Energy reconstruction with artificial neural networks on LDMX simulations

Daniel Magdalinski

Thesis submitted for the degree of
Bachelor of Science

Project duration: 4 months-halftime

Supervised by Ruth Pöttgen

LUND UNIVERSITY

Department of Physics
Division of Particle Physics
Lund University

VT 2020

Abstract

It is clear from evidence such as rotational curves and cosmic microwave background measurements that dark matter exists. The light dark matter experiment (LDMX) will search for dark matter in the sub-GeV range. It will do this using missing-momentum measurements of electrons interacting with a Tungsten target. The electron will recoil and be measured in the electromagnetic calorimeter (ECal) of the experiment. The accuracy of this measurement is vital for the result of the experiment. Therefore, the ECal design will draw from the Phase-II high granularity upgrade of the Compact muon solenoid (CMS) forward ECal.

This thesis have investigated the possibility of using artificial neural networks (ANNs) to improve the energy resolution of the ECal. This was performed on simulation data based on the LDMX framework. Both convolutional neural networks (CNNs) and dense neural networks (DNNs) were trained on the data and compared with a linear fit between ECal readout energy and the original electron energy.

The analysis have shown that CNNs can improve the energy resolution of the ECal compared to both the DNN and linear fit who perform similarly. Some inconsistencies in how the models performed on different energies was discovered. Finally, solutions to this and suggestions for future work is discussed.

List of abbreviations

ANN = Artifical neural network
CNN = Convolutional neural network
DNN = Dense neural network
QED = Quantum electron dynamics
QCD = Quantum chromo dynamics
LDMX = Light dark matter experiment
DM = Dark matter
ECal = Electromagnetic calorimeter
HCal = Hadronic calorimter
QFT = Quantum field theory
WIMP = Weakly interacting massive particle
SM = Standard model
CMS = Compact muon solenoid
MSE = Mean squared error
MLP = Multi layered perceptron
SGD = Stochastic gradient descent

Contents

1 Introduction

Particle physics attempts to describe the smallest form of matter and how it interacts. The Standard Model (SM) governs our current understanding of particle physics. There is however a lot of discoveries left to be made. Dark matter (DM) is one problem that the SM has yet to find a confirmed solution to. There exist much evidence for the presence of dark matter in the Universe. The first was formed by Fritz Swicky [1] in the 1930s. He observed that galaxies in the Coma Cluster were moving too fast for the visible matter that he could observe. Rotational curves of galaxies were studied by Vera Rubin and William Kent Ford Jr [2] in the 1970s. The velocity measured in these curves should decrease steadily towards the edge of the visible matter in the galaxies. In contrast to this hypothesis, the curves were mostly flat indicating a halo of dark matter surrounding the galaxies.

The light dark matter experiment [3] (LDMX) will as its primary focus try to detect light thermal dark matter in the mass range of $\sim$ MeV $-$ GeV. The experimental design is unique in its current form and the experiment will implement an extensive and promising search for light dark matter. The experiment will perform a missing-momentum measurement with a 4-16 GeV electron beam impinging on a Tungsten fixed target. The signal event is produced through "dark bremsstrahlung" where dark matter particles carry away a large fraction of the electron energy. This electron recoils of the target with transverse momentum and is measured in the electromagnetic calorimeter (ECal). If a big fraction of the electron energy is missing from the measurement it could be a signal of dark matter detection. A hadronic calorimeter is situated behind and beside the ECal to veto background events were the electron interacts with the target to create non dark matter particles. The energy and momentum measurement of the electron is very important to the sensitivity of the experiment. The ECal design is therefore based on upgrade on the CMS forward ECal [4], which has a very high granularity.

Artificial neural networks (ANNs) is a very prominent machine learning technique that emulate how the neurons in our brain works. They can approximate complex functions by using many connected nodes. CNNs is one type of ANN that have had a rise in the recent decade with improvements within model architecture as well as in CPU and GPU performance. CNNs are extensively used for image analysis which makes it suitable for tasks that are image-like. The aim of this thesis is therefore to attempt to study the energy resolution (i.e. how well the calorimeter can measure a showering particle's energy) of the LDMX ECal with convolutional neural networks (CNNs). The high granularity in the LDMX design gives good grounds to enable the high pattern recognition ability of CNNs. Previous attempts at this task has also shown promising results [5, 6, 7], although at different energy ranges.

The thesis will cover the relevant physics in chapter 2 and 3, followed by a description of LDMX in chapter 4. Chapter 5 will give an overview in how neural networks work. The methodology of the thesis is covered in chapter 6 and the analysis in the form of results and discussion in chapter 7 and 8. Lastly, conclusion are given in chapter 9 with an outlook and overview of further studies in chapter 10.

2 The Standard Model

The standard model describes the different particles in nature and how they interact through forces. The particles are divided into two main groups: fermions and bosons. Fermions are half integer spin particles that follow Fermi-Dirac statistics. Particularly, this means that they follow the Pauli exclusion principle which says that two indistinguishable fermions cannot occupy the same quantum state. The fermions are further divided into leptons and quarks. Leptons consist of the electron, muon and tauon as well as their neutrino counterpart and the anti-particles as shown in the lower part of figure 1. Quarks are divided into six flavours: up, down, charm, strange, top and bottom and their respective anti-particles. These are, similar to the leptons, divided into three generations. The mass of the particles increase with increasing generations. All of the visible matter is made up of fermions. The other group called bosons are the mediators of the forces that determine the interactions between the fermions. The bosons have integer-spin and follow Bose-Einstein statistics, which therefore means they can occupy the same quantum state. The mediators of the forces are called gauge bosons and have spin-1. There also exists a spin-0 boson called called the Higgs boson which is the particle that give mass to the others [8].

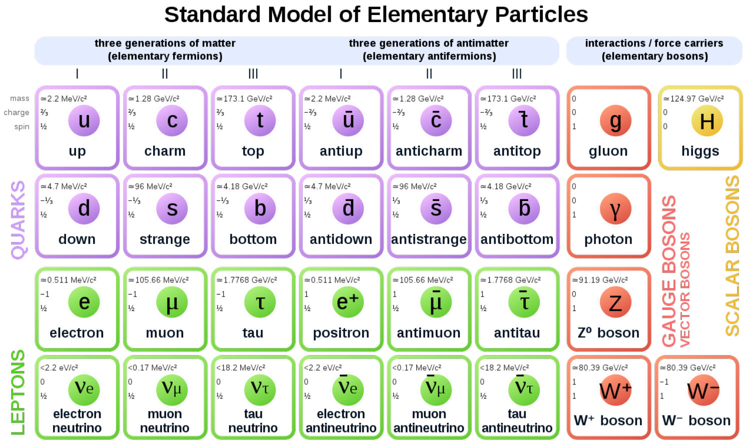

Standard Model of Elementary Particles
three generations of matter (elementary fermions)
three generations of antimatter (elementary antifermions)
interactions / force carriers (elementary bosons)
I
II
III
I
II
III
mass
charge
spin
≃2.2 MeV/c² ⅔ ½ u up
≃1.28 GeV/c² ⅔ ½ c charm
≃173.1 GeV/c² ⅔ ½ t top
≃2.2 MeV/c² −⅔ ½ ū antiup
≃1.28 GeV/c² −⅔ ½ c̄ anticharm
≃173.1 GeV/c² −⅔ ½ t̄ antitop
0 0 1 g gluon
≃124.97 GeV/c² 0 0 H higgs
≃4.7 MeV/c² −⅓ ½ d down
≃96 MeV/c² −⅓ ½ s strange
≃4.18 GeV/c² −⅓ ½ b bottom
≃4.7 MeV/c² ⅓ ½ d̄ antidown
≃96 MeV/c² ⅓ ½ s̄ antistrange
≃4.18 GeV/c² ⅓ ½ b̄ antibottom
0 0 1 γ photon
≃0.511 MeV/c² −1 ½ e electron
≃105.66 MeV/c² −1 ½ μ muon
≃1.7768 GeV/c² −1 ½ τ tau
≃0.511 MeV/c² 1 ½ e⁺ positron
≃105.66 MeV/c² 1 ½ μ̄ antimuon
≃1.7768 GeV/c² 1 ½ τ̄ antitau
≃91.19 GeV/c² 0 1 Z Z⁰ boson
<2.2 eV/c² 0 ½ νe electron neutrino
<0.17 MeV/c² 0 ½ νμ muon neutrino
<18.2 MeV/c² 0 ½ ντ tau neutrino
<2.2 eV/c² 0 ½ ν̄e electron antineutrino
<0.17 MeV/c² 0 ½ ν̄μ muon antineutrino
<18.2 MeV/c² 0 ½ ν̄τ tau antineutrino
≃80.39 GeV/c² 1 1 W⁺ W⁺ boson
≃80.39 GeV/c² −1 1 W⁻ W⁻ boson
QUARKS
LEPTONS
GAUGE BOSONS
VECTOR BOSONS
SCALAR BOSONS

Figure 1: The particle in the standard model [9].

There exists four fundamental fources and among these two are easier experienced in the macro scale of everyday life. The graviational force is what holds us on the surface of the earth and the electromagnetic force can be seen acting everywhere in our current digital world. The gravitational force is the weakest of the forces which makes it hard to study within the field of particle physics. A mediator particle called the graviton has been theorized but have so far not been detected. Therefore the gravitational force is mainly understood on macro-scales where it has a noticeable effect, which means it is not described within the standard model. The electromagnetic force describes the interactions between electrical charges and the interactions between light and matter. This force was well described already in 1865 by James Clerk Maxwell. The force is mediated by the massless photon, which means that the range of the force is infinite although it decays with $\frac{1}{d^2}$, where d is the distance between two charges. After the arrival of quantum mechanics in the beginning of the 20th century a quantum field theory (QFT), quantum electrodynamics (QED) was formulated by Richard Feynman in the 1960s [10].

The source of why the strength of the electromagnetic force decays over distance in QED is called screening. This property comes from quantum fluctuation, where a particle can emit and absorb another particle during a very short time. This phenomenon is explained by Heisenberg's uncertainty principle, which states that the product of the uncertainty of two complementary observables in a system must be greater that $\frac{\hbar}{2}$. This can be applied to the energy and time of a system: $\Delta t \propto \frac{\hbar}{\Delta E}$. Therefore, a certain amount of energy can be created during a certain amount of time without violating conservation of energy. These quantum fluctuations lead to the fact that electrons continuously emit and absorb photons. These photons can split into electron-positron pairs, which screen the original charge of the electron. This means that at increasing distances the effective charge of the electron is reduced, thereby reducing the strength of the electromagnetic force [10].

Contrary to these long range forces the weak and strong nuclear forces only apply on short distances. The mediators of the weak force are the W^-, W^+ and Z^0 bosons. These mediator particle are all very heavy, about 80-90 times the mass of the proton. This means that the interaction length of the weak force is very small following the argument of quantum fluctuations. A higher mass means higher energy therefore the lifetime of the quantum fluctuated mediator particle will be shorter giving a shorter range to interact. The main example of the weak force is β^--decay, where a neutron decays into a proton, an electron and an electron antineutrino [8].

A theory called the electroweak theory combines the electromagnetic force and the weak nuclear force at high energies. The idea for the theory comes from the similar interactions mediated by the Z^0 and the γ. To unify these forces, two neutral bosons B and W^0 had to be introduced. The γ and Z^0 are combinations of the W^0 and B. This combination is achieved assuming gauge invariance under a certain transformation. A problem with this assumption is that gauge invariance implies that the spin-1 gauge bosons have zero mass. This is not the case for the weak force mediators. To combat this problem the Higgs field was theorized. Certain particles can interact with the Higgs field through the mediator spin-0 particle called the Higgs boson. The important property of the Higgs field is that it has a non-zero value in its vacuum state, which is not invariant under a gauge transformation. This breaks the symmetry of the electroweak theory, thus enabling the massive weak bosons [8].

The last force is the strongest, therefore the name the strong nuclear force. The strong force only interacts with the quarks and its own mediator particle, with a range of 10^{-15} m. Because of this short range of the force it was first theorized to have a heavy mediator particle similar to the weak force. The π-meson was discovered in 1947 and was thought to be the mediator particle. This idea was soon after discarded as the π-meson had too weak of an interaction to be the mediator particle. The interaction that the π-meson mediates is a residual effect of the strong force and is the interaction that binds protons and neutron together in the nucleus. During the following years more and more hadrons were discovered which lead to the idea that these were composite particles. The quark model was introduced in 1963 and was successful at explaining the currently observed hadrons. Two problems emerged with the initial model. Firstly, no observations of a single quark had been made, only composite particles made from them. Secondly, one hadron(Δ^{++}) seemed to imply that the quarks violated Pauli's exclusion principle. From these problems a new quantum variable called color was introduced. The QFT for the strong force is therefore called Quantum Chromo Dynamics (QCD). The color states were called red, green, blue as well as their three respective anti-colors. This solved the second problem because the quark could differ in the color quantum number. It also posed a solution to the first problem with the hypothesis that only color-neutral states could be observed. Color-neutral meaning a combination of the three colors or a combination of a color and its anti-color [10].

The mediator particle is the massless gluon. The color charge of a quark can be changed by the emission or absorption of a gluon. In order to conserve the color charge quantity the gluon must carry the initial color as well as the anti-color of the final color. This means that contrary to the electromagnetic force where the photon has no charge, the gluon will be able to self-interact. This self-interaction leads to two important properties of the strong force called color confinement and asymptotic freedom. Color confinement means that the force becomes stronger with increasing distance leading to the quarks being confined within hadrons. Asymptotic freedom means that the interactions gets weaker at shorter distances leading to the quarks behaving more like free particles. In QCD the phenomena of screening exists similar to QED as the gluon can pair produce into a quark-antiquark pair. Additionally, because of the self-interacting nature of the gluon it can also fluctuate into a gluon pair that anti-screens the charge of the original quark. The effect of anti-screening is much larger than screening which gives this property that the strong force coupling increases at long range and becomes lower at short range [10].

2.1 Calorimetry

Most particle physics experiments today uses some form of calorimetry to detect and measure energies of particles. A calorimeter is a block of material that absorbs particles and converts the energy absorbed into an electric signal. The interactions between the incoming particle and the calorimeter creates showers of secondary particles with lowering energy. Calorimeters are usually divided into an electromagnetic and a hadronic part. In this thesis the focus will be on the electromagnetic calorimeter so therefore we will only cover that type of calorimeters here [11].

The interactions between electromagnetic calorimeters and particles like photons and electrons are well understood through a few QED-processes. The average energy loss of an electron is illustrated in figure 2. From this it is clear that for energies above 20 MeV the process of bremsstrahlung dominates. This process occurs when electrons get deflected and decelerated by a positive nucleus, whereby the electron radiates a photon. At these energies pair-production is the dominant process for photons. These two processes together create a cascading effect of generating more and more photons and electron-positron pairs with decreasing energy into what is called a electromagnetic shower. At lower energies ionization and thermal excitation dominate for electrons and positrons, for photons it is Compton scattering and photoelectric effect. At these lower energies, energy is therefore dissipated slowly into the atoms of the material instead of cascading into new particles [11].

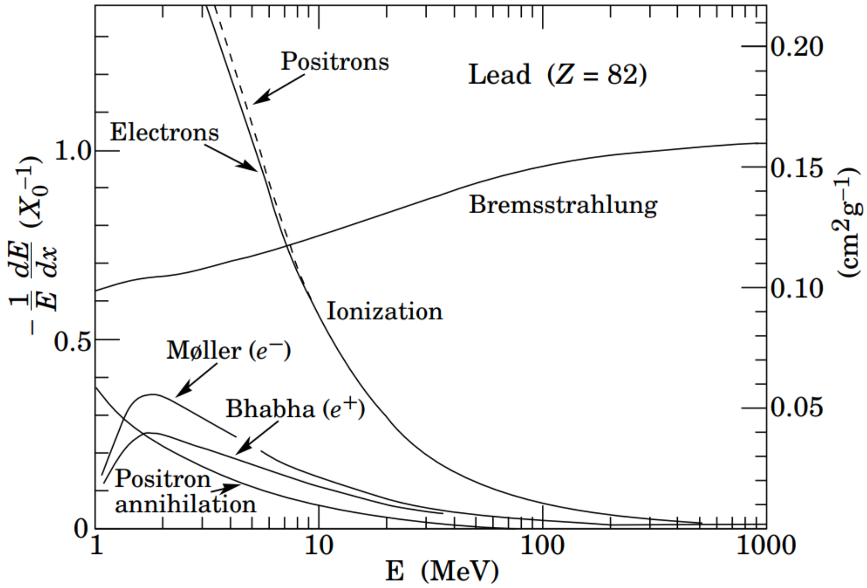


Figure 2: Fractional energy loss per radiation length for electrons and positrons in lead as a function of energy [12].

Two different kinds of ECals exists called homogeneous and sampling. Homogeneous calorimeters consist of one homogeneous medium that both absorbs and measures energy at the same time. Sampling calorimeters have instead two different mediums designed for absorbing and measuring. The main advantage of the homogeneous calorimeter is the increased energy resolution, which is a result of the particles depositing all of its energy within the measuring material. In contrast to the sampling case where some energy is always not measured as it deposits in the absorbing layer. The advantage of sampling calorimeters is that both materials can be optimized to its task. This also means that sampling calorimeters can be easier to segment given them better positional resolution and particle identification. Therefore sampling calorimeters is the main calorimeter used in detectors today [11].

A very important property of an electromagnetic calorimeter is the energy resolution. This is a measurement of how accurate the energy of an incident particle can be determined. The resolution is computed by first creating a distribution of the difference between the particles energy and the measured energy $\Delta E = E_{true} - E_{measured}$. This distribution should have a Gaussian form. By fitting a Gaussian to the distribution and taking the first standard deviation (this corresponds to 68% of the events) of this distribution one get a estimate of how wide the inaccuracy is. This sigma is then divided by the correct energy of the events to get the energy resolution. The resolution is usually expressed in three different terms,

$$\frac{\sigma(\Delta E)}{E} = \frac{a}{\sqrt{E}} \oplus \frac{b}{E} \oplus c \tag{1}$$

where $\oplus$ indicates a quadratic sum. The three terms on the right hand side represent different processes that can generate uncertainty in the energy resolution. The first term is called the stochastic term and represents fluctuations in the creation and absorption of shower particles. Differences in how the electron deposits its energy in the layers of sampling calorimeters will also add to the stochastic term. The second term is a noise term that comes from electronic noise in the readout electronics. This is therefore very dependent on the setup used and it becomes the dominant term over the stochastic at lower energies. The third term is a constant that includes all effects that do not depend on the energy of incident particle. These come from imperfections in the calorimeter material and readout system, aging, radiation damage, etc. This term is primarily important at higher energies where the other terms fall off [11].

3 Dark Matter

The first experimental evidence for dark matter was first discovered in the 1930s by Fritz Zwicky [1]. Zwicky observed that the galaxies in the Coma Cluster were moving faster than what the visible matter in the cluster would predict. From this he postulated that there must exists some form of dark matter that would contribute to the gravitational force needed for the faster orbits. Further evidence was put forth by Vera Rubin and William Kent Ford Jr [2] in the 1970s. They observed the rotational curves of galaxies similar to the one in figure 3, which is a plot of the angular velocity of the gas and stars as a function of the distance from the center. According to the visible mass in the galaxies these rotational curves where expected to decrease slowly as the distance from the center increased. Instead Rubin and Ford measured that the curve remained mostly flat in several galaxies. To explain this behavior, galaxies should have a halo of dark matter that adds the extra mass to support the higher velocities measured.

Additionally, evidence for dark matter have also come in the form of gravitational lensing measurements. Einstein's theory of general relativity predicts that mass can bend light. This phenomena can then be used to infer the existence of mass even when it can not be observed. These measurements generated evidence that the dark matter in galaxies spread out much further than the visible matter in the galaxies. This had not previously been observed as rotational curves only give measurements were there is visible mass.
The estimation of how much dark matter exists in the Universe comes from measurements of the cosmic microwave background (CMB). The CMB is remnant radiation originating from the hot early Universe. Measurements of this radiation gives valuable insight into the early Universe and therefore estimations for many cosmological constants and densi-

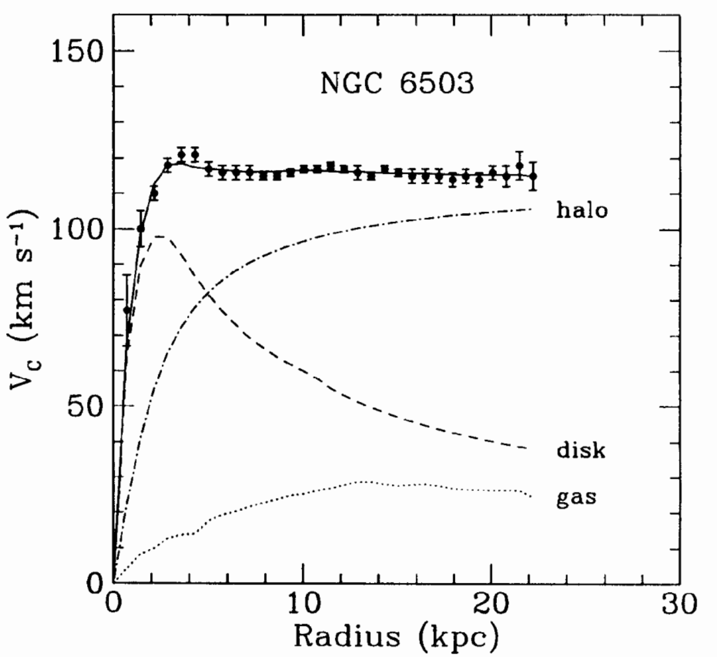


Figure 3: Galactic rotational curve of NGC 6503 showing the data collected and predicted contributions from disk matter, gas matter and a dark matter halo [13].

ties. These measurements gives an estimate that 5% of the energy in the Universe comes from visible matter and 26% from dark matter. The rest is called dark energy and is another mystery within physics that will not be further discussed here [14].

One of the most common theories for dark matter is that it comes from non-gravitational interactions during the early hot Universe. If the DM-SM interaction rate exceeds the Hubble expansion rate during the early Universe, a lingering concentration of DM is assured. This theoretical scenario is called thermal dark matter and has an allowed mass range between $\sim$ MeV$-$ $\sim$ 10 TeV [3]. In the upper half of that mass window $\sim$ GeV$-$ $\sim$ 10 TeV is one of the most promising candidates for dark matter called weakly interacting massive paricles (WIMPs). These theorized particles that interact through a force of similar strength to the electroweak force, already existed in another field of theoretical particle physics called super symmetry. Therefore WIMPs were seen as an excellent candidate that would solve two questions at once. So far no conclusive detection evidence of WIMPs has been found [14]. The lower half of the mass window $\sim$ MeV $-$ GeV has been hard to probe with traditional experiments. This scenario has been raised as critical to test for in both the CERN and US particle physics strategy reports [15, 16]. The scenario is motivated by "hidden-sector" [17] theories where dark matter is neutral under SM and interacts trough a dark mediator particle like the dark photon mentioned in the next section, but has enough of a coupling to ordinary matter to enable the interactions needed for the creation of dark matter during the early Universe.

4 Light Dark Matter eXperiment (LDMX)

The primary goal of LDMX [3] is to search for thermal light dark matter in the sub-GeV range but also with sensitivity for other dark matter particles at a mass range extending to below keV. This is done through a missing-momentum measurement in a fixed-target experiment with an electron beam. The momentum of the outgoing particles is measured and compared to the incident electron. If a difference is observed it could indicate the production of a new particle. The theorized processes for producing dark matter are illustrated in figure 4. The two processes occur through either a dark matter version of bremsstrahlung followed by decay of the dark photon into dark matter particles or direct production. This particle can not be detected with the detector setup so there will be missing momentum in the measurement [3].

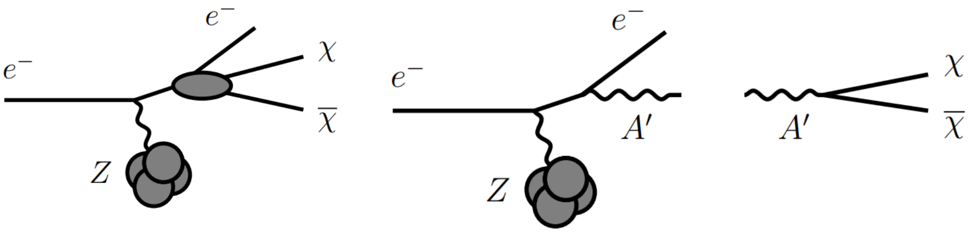


Figure 4: The two processes that LDMX especially wants to look for. On the left is direct darkmatter production and on the right production through a mediator particle [3].

A dark matter signal contains three distinct features, these include a recoiled electron(i.e. a electron that interacts with the target and deflects with reduced energy) with energy far less than the beam energy but detectable, non-zero transverse momentum and a lack of signal in the remainder of the detector setup. To perform an accurate measurement of the recoil electron, the setup that can be viewed in figure 5 will consist of a recoil tracker

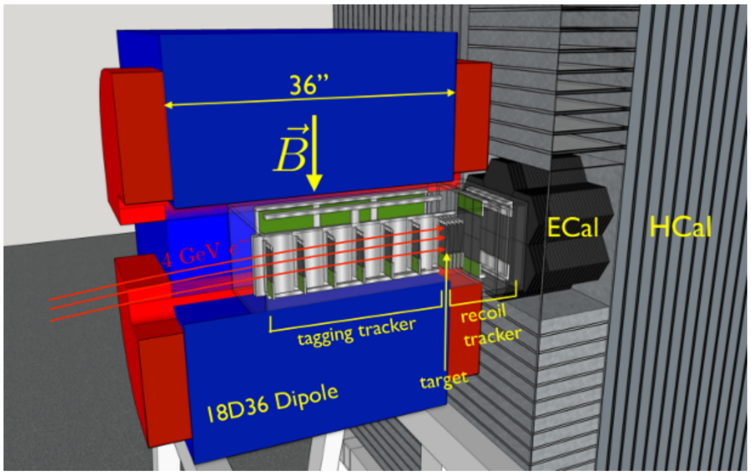


Figure 5: A cut-through view of the detector setup for LDMX, showing the guiding 18D36 dipole magnet as well as tracking, target, ECal and HCal [3].

following the target followed by a electromagnetic calorimeter (ECal). The recoil tracker will measure the path and the momentum of the outgoing electron. The ECal, a high granularity sampling calorimeter will mainly measure the energy of the recoiling electron as well as other particles created in the target that shower electromagnetically. A tagging tracker upstream of the target aims to veto events with stray low-energy electron coming from beam defects. A hadronic calorimater (HCal) follows the ECal to veto background events where hadronic particles are created that would escape the ECal. To achieve sufficient statistics for the experiment, the beam is proposed to have a high-bunch repetition ($\approx$ 40MHz) with 10^8 electrons per second with very few electrons interacting with the detector setup at the same time. The Phase-I beam energy will be 4 GeV, following by a increase in steps to potentially 16 GeV to be able to probe most of the sub-GeV dark matter candidates [3].

The ECal design for LDMX draws from the Phase-II high granularity calorimeter upgrade for the CMS detector [4]. The Tungsten-Silicon sampling calorimeter design has a hexagonal sensor structure and the LDMX setup will contain 32 layers with 7 hexagons as shown by figure 6. The CMS design will have a maximum of 320 μm thick silicon, to get better energy resolution and better detection of hadrons LDMX will increase the thickness to $\approx 500 - 700$ μm. The highest granularity considered for the CMS design is $\approx$ 432 pads with an area of 0.52 cm^2 on a sensor made out of a 8" wafer [3].

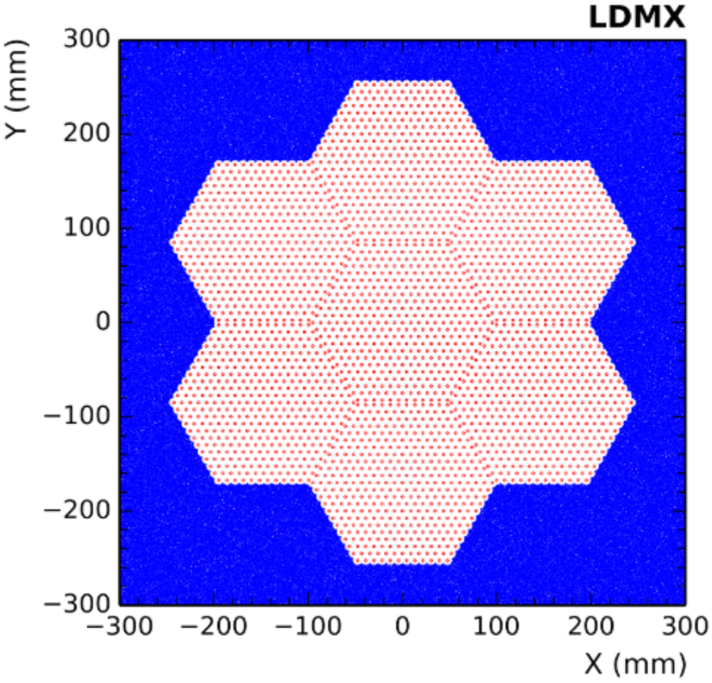


Figure 6: The xy distribution of the ECal cell centers are shown in red. The ECal is clearly structured in hexagonal divisions [3].

5 Artificial Neural Networks

Artificial neural networks(ANN) is a machine learning technique that is inspired after the neurons in our human brain. The network consists of multiple connected nodes that add up to construct a complicated functionality. The network is trained on data with known targets to create an approximate function that maps the input to the target output. In this thesis all information and details about ANNs are gathered from lectures notes by M. Ohlsson and P. Edén [18].

ANNs have mainly been used for regression tasks, classification and in recent years image recognition and generation through CNNs. In regression problems the goal is to predict a certain value from the input data. This can be a simple example like linear regression or a more complicated function like the one attempted in this project, mapping the energy of a shower in a detector from the calorimeter hit data. Classification is a task where the network tries to label the data according to the correct class label. An example of this is the very popular MNIST problem where pictures of handwritten numbers between 0-9 are labeled by the ANN. These are all examples of supervised learning were the data comes with labels. There are many other learning methods that will not be covered here.

5.1 The Perceptron

The simplest form of a neural network is called a peceptron. An illustration of this construct is shown in figure 7.

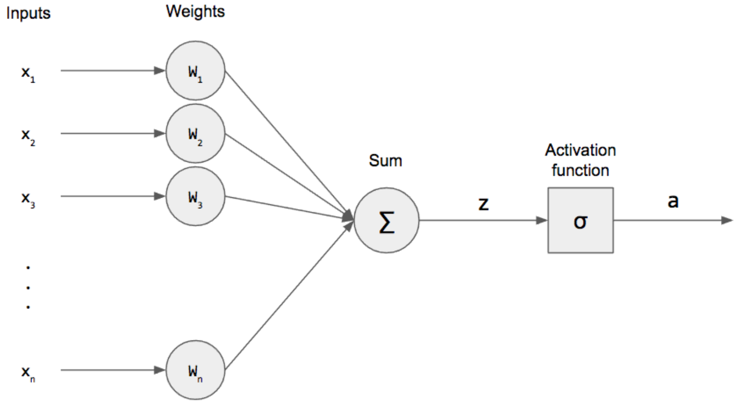


Figure 7: Illustration of the perceptron and how it computes output from input. Inputs are multiplied with the weights and summed up and then finally sent through an activation function [19].

The output y is computed by adding all the inputs x_i multiplied with their corresponding weight ω_i. To the sum a bias is also added to more easily enable shifts in the model function. To make notation simpler usually $x_0 = 1$ and ω_0 is the value of the bias.

$$y(x) = \sigma\left(\sum_{i=0}^{N} x_i \cdot \omega_i\right) \tag{2}$$

The activation function σ can take a few different forms. The linear activation function $\sigma(x) = x$ is the original function used and is always used as output activation in regression tasks where the value of the output is important. Rectified linear unit (ReLU) is another activation function shown in equation 3 which is very popular currently especially in use with convolutional neural networks (CNNs). These are the two activation functions used in this project but there are many other such as tangential, sigmoid and softmax.

$$\sigma(x) = \begin{cases} x \text{ if } x \geq 0 \\ 0 \text{ if } x < 0 \end{cases} \tag{3}$$

5.2 Multi layered perceptron

With the perceptron only basic problems that are linear can be solved. To achieve a non-linear function more layers need to be added with non-linear activation functions. The equation for a multi layered perceptron (MLP) with two layers, K inputs, J nodes in the hidden layer, n events and a single output is,

$$y(x_n) = \sigma_o \left(\sum_j^J \omega_j \cdot \sigma_h \left(\sum_k^K \widetilde{\omega}_{jk} x_{nk} \right) \right) \tag{4}$$

where h and o stands for hidden and output. The amount of layers and nodes are hyper parameters that can be adjusted depending on how complex a function the network attempts to learn.

5.3 Training the network

As previously mentioned the network is trained on labeled data where the answer is given. This is effectively done by computing an error between the output of the network y and the labels d. There are several error functions but the most common error function used in regression problems is the mean squared error (MSE). By convention a $\frac{1}{2}$ factor is introduced to the MSE to cancel the 2 that comes from the derivative.

$$E(\omega) = \frac{1}{2N} \sum_{n=1}^{N} (d_n - y(x_n))^2 \tag{5}$$

By minimizing E with respect to ω the function y can approximate the mapping that gives the labels d. The algorithm used for minimizing the error is called gradient descent. When it comes to the MLP this process is usually called back-propagation. The algorithm works by initiating all weights to a random value. Then the output $y(x_n)$ is computed for each input x_n. Then the gradient of the error function w.r.t the weights is computed and a step is taken in the negative direction. The weight update for each layer of weights from the MLP defined in equation 4 is,

$$\Delta\omega_j = -\eta \frac{\partial E}{\partial \omega_j} \text{ , } \Delta\widetilde{\omega}_{jk} = -\eta \frac{\partial E}{\partial \widetilde{\omega}_{jk}}. \tag{6}$$

, where η is called learning rate and defines how big the updates should be. This method averages the weight update over all inputs x_n because $E = \frac{1}{N}\sum_n E_n$, with $E_n = (d_n - y(x_n))^2$. Stocastic gradient descent (SGD) offers a slight variation to this. Instead of the full batch of all inputs, weights are updated in mini-batches instead,

$$\Delta\omega_j = -\eta\frac{1}{P}\sum_{p=1}^{P}\frac{\partial E_p}{\partial\omega_j}\ , \ \Delta\widetilde{\omega}_{jk} = -\eta\frac{1}{P}\sum_{p=1}^{P}\frac{\partial E_p}{\partial\widetilde{\omega}_{jk}}, \tag{7}$$

where P is the mini-batch size and is treated as a hyperparameter. The inputs are divided into different mini-batches and after one mini-batch update one iteration has been completed. After all iterations are done and the input set has been trained on once, an epoch has been completed. One might worry that SGD will give worse performance than gradient descent as taking all inputs to consideration would give a more consistent result. This is true but gradient descent can be very computationally heavy and also training too well is a problem for networks that is covered in section 5.4.

The data set that is used to train and evaluate an ANN model is usually divided into a training set, validation set and testing set. The biggest set is the training set that is used during training of the weights. The validation set is used during training after each epoch to test the performance of the network on data that it has not trained on. This data set is usually used to differentiate between different models and to search for optimal hyperparameters. Because the validation set is used in determining models and hyperparameters, the model might have bias to that data. Therefore, an additional test set is used to give an unbiased performance estimate of the model.

5.4 Generalization

The model can usually train very well but the network is built to be applied to other unseen data. Therefore, the generalization performance is important i.e. the performance of the ANN model on new unseen data. The essential problem within neural networks is the balance between underfitting and overfitting. Overfitting is when the network train so much on the training data that it performs worse on new data as illustrated by the left side

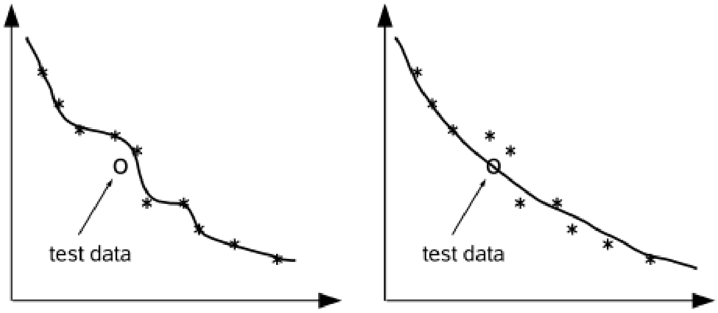


Figure 8: Left: The network gives a good fit on the training data but does not generalize well. Right: The network generalizes better to the new unseen data [18].

of figure 8. Underfitting is instead when the network performs bad on new data because it has not achieved a well enought mapping of the desired function. Balancing these two problems is the key to the networks generalization performance.

5.5 Dropout

One very prominent method used to generalize the network is dropout [20]. This method removes nodes during training with a probability p as illustrated by figure 9. Removing the node means that all input and output weights are not involved in that part of the training. This procedure is randomized each mini-batch so that each iteration trains on a slightly different network.

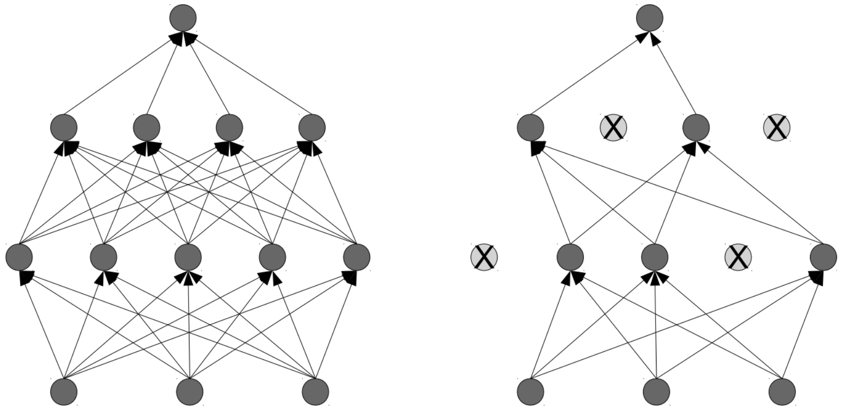


Figure 9: Left: An MLP with two hidden layers. Right: The same network but with dropout applied where some of the hidden nodes have been removed as well as their weights [18].

The method ensures that the network cannot train too well on the data set. The final network will be a complex ensemble of many slightly different networks. In fact there is a concept in neural networks called ensemble machines were an averaged output of several different network usually outperforms the best network of that set.

5.6 Convolution neural network (CNN)

CNNs are mostly used for image analysis or on data that is "image-like". CNNs utilise image filters instead of nodes that scan over the input data with the same set of weights applying to all of the input. The filter applies the process of convolution, in practise cross-correlation is done instead which is the same thing just that the output image will not be flipped. The equation for cross-correlation with $I(i,j)$ being the input image, $K(n,m)$ being the filter and $O(i,j)$ being the output,

$$O(i,j) = (I * K)(i,j) = \sum_n \sum_m I(i+n, j+m)K(n,m). \quad (8)$$

Illustrating this with a 3x3 image matrix cross-correlated by a 2x2 filter,

$$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} * \begin{bmatrix} 1 & 2 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} (1+8+2-5) & (4+14+5-8) \\ (2+10+3-6) & (5+16+6-9) \end{bmatrix} = \begin{bmatrix} 6 & 15 \\ 9 & 18 \end{bmatrix} \tag{9}$$

Note that the output image is smaller than the input image. Usually the image is padded with zeroes so that the output image retains the original size. Another operation called pooling will instead downsample the image, pooling will be covered in section 5.6.2.

5.6.1 Sparse connectivity and weight sharing.

The important properties that differentiate CNNs from normal ANNs is sparse connectivity and sharing of weights. If we imagine this in 1D, sparse connectivity is illustrated in figure 10. For the MLP every input adds to every hidden nodes through its weights. For the CNN this is restricted through the size of the filter. For the figure the filter has a size of 3 meaning that only three input values contributes to a hidden node.

Sharing of weights is illustrated by the color in the figure below. The convolution filter that scans the input stays the same for all inputs so that weights are shared between the inputs. The idea here is that properties of the image that are important for the network is important in equal ways over the entire image. This means that the CNN usually has way fewer weights to train for the same feature extraction on large images.

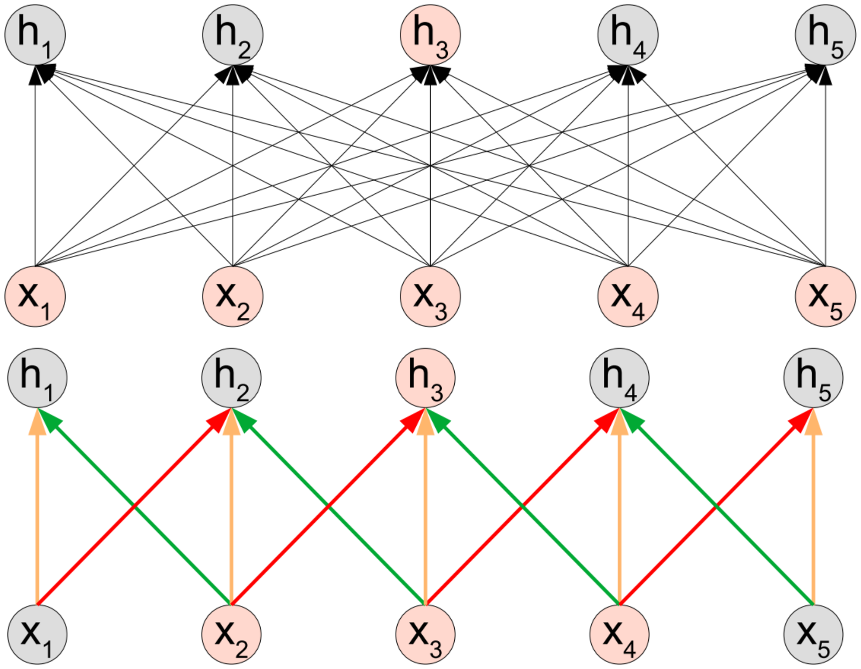


Figure 10: Above: The figure illustrates how weights behave within the MLP, all nodes are connected with indidividual weights. Below: For the CNN one hidden node is only connected to a limited amount of input nodes. Also the weights are shared illustrated by the colors so that the same weight is applied at multiple points [18].

5.6.2 Building the CNN

The CNN usually consist of layers with three components: the convolution stage, activation stage and pooling stage. The activation stage usually consists of a ReLU activation function which is also what will be implemented in this project. The pooling stage takes the values of the hidden layer and summarize them according to some algorithm. The most common pooling is max pooling which replaces the value of the node in the layer by the maximum in a given neighbourhood. This is illustrated on the left side of figure 11. Pooling is very often used to down-sample the data. This is illustrated in the right part of figure 11 where the maxpooling filter has a size of 3 and moves with 2 steps so that the size of the data is reduced from 7 to 3. The idea with pooling is to make the output invariant to small differences in the input.

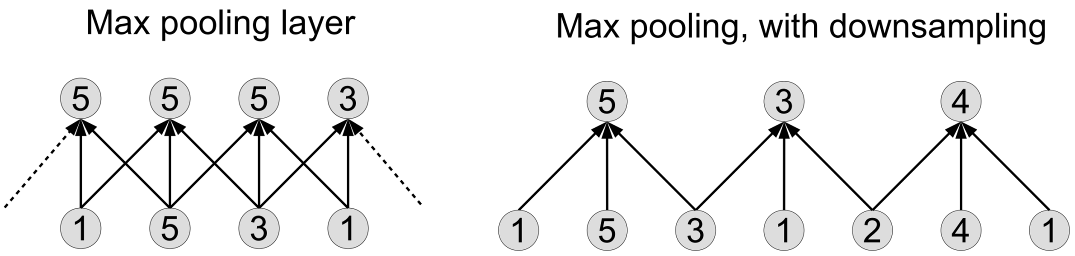


Figure 11: Left: The max pooling layer takes the input of in this case two nodes around the current node and replaces it with the maximum value. Right: Here the max pooling move with two nodes each time which means that the image is downsampled [18].

To finally build the complete CNN several parameters need to be considered. Firstly how many convolutional layers that will be used. For each layer the filter sizes for both the convolutional stage and the pooling stage. The amount of pixels that the filter and the pooling operations move, also called stride, is also a hyperparameter that needs to be determined.

6 Method

6.1 Software

The Keras API [21] is used running together with the machine learning library Tensorflow [22]. All code is written in Python with PyROOT [23] used to access the generated data. All code was run on the Aurora Cluster at the center for scientific and technical computing at Lund University (LUNARC).

6.2 Data

All the data used in this thesis is simulated using the Geant4-based [24] LDMX framework. Two data sets were generated with 100000 events from a linear 0-3 GeV distribution shown in figure 12 in red. This energy range was chosen as it would emulate the energy range of a recoil electron coming from a beam with an energy of either 4 GeV(0-1.5 GeV) or 8 GeV(0-3 GeV). Both sets generated an electron close to the ECal, 200 mm ahead of the target. This was done to ensure better control over the electron entering the ECal. The first data set has no angle distribution and will therefore enter the ECal in the center. The second set has an isotropic angle distribution between 160°- 180°or $\frac{17}{9}\pi - 2\pi$ for θ and 0°to 360°for ϕ. θ is defined as the angle to the z-axis with 180°being directed downstream towards the ECal and 0°being directed upstream towards the target. ϕ is defined as the angle on the transverse plane to the z-axis. The electron will therefore be generated in a 20°cone distribution towards the ECal. The x, y and z distributions for both these data sets are visible in figure 13. It's clear that the angle set has a wider distribution in x and y while the z distribution is almost identical. The discreetness from the values in figure 13 comes from the position of the ECal cells, the x-distribution is separated by 4.3 mm for all cells. The y-distribution is different depending on which hexagon in the ECal observed. Within a hexagon all y positions are separated by 7.5 mm. Between some ECal hexagons there is a shift in 2.5 mm for the y-distribution.

For both data sets two separate kinds of data exists. The simulation comes with ECal simulation hits, which are generated by how the particle interact with the ECal in the simulation and where each hit has a value that corresponds to the energy deposited. Using a reconstruction algorithm from the LDMX software hits more similar to actual calorimeter hits were created. With this reconstruction, layer weights are applied to the different z-layers, also some small noise are randomly added to the data. The data was converted from the initial data sets in ROOT [23] to 3D array in Numpy [25] format. The energy distributions of the sum of the ECal hits in these arrays is shown scaled to the range of the true values in figure 12 together with the true distribution. The dimensions in (x,y,z) were for the first set: (24,48,28) and for the second set: (32,54,26). The size of these array comes from looking at the x,y,z distributions and also from consideration of the memory size of the data. The cut into arrays therefore excludes some hits from the simulation, for the first set the fraction of include hits over all hits is 0.966 and for the second set it is 0.970.

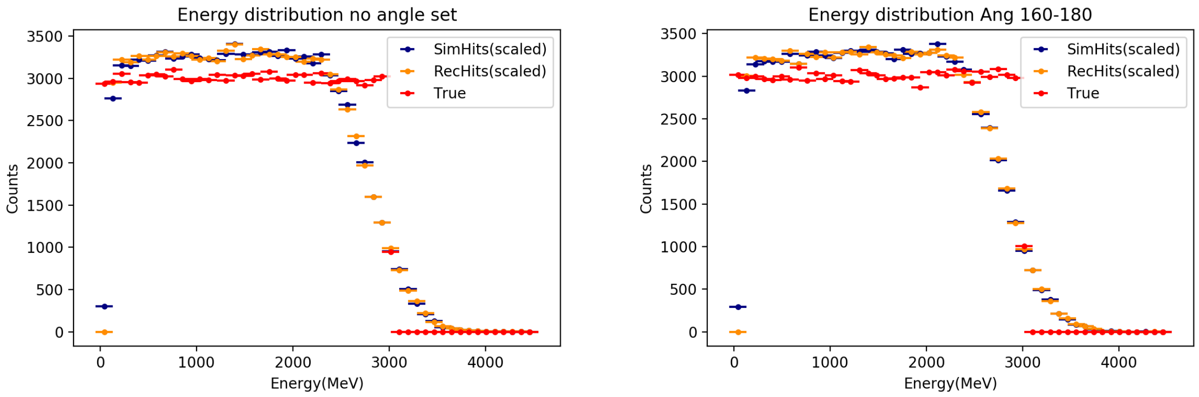


Figure 12: Energy distribution of the data sets with the set without angle dist. on the left and the 160°-180°set on the right. The distribution of the sim hits and recon hits array energy sum is also added and scaled to the same range as the true energy.

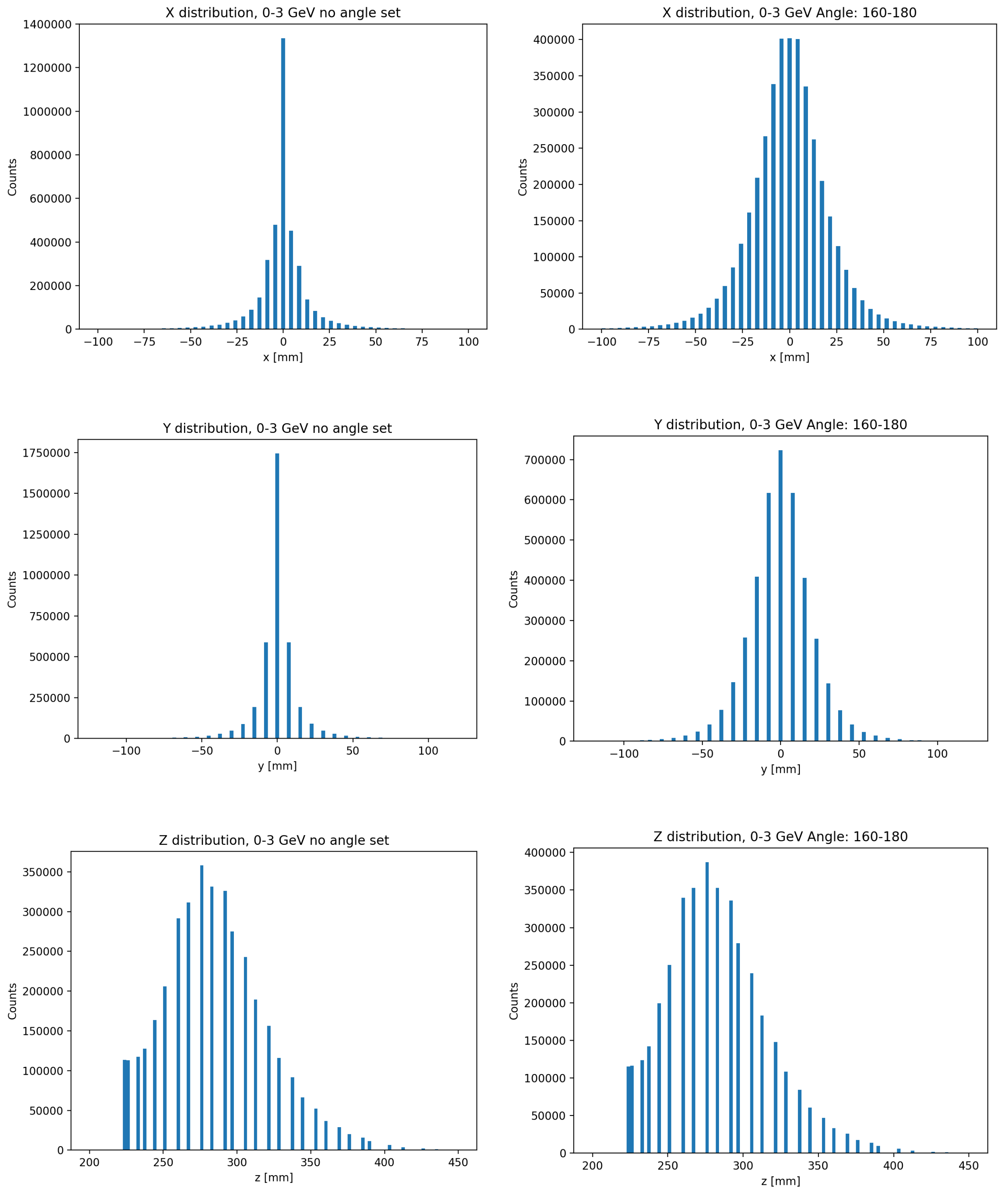


Figure 13: Left: The x, y and z distribution from the simulation of the first no angle data set. Right: The x, y and z distribution from the simulation of the second 160°-180°data set.

An example array of what is fed into the network is shown in figure 14. This event is from the reconstruction hit 160°-180° data set. Each dot corresponds to where a particle has deposited energy in the calorimeter material.

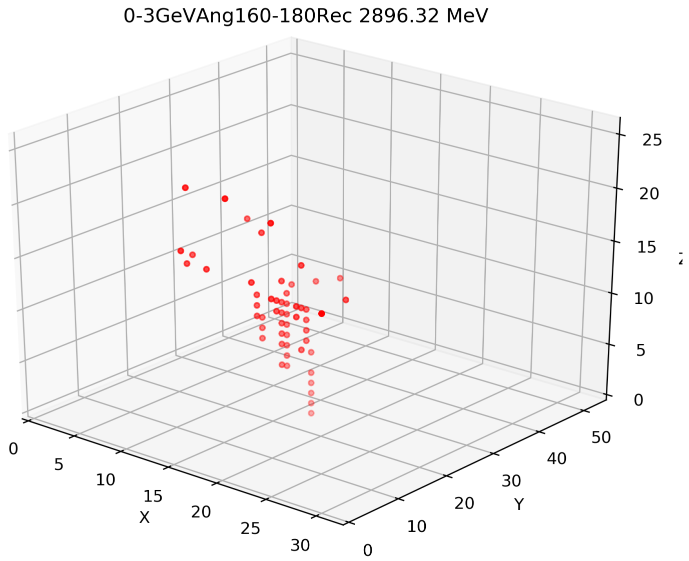


Figure 14: The figure shows the 3D array in x,y and z of an events with an energy of 2896.32 MeV. The intensity of the color of the dot indicates if the hit is close of far from the view point. A dot with less intensity is further into the array than a dot with more intensity. No indication in the figure is given to the size of the energy deposition.

6.3 Training

Three different models were used:

- One linear regression model which is comparable to conventional energy reconstruction.
- One dense(i.e. fully connected MLP) neural network (DNN). The 3D data is flattened into a 1D array with $24 \times 48 \times 28 = 32256$ and $32 \times 54 \times 26 = 44928$ inputs. The flattened array is then fed as the input to the first layer of the network. The first layer is followed by a number of hidden layers all using ReLU activation functions with dropout applied. The number of layers, dropout probability and number of nodes is treated as hyperparameters. The last layer feeds into a single output node with a linear activation function. The DNN was trained for 10 epochs.
- The CNN consists of 3D convolutions layers with ReLU activation and maxpooling. The filter sizes, stride and amount of filters for convolutions and pooling was treated as hyperparameters. After the convolutions the output array is flattened and feed to an MLP. The number of layers, dropout probability and number of nodes is also treated as hyperparameters. The output node has a linear activation similar to the DNN. The CNN was trained for 20 epochs.

The error function for both network models was MSE. The Adam optimizer was used to train the network. Both the learning rate and batch size was treated as hyperparameters.

The networks were trained using a manual hyperparameter search which was performed by training different models and then investigating new parameters based on the models performance. The data set of 100000 events was divided into 87500 event for training and 12500 for validation.

After the top performing models based on validation loss for each data set were determined, energy resolution plots for all sets were computed. To do this test data sets with 10000 events each were generated with energies between 500-2500 MeV in 250 MeV steps. Distributions of $\Delta E = E_{true} - E_{predicted}$ were computed for all models and then a Gaussian fit was made to these distributions to extract a 1-sigma estimate. Some of these Gaussian fits made can be seen in figure 20 in the Appendix. The sigma estimate was then plotted and a fit was made for each model to only the stochastic term in equation 1. This decision was done because of the simulation data having almost no noise (small amount of noise is added to the recon hits) and the detail of the simulations not being good enough leading to the the constant term being unnecessary. Both fits (Gaussian and energy resolution) was done using the curve_fit method from Pythons scipy library.

7 Results

For all CNN models two convolutional layers was deemed slightly more optimal, two layers improved the validation MSE by in general $\approx 1000 - 3000$ while 3 layers was deemed too computationally heavy to search for. Earlier research [5] has also indicated that more layers are usually not worth the computational cost. The hyperparameters for the convolutions were optimized to: The first convolution has 20 filters with a size of (8x8x8) and stride of (1x1x1) followed by (2x2x2) max pooling with a stride of (2x2x2). The second convolution has 10 filters with a size of (4x4x4) and stride of (1x1x1) followed by (2x2x2) max pooling with a stride of (2x2x2) again. This was followed by a MLP with 1500 nodes in each layer with 5 layers. The rest of the optimized hyperparameters differ between data sets and are shown in table 1.

Table 1: Table shows the most optimal hyperparameters for the CNNs as well as the validation loss that the model achieved during training.

Data type	No angle set				160°-180°set			
Hit type	Sim hits		Recon hits		Sim hits		Recon hits	
Model sign	S1	S2	R1	R2	AS1	AS2	AR1	AR2
Dropout probability	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.12
learning rate	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
batch size	128	256	64	128	128	256	256	256
Validation loss(MSE)	34129	33130	34853	34179	34861	36991	36424	36150

For the DNNs the optimal model was very similar between the different data sets as shown in table 2.

Table 2: Table shows the most optimal hyperparameters for the DNNs as well as the validation loss that the model achieved during training.

Data type	No angle set		160°-180°set	
Hit type	Sim hits	Recon hits	Sim hits	Recon hits
# of layers	5	5	5	5
# of nodes	512	512	512	400
Dropout probability	0.04	0.04	0.04	0.04
learning rate	0.00005	0.00005	0.00005	0.00001
batch size	64	32	128	32
Validation loss(MSE)	45620	53078	56306	60262

The linear fits for the four different data sets are shown in figure 15. The fits are drawn on top of a heat map of the distribution between the sum of hit energy in the array and the true energy of that event.

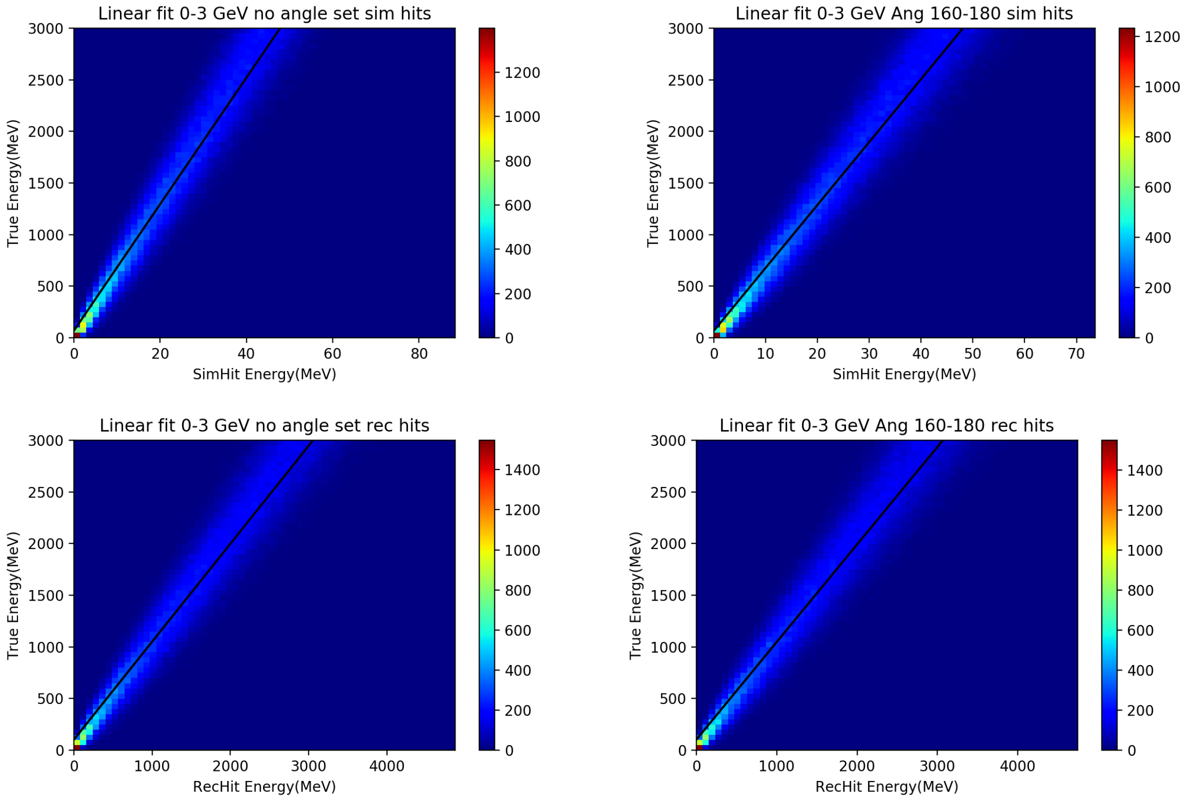
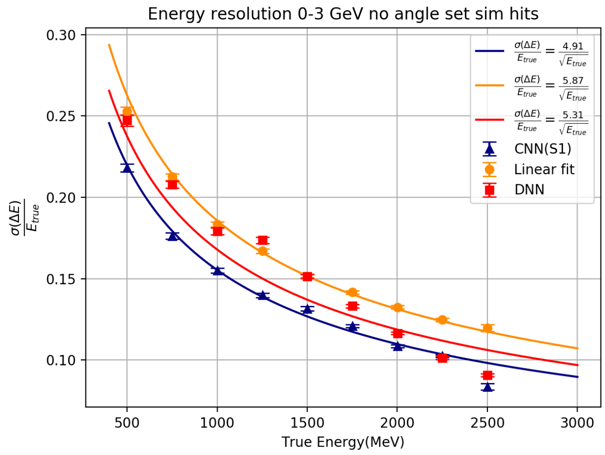
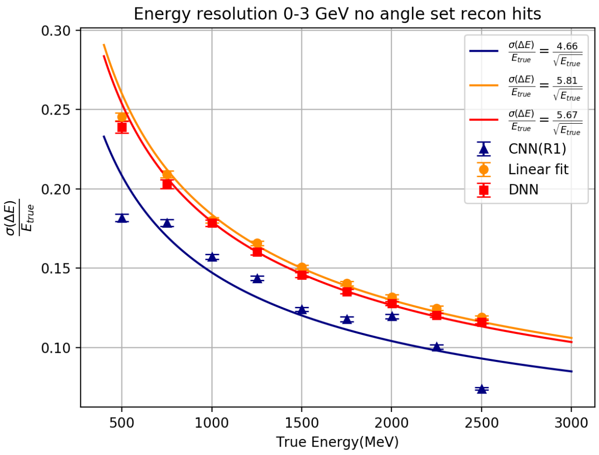


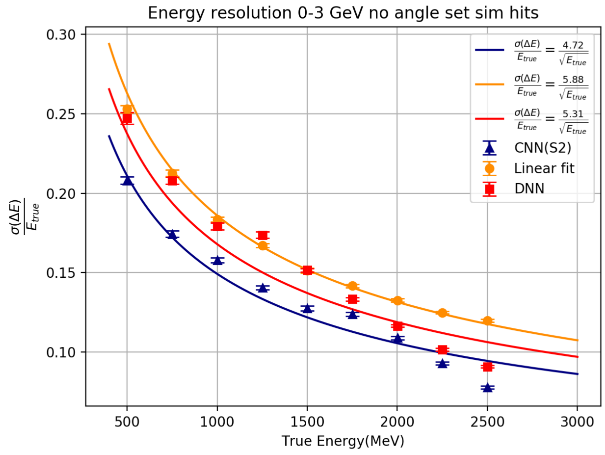
Figure 15: Linear fits from the four different data sets. The heat map shows the 2D distribution of the event energy vs true energy. Event energy is computed from the sum of all the hits within the array cutouts.

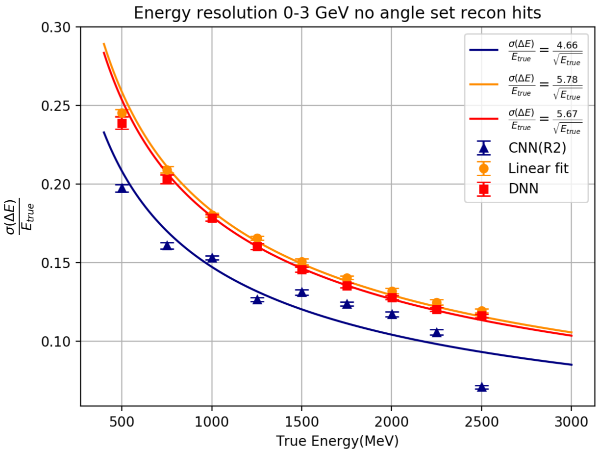
The energy resolution of the CNNs, DNNs and linear fits is shown in figure 17. The 8 figures correspond to the 8 different models in table 1. Each data point corresponds to a 1-sigma estimate from a Gaussian fit of the energy difference ($\Delta E = E_{true} - E_{predicted}$)

distribution for that energy. All the Gaussian fits for the data points in the AS2 model figure are shown in figure 20 in the Appendix to illustrate how the fits look. The estimate for the energy resolution fit parameter is shown in the legend. The CNNs performs in general better than both the DNN and the linear fit. For the sim hits the DNN has a very poor fit to the energy resolution function, which can be seen from the uncertainty in the fit parameter in table 3. The CNN also has some inconsistencies in this regard but its behavior is more randomly distributed over different data sets.









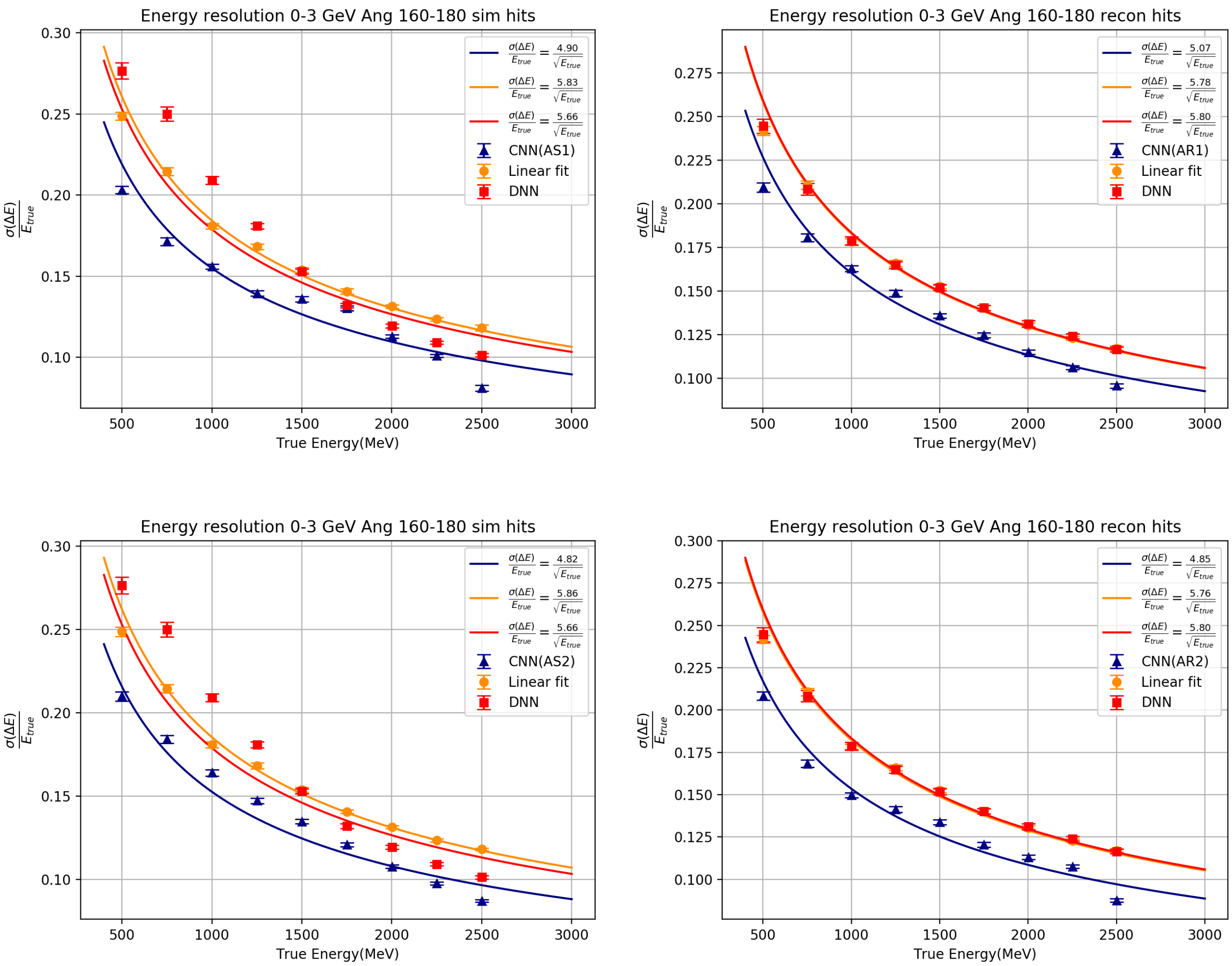


Figure 17: Energy resolution plots with two CNN models for both data sets and both data types. Each data point is a 1-sigma estimate from a Gaussian fit to a energy difference ($\Delta E = E_{true} - E_{predicted}$) distribution.

From the energy resolution fits in figure 17 an estimate and a uncertainty for the stochastic term constants is shown in table 3.

Table 3: Table shows the energy resolution fit value with uncertainty for all data sets. All values have unit $\sqrt{\text{MeV}}$.

Data type	No angle set		160°-180°set	
Hit type	Sim hits	Recon hits	Sim hits	Recon hits
CNN 1	4.91 ± 0.05	4.66 ± 0.19	4.90 ± 0.11	5.07 ± 0.07
CNN 2	4.72 ± 0.13	4.66 ± 0.16	4.82 ± 0.11	4.85 ± 0.09
DNN	5.31 ± 0.18	5.67 ± 0.04	5.66 ± 0.20	5.80 ± 0.04
Linear	5.88 ± 0.02	5.80 ± 0.04	5.85 ±0.03	5.77 ± 0.04

8 Discussion

An important aspect of the data sets is that they mainly use only the center hexagon of the ECal. This might give a good estimation of the energy resolution of the whole detector but the data does not fully represent the raw data that the experiment will produce. The data handling of the conversion to arrays could probably have been done better. Because of the ECal design described in the method leading to gaps in the y-distribution, $\approx \frac{2}{3}$ of the memory is left unused in the arrays generated. Seeing as the arrays take quite a lot of memory (24GB and 34GB for the two data sets), optimizing this could have given space for more events to be trained on.

One important check to establish that the results are reasonable is to compare to the energy resolution of the ECal upgrade of CMS [4]. Because the design is almost identical to LDMX, the results should be comparable. Comparing figure 18 to figure 17 and taking units into account we see that $0.199\sqrt{\text{GeV}} \approx 6.29\sqrt{\text{MeV}}$. Comparing this value to the linear fit values from the results in table 3 being in the range $5.67 - 5.88\ \sqrt{\text{MeV}}$ we see that the results seems reasonable. The energy resolution being slightly better in the results are reasonable because of LDMX using thicker silicon in the simulation setup.

It's interesting to note that the linear fit performs worse on the sim hit data sets than the recon hit ones especially for the no angle data set. This is probably because of this fit not being able to take the layer weights into account that are applied for the recon-

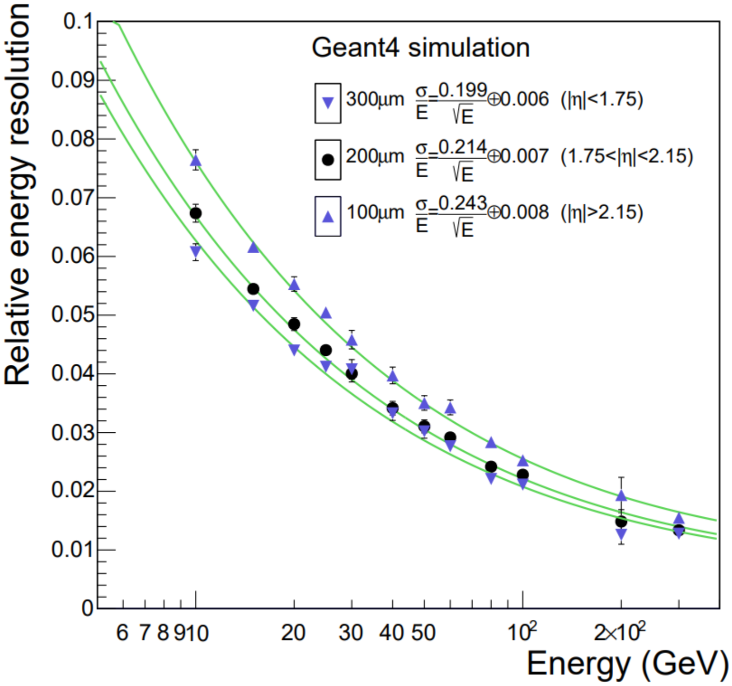


Figure 18: Energy resolution plot from the simulations of the high granularity upgrade of the CMS forward ECal[4].

structed hits. This means that comparing the linear fit for the sim hit data with the model performance does not really represent how they can perform on real data, because of the networks abilities to learn layer weights. Comparing model performance on the sim hit data sets with the linear fit on reconstructed data instead could be an option. One problem with this is that noise is added to the recon hits which gives the reconstructed linear fit worse performance. To combat this another data set could have been created applying the layer weights to the sim hit data sets but unfortunately no time remained for this task. With the current results the reconstructed data is a better indication for how the models would perform and compare to the linear fit on real data.

Furthermore, the DNN performs very inconsistent on the sim hit data sets. This seems like a weird behavior because as said before the only change between recon and sim hits is layer weights and noise. The DNN should easily be able to learn the layer weights by itself as the flattened array is still ordered. But it seems this is not the case and it might just be the fact that the layer weights from the recon data gives the network a jump start in the correct direction.

Something that could be improved with the method of this thesis is the hyperparameter search. Doing the search manually was all I had time with this thesis but using a more sophisticated search algorithm would be preferred to find the best performing model. Something like grid search would give better grounds to the models being optimal.

Another aspect that could have improved the results is using ensemble machines. An ensemble machine is a combination of several machine learning models that have trained well on the data set. The combined validation performance of the ensemble usually outperforms the best model and the prediction should also be more consistent. Thereby reducing the inconsistencies in energy resolution shown by the CNN and the DNN in figure 17.

A time evaluation of how long the network takes to predict energies could have been made. Because of the enormous amounts of data coming from particle physics experiments it's important that algorithms are fast to enable for event triggering. Because the network will act on the reconstruction of event for analysis it might not need to be used during the fast real-time triggering.

9 Conclusion

This thesis has investigated whether neural network models can improve the energy resolution of a calorimeter. This task was applied to data generated in LDMX simulations. The CNNs performed better than both the DNN and the linear fit for all data sets. Some inconsistencies in how the model follow the expected energy resolution fit was apparent. This might be because of what the model looks for, the model not trained very well or just because of the nature of how neural networks function.

10 Outlook

The thesis shows that neural networks is a method that particle physics could use in the future for improving on the energy resolution of calorimeters. Some questions remain considering the consistency of the network. Some things to consider for further implementation is that the network might perform worse on real data that is not simulated. Training on more data, using a more incorporating hyperparameter search or perhaps implementing an ensemble machine could give better and more consistent results.

11 Acknowledgements

Firstly, I want to thank my supervisor Ruth Pöttgen for all the support, feedback and guidance. I would also like to thank the Lund LDMX collaboration for great input and especially thank Leo Östman for getting me started quickly and helping me with various questions. I want to thank my girlfriend for continuous support during this thesis. The computations/simulations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at the center for scientific and technical computing at Lund University (LUNARC) partially funded by the Swedish Research Council through grant agreement no. 2016-07213.

References

[1] F. Zwicky, translation performed by H.Andernach, *English and Spanish Translation of Zwicky's (1933) The Redshift of Extragalactic Nebulae*, 2017, arXiv:1711.01693 [astro-ph.IM].

[2] V. Rubin and W. Ford, Astrophys. J. 159, 379 (1970).

[3] T. Åkesson *et al.* [LDMX working group], *Light Dark Matter eXperiment (LDMX)*, 2018, arXiv:1808.05219 [hep-ex].

[4] J. Butler, D. Contardo, M. Klute, J. Mans, L. Silvestris, *Technical Proposal for the Phase-II Upgrade of the CMS Detector*, 2015, CMS Technical proposal CERN-LHCC-2015-010, LHCC-P-008, CMS-TDR-15-02, `https://cds.cern.ch/record/202088`.

[5] D. Belayneh *et al.*, *Calorimetry with Deep Learning: Particle Simulation and Reconstruction for Collider Physics*, 2019, arXiv:1912.06794 [physics.ins-det].

[6] F. Carminati, G. Khattak, M. Pierini, S. Vallecorsafac, A. Farbin, *Calorimetry with deep learning: particle classification, energy regression, and simulation for high-energy physics*, In-NIPS, 2017, `https://dl4physicalsciences.github.io/files/nips_dlps_2017_15.pdf`.

[7] J. Damgov, L. Litov, *Application of neural networks for energy reconstruction*, Nucl. Inst. & Meth. in Phys. Res. A 482 (2003), 776-788, ISSN: 0168-9002, `http://www.sciencedirect.com/science/article/pii/S0168900201018514`

[8] B.R. Martin, G. Shaw, *Particle Physics*, 3rd edition, 2008, John Wiley & Sons, ISBN: 978-1-118-68166-4.

[9] Wikipedia, *Particle physics*, `https://en.wikipedia.org/wiki/Particle_physics`, Accessed: 2020-04-10.

[10] Leif Jönsson, *Lectures in Particle Physics*, 2005, revised: 2018, Lund University.

[11] C.W Fabjan and F. Gianotti, *Calorimetry for particle physics*, 2003, American Physical Society, Rev. Mod. Phys. 75, 1243-1286, DOI: `https://doi.org/10.1103/RevModPhys.75.1243`

[12] M. Tanabashi et al. [Particle Data Group], *Passage of Particles Through Matter* 2018, Phys. Rev. D 98, 030001, `http://pdg.lbl.gov/2019/reviews/rpp2019-rev-passage-particles-matter.pdf`.

[13] A.G, Doroshkevich, V.N. Lukash, E.V. Mikheeva, *A solution to the problems of cusps and rotation curves in dark matter halos in the cosmological standard model*, 2012, Phys. Usp. 55 3-17, arXiv:1209.0388 [astro-ph.CO].

[14] K. Freese, *Status of dark matter in the Universe*, 2017, arXiv:1701.01840 [astro-ph].

[15] M. Battaglieri *et al.* , *US Cosmic Visions: New Ideas in Dark Matter 2017: Community Report*, 2017, arXiv:1707.04591 [hep-ph].

[16] J. Beacham *et al.* ,*Physics Beyond Colliders at CERN Beyond the Standard Model Working Group Report*, 2019, arXiv:1901.09966 [hep-ex].

[17] R. Foot and S. Vagnozzi, *Dissipative hidden sector dark matter*, Phys. Rev. D 91 (2015), arXiv:1409.7174 [hep-ph].

[18] M. Ohlsson and P. Edén, *Lecture Notes on Introduction to Artificial Neural Networks and Deep Learning*, 2019, Department of Astronomy and Theoretical Physics, Lund University.

[19] `https://pythonmachinelearning.pro/perceptrons-the-first-neural-networks/`, Accessed: 2020-04-02.

[20] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, *Improving neural networks by preventing co-adaptation of feature detectors*, 2012, arXiv:1207.0580 [cs.NE].

[21] F. Chollet, *Keras*, 2015, `https://keras.io/`.

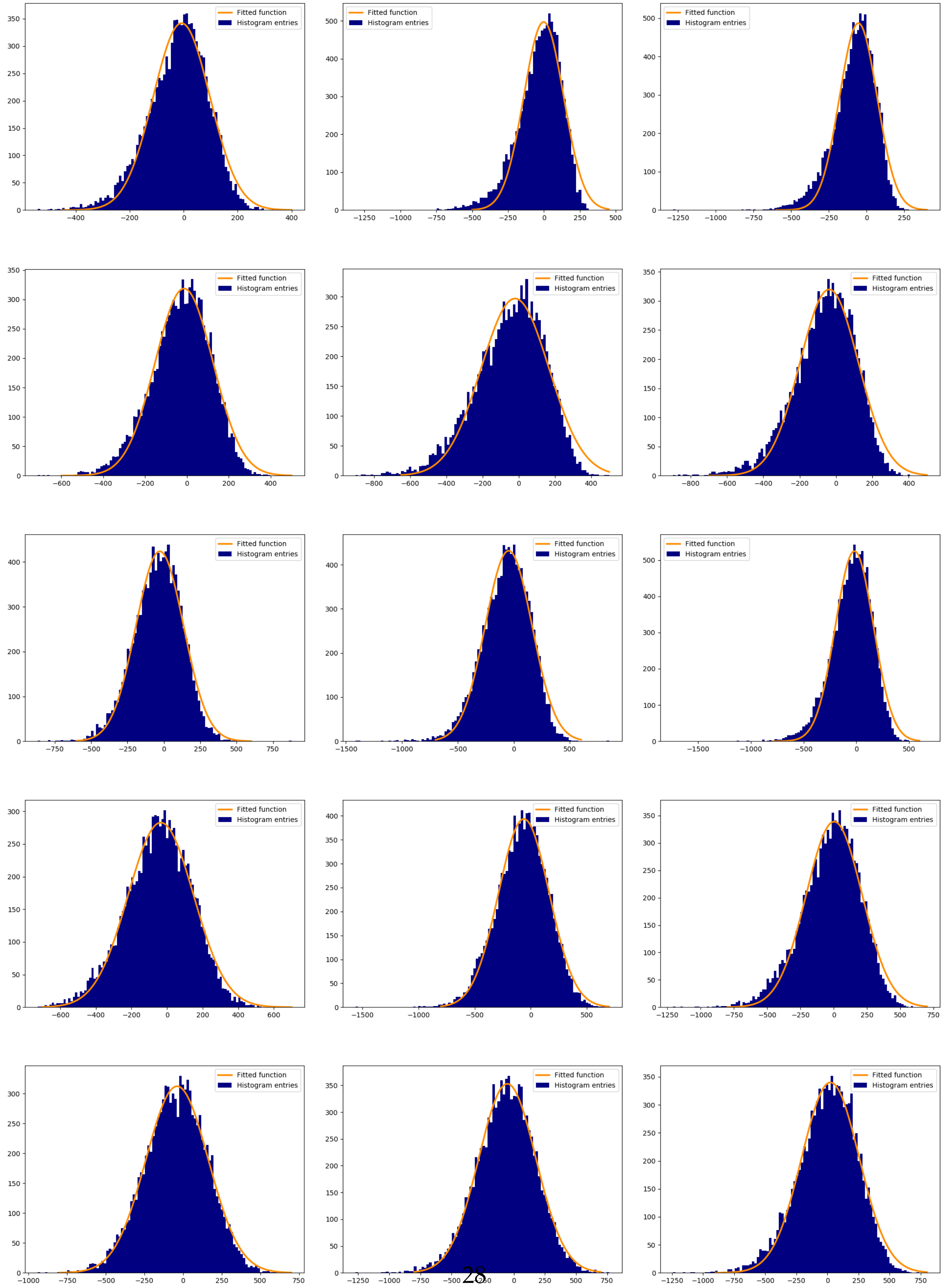
[22] M. Abadi *et al.*, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, 2015, `https://www.tensorflow.org/`.

[23] Rene Brun and Fons Rademakers, *ROOT - An Object Oriented Data Analysis Framework*, Proceedings AIHENP'96 Workshop, Lausanne, Sep. 1996, Nucl. Inst. & Meth. in Phys. Res. A 389 (1997) ,81-86, `https://root.cern.ch/`

[24] S. Agostinelli *et al.*, *Geant4-a simulation toolkit*, Nucl. Inst. & Meth. in Phys. Res. A 506 (2003), 250-303, doi: `10.1016/S0168-9002(03)01368-8`, `https://geant4.web.cern.ch/`.

[25] Travis E. Oliphant, *A guide to NumPy (Vol. 1)*, 2006, Trelgol Publishing USA, `https://numpy.org/`.

Appendix


Fitted function
Histogram entries
0 50 100 150 200 250 300 350
−400 −200 0 200 400
0 100 200 300 400 500
−1250 −1000 −750 −500 −250 0 250 500
0 100 200 300 400 500
−1250 −1000 −750 −500 −250 0 250
0 50 100 150 200 250 300 350
−600 −400 −200 0 200 400
0 50 100 150 200 250 300
−800 −600 −400 −200 0 200 400
0 50 100 150 200 250 300 350
−800 −600 −400 −200 0 200 400
0 100 200 300 400
−750 −500 −250 0 250 500 750
0 100 200 300 400
−1500 −1000 −500 0 500
0 100 200 300 400 500
−1500 −1000 −500 0 500
0 50 100 150 200 250 300
−600 −400 −200 0 200 400 600
0 50 100 150 200 250 300 350 400
−1500 −1000 −500 0 500
0 50 100 150 200 250 300 350
−1250 −1000 −750 −500 −250 0 250 500 750
0 50 100 150 200 250 300
−1000 −750 −500 −250 0 250 500 750
0 50 100 150 200 250 300 350
−1250 −1000 −750 −500 −250 0 250 500 750
0 50 100 150 200 250 300 350
−1250 −1000 −750 −500 −250 0 250 500 750

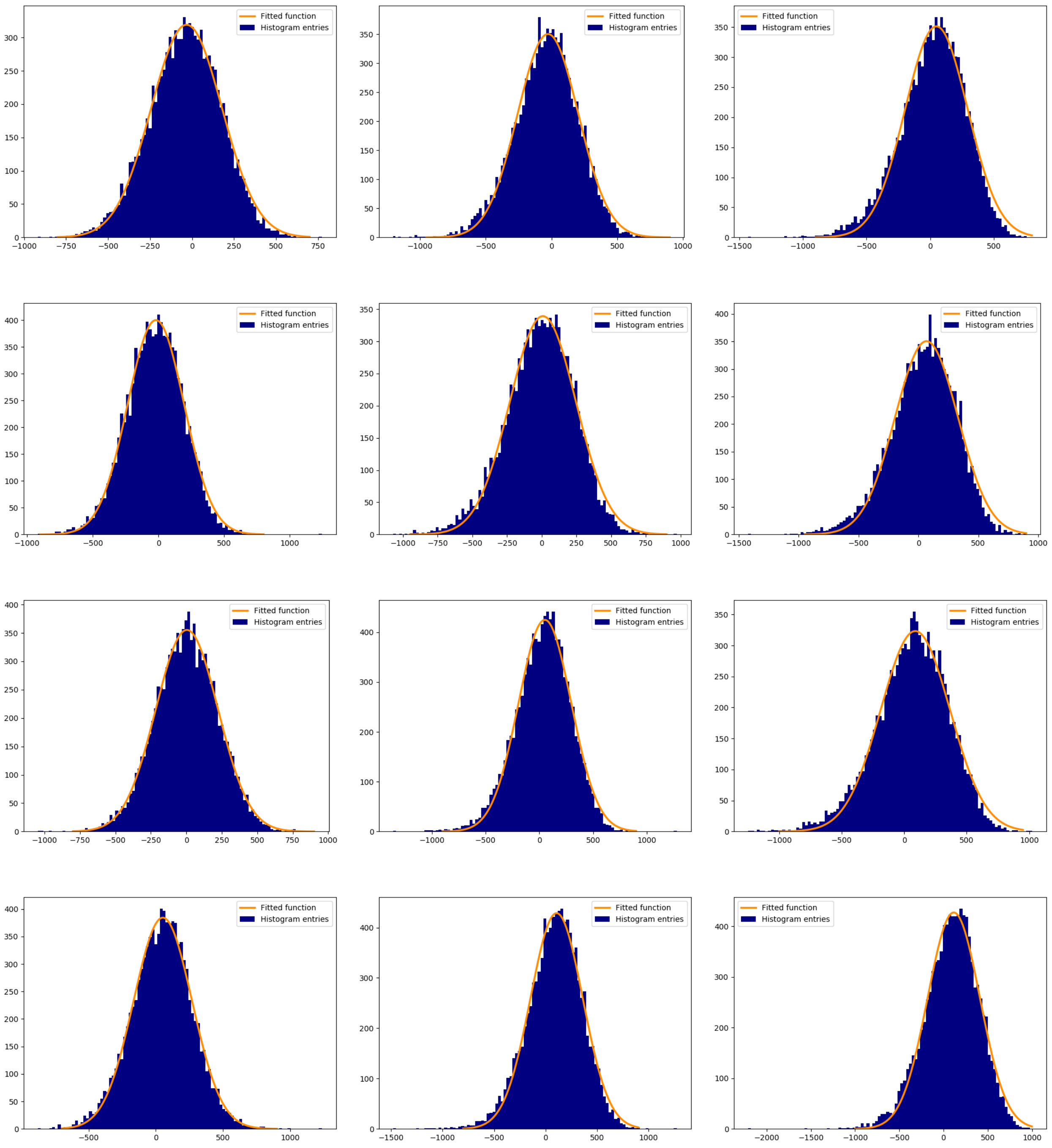


Figure 20: An example of how the Gaussian fit figure looks like. All plot the distribution of the difference ΔE between the energy predicted by the model and the correct energy. In the left row is the model AS2, in the middle row is the sim hit Ang 160°-180°DNN and in the right row is the linear fit. These figures correspond to all 1-sigma estimates for the AS2 plot in figure 17.