Google Summer of Code

Deep Regression Techniques for Decoding Dark Matter with Strong Gravitational Lensing

Organisation

ML4SCI

Duration

175 hours

Mentors

Michael Toomey, Stephon Alexander, Brandon Ames, Sanaz Kiyadeh, Yurii Halychanskyi, Saranga Mahanta, Karthik Sachdev

By

Alisetti Sai Vamsi

Contents

Basic Information

Name	Alisetti Sai Vamsi
University	Indian Institute of Technology (IIT) Palakkad
Field of Study (Major)	Computer Science & Engineering (2018 - 2022)
Email	saivamsi.ds123@gmail.com alisetti.vamsi@gmail.com
GitHub	Vamsi995
LinkedIn	https://www.linkedin.com/in/sai-vamsi-alisetti-4892a4197/
Phone Number	+91 7995939741
Country/Region/Timezone	Telangana, India, Indian Standard Time (GMT+5:30)
Resume	AlisettiSaiVamsi_Resume.pdf
Profession	Software Development Engineer @ Airasia

About Me

I am a fresh graduate from IIT Palakkad majoring in Computer Science & Engineering with a potent blend of industry and research experiences. I am interested in problem-solving be it in a social aspect or analytical aspect, and nothing brings me more joy than solving something that is on the cutting edge of research and on the cusp of being novel. I draw my passion from my deep interest in applied mathematics, and how math can disentangle convoluted symmetries and abstract patterns in nature through simple equations.

According to my philosophy, the reason why we humans developed into a civilisation and have continued to do so is because of cooperation. Everyone on the globe makes some kind of contribution to our survival, which may seem strange. In my instance, my contribution suffuses in the form of quality code. And because of this, I find myself pulled to the open-source programming community, where programmers from all across the world work together.

I have an immense amount of interest in machine learning and astronomy. Although most of my astronomy knowledge stems from youtube videos, I have a good amount of understanding of the technical jargon. Apart from reading books and programming, I do enjoy playing football, chess, video games, and making pencil sketches.

Project Details

Overview

Galaxies in our universe seem to rotate at very high speeds that the gravitational force generated by the matter inside the galaxy is not enough to hold the galaxy together. In theory, they should rip themselves apart because of the immense centrifugal force that counteracts gravity. But the fact that these galaxies are intact gives us a hint that there is some extra mass that is aiding the gravitational force but is invisible. The visible matter in the universe only forms a very small fraction of all the matter in the universe. There is about six times more matter than what we can see, but except it is hidden/invisible. This invisible matter is called dark matter. Dark matter does not interact with electromagnetic fields, i.e it does not absorb, reflect, or emit electromagnetic radiation. Hence it is difficult to detect but its only interaction with the visible matter is through gravity. Strong galaxy-galaxy lensing provides strong evidence of dark matter because most of the light is lensed through the space distorted due to the dark matter halos surrounding the lensing galaxy. The way in which these dark matter halos lens the source galaxies provides us means to study the properties of dark matter and possible ways of decoding the structure inside these dark matter halos.

The volume and complexity of data in astronomy are constantly increasing, making conventional algorithms for data analysis very difficult. The rapid advancement of machine learning and advanced deep learning techniques allows us to solve these problems in a variety of ways.

Regression analysis in machine learning is a tool that is used for predicting a continuous output variable (y) based on the values of one or more dependent variables (x). There are several forms of regression possible out of which I am using polynomial regression, and hence the use of neural networks. These are function approximators in a higher dimension. Hence in the case of a dense neural network (ANN), the number of weight parameters are nothing but the dependent variables that will be adjusted in each backpropagation iteration to predict the correct output. The focus of this project is on using deep regression techniques for estimating dark matter properties, including population-level quantities (i.e learning the total mass of a halo).

The evaluation task that is done on the deep regression analysis which included learning the dark matter halo mass from the simulated images, has given me a deep insight as to what the project needs.

About Evaluation Task

GitHub: https://github.com/Vamsi995/DeepLense-ML4SCI-EvalTest

Research:

My first challenge was to understand the concept of dark matter. I learned about the possible candidates of dark matter i.e axion, WIMP, MACHOS, etc.., and how dark matter interacts with visible matter only through gravity and hence the use of lensing to observe its substructure. These two papers Decoding Dark Matter Substructure without Supervision and Deep Learning the Morphology of Dark Matter Substructure helped me understand the dark matter substructure in much more detail. The first paper explains about dark matter substructure and expected signatures from lensing images, and then it talks about the generation of the dataset using the PyAutoLens simulations (Table 1), and then presents two methods (supervised and unsupervised) which classify the data into dark matter halos having no substructure, subhalos as substructure, vortices as substructure. The supervised method includes using a ResNet18 for image classification (this is where I drew inspiration to use ResNet18 as a feature extractor) and the unsupervised method includes the usage of convolutional autoencoders. The novelty of the first paper is the unsupervised classification method which is compared with the supervised method for classification. The second paper talks about the theory of superfluidity in dark matter and also performs the same supervised analysis as the first paper on the classification of the substructure inside dark matter halos of lensing galaxies from lensing simulations. The third paper Domain Adaptation for Simulation-Based Dark Matter Searches Using Strong Gravitational Lensing however talks about the poor performance of these supervised methods naively applied on real world data as these methods are trained on simulated data. To overcome these shortcomings this paper talks about using UDA (unsupervised domain adaptation) algorithms that learn the substructure representation from a source dataset and identifies these representations in the target dataset. Unfortunately, because of lack of real data both these datasets were simulated with different complexities. The main takeaway of this paper is that it demonstrates that application of UDA algorithms increases the performance of all the algorithms on a much complex target dataset.

I have faced many issues while doing my evaluation task, especially while building the model architecture. For the first question the loss was not reducing at first, and upon further investigation it turns out that I was applying softmax layer in the model which I should not when using nn.CrossEntropy loss as it intrinsically applies a log softmax function to the labels. Another issue I faced was in the second question where the loss was not reducing further than expected. This was fixed when I applied a learning rate scheduler to peak in the middle of the training and gradually reduce the lr towards the end.

The following talk "Decoding Dark Matter Substructure without Supervision" Michael Toomey (Brown) - CFPU SMLI, explains the first paper in brief and helped me understand the bigger picture since all the pieces came together after watching this talk. From the evaluation test, I have understood how the dataset is structured and have already performed necessary data preprocessing steps to clean the data including data augmentations for improving the performance. The first and second question, which included training a deep learning model, gave me enough idea on how to transform the data to feed it into any deep learning network. Overall the evaluation test enhanced my skills to perform quality research and gave me an idea as to what the project needs and what is to be done.

Goals/Deliverables

1. Domain Based Feature Engineering

2. Research on optimal model tuning for extracting better regression performance

3. AlterNet Implementation/Analysis

4. EfficientNetV2 Implementation/Analysis

5. Integrating XResNet and AlterNet

6. Writing Blog Posts and articles for creating more awareness.

7. Aiming for a paper in this direction by making a custom deep learning architecture.

Approach/Methods

Note: This is a tentative outline of the final outcome, further refinements will be made during the project.

This project involves a significant research aspect. Therefore I also included some research time in between so that I can discuss with the mentors in detail. The amount of research studies on deep regression on vision tasks is quite limited and hence is a niche area. The papers that have been published on this topic are either very specific to the task or the dataset or they are very old. One of the only reliable and generalized sources that I could find is this paper A Comprehensive Analysis of Deep Regression which provides a detailed analysis of deep regression on only two selected models ResNet-50 and VGG-16.

Deep representational models are data hungry i.e they need a dataset size comparable to ImageNet so that the network weights saturate to an optimal value. But the dataset we are dealing with is comparably small and hence very deep models (especially vision transformer) performs relatively poor. To avert this issue, is to adapt our model to our domain i.e, transform the architecture according to the domain of the images that are being trained. We can perform some image processing techniques to extract better features from our images and tune our hyperparamters to extract the optimal performance from any model we train.

Taking inspiration from my evaluation task, I will be dividing the project into several parts from designing the data pipeline to model building and training. A typical workflow for a model would look as follows:

Model Workflow

- **Data Pipeline**
  - Analyzing Dataset Properties
    - Input/Output Analysis (Here in our case its images/masses)
    - Domain Based Feature Engineering for images
    - Normalizing input/output ranges to avoid weight saturation
  - PyTorch Adaptation
    - Custom Datasets
    - Custom Dataloaders
    - Stratified Data Sampling Split
    - Data Transformations/Augmentations
- **Model**
  - Model Definition
  - Loss Function
  - Optimizer
    - Learning Rate Scheduler
- **Training**
  - Regression Metrics
  - Training Strategies (Early Stoppoing, Checkpointing, ..etc)

A typical regression model would have a feature extractor part and a dense network attached at the end. For the feature extractor part, I have taken some SOTA CNN architectures that have performed well in the image classification tasks. The dense network takes takes in these CNN features and produces the regression output. We can follow a typical MLP for the dense network with regluarizaiton layers in between.

1. **Domain Based Feature Engineering**

   This is a research aspect that I want to discuss with the mentors briefly and get their opinions.

   The first step to any machine learning task is to understand the dataset thoroughly. I want to explore if any image processing modifications to the dataset can improve the regression metrics. I also want to play around with the data transformations/augmentations and see the most optimal choice of transformation for better performance. I will be following the given papers and blogs to enhance my knowledge on feature engineering.

   - https://www.frontiersin.org/articles/10.3389/fenrg.2021.652801/full
   - https://towardsdatascience.com/data-preprocessing-in-data-mining-machine-learning-79a9662e2eb
   - https://towardsdatascience.com/feature-engineering-for-machine-learning-with-picture-data-d7ff8554920
   - https://www.kaggle.com/code/lorinc/feature-extraction-from-images

   This task needs a bit more exploration, but at the time of writing this is all the relevant information I could procure related to this.

2. **Research on optimal model tuning for extracting better regression performance**

   Tuning hyperparameters is a crucial part of any deep learning model. The choice of optimizers and loss functions used can greatly affect the representations learnt by our model. There are some optimizations that i missed in my evaluation tasks that I will add here.

   Loss Functions:

   We can use a better loss function as compared to MSE and MAE since both these loss functions suffer from some drawbacks. MSE is very sensitve to outliers whereas MAE is not a smooth function. We can use Huber Loss Function which is a combination of MAE and MSE. Huber Loss function provides a smooth loss function that isnt very sensitive to outliers in the data. Pytorch already has an implementation of Huber Loss in the torch optim package. Huber loss function is piece-wise function and the transition is dependent on an input parameter (delta) to the loss function. This delta parameter adds another layer of complexity since tweaking this provides different loss curves. The choice of optimal delta value is something to be explored.

- How to choose delta parameter in Huber Loss?
- Pytorch Docs on Huber Loss
- Generalized Huber Loss for Robust Learning and its Efficient Minimization for a Robust Statistics

Regularization:

One problem that I faced during the training was the erratic values of validation loss as compared to the relatively smooth decent of training loss. This clearly indicates that the model is trying to overfit on the training data. One way to avoid this erratic validation loss is to apply regularization techniques.

L1/L2 Regularization:

L1/L2 regularization add a weight penalty term to the loss function so that the model loss focuses on reducing the loss criterion as well as makes sure that the weights dont reduce beyond a limit which avoids the model to overfit.

- https://icml.cc/Conferences/2004/proceedings/papers/354.pdf
- Pytorch L1 & L2 Regularization

Drop Out Layers:

Another method to avoid overfitting is by adding dropout layers in the network. The training algorithm uses a random subset of the network every iteration. This approach encourages neurons to learn useful features on their own without relying on other neurons.

- https://www.cs.toronto.edu/~rsalakhu/papers/srivastava14a.pdf
- Pytorch Dropout

Model Weight Initialization

The mean and variance of activations can suddenly rise to extremely high values or fall to zero because Neural Networks use a lot of matrix multiplications. As a result, our layers' local gradients will become NaN or zero and our network won't be able to learn anything. Using the most recent methods to initialise your network's weights is a popular method for preventing this. For instance, you must initialise your weights with Kaiming He initialization and set the biases to zero if you're utilising ReLU activation after a layer as per ImageNet paper. This makes sure that all layers' activations have a mean and standard deviation that are near to 0 and 1, respectively. Intitalizing the network with the right weights can make the difference between converging in a few epochs versus not converging at all.

- Weight Initialization Strategies
- WEIGHT INITIALIZATION TECHNIQUES FOR DEEP LEARNING ALGORITHMS
- Initializing Neural Networks

Learning Rate Finder

The learning rate finder, or more correctly the learning rate range finder, is a technique that Leslie Smith described in a study from Cyclical Learning Rates for Training Neural Networks. The paper introduces the idea of cyclical learning rates, or cycling repeatedly between learning rates between a predetermined minimum and a predetermined maximum, which has been shown to be effective for training. The minimum and maximum values to cycle through are determined by a function that Leslie Smith refers to as the "LR range test."

- Implementing a Learning Rate Finder from Scratch
- PyTorch learning rate finder

Optimizers

Deep Learning optimization involves minimizing a high-dimensional loss function in the weight space which is often perceived as difficult due to its inherent difficulties. There is no one generic optimizer that is good for all neural networks, it mainly depends on the loss landscape and learning rate. So I would like to make a catalouge of all the SOTA optimizers and plot the regression metrics for the same. I will be using the following optimizers from the paper A survey of deep learning optimizers-first and second order methods. From this I will be selecting the better performing optimizer for further developed models. I also want to try out the latest optimizer that is purported to be better than Adam class of optimizers called the LION optimizer.

- Symbolic Discovery of Optimization Algorithms

Learning Rate Schedulers

I would like to try out different learning rate schedulers and plot the regression metrics for the same, and find the best scheduler out of all. I also want to use the following technique of auto learning rate scheduler from the paper Automated Learning Rate Scheduler for Large-batch Training and adapt it to smaller batches for training.

- How to choose LR Scheduler?

Batch Normalization Layers

Typically batch normalization layers are added to accelerate the training process of neural networks. Batch normalization makes sure that the data is standardized at each layer. Because of its several drawbacks stated in this blog Curse of Batch Norm, I am willing to explore how the model performs in its absence and also introduce a new normalization layer called Proxy Normalization.

- Layer Normalization
- Group Normalization
- Instance Normalization

Batch Size Impact

I want to analyze the batch size impact on resnet like model networks. Luckily A Comprehensive Analysis of Deep Regression already puts out the impact of batch size on these structures for various datasets. I want to adapt this study to our dataset and provide further analysis on batch sizes.

- A DISCIPLINED APPROACH TO NEURAL NETWORK HYPER-PARAMETERS: PART 1 – LEARNING RATE, BATCH SIZE, MOMENTUM, AND WEIGHT DECAY

Activation Functions

As their name implies, activation functions are those that cause the neurons in any neural network to fire. These mathematical operations are connected to neurons and control whether the current neuron will fire or not (outputs 1). Based on whether the neuron's input is important for a model's prediction, activation functions take this action. To do this, it normalises any neuron's output between 0 and 1 or -1 and 1. (some exceptions are there). The neural network also gains non-linearity via activation functions. Hence its imperative to choose the right combination of activation functions for our network. I am willing to explore the following papers and blogs to find the best activation function for my model.

- Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark
- Activation Functions Compared With Experiments
- Neural Network composed of multiple activation functions

Deep Vs Wide Networks

How deep can we actually go, and is it worth it? This is something I want to explore with this task. I want to find the maximum depth where the network bottlenecks. I am also curious as to why not build wider networks instead of deep networks which have gradient vanishing problems. Below are the sources that I am curious of exploring.

- Do Wide and Deep Networks Learn the Same Things?
- Do Wide and Deep Networks Learn the Same Things? Uncovering How Neural Network Representations Vary with Width and Depth

3. AlterNet Implementation/Analysis

AlterNet is the latest innovation in CNN architectures. AlterNet proposes the use of MSA (Multi Head Self Attention) blocks as supposed to typical CNN layers. MSAs improve not only accuracy but also generalization by flattening the loss landscapes. The alternet design replaces Convolution blocks at the end of a stage with MSA blocks. AlterNet outperforms CNNs not only in large data regimes but also in small data regimes. These MSA layers can be adapted into our ResNet like models.

- How Do Vision Transformers Work?

4. EfficientNetV2 Implementation/Analysis

EfficientNetV2 introduces an efficient model architecture that requires 6.8x times lesser parameters to achieve the same SOTA results. EfficientNetV2 introduces new concepts like progressive learning and adaptive regluarization that aids in faster training times and better accuracies. I want to study this architecture in detail and adapt this model to our dataset.

- EfficientNetV2: Smaller Models and Faster Training

5. **Integrating XResnet Architecture with AlterNet**

    This is a new combination that I want to try out. I am not sure if it will result in better results. The idea is to introduce XResnet Architecture that was used earlier in DeepLense and combine it with AlterNet by introducing MSA blocks into XResNet and compare the results with the previous models. I will also add my learnings from EfficientNetV2 into this custom model construction.

    - Predicting galaxy spectra from images with hybrid convolutional neural networks (XResnet Paper)

6. **Writing Blog Posts and articles for creating more awareness**

    During the final phase of the project, I would very much like to document my entire work and also write some blogs relating to this topic, to spread the word out about the advances of deep learning in astronomy. And also I would like to write blogs explaining the science behind lensing and dark matter to the best of my knowledge.

7. **Aiming for a paper in this direction by making a custom deep learning architecture**

    This is my main goal for this project. I am willing to spend more time even after my project ends with GSoC to develop my own architecture for this problem. As of now, I don't have a solid conclusive idea, but I am fairly confident that after performing extensive research on the literature, I will be able to devise the architecture that I will be developing. I am ready to give a commitment towards this and give my time even after this project.

Future Work

I will be more than glad to work on this project throughout this year, to complete my main goal, i.e to make my own architecture for this problem. I would want the mentor's help and guidance in this aspect and would love to stay in touch with the mentors even after the project is done and hopefully work on future projects too.

Timeline

Dates	Tasks
Pre GSoC Period	Research on Feature Extractors/Domain Adaptation
Community Bonding Period	
May 4 - 28	Getting to know the community and the mentors. Discussing the project, with mentors and analyzing the problem areas/requirements to set up the final project goals and approach. Understand the science behind the working of deeplense to get more knowledge on the domain to prune my approach.
Phase - I (May 29 - July 10)	

Week 1 - 2
 	
May 29 - June 12
	• Setting up deep learning workflow
	    ○ Data Preprocessing
	    ○ Model Setup
	    ○ Training Metrics
	• Domain Based Data Preprocessing Techniques
	• Defining Regression Metrics for comparision

Week 3 - 4 **June 13 - 26**	Research on Optimal Model tuning strategies Apply these techniques to all the previous models built
Week 5 **June 27 - July 3**	Research on AlterNet Model Architecture

Week 6 **July 4 - July 10**	AlterNet Implementation/Analysis in PyTorch
Phase - I Evaluation Period (July 10 - July 14)	
Phase - II (July 14 - Aug 21)	
Week 7 **July 15 - July 22**	Research on EfficientNetV2 Architecture
Week 8 - 9 **July 23 - Aug 7**	EfficientNetv2 Implementation/Analysis in PyTorch
Week 10 **Aug 8 - Aug 15**	Research on Integrating XResNet and AlterNet
Week 11 - 12 **Aug 16 - Aug 28**	Implementation of XResNet + AlterNet Model
Phase - II Final Evaluation Period (Aug 28 - Sept 4)	

Why Me?

I think I am a good technical fit for this project because I have a good amount of experience in PyTorch, and have in-depth knowledge in Deep Learning / Machine Learning. I have done Andrew Ng's course on deep learning, and also took some courses related to it in my institute like Linear Algebra, Probability, Reinforcement Learning, Foundations of Data Science, and Machine Learning. Overall I have more than intermediate knowledge in Deep Learning. Coming to programming in general, I have been programming in python for more than 4 years now, and I am very proficient in the language. I have a good amount of research experience through my internships and my projects. One of the first projects that I was part of was the Pesticide Spraying Robot, which won the Gold Medal in the Inter IIT Tech Meet 2018 conducted in IIT Bombay. I have also participated in international hackathons like HackMIT and ShellHacks which helped me in improving my time management strategies and leadership skills. I have a good research background wherein I worked under Dr. Tarun Rambha, as a summer intern at IISc Bangalore on Reinforcement Learning applications in optimizing traffic network flows, and microscopic driver behaviour.

I was recently drawn to open source, wherein I first started by completing the HacktoberFest challenge as a participant. This gave me a huge confidence boost as well as made me more interested in the open-source world. From here I participated in the Winter Of Code program conducted by the DSC NSEC Club, wherein I was awarded as one of the top contributors. This made me a pro in the usage of Git and Github. With this, I made up my mind to commit to open source programming, and hence thought of applying to GSoC this year. I have applied in the year 2020 for this same program, but I had to withdraw my application in the last moment because of a medical complicaiton and I have discussed this with Prof. Sergei Glyzer as well who was interested in my proposal then.

Astronomy has always excited me, and the amount of time I spend on youtube videos in this aspect is immeasurable. I always wanted to apply my knowledge of programming in these fields but I always ended up assuming that it would be too complex. When I explored it in-depth, I felt very comfortable reading the papers that integrated deep learning with astronomy. I am very much willing to learn more through this project. I want to explore more in this field and I hope I get the chance to do so.

Availability

I wont be working on any other project apart from my day job, so I will be devoting about 25-30 hours per week to the project during GSoC with ML4SCI. If in any case, I happen to fall behind my schedule, I am ready to commit more time to this project by working overtime.

Community Outreach

Being part of such a flourishing community like ML4SCI is a blessing to me. And I plan to stick around for the long run. Even after the GSoC period is over I would like to still work on more projects or develop this one further. I have joined the necessary communication channels for the same. I will also be providing weekly progress updates, blog posts, articles through the communication channels.