABS positions and associations based on changing user distributions.

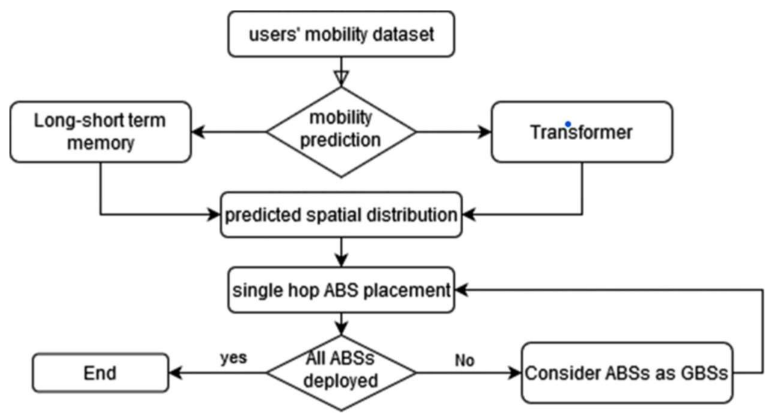
Problem - Design a robust solution for predicting human mobility to enhance the deployment of ABSs in a wireless communication system. The focus is on maximizing cellular network coverage and serving a maximum number of mobile User Equipment's (UEs) efficiently.

Section 3. Multi-Hop SSO- framework for 3D placement of ABS

User locations are collected at each time slot T, divided into T time slots for optimizing ABS trajectories. Prior to each time slot, predictions for the user distribution in the next T time slots are generated. ABS positions are then computed by solving an optimization problem considering user associations and backhaul links. The Adaptive SSO-based Framework (ASF) iteratively refines ABS placement through single-hop ABS positioning and adaptation stages, ensuring optimal joint association of UEs with ABSs. The Social Spider Optimization (SSO) algorithm efficiently addresses the mixed-integer nonlinear programming (MINLP) nature of the ABS positioning problem, contributing to the effective deployment of multi-hop networks while accommodating user mobility predictions.

Section 4. Users Mobility Prediction

In our user mobility prediction, we employ advanced machine learning models to tackle the intricacies of the task. Specifically, we harness the Transformer Model, renowned for its success in natural language processing, and the proven Long-Short Term Memory (LSTM) Encoder-Decoder architecture, well-suited for handling sequence dependencies. The LSTM utilizes a gating mechanism with forget, input, and output gates, while the Transformer leverages attention mechanisms, eliminating the need for recurrence and convolutions. The self-attention feature enables simultaneous focus on various input aspects, and the multi-head attention model enhances the collaborative processing of diverse representations, contributing to heightened predictive accuracy.



[1]https://ieeexplore.ieee.org/abstract/document/9707619

Fig. Adaptive SSO-based framework for ABSs placement.

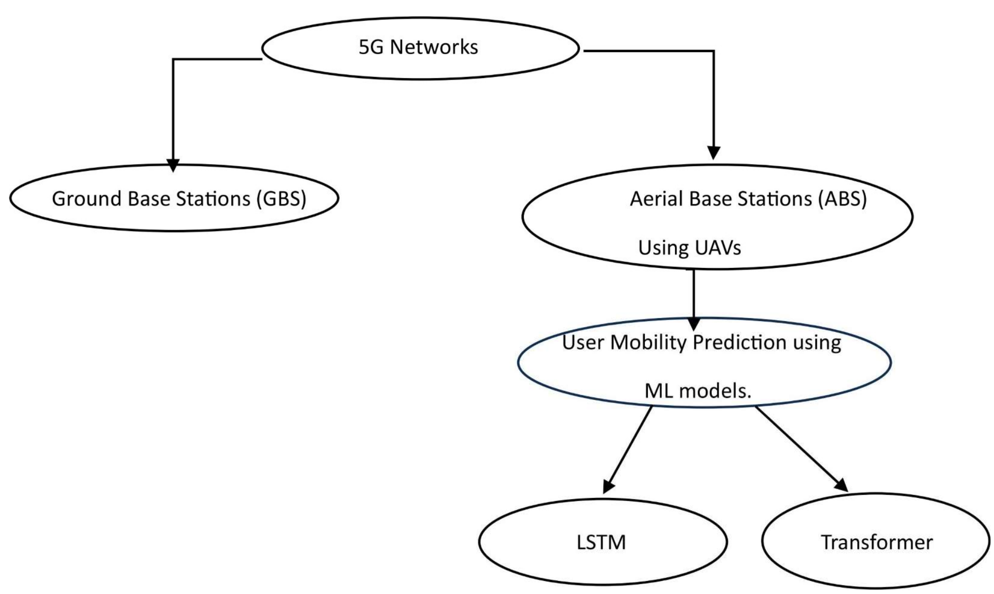


Fig. Flowchart of proposed model

3.2 Overview of Transformers

The Transformer model, initially proposed by Google for machine translation [55], features an encoder-decoder architecture. The encoder maps the input sequence Y to a continuous sequence H, utilized by the decoder for generating the target sequence Y'. Notably, the Transformer relies solely on attention mechanisms, eliminating recurrence and convolutions. Positional encoding is incorporated for sequence order, and the encoder comprises stacked identical layers, each housing a multi-head self-attention and point-wise feed-forward network (FFN) layer. Residual connections, followed by normalization layers [56], are applied around both multi-head and feed-forward network layers.

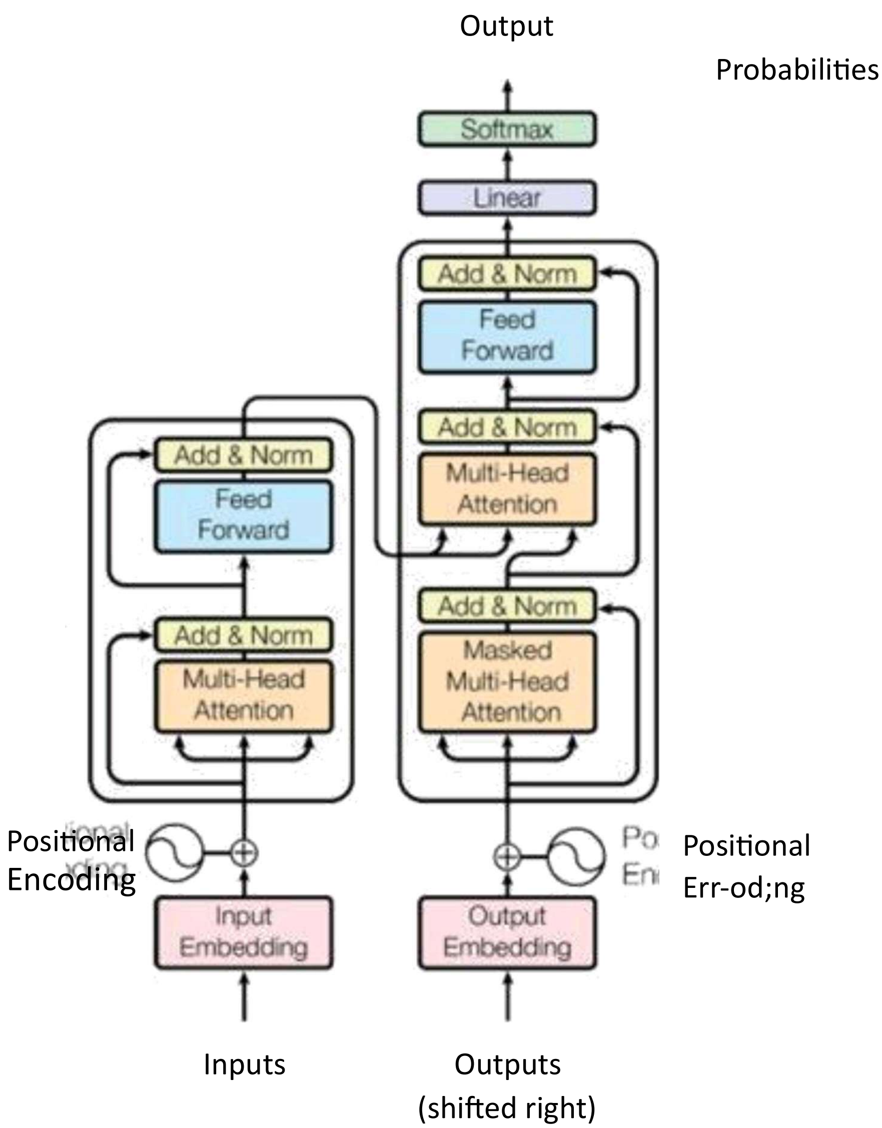


Fig 1. Transformer model architecture[361

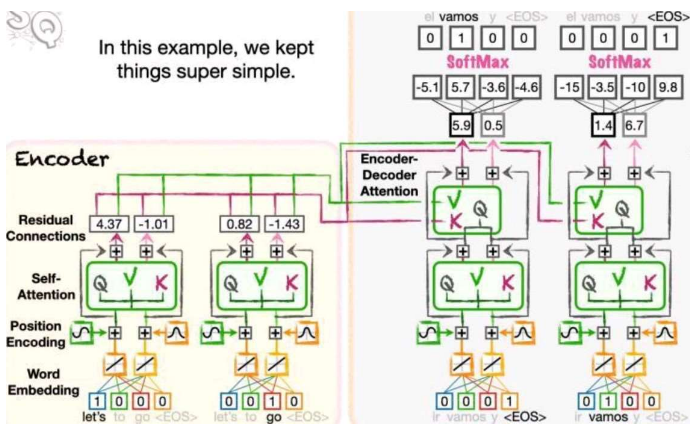
3.3 What are LSTM [7] in neural networks

The deepMove mpdel relies on Long short term memory, or LSTM, which is quite famous deep learning strategy. Many RNNs have the capacity to recall long-term dependencies, which is helpful in situations involving sequence prediction. Speech recognition, machine translation, and other areas are applications for LSTM, which has feedback connections.

3.4 The logic behind Transformer encoder- decoder

First, the output of an LSTM is fundamentally dependent on three factors at any one time:

- Encoder-Decoder Architecture: Transformer uses a dual structure with an encoder mapping input sequence Y to a continuous sequence H, and a decoder generating the target sequence Y'.



Decoder

Fig 2. Encoder decoder model [35]

- Attention Mechanisms: It relies solely on attention mechanisms, dispensing with recurrent and convolutional layers. This allows the model to capture dependencies across the entire input sequence.

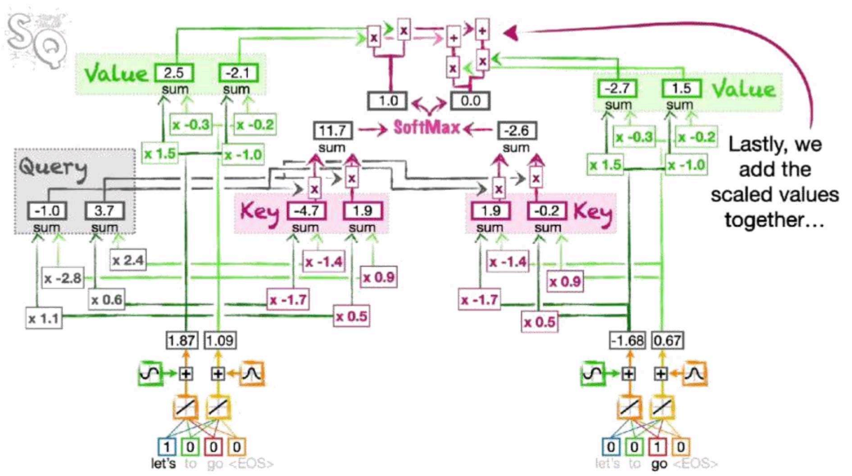


Fig 3. Self-Attention in encoder [35]

- Positional Encoding: To provide sequence order information, positional en-coding is added to both input and output embeddings.

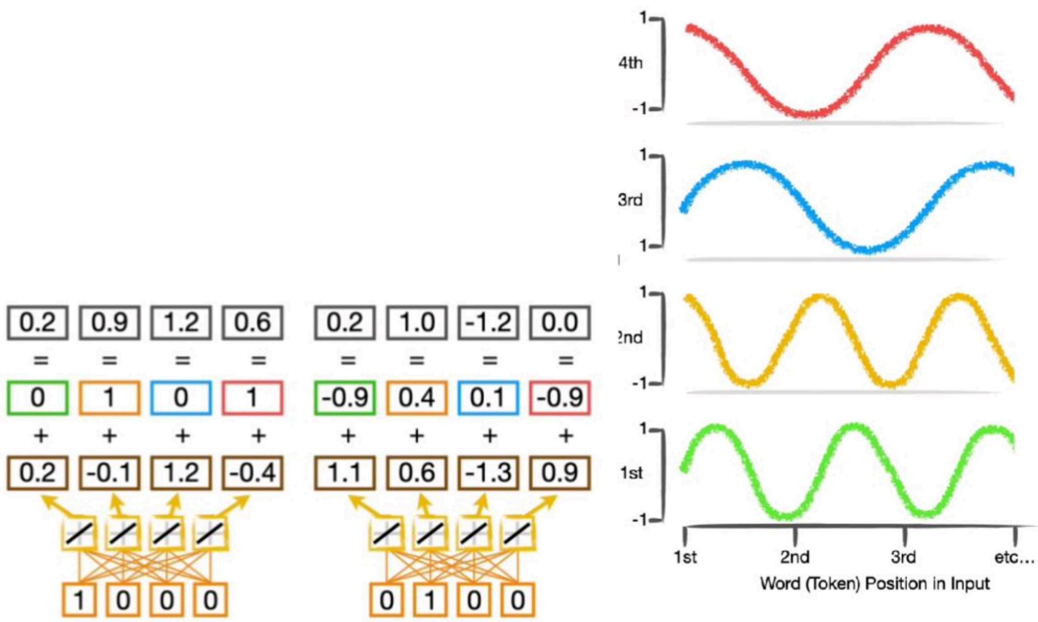


Fig 4. Positional Encoding in encoder/decoder[35]

Fig 5. Word embedding[35]

- Multi-Head Self-Attention: The encoder comprises stacked identical layers, each containing a multi-head self-attention mechanism, enabling the model to consider multiple aspects of the input simultaneously.

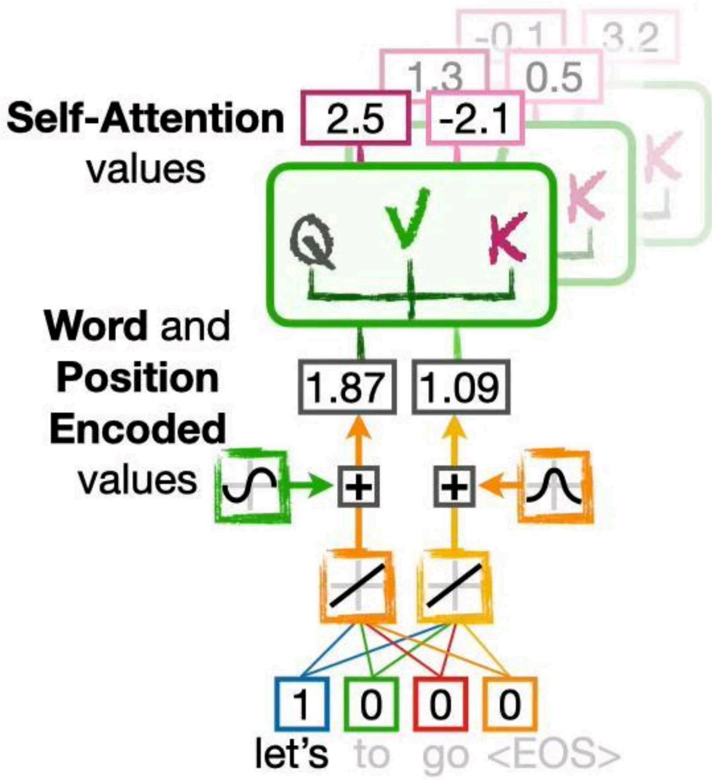


Fig 6. Multi-head attention[35]

- Feed-Forward Network (FFN) Layers: Each layer in the encoder includes a point-wise feed-forward network (FFN) layer to process the attention layer's output.

- Residual Connections and Normalization: Residual connections, along with normalization layers, are applied around both the multi-head and feed-forward network layers, aiding in training stability.

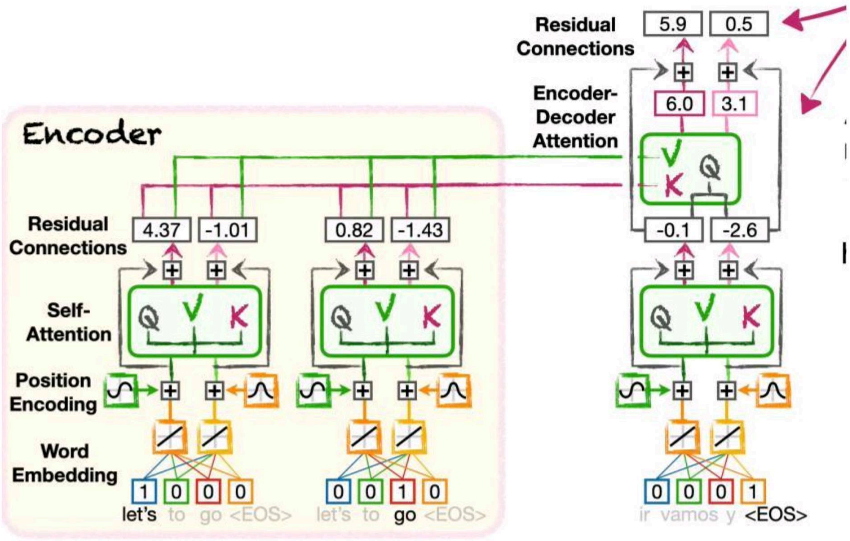


Fig 7. Encoder-decoder attention[35]

3.5 Optimizer [9]:

Optimizer algorithms play a crucial role in enhancing the performance of deep learn-ing models. An optimization algorithm is utilized to discover the optimal values of the weight in order to minimize the error. During deep learning model training, optimizers modify neural network attributes, including weights and learning rates, to minimize the loss function at each epoch. This leads to improved accuracy of the model.

3.5.1 Adam :

The Adam optimizer combines the advantageous aspects of both the Adagrad and RMSprop algorithms. Unlike stochastic gradient descent, which maintains a fixed learning rate throughout training, the Adam optimizer updates the learning rate individually for each weight in the neural network. With the Adam optimizer, each weight in the neural network receives a personalized learning rate, leading to enhanced optimization by considering the unique characteristics of each weight. This adaptivity helps achieve better convergence and handles sparse gradients effectively, making the Adam optimizer a widely utilized choice in deep learning .

Chapter 4

Experimental Setup and Results Analysis

4.1 System Requirements

The project can be easily run on consumer-grade hardware. Any modern computer with 4 GB of RAM and a 2 GHz or higher processor, should have no trouble handling the same.

4.2 Software Requirements

- python 3.8.2
- numpy 1.18.1
- torch 1.4.0

4.2.1 VS Code installation :

- Download the visual studio Code installer for specific system from the official VS code site.
- Run the installer(VSCodeUserSetup-version), install in default setting.

- By default for windows, VS code is installed under C: Microsoft VS Code.

4.2.2 Python:

- Go to the official Python download page for Windows.
- Downloading the Python Installer.
- After the installer is downloaded, execute the .exe file in default setting.
- Add python to the environment variable(opt).

4.2.3 Pytorch :

- Visit the official Pytorch website, there choose system specific requirements and run the command generated in your terminal.
- Eg : “pip3 install torch torchvision torchaudio –index-url
- Link : Download PyTorch

4.3 Running the project :

- Open terminal at the project location.
- ”python train.py –data name=foursquare –data path=../data/”
- Change the command according to the need of arguments.

4.4 Dataset

1) The Foursquare dataset is a collection of location check-ins obtained from the Foursquare app, which allows users to check-in at various venues such as restaurants, bars, parks, and other points of interest. The dataset contains information about the user’s location (latitude and longitude), the time of the check-in, and the venue’s

name and category. It has been used in various studies related to human mobility, including trajectory prediction and location recommendation systems.

- Link :

https://www.kaggle.com/datasets/chetanism/foursquare-nyc-and-tokyo-checkin-dataset

2)The GeoLife dataset is a widely used dataset in the field of human mobility research. It consists of GPS trajectory data collected from the Microsoft Research Asia GeoLife project, where participants used GPS-enabled devices to record their movements over an extended period. The dataset contains diverse mobility patterns, including various transportation modes and activities, making it valuable for studying location-based services, mobility prediction, and urban analytics. Researchers commonly leverage GeoLife for developing and testing algorithms related to trajectory analysis, route recommendation, and understanding human mobility behaviour in different contexts.

- Link :

https://www.microsoft.com/en-us/download/details.aspx?id=52367

4.5 Changes in Model :

- Markov Model which is widely used in human mobility prediction.
- RNN[1,2] model produced from the paper, which either feeds data to model day by day or whole month trajectory at once.
- The default training setting for the data is :

Training Settings	Value	Feature (Size)	Input	Output
learning rate (*lr*)	1e-3	location	≈10000	256
the decay of *lr*	0.1	time	{48,168}	16
L2 penalty	1e-5	user ID	≈1000	16
gradient clip	1.0	hidden state	300	256

Table 2: The default parameter settings for DeepMove.

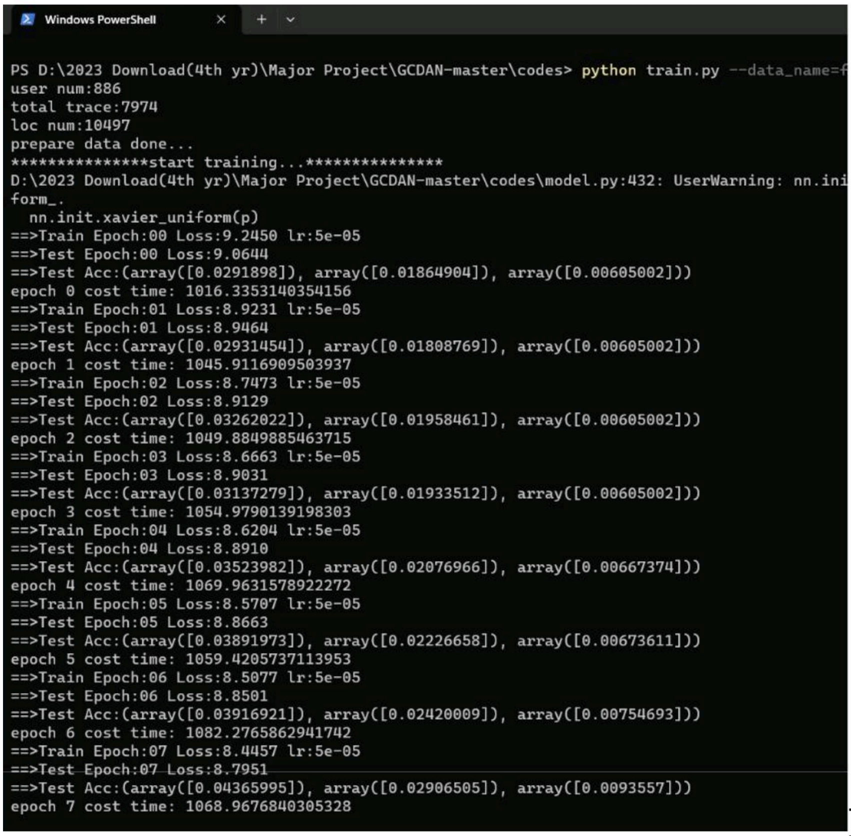
- The model is trained in train-test mode where 70 percent of the trajectory acts as the training dataset while the remaining 30 percent is used as the testing dataset.

- We can also specify the optimizer[9,10] between Adam and SGD.

- The max epoch is set to 50 by default and learning rate is set to default = 5*1e-4 .

4.6 Accuracy

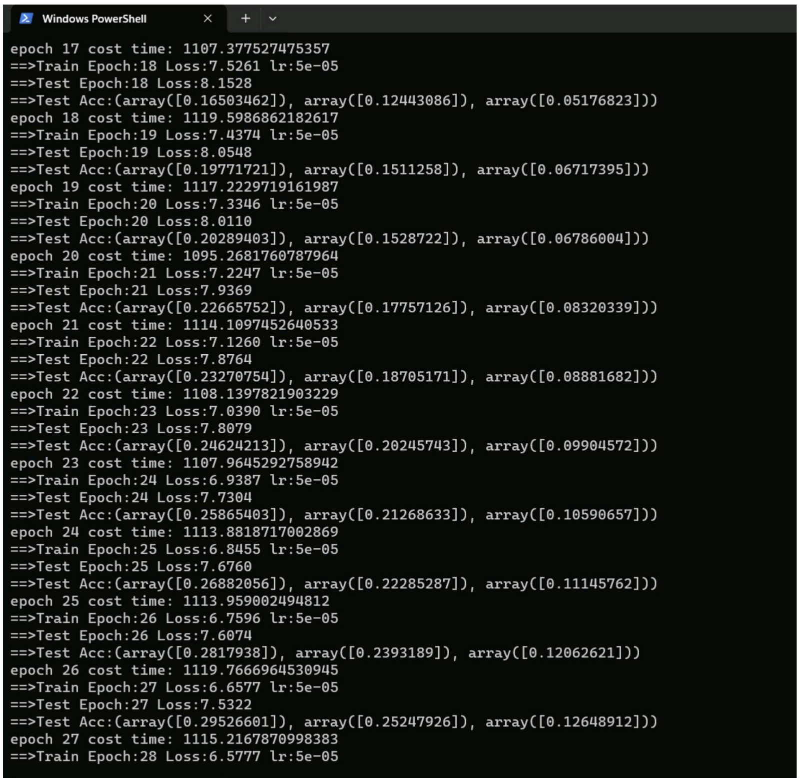
- The accuracies achieved by us :



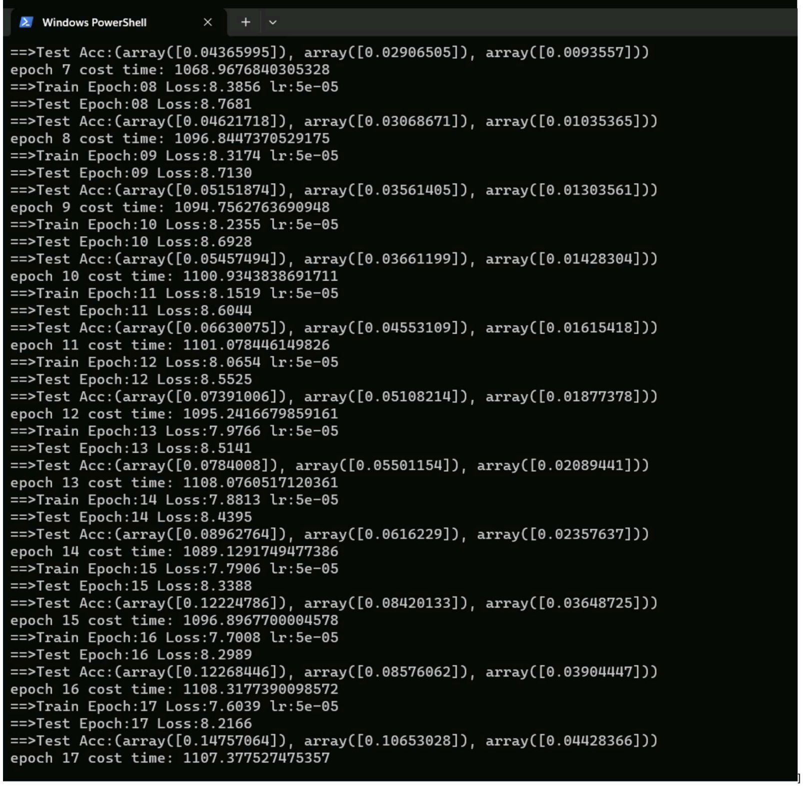
```
Windows PowerShell

PS D:\2023 Download(4th yr)\Major Project\GCDAN-master\codes> python train.py --data_name=f
user num:886
total trace:7974
loc num:10497
prepare data done...
***************start training...***************
D:\2023 Download(4th yr)\Major Project\GCDAN-master\codes\model.py:432: UserWarning: nn.ini
form_.
  nn.init.xavier_uniform(p)
==>Train Epoch:00 Loss:9.2450 lr:5e-05
==>Test Epoch:00 Loss:9.0644
==>Test Acc:(array([0.0291898]), array([0.01864904]), array([0.00605002]))
epoch 0 cost time: 1016.3353140354156
==>Train Epoch:01 Loss:8.9231 lr:5e-05
==>Test Epoch:01 Loss:8.9464
==>Test Acc:(array([0.02931454]), array([0.01808769]), array([0.00605002]))
epoch 1 cost time: 1045.9116909503937
==>Train Epoch:02 Loss:8.7473 lr:5e-05
==>Test Epoch:02 Loss:8.9129
==>Test Acc:(array([0.03262022]), array([0.01958461]), array([0.00605002]))
epoch 2 cost time: 1049.8849885463715
==>Train Epoch:03 Loss:8.6663 lr:5e-05
==>Test Epoch:03 Loss:8.9031
==>Test Acc:(array([0.03137279]), array([0.01933512]), array([0.00605002]))
epoch 3 cost time: 1054.9790139198303
==>Train Epoch:04 Loss:8.6204 lr:5e-05
==>Test Epoch:04 Loss:8.8910
==>Test Acc:(array([0.03523982]), array([0.02076966]), array([0.00667374]))
epoch 4 cost time: 1069.9631578922272
==>Train Epoch:05 Loss:8.5707 lr:5e-05
==>Test Epoch:05 Loss:8.8663
==>Test Acc:(array([0.03891973]), array([0.02226658]), array([0.00673611]))
epoch 5 cost time: 1059.4205737113953
==>Train Epoch:06 Loss:8.5077 lr:5e-05
==>Test Epoch:06 Loss:8.8501
==>Test Acc:(array([0.03916921]), array([0.02420009]), array([0.00754693]))
epoch 6 cost time: 1082.2765862941742
==>Train Epoch:07 Loss:8.4457 lr:5e-05
==>Test Epoch:07 Loss:8.7951
==>Test Acc:(array([0.04365995]), array([0.02906505]), array([0.0093557]))
epoch 7 cost time: 1068.9676840305328
```

]

Fig 7. Accuracy at different epoch stages



```
Windows PowerShell
epoch 17 cost time: 1107.377527475357
==>Train Epoch:18 Loss:7.5261 lr:5e-05
==>Test Epoch:18 Loss:8.1528
==>Test Acc:(array([0.16503462]), array([0.12443086]), array([0.05176823]))
epoch 18 cost time: 1119.5986862182617
==>Train Epoch:19 Loss:7.4374 lr:5e-05
==>Test Epoch:19 Loss:8.0548
==>Test Acc:(array([0.19771721]), array([0.1511258]), array([0.06717395]))
epoch 19 cost time: 1117.2229719161987
==>Train Epoch:20 Loss:7.3346 lr:5e-05
==>Test Epoch:20 Loss:8.0110
==>Test Acc:(array([0.20289403]), array([0.1528722]), array([0.06786004]))
epoch 20 cost time: 1095.2681760787964
==>Train Epoch:21 Loss:7.2247 lr:5e-05
==>Test Epoch:21 Loss:7.9369
==>Test Acc:(array([0.22665752]), array([0.17757126]), array([0.08320339]))
epoch 21 cost time: 1114.1097452640533
==>Train Epoch:22 Loss:7.1260 lr:5e-05
==>Test Epoch:22 Loss:7.8764
==>Test Acc:(array([0.23270754]), array([0.18705171]), array([0.08881682]))
epoch 22 cost time: 1108.1397821903229
==>Train Epoch:23 Loss:7.0390 lr:5e-05
==>Test Epoch:23 Loss:7.8079
==>Test Acc:(array([0.24624213]), array([0.20245743]), array([0.09904572]))
epoch 23 cost time: 1107.9645292758942
==>Train Epoch:24 Loss:6.9387 lr:5e-05
==>Test Epoch:24 Loss:7.7304
==>Test Acc:(array([0.25865403]), array([0.21268633]), array([0.10590657]))
epoch 24 cost time: 1113.8818717002869
==>Train Epoch:25 Loss:6.8455 lr:5e-05
==>Test Epoch:25 Loss:7.6760
==>Test Acc:(array([0.26882056]), array([0.22285287]), array([0.11145762]))
epoch 25 cost time: 1113.959002494812
==>Train Epoch:26 Loss:6.7596 lr:5e-05
==>Test Epoch:26 Loss:7.6074
==>Test Acc:(array([0.2817938]), array([0.2393189]), array([0.12062621]))
epoch 26 cost time: 1119.7666964530945
==>Train Epoch:27 Loss:6.6577 lr:5e-05
==>Test Epoch:27 Loss:7.5322
==>Test Acc:(array([0.29526601]), array([0.25247926]), array([0.12648912]))
epoch 27 cost time: 1115.2167870998383
==>Train Epoch:28 Loss:6.5777 lr:5e-05
```



```
Windows PowerShell
==>Test Acc:(array([0.04365995]), array([0.02906505]), array([0.0093557]))
epoch 7 cost time: 1068.9676840305328
==>Train Epoch:08 Loss:8.3856 lr:5e-05
==>Test Epoch:08 Loss:8.7681
==>Test Acc:(array([0.04621718]), array([0.03068671]), array([0.01035365]))
epoch 8 cost time: 1096.8447370529175
==>Train Epoch:09 Loss:8.3174 lr:5e-05
==>Test Epoch:09 Loss:8.7130
==>Test Acc:(array([0.05151874]), array([0.03561405]), array([0.01303561]))
epoch 9 cost time: 1094.7562763690948
==>Train Epoch:10 Loss:8.2355 lr:5e-05
==>Test Epoch:10 Loss:8.6928
==>Test Acc:(array([0.05457494]), array([0.03661199]), array([0.01428304]))
epoch 10 cost time: 1100.9343838691711
==>Train Epoch:11 Loss:8.1519 lr:5e-05
==>Test Epoch:11 Loss:8.6044
==>Test Acc:(array([0.06630075]), array([0.04553109]), array([0.01615418]))
epoch 11 cost time: 1101.078446149826
==>Train Epoch:12 Loss:8.0654 lr:5e-05
==>Test Epoch:12 Loss:8.5525
==>Test Acc:(array([0.07391006]), array([0.05108214]), array([0.01877378]))
epoch 12 cost time: 1095.2416679859161
==>Train Epoch:13 Loss:7.9766 lr:5e-05
==>Test Epoch:13 Loss:8.5141
==>Test Acc:(array([0.0784008]), array([0.05501154]), array([0.02089441]))
epoch 13 cost time: 1108.0760517120361
==>Train Epoch:14 Loss:7.8813 lr:5e-05
==>Test Epoch:14 Loss:8.4395
==>Test Acc:(array([0.08962764]), array([0.0616229]), array([0.02357637]))
epoch 14 cost time: 1089.1291749477386
==>Train Epoch:15 Loss:7.7906 lr:5e-05
==>Test Epoch:15 Loss:8.3388
==>Test Acc:(array([0.12224786]), array([0.08420133]), array([0.03648725]))
epoch 15 cost time: 1096.8967700004578
==>Train Epoch:16 Loss:7.7008 lr:5e-05
==>Test Epoch:16 Loss:8.2989
==>Test Acc:(array([0.12268446]), array([0.08576062]), array([0.03904447]))
epoch 16 cost time: 1108.3177390098572
==>Train Epoch:17 Loss:7.6039 lr:5e-05
==>Test Epoch:17 Loss:8.2166
==>Test Acc:(array([0.14757064]), array([0.10653028]), array([0.04428366]))
epoch 17 cost time: 1107.377527475357
```

Fig 7. Accuracy at different epoch stages

```
epoch 42 cost time: 1132.2770342826843
==>Train Epoch:43 Loss:5.3038 lr:5e-05
==>Test Epoch:43 Loss:6.7844
==>Test Acc:(array([0.38707665]), array([0.32545375]), array([0.15062683]))
epoch 43 cost time: 1129.1004071235657
==>Train Epoch:44 Loss:5.2144 lr:5e-05
==>Test Epoch:44 Loss:6.7626
==>Test Acc:(array([0.383272]), array([0.32046404]), array([0.14987838]))
epoch 44 cost time: 1241.452297449112
==>Train Epoch:45 Loss:5.1522 lr:5e-05
==>Test Epoch:45 Loss:6.7399
==>Test Acc:(array([0.38557974]), array([0.32376972]), array([0.15000312]))
epoch 45 cost time: 1154.7386920452118
==>Train Epoch:46 Loss:5.0901 lr:5e-05
==>Test Epoch:46 Loss:6.6814
==>Test Acc:(array([0.39169213]), array([0.32844758]), array([0.15075157]))
epoch 46 cost time: 1208.847817659378
==>Train Epoch:47 Loss:4.9955 lr:5e-05
==>Test Epoch:47 Loss:6.6710
==>Test Acc:(array([0.39487307]), array([0.33119192]), array([0.15087632]))
epoch 47 cost time: 1563.9142549037933
==>Train Epoch:48 Loss:4.9513 lr:5e-05
==>Test Epoch:48 Loss:6.6480
==>Test Acc:(array([0.39711844]), array([0.33112955]), array([0.15093869]))
epoch 48 cost time: 1404.1045117378235
==>Train Epoch:49 Loss:4.8628 lr:5e-05
==>Test Epoch:49 Loss:6.6385
==>Test Acc:(array([0.39593339]), array([0.32944552]), array([0.1515624]))
epoch 49 cost time: 1171.910620212555
ours_acc:[0.1515624]
PS D:\2023 Download(4th yr)\Major Project\GCDAN-master\codes>
```

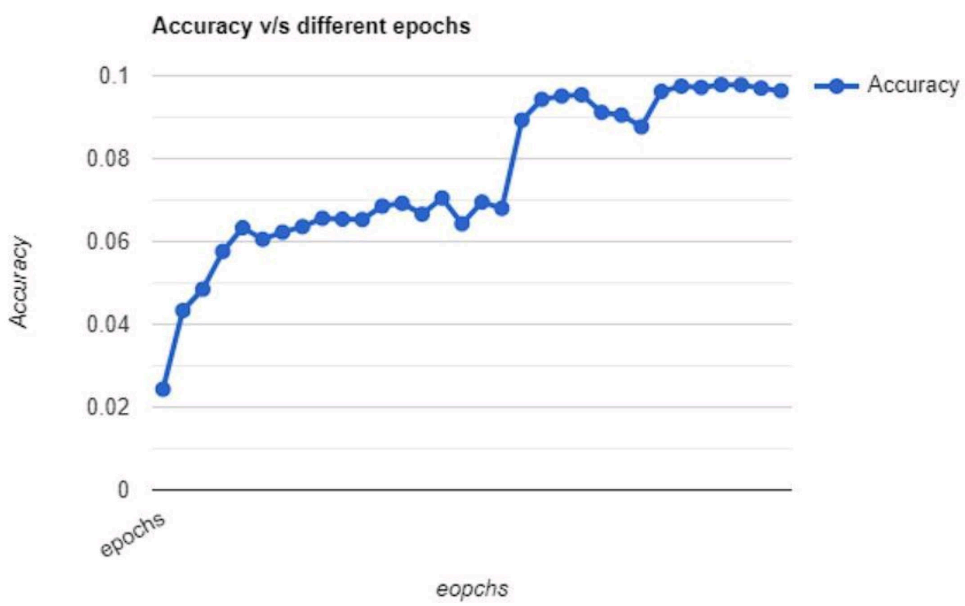


Fig 8. Accuracy Vs Epochs

4.7 Pretain Vs New Train

The results of Pretrained Model which we compared with :

model_in_code	model_in_paper	top-1 accuracy (pre-trained)
markov	markov	0.082
simple	RNN-short	0.096
simple_long	RNN-long	0.118
attn_avg_long_user	Ours attn-1	0.133
attn_local_long	Ours attn-2	0.145

Table 2: Neural Network Models V/s accuracy

Link : DeepMove

- Accuracy of Transformer model :

Fig 9. Final accuracy achieved

4.8 Comparison(BaseLine Method [14])

- Comparisons in between :

  [1] Recurrent Neural Networks(RNN)

  [2] Deep Move(2018) - Hybrid model which is a mixture of RNN and attentional neural network.

  [3] Transformer encoder-decoder model.

• Evaluation Metric - We define accuracy @1 as the function of records whose corresponding locations are involved in the top - k locations given by the prediction.

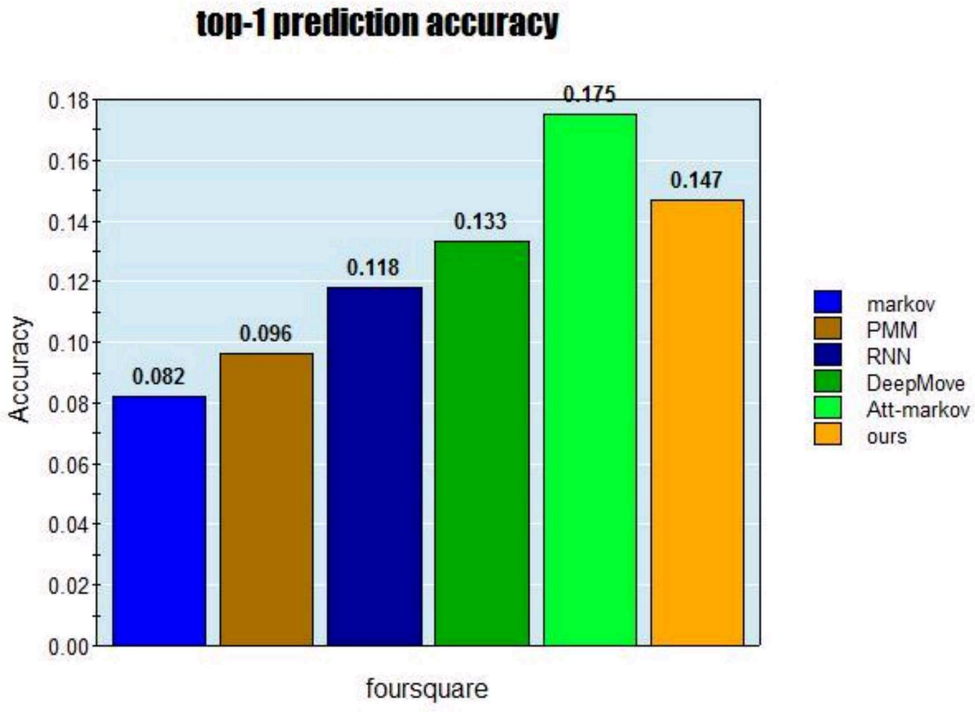


Fig 10. Accuracy v/s models graph

Chapter 5

Conclusion and Future Work

With the model in paper we are able to achieve an accuracy of 0.152 on transformer encoder-decoder model which is trained on foursquare dataset and geolife dataset.

We are able to achieve higher accuracy than the proposed model and from our previous work implementation of DeepMove by optimally choosing optimizer as Adam, activation function as log softmax, learning rate of 0.005 and max epochs of 50. We plan to incorporate a better algorithm/model based on Networking and Cryptography to further improve the models accuracy and introduce the second half of the paper with OPTIMIZATION OF ABSS PLACEMENT.

References

[1] C. V. N. Index, “Cisco annual internet report (2018–2023) white paper,” White Paper, Mar. 2020.

[2] F. Lagum, I. Bor-Yaliniz, and H. Yanikomeroglu, “Strategic densification with UAV-BSs in cellular networks,” IEEE Wireless Commun. Lett., vol. 7, no. 3, pp. 384–387, Jun. 2018.

[3] A. A. Khuwaja, Y. Chen, N. Zhao, M.-S. Alouini, and P. Dobbins, “A survey of channel modeling for UAV communications,” IEEE Commun. Surv. Tut., vol. 20, no. 4, pp. 2804–2821, Oct.–Dec. 2018.

[4] N. Zhao et al., “UAV-assisted emergency networks in disasters,” IEEE Wireless Commun., vol. 26, no. 1, pp. 45–51, Feb. 2019.

[5] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, “A tutorial on UAVs for wireless networks: Applications, challenges, and open problems,” IEEE Commun. Surv. Tut., vol. 21, no. 3, pp. 2334–2360, Jul.–Sep. 2019.

[6] A. Fotouhi et al., “Survey on UAV cellular communications: Practical aspects, standardization advancements, regulation, and security challenges,” IEEE Commun. Surv. Tut., vol. 21, no. 4, pp. 3417–3442, Oct.–Dec. 2019.

[7] D. W. Matolak and R. Sun, “Air-ground channel characterization for unmanned aircraft systems–Part III: The suburban and near-urban environments,” IEEE Trans. Veh. Technol., vol. 66, no. 8, pp. 6607–6618, Aug. 2017.

[8] Y. Zeng, X. Xu, and R. Zhang, “Trajectory design for completion time minimization in UAV-enabled multicasting,” IEEE Trans. Wireless Commun., vol. 17, no. 4, pp. 2233–2246, Apr. 2018.

[9] H. Huang and A. V. Savkin, “An algorithm of efficient proactive placement of autonomous drones for maximum coverage in cellular networks,” IEEE Wireless Commun. Lett., vol. 7, no. 6, pp. 994–997, Dec. 2018.

[10] C. Caillouet, F. Giroire, and T. Razafindralambo, “Optimization of mobile sensor coverage with UAVs,” in Proc. IEEE Conf. Comput. Commun. Workshops, 2018, pp. 622–627.

[11] M. Mozaffari, A. T. Z. Kasgari, W. Saad, M. Bennis, and M. Debbah, “3D cellular network architecture with drones for beyond 5G,” in Proc. IEEE Glob. Commun. Conf., 2018, pp. 1–6.

[12] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Drone small cells in the clouds: Design, deployment and performance analysis,” in Proc. IEEE Glob. Commun. Conf., 2015, pp. 1–6.

[13] E. Kalantari, M. Z. Shakir, H. Yanikomeroglu, and A. Yongacoglu, “Backhaul-aware robust 3D drone placement in 5G wireless networks,” in Proc. IEEE Int. Conf. Commun. Workshops, 2017, pp. 109–114.

[14] R. I. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, “Efficient 3-D placement of an aerial base station in next generation cellular networks,” in Proc. IEEE Int. Conf. Commun., 2016, pp. 1–5.

[15] M. Alzenad, A. El-Keyi, F. Lagum, and H. Yanikomeroglu, “3-D placement of an unmanned aerial vehicle base station (UAV-BS) for energyefficient maximal coverage,” IEEE Wireless Commun. Lett., vol. 6, no. 4, pp. 434–437, Aug. 2017.

[16] E. Chaalal, L. Reynaud, and S. M. Senouci, “A social spider optimisation algorithm for 3D unmanned aerial base stations placement,” in Proc. IFIP Netw. Conf., 2020, pp. 544–548.

[17] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," Nature, vol. 453, no. 7196, pp. 779–782, 2008.

[18] J. Chen and D. Gesbert, "Optimal positioning of flying relays for wireless networks: A los map approach," in Proc. IEEE Int. Conf. Commun., 2017, pp. 1–6.

[19] Z. Becvar, M. Vondra, P. Mach, J. Plachy, and D. Gesbert, "Performance of mobile networks with UAVs: Can flying base stations substitute ultra-dense small cells?," in Proc. 23th Eur. Wireless Conf., 2017, pp. 1–7.

[20] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," IEEE Wireless Commun. Lett., vol. 3, no. 6, pp. 569–572, Dec. 2014.

[21] P. Zhan, K. Yu, and A. L. Swindlehurst, "Wireless relay communications using ANN unmanned aerial vehicle," in Proc. IEEE 7th Workshop Signal Process. Adv. Wireless Commun., 2006, pp. 1–5.

[22] E. P. De Freitas et al., "UAV relay network to support wsn connectivity," in Proc. IEEE Int. Congr. Ultra Modern Telecommun. Control Syst., 2010, pp. 309–314.

[23] D. Orfanus, E. P. de Freitas, and F. Eliassen, "Self-organization as a support-ing paradigm for military UAV relay networks," IEEE Commun. Lett., vol. 20, no. 4, pp. 804–807, Apr. 2016.

[24] J. Korhonen, "Enhanced lte support for aerial vehicles," 3rd Generation Part-nership Project (3GPP), Tech. Rep. 36.777, 2018.

[25] Y. Zheng, Y. Wang, and F. Meng, "Modeling and simulation of pathloss and fading for air-ground link of HAPs within a network simulator," in Proc. IEEE Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discov., 2013, pp. 421–426.

[26] J. Holis and P. Pechac, "Elevation dependent shadowing model for mobile communications via high altitude platforms in built-up areas," IEEE Trans. Anten-nas Propag., vol. 56, no. 4, pp. 1078–1084, Apr. 2008.

[27] Q. Feng, E. K. Tameh, A. R. Nix, and J. McGeehan, “WLCp2-06: Modelling the likelihood of line-of-sight for air-to-ground radio propagation in urban environ-ments,” in Proc. IEEE Globecom, 2006, pp. 1–5.

[28] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, “Modeling air-toground path loss for low altitude platforms in urban environments,” in Proc. IEEE Glob. Commun. Conf., 2014, pp. 2898–2904.

[29] W. Khawaja, I. Guvenc, D. W. Matolak, U.-C. Fiebig, and N. Schnecken-burger, “A survey of air-to-ground propagation channel modeling for unmanned aerial vehicles,” IEEE Commun. Surv. Tut., vol. 21, no. 3, pp. 2361–2391, Jul.–Sep. 2019.

[30] Y. Zeng, R. Zhang, and T. J. Lim, “Wireless communications with unmanned aerial vehicles: Opportunities and challenges,” IEEE Commun. Mag., vol. 54, no. 5, pp. 36–42, May 2016.

[31] F. Jiang and A. L. Swindlehurst, “Optimization of UAV heading for the ground-to-air uplink,” IEEE J. Sel. Areas Commun., vol. 30, no. 5, pp. 993–1005, Jun. 2012.

[32] G. Zhang, Q. Wu, M. Cui, and R. Zhang, “Securing UAV communications via trajectory optimization,” in Proc. IEEE Glob. Commun. Conf., 2017, pp. 1–6.

[33] https://github.com/GCDAN/GCDAN/tree/master

[34] https://www.youtube.com/watch?v=zxQyTK8quyY&list=PLblh5JKOoLUIxGDQs4LFFD--41Vzf-index=21&ab_channel=StatQuestwithJoshStarmer

[35] https://pytorch.org/docs/stable/_modules/torch/nn/modules/transformer.html#Transformer

[36] https://ieeexplore.ieee.org/document/9413901