NTNU
Kunnskap for en bedre verden

Department of ICT and Natural sciences

AIS2900 - bacheloroppgave ingeniørfag

Intergalactic Machine Learning

Author:
Ahmed Mahammed Aseyr
Ervik Lars-Joar

21.05.2024

Preface

This Bachelor thesis was written by two students from ”Automation and intelligent system” at NTNU Ålesund with the goal of optimizing a machine learning algorithm in a research project with the goal of using the estimated roulette amplitudes to be able to then again estimate dark matter in space.

This project brought a lot of new challenges and learning materials which is the reason we first chose to write this as our Bachelor thesis. We would like to extend our gratitude towards our supervisors Ben David Normann and Hans Georg Schaathun for their help and insight. This project would not been possible without their wisdow and guidance.

Summary

Gravitational lensing is an an phenomenon that have gained some attention for being the key to mapping and understanding the mysterious dark matter. This project further develops on an research project from NTNU. The task this time was the optimization of the machine learning algorithm that was developed by earlier group, by testing and training different hyper parameters. This process gave good results with the inception network, other test that was done with the AlexNet did not give the results that was expected. In this project there was also an attempt at trying to estimate the mass of dark matter, but there was not enough time to do both of these tasks. The attempt however was good enough to be the start of something, and with the optimized machine learning algorithm, could lead to estimated roulette amplitudes good enough so the mass can be estimated.

Contents

Terminology

IDUN Remote High Performance Computing Clusters **Ground truth** Correct result that is hidden until a prediction has been made

Notation

κ Surface mass density

Σ Projected mass density

$\Sigma_{\mathbf{crit}}$ Critical density

ψ Lens potential

ξ Distance between the lens and the distorted image

σ Projected mass distribution

D_L Distance from observer to lens plane

D_S Distance from observer to source plane

R_E Einstein radius

Abbreviations

GL Graviational lensing

ML Machine learning

AI Artificial intelligence

DL Deep learning

ANN Artificial Neural Network

CNN Convolutional Neural Network

List of Figures

Chapter 1

1 Introduction

Today we are able to directly observe only around 5% of the universe, this is the commonly named luminous matter. Approximately 68% of the all matter in the universe is dark energy while the remaining 27% is dark matter(DM). DM can not reflect nor emit light, and thus making it hard to observe through normal methods. The phenomenon of gravitational lensing(GL) gained recognition in 1919 when Arthur Eddington confirmed Albert Einstein's Theory of relativity by measuring the sunlight curved during an solar eclipse. Since then, GL have become one of the main methods to collect information from the night sky[8]. GL happens when light passes through a gravitational field and is deflected, this effects light and dark matter indiscriminately However, GL is particular useful when observing DM, by measuring how much the gravity bends the light DM becomes somewhat observable. This however is time consuming task that can take days. Machine learning algorithms are increasingly being used as an efficient tool to help with these calculations.

1.1 Background

This research project started at the university and this group being the third one to write their bachelor thesis on this subject. The first group[19], developed a tool for simulating different lensing models and used ML to analyse images with GL. the second group[10], further developed on the ML. This years goal will be to optimize the ML. There have also been other work done on this project by masters students as well as professors at the university. The collective goal for this is the complete mapping of the universe, creating an optimal ML to figure out the mass and size of DM will be of great help on the way to doing this.

1.2 Problem

This report is a deep dive into how the group took on the challenge of making an optimal ML algorithm for calculating GL and our look at the process of estimating the mass of dark matter, which required understanding of the math equations. Roughly explained the process for reaching this goal, use a preexisting simulator[18] to generate images, these images are made up of different parameters and it is these parameters that is the base for our machine learning. Datasets are created and then is put in a prototype ML system[17] to train and test the data. Since the objective is to optimize the ML algorithm, the group will further develop the ML code by testing and training different networks to see how they are affected by tuning the hyper parameters. The prototype ML system are developed by the supervisor and earlier groups, even though this makes the start up of our project simpler, this also came with its own set of challenges, one being understanding the code for the simulator and especially the ML code. But both these codes are quite complex and is not as well documented as it could be. This meant that some time went to understanding the code and learning almost everything from the ground up.

Report structure

The report is structured as:

- **Chapter 2 - Theoretical basis** This chapter is divded into main parts: cosmology and machine learning. This chapter serves as theoretical background for everything that is to come later in the report.

- **Chapter 3 - Method** This chapter contains how the experimentation was done.
- **Chapter 4 - Result** This part will present the result of the data from the experiments, and what was learnt.
- **Chapter 5 - Discussion** The discussion chapter will contain reasons for why the methodology chosen was right and/or wrong but also what could have been done differently.
- **Chapter 6 - Conclusion** The final chapter will have an overall conclusion on the thesis, were the group discusses what it has learnt over the span of the experiment and the challenges faced.

Chapter 2

Theoretical basis

This section works as a brief introduction to the basics of both cosmology and machine learning which may be needed to fully understand the work being done in this report. A lot of the info found here will be the same as in the previous year's bachelor thesis surrounding the same project.

Cosmology is the study of the content and evolution of our universe[6]. As time progressed throughout the years studying these processes have become more complex, as such the tools needed to observe our universe also required an update. Looking forward on the next decade of cosmology while considering the vast amount of data being delivered, integrating AI and ML in cosmology is the logical next move. In some aspect ML have already made a significant impact, showing promising results in overcoming some of the computational bottlenecks one usually finds with traditional statistical techniques. Machine learning techniques will be crucial for detecting and classify cosmological sources, extract information from images, and optimize observing strategies[6]. Examples of these are of course this research project with the goal of mapping dark matter and some of its core theme, such as galaxy clustering, strong- and weak lensing but also supernovas and cosmic microwave background.

1.3 Cosmology

Cosmology is the term used when talking about the study and knowledge about the universe.

1.3.1 Gravitational lensing

Every object in space emits some form of light, that is the way we can see these objects, even if they are far away, but these light ray photons are prone to being affected by gravity, and considering gravity is dictated by how big a certain mass is we can somewhat know how big a mass is by seeing how much the light swerves from its original trajectory. Gravitational lensing can be split into three subcategories, strong lensing, weak lensing and microlensing. In this report the only categories we will be looking at is strong lensing and weak lensing. [5]

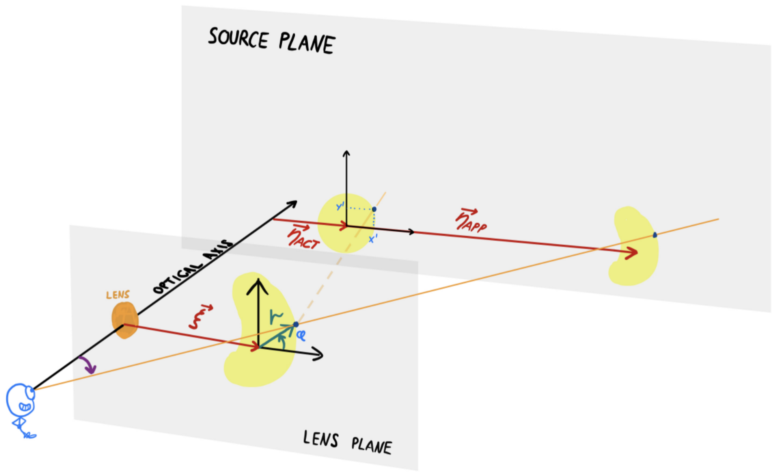


Figure 1: Illustration of GL. source: [7]

Strong Lensing occurs when a nearer object splits the light from a more distant source into multiple resolved images [5]. For strong lensing to occur the lens and the source need to be aligned. Strong lensing is simple to identify when found, the reason for this being that images merge together, creating structures that are visually striking[5]. This can be seen clearly on figure 2, which is a image of a galaxy cluster, the long structure to the lower left of the center was the first lensed arc to be discorvered[5].In contrast to GL that makes it possible to detect both dark and luminous matter other methods usually focus only on the luminous matter. It is possible to learn about dark matter by comparing the findings of other methods and comparing it to GL.

Figure 2: Galaxy cluster. Source:[5]

Weak Lensing occurs when the light emitting from a source is affected but not split into more different visible light sources. This model unlike the other two can not be inferred from a single object but is observable when many source can be analyzed to give a significant signal[5]. Multiple images will form when the surface mass density of the lens is above a certain threshold[5].

Roulette Formalism

Roulette formalism is a mathematical framework that utilizes weak lensing to simulate strong lensing. Weak lensed objects can then have more prevalent features and give another way of simulating strong lensing [7].

Cosmological Redshift

When light travels across great distances, as it does in space, the light waves will be stretched whilst traveling. The light will appear as the color red since red has the longest wave length in the color spectrum, thus objects that are at greater distances from the observer will appear red. The redness of the light can then be used to measure the distance between observer and observed object, like for example a galaxy [19].

Einstein radius

Einstein radius occurs when the light source, lens and observer all lined up. The image of the observed objects forms a ring around the lens[9]. For this to occur the object must be large like a galaxy or a black hole, the distance between the light source and the observer also plays a big role for this to happen. Figure 3 shows the Einstein rings phenomenon.

Figure 3: Einstein radius. [5]

1.3.2 Lenses

In cosmology, a seemingly simple lens transforms into an sophisticated tool, crucial for mapping the universe. Known as cosmic lenses, these phenomena bend light in space due to the gravitational influence from massive object such as galaxies and galaxy clusters. Each lens is defined by its lensing potential, denoted by ψ which tells about the extent of the bending. There are many types of lens models and some of them will be mentioned briefly.

Point mass is the most simple lens one can study, point mass is a lens where all the mass is concentrated at one point. This model however is only used for convince sake the reason for this being that such an idea of all the mass of an lens being concentrated into an infinite small point is believed to only exist in black holes. In practical situations, more complex models are used.

singular isothermal sphere (SIS) is another simple and effective lens model. SIS assumes that the lensing galaxy has a spherical symmetry and follows an isothermal density profile, meaning that the mass density decreases inversly with the square of distance from the center . The following is the lens potential for the point mass and SIS lens.

$$\psi_{sis}(\xi) = \frac{R_E}{D_L^2} * \xi \tag{1}$$

In equation 1 R_E is the Einstein radius

$$\psi_{PM}(\theta) = \theta - \frac{\theta_E^2}{\theta} \tag{2}$$

1.3.3 Mass reconstruction

The following equations was given by our supervisors, more detailed explanation on the equations can be found in attachments.

The relation between the amplitude and potential is expressed as:

$$
\begin{aligned}
\psi_x &= -\frac{1}{D_L}\alpha_1^0, & \psi_y &= -\frac{1}{D_L}\beta_1^0, & & \\
\psi_{xx} &= -\frac{1}{D_L^2}\left(\alpha_0^1+\alpha_2^1\right), & \psi_{yy} &= -\frac{1}{D_L^2}\left(\alpha_0^1-\alpha_2^1\right), & \psi_{xy} &= -\frac{1}{D_L^2}\beta_1^1, \\
\psi_{3x} &= -\frac{1}{D_L^3}\left(\alpha_1^2+\alpha_3^2\right), & \psi_{3y} &= -\frac{1}{D_L^3}\left(\beta_1^2-\beta_3^2\right), & & \\
\psi_{xxy} &= -\frac{1}{3D_L^3}\left(\alpha_1^2-3\alpha_3^2\right), & \psi_{xxx} &= -\frac{1}{3D_L^3}\left(\beta_1^2+3\beta_3^2\right), & & \\
\psi_{4x} &= -\frac{1}{D_L^4}\left(\alpha_0^3+\alpha_2^3+\alpha_4^3\right), & \psi_{4y} &= \frac{1}{D_L^4}\left(\alpha_0^3-\alpha_2^3+\alpha_4^3\right), & & \\
\psi_{2x2y} &= -\frac{1}{3D_L^4}\left(\alpha_0^3-3\alpha_4^3\right), & \psi_{y3x} &= \frac{1}{2D_L^4}\left(\beta_2^3+2\beta_4^3\right), & \psi_{x3y} &= -\frac{1}{2D_L^4}\left(\beta_2^3-2\beta_4^3\right), \\
\psi_{5x} &= -\frac{1}{D_L^5}\left(\alpha_1^4+\alpha_3^4+\alpha_5^4\right), & \psi_{5y} &= \frac{1}{D_L^5}\left(\beta_1^4-\beta_3^4+\beta_5^4\right), & & \\
\psi_{4xy} &= -\frac{1}{5D_L^5}\left(\beta_1^4+3\beta_3^4+5\beta_5^4\right), & \psi_{x4y} &= -\frac{1}{5D_L^5}\left(\alpha_1^4-3\alpha_3^4+5\alpha_5^4\right), & & \\
\psi_{3xyy} &= \frac{1}{5D_L^5}\left(-\alpha_1^4+\alpha_3^4+5\alpha_5^4\right), & \psi_{xxx3y} &= -\frac{1}{5D_L^5}\left(\beta_1^4+\beta_3^4-5\beta_5^4\right). & &
\end{aligned}
\tag{3}
$$

For circular symmetry, mass is expressed as:

$$M(\xi) = \int_0^{\xi'} d\xi'\xi'\Sigma(\xi') \tag{4}$$

In circular symmetry it is only necessary to consider:

$$\mu_s^m = \sqrt{(\alpha_s^m)^2+(\beta_s^m)^2} \tag{5}$$

This gives these relations:

$$\frac{M}{2\pi\xi^2\Sigma_{\mathrm{crit}}} = \frac{1}{2}\left(\mu_2^1+\mu_0^1\right), \tag{6}$$

$$\frac{M'}{2\pi\xi^2\Sigma_{\mathrm{crit}}} = \mu_0^1, \tag{7}$$

$$\frac{M''}{2\pi\xi^2\Sigma_{\mathrm{crit}}} = -\frac{2}{3}\frac{\xi}{D_L}\mu_1^2+\mu_0^1, \tag{8}$$

$$\frac{M'''}{2\pi\xi^2\Sigma_{\mathrm{crit}}} = \frac{4}{3}\left(\frac{\xi}{D_L}\right)^2\mu_0^3-\frac{2}{3}\frac{\xi}{D_L}\mu_1^2, \tag{9}$$

$$\frac{M''''}{2\pi\xi^2\Sigma_{\mathrm{crit}}} = -\frac{4}{5}\left(\frac{\xi}{D_L}\right)^3\mu_1^4+\frac{8}{3}\left(\frac{\xi}{D_L}\right)^2\mu_0^3+\frac{2}{3}\frac{\xi}{D_L}\mu_1^2, \tag{10}$$

And this goes on for higher order derivatives. The critical density(Σ_{crit}) is defined as:

$$\Sigma_{crit} = \frac{C^2}{4\pi G}\frac{D_s}{D_{LS}} \tag{11}$$

for the SIS model:

$$\Sigma = \frac{R_E}{2\xi}\Sigma_{crit} \tag{12}$$

where Σ is the projected mass distribution and is defined as:

$$\Sigma(\xi) = \int \rho d\chi \tag{13}$$

ρ represent the three-dimensional mass distribution, and χ is the co-ordinate along the optical axis.

By integrating equation 4 the mass for a SIS lens can be defined as:

$$M(\xi) = \Sigma_{crit}\pi R_E\xi \tag{14}$$

$$M'(\xi) = \Sigma_{crit}\pi R_E \tag{15}$$

$$M''(\xi) = 0 \tag{16}$$

By using Equation 6 and 7, and substituting for the M we found in 14 and 15 it becomes:

$$\frac{\Sigma_{crit}\pi R_E\xi}{2\pi\xi^2\Sigma_{crit}} = \frac{1}{2}(\mu_2^1 + \mu_0^1) \tag{17}$$

After simplifying the equation:

$$\frac{R_E}{\xi} = (\mu_2^1) \tag{18}$$

We can do the same for the first derivatives as well(11)

$$\frac{\Sigma_{crit}\pi R_E}{2\pi\xi^2\Sigma_{crit}} = \mu_0^1 \tag{19}$$

It possible to simplify this as done with the previous equation above:

$$\frac{R_E}{2\xi^2} = \mu_0^1 \tag{20}$$

Now that there is two equations with two unknowns, we can use these unknowns in relations to each other and find the equations for the amplitude:

$$\begin{aligned} \mu_2^1 &= \frac{R_E(\pi\xi - 1}{2\pi\xi^2} \\ \mu_0^1 &= \frac{\xi_0}{2\pi\xi^2} \end{aligned} \tag{21}$$

Using these equations one can find the amplitudes, which makes it possible to estimate the mass of the dark matter.

1.4 Machine Learning

This section will explain the machine learning part of the project.

1.4.1 Artificial Neural Networks

Artificial neural network or simply just neural network is a subsection in machine learning which is used a lot within image processing. Neural network tries to mimic the human brain in the way it works, the network is connected trough data elements called neurons. Easier way to think of neurons are "a thing that holds a number", these numbers are called activation. Neurons can send and receive values between each other. A basic outline of how a ANN looks like is can be found in figure 4

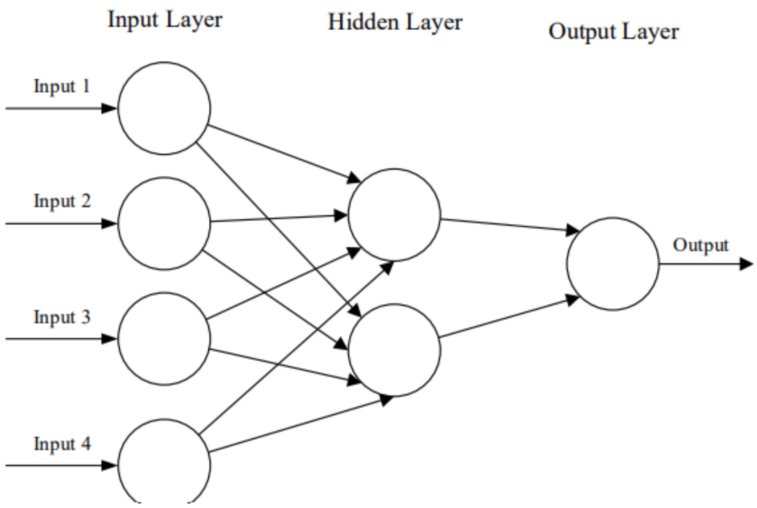


Figure 4: Basic outlay of an ANN. [16]

The neurons are organised in layers, The first layer receives the inputs where as the last layer gives out the output of the network. There are other layers in between which are "hidden", the hidden layers will make decisions from the previous layers, this is the learning part of the process, having multiple hidden layers upon each-other is known as deep learning [13]. The numbers of neurons depend on the task intended for the network. If the task is processing images then each pixel in the image will be represented by an neuron in the input layer. [16]

1.4.2 Convolutional Neural Network

CNN is a version of ANN and thus they work in similar ways, they are both comprised of neurons that take information and improve upon it. The area where CNN truly shines is image recognition, CNN comes with a solution to the limitation of the traditional ANN that it struggles with computing image data the reason for this being that ANN lacks the computational complexity [16]. CNN uses convolution between a set of kernels and the image. CNN are made up of three types of layers: Convolutional layers, pooling layers and fully-connected layers[16]:

- **The convolutional layers** convolve the inputs given to it using a kernel of selected size, and gives the results to the next layer.
- **Pooling layers** reduce the data down by combining n x n regions of the input data using either average or max values.

- **Fully-connected layers** does the same in CNNs as they do in traditional ANNs. They connect all neurons in one layer to all neurons in the next layer.

1.4.3 Deep learning

Since the 1950s a small subset of artificial intelligence(AI), refereed to as machine learning took the field by storm. Within machine learning there exists branch called Neural network, and this is where the concept of deep learning(DL) comes from. Deep learning have done big strides and revolutionised work in several fields since it was founded. Figure 5 shows how AI is build up.

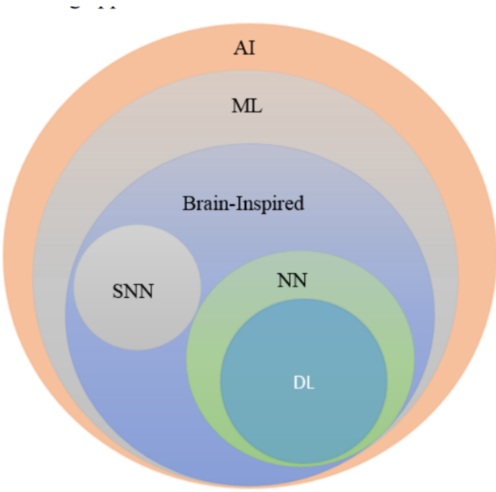


Figure 5: How AI is build up [11]

DL can be broadly categorized into four main types, supervised, semi-supervised/partially supervised, unsupervised and reinforcement learning:

- **Supervised learning** is the most common type of DL, this technique uses labeled data. Meaning that each input data (or instance) in the dataset is paired with an output label that provides information. In other words each sets of input have an corresponding output (x_t, y_t).
- **Semi-supervised learning** happens when parts of the datasets used for training is labeled while the rest is unlabeled. Semi-supervised learning uses the small amount of labeled data together with the large amount of unlabeled data as guide in the learning process.
- **Unsupervised learning** consist entirely of unlabeled datasets, this technique learns the internal representation or important features to discover un-

known relationships within the input data [11]. Unsupervised learning uses approaches such as clustering, dimensional reduction and generative techniques.

- **Reinforcement learning**(RL) is a used for unknown environments. RL is sometimes called semi-supervised learning, and many semi-supervised and unsupervised techniques that have been implemented based on RL[11]. RL do not have as simple loss function as the Traditional supervised learning , which makes learning harder compared with RL[11].

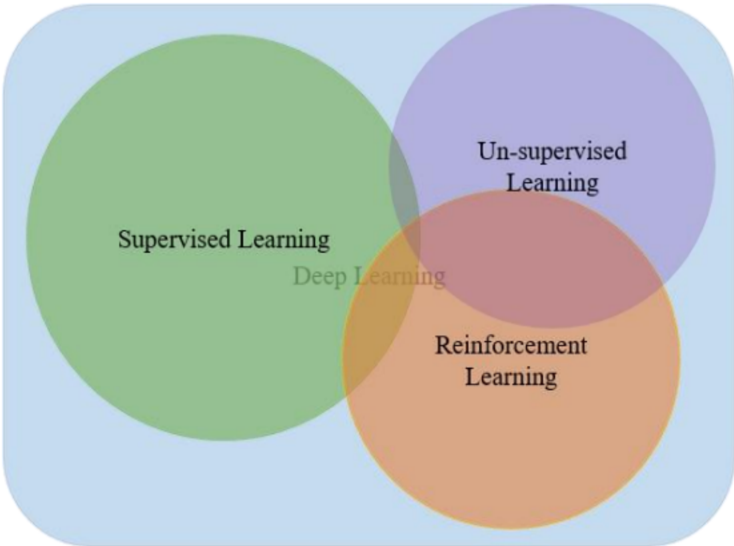


Figure 6: Categories of deep learning. Source: [11]

1.4.4 Networks

In machine learning, architecture and particularly neural networks, play a central role in modeling complex patterns and relationships in data. In this project there are two types of networks that have been the main focus, AlexNet and Inception v.3. To clarify, these are types CNN architectures, and as mentioned earlier CNN is a category of ANN.

- **AlexNet** played a significant role in the field of deep learning, AlexNet excels when it comes to image recognition. AlexNet consist of eight layers where the first five are convolutional and the following three are fully-connected.Figure 7shows a how a basic layout of an AlexNet looks like. AlexNet was first recognized 2012 when Alex Krizhevesky and others won the most difficult "ImageNet" challenge for visual object recognition called the "ImageNet Large

Scale Visual Recognition Challenge (ILSVRC)". Readers who are interested in reading more about AlexNet can find it here[2]

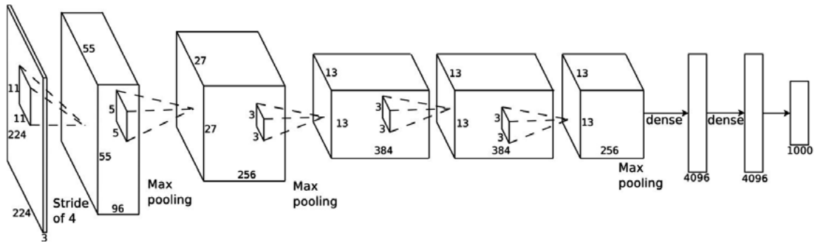


Figure 7: AlexNet

- **Inception V.3** builds on its predecessors foundation, the model was created after a lot of different ideas were tested by many researchers over the years. The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concatenations, dropouts, and fully connected layers[1]. This architecture focuses on several innovative features to be as optimal as possible. The key improvements being factorization into smaller convolutions, this is to reduce computational cost without sacrificing width or depth. Label smoothing is another technique used in this model, this is to minimize the confidence of the model to better the learning process. Inception V.3 is an improvement of architectures like AlexNet and VGGNet.

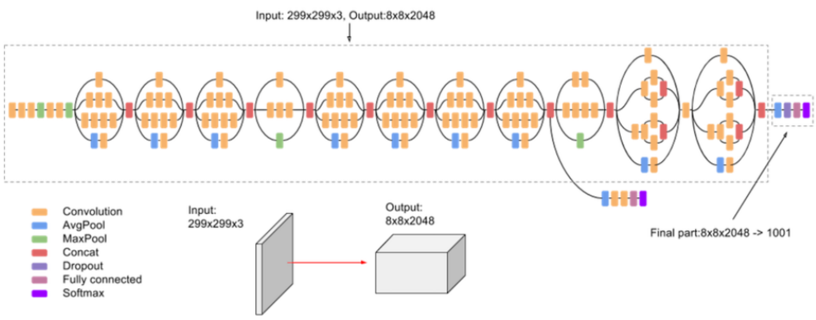


Figure 8: High level diagram of Inception V.3: Source: [1]

1.5 Parameters and Hyper Parameters

Parameters

Are variables that the model uses to make predictions, these are learned from the data during training. Model training usually starts with setting parameters to some values, these are either random or set to zero. These values are then adjusted by an algorithms such as gradient descent and other learning algorithms. These algorithms are continuously updating the parameter values as the learning progress[12]. Examples of parameters are: Weights and biases of a NN and The cluster centroids in clustering.

Hyper Parameters

are parameters whose values control the learning process and determine the values of model parameters that a learning algorithm ends up learning[12]. Usually when designing a machine learning model, the hyper parameters are chosen before the training of the model begins, in other words hyper parameters are external to the model. The hyper parameters are used by the model when learning but its not part of the result. Some examples of hyper parameters are: Batch size, learning rate, epochs and more. The process of setting the hyper parameters requires a considerable amount of trial and error.

Learning rate and gradient descent

One of the most important hyper parameters is learning rate, the learning rate or speed at which the model learns is controlled by how much to change the model's weights with respect to the gradient of the loss function. Learning rate is usually categorized in two, high learning rate and low learning rate. a smaller learning rate requires more training epochs, the reason for this being that the smaller changes made to the weights each update. While a larger learning rate result in rapid changes and thus require fewer training epochs[3].

Gradient descent are used for training when working with deep learning neural networks. Gradient descent using an optimization algorithm used to find the vales of parameters. Gradient descent is best used when the parameters cannot be calculated using algebra[3].

1.6 Previous work with machine learning on Gravitational lensing

In the recent years there have been a lot of significant work done not only in machine learning but also gravitational lensing to better understand what dark matter is. There are different approaches that have been used to reach these goals but almost all of the successful ones have used CNN such as the paper by Joshua Wilde and more [20], where they had some of the same goals as this project but also focused on proving why CNN is the way to go forward with this research. There have also been some that have developed their own CNN network like this with this project by Samira Rezaei and more [15]. Other than these two examples there are a lot of work being done using ML on GL and this project hope to make some dent into the research process.

Chapter 3

Method

This chapter will be detailing the experimentation process of the project from start to end. This chapter will mostly consist of machine learning, from how we tested the work previous groups have done and how we attempted to optimize the machine learning algorithm.

1.7 Going through earlier work

As mentioned earlier this is not the first time students have chosen this research subject as their Bachelor thesis, therefore the first step was to go through the work of the previous groups. This together with the code bases from the supervisor, gave enough tools to test and start with the research part of the project. However testing the code proved more challenging then expected, there was documentation to help on the way to understand the code bases, but a lot of it had become outdated over the year, with newer modules being used, and processes being revised, too much time was spent unnecessarily. Also with the addition challenge of having to use new hardware IDUN, another learning challenge had to be passed to begin using the previous work.

1.8 Generating images

For images a preexisting simulator was used, with the main task of generating images for training and testing data. It generates images of an un-distorted light source and creates a distorted image by using equations described in the lens model[19]. The distortion depends on the lens model chosen together with some randomly generated variables. Figure 9 shows the GUI of the simulator, with the image on the left being the reference image, while the one the right is how the distorted image looks.

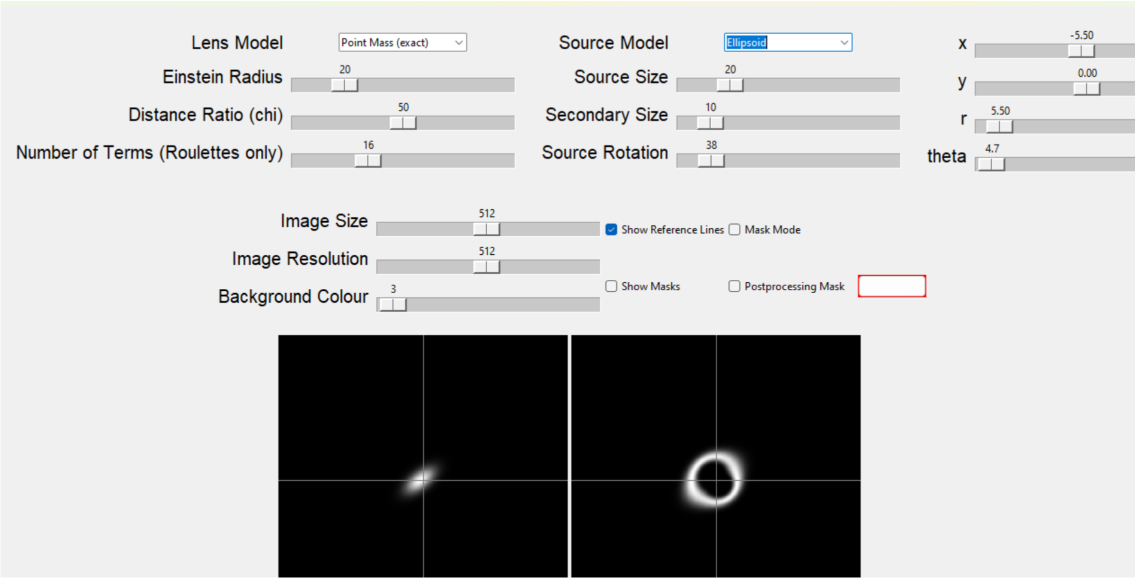


Figure 9: Simulator GUI

The simulator was setup to create datasets of 20000 each, these images are created using the processing power of the IDUN computers, and when creating 7 datasets of 20000 each, the time of creating was approximately 1.5 to 2 hours. These images are created in a 800x800 size, but after being centered and cut down to a 400x400 size around the center of the light source, this is to make the light source be the center of the image.

1.9 Machine learning

1.9.1 Why Machine learning

As talked about in the theory part of this thesis, machine learning is a powerful tool that can be utilized to do the heavy work that people otherwise would need a lot of time and effort to solve. In this research project machine learning is used to reverse warped or observed images into the images that would be seen had the picture been taken from a distance where a gravitational lensing would not take affect. As it stands photos are generated and not ”real” per-se, the images are generate with different parameters, these parameters are used as a ground truth to be compared to the values gained from the machine learning algorithm after it has undergone a training cycle, where it has x amount of pictures and can learn how it will look with different parameters. The machine learning algorithm can then process the test data and generate variables that are subsequently utilized in the reconstruction of the images.

1.9.2 Choosing network

The previous group had done rigorous testing of different networks[10], and seeing the networks implemented within the machine learning code from the supervisor, the chosen networks became Inception v.3 and the AlexNet networks.

1.9.3 Hyper parameters

Considering the previous group had done most of the general network testing, this year would consist of using said networks and further optimizing to find the "optimal" hyper parameters that can strike a good balance between time and good results. To find the optimal parameters there was done testing to see how the different parameters altered the final outcome, for example by changing the number of epochs the algorithm goes through or changing the learning rate and seeing how the values change when going through the layers.

1.9.4 IDUN

IDUN is a Cloud service specializing in high power workstations, this is not a service specifically made for NTNU, but taken in by users and professors of NTNU to be used to run script that otherwise would take a lot of time on personal computers. Using IDUN one can take these heavy load script and run them on workstations composed of components specifically picked to run said heavy tasks. IDUN was used by the group to run all the script steps that go into running a single test, from creating images, running machine learning, comparing the found results and plotting scripts.

IDUN has capabilities of running multiple test simultaneously, this was good help in this thesis. Running multiple tests at the same time was needed to find the best results in the smallest time. Running multiple tests allowed us as the user to change the hyper parameters a little bit and run it while the same test with a different hyper parameter ran in parallel.

IDUN Scripts

IDUN has its own way of running code based on certain setup of code. IDUN uses a file type known as "slurm", these files are comprised of all the data IDUN needs to run, including modules needed, the variables used and the scripts of snippets of code the gets ran. These were previously made by the previous years group, and just altered to fit our means.

These are the scripts used in this experiment the sequence and what they do:

- 01dataset.slurm: Creates Images using the simulator
- 02roulette.slurm: Creates the roulette test and train datasets for use in the testing
- 03ml.slurm: Runs the machine learning algorithm based on chosen network and hyper parameters
- 04reconstruct.slurm: Reconstructs roulette amplitudes from the outputs of the machine learning and reconstructs images based on the amplitudes
- 05simplecompare.slurm: Compares the values of the machine learning outputs and the ground truth.
- 06plot.slurm: Creates graphs based on the findings in the "simplecompare" script.

1.9.5 Problems with the machine learning code

The machine learning code was preexisting, which gave a good starting point and was huge help on the way, but the fact that it was written by somebody else made it challenging to understand all the details of how the code was built and understanding its functionality, considering the code being such a complex and extensive one, understanding was very important. As the time spent on code increased, knowledge of the code also increased, after some time the problems with the code started appearing, which then led to a debugging phase.

1.9.6 Debugging

The IDUN scripts made from previous groups were somewhat outdated, so to get them to run the modules used to run the scripts needed to be updated from what it had been earlier. The GitHub repository that has the code and documentation were somewhat bare, that led to a bit of struggling to make the code run successfully and understanding everything.

1.10 Experiment

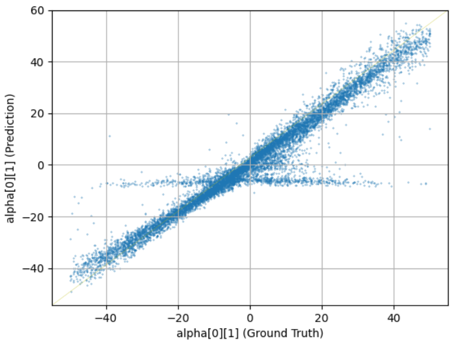
To find the difference between different hyper parameters, a base line had to be established. Since there was no prior fixed machine learning hyper parameters, the choice of using the values that came with the code when using it was as good as any. Hyper parameters of learning rate set to 0.001, epoch amount set to 50 and a drop off rate set to 0.5. To make the test not take that much time the conscious decision to use a train data size of 20000 and test data size of 10000, different results might be found using a higher train data size, but when trying to find out how the different hyper parameters affects the result, a smaller size was chosen.

1.10.1 Base Results

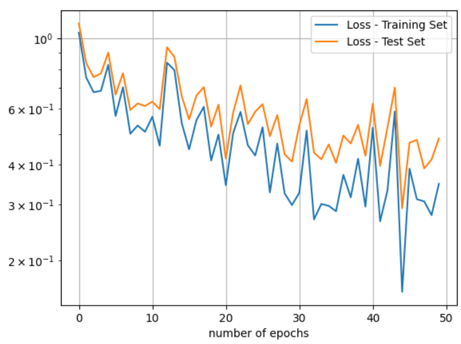
Below are a few of the resulting graphs using the base hyper parameters 10. They act as a guideline to see how the different hyper parameters act. The angle of deflection graphs 10a is the first roulette amplitude, and is chosen because it has to be well estimated to be able to do one of the more crucial calculations. The loss function 10b is chosen to give a debugger view on the situation, seeing how the algorithm behaves between each epoch was good for the tuning part of the subject. Within a Gaussian distribution the values are correct if they fall within a certain frame, using that as a guideline for our data was helpful do dictate a good or bad output 10c, and using the line going diagonally as our guide point, the values should ideally be symmetrically around the line. 10d is a graph that shows the x and y positions of the lens potential, where ξ_1 is x and ξ_1 is y, the ξ is used to dictate what kind of values they are.

Looking at the values from the base test, one can see that the results are not really satisfactory, looking at 10a, we can see that there are a lot of the predicted results laying horizontally, these should be following the steep line with the rest of the results. Otherwise values at either end of the graph also looks to be spaced about, which is a indication to that these values are not well predicted. For 10c the values should be following the line perfectly but the lower values, almost half the values, get systematically overestimated while some of the upper values get systematically underestimated. Also looking at the loss functions 10b, the learning could be better. In a perfect world there would not be as much fluctuation, and rather a smooth line with a downwards curve towards zero. During certain times, the loss function neither dips nor rises of great value, this is the sort of look we would want on the entire graph, that would indicate that the predicted value does not need much change to reach its "correct" value, the learning rate needs to be paired up with a good epoch amount to function to its best potential, which will be looked at later.

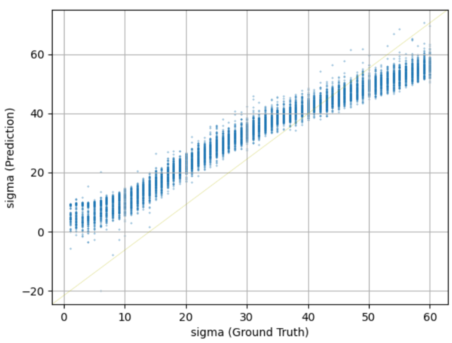
With all this in mind, looking at the tests done one can see whether different hyper parameters works to solve the bias and errors of the base machine learning algorithm.



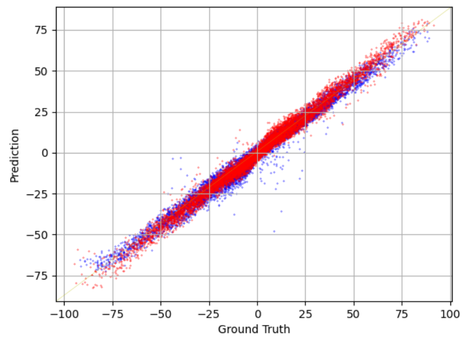
(a) The Angle of deflection



(b) Loss Function



(c) Gaussian distribution



(d) Comparison of estimates and ground truth for ξ_1' (blue) and ξ_2' (red)

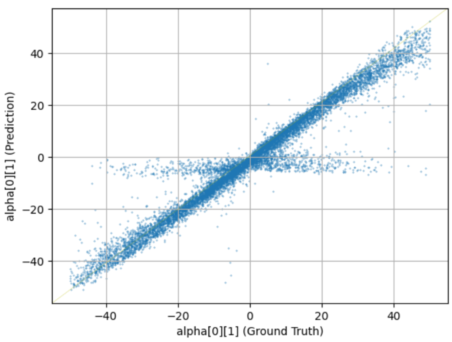
Figure 10: Inception v.3 base

1.10.2 Low Learning Rate

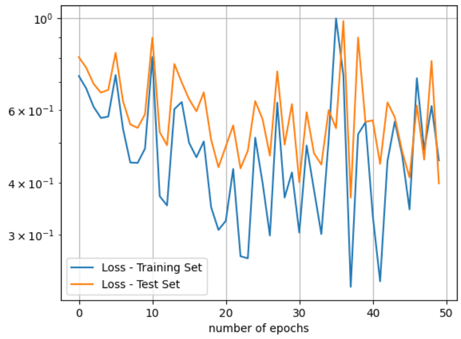
A lower learning rate should try to not change the values that much between each epoch, to help it from overshooting, but may end up getting stuck in a sub optimal position.

A few cliff notes to take away from the graphs of the machine learning using a lower learning rate (0.0001).The loss function is not as frantic as the one in the base tests case,it might look more frantic but it spans a smaller area(barely), but it is still not exactly what we would want, but not necessarily enough to make a observable difference.

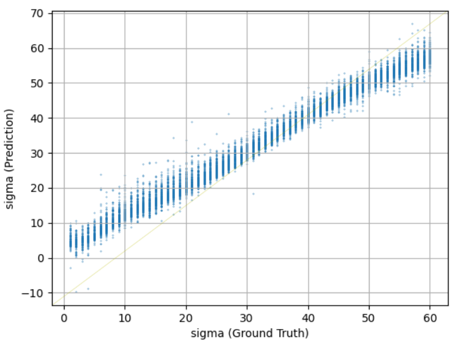
Again when looking at the Gaussian distribution almost half of all the total values are over estimated, while also some of the values in the top gets underestimated.



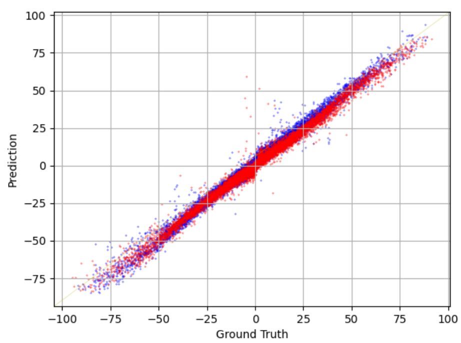
(a) The Angle of deflection



(b) Loss Function



(c) Gaussian distribution



(d) Comparison of estimates and ground truth for ξ_1' (blue) and ξ_2' (red)

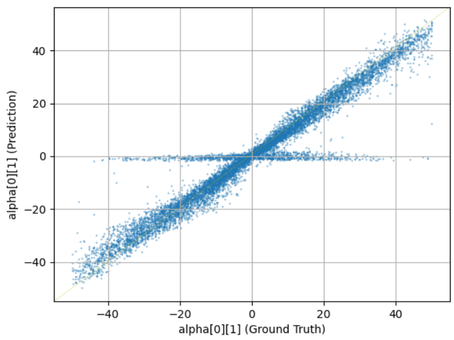
Figure 11: Smaller learning rate graphs

1.10.3 High Learning Rate

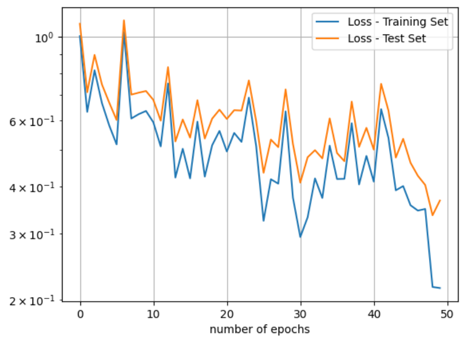
A higher leaning rate should help the algorithm make changes between each epoch to further its search for the correct prediction, but having this too high can make it unable to and up on the correct result.

When looking at the graphs of the machine learning algorithm with a high learning rate (0.01), one can again see that almost all the lower half values are getting overestimated 12c, but not to the same extent of the lower learning rates test.

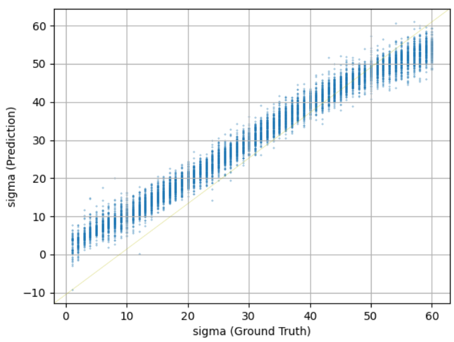
Other then that, the values of the angle of deflection 12a has values at either end that are somewhat strong outliers, and not enough clustering of the values in the expected way.



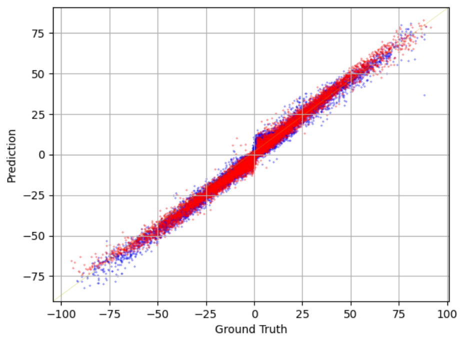
(a) The Angle of deflection



(b) Loss Function



(c) Gaussian distribution



(d) Comparison of estimates and ground truth for ξ_1' (blue) and ξ_2' (red)

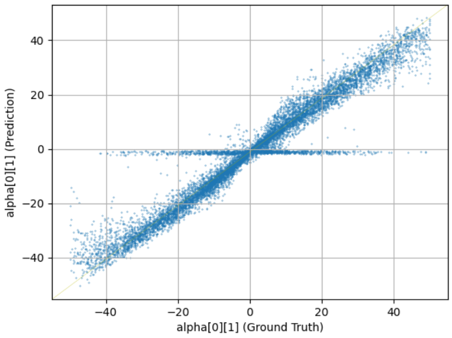
Figure 12: Higher learning rate graphs

1.10.4 Small number of Epochs

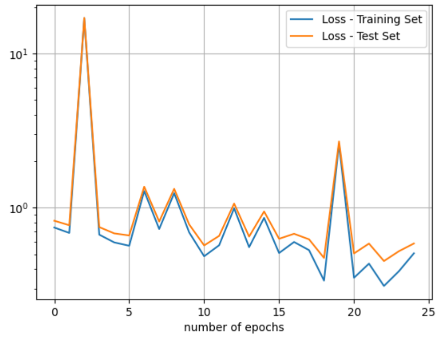
A smaller number of epochs limits the time the machine learning algorithm runs, changing this to a lower value can help it not get caught up in an exponential faulty prediction, but may also not give it enough time to zero in on the correct result.

The smaller number of epochs(25) is interesting in a way that none of the others are, with the fact that the algorithm having less time to maybe shoot off, the loss function is the most calm of all the tests done, when looking at the big spike as an outlier, which lead to not much discrepancy between the values for each epoch. This could be seen as just a fluke, but interesting to think about, considering this is the same learning rate as the base result but has a calmer loss function, but it might be that it is that zoomed in because of the small number of epochs.

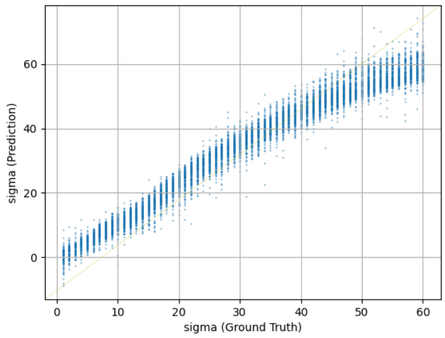
But the loss function is the only positive to take away from it, having this few epochs does not give the algorithm enough time to be able to train itself good enough to be able to predict good enough results. Looking at both 13a and 13d there is a lot of spreading between the values, indicating a poor estimation compared to the ground truth values.



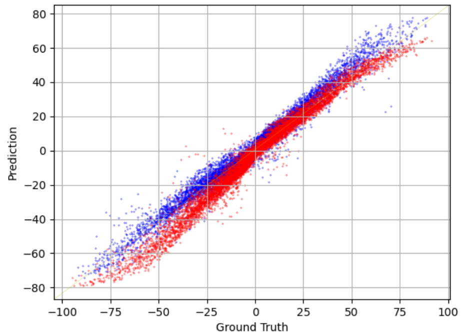
(a) The Angle of deflection



(b) Loss Function



(c) Gaussian distribution



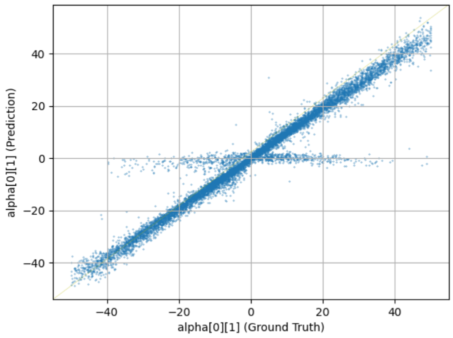
(d) Comparison of estimates and ground truth for ξ_1' (blue) and ξ_2' (red)

Figure 13: Smaller amount of epochs graphs

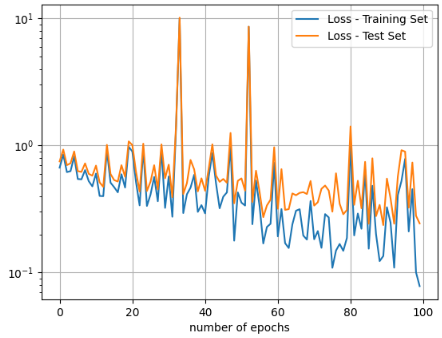
1.10.5 Bigger number of Epochs

Using a higher epoch number gives the algorithm more time to run through the data and learn, thus making the final predicted result better, but could as mentioned in the smaller epochs section constantly wrongly predict the result.

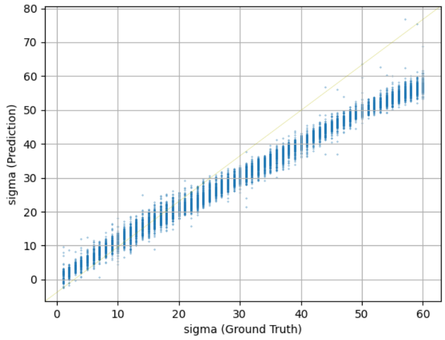
When using a higher number of epochs (100), we do not get the same drastic changes like with the smaller epochs. But looking at 14a and 14d, we can see that there is a lot more clustering at the ends, with the outliers not being as strong as earlier. Another thing to notice is that just a few of the values in the Gaussian distribution are overestimated, but almost all the values in the upper half get underestimated.



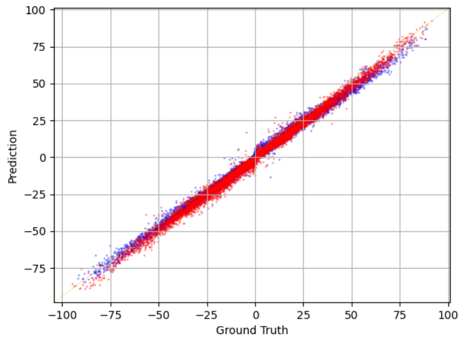
(a) The Angle of deflection



(b) Loss Function



(c) Gaussian distribution



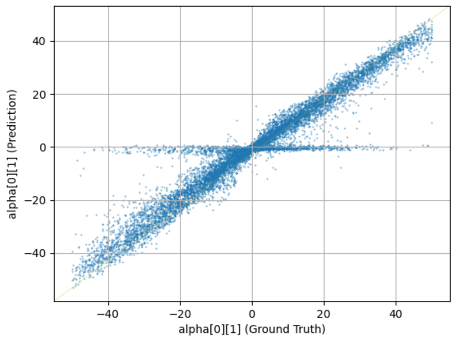
(d) Comparison of estimates and ground truth for ξ_1' (blue) and ξ_2' (red)

Figure 14: Bigger amount of epochs graphs

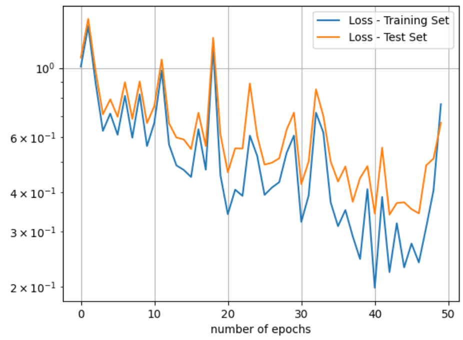
1.10.6 High drop off rate graphs

Drop rates aim to provide randomness to the machine learning, which will test it in its training capabilities when it hits obstacles [14], having a high enough drop off rate can also help the machine learning algorithm remove values that would otherwise be considered as noise or data not needed.

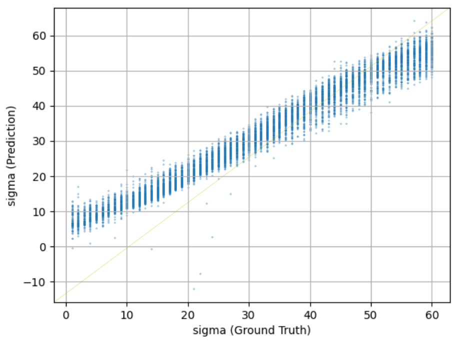
Using a higher drop off rate (0.75), results in graphs looking really similar to the base graphs (10), with overestimation in the the lower half of the values of the Gaussian distribution 15c, and notable spread in the estimation of the angle of deflection 15a.



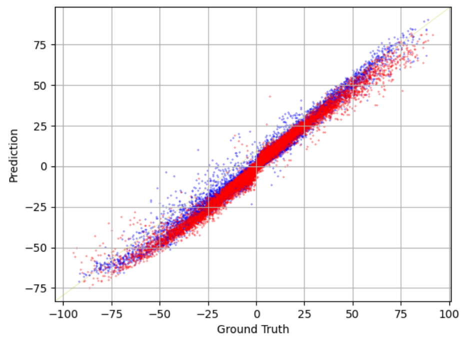
(a) The Angle of deflection



(b) Loss Function



(c) Gaussian distribution



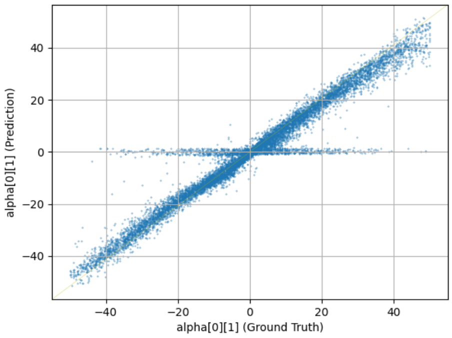
(d) Comparison of estimates and ground truth for ξ_1' (blue) and ξ_2' (red)

Figure 15: Bigger drop off rate graphs

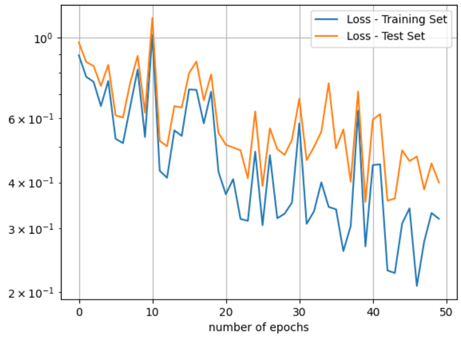
1.10.7 Low drop off rate

As mentioned changing the drop off rate can help to remove data that could be seen as noise from altering the outcome [14].

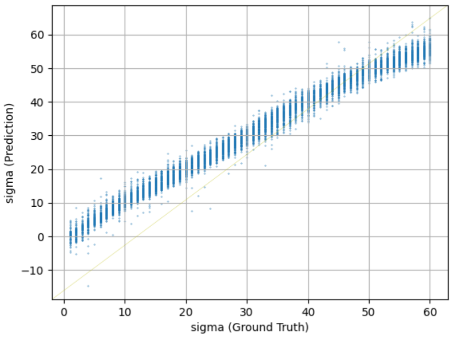
Having a lower drop rate creates less spread in 16a then with a higher drop rate. But outside of that there is not much change provoked by setting a smaller drop off rate. The lower values of the Gaussian distribution in overestimated and only the last values of the upper half are under estimated.



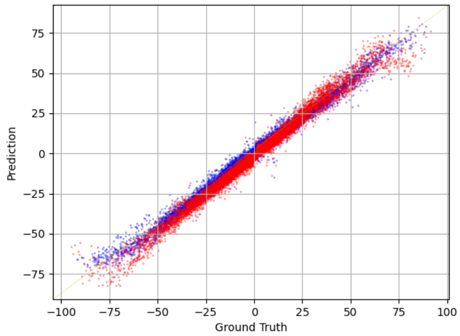
(a) The Angle of deflection



(b) Loss Function



(c) Gaussian distribution



(d) Comparison of estimates and ground truth for ξ_1' (blue) and ξ_2' (red)

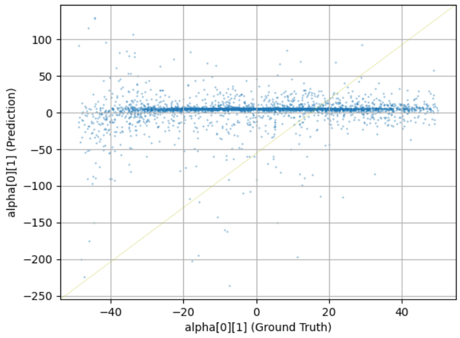
Figure 16: Lower drop off rate

1.11 AlexNet

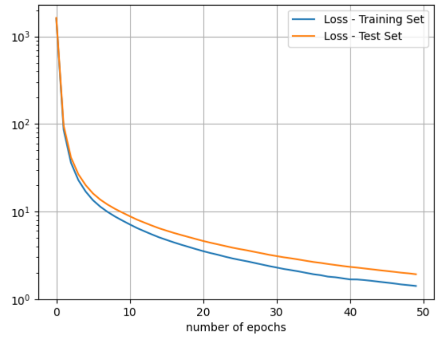
To widen the horizon of what networks could possibly be the best, tests with the AlexNet were also done, considering this was another one of the networks found to give good results [10].

1.11.1 Base

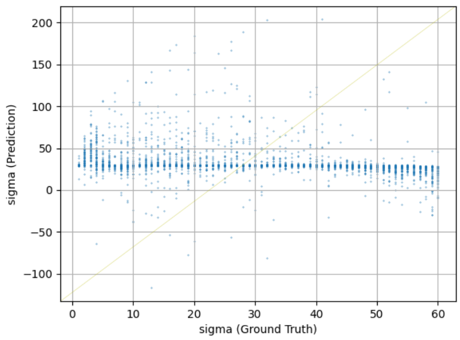
The AlexNet proved some problems during testing, a lot of the graphs and outputs being produced from the use of AlexNet gave really bad results. A lot of the functions gave the output of a flat curve, this is probably due to the fact that the data is being predicted with the same value, most likely the average. The model finds a local optimum point and can not seem to exit it and find the next one, when this happens the loss function will flatten out over the course of many epochs, which looking at the loss graph 15b this makes sense.



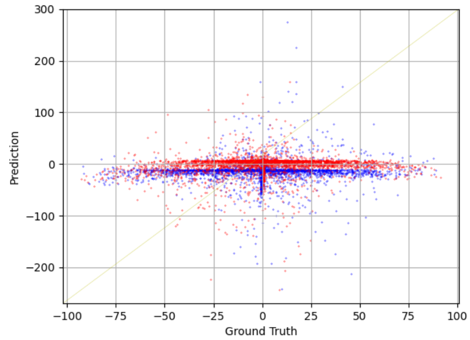
(a) The Angle of deflection



(b) Loss Function



(c) Gaussian distribution

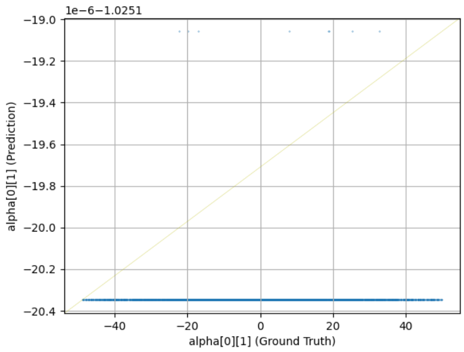


(d) Comparison of estimates and ground truth for ξ_1' (blue) and ξ_2' (red)

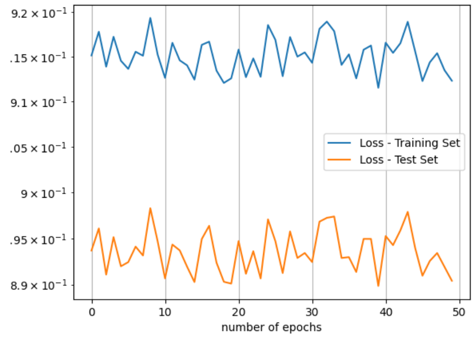
Figure 17: AlexNet Base

1.11.2 High Learning Rate

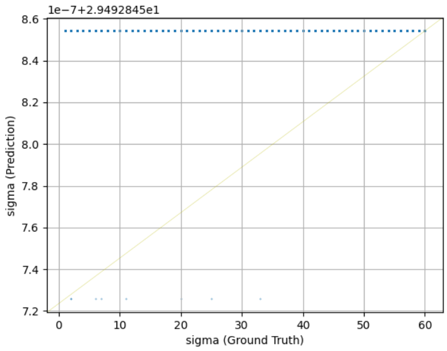
Trying the AlexNet neural network, but with a higher learning rate(0.01), the graphs continue to have the same problems as with the base graphs.



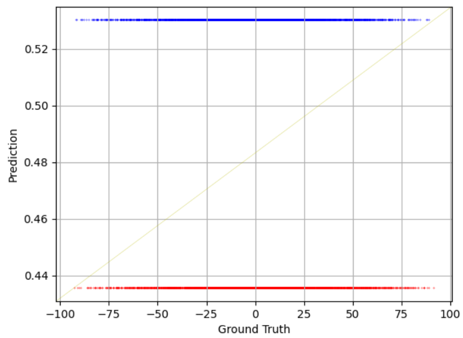
(a) The Angle of deflection



(b) Loss Function



(c) Gaussian distribution

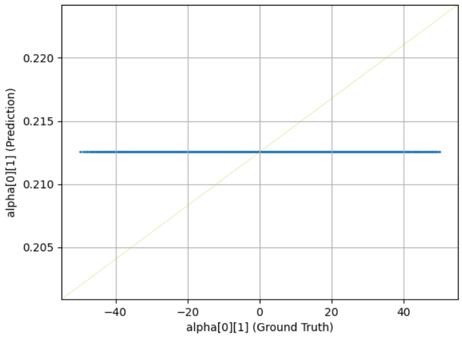


(d) Comparison of estimates and ground truth for ξ_1' (blue) and ξ_2' (red)

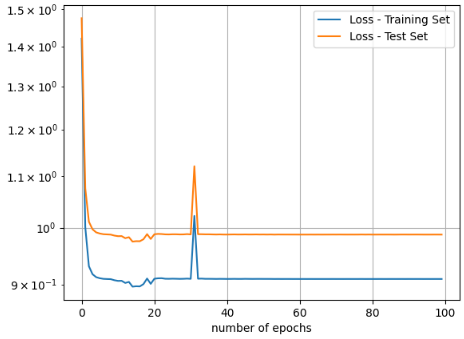
Figure 18: Bigger learning rate

1.11.3 More Epochs

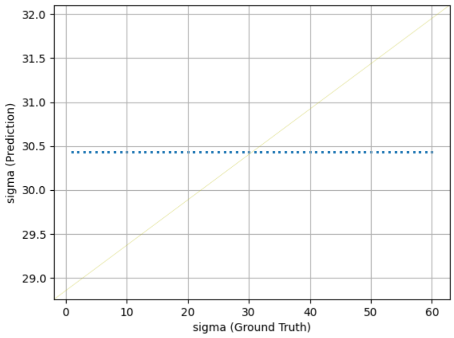
It seems to appear that the AlexNet is really dependent on good hyper parameters, changing the number of epochs from 50 to 100 seemed to not give any different output.



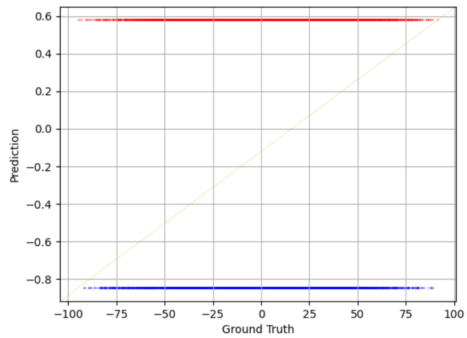
(a) The Angle of deflection



(b) Loss Function



(c) Gaussian distribution



(d) Comparison of estimates and ground truth for ξ_1' (blue) and ξ_2' (red)

Figure 19: Bigger number of epochs

1.11.4 Other hyper parameters

Testing using the same hyper parameters as with the Inception v.3 network yielded more of the same graphs as with the other tests, hence there will be no more graphs of this and instead focus was put on the Inception v.3 network and seeing how mixing the hyper parameters looked.

2 Results

This chapter will present the result we got from the training and testing that was done and mentioned in the previous chapter,

2.1 Takeaways

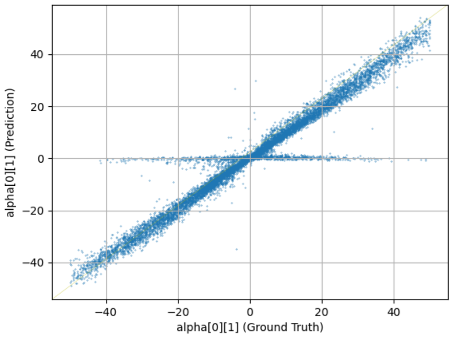
Looking at all of the different tests done, and just changing one hyper parameter, a lot of the algorithms ended up overestimating the values in the lower half, only the test using a higher amount of epochs ended up having lower half values of the Gaussian distribution following the expected curvature, and having the values well estimated.

Mostly if not all the hyper parameters managed to make a difference on the final results, which is good if you want to learn something from it, which was a big part of this thesis.

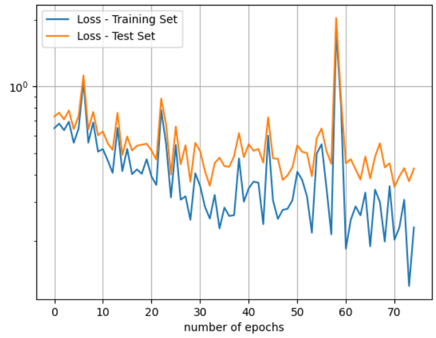
2.2 After using knowledge of testing

After looking at the results from the experiment, and trying to implement it into the machine learning algorithm(Learning rate of 0.0007 and 75 epochs), the results are promising to some extent. The Gaussian distribution has values that are looking really good except for some outliers in the top and bottom end, where they are over- and underestimated respectively. The loss function is also really nice before reaching around 35-40 epochs where it starts to swing, being an indication that the learning rate might be too high.

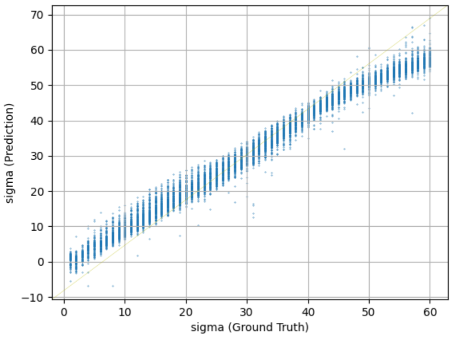
These were interesting results, using what was found in the experimental part of this thesis and implementing and getting promising results was a good step in the right direction.



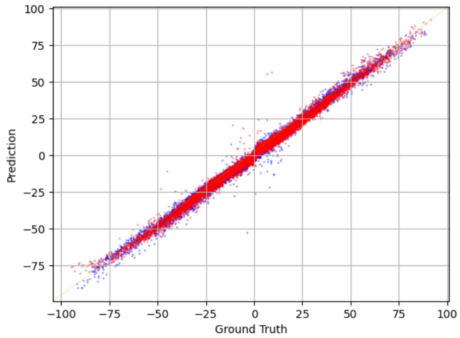
(a) The Angle of deflection



(b) Loss Function



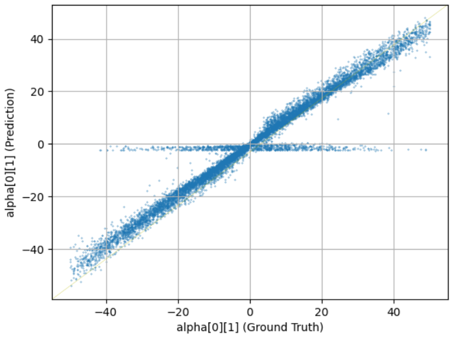
(c) Gaussian distribution



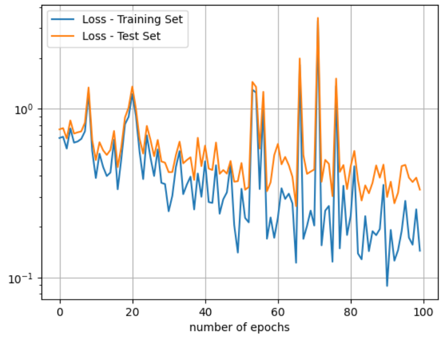
(d) Comparison of estimates and ground truth for ξ_1' (blue) and ξ_2' (red)

Figure 20: First test after experimentation

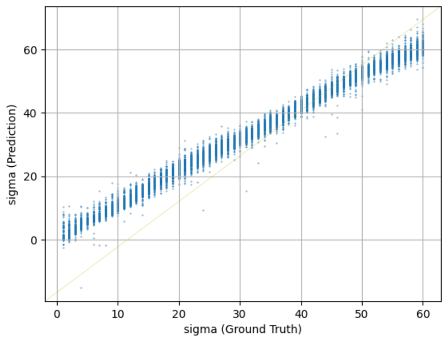
After tweaking the hyper parameters a small bit more(100 epochs and 0.0005 Learning rate), we get somewhat worse results, at least if we look at the Gaussian distribution, as again, the lower values get overestimated. The loss function has times when it actually is bettering an zeroing in to zero, but after 50 epochs the learning rate ends up maybe being too high.



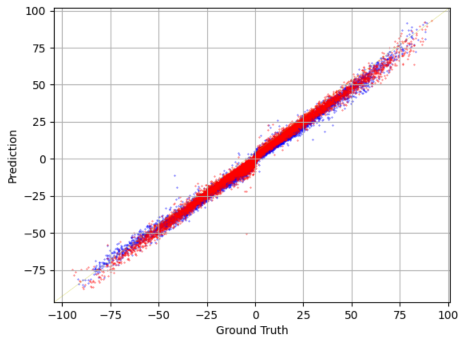
(a) The Angle of deflection



(b) Loss Function



(c) Gaussian distribution



(d) Comparison of estimates and ground truth for ξ_1' (blue) and ξ_2' (red)

Figure 21: Second test after experimentation

Chapter 5

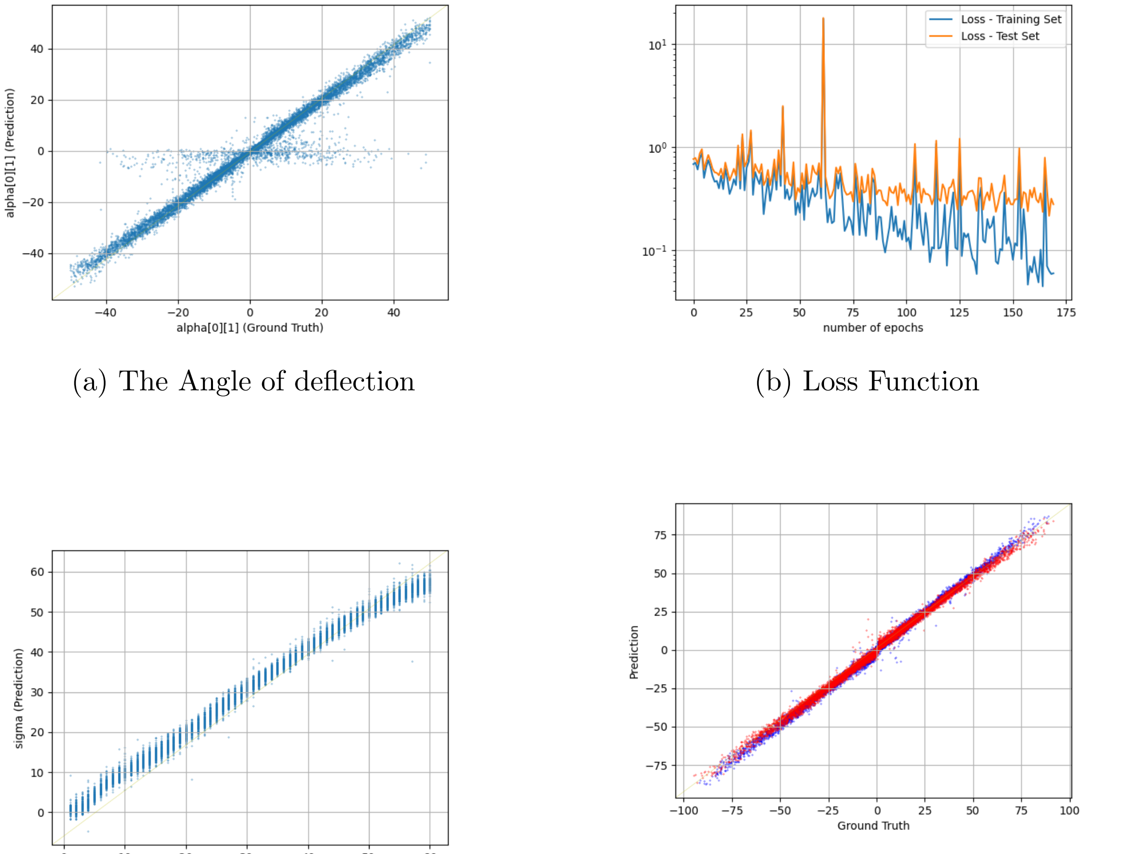
3 Discussion

This chapter will aim to discuss the result of our findings, the methods chosen and the group work.

3.0.1 The Machine Learning

The machine learning ended with a somewhat satisfactory result, but only by changing the hyper parameters and tuning it according to how different hyper parameters react. Using the Inception3 network makes a lot of sense for this type of task in particular, it is documented that it can produce results up to 78.1% accuracy if the number of epochs are higher that 170 [1]. Also on the same page is different ways of optimizing the Inception3 network in particular, one of the ways of doing this was "Ramp-up", a way of changing the learning rate after certain amounts of epochs, at the beginning the learning rate is set to 10% of what it would be normally, and from there it linearly rises within a set of epochs before it start to decay exponentially, this is how the makers of the network managed to hit their 78.1% accuracy result. There are more techniques to optimize the model, a simple yet very efficient way to update the model is by using Stochastic gradient descent(SGD), this is done by nudging the weights in negative direction. Much like SGD momentum it updates the weights but also adds a component in the direction of the previous update[1]. It would be an idea to try and implement these into the code given to see if it could work for the data being used, but not enough time was set aside to try this endeavour.

Just as a test, the machine learning algorithm was tested with 170 epochs, and the graphs showed why google could claim the high accuracy. But the amount of time needed to run this test could be equivalent to running many of the other designed and ran tests. The exact time was not taken, but the machine learning script was ran for over 8 hours at last time of check in, the test was ran on a NVIDIA A100 GPU, which is designed to do heavy load computing, and even with that it took a long time. Also comparing it to the first test after experimentation 20, the discrepancy might not be enough to justify the time used.



(a) The Angle of deflection

(b) Loss Function

(c) Gaussian distribution

(d) Comparison of estimates and ground truth for ξ_1' (blue) and ξ_2' (red)

Figure 22: Using 170 epochs

Otherwise the results of the AlexNet was not really satisfactory, using different hyper parameters did change the graphical outputs, but was not able to make the same impact so it gave similarly good results as with using the Inception3 network. It might be just the AlexNet being really sensitive to the hyper parameters chosen or a change within the hidden layers could help, perhaps maybe another network should be used and implemented in its stead.

3.1 Debugging

As mentioned previously, some of the code needed refurbishing. Within the Inception network, it was mostly changing the modules that the IDUN script needed to fetch. But some changes done to the code were not as well documented, note the code base is not finished and is being constantly developed, some of the documentation and sections walking you through the recipe on how to use it was outdated and some processes were changed. But having it not be super complete led to us as users needing to understand how to look learn error codes and how to act upon them.

Trying to implement the code on local computer gave some difficulties, considering the want to use this on a windows computer whilst it being set up to run on the IDUN computers and their OS. But after some time it managed to complete a test, but with it taking long time and not having the capability of taking processing power from different computers like with IDUN made it somewhat obsolete to use in the long run.

3.2 Finding the mass of dark matter

The formulas for finding was given by our supervisor Ben David Normann, and the full paper can be found in Appendix A. the formulas are deduced from Clarkson 2016b which also gives a more detailed explanation[4]. Equation11 details how to find the critical density, which is substituted into equation 15 and equation16, these are the first and second derivative of the mass. Notably, the second derivative is zero the second derivative will of course be zero which confirms the increasing of the mass. These are then substituted for the mass values in equation 6 and 7. Solving these two new equation for μ_2^1 and μ_0^1, the end result will be the expression for finding the amplitudes for the mass. This process is essential for precisely modeling and predicting the behaviour of the system. This can be taken further with the machine learning, if the algorithm is good enough then finding a good approximation of the mass of dark matter will be possible.

Chapter 6

4 Conclusion

with the purpose of assisting the research group at NTNU on the way to map dark matter, we chose our goal for this project to be to optimize the machine learning algorithm by testing and training networks. The following will detail if we think we managed that or not.

4.1 Machine Learning

After the testing done and looking at the outputs and plots, we can with some confidence say that tuning the hyper parameters is a difficult process, with our test the best results came when using a somewhat high amount of epochs, more then 50 but less then 100, and with a learning rate between 0.001 and 0.0007 (At least with our testing).

Using the aforementioned ways to optimize the machine learning would also be a interesting test, just to see if it could give any different results. The algorithm was sensitive when changing the learning rate, so seeing how a dynamic learning rate could impact the outputs could be interesting.

There is also simpler ways of tuning hyper-parameters that probably should have been looked into more thoroughly then just brute forcing the way it was done from us. Ways like grid search and random search can be time consuming but would be able to get the best out of the machine learning if done properly.

Other hyper parameters to test could have been tested more were changing the amount of hidden layers in the algorithm or even changing the amount neurons within each layers. These processes were looked into to some extent, but a lot of the testing ended in a bunch of debugging and error codes, most likely due to the lack off knowledge on the subject and lack of time set aside to look into fixing it.

After changing every hyper parameter and testing the AlexNet and getting no decent outputs, it might have been a good idea to try a different network, networks with similar use cases are ResNet, but being stuck trying to make the Inception network work as good as possible might not have been the best use of time.

Some of the latter testing, which included the testing of different numbers of hidden layers or changing the number of neurons was also attempted, but at the end of the research time IDUN was under heavy load and getting time on the GPUs was difficult, and under a development phase were tests could last seconds and spit out an error code but the wait was up to and hour, there was not enough time, even after changing the requirements to not look for a specific GPU and just taking one that is free.. In retrospect it should have been implemented on personal computers for the debugging and development.

4.2 Cosmology

The cosmology part of this project introduced several complex and previously unfamiliar formulas and concepts, such as Einsteins radius and Graviational Lensing. The next step on this part will be using the machine learning to determine the mass of dark matter. by integrating the discussed formulas with the machine learning algorithms outputs it should be possible to estimate the mass with some confidence, although this will be another substantial task this project sets a good foundation for.

4.3 What we learned

As engineering students in a technology field we dipped our toes in machine learning, we had one semester of a subject covering machine learning. But this was nothing more than a little introduction compared to what we learned from this project. The machine learning part of this project required a lot coding, testing and training. The coding was quite challenging, it was mostly new technology and new code, in the form of TensorFlow for machine learning and using IDUN and its Linux operating system and ways of submitting jobs.

What we did not have any prior knowledge about is cosmology, even though the machine learning was difficult to understand and a lot of new learning material. The cosmology was even more challenging. Learning not only the basics of space and universe, but understanding a fairly new concept within cosmology is also not an easy feat. We gained a bigger understanding of the universe and truly how fascinating it is, but with that came also the understanding of how truly complex the universe is.

Bibliography

[1] *Advanced Guide to Inception v3*. 2023. URL: https://cloud.google.com/tpu/docs/inception-v3-advanced (visited on May 3, 2024).

[2] Geoffrey E. Hinton Alex Krizhevsky Ilya Sutskever. *ImageNet Classification with Deep Convolutional Neural Networks*. URL: https://papers.nips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf (visited on May 6, 2024).

[3] Jason Brownlee. *Understand the Impact of Learning Rate on Neural Network Performance*. URL: https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/ (visited on May 15, 2024).

[4] Chris Clarkson. *Roulettes: A weak lensing formalism for strong lensing - II. Derivation and analysis*. URL: https://arxiv.org/abs/1603.04652 (visited on May 19, 2024).

[5] Arthur B. Congdon and Charles R. Keeton. *Principles of gravitational lensing: Light replection as a Probe of Astrophysics and Cosmology*. Springer Praxis books, 2018.

[6] Siddharth Mishra-Sharma Cora Dvorkin. *Machine Learning and Cosmology*. URL: https://arxiv.org/pdf/2203.08056 (visited on May 13, 2024).

[7] Ben David Normann Hans Georg Schaathun. *On the simulation of gravitational lensing*. URL: https://www.researchgate.net/publication/369414016_On_the_simulation_of_gravitational_lensing (visited on May 1, 2024).

[8] Kenny Solevåg-Hoti Hans Georg Schaathun Ben David Normann. *Om å kartleggja mørk materie medmaskinlæring*. URL: https://www.ntnu.no/ojs/index.php/nikt/article/view/5666/5106 (visited on May 1, 2024).

[9] Lynn Cominsky Kim Coble Kevin McLin. *Lensing by Point Masses*. URL: https://phys.libretexts.org/Bookshelves/Astronomy__Cosmology/Big_Ideas_in_Cosmology_%28Coble_et_al.%29/12%3A_Gravitational_Lenses/12.02%3A_Lensing_by_Point_Masses (visited on May 7, 2024).

[10] Modestas Sukarevicius Kristian Lynghjem Vegsund. *Intergalactic Machine Vision*. URL: https://ntnuopen.ntnu.no/ntnu-xmlui/discover?scope=%2F&query=Intergalactic+Machine+Vision&submit= (visited on Apr. 24, 2024).

[11] Tarek M. Taha Md Zahangir Alom. *The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches*. URL: https://arxiv.org/pdf/1803.01164 (visited on Apr. 30, 2024).

[12] Kizito Nyuytiymbiy. *Parameters and Hyperparameters in Machine Learning and Deep Learning*. URL: https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac (visited on May 2, 2024).

[13] Keiron O'Shea and Ryan Nash. *An Introduction to Convolutional Neural Networks*. URL: https://arxiv.org/pdf/1511.08458 (visited on Apr. 22, 2024).

[14] Neri Van Otten. *Understanding dropout in neural network: Enhancing robustness and generalization*. May 2024. URL: https://spotintelligence.com/2023/08/15/dropout-in-neural-network/#How_does_dropout_work_in_a_neural_network.

[15] S Rezaei et al. "A machine learning based approach to gravitational lens identification with the International LOFAR Telescope". In: *Monthly Notices of the Royal Astronomical Society* 517.1 (July 2022), pp. 1156–1170. ISSN: 0035-8711. DOI: 10.1093/mnras/stac2078. eprint: https://academic.oup.com/mnras/article-pdf/517/1/1156/46441535/stac2078.pdf. URL: https://doi.org/10.1093/mnras/stac2078.

[16] Grant Sanderson. *Neural network*. URL: https://www.3blue1brown.com/lessons/neural-networks (visited on Apr. 22, 2024).

[17] Hans Georg Schaathu. *CosmoML*. URL: https://github.com/CosmoAI-AES/CosmoML/tree/master (visited on May 19, 2024).

[18] Hans Georg Schaathu. *CosmoSim*. URL: https://github.com/CosmoAI-AES/CosmoSim/actions (visited on May 19, 2024).

[19] Sondre Westbø Remøy Simon Ingebrigtsen. *CosmoAI: A study of gravitational lensing through simulation and machine learning*. URL: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/3003634 (visited on Apr. 24, 2024).

[20] Joshua Wilde et al. "Detecting gravitational lenses using machine learning: exploring interpretability and sensitivity to rare lensing configurations". In: *Monthly Notices of the Royal Astronomical Society* 512.3 (Feb. 2022), pp. 3464–3479. ISSN: 0035-8711. DOI: 10.1093/mnras/stac562. eprint: https://academic.oup.com/mnras/article-pdf/512/3/3464/43249501/stac562.pdf. URL: https://doi.org/10.1093/mnras/stac562.

Appendix

A time list

Lars

Dato	Lars	Timer	Gjøremål
10.01.2024	kl.12-16	4	forprosjektrapport
11.01.2024	kl.11-15	4	forprosjektrapport
12.01.2024	kl.12-14	2	forprosjektrapport
16.01.2024	kl.14.15-30	1.50	møte med veileder
23.01.2024	kl.13-13.30	1.50	internt møte
24.01.2024	kl.14-18	4	forprosjektrapport
26.01.2024	Kl.09.00-10.00	1.00	møte med veileder
30.01.2024	kl.11-16	5	simulator
01.02.2024	kl.10-16	6	simulator
02.02.2024	kl.10-16	6	Simulator + maskinlæring
07.02.2024	kl.10-16	6	Simulator + maskinlæring
08.02.2024	kl.10-16	6	Research
09.02.2024	kl.10-16	6	Research+møte
10.02.2024	kl.10-16	6	Research
13.02.2024	kl.10-16	6	Simulator + maskinlæring
14.02.2024	kl.11-17	6	Simulator + maskinlæring
15.02.2024	kl.10-15	5	Simulator + maskinlæring
16.02.2024	kl.10-15	5	Simulator + maskinlæring
19.02.2024	kl.10-15	5	Simulator + maskinlæring
20.02.2024	kl.12-16	4	Simulator + maskinlæring
21.02.2024	kl.10-18	8	Simulator
03.07.2024	kl.11-15	4	Maskin Læring
21.03.2024	kl.12-15	3	Maskin Læring
22.03.2024	kl.10-15	5	Maskin Læring
24.03.2024	kl.10-15	5	Maskin Læring
25.03.2024	kl.10-16	6	Maskin Læring
26.03.2024	kl.10-16	6	Maskin Læring
27.03.2024	kl.10-16	6	Maskin Læring

28.03.2024	kl.10-15	5	Maskin Læring
02.04.2024	kl.10-18	8	Simulator
03.04.2024	kl.10-16	6	Simulator
04.04.2024	kl.10-15	5	Simulator
05.04.2024	kl.10-15	5	Simulator + rapport
08.04.2024	kl.10-15	5	Simulator + rapport
09.04.2024	kl.10-15	5	Maskin Læring
10.04.2024	kl.11-15	4	Maskin Læring
11.04.2024	kl.10-15	5	Maskin Læring
12.04.2024	kl.10-15	5	Maskin Læring
15.04.2024	kl.10-15	5	Maskin Læring
16.04.2024	kl.15-20	5	Maskin Læring
17.04.2024	kl.10-18	8	Maskin Læring
18.04.2024	kl.11-17, 19-21	10	Maskin Læring
19.04.2024	kl.9-15	6	Maskin Læring
21.04.2024	kl.10-16	6	Maskin Læring
22.04.2024	kl.10-16	6	Maskin Læring
23.04.2024	kl.10-16	6	Maskin Læring
24.04.2024	kl.10-16	6	Maskin Læring
25.04.2024	kl.10-18	8	Maskin Læring
26.04.2024	kl.10-16	6	Maskin Læring
27.04.2024	kl.10-16	6	Maskin Læring
28.04.2024	kl.10-18	8	Maskin Læring
29.04.2024	kl.10-16	6	Maskin Læring
30.04.2024	kl.10-16	6	Maskin Læring
01.05.2024	kl.10-16	6	Testing
02.05.2024	kl.09-18	9	Testing
03.05.2024	kl.10-18	8	Testing
04.05.2024	kl.10-18	8	Testing
05.05.2024	kl.10-18	8	Testing
06.05.2024	kl.10-18	8	Testing
07.05.2024	kl.10-18	8	Testing + Rapport
08.05.2024	kl.10-16	6	Rapport
11.05.2024	kl.12-15	3	Rapport
12.05.2024	kl.12-16	4	Poster

13.05.2024	kl.18-23	5	Testing + Rapport
14.05.2024	kl.10-18	8	Rapport
15.05.2024	kl.10-15	5	Testing + Rapport
16.05.2024	kl.10-18	8	Rapport
17.05.2024	kl.10-18	8	Testing + Rapport
18.05.2024	kl.10-18	8	Rapport
19.05.2024	kl.10-16	6	Rapport
20.05.2024	kl.10-00	14	Rapport
Sum		**414**	

Mahammed

Dato	Mahammed	Timer	Gjøremål
10.jan	kl.12-18	6	forprosjektrapport
11.jan	kl.11-15	4	forprosjektrapport
12.jan	kl.14-17	3	forprosjektrapport
16.jan	kl.14-15.30	1.50	møte med veileder
23.jan	kl.13-13.30	1.50	internt møte
24.jan	kl.14-16	2	forprosjektrapport
26.jan	kl.09.00-10.00	1	møte med veileder
30.jan	kl.12-13	1	simulator
01.feb	kl.10-15	5	simulator
02.feb	kl.10-15	5	Simulator + maskinlæring
07.02.2024	kl.10-15	5	Simulator + maskinlæring
08.02.2024	kl.10-15	5	Research
09.02.2024	kl.10-15	5	Research+møte
10.02.2024	kl.10-15	5	Research
13.02.2024	kl.12-16	4	Simulator + maskinlæring
14.02.2024	kl.12-16	4	Simulator + maskinlæring
15.02.2024	kl.10-15	5	Simulator + maskinlæring
16.02.2024	kl.10-15	5	Simulator + maskinlæring
19.02.2024	kl.10-15	5	Simulator + maskinlæring
20.02.2024	kl.12-16	4	Simulator + maskinlæring
21.02.2024	kl.12-16	4	Simulator + maskinlæring

22.02.2024	kl.10-15	5	Research+rapport
23.02.2024	kl.10-15	5	Research+rapport
26.02.2024	kl.10-15	5	Research+rapport
27.02.2024	kl.12-16	4	Research+rapport
28.02.2024	kl.12-16	4	Research+rapport+møte
29.02.2024	kl.10-15	5	Simulator + rapport
04.03.2024	kl.10-15	5	Maskin Læring
05.03.2024	kl.12-15	3	Maskin Læring
06.03.2024	kl.12-15	3	Research+rapport
07.03.2024	kl.12-15	3	Research+rapport
08.03.2024	kl.12-15	3	Research+rapport
14.03.2024	kl.10-15	5	Maskin Læring
15.03.2024	kl.10-15	5	Maskin Læring
18.03.2024	kl.10-15	5	Maskin Læring
19.03.2024	kl.10-15	5	Maskin Læring
20.03.2024	kl.10-15	5	Maskin Læring
25.03.2024	kl.12-15	3	Research
26.03.2024	kl.12-16	4	Research
27.03.2024	kl.12-15	3	Research
03.04.2024	kl.10-15	5	Maskin Læring
04.04.2024	kl.10-15	5	Maskin Læring
05.04.2024	kl.10-15	5	Maskin Læring
08.04.2024	kl.10-16	6	Maskin Læring
09.04.2024	kl.10-16	6	Maskin Læring
10.04.2024	kl.10-16	6	simulator
11.04.2024	kl.12-15	3	simulator
12.04.2024	kl.11-16	5	simulator
14.04.2024	kl.10-11.30	1.5	møte med veileder
15.04.2024	kl.11-16	5	simulator
16.04.2024	kl.11-16	5	maskin læring +rapport
17.04.2024	kl.10-15	5	maskin læring +rapport
18.04.2024	kl.10-15	5	maskin læring +rapport
19.04.2024	kl.10-15	5	maskin læring +rapport
20.04.2024	kl.10-15	5	maskin læring +rapport
21.04.2024	kl.10-16	6	maskin læring +rapport

22.04.2024	kl.10-16	6	maskin læring +rapport
23.04.2024	kl.10-16	6	maskin læring +rapport+møte
24.04.2024	kl.10-16	6	maskin læring +rapport
25.04.2024	kl.10-16	6	maskin læring +rapport
26.04.2024	kl.10-16	6	maskin læring +rapport
27.04.2024	kl.10-16	6	maskin læring +rapport
28.04.2024	kl.10-16	6	maskin læring +rapport
29.04.2024	kl.10-16	6	maskin læring +rapport+møte
30.04.2024	kl.10-16	6	maskin læring +rapport
01.05.2024	kl.10-16	6	Rapport
02.05.2024	kl.10-16	6	Rapport
03.05.2024	kl.10-16	6	Rapport
04.05.2024	kl.10-18	8	Rapport
05.05.2024	kl.13-20	7	Rapport
06.05.2024	kl.10-16	6	Rapport
07.05.2024	kl.10-16	6	Rapport
12.05.2024	kl.12-23	11	Rapport+poster
13.05.2024	kl.09-17, 20-23.00	11	rapport
14.05.2024	kl.10-20	10	rapport
15.05.2024	kl.10-20	10	Rapport
16.05.2024	kl.10-20	10	Rapport
18.05.2024	kl.10-20	10	rapport
19.05.2024	kl.12-00	12	rapport+presentasjon
20.05.2024	kl.10-00	14	rapport+presentasjon
Sum		**412.5**	

B Attachments

the followings are can be found in attachments:

Meeting notes

The following are notes we took from some of the meetings we had with the supervisors, it is important to note that we had more meetings then the ones shown on the notes. Since one of our supervisor got sick we had a lot of our talks with him though mail and teams, there was also other instances where we just had ”small talks” that did not need to be noted.

making sense of the roulette formalism

This contains complete instructions on how to use the equations that we mentioned in chapter three about mass reconstruction. These are also more detailed and was given to us by one of our supervisors.

Other results

there are other results we obtained through test and training. These are categorized in two main folders, AlexNet and Inception, and further into more folders based on the types of learning rates.

Pre-project report

This is an report that we wrote before we started with the project, this report detailed how and what we was going to do to reach our goal in this thesis.

progress report

These are short reports on how the progress went through the months we was working on the project.