PERSONAL DETAILS

Name: Rajat Gupta

Course: PGDBA (Post Graduate Diploma in Business Analytics)

Course Website: https://www.isical.ac.in/~pgdba/Programme_Overview.html

Universities: IIM Calcutta, IIT Kharagpur, ISI Kolkata

Time Zone: Indian Standard Time (IST) – GMT + 5:30

GitHub: https://github.com/rajatguptakgp

LinkedIn: https://www.linkedin.com/in/rajatguptakgp

Contact Details:

1. **Primary email-id:** rajatgba2021@email.iimcal.ac.in
2. **Secondary email-id:** rajatgupta069@gmail.com

BIOGRAPHICAL INFORMATION

Hi, I am Rajat Gupta, student of the PGDBA program jointly offered by IIM Calcutta, IIT Kharagpur and ISI Kolkata. I am currently interning at **Amazon** under the role of **Data Scientist II**. My internship started on **4th January 2021** and will end on **4th June 2021**.

I have been a part of variety of projects in the fields of **Computer Vision**, **NLP**, and **Reinforcement Learning** among others that I have done in the duration of course, as can be seen in my resume. I have a strong knowledge of **Python, PyTorch/Tensorflow/Keras** and machine learning algorithms from Linear/Logistic Regression to ensemble methods like bagging and boosting to advanced neural network frameworks like Autoencoders and GANs. I am also an enthusiastic participant in Data Science competitions and quizzes. Last year, I stood **1st** among **2035 aspirants** of Data Science Quiz, conducted by Analytics Club, IIT Guwahati.

I have been part of three remote academic internships during the course. Last year, I worked under my two professors from **Computer Science & Engineering Department, IIT Kharagpur** for two projects for a duration of four months from April '20 till July '20. The two projects that I had worked on were:

1. **Generating explainable recommendations**, where I built a machine learning framework using graphs (Python library used: **Networkx**)
2. **Building knowledge-aware chatbots:** Transformer framework with problem modelled as Reinforcement Learning

I am also an active open-source contributor. I maintain my code works of all the projects and learnings in my GitHub repository. I like to code machine learning algorithms from scratch to have a deeper knowledge of algorithms and to be an efficient coder.

I want to participate in GSOC '21 as this is my last opportunity to be a part of something so unique and contributing to open-source. In order to showcase my skills, sincerity and dedication, and to increase chances of selection, I have worked on evaluation tasks of **six** projects offered by ML4SCI organization and as listed below and **all of my three proposals** are for these projects:

1. Deep Regression Techniques for Decoding Dark Matter with Strong Gravitational Lensing
2. Domain Adaptation for Decoding Dark Matter with Strong Gravitational Lensing
3. Equivariant Neural Networks for Dark Matter Morphology with Strong Gravitational Lensing
4. Machine Learning for Turbulent Fluid Dynamics
5. Decoding quantum states through nuclear magnetic resonance
6. Dimensionality Reduction for Studying Diffuse Circumgalactic Medium

SYNOPSIS

PROJECT PROPOSAL 1: Deep Regression Techniques for Decoding Dark Matter with Strong Gravitational Lensing

This project focuses on building a deep learning framework to understand the structure of dark matter and estimating its properties. We start with building a deep regression framework using **Multi-Layer Perceptron (MLP)** followed by improving model performance with hyperparameter tuning and ensuring that the model does not overfit. Next, we explore and build advanced deep regression frameworks like **Radial Basis Function Neural Networks (RBNN)** and **Generalized Regression Neural Networks (GRNN).** Finally, we compare the metrics and conclude at the best model.

PROJECT PROPOSAL 2: Equivariant Neural Networks for Dark Matter Morphology with Strong Gravitational Lensing

In order to better understand the distribution of substructure of dark matter, we will explore equivariant neural networks to distinguish images with and without sub-structure. Equivariant neural networks are networks which are invariant to data transformations like translation, rotation and permutations. For example - CNNs are translationally equivariant.

We will start with building a base model using **Multi-Layer Perceptron (MLP)** followed by improving model performance with hyperparameter tuning and ensuring no overfitting. Next, we explore and build advanced equivariant neural networks like **Semi-Supervised GANs (SGAN)** and **autoencoders**. Finally, we compare the metrics and conclude at the best model.

PROJECT PROPOSAL 3: Domain Adaptation for Decoding Dark Matter with Strong Gravitational Lensing

This project entails building domain adaptation models and learning distribution of lensing images with/without substructure for use cases like anomaly detection. We start the project by training and evaluating generated simulated images on **all models (VAE, AAE, DCAE, RBM)** of library **unsupervised-lensing** and comparing judging metrics, followed by hyperparameter tuning the frameworks to achieve best model performances.

Next, we leverage **transfer learning** with famous CNN architectures like **VGG, ResNet and AlexNet** and fine-tune the models based on our dataset. Finally, we compare the metrics across all seven architectures (4 + 3) and conclude with the best model.

PROJECT PROPOSAL 1: Deep Regression Techniques for Decoding Dark Matter with Strong Gravitational Lensing

PROJECT MILESTONES + DELIVERABLES

Week Number	Dates	Milestones	Deliverable (apart from Weekly Report)
Week 1	June 7 – June 14	Simulations in PyAutoLens	
Week 2	June 14 – June 21		**Dataset** to be used for analysis
Week 3	June 21 – June 28	Exploring deep regression frameworks	
Week 4	June 28 – July 5	Building a base deep regression framework and getting judging metrics (MSE)	
Week 5	July 5 – July 12		**Full code-base** with metrics calculated on dataset
Week 6	July 12 – July 19	**Mid - Evaluation**	**Presentation + Documentation of work done in past 5 weeks**
Week 7	July 19 – July 26	Improving upon base metrics - Hyperparameter tuning + Model Evaluation (Overfitting/Underfitting)	
Week 8	July 26 – August 2	**Building advanced deep regression frameworks** + comparing judging metrics	Literature Survey of advanced frameworks – **RBNN (Radial Basis Function NN) and GRNN (General Regression NN)**
Week 9	August 2 – August 9		Building two architectures: **RBNN, GRNN**
Week 10	August 9 – August 16	Analysis wrap-up	Model Evaluation, **Code Base** + **Documentation of work done in last 4 weeks**
Week 11	August 16 – August 23	**Code Submission + Final Evaluation**	Final presentation

Deliverables:

1. I will be submitting a weekly report to give updates about the work that I have done for the week. This report can go up to 3-4 pages depending on the level of detail. This report will help me align with the objectives of the project and will also give a clear idea about my update to my mentor.
2. Apart from the weekly report, I intend to deliver the following:
   a. Dataset to be used for analysis: Using Simulations by PyAutoLens
   b. Full-code base for the base model of deep regression neural network + Documentation of work done till Mid-Evaluation

c. Full-code base for approaches that I plan to follow for building advanced deep regression neural networks, with the guidance of my mentor: Radial Basis Function Neural Network (RBNN) and General Regression Neural Network (GRNN).

d. Documentation for the complete project

PROJECT OUTLINE

Insights from the test: For this task, I had used a Multi-Layer Perceptron and the results that I got are as follows:

Metric	**Value**
Validation MSE	2.469
Test MSE	2.449
Test R2	0.9998

We can notice that since **there isn't much difference between the Validation MSE and Test MSE,** which might imply that the model is not overfitting, however we cannot be sure of it since MSE does not have a fixed range unlike R2.

And so, if we look at Test R2, we realize that it is very close to 1, which means that there exists a possibility that the **model might be overfitting**.

And so, my approach for the project is going to be as follows:

1. After running simulations using PyAutoLens and generating the dataset, **I would like to start my project by understanding whether my approach leads to an over-fitted model or not,** and if that's really the case, take appropriate measures to reduce overfitting.
2. Next, I would like to build a base deep regression neural network (MLP) on the generated set of simulated images and look at judging metrics. This shall be my base-model which I intend to complete in my mid-evaluation.
3. Next, in order to improve the base model and ensure that the model is not overfitting, I would like to do hyperparameter tuning. Common hyper parameters/techniques for a neural network to improve performance as described above are:
   a. Learning Rate
   b. Early Stopping criteria
   c. Dropout
   d. Batch Normalization
4. Next, in order to improve upon the base model, I would like to pursue two advanced neural network frameworks under the guidance of my mentor –
   a. **Radial Basis Function Neural Network (RBNN)**: Structurally similar to MLP, with exactly one hidden layer. **It increases dimension of feature vector (hidden layer) using transformations to get linear separability** – a concept very similar to what **kernels** do in **Support Vector Machines (SVMs).**
   b. **Generalized Regression Neural Network (GRNN)**: A variation of RBNN and an improved technique based on non-parametric regression. However, it is computationally expensive.
5. Comparing results of all the three models – **MLP, RBNN and GRNN.** It would be interesting to understand their metrics since these three frameworks are quite similar to each other.

PROJECT PROPOSAL 2: Equivariant Neural Networks for Dark Matter Morphology with Strong Gravitational Lensing

PROJECT MILESTONES + DELIVERABLES

Week Number	**Dates**	**Milestones**	**Deliverable (apart from Weekly Report)**
Week 1	June 7 – June 14	Simulations in PyAutoLens	
Week 2	June 14 – June 21		**Dataset** to be used for analysis
Week 3	June 21 – June 28	Understanding Equivariant Neural Networks	
Week 4	June 28 – July 5	Building a base equivariant neural network (MLP) and getting judging metrics (AUROC)	
Week 5	July 5 – July 12		**Full code-base** with metrics calculated on dataset
Week 6	July 12 – July 19	**Mid - Evaluation**	**Presentation + Documentation of work done in past 5 weeks**
Week 7	July 19 – July 26	Improving upon base metrics - Hyperparameter tuning + Model Evaluation (Overfitting/Underfitting)	
Week 8	July 26 – August 2	**Building advanced equivariant neural networks** + comparing judging metrics (AUROC) with base model	**Literature Survey** of advanced neural networks - Semi-Supervised GANs (SGAN) + Autoencoders
Week 9	August 2 – August 9		Building two architectures: Autoencoders, SGAN
Week 10	August 9 – August 16	Model Evaluation for Autoencoders, SGAN + Analysis wrap-up	**Code Base** + **Documentation of work done in last 4 weeks**
Week 11	August 16 – August 23	**Code Submission + Final Evaluation**	Final presentation

I plan to deliver the following during the course of project:

1. I will be submitting a weekly report to give updates about the work that I have done for the week. This report can go up to 3-4 pages depending on the level of detail. This report will help me align with the objectives of the project and will also give a clear idea about my update to my mentor.
2. Apart from the weekly report, I intend to deliver the following:
    a. Dataset to be used for analysis: Using Simulations by PyAutoLens
    b. Full-code base for the base model of equivariant neural network + Documentation of work done till Mid-Evaluation
    c. Presentation + Documentation for mid-evaluation

d. Full-code base for approaches that I plan to follow for building equivariant neural networks, with the guidance of my mentor: Autoencoder + Semi-Supervised GANs (SGAN).

e. Presentation for final evaluation + documentation for the complete project

PROJECT OUTLINE

Insights from the evaluation test: In this task, I had modelled a Multi-Layer Perceptron (MLP) and the results that I got are as follows:

Metric	Value
Validation Accuracy	96.83%
Test Accuracy	97.3%
Test AUROC	0.973

We can notice that since **there isn't much difference between the Validation Accuracy and Test Accuracy,** this might imply that the model is not overfitting.

And so, my approach for the project is going to be as follows:

1. After running simulations using PyAutoLens and generating the dataset, **I would like to start my project by diving down** and understand if that's really the case **(model overfitting)**, and take appropriate measures to reduce overfitting.
2. Next, I would like to look at possible approaches of building equivariant neural networks under the guidance of my mentor. **Just like CNNs are transitionally equivariant, I would like to build networks which are rotationally equivariant as well.** In that case, one can perform data augmentation techniques like rotation on original set of images to generate a larger set of images and check if there is an improvement in judging metrics. This shall be my base model which I intend to complete in my mid-evaluation.
3. In order to improve the base model and ensure that the model is not overfitting, I would like to do hyperparameter tuning. Common hyper parameters for a neural network are:

Hyperparameter	Description
Learning Rate	Using adaptive learning rates to reach extrema
Early Stopping	Early stop the training to prevent too much of learning and hence overfitting
Dropout	Randomly turning of neurons to make model robust

Apart from this Batch Normalization, which is normalizing model parameters using batch statistics for faster convergence and inducing regularization can help prevent overfitting.

4. In order to improve the base model, I would like to explore advanced architectures as follows:
   a. **Autoencoders:** Autoencoders are unsupervised neural networks which aim to learn the latent representation of data and are famous for anomaly detection and rare event classification. If our use case matches with it, I would like to explore them.
   b. **Semi-Supervised GANs:** SGANs will enable us to learn a classifier as well as distribution of images to be able to generate them from the generator model.

PROJECT PROPOSAL 3: Domain Adaptation for Decoding Dark Matter with Strong Gravitational Lensing

PROJECT MILESTONES + DELIVERABLES

Week Number	**Dates**	**Milestones**	**Deliverable (apart from Weekly Report)**
Week 1	June 7 – June 14	Simulations in PyAutoLens	
Week 2	June 14 – June 21		**Dataset** to be used for analysis
Week 3	June 21 – June 28	Exploring unsupervised frameworks for Anomaly Detection	
Week 4	June 28 – July 5	Comparing judging metrics with existing frameworks (Library: **unsupervised-lensing**) on simulated data	
Week 5	July 5 – July 12		**Full code-base** with detailed summary of metrics calculated on dataset
Week 6	July 12 – July 19	**Mid - Evaluation**	**Presentation + Documentation of work done in past 5 weeks**
Week 7	July 19 – July 26	Improving upon base metrics - Hyperparameter tuning + Model Evaluation (Overfitting/Underfitting)	
Week 8	July 26 – August 2	Leveraging **Transfer Learning**	Exploring architectures like ResNet, VGG, AlexNet
Week 9	August 2 – August 9		Fine - tuning model performance on above architectures for our dataset
Week 10	August 9 – August 16	Model evaluation + Analysis wrap-up	**Code Base + Documentation of work done in last 4 weeks**
Week 11	August 16 – August 23	**Code Submission + Final Evaluation**	Final presentation

I plan to deliver the following during the course of project:

1. I will be submitting a weekly report to give updates about the work that I have done for the week. This report can go up to 3-4 pages depending on the level of detail. This report will help me align with the objectives of the project and will also give a clear idea about my update to my mentor.
2. Apart from the weekly report, I intend to deliver the following:
    a. Dataset to be used for analysis: Using Simulations by PyAutoLens

b. Full-code base for all existing frameworks trained and tested on generated simulated data + **detailed summary of metrics in one table +** Documentation of work done till Mid-Evaluation

The summary table highlighting model performances should look as follows:

Frameworks from library **unsupervised-lensing**	**Train AUROC**	**TEST AUROC**
AAE (Adversarial Autoencoder)	---	---
VAE (Variational Autoencoder)	---	---
DCAE (Deep Convolutional Autoencoder)	---	---
RBM (Restricted Boltzmann Machine)	---	---

c. Full-code base for approaches leveraging **transfer learning** – fine-tuning architectures on our dataset and comparison with previous metrics.

The summary table highlighting model performances should look as follows:

Comparison with existing frameworks	**Train AUROC**	**TEST AUROC**
ResNet	---	---
VGG	---	---
AlexNet	---	---
AAE (Adversarial Autoencoder)	---	---
VAE (Variational Autoencoder)	---	---
DCAE (Deep Convolutional Autoencoder)	---	---
RBM (Restricted Boltzmann Machine)	---	---

d. Presentation + Documentation for the complete project

PROJECT OUTLINE

Insights from the evaluation test: In this task, I had used Adversarial Autoencoder (AAE) framework from the library **unsupervised-lensing** to train the model on given dataset leveraging transfer learning. I had trained the model to learn distribution of lensing images with no substructure.

The results that I got are as follows:

Metric	**Value**
Mean reconstruction loss for images with no substructure	4.26E-5
Mean reconstruction loss for images with substructure	2.8E-4
Test AUROC	0.853

We can notice that the mean reconstruction loss for images with substructure (anomalies) is **~6.76** times mean reconstruction loss for images with no substructure. Also, the test AUROC is a decent number to start with.

Hence, we can conclude that the **approach seems promising** and it should be interesting to do the **same analysis across other existing frameworks (VAE, DCAE, RBM)** in the unsupervised-lensing library, and compare the metrics.

And so, my approach for the project is going to be as follows:

1. After running simulations using PyAutoLens and generating the dataset, **I would like to start my project** by training and evaluating all models in unsupervised-lensing library and compare the metrics.
2. Next, with the code base laid out, I would like to **hyperparameter tune the architectures** to **achieve their best performances** while ensuring that the models do not overfit by comparing Train and Test ROC Curves, and create the **metric summary table** as explained above. I intend to complete this in my mid-evaluation.
3. Next, I would like to leverage **transfer learning** with famous CNN architectures like VGG, ResNet and AlexNet, and fine-tune the models for our dataset.
4. With achieving best model performances across these architectures, I will create the final summary table listing all deep learning frameworks evaluated on the dataset. It would be interesting to understand comparison of judging metrics of these architectures with previously discussed architectures.

RELATED WORK

Unsupervised-lensing: https://pypi.org/project/unsupervised-lensing/

RBNN: https://pythonmachinelearning.pro/using-neural-networks-for-regression-radial-basis-function-networks/

GRNN: https://www.dtreg.com/solution/probabilistic-and-general-regression-neural-networks

SGAN: https://towardsdatascience.com/semi-supervised-learning-with-gans-9f3cb128c5e