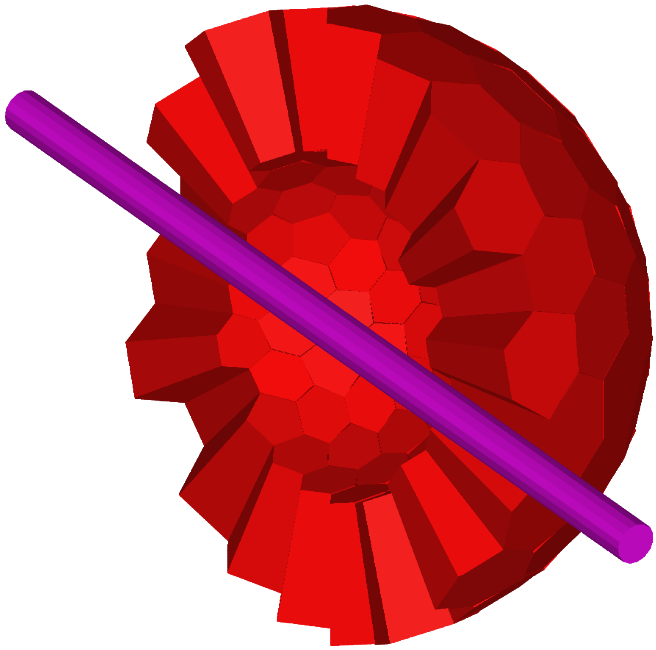
CHALMERS



Rekonstruktion från detektordata med hjälp av neurala nätverk

En studie i tvärsnittet mellan kärnfysik och maskininlärning

Kandidatarbete i teknisk fysik: TIFX04-18-08

Jacob Olander
Miriam Skarin
Pontus Svensson
Jakob Wadman

Handledare: Håkan T Johansson
Andreas Heinz

Avdelningen för subatomär fysik och plasmafysik, Institutionen för fysik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2018

Framsida: Datorgenererad visualisering av partikeldetektorn Crystal Ball.

Rekonstruktion från detektordata med hjälp av neurala nätverk
En studie i tvärsnittet mellan kärnfysik och maskininlärning
Jacob Olander
Miriam Skarin
Pontus Svensson
Jakob Wadman
Institutionen för fysik
Chalmers tekniska högskola

Sammandrag

Denna rapport demonstrerar hur artificiella neurala nätverk utifrån data från partikeldetektorer kan rekonstruera relevanta egenskaper hos partiklar skapade vid kärnfysikexperiment. Enkla fall undersöks där energin hos en γ-partikel ska bestämmas, men även mer komplexa fall där både energin och emissionsriktningen hos flera partiklar ska rekonstrueras. Arbetsprocessen har till stor del varit iterativ, där mycket fokus lagts på att optimera kostnadsfunktionen. Rekonstruktionen kan utföras för både en och flera partiklar och är inte begränsad till att behandla en partikelegenskap åt gången. Slutsatsen av undersökningen är att artificiella neurala nätverk kan användas för att rekonstruera partikelegenskaper, men att de metoder som utvecklats i detta arbete, ännu inte är bättre än befintliga rekonstruktionsmetoder grundade i fysik. Fortsatta studier behövs för att definitivt kunna avgöra om artificiella neurala nätverk kan användas för att göra bättre rekonstruktioner än vad som i dagsläget är möjligt.

Abstract

This report demonstrates how artificial neural networks, using data from particle detectors, can be applied to reconstruct relevant features of γ-particles from nuclear reaction experiments. Simple events are investigated, where the energy of one γ-particle is predicted, as well as more complex events, where both the energy and the direction of emission are reconstructed for multiple particles at once. A trial-and-error work process has been used, where the focus was to optimize the cost function. The reconstruction can be made for multiple particles and is not limited to treating one particle feature at a time. The conclusion of the investigation is that artificial neural networks can be used to reconstruct particle features, but that the methods developed in this project, not yet outperform existing, physics based, methods. Further studies are needed to determine whether artificial neural networks can be used to make better reconstructions than what is currently possible.

Nyckelord: neurala nätverk, partikeldetektorer, datarekonstruktion, FAIR, R^3B, CALIFA, Crystal Ball, addback.

Innehåll

1 Introduktion **1**
1.1 Syfte . . . 2
1.2 Avgränsningar . . . 2

2 Teoretisk bakgrund **3**
2.1 Fysikalisk bakgrund och partikeldetektorer . . . 3
2.1.1 Reaktionstvärsnitt . . . 3
2.1.2 Material med godtycklig tjocklek och medelfri väg . . . 4
2.1.3 Energiförlust i materia för laddade tunga partiklar . . . 4
2.1.4 Energiförlust i materia för elektroner och positroner . . . 5
2.1.5 Fotoners växelverkan med materia . . . 5
2.1.6 Simuleringsverktyget GEANT4 . . . 7
2.1.7 Partikeldetektorer . . . 7
2.1.8 Relativistiska effekter . . . 9
2.2 Artificiella neurala nätverk . . . 9
2.2.1 Principer för artificiella neurala nätverk . . . 9
2.2.2 Att träna ett neuralt nätverk . . . 11
2.2.3 Kostnadsfunktion . . . 11
2.2.4 Bakåtfortplantning och optimering av nätverkets parametrar . . . 12
2.2.5 Optimeringsalgoritmen Adam . . . 12
2.2.6 Uppdelning av data och överträning . . . 12
2.2.7 Faltande nätverk . . . 13

3 Konstruktion av modell **15**
3.1 Simulering av data . . . 15
3.1.1 Simuleringsparametrar . . . 16
3.2 Beräkningsresurser . . . 16
3.3 Programvara för uppbyggnad av neurala nätverk . . . 16
3.4 Energi- och vinkelrekonstruktion för Crystal Ball . . . 17
3.4.1 Utveckling av kostnadsfunktion för en γ-partikel . . . 17
3.4.2 Användning av kostnadsfunktion för flera γ-partiklar . . . 19
3.4.2.1 Jämförelse mellan absolut och relativ kostnadsfunktion . . . 20
3.4.3 Beräkningsmetoder för kostnadsfunktion för flera γ-partiklar . . . 21
3.4.4 Viktning av kostnadsfunktionens termer . . . 23
3.4.5 Rekonstruktionens energiberoende . . . 24
3.4.6 Konvergens av nätverk och variation av nätverksarkitektur . . . 25
3.4.7 Kostnadsfunktion för felminimering av Dopplerkorrigerad energi . . . 25
3.5 Jämförelse av olika faltningsmetoder . . . 26
3.5.1 Feluppskattning . . . 28

3.5.2 Faltning för energi- och vinkelrekonstruktion 29

4 Jämförelse med nuvarande metoder och applicering på annan detektor 31
4.1 Jämförelse med addback-rutin 31
4.2 Rekonstruktion för detektorn DALI2 33

5 Diskussion och sammanfattning 35
5.1 Övergripande diskussion 35
5.2 Slutsatser 36
5.3 Fortsatt arbete 36

Referenser 37

A Detaljer kring faltningsmetoder I
A.1 Metoder I
A.1.1 Klassificeringsnätverk för multiplicitetsbedömning I
A.1.2 Ett faltningslager med närmsta grannar (NG) II
A.1.3 Faltning med närmsta grannar två gånger i följd (NG-NG) II
A.1.4 Två faltningslager med medelvärdes-pooling (NG-pool-NG-pool) . . III
A.1.5 Två faltningslager utan pooling (NG-NNG) III
A.2 Träningstider IV

B Nätverkets struktur V
B.1 Konvergens av nätverk V
B.2 Variation av nätverksarkitektur V
B.3 Fördelning av felen i rekonstruktionen VII

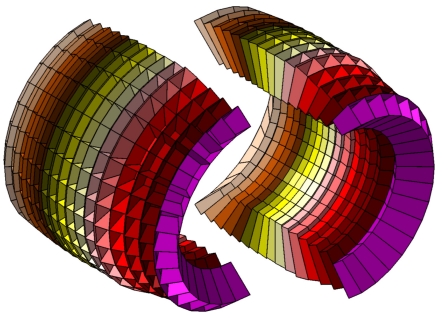
C Anpassningar av rekonstruerade spektrum för jämförelse av realistisk data IX

1

Introduktion

Ett vanligt sätt att undersöka atomkärnors struktur är genom att accelerera joner mot ett strålmål för att sedan studera spridningen och sönderfallet av de exotiska kärnor som ibland bildas som följd av reaktioner i strålmålet. För att kunna studera dessa fenomen behövs omfattande detektoruppställningar kring strålmålet. Eftersom strålningens interaktion med detektorn ofta präglas av hög komplexitet måste resultatet från detektorn tolkas ingående för att ge en korrekt bild av vad som hände i detektorn.

En inom några år kommande partikelacceleratoranläggning är FAIR (Facility for Antiproton and Ion Research), belägen utanför Darmstadt, Tyskland [2]. Ett av projekten knutit till anläggningen är R^3B-experimentet (Reactions with Relativistic Radioactive Beams) [3] där en nyckeldel av mätuppställningen är detektorn CALIFA (CALorimeter for In-Flight detection of gamma-rays and high energy charged pArticles) som kommer att omsluta R^3Bs strålmål. Se figur 1.1 för en modell av detektorn. CALIFA har 1952 scintillerande CsI(Tl)-kristaller och kommer fungera som en γ-kalorimeter och -spektrometer, men kommer även kunna detektera laddade partiklar som produceras vid kollisionerna i strålmålet [4]. Denna mångsidighet ställer höga krav på databehandling och tolkning av resultat så att en så korrekt rekonstruktion som möjligt av resulterande partiklars egenskaper kan göras.



Figur 1.1: *Artistisk visualisering av en del av detektorn CALIFA som är planerad att sättas i bruk på FAIR inom några år. Bild hämtad från [1].*

I dagsläget finns ett flertal metoder baserade på fysikaliska modeller för att utföra rekonstruktionen för detektorer som liknar CALIFA [5], men dessa är ofta begränsade. Det är exempelvis svårt att skilja händelser där endast en liten andel av partikelenergin har deponerats i detektorn från händelser med en partikel vars energi var lägre från början. Genom att studera och upptäcka mönster i detektorn kan kanske rekonstruktionen göras bättre.

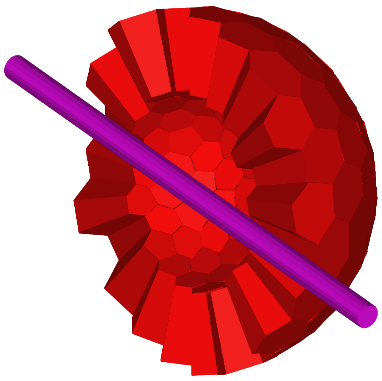
En metod för att studera mönster i andra sammanhang är neurala nätverk. Neurala nätverk är inspirerade av den mänskliga hjärnan och har länge använts inom forskning och industri, men framförallt har det under senare år visats att dessa algoritmer, med stor framgång, kan identifiera mönster vid exempelvis bildigenkänningsproblem [6]. Mot denna bakgrund presenterar denna rapport huruvida rekonstruktion för detektorer liknande CALIFA kan göras med hjälp av neurala nätverk.

1.1 Syfte

Syftet med denna rapport är att demonstrera hur neurala nätverk kan användas för att rekonstruera relevanta egenskaper hos resulterande partiklar från en kärnreaktion i en partikeldetektor.

1.2 Avgränsningar

Eftersom CALIFA är oregelbunden i sin geometri och består av så pass många kristaller, fokuserar denna rapport på två andra befintliga detektoruppställningar som bygger på samma principer som CALIFA. Den detektor som har studerats är framförallt *Crystal Ball* [7], men även *DALI2* [8] har behandlats. Båda detektoruppställningarna är uppbyggda av scintillerande kristaller. Crystal Ball är uppbyggd av 162 NaI-kristaller sfäriskt symmetriskt placerade kring strålmålet. Figur 1.2 visar halva detektorn och strålrörets position. Genom att studera två olika detektorer som liknar CALIFA är förhoppningen att metoderna som tas fram kan generaliseras till den nya detektorn. Den data som använts är simulerad data från modeller av de givna detektoruppställningarna.



Figur 1.2: *Halva detektorn Crystall Ball inklusive position för strålröret. På grund av dess sfäriska symmetri lämpar den sig för det här arbetet.*

Arbetet har begränsats till rekonstruktion av γ-partiklar med energier som typiskt ses vid kärnfysikexperiment, 50 keV-10 MeV. Undre gränsen härrör framförallt från hur små energier som detektorn kan detektera.

Ett neuralt nätverk kan konstrueras på många olika sätt. Det krävs en metod för att bestraffa nätverket när det har fel vilket realiseras av en så kallad kostnadsfunktion. Rapporten begränsar sig till stor del att diskutera för- och nackdelar med olika typer av kostnadsfunktioner och inte primärt nätverkens arkitektur.

2
Teoretisk bakgrund

Eftersom projektet grundar sig i både kärnfysik och maskininlärning presenterar detta kapitel den mest väsentliga teorin inom båda områdena. En fysikalisk bakgrund till projektet och en introduktion till artificiella neurala nätverk ges.

2.1 Fysikalisk bakgrund och partikeldetektorer

Detta avsnitt behandlar de mest relevanta fysikaliska processer som sker i detektorn och hur den är uppbyggd. En mer detaljerad beskrivning av de fysikaliska processerna ges av exempelvis Leo [9], som de kommande avsnitten till stor del bygger på[1].

Strålning med olika massa interagerar med materia på olika sätt och beskrivs därför bäst av olika modeller. Fotoner, även kallade γ-partiklar, behandlas separat eftersom deras interaktion skiljer sig från massiva partiklars. Hur strålning interagerar med materia ligger till grund för alla sorters partikeldetektorer eftersom det är energin som partikeln deponerar i detektormaterialet som detektorn (teoretiskt) kan registrera. Simuleringsverktyget GEANT4 (se avsnitt 2.1.6) som använts i det här arbetet bygger på teorin beskriven i det här avsnittet.

2.1.1 Reaktionstvärsnitt

För att beskriva reaktioner som uppstår när en partikelstråle riktas mot ett material är det lämpligt att använda sig av konceptet reaktionstvärsnitt, en storhet betecknad σ. Tvärsnittet är en fysikalisk observabel och står i proportion till sannolikheten att en reaktion ska inträffa. Om strålen antas ha ett konstant flöde av partiklar per area- och tidsenhet, F, och N_s är det genomsnittliga antalet spridningar per tidsenhet in i en rymdvinkel Ω, så definieras det differentiella tvärsnittet som

$$\frac{d\sigma}{d\Omega}(E, \Omega) = \frac{1}{F}\frac{dN_s}{d\Omega}, \tag{2.1}$$

där E är partikelns energi. Uttrycket i (2.1) beskriver alltså den genomsnittliga andelen av partiklar som sprids in i rymdvinkelelementet $d\Omega = d(\cos\theta)d\phi$ per tids- och flödesenhet. Det totala tvärsnittet för en given energi kan beräknas genom att integrera (2.1) över alla rymdvinklar,

$$\sigma(E) = \int \frac{d\sigma}{d\Omega}(E, \Omega)d\Omega. \tag{2.2}$$

[1] *Techniques for Nuclear and Particle Physics Experiements* av W. R. Leo är läsvärd och rekommenderas för en mer fullständig diskussion.

En naiv tolkning av σ är att se det som det geometriska tvärsnittet mellan partiklarna i en reaktion eftersom dess dimension är area. Detta hade varit fallet om partiklarna beskrivits som solida sfärer som endast interagerar genom mekanisk kontakt. Interaktionen sker exempelvis genom elektromagnetisk växelverkan och därför behövs kvantmekanik för att formulera en mer verklighetstrogen beskrivning.

Strålmålet utgörs av ett objekt med en viss tjocklek Δx som innehåller många möjliga spridningspunkter, exempelvis atomkärnor. Om det antas att punkternas positioner är likformigt fördelade samt att objektet är tunt nog för att sannolikheten för överlappande punkter ska vara låg, ges antalet möjliga spridningspositioner per areaenhet vinkelrätt mot strålens riktning av $N\Delta x$ där N är volymdensiteten av sådana punkter. Det genomsnittliga antalet spridningar per tidsenhet in i $d\Omega$ ges då av $N_s(\Omega) = FAN\Delta x\frac{d\sigma}{d\Omega}$, där A är den geometriska tvärsnittsarean mellan strålen och strålmålet. Den totala sannolikheten för en partikel att spridas i intervallet Δx ges av $N\sigma\Delta x$.

2.1.2 Material med godtycklig tjocklek och medelfri väg

För att härleda sannolikheten för en partikel att interagera med ett material efter att ha färdats en godtycklig sträcka x, betraktas sannolikheten för att inte växelverka, $P(x)$. Om wdx är sannolikheten för att interagera i intervallet $[x, x + dx]$ gäller att

$$P(x + dx) = P(x)(1 - wdx) \implies P(x) = Ce^{-wx}, \tag{2.3}$$

där C är en konstant. Med randvillkoret $P(0) = 1$ fås $C = 1$. Sannolikheten för att partikeln rör sig framåt opåverkad avtar alltså exponentiellt i materialet. Detta betyder att partikeln interagerar med materialet med sannolikhet $P_{\text{int}}(x) = 1 - e^{-wx}$.

Partikelns medelfria väg i materialet λ kan beräknas enligt

$$\lambda = \frac{\int xP(x)dx}{\int P(x)dx} = \frac{1}{w}. \tag{2.4}$$

Värdet på λ beror på material och reaktionstvärsnitt. Genom serieutveckling av P_{int} för en tunn tjocklek Δx fås $P_{\text{int}} \approx \frac{\Delta x}{\lambda}$, vilket genom jämförelse med resultatet i avsnitt 2.1.1 och tillsammans med $\Delta x = dx$ ger

$$\frac{1}{\lambda} = N\sigma \implies P_{\text{int}}(x) = 1 - e^{-N\sigma x}. \tag{2.5}$$

Det framgår att P_{int} ökar med tätheten av spridningspunkter i materialet N och reaktionstvärsnittet σ.

2.1.3 Energiförlust i materia för laddade tunga partiklar

Energiförlusten en laddad partikel genomgår när den färdas genom ett material domineras av dess interaktion med atomära elektroner i materialet. För tunga partiklar med massa m, mycket större än elektronmassan m_e, kommer varje enskild ”kollision” endast bidra till en liten förändring i partikelns bana. Om elektronen ses som obunden och i vila och dess förflyttning under interaktionen antas vara så pass liten att det elektriska fältet kan antas vara konstant, kan energiförlust per längdenhet för den tunga partikeln i materialet härledas. För en noggrann analys av systemet måste elektronen och den tyngre partikeln behandlas kvantmekaniskt vilket först gjordes av Bethe och Bloch *et al.*. Detta ger energideponeringen per längdenhet i materialet som

$$-\frac{dE}{dx} = 2\pi N_a r_e^2 m_e c^2 \rho \frac{Z z^2}{A\beta^2}\left[\ln\left(\frac{2m_e\gamma v^2 W_{\max}}{I^2}\right) - 2\beta^2 - \delta - 2\,\frac{C}{Z}\right], \tag{2.6}$$

med storheter enligt

r_e:	klassisk elektronradie	ρ:	densitet av material
m_e:	elektronens massa	z:	partikelladdning i e
N_a:	Avogradros tal	β:	v/c normerad hastighet
I:	genomsnittlig excitationspotential	γ:	$1/\sqrt{1-\beta^2}$
Z:	atomnummer för material	δ:	densitetskorrektion
A:	atomärt masstal	C:	orbitalkorrektion
$W_{\max}$:	maximalt överförbar energi i en kollision		

För ett givet material ses att

$$-\frac{dE}{dx} = \frac{z^2}{\beta^2} f(\beta), \tag{2.7}$$

där $f(\beta)$ är en funktion av partikelns hastighet (eller energi), vilket medför att en partikel med större laddning penetrerar väsentligt kortare in i materialet. Notera även att den deponerade energin går som β^{-2}. Det innebär att en stor del av partikelns energin kommer att deponeras precis innan partikeln stannar i materialet. Klassiskt kan detta förklaras med att partikeln spenderar en längre tid nära en elektron i materialet.

2.1.4 Energiförlust i materia för elektroner och positroner

Energiförlusten i ett material för elektroner och positroner beror likt fallet för tyngre laddade partiklar till viss del på kollisioner med elektroner i materialet. Skillnaden är att massan är betydligt lägre vilket gör antagandet om att partikelns bana förblir oförändrad ogiltigt. Dessutom kommer energi gå förlorad till bromsstrålning[2] när elektronen eller positronen accelereras i en atomkärnas elektriska fält.

För att ta fram energiförlusten per längdenhet för elektroners och positroners kollisioner i material, tas hänsyn till den lägre massan men även till omöjligheten att särskilja partiklar vid en e^-e^- kollision. Detta gör att den maximalt överförbara energin i en kollision blir $W_{\max} = T_e/2$, där T_e är den kinetiska energin för den ingående elektronen eller positronen. För en detaljerad beskrivning och ett explicit uttryck för $\frac{dE}{dx}$ hänvisas läsaren till Leo [9, s. 35].

2.1.5 Fotoners växelverkan med materia

En stråle av γ-partiklar som passerar genom ett material avtar exponentiellt i intensitet medan enskilda icke växelverkande kvantas energier förblir oförändrade genom materialet. Detta är tvärtemot hur laddade partiklar interagerar i ett material, eftersom de förlorar energi kontinuerligt. Skillnaden är att γ-partiklar har betydligt mindre reaktionstvärsnitt och att de processer som γ interagerar genom avlägsnar partiklarna från strålen:

[2] Ofta används tyskans *Bremstrahlung* istället för bromsstrålning.

- **Fotoabsorption (fotoelektrisk effekt):** Process där γ-kvanta absorberas av bundna elektroner i materialet där atomkärnans närvaro krävs för rörelsemängdskonservering. Den bundna elektronen frigörs med en kinetisk energi som är skillnaden mellan γ-partikelns energi och elektronens bindningsenergi. Reaktionstvärsnittet beror på tillgången av elektroner och minskar med γ-energin, E_γ, och maximeras därför precis över bindningsenergin för elektronen. När alla elektroner är tillgängliga för reaktionen men E_γ är mycket mindre än viloenergin för elektronen går tvärsnittet som $\sigma_\gamma \propto \frac{Z^5}{E_\gamma^{7/2}}$ där Z är atomtalet för materialet.

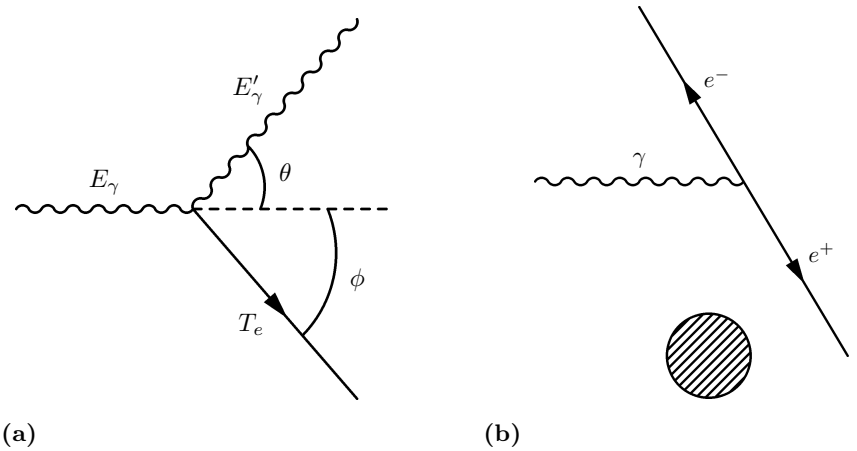
- **Comptonspridning:** När γ sprids mot fria elektroner[3] i materialet. En schematisk beskrivning av Comptonspridning ses i figur 2.1a. Bevarande av relativistisk rörelsemängd ger

$$E'_\gamma = \frac{E_\gamma}{1+\gamma(1-\cos\theta)} \quad \text{och} \quad \cot\phi = (1+\gamma)\tan\frac{\theta}{2}, \tag{2.8}$$

  där $\gamma = \frac{E_\gamma}{m_e c^2}$. Energin överförd till elektronen är skillnaden $T_e = E_\gamma - E'_\gamma$. Det differentiella tvärsnittet för reaktionen fås av Klein-Nishina-formlen [10] beräknad med hjälp av kvantelektrodynamik 1928:

$$\frac{d\sigma}{d\Omega} = \frac{r_e^2}{2[1+\gamma(1-\cos\theta)]^2}\left(1+\cos^2\theta + \frac{\gamma^2(1-\cos\theta)^2}{1+\gamma(1-\cos\theta)}\right), \tag{2.9}$$

  där r_e är den klassiska elektronradien. Det gäller att det integrerade tvärsnittet minskar med fotonenergin. Tvärsnittet kan delas upp i en spridningsdel, andelen energi överförd till den spridda γ-partikeln, och en absorptionsdel, energi överförd till elektronen. Dessa summerar till tvärsnittet för reaktionen. För låga energier ($\ll$1 MeV) dominerar absorptionsdelen medan spridningsdelen dominerar vid höga energier.



Figur 2.1: *Schematiska visualiseringar av (a) Comptonspridning och (b) parproduktion.*

- **Parproduktion:** En γ-partikel sönderfaller till ett elektron-positronpar. Se figur 2.1b för schematisk visualisering. På grund av energi- och rörelsemängdskonservering måste γ-energin vara större än 1022 keV (viloenergin för partikelparet) och det

[3] Om fotonenergierna är mycket större än bindningsenergierna kan dessa försummas och elektronerna behandlas som fria.

måste finnas en tredje kropp i systemet som kan ta upp en rekyl. Positronen kommer saktas ner i materialet genom Coulombväxelverkan och kan då bilda ett bundet tillstånd med en elektron. Slutligen kommer positronen att annihileras[4] med en elektron i materialet och emittera två γ-partiklar spridda 180° mot varandra med energin 511 keV vardera. För relativistiska energier kan tvärsnittet approximeras med Born-approximationen men behöver i allmänhet integreras fram numeriskt, se Bethe och Ashkin [11, s. 325-331]. Parbildning förekommer framförallt för högenergetiska γ-partiklar.

De olika typerna av reaktioner ter sig olika och kommer därför att påverka detektorer på olika sätt. Detta återkommer i avsnitt 2.1.7.

2.1.6 Simuleringsverktyget GEANT4

På grund av komplexiteten i fysiken bakom partiklars växelverkan med materia ställs höga krav på den mjukvara som används för att simulera dessa förlopp. Moderna experiment inom kärn- och partikelfysik har i och med att de skalats upp [12] ökat efterfrågan på verktyg som kan simulera experiment innan de utförs. Ett sådant simuleringsverktyg är det C++-baserade GEANT4 [13][14], som används för att studera allt från enskilda fysikaliska fenomen till fullskaliga experiment. GEANT4 har tagits fram i ett internationellt samarbete mellan organisationer, däribland CERN. Programpaketet är kapabelt att hantera de flesta aspekter av en simulering inklusive ett systems geometri, fundamentala partiklar, partiklars interaktion i materia, elektromagnetiska fält etc. Eftersom dessa system ofta består av ett stort antal kopplade frihetsgrader är de svåra att modellera deterministiskt och simuleras därför med hjälp av Monte Carlo-metoder. Dessa metoder löser matematiska problem med hjälp av sampling av slumpvariabler från givna sannolikhetsfördelningar. Den intresserade läsaren hänvisas exempelvis till Carter och Cashwell [15] för vidare läsning om hur detta kan tillämpas.

2.1.7 Partikeldetektorer

En vanlig metod för att studera atomkärnans struktur är att accelerera joner mot ett strålmål och sedan studera spridning och sönderfall av instabila isotoper som bildas i strålmålet. Denna rapport behandlar en typ av detektorer som används för att detektera och mäta denna strålning, framförallt läggs fokuset på γ-partiklar men även protoner diskuteras. Detektorn består av kristaller som placeras kring strålmålet för att täcka så stor del av rymdvinkeln som möjligt, sett från strålmålet med undantag för där strålröret går igenom.

Varje kristall i detektorn är i sig själv en egen detektor. Mätuppställningarna för både Crystal Ball och DALI2, men även i framtiden för CALIFA, detekterar partiklar genom att kristallen emitterar ljus när partiklar passerar igenom och deponerar energi. Kristallerna scintillerar. Genom att förstärka och konvertera ljussignalen till en elektrisk signal med hjälp av fotomultiplikatorrör (PMT) kan partiklarna registreras. Den emitterade ljussignalens intensitet i kristallen är till god approximation proportionell mot den deponerade energin så partikelns energi kan bestämmas genom att mäta amplituden på den elektriska signalen.

Hela detektorn är segmenterad i olika kristaller för att kunna bestämma var partiklarna detekterades. Det finns detektoruppställningar som inte är segmenterade och har därmed

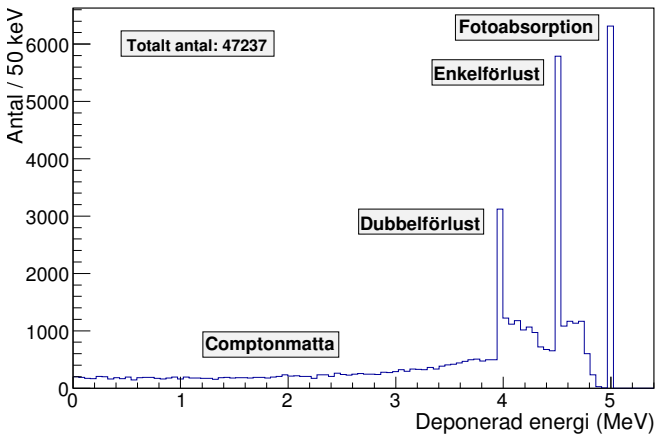
[4] Eftersom positronen är elektronens antipartikel kommer de båda partiklarnas massor omvandlas till energi i form av strålning.

ingen rumsupplösning. Vid experiment som R^3B är den rumsliga upplösningen dock av stor vikt för att på så sätt kunna bestämma riktningen som partikeln emitteras med från strålmålet. Med information om riktningen kan en så kallad Dopplerkorrektion av energin utföras. Detta behövs eftersom partiklarna i jonstrålen är relativistiska vilket beskrivs närmare i avsnitt 2.1.8.

I en segmenterad detektor finns det stor risk att en partikel inte deponerar all sin energi i en kristall innan den sprids in i en annan kristall eller ut från detektorn helt och hållet. I figur 2.2 ses ett simulerat energispektrum där en γ med 5 MeV faller in mot en NaI-kub med sidlängd 5 cm och deponerar energi i kristallen. Effekterna av de tre sätt en γ-partikel framförallt interagerar med materia på, som beskrevs i 2.1.5, ses tydligt i spektrumet.

- **Fotoabsorptionstoppen:** En topp vid fotonernas ursprungliga energi. Ger en korrekt energimätning i detektorn.
- **Enkel- och dubbelförlust vid parproduktion:** Två toppar vid 511 keV och 1022 keV under fotoabsorptionstoppen som uppstår då en eller båda av fotonerna som bildas efter annihileringen lämnar kristallen och förloras. Teoretiskt sett är denna effekt lätt att kompensera för eftersom det är en diskret förflyttning av spektrumet som kan överföras till fotoabsorptionstoppen.
- **Comptonmattan:** Fotonen har vid en Comptonspridning endast lämnat en del av sin energi i kristallen och sedan försvunnit. Vid rekonstruktion är det svårt att se skillnad på en träff där endast en del av energin har deponerats genom Comptonspridning eller om det var en foton med lägre energi. Effekten ger upphov till ett kontinuerligt spektrum vilket betyder att den inte kan kompenseras för på samma sätt som effekterna som uppkommer från parproduktion.

Eftersom energimätningen inte enbart ger en signal vid den korrekta energin (majoriteten av fotonena interagerar med detektorn genom Comptonspridning) krävs det en metod för att rekonstruera vad för partikel detektorn egentligen detekterade.



Figur 2.2: *Simulerat energispektrum för deponerad energi i en kub av NaI med sidlängd* 5 cm *där en γ-partikel med energin* 5 MeV *emitterats mot kuben. Effekterna av Comptonspridning, enkel- och dubbelförlust, som alla försvårar energimätningen, framgår tydligt. Notera att endast ca 13% av alla γ genomgår fotoabsorption.*

Utöver energin som avges av partikeln i kristallen är det även möjligt att registrera när i tid träffen skedde. Tidsinformationen är ofta tillräcklig för att kunna särskilja olika reaktioner i strålmålet (händelser), men ofta otillräcklig för att kunna avgöra i vilken ordning kristallerna träffats i varje händelse. För Crystal Ball är tidsupplösningen ca 3,5 ns och dess innerradie 25 cm. Detta betyder att en partikel som färdas nära ljusets hastighet, 30 cm/ns, kan färdas fram och tillbacka i detektorn flertalet gånger inom tidsupplösningen.

2.1.8 Relativistiska effekter

Acceleratoranläggningar, som exempelvis FAIR, använder sig ofta av jonstrålar med så pass höga kinetiska energier (ca 50-70% av ljushastigheten c) att relativistiska effekter måste beaktas. Två relativistiska fenomen relevanta för det här arbetet är Dopplereffekten respektive strålkastareffekten[5] som ger upphov till skillnader i energier respektive emissionsvinklar i olika inertialsystem[6]. Effekterna uppstår på grund av att reaktionerna mellan jonstrålen och strålmålet sker i strålens inertialsystem, medan detektorn mäter resultatet i labbets inertialsystem[7] [16]. Om strålen har en hastighet βc längs med z-axeln transformerar energi och emissionsvinkel för en γ mellan systemen enligt:

$$E' = \frac{1 - \beta \cos\theta}{\sqrt{1-\beta^2}} E \qquad \cos\theta = \frac{\beta + \cos\theta'}{1 + \beta\cos\theta'} \tag{2.10}$$

där E och θ är partikelns energi och polära emissionsvinkel i labbsystemet samt E' och θ' är motsvarande värden i strålens system [17]. Energin E' benämns som den Dopplerkorrigerade energin och för $\beta = 0,6$ kan den skilja sig från E upp till en faktor två. Eftersom Dopplerkorrektionen är vinkelberoende är det viktigt att kunna bestämma $\cos\theta$. Vinkeltransformationen innebär att då partiklar emitteras isotropt i strålens system kommer de att fokuseras framåt i labbsystemet och detektorn, därav namnet strålkastareffekt.

Dopplerkorrektionen beror endast på θ och inte på den azimutala vinkeln ϕ i xy-planet. Vid vissa typer av reaktioner finns dock en tydlig koppling mellan reaktionsprodukternas inbördes ϕ-värden som uppkommer på grund av rörelsemängdsbevaring. Ett exempel på en sådan reaktion är när en proton i jonstrålen genomgår kvasifri spridning mot en enda nukleon i en atomkärna, detta studeras exempelvis i [18].

2.2 Artificiella neurala nätverk

Maskininlärning är en gren av datavetenskapen som spridits till många tillämpningsområden [19, s. 17]. För att hantera komplexa problem är ett artificiellt neuralt nätverk (ANN) en av de mest effektiva maskininlärningsmetoderna [19, s. 81]. Inspirationen till ANN kommer från den biologiska hjärnan och hur information går via neuroner ihopkopplade med varandra. Detta avsnitt kommer ge en övergripande beskrivning av de tekniker inom ANN som används i projektet. För en utförligare beskrivning av dessa tekniker hänvisas läsaren till exempelvis Mitchell [19] och Goodfellow *et al.* [20].

2.2.1 Principer för artificiella neurala nätverk

Ett ANN består av noder placerade i lager där varje nod får reellvärd indata $\mathbf{x}$ från n noder i föregående lager. I denna rapport behandlas framåtpropagerande nätverk, även kallat *multilayer perceptrons* eller *feedforward networks*, där liksom namnet antyder, informationen endast flödar framåt [20, s. 164]. Nätverket kan därmed beskrivas som en riktad graf vars kanter utgörs av kopplingen mellan noder i två på varandra följande lager, se figur 2.3. Vidare kommer förkortningen MLP användas för dessa nätverk.

I ett MLP får varje lager data från det föregående lagret genom två på varandra följande operationer. Den första operationen är en affin transformation, se ekvation (2.11),

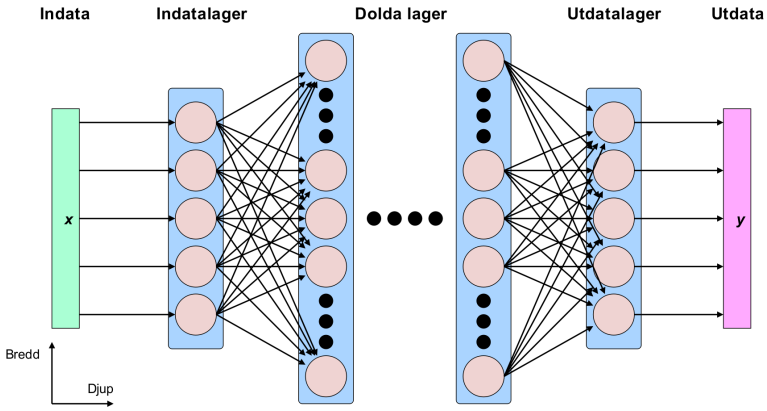
[5] På engelska *headlight effect* eller *Lorentz boost*.
[6] Olika referenssystem som rör sig med konstant hastighet i förhållande till varandra.
[7] Detektorn är stationär i labbet.

och den andra är applicering av en aktiveringsfunktion. Aktiveringsfunktioner behandlas senare i avsnittet. Om $\mathbf{x}$ och $\mathbf{z}$ är radvektorer innehållandes noderna i det föregående respektive det följande lagret (före tillämpning av aktiveringsfunktion), och $\mathbf{x}_n$ innehåller de n stycken noder som ska användas för att beräkna $\mathbf{z}$, ges $\mathbf{z}$ av

$$\mathbf{z} = \mathbf{x}_n\mathbf{W} + \mathbf{b}, \tag{2.11}$$

där $\mathbf{W}$ brukar benämnas som viktningsmatrisen och $\mathbf{b}$ som bias-vektorn. Varje element i dessa tensorer är parametrar som uppdateras när nätverket tränas. Motiveringen bakom dessa benämningar är att varje enskild nod i det följande lagret, z_i, beror på alla noder i $\mathbf{x}_n$ plus en bias b_i, där inverkan från varje element i $\mathbf{x}_n$ styrs av viktningsfaktorerna i $\mathbf{W}$-matrisens kolonn i. Framöver kommer w och b att användas för att beskriva alla viktnings- respektive bias-parametrar hos ett nätverk.

Det första och sista lagret i ett MLP kallas för indatalagret respektive utdatalagret, och lagren däremellan kallas dolda lager[8]. Antalet lager ger djupet[9] av ett nätverk medan antalet noder ger bredden för varje lager och allt sammantaget utgör nätverkets arkitektur. Ett nätverk som har varje nod i ett lager kopplat till alla noder i nästföljande kallas fullt kopplat. I figur 2.3 visualiseras ett sådant nätverk, där de olika benämningarna på lagren är markerade.



Figur 2.3: *En figur av ett fullt kopplat neuralt nätverk som kan ses som en riktad graf där indatan* $\mathbf{x}$ *flödar genom nätverket och till sist ges utdata* $\mathbf{y}$ *från utdatalagret. Varje nod har en koppling mellan sig och alla noder i lagren före och efter. Figuren visar endast ett exempel, antal noder i varje enskilt lager kan variera mellan lagren.*

Vad varje lager ger för utdata bestäms av den icke-linjära aktiveringsfunktion, $g(z)$, som appliceras på lagrets noder. Aktiveringsfunktionen tar nodens utsignal z enligt ekvation (2.11) som argument och vilken funktion som används beror på uppgiften nätverket ska utföra. Den aktiveringsfunktion som i dagsläget är mest använd på de dolda lagren är *rectified linear unit*, ofta kallad ReLU [20, s. 171], definierad som $g(z) = \max\{0, z\}$.

Två vanliga problem där neurala nätverk används är *klassificering* och *regression* [20, kapitel 5]. Vid klassificering ska nätverket ange vilken kategori k indatan $\mathbf{x}$ tillhör. Vid regression ska nätverket istället, givet indata $\mathbf{x}$, ge utdata som ett kontinuerligt värde. I

[8] På engelska *hidden layers*.
[9] Uttrycket *Deep Neural Networks* används för nätverk med många lager.

den här rapporten kommer problemet med att ange multipliciteten av partiklar behandlas som ett klassificeringsproblem. Att få nätverket att ange rätt energi för en specifik partikel eller korrekt emissionsvinkel behandlas istället ett regressionsproblem.

2.2.2 Att träna ett neuralt nätverk

Träning är den process där nätverket lär sig från data och optimerar sina inre parametrar w och b för att kunna prestera så bra som möjligt [20, s. 164]. För att kvantifiera hur väl nätverket presterar införs en kostnadsfunktion som motsvarar felet mellan önskad utdata och den data nätverket ger. Genom att använda träningdata som är associerad med en önskad utdata kan kontrollerad träning[10] utföras. Termen kontrollerad träning kommer från att algoritmen förses med information om vad den ska åstadkomma, likt en tränare eller lärare som visar hur något ska ske [20]. Om träningsdatamängden inte är associerad till en önskad utdata kan nätverket istället försöka hitta en viss struktur eller beteende i indatamängden och rekonstruera detta. Det kallas då okontrollerad träning[11] och kräver ett annat mått på hur väl nätverket presterar. I denna rapport kommer endast kontrollerad träning att användas eftersom det finns en önskad utdata specificerad.

2.2.3 Kostnadsfunktion

Kostnadsfunktionen är den funktion som ett ANN försöker att minimera under träningsprocessen och valet av kostnadsfunktion är ofta kopplat till valet av aktiveringsfunktion [20, s. 177]. En grundläggande kostnadsfunktion är det genomsnittliga kvadratfelet mellan en vektor y bestående av värdena på utnoderna och en vektor $\hat{y}$ som beskriver önskad utdata för en viss händelse:

$$C(y(w,b),\hat{y}) = \frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2, \tag{2.12}$$

där n är antalet utnoder n och $C(y(w,b),\hat{y})$ betecknar kostnadsfunktionens beroende av vikterna w och biaserna b. Denna kostnadsfunktion är som mest användbar vid regressionsproblem [20].

Vid klassificeringsproblem där endast en kategori åt gången är korrekt är det vanligt att först konvertera värdena hos utnoderna till en sannolikhetsfördelning med hjälp av aktiveringsfunktionen softmax: [20, s. 180]:

$$\text{softmax}(y_i) = \frac{\exp(y_i)}{\sum_{j=1}^{n}\exp(y_j)}. \tag{2.13}$$

Det korrekta svaret kan representeras som en vektor där den rätta kategorin motsvaras av ett element med värdet 1 medan de andra har värdet 0, en så kallad one-hot-vektor. Den vanligaste kostnadsfunktionen vid klassificeringsproblem är cross-entropy [6, kap. 3]. Eftersom både utdata från nätverket och rätt svar nu är sannolikhetsfördelningar, kan cross-entropy-funktionen för sannolikhetsfördelningar användas:

$$H(y(w,b),\hat{y}) = -\sum_{i}\hat{y}_i \ln(y_i). \tag{2.14}$$

[10] På engelska *Supervised learning.*
[11] På engelska *Unsupervised learning.*

Fördelen med cross-entropy är att inlärningstakten ökar ju större fel nätverket har vilket inte är fallet med genomsnittliga kvadratfelet [6, kap. 3]. För att utvärdera ett nätverk där endast en kategori kan vara rätt åt gången, jämförs indexet för det maximala argumentet hos utnoderna med indexet för ettan i one-hot-vektorn.

2.2.4 Bakåtfortplantning och optimering av nätverkets parametrar

Under träningen är målet för nätverket att minimera kostnadsfunktionen genom att optimera de inre parametrarna samt hitta den bästa kombinationen av dessa [19]. Ett vanligt tillvägagångssätt är träning baserad på gradienten av kostnadsfunktionen, så kallad *gradient descent*. För att beräkna gradienten av kostnadsfunktionen används metoden bakåtfortplantning. Gradienten beräknas iterativt med avseende på parametrarna i varje lager θ, det vill säga w och b, genom att gå bakåt i nätverket och med hjälp av kedjeregeln återanvända resultatet från lagret innan. Parametrarna uppdateras sedan genom att ta ett litet steg i negativa gradientens riktning enligt

$$\theta_{n+1} = \theta_n - \epsilon \nabla_{\theta_n} C(\theta_n) \tag{2.15}$$

där ϵ är inlärningstakten[12], som justerar storleken på steget. För att minska beräkningsbelastningen kan träningen och gradientens beräkning utföras på en mindre slumpmässigt vald delmängd av träningsdatamängden istället för på hela. Detta tillvägagångssätt kallas *stochastic gradient descent* (SGD). Gradienten uppdateras på så sätt snabbare men med mindre precision.

En variant av SGD är SGD med momentum. Eftersom SGD beräknar gradienten på en delmängd av datamängden kan gradienten bli ”brusig” och för att erhålla bättre säkerhet kan stegriktningen delvis bero av riktningen på steget innan [19, s. 100]. Det gör även att lokala minima lättare undviks eftersom riktningen tenderar att fortsätta peka åt samma håll.

2.2.5 Optimeringsalgoritmen Adam

Det finns olika träningsmetoder och optimeringsalgoritmer som använder sig av varianter av bakåtfortplantning. I den här rapporten används Adam, en optimeringsalgoritm som bygger på SGD med momentum. Istället för att samma inlärningstakt används för alla parametrar så anpassar Adam individuella värden på takten och erhåller på så sätt olika steglängder. Detta eftersom någon parameter gynnas av att gå snabbare åt ett håll medan andra istället hamnar längre från sitt egna minimum vilket inte är önskvärt. För en utförligare beskrivning av optimiseringsalgoritmen Adam hänvisas läsaren till artikeln *Adam: A method for stochastic optimization* [21].

2.2.6 Uppdelning av data och överträning

För att erhålla bäst generalisering hos nätverket brukar datamängden delas in i tre olika delar: träningsdata, evalueringssdata och testdata. Träningsdatamängden är den del som används som indata vid nätverkets träning och som påverkar de interna parametrarna. Evalueringsdatamängden är den data som används vid evaluering av nätverkets prestation efter visst antal träningsiterationer. Resultatet används sedan för att optimera nätverkets hyperparametrar [20, kap. 5] vilka är de parametrar som bestämmer kapaciteten hos nätverket, alltså förmåga att hitta passande funktioner. Om testdatamängden

[12] På engelska *learning rate*.

användas för optimering av hyperparametrarna skulle inte en pålitlig uppskattning av generaliseringen kunna göras med samma data efteråt. Att nätverket lyckas bra med att associera indata $\mathbf{x}_{\text{träning}}$ till $\hat{\mathbf{y}}_{\text{träning}}$ betyder dock inte att den kommer prestera bra på en annan datamängd, som till exempel testdatamängden. Nätverket kan bli övertränat, så kallad *overfitted*, och hitta funktioner som bara passar till träningsdata, vilket leder till dålig generalisering.

Det finns diverse metoder för att undvika överträning. Ett sätt är att använda en tillräckligt stor datamängd med avseende på nätverkets kapacitet. I det här projektet har inte datastorleken varit ett problem. En annan metod som används för att undvika överträning är *early stopping* [20, s. 243], där träningen av nätverket avbryts om kostnadsfunktionen, beräknad med evalueringsdata, inte blivit bättre än sitt tidigare minima inom ett visst antal iterationer. Ytterligare ett sätt att undvika överträning är att använda *dropout* [20, s. 255]. Det innebär att godtyckligt valda noder blockeras och inte kommer ha någon inverkan under en träningsiteration. Därmed minimeras risken att nätverket helt kan förlita sig på informationsflöde via specifika noder.

2.2.7 Faltande nätverk

För bildigenkänning är ett faltande nätverk, även kallat *convolutional neural network* (CNN), att föredra [6, kap. 6]. Ett CNN är ett framåtpropagerande nätverk med ett eller flera faltningslager. Ett faltningslager använder bildens struktur genom att translatera ett filter över bilden. Ett filter är uppbyggt som en flerdimensionell matris där varje element är en vikt som uppdateras efter varje träningssteg. För en 2D-bild är det exempelvis en rektangulär matris. För varje position filtret har på bilden fås en skalär som utdata vilken beräknas genom att göra elementvis multiplikation mellan filtret och den matris av noder som filtret täcker, och sedan addera alla element. Till detta värde adderas en bias specifik för varje filter. Precis som för tidigare beskrivna lager, appliceras oftast aktiveringsfunktionen ReLU på faltningslagren. Varje filter tränas till att känna igen en specifik egenskap hos bilden.

Ett CNN har ofta *pooling*-lager placerat efter ett eller flera av faltningslagren, där en översättning av pooling skulle kunna vara sammanslagning. Syftet med pooling-lagren är att sammanfatta informationen från faltningslagret [6, kap. 6]. Detta sker med ett filter som betraktar ett område av närliggande noder, sammanfattar deras värden till en skalär och sedan förflyttas till ett nytt område. På så sätt minskar mängden data till nästkommande lager och träningen kan snabbas upp. Att använda pooling ger även en liten translationsinvarians vilket betyder att en liten förflyttning av innoderna oftast inte förändrar pooling-lagrets utdata [20]. Detta är användbart om närvaron av en viss egenskap är viktig, snarare än att veta dess exakta placering. Två exempel på pooling är max-pooling och medelvärdes-pooling, där varje filter tar maxvärdet respektive medelvärdet av de noder filtret verkar på.

3

Konstruktion av modell

Detta kapitel presenterar tillvägagångssättet för konstruktion och träning av neurala nätverk för rekonstruktion från Crystal Ball-data. Arbetsprocessen var till stor del iterativ, och på grund av detta presenteras även de resultat som följde av nya metodval. Vissa av dessa resultat följs av en kortare diskussion. Orsaken till den iterativa arbetsprocessen var att få liknande arbeten kunde hittas och bristen på ett gediget teoretiskt ramverk kring neurala nätverk [22]. Den slutgiltiga evalueringen av de metoder i detta kapitel som bedömts vara de mest relevanta, görs i kapitel 4.

Två olika nätverksstrukturer har konstruerats: ett nätverk som fokuserat på att med regression rekonstruera γ-partiklars energi samt emissionsriktning i termer av vinklar och ett klassifikationsnätverk vars uppgift var att rekonstruera multipliciteten. Syftet med nätverket för multiplicitetsrekonstruktion var att använda det för att undersöka olika strategier för faltning. Den största utmaningen vid utvecklandet av nätverket för energi- och vinkelrekonstruktion, var kostnadsfunktionens utformning, och detta kapitel fokuserar därför till största del på denna.

3.1 Simulering av data

För att träna ett neuralt nätverk krävs stora mängder realistisk data. I detta arbete används simulerad data producerad med *ggland* [23], ett program som med GEANT4-verktyget (se avsnitt 2.1.6) simulerar kärnfysikexperiment. Programmet ggland innehåller färdiga modeller av flertalet detektorer som exempelvis Crystal Ball och DALI2, vilket underlättar simuleringsprocessen. I simuleringarna placeras ett godtyckligt antal partikelkällor, kallade *guns*, i detektorns centrum. Dessa kan i varje händelse avge en partikel av vald typ med energi och riktning slumpmässigt genererade från valda fördelningar för att simulera resultatet som kan komma från strålmålet i ett verkligt experiment. Partiklarna färdas sedan fram till och in i detektorns kristaller där energi möjligen deponeras, och reaktioner som kan skapa sekundära partiklar kan äga rum. Hädanefter kommer ordet ”händelse” referera till just denna process med ett visst antal guns.

Från ggland erhålls data från varje kristall i vald detektor för varje simulerad händelse, vilket ska motsvara det som fås ur ett verkligt experiment. I simuleringen läggs ett normalfördelat fel till på varje uppmätt energi, där standardavvikelsen är 5 % av respektive energideponering, vilket ska motsvara detektorns mätosäkerhet. Denna data ges till nätverket vars uppgift är att rekonstruera de ursprungliga partiklarnas energier och emissionsvinklar. Från ggland hämtas även de värden som beskriver de ursprungliga partiklarna, vilka används av nätverket för att träna emot.

Samtliga data från simuleringen sparades som C++-objekt i en trädstruktur i en ROOT[1]-fil [24][25]. Från ROOT-filen konverterades simulerad data över till textfiler för inläsning av programmet som definierar nätverket. Träffdata från varje händelse representerades i en textfil som en lista där elementen motsvarar uppmätt deponerad energi i varje detektorkristall. Indata till nätverket kan därför ses som en endimensionell bild med energier som pixelvärden. Egenskaperna för de ursprungliga partiklarna sparades i liknande format.

För att undvika händelser där en eller flera fotoner inte deponerar någon eller endast en liten del av sin energi, för att sedan lämna detektorn, sattes ett tröskelvärde så att endast händelser där varje foton deponerar minst 90 % av sin energi sparas. Det hade varit en otacksam uppgift för nätverket att behöva skilja på händelser med en foton och händelser med tre fotoner varav två lämnat detektorn utan att deponera någon energi. Detta scenario kan naturligtvis uppkomma i verkligheten, men träningen underlättas av att försaka dem.

3.1.1 Simuleringsparametrar

Vid simuleringarna genererades γ-partiklar från punktkällor placerade i centrum av vald detektor (Crystal Ball eller DALI2). Koordinatsystemet orienterades med origo placerad i detektorns centrum och z-axeln längs med den tänkta strålen. Partikelenergierna varierades likformigt i intervallet 0,05 till 10 MeV och emissionsvinklar fördelades isotropt i labbsystemet. För varje γ-multiplicitet mellan 1 och 6 simulerades ungefär 10^6 händelser. Genom att kombinera data med olika multipliciteter kunde nätverket exponeras för varierande antal γ.

3.2 Beräkningsresurser

För att utföra de beräkningar som krävs för att träna ett neuralt nätverk är det mer effektivt att använda grafikprocessorer (GPU) snarare än konventionella centralprocessorer (CPU). Det beror på att GPUer är explicit designade för att effektivt utföra beräkningar parallellt, vilket neurala nätverk till stor del bygger på.

I det här arbetet har en del av beräkningarna utförts på datorfarmen Kebnekaise [26] vid High Performance Computing Center North (HPC2N) i Umeå. Där användes grafikkort av modell Nvidia Tesla K80. Utöver detta har även beräkningar utförts på grafikkort av modell Geforce GTX 1080 Ti och Geforce GTX 1060 på lokala arbetsstationer på Chalmers.

3.3 Programvara för uppbyggnad av neurala nätverk

För att konstruera neurala nätverk har den fria programvaran TensorFlow [27] i kombination med programmeringsspråket Python använts. TensorFlow är utvecklat för maskininlärning och kan användas på många olika nivåer, antingen genom högnivå-gränssnitt eller som i det här arbetet, lågnivå-gränssnitt. Som en introduktion till maskininlärning utnyttjades introduktionsguider [28] från TensorFlows hemsida och stor del av arbetets första tid gick åt till att bekanta sig med programvaran. Eftersom TensorFlow bara är ett verktyg i det här arbetet lades fokus på att förstå de mest väsentliga delarna.

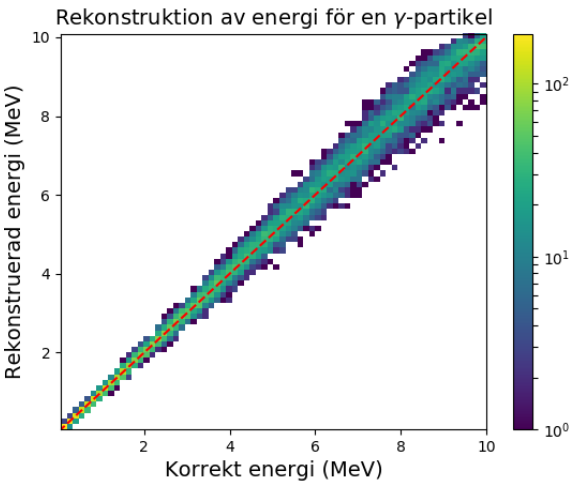
[1] ROOT är ett ramverk för datahantering utvecklat av CERN.

3.4 Energi- och vinkelrekonstruktion för Crystal Ball

Ett MLP för regression konstruerades med 162 innoder och ett varierande antal utnoder. Antalet innoder var lika med antalet kristaller i Crystal Ball och utnodernas antal berodde på hur många parametrar som skulle återges. Vid träning har simulerad data använts enligt avsnitt 3.1. Träningsdata delades slumpvis upp i mindre grupper om 100 händelser och träningen utfördes med hjälp av optimeringsmetoden Adam, tidigare beskriven i avsnitt 2.2.5. Alla lager i nätverket var fullt kopplade och om inget annat anges bestod arkitekturen av 10 dolda lager med 256 noder vardera. Detta valdes som en godtycklig standard eftersom ingen större skillnad från andra val kunde observeras, se appendix B.2.

Att rekonstruera en händelse i en detektor är ett komplext problem eftersom alla inkommande partiklar samtidigt ska få korrekt energi och korrekta emissionsvinklar. Problemet delades därför upp i delproblem och olika varianter av genomsnittliga kvadratfelet implementerades som kostnadsfunktion för nätverket, där utseendet berodde på vilket delproblem som var i fokus. Att hitta ytterligare inspiration till kostnadsfunktionen var svårt eftersom applicering av neurala nätverk på denna typ av problem i dagsläget inte är vanligt förekommande i litteraturen.

3.4.1 Utveckling av kostnadsfunktion för en γ-partikel



Figur 3.1: *Energirekonstruktion för en γ-partikel. Den ljusare nyansen kring den streckade linjen indikerar att nätverket lyckats bra med att rekonstruera energin för många händelser. Konformen uppkommer antagligen på grund av det procentuella mätfelet som läggs till vid simuleringen.*

Ett enkelt delproblem är att få nätverket att rekonstruera energin för en enda γ emitterad från strålmålet inuti Crystal Ball. I detta fall har nätverket en utnod som anger energin som ett flyttal. Kostnadsfunktionen definierades som det genomsnittliga kvadratiska felet mellan den givna energin $\hat{E}$ och den av nätverket återgivna energin, E, enligt ekvation (2.12). Hädanefter kommer ett korrekt värde på en parameter indikeras med ”$\widehat{\text{parameter}}$”. Figur 3.1 visar hur nätverket rekonstruerat energin från evalueringsdata efter $3 \cdot 10^5$ träningsiterationer. Färgskalan i figuren anger antalet rekonstruerade partiklar med motsvarande energi. Den höga tätheten kring den streckade linjen visar att nätverket lyckats väl för majoriteten av händelserna. På grund av tröskelvärdet på 90 % (se avsnitt 3.1) så räcker det dock att addera de deponerade energivärdena för att göra en framgångsrik rekonstruktion av partikelns deponerade energi. Spridningen av rekonstruerad energi ökar med korrekt energi och ses som en konformad fördelning i figur 3.1. Detta beror troligtvis på det procentuella mätfel som läggs till vid simuleringen.

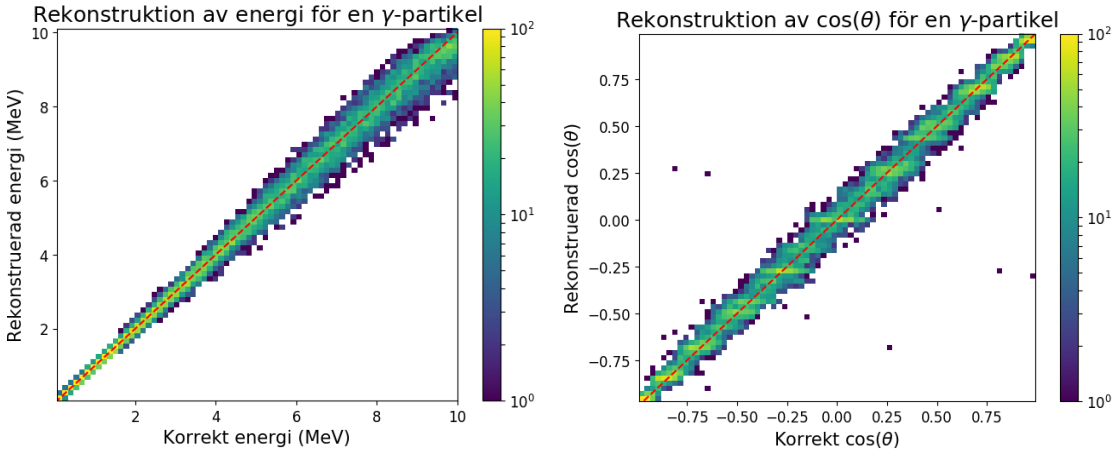
För att även kunna inkludera partikelns emissionsvinkel θ gavs nätverket två utnoder, en för energi och en för $\cos\theta$. Att använda $\cos\theta$ som träningsparameter var naturligt, eftersom den används för Dopplerkorrektion, se ekvation (2.10). Kostnadsfunktionen utö-

kades med en term till

$$C(y(w,b),\hat{y}) = \frac{1}{n}\sum_{j=1}^{n}\left[\lambda\left(\frac{\Delta E_j}{\hat{E}_j}\right)^2 + (\Delta\cos\theta_j)^2\right], \tag{3.1}$$

där n är antalet händelser funktionen evalueras på, $\Delta E_j = E_j - \hat{E}_j$ och $\Delta\cos\theta_j = \cos\theta_j - \widehat{\cos\theta_j}$. Viktfaktorn λ användes för att få ett så optimalt förhållande mellan termerna som möjligt. Effekten av λ studeras närmare i avsnitt 3.4.4. Energitermen i funktionen sattes även till att beskriva det relativa felet istället för det absoluta. Detta för att nätverket ska straffas mindre om det exempelvis säger fel på 1 MeV för en partikel med energi 9 MeV än en partikel med energi 2 MeV.

Figur 3.2 visar resultatet från en träning med kostnadsfunktionen enligt ekvation (3.1). Resultatet indikerar att nätverket klarar av rekonstruktion av båda parametrarna samtidigt och att ingen större skillnad i kvalitén för energirekonstruktionen uppkommer jämfört med figur 3.1 där en absolut kostnadsfunktion använts. De partier med högre intensitet i figur 3.2b kan kopplas till Crystal Balls segmentering. Eftersom nätverket inte kan avgöra var i en kristall en träff skett är det kristallens placering som tenderar att återges medan det korrekta svaret är partikelns faktiska emissionsvinkel.

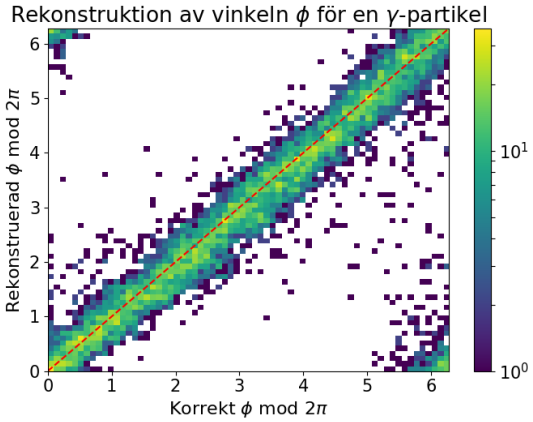


(a) **(b)**

Figur 3.2: *Energi- och vinkelrekonstruktion i (a) respektive (b) för en γ. Den streckade linjen indikerar korrekt svar och båda resultaten följer denna linje väl. De ljusa platåerna i (b) beror på de diskreta vinkelvärdena hos detektorns kristaller. Till skillnad från figur 3.1 användes en kostnadsfunktion där energitermen bestod av det relativa felet istället för det absoluta felet, men energirekonstruktionen uppvisar inte några större skillnader.*

Till sist adderades en term för återgivning av emissionsvinkeln ϕ i xy-planet. Denna vinkel är primärt inte av intresse för γ-partiklar i detta fall, men i de fall där kvasifri spridning studeras är det önskvärt att kunna kartlägga vilka träffar som har en skillnad $\Delta\phi \approx 180°$, se avsnitt 2.1.8. Kostnadsfunktionen med avseende på alla tre parametrar blir

$$C(y(w,b),\hat{y}) = \frac{1}{n}\sum_{j=1}^{n}\left[\left(\frac{\Delta E_j}{\hat{E}_j}\right)^2 + (\Delta\cos\theta_j)^2 + \left(\frac{(\Delta\phi_j+\pi)\mathrm{mod}2\pi) - \pi}{2\pi}\right)^2\right], \tag{3.2}$$



Figur 3.3: *Rekonstruktion av emissionsvinkeln ϕ för en γ. Detta har skett i kombination med rekonstruktion av energi och $\cos\theta$ vilket innebär att alla tre parametrar går att rekonstruera samtidigt. Vinkelns periodicitet är anledningen till effekterna i två av figurens hörn. Notera att det på samma sätt som för $\cos\theta$ uppkommer diskreta värden med högre densitet kring den streckade linjen.*

där $\Delta\phi_j = \phi_j - \hat{\phi}_j$ och modulusoperatorn tillåter nätverket att ange värden på ϕ inom valfritt intervall mod 2π. Detta på grund av vinkelns periodicitet. Genom att addera π till differensen och beräkna modulus av $\pi + \Delta\phi$, uppstod inte problemet med att en liten negativ differens kan tolkas som ett stort fel. Värdet på differensen förskjuts π radianer och på så sätt bort från brytpunkten.

Figur 3.3 visar att rekonstruktion av vinkeln ϕ med kostnadsfunktionen enligt (3.2) är möjlig. En effekt av vinkelns periodicitet syns tydligt i två av figurens hörn. Notera även att de diskreta vinkelvärdena inte är lika tydliga som i figur 3.2b. Detta beror troligtvis på hur kristallerna är placerade runt z-axeln, något som kan ses i figur 1.2.

3.4.2 Användning av kostnadsfunktion för flera γ-partiklar

Med hjälp av den framtagna kostnadsfunktionen för rekonstruktion av händelser med en γ-partikel, beskriven i föregående avsnitt, utfördes tester med fler än en γ. Antalet utnoder hos nätverket ändrades till $m \cdot p$ där m var maximala antalet partiklar och p antalet parametrar som skulle bestämmas. Vid träning användes först data med fixerad multiplicitet där antalet γ som mest var sex, och sedan användes träningsdata med varierande multiplicitet. I båda fallen testades den absoluta och relativa kostnadsfunktionen.

Ett problem som uppstår vid varierande multiplicitet är formatet på nätverkets indata då den måste vara identisk under hela träningsprocessen. Detta löstes genom att fylla ut data med färre antal partiklar än det maximala med nollor, och på så sätt generera ”partiklar” med värdet 0 MeV på energin och värdet 0 på $\cos\theta$. För att fortsättningsvis kunna använda det relativa energifelet i kostnadsfunktionen introducerades en offset, E_{offset}, till nämnaren för att undvika division med 0. Den valdes till 0,1 MeV för att vara i storleksordning med de lägsta energivärdena och E_{offset} kan ses som en av nätverkets hyperparametrar som kan optimeras.

För att undersöka om neurala nätverk kan rekonstruera en verklig händelse behövs primärt värdet på energin och $\cos\theta$. Relativa kostnadsfunktionen för detta i fallet med flera γ ges av

$$C(y(w,b),\hat{y}) = \frac{1}{n}\sum_{j=1}^{n}\left[\sum_{i=1}^{n_\gamma}\lambda\left(\frac{\Delta E_{j,i}}{\hat{E}_{j,i}+E_{\text{offset}}}\right)^2 + (\Delta\cos\theta_{j,i})^2\right], \tag{3.3}$$

där n är antalet händelser funktionen evaluerades över och n_γ är antalet γ-partiklar som ska rekonstrueras. För att använda den absoluta kostnadsfunktionen sätts nämnaren i energidelen till en konstant, E_{norm}. Även kostnadsfunktionen enligt ekvation (3.2) som

används för rekonstruktioner där ϕ inkluderas, anpassades till flera partiklar med lyckat resultat. Detta har dock inte undersökts närmare.

Den största utmaningen när händelser med flera partiklar undersöktes, var den betydligt mer komplicerade träningsprocessen. Rekonstruktionen för en partikel kan ses som en grupp av utnoder där noderna internt alltid placeras i samma ordning. När energi och vinklar ska återges för flera partiklar i samma händelse uppkommer problemet att avgöra vilken grupp av utnoder som tillhör vilken partikel. Eftersom nätverket strävar efter att minimera kostnadsfunktionen görs ett antagande att den korrekta kombinationen är när en grupp av utnoder matchar med korrekta värden så att kostnadsfunktionen blir minimal. Tre olika metoder baserade på denna strategi testades och beskrivs närmare i avsnitt 3.4.3.

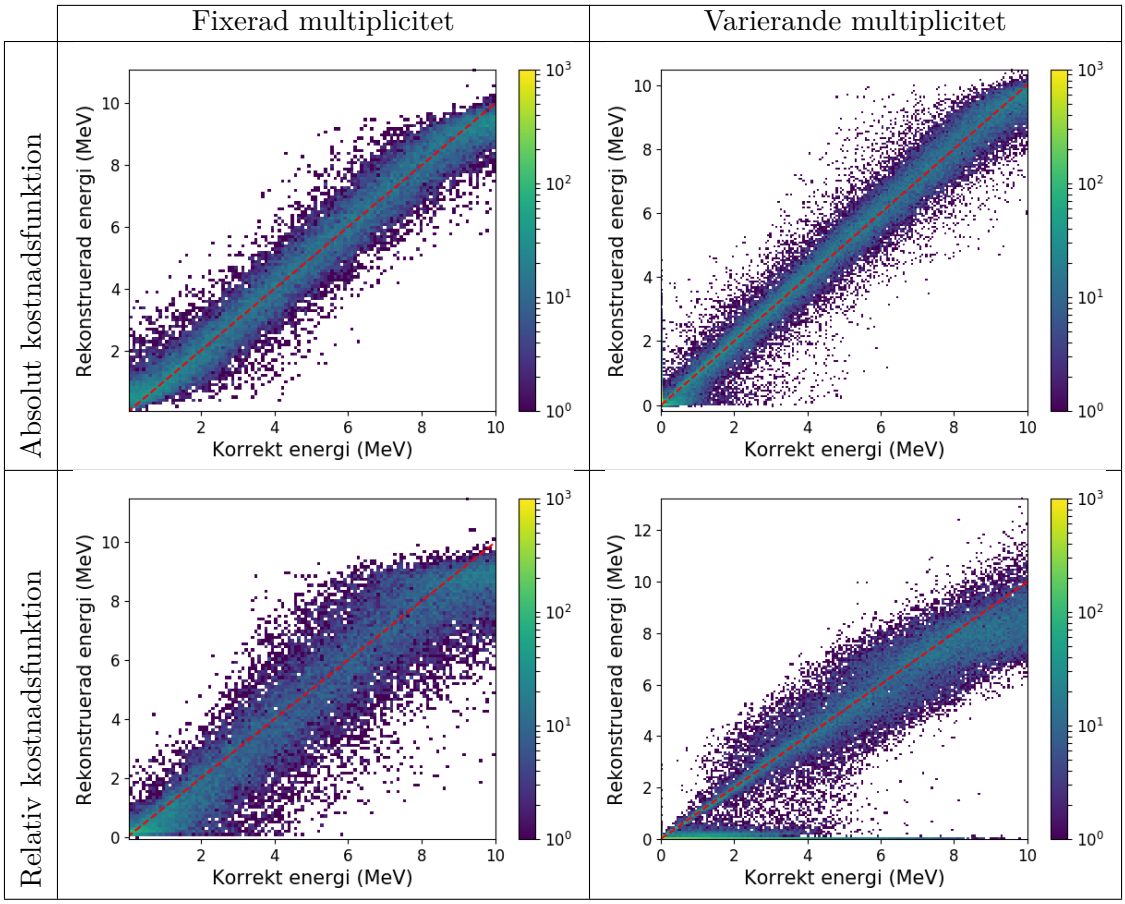
3.4.2.1 Jämförelse mellan absolut och relativ kostnadsfunktion

I figur 3.4 jämförs energirekonstruktion med absolut respektive relativ kostnadsfunktion. För att isolera kostnadsfunktionerna rekonstruerades här enbart energier. Nätverken har tränats och evaluerats med data där multipliciteten fixerats till tre γ eller varierat mellan 1–3 γ. De händelser där den korrekta energin var 0 MeV och nätverket rekonstruerat den som mindre än 25 keV, har tagits bort för att göra figuren tydligare. De streckade linjerna indikerar placeringen för en perfekt rekonstruktion av energin.

Att notera är den s-form som uppkommer med kombinationen av absolut kostnadsfunktion och fixerad multiplicitet vilken inte är lika tydlig i något av de andra tre fallen. Nätverket tenderar att återge de högsta energierna för lågt och de lägsta energierna för högt, något som även skiljer sig från enpartikelfallet där en konform uppkom i rekonstruktionen, se figur 3.1. S-formen antas därför bero på att den totala energin i detektorn nu fördelas på flera γ istället för en. Sannolikheten att en av dessa ska ha energi i extremfallen verkar enligt nätverket vara liten. Vid användning av relativ kostnadsfunktion och fixerad multiplicitet försvinner s-formen för de lägre energierna, vilket rimligen beror på att nätverket straffas hårdare för fel vid dessa värden. För att minimera s-formen för de högre energierna kan träning ske på ett större intervall än evalueringen. Detta kan ses i samma figur genom att bara studera evalueringsenergin upp till exempelvis 6 MeV.

Ytterligare en utmärkande effekt är den ”matta” av rekonstruerade energivärden som uppkommer i figur 3.4 vid varierande multiplicitet och relativ kostnadsfunktion. Eftersom ”mattan” inte uppkommer i något annat av fallen kan den inte enbart förklaras av att antalet frihetsgrader ökas till följd av varierande multiplicitet eller enbart av att en relativ bestraffning används. Däremot skulle de många nollor som tillförts för att förmedla korrekt svar i rätt format, påverka kombinationen av dessa så pass mycket att ”mattan” uppkommer. På grund av omformateringen blev så många som 1/3 av alla korrekta värden nollor. Eftersom nätverket för låga energier straffas hårt även för relativt små fel, kan det i genomsnitt blir mer lönsamt för nätverket att sätta partikelenergier till 0 MeV, då det värdet är så vanligt förekommande. Detta verkar dock inte gälla för alla händelser.

Det utfördes småskaliga tester med en kontrollnod där nätverket även fick förutspå om indata var en partikel- eller utfyllnadsenergi. Om nätverket återgav att en energi var en utfyllnad straffades det inte med avseende på denna. Nätverket var nu inte tvungen att rekonstruera så många 0 MeV-partiklar och förhoppningen var att effekten av utfyllnaden skulle minska. Testerna gav inget större resultat och idén behöver utvecklas för en bättre



Figur 3.4: *Skillnaden mellan absolut och relativ kostnadsfunktion samt fixerad och varierande multiplicitet. För fixerad multiplicitet i kombination med absolut kostnadsfunktion uppkommer en s-form vilket inte tydligt kan ses i något av de övriga fallen. För relativ kostnadsfunktion med varierande multiplicitet uppkommer en "matta" av energier som, vid jämförelse med de två närliggande rekonstruktionerna, kan antas bero på två saker: format på den indata som användes och den relativa bestraffningen vid träning.*

utvärdering.

Studeras enbart energirekonstruktionen enligt figur 3.4 kan den absoluta kostnadsfunktionen ses som det optimala valet då den har en jämn spridning av energierna och ingen "matta" uppkommer. För att konstatera vilken kostnadsfunktion som är bäst lämpad för rekonstruktion av en verklig händelse kan dock en bättre indikering erhållas vid jämförelse med befintliga algoritmer, vilket studeras i kapitel 4.

3.4.3 Beräkningsmetoder för kostnadsfunktion för flera γ-partiklar

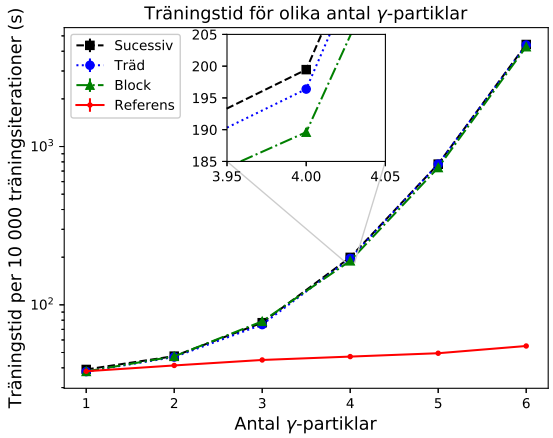
Som beskrivits i avsnitt 3.4.1 studeras alla permutationer av förutspådda partiklar och jämförs med de korrekta för att beräkna kostnadsfunktionen. Det betyder att för n partiklar finns det $n!$ fall att studera.

För att finna den permutation som minimerar kostnadsfunktionen användes TensorFlows inbyggda binära operator *minimum* som agerar på två tensorer av samma storlek och ger ett elementvis minimum. Tre olika metoder har utvecklats:

- **Successiv:** Itererar över alla permutationer och finner minimum mellan den hittills

bästa och den nuvarande permutationen.

- **Träd:** Minimumoperatorn används i en trädstruktur. I varje minimeringssteg paras alla permutationer ihop och minimumet tas mellan paren. Om antalet kvarvarande permutationer är udda förs endast den kvarvarande permutationen vidare till nästa steg. Denna metod gör att flera steg kan göras parallellt internt i TensorFlow jämfört med den successiva metoden.
- **Block:** Alla permutationer klumpas ihop till en matris. Om matrisen innehåller ett jämnt antal permutationer delas matrisen i två lika stora delar och minimumoperatorn agerar på dessa. Om matrisen innehåller ett udda antal permutationer tas först en permutation bort så att minimumet kan beräknas som i det jämna fallet varefter den udda permutationen läggs till igen. Denna metod kräver färre funktionsanrop på minimumoperatorn är de tidigare metoderna.



Figur 3.5: *Träningstid för de olika metoderna för att finna den korrekta permutationen. Referenskurvan är en kostnadsfunktion där alla permutationer av kostnadsfunktionen konstrueras utan minimumoperationer. Träningstiden ökar snabbare än exponentiellt med antalet γ-partiklar. Beräkningen är gjord på en Tesla K80 GPU med en gruppstorlek på 100 händelser. Notera att felstaplarna är mindre än markörerna.*

Även om minimumoperatorn anropas färre gånger av vissa metoder så är fortfarande antalet jämförelser på lägsta nivå samma. I tabell 3.1 ses antal steg metoderna behöver och antal anrop utförda av minimumoperatorn. I figur 3.5 ses hur träningstiden för ett konstant antal träningsiterationer ökar med antalet partiklar som ska rekonstrueras. Träningstiden växer snabbare än exponentiellt i detta fall. På en Tesla K80 med en gruppstorlek på 100 händelser under träningen kan träningstiden approximeras med $T \propto \exp(0.05 n_\gamma^{2.6})$ där n_γ är antalet γ-partiklar som ska rekonstrueras. De olika metoderna ger ungefär samma tidsåtgång där blockmetoden är marginellt snabbare. Från figur 3.5 ses även att det är minimumberäkningen som är tidskrävande och inte att konstruera permutationerna. Beräkning av minimum är dock relativt enkel operation. Troligt är att det är kopplingen mellan minimumet och bakåtfortplantningen som är krävande.

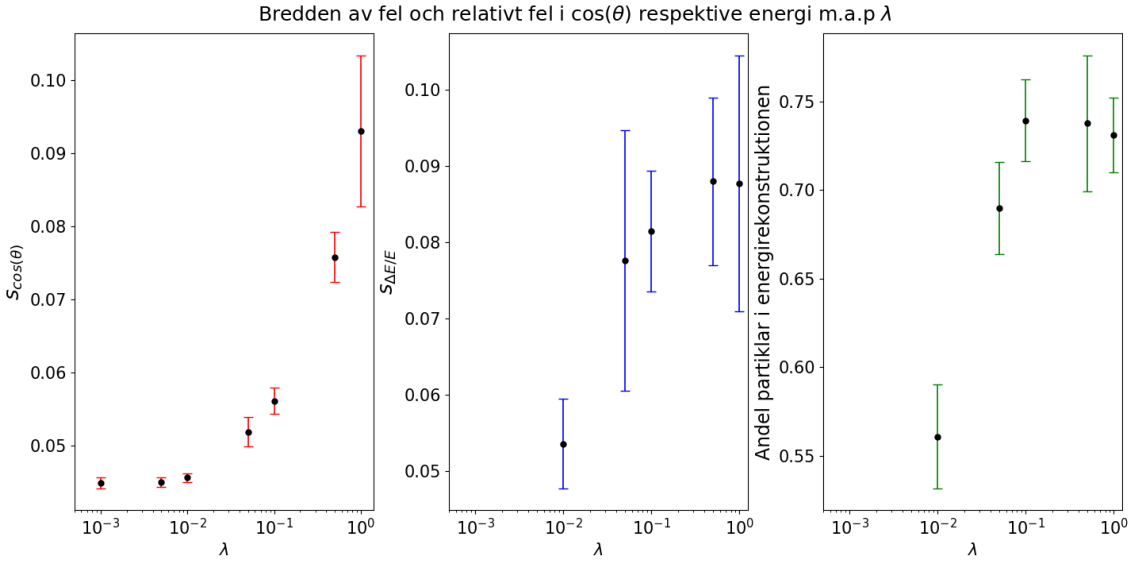
Tabell 3.1: *Jämförelse mellan de tre metoder som använts för att beräkna kostnadsfunktionen för olika antal partiklar. Djupet är antalet steg som behöver göras i följd och därmed inte kan göras parallellt. Anrop är antalet gånger minimumfunktionen anropas.*

Metod / Partiklar	Successiv		Träd		Block	
	Djup	# Anrop	Djup	# Anrop	Djup	# Anrop
2	2	1	2	1	2	1
3	6	5	4	5	4	3
4	24	23	6	23	6	5
5	120	119	8	119	8	7
6	720	719	11	719	11	10

3.4.4 Viktning av kostnadsfunktionens termer

Som beskrivits i avsnitt 3.4.2 multiplicerades en viktningsfaktor λ med energitermen i kostnadsfunktionen för att undersöka det optimala förhållandet mellan termerna. Nätverk tränades i $3 \cdot 10^5$ iterationer med kostnadsfunktionen enligt ekvation (3.3), där λ antog värdena 1, $5 \cdot 10^{-1}$, $1 \cdot 10^{-1}$, $5 \cdot 10^{-2}$, $1 \cdot 10^{-2}$, $5 \cdot 10^{-3}$ och $1 \cdot 10^{-3}$. Antalet γ-partiklar i indatamängden varierades mellan 1–3 och för varje λ tränades tio nätverk. Det relativa felet för energin och felet i $\cos\theta$ beräknades var för sig, och en t2-fördelning anpassades för att mäta storleken av dessa; detta efter att en analys av felen visat att de inte var normalfördelade, se appendix B.3. För varje värde på λ beräknades sedan ett medelvärde av de tio nätverkens felbredder. Andelen rekonstruerade partikelenergier i anpassningen beräknades också genom vetskapen att den normerade anpassningen vid perfekt rekonstruktion skulle integrera till ett.

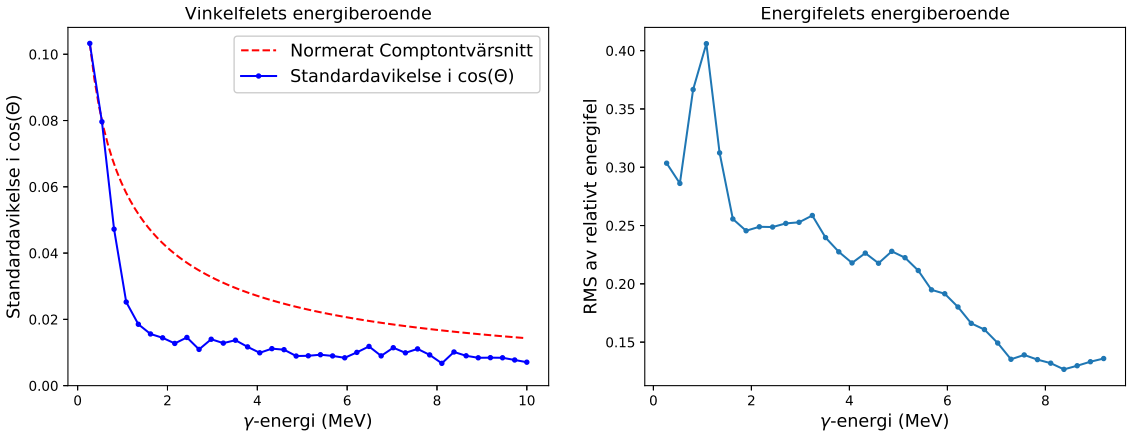
Figur 3.6 visar hur breddningarna, $s_{\cos\theta}$ och $s_{\Delta E/E}$, samt andelen lyckade energirekonstruktioner, varierar med λ. Notera att enbart $s_{\cos\theta}$ innehåller alla sju värden vilket är en följd av att energirekonstruktionen inte lyckades för de lägsta λ. Det är inte överraskande eftersom energidelen där har låg prioritet hos nätverket. I figur 3.6 framgår det att $s_{\cos\theta}$ ökar med λ. Även $s_{\Delta E/E}$ tenderar att öka med λ men något att notera är andelen lyckade energirekonstruktioner för dessa värden. En låg andel lyckade rekonstruktioner indikerar att $s_{\Delta E/E}$ kan ge ett missvisande värde. Att använda ett värde på $\lambda < 0{,}05$ kan anses omotiverat eftersom ungefär hälften av energirekonstruktionerna misslyckas och att det inte sker någon större förbättring av $s_{\cos\theta}$ jämfört med $\lambda = 0{,}05$.



Figur 3.6: *Bredden av felet i $\cos\theta$, $s_{\cos\theta}$, och bredden av relativa felet i energin, $s_{\Delta E/E}$, med avseende på viktningsparametern λ som multiplicerats med energidelen i kostnadsfunktionen. Även andelen partikelenergier som kan anses rekonstruerade visas. Notera att enbart $s_{\cos\theta}$ har med alla sju värden på λ. Att använda ett värde på $\lambda < 0{,}05$ kan anses omotiverat eftersom en betydande del av energirekonstruktionerna blir misslyckade samt att det inte sker någon större förbättring av $s_{\cos\theta}$ jämfört med $\lambda = 0{,}05$.*

3.4.5 Rekonstruktionens energiberoende

I figur 3.4 ses antydningar till att rekonstruktionen fungerar bättre i vissa energiintervall och därför studeras rekonstruktionens energiberoende för fallet med relativ kostnadsfunktion och varierande multiplicitet närmare detta avsnitt. För att få en mer renodlad analys så har två nätverk tränats, vilka endast försöker rekonstruera θ-vinkeln ($\lambda = 0$) respektive energin ($\lambda \to \infty$) detta eftersom kopplingen mellan energi och vinkeldelen i kostnadsfunktionen har studerats i 3.4.4. Nätverken som användes i analysen tränades i $5{\cdot}10^5$ iterationer.



(a) **(b)**

Figur 3.7: *(a) Vinkelrekonstruktionens energiberoende. Vinkelrekonstruktionen är betydligt sämre för energier lägre än* 2 MeV*, därefter är felet konstant, vilket troligen beror på detektorupplösningen. Även Comptontvärsnittet (normerat till första mätpunkten) visas i samma figur och den minskar signifikant i det intressanta intervallet. (b) Energirekonstruktionens energiberoende, där de partiklar för vilka nätverket har gissat mer än* 75% *fel har ignorerats, för att undgå "mattan" i figur 3.4.*

Figur 3.7 visar rekonstruktionens energiberoende och det är tydligt att nätverket presterar bättre för höga energier, både med avseende på vinkel och energi. I vinkelrekonstruktionen, 3.7a, ses en betydande försämring för energier under 2 MeV. Comptontvärsnittet som även det kan ses i figuren, ökar betydligt för låga energier och kan vara en förklaring till att rekonstruktionen blir sämre. Comptonspridning är hämmande för god vinkelrekonstruktion eftersom spridningen kan ta en betydande andel av energin från γ-partikeln till en annan kristall. För energier över 2 MeV ses inget energiberoende i rekonstruktionen och den begränsande faktorn är troligen detektorupplösningen.

Även energirekonstruktionen är bättre för höga energier, se figur 3.7b. I analysen har partiklar som nätverket gissat mer än 75% fel ignorerats för att undvika "mattan" som ses i figur 3.4. Likaså har enbart energier upp till 9 MeV studerats. Nätverket tränades på data upp till 10 MeV för att undvika övre delen av s-formen (effekt av träningsintervall, se 3.4.2.1). Toppen som ses beror troligen på att "mattan" kommer så pass nära det korrekta värdet att den inkluderas. Att metoden inte försämras för energier över 1,022 MeV tyder på att nätverket kan hantera parproduktion på någon nivå. För högre energier minskar även sannolikheten för Comptonspridning men den är fortfarande större än sannolikheten för fotoabsorption. Därför tyder resultatet på att för höga energier då andelen parproduktion till Comptonspridning ökar fås mer av en process som nätverket hanterar bättre

och därav förbättras nätverkets prestation. Att nätverket behandlar parbildnig bättre än Comptonspridning beror troligen på parbildnings tydliga signatur (enkel- och dubbelförlust, se avsnitt 2.1.7) och att stor andel av γ-partikelns energi kan deponeras i kristallen.

3.4.6 Konvergens av nätverk och variation av nätverksarkitektur

För optimering av nätverkets prestation finns flertalet hyperparametrar att variera, framförallt i nätverkets arkitektur i form av antal lager och noder per lager. För att jämföra olika nätverk behövs dock ett konsekvent villkor för att avgöra om ett stationärläge nåtts. Två varianter har testats: först genom att studera kostnadsfunktionens förändringstakt och sedan genom en *early stopping*-metod (beskriven i avsnitt 2.2.6). Dessa metoder verkar dock vara mindre lämpade för det aktuella problemet. Detta tillsammans med ett svep över olika nätverksarkitekturer beskrivs i appendix B.1 respektive B.2. Det är definitivt av intresse att optimera nätverksmodellerna och att hitta ett fungerande konvergensvillkor. Det är däremot inte det primära fokuset för det här arbetet, och att ha löst dessa problem tycks inte vara absolut nödvändigt för rapportens syfte, att påvisa huruvida rekonstruktion med ANN är möjlig.

3.4.7 Kostnadsfunktion för felminimering av Dopplerkorrigerad energi

Som observerats i avsnitt 3.4.4 beror energi- och vinkelrekonstruktionen på viktningen mellan termerna i kostnadsfunktionen. Vid experiment är det ofta den Dopplerkorrigerade energin (beskriven i avsnitt 2.1.8) som är relevant. Med detta som motivering användes Gaussisk felfortplantning av den Dopplerkorrigerade energin som utgångspunkt för en alternativ kostnadsfunktion, där viktningen mellan termerna är fysikaliskt motiverad:

$$(\Delta E')^2 = \left(\frac{1-\beta\cos\theta}{\sqrt{1-\beta^2}}\right)^2 (\Delta E)^2 + \left(\frac{\beta E}{\sqrt{1-\beta^2}}\right)^2 (\Delta\cos\theta)^2, \tag{3.4}$$

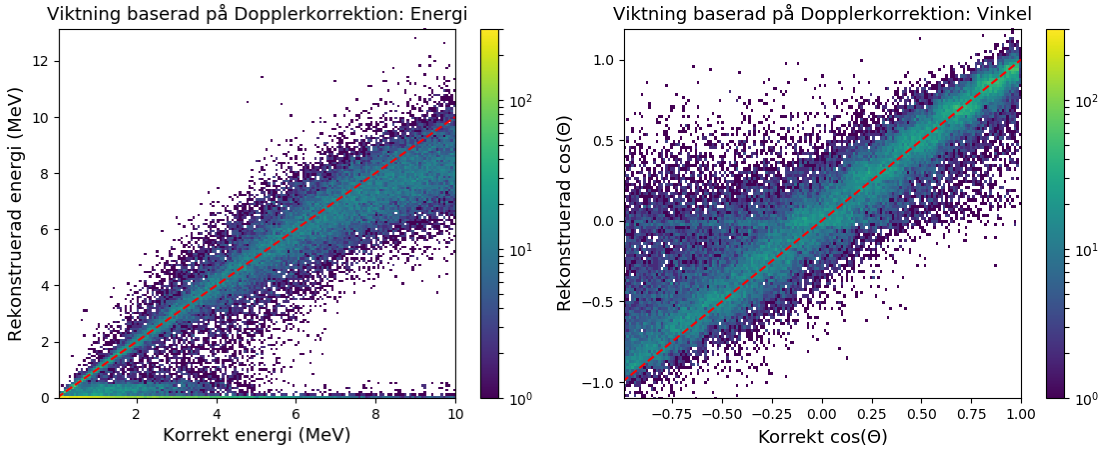
där E' är den Dopplerkorrigerade energin, $\beta = \frac{v}{c}$ är den normerade hastigheten och Δ representerar i detta fall mätfelet i respektive variabel. Denna feluppskattning kan användas som kostnadsfunktion och kallas hädanefter Absolut Doppler (AD). Samma typ av kostnadsfunktion kan göras relativ genom att studera $(\frac{\Delta E'}{E})^2$, och som i avsnitt 3.4.2 introduceras en E_{offset} i nämnaren för att kunna hantera flera partiklar. Kostnadsfunktionen, Relativ Doppler (RD), blir

$$C(y(w,b),\hat{y}) = \frac{1}{n}\sum_{j=1}^{n}\left[\sum_{i=1}^{n_\gamma}\left(\frac{\Delta E_{j,i}}{\hat{E}_{j,i}+E_{\text{offset}}}\right)^2 + \left(\frac{\beta}{1-\beta\widehat{\cos\theta_{j,i}}}\right)^2 (\Delta\cos\theta_{j,i})^2\right], \tag{3.5}$$

där $\Delta E_{j,i} = E_{j,i} - \hat{E}_{j,i}$, $\Delta\cos\theta_{j,i} = \cos\theta_{j,i} - \widehat{\cos\theta_{j,i}}$, n_γ är maximala antalet partiklar och n är antalet händelser som kostnadsfunktionen beräknas över. För att para ihop korrekt och rekonstruerad data konstruerades alla permutationer (se avsnitt 3.4.3) för både AD och RD. Kostnadsfunktionen blir nu beroende av parametern β, vilket betyder att rekonstruktionen specialiseras för en specifik hastighet på jonstrålen. I jämförelsen med befintliga algoritmer, i avsnitt 4.1, där ett experiment simuleras, varierar β mellan 0,6 och 0,7. Även om β varierar bör ett karakteristiskt β kunna användas under träningen som en approximation, eftersom det handlar om ett litet intervall.

Eftersom relativa vikten mellan energi- och vinkeltermen nu beror av vinkeln för partikeln,

kommer felen inte längre att vara symmetriska kring den korrekta rekonstruktionen, vilket ses i figur 3.8 där RD använts i en situation med 1–3 γ-partiklar. Av denna anledning görs den slutliga jämförelsen mellan RD, AD och de tidigare kostnadsfunktionerna i avsnitt 4.1. Spektrumet innehåller flera av de karakteristiska dragen från de tidigare relativa kostnadsfunktionerna, men fokus ligger framförallt på vinklarna i framåtriktningen medan vinklar bakåt rekonstrueras dåligt. Detta ses i ekvation (3.5), där ett givet fel i framåtriktningen viktas betydligt högre än samma fel i bakåtriktningen.



(a) **(b)**

Figur 3.8: *Rekonstruktion av 1–3 γ-partiklar med användning av kostnadsfunktionen enligt ekvation* (3.5)*. Inga större skillnader framgår för energirekonstruktionen i jämförelse med en konstant viktning mellan energi- och vinkeltermen. Däremot ses en betydlig skillnad i vinkelrekonstruktionen, där asymmetrin till följd av RD-kostnadsfunktionen framkommer tydligt.*

3.5 Jämförelse av olika faltningsmetoder

För att undersöka effekten av faltningslager för rekonstruktion från Crystal Ball, användes till en början inte regressionsnätverket för energi- och vinkelrekonstruktion (se avsnitt 3.4), utan ett klassificeringsnätverk för att enbart rekonstruera multipliciteten av γ. Det fanns två anledningar till detta. Den första var att effekterna av faltning ansågs vara som lättast att urskilja för stora multipliciteter, men detta gör att regressionsnätverket tar mycket lång tid att träna, se figur 3.5. Den andra anledningen var att resultatet från ett klassificeringsproblem är mycket smidigt att utvärdera genom att bestämma pricksäkerheten[2]. Eftersom kunskaper om γ:s energispridning måste tas i beaktning vid utredandet av multipliciteten, var förhoppningen att en faltningsstrategi väl lämpad för multiplicitetsbedömning, också skulle ge goda resultat för regressionsproblemet.

Det väsentliga med detta avsnitt är alltså inte själva klassificeringsnätverket för multiplicitet, och därmed läggs inte fokus på att beskriva detaljer kring nätverket eller träningsprocessen. För information om bland annat klassificeringsnätverkets arkitektur och kostnadsfunktion samt hur pricksäkerheten utvärderades, hänvisas istället till appendix A. Där beskrivs också hur de olika nätverksstrukturerna kunde jämföras trots avsaknaden av ett konsekvent konvergensvillkor, se avsnitt 3.4.6.

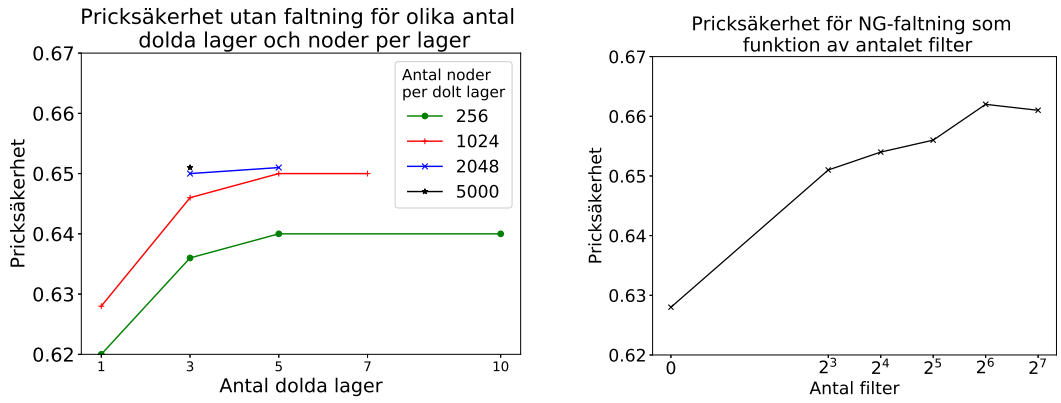
[2]Pricksäkerheten definieras som sannolikheten att nätverket ge rätt multiplicitet för en godtycklig händelse i evalueringsdatamängden.

Fyra olika tillvägagångssätt med faltning utreddes på data med upp till sex stycken γ och energin för varje partikel varierade mellan 0,1 och 10 MeV. Dessa presenteras i tabell 3.2. Metoderna kommer att beskrivas kort nedan, men för en utförligare beskrivning, bland annat om hur detektorns sfäriska utformning hanterades, hänvisas till appendix A.

Tabell 3.2: *Presentation av de faltningsmetoder som undersökts.*

Förkortning	**Metodbeskrivning**
NG	Ett faltningslager med närmsta grannar
NG-NG	Två stycken NG-lager i följd
NG-pool-NG-pool	Ett NG-lager, följt av ett pooling-lager, följt av ett NG-lager och till sist ett pooling-lager.
NG-NNG	Ett NG-lager följt av ett faltningslager med näst närmsta grannar.

Ett NG-lager innebär ett faltningslager där varje filter translateras över varje kristall och täcker den och de närmsta grannarna. I figur 3.9b visas att pricksäkerheten ökar kraftigt när ett NG-lager placeras före ett 1024 noder brett, fullt kopplat nätverk. Av figur 3.9a framgår att motsvarande ökning utan faltning knappt är möjlig, och skulle i så fall kräva flera storleksordningar fler parametrar. Antal filter för jämförelsen mellan de olika faltningsmetoderna, valdes till 32 för NG-lagret, för att inte två NG-lager i följd (NG-NG) skulle riskera att ta lång tid att träna.



(a) **(b)**

Figur 3.9: *(a) Pricksäkerhet för multiplicitet för olika antal dolda lager och noder per lager utan faltning. Det bästa resultatet åstadkoms med 5 dolda lager med 2048 noder vardera. Anledningen till varför så få körningar gjordes med 2048 och 5000 noder var att nätverket redan ansågs väloptimerat utifrån trenderna i figuren. I avsnitt 3.5.1 presenteras argument för att felmarginalen för varje mätpunkt är ± 0,0015. (b) Hur antalet filter i en NG-faltning påverkar pricksäkerheten för multiplicitetsbestämningen, där 0 filter motsvarar ett nätverk utan faltningslager. Det fullt kopplade nätverket bestod för alla mätningar av ett dolt lager med 1024 noder. En tydlig ökning av pricksäkerheten observeras när antalet filter ökas, förutom från 64 till 128 filter då pricksäkerheten verkar ha konvergerat. NG-lagret följdes av ett fullt kopplat lager med 1024 noder. I avsnitt 3.5.1 presenteras argument för att felmarginalen för varje mätpunkt är ± 0,0015.*

Pooling-filtren som användes i NG-pool-NG-pool var av samma storlek som NG-filtren, men tilläts bara överlappa med en kristall. På grund av överlappet användes inte max-pooling eftersom flera filter då skulle riskera att få samma värde, utan medelvärdes-pooling bedömdes vara mer passande. Anledningen till varför filtren tilläts att överlappa var att bevara symmetrin runt varje nod, eftersom det gjorde implementeringen av metoden lättare. Det första NG-lagret hade 32 filter och det andra 64. NG-NNG-strategin gick ut på att filter som täckte både de närmsta grannarna och grannarnas grannar, translaterades över resultatet från NG-lagret med mycket litet överlapp. Denna metod utvecklades för att efterlikna effekterna av NG-pool-NG-pool-metoden, men att den skulle vara mindre beräkningstung. Likt NG-pool-NG-pool-metoden sattes antal filter i första lagret och andra lagret till 32 respektive 64.

I figur 3.9a jämförs pricksäkerheten för olika kombinationer av lager och noder för ett nätverk utan faltning. När det utifrån trenderna verkade som att ett nästintill optimerat nätverk hittats, avbröts jämförelsen. Det är anledningen till varför så få mätningar gjordes för stora nodantal.

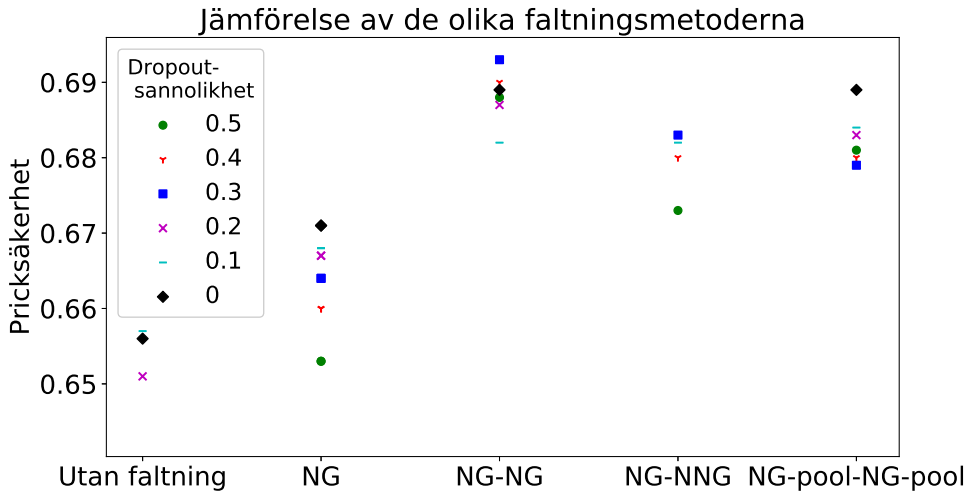
Resultatet från jämförelsen mellan de olika metoderna för faltning presenteras i figur 3.10. De jämförs också mot det mest framgångsrika nätverket utan något faltningslager. Efter faltningslagren följde alltid två stycken fullt kopplade lager på 2048 noder, vilket var ett relativt godtyckligt val, det viktigaste var att ge så lika förutsättningar som möjligt. För att se till att alla nätverk var optimerade tillämpades olika dropout-sannolikheter[3], se avsnitt 2.2.6, på de fullt kopplade lagren. Resultaten visar att alla metoderna presterar bättre än nätverket utan faltning, men att NG-NG, NG-NNG och NG-pool-NG-pool sticker ut positivt. Det bästa resultatet erhålls med NG-NG-nätverket med en dropout-sannolikhet på 0,3. En observation är att pricksäkerheten inte är konstant avtagande ju större avvikelsen är från den optimala dropout-sannolikheten är. För en mer rättvis jämförelse rekommenderas att hålla antalet filterparametrar för metoderna konstant. För träningstiderna som krävdes för ett konvergerat nätverk med de olika metoderna, se appendix A.

Undersökningen av olika faltningstillvägagångssätt för att förutsäga multiplicitet, visar att faltning kan ha betydande positiv inverkan på hur ett neuralt nätverk tolkar informationen från Crystal Ball och borde därmed också kunna ha positiv inverkan på liknande detektorer. För att förbättra pricksäkerheten ytterligare skulle olika kombinationer av noder och lager kunna undersökas efter faltningsmetoderna. Att just 2 dolda lager och 2048 noder per lager användes var som nämns relativt godtyckligt. Ytterligare något som skulle vara intressant att undersöka är hur pricksäkerheten ändras när åtskilliga NG-lager placeras efter varandra.

3.5.1 Feluppskattning

För att uppskatta ett konfidensintervall för pricksäkerheter undersöktes endast ett av fallen, det med NG-NG-faltning med en dropout-sannolikhet på 0,3. Fem träningsprocesser gjordes som grund för undersökningen och resultaten blev: 0,6907, 0,6910, 0,6937, 0,6907 och 0,6918. Dessa värden ger ett 95-procentigt konfidensintervall för pricksäkerheten på $0{,}6915 \pm 0{,}0015$. Denna felmarginal antas vara snarlik för övriga mätpunkter eftersom grundarkitekturen och uppgiften för nätverken är mycket lika.

[3]Notera att det inte är sannolikheten att den nod ska behållas som syftas på, utan just sannolikheten för att en viss nod ska blockeras.



Figur 3.10: *Pricksäkerheten för olika faltningsmetoder som funktion av olika dropout-sannolikheter. För alla nätverk med faltning användes två fullt kopplade lager med 2048 noder i vardera. För nätverket utan faltning valdes det bästa nätverket från svepet över antal lager och noder, det vill säga 5 dolda lager med 2048 noder. Det bästa resultatet åstadkoms alltså med NG-NG-metoden med en dropout-sannolikhet på 0,3. Detta talar emot att resultatet alltid skulle försämras längre bort från den optimala dropout-sannolikheten. I avsnitt 3.5.1 presenteras argument för att felmarginalen för varje mätpunkt är ± 0,0015.*

3.5.2 Faltning för energi- och vinkelrekonstruktion

Efter att effekterna av faltning undersökts för rekonstruktion av multipliciteten, gjordes en mindre undersökning av faltning tillämpat på regressionsnätverket beskrivet i avsnitt 3.4. I denna undersökning användes NG-NG-faltning eftersom denna faltningsstrategi gav bäst resultat för multiplicitetsrekonstruktionen. Efter de två faltningslagren följde fullt kopplade lager där ingen dropout tillämpades. Två kombinationer av lager och noder per lager användes: 2 lager med 2048 noder (den som användes för faltningsjämförelsen) och 8 lager med 256 noder (användes för att efterlikna 10 lager med 256 noder per lager som användes i avsnitt 3.4). Kostnadsfunktionen som användes för undersökningen var RD, se ekvation (3.5).

För att evaluera konsekvenserna av faltning tillämpat på nätverket för energi- och vinkelrekonstruktion, studerades de rekonstruerade energispektrumen som de två nätverken gav upphov till när de utsätts för realistisk data, i enlighet med avsnitt 4.1. Resultatet var dessvärre mycket sämre jämfört med när ingen faltning använts, framförallt i termer av signal-till-bakgrund-förhållandet.

Även om tillämpning av faltning gav sämre resultat för experimentliknande data, var den lika bra eller bättre på att minska värdet av RD-kostnadsfunktionen för evalueringsdatamängden. En förklaring till detta kan vara att filtren lärt sig mönster specifika för γ emitterade från vila, som inte är lika vanligt förekommande i experimentliknande data, där partiklarna emitteras i strålens inertialsystem. Det skulle även kunna vara så att problemen endast uppstår på grund av RD-kostnadsfunktionen och att faltningsmetoder hade kunnat uppvisa bättre resultat om en annan kostnadsfunktion hade använts.

4

Jämförelse med nuvarande metoder och applicering på annan detektor

För att tolka hur väl neurala nätverk rekonstruerar detektordata behöver resultaten jämföras med nuvarande metoder. Eftersom arbetet endast behandlat Crystal Ball är det även lämpligt att testa nätverkens rekonstruktionsförmåga för en annan detektor och därigenom visa på dess allmängiltighet. Detta kapitel jämför nätverk med olika kostnadsfunktioner med en så kallad addback-rutin som i nuläget används för att rekonstruera partikelenergier för Crystal Ball. Därefter presenteras ett nätverks rekonstruktion av data från detektorn DALI2 [8], som likt Crystal Ball är en γ-kalorimeter med NaI-kristaller men skiljer sig i sin geometri.

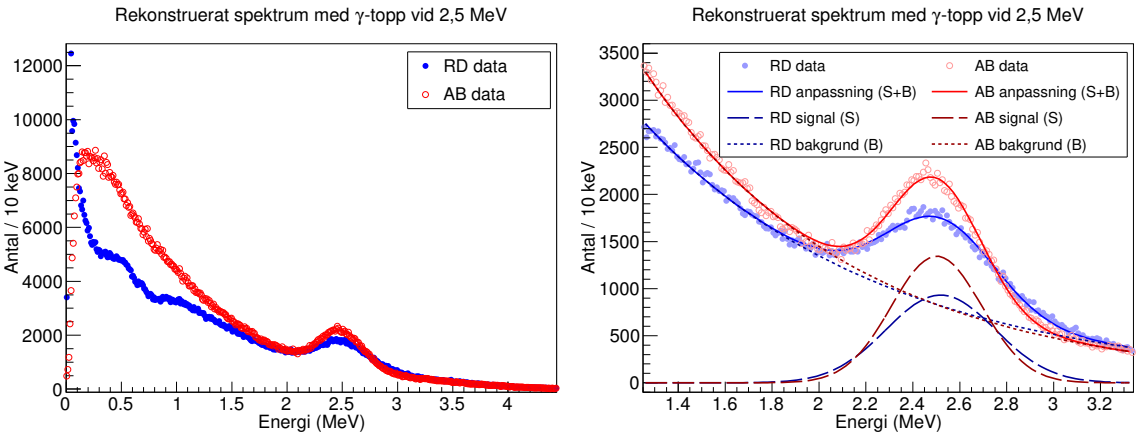
4.1 Jämförelse med addback-rutin

Addback-rutinens princip är relativt enkel. Den bygger på att för varje händelse lokalisera den kristall med högst energi och sedan anta att denna energi härrör från en primärdeponering. Energivärden i omgivande kristaller antas innehålla spridda sekundära partiklar. Dessa energier tas sedan bort och processen upprepas tills alla energivärden behandlats. Positionen för den primära kristallens mittpunkt används som vinkeldata för varje partikel. För en mer utförlig beskrivning av hur addback-rutinen tillämpas, se Lindberg [5].

I praktiken är det ofta av intresse att kunna skilja på detektordata som kommer från primära reaktioner i strålmålet och data från bakgrundsstrålning. Detta kan göras genom att försöka identifiera toppar i det uppmätta energispektrumet som kommer från primära reaktioner.

I figur 4.1a jämförs ett exempel på rekonstruerade spektrum från ett nätverk tränat med den Relativa Doppler-kostnadsfunktionen (RD) presenterad i avsnitt 3.4.7 med motsvarande rekonstruktion från addback-rutinen (AB). Figur 4.1b visar de kurvor som anpassats till de båda rekonstruktionerna kring en topp. Signalen respektive bakgrunden är i båda fallen anpassade med en normalfördelningskurva respektive en exponentiellt avtagande kurva i ett intervall kring centrallinjen. Den data som rekonstruerats kommer från 10^6 simulerade händelser varav 10% innehåller en primär γ, isotropt emitterad med 2,5 MeV i strålens inertialsystem. Strålens normerade hastighet β, är i intervallet 0,6 till 0,7. Bakgrunden utgörs av två γ som individuellt emitteras med 50% sannolikhet per händelse, den ena med en energi i intervallet 0,05 till 1 MeV i labbsystemet och den andra med en energi i intervallet 0,05 till 2 MeV i labbsystemet. Det betyder att det kan finnas maximalt 3 partiklar i varje händelse. Histogramdata är Dopplerkorrigerad med hjälp av rekonstru-

erade vinkelvärden från varje metod så att den presenteras i strålens inertialsystem. Det framgår i figur 4.1a att AB rekonstruerar en tydligare topp vid $\sim 2{,}5\,\text{MeV}$ samt att RD förvränger bakgrunden vid lägre energier. Notera hur antalet för RD växer för energier nära noll vilket inte stämmer överens med simuleringen, AB visar på motsatt beteende. Anpassningarna i figur 4.1b tydliggör de två topparnas skillnader.



(a) **(b)**

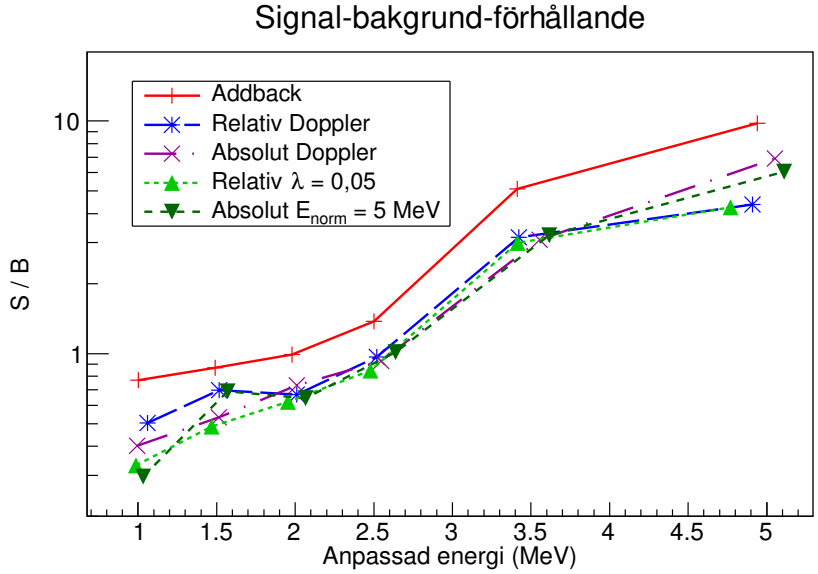
Figur 4.1: *I (a) jämförs hur ett nätverk tränat med den relativa Dopplerkostnadsfunktionen (RD) rekonstruerar simulerad data med motsvarande rekonstruktion utförd med en addback-rutin (AB). Kurvanpassningar för signal respektive bakgrund för de två histogrammen visas i (b). Det är tydligt att AB lyckas rekonstruera en mer framstående topp än vad RD gör.*

Bakgrunden är i allmänhet stor i förhållande till den eftersökta signaltoppen och för att statistiskt verifiera att en signal hittats studeras förhållandet mellan signal och bakgrund. Signal-bakgrund-kvoten S/B kan beräknas genom att integrera fördelningarna av bakgrund och signal, $f_B(E)$ och $f_S(E)$, över ett två standardavvikelser, $2\sigma_S$, brett intervall centrerat kring signalens centrallinje μ_S:

$$S/B = \frac{\int_{\mu_S-\sigma_S}^{\mu_S+\sigma_S} f_S(E)dE}{\int_{\mu_S-\sigma_S}^{\mu_S+\sigma_S} f_B(E)dE}. \tag{4.1}$$

Beräknade värden på S/B för olika rekonstruktionsmetoder är utritade i figur 4.2 som funktioner av anpassade signalpositioner. Beräkningarna är gjorda utifrån samma typ av anpassningar som ses i figur 4.1b för addback (AB) samt fyra nätverk tränade med olika kostnadsfunktioner: Relativ Doppler (RD), Absolut Doppler (AD), relativ med viktningsfaktor $\lambda = 0,05$ och absolut med normeringsfaktor $E_{\text{norm}} = 5\,\text{MeV}$. För fler exempel på anpassningar, se appendix C. Rekonstruktionen är gjord utifrån 6 omgångar simulerad data med γ-toppar vid $E = 1\,\text{MeV}$, $1{,}5\,\text{MeV}$, $2\,\text{MeV}$, $2{,}5\,\text{MeV}$, $3{,}5\,\text{MeV}$ och $5\,\text{MeV}$. Det framgår att AB har högre S/B-värde än alla nätverk vid samtliga energier vilket är förväntat med tanke på utseendet hos exempelvis anpassningarna i figur 4.1b. De olika kurvorna visar på viss kovariation vilket tyder på att anpassningarna är konsistenta. Skillnader mellan de olika nätverken är marginella förutom vid $5\,\text{MeV}$ där de absoluta varianterna presterar bäst.

Observera att figur 4.2 är framtagen från endast en typ av simulering med en relativt enkel bakgrund. Resultatet skulle i ett annat fall kunna se annorlunda ut. Det är dock från

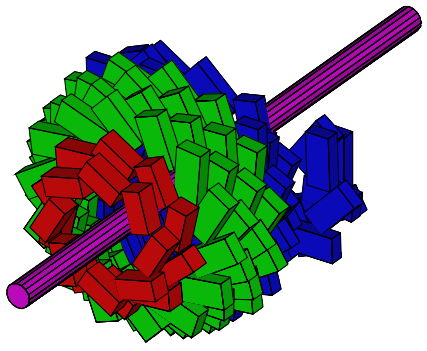


Figur 4.2: *Jämförelse av förhållanden mellan signal och bakgrund för olika rekonstruktionsmetoder vid olika primära γ-energier. Addback-rutinen tycks prestera bäst vid samtliga energier.*

resultaten ovan tydligt att AB i detta fall lyckas bättre med att rekonstruera det simulerade energispektrumet. Även om RD lyckas identifiera toppen vid 2,5 MeV i figur 4.1a är det oroväckande att bakgrunden vid låga energier förvrängs på ett sådant sätt att det skulle kunna tolkas som fler förekommande toppar i spektrumet. Detta är ett problem som måste åtgärdas för att metoden ska kunna bli pålitlig. Anledningen till problemet är troligtvis nätverkets sämre förmåga att rekonstruera partikelegenskaper vid låga energier vilket kan ses i figurer 3.7b och 3.7a. Även RDs rekonstruktion av träningsdata som kan ses i figurer 3.8a och 3.8b tyder på sämre prestanda vid låga energier och vinklar i bakåtriktningen.

4.2 Rekonstruktion för detektorn DALI2

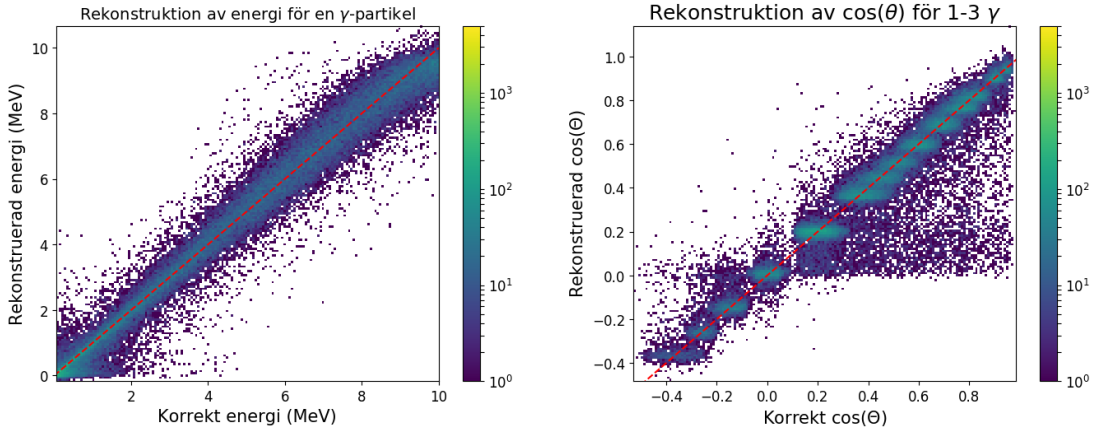
För att visa att metoden med neurala nätverk är generell testades även nätverk på detektorn DALI2 som kan ses i figur 4.3. Likt Crystal Ball är DALI2 uppbyggd av scintillerande kristaller (NaI(Tl)) placerade runt strålmålet men med en mer oregelbunden geometri. I kapitel 3 beskrivs nätverksmodellen och tillvägagångssättet för applicering av ANN på Crystal Ball. Den enda skillnaden för applicering på DALI2 är antalet innoder som anpassats till DALI2s kristallantal, 140, samt att ny träningsdata simulerats med DALI2 som vald detektor.



Figur 4.3: *Visualisering av detektorn DALI2 och strålrörets position. Kristallernas olika nyanser betyder att de är av olika typ. I figuren är strålen riktad framåt.*

Figur 4.4 visar resultatet från rekonstruktion av energi och $\cos\theta$. Här har den absoluta kostnadsfunktionen enligt ekvation (3.1) använts med

$E_{\text{norm}} = 5\,\text{MeV}$. Notera att nätverket inte rekonstruerar vinklar där $\cos\theta < -0,4$, vilket är vinklar där partikeln emitterats bakåt sett till strålmålet. Detta kan förklaras av att DALI2 inte har några kristaller för dessa vinklar. Detektorns kristaller har oregelbunden placering och det uppkommer även flera hål för viktig apparatur, återigen se figur 4.3. I övrigt kan vi i likhet med tidigare resultat se att rekonstruktion av både energi och emissionsvinkel är principiellt möjlig. Detta indikerar att en generalisering till andra liknande detektorer är genomförbart.



(a) **(b)**

Figur 4.4: *Rekonstruktion av energi och* $\cos\theta$ *för detektorn DALI2 visas i (a) respektive (b). I vinkelrekonstruktionen tenderar vinklar med värde på* $\cos\theta < -0,4$ *inte rekonstrueras. Detta kan förklaras av detektorns utformning och geometri, vilket kan ses i figur 4.3.*

5
Diskussion och sammanfattning

Syftet med denna rapport var att redovisa för en undersökning av om det var möjligt att rekonstruera partikelhändelser från detektordata i ett kärnfysiksammanhang med hjälp av neurala nätverk. För detta syfte utvecklades kostnadsfunktioner och efter träning med dessa kunde nätverken rekonstruera energin och de två emissionsvinklarna som partikeln strålades ut med från strålmålet. Rekonstruktionen kan göras både för en och flera partiklar. Vid varierande antal partiklar och en relativ kostnadsfunktion tenderar nätverket att återge ett mindre antal partiklar än det givna, något som fortfarande är ett olöst problem.

Det har även visats att metoden inte är detektorspecifik eftersom den med framgång kan appliceras på två olika detektormodeller, Crystal Ball och DALI2. Därmed finns det goda förutsättningar för att metoden ska kunna generaliseras även till CALIFA. Med nätverksmodellens nuvarande utformning kan signaler i de tänkta experimenten ses, men metoden presterar inte bättre än den redan befintliga rekonstruktionsalgoritmen för Crystal ball. Nätverket har å andra sidan många hyperparametrar som inte har studerats till fullo, varför det troligen finns förbättringspotential.

5.1 Övergripande diskussion

Som ett resultat från avsnitt 3.4.2 verkar införandet av relativ bestraffning i kostnadsfunktionen ha stor effekt på energirekonstruktionen. Vidare visade det sig att inkludera faltningslager inte gav någon positiv inverkan vid jämförelse med addback-rutinen, trots att det gav det vid multiplicitetsrekonstruktion. Hur de metodval som gjorts under projektets gång påverkat nätverket i detalj är svårt att avgöra då träningen sker som i en svart låda. Detta problem är något som präglar all användning av ANN och diskuteras inom datavetenskapen, se exempelvis [22].

När nätverket har två termer att minimera i kostnadsfunktionen vill vi att nätverket gör detta så att den bästa sammanlagda händelserekonstruktionen erhålls. Genom utvecklingen av AD och RD i avsnitt 3.4.7 togs inspiration från avsnitt 2.1.8 om relativistiska effekter för att försöka optimera viktningen mellan termerna. I figur 4.2 sågs en antydan till att en fysikaliskt baserad kostnadsfunktion fungerar bäst ses för de högre energierna. Att ge en rekommendation om en kostnadsfunktion är däremot svårt, eftersom jämförelsen med addback-rutinen visar hur deras kvalité varierar för olika energier.

Slutligen kan den iterativa process som använts som arbetsmetod anses ha fungerat väl för projektets syfte. Genom kontinuerlig utvärdering av delresultat har arbetet kunnat fortskrida och analys av enskilda rekonstruktioner har gjort att särskiljande egenskaper hos kostnadsfunktionerna upptäckts. Att till sist jämföra olika nätverk med den befintliga addback-rutinen gav ett bra mått på hur väl nätverken egentligen utförde händelserekon-

struktion, speciellt eftersom rekonstruktionen av parametrarna utan ett sammanhang inte explicit visade detta.

5.2 Slutsatser

Denna rapport har presenterat ett tillvägagångssätt för att använda neurala nätverk vid rekonstruktion av händelser i en partikeldetektor, det vill säga, att återge inkommande partiklars relevanta egenskaper. Vi kan dra slutsatsen att neurala nätverk fungerar för denna typ av rekonstruktion, men att den ännu inte är bättre än den befintliga addback-rutinen. Vidare kan vi sluta oss till att hur väl nätverket presterar vid rekonstruktion beror på dess många hyperparametrar och att det krävs vidare studier för att optimera dessa.

5.3 Fortsatt arbete

För att den utvecklade metoden ska nå sin fulla potential behöver det mångdimensionella parameterrum av hyperparametrar studeras vidare. Exempel på dessa är:

- Nätverkets arkitektur och aktiveringsfunktioner,
- Gruppstorleken för träningsdata,
- Parametrar i kostnadsfunktionen så som E_{offset} och λ.

Ytterligare något att studera vidare är den ”matta” som visas i figur 3.4. Utförligare tester med kontrollnod kan vara intressant. Det primära är dock att få förståelse för vid vilka tillfällen fenomenet uppstår eftersom ”mattan” inte alltid förekommer.

I och med att faltning är ett kraftfullt verktyg vid bildanalys och att de faltningsmetoder som tagits fram i detta arbete visat sig användbara vid multiplicitetsrekonstruktion, vore det intressant att fortsätta utreda hur faltning kan användas för energi- och vinkelrekonstruktion. Detta skulle till exempel kunna göras genom att träna nätverket med mer experimenttrogen data.

För att rekonstruera händelser i detektorer kan eventuellt en trädstruktur av nätverk användas, där multipliciteten hos partiklarna först bestäms för att sedan rekonstruera deras parametrar med givet antal partiklar. I denna rapport kan vi se att ett fixt antal partiklar verkar ge en bättre energirekonstruktion och en trädstruktur skulle innebära att ett nätverk bara behöver tränas för ett specifikt ändamål istället för flera.

En annan strategi som skulle kunna undersökas är en kombination av fysikaliskt grundade metoder och ANN. Ett sätt är att använda ett ANN för att rekonstruera i vilka kristaller primärträffar skett, och sedan använda denna information i en fysikaliskt grundad metod. Ytterligare ett sådant exempel är att, tillsammans med informationen om den deponerade energin i varje kristall, ge resultatet från addback-rutinen som indata till ett ANN. Eventuellt skulle nätverket då kunna prestera bättre än addback-rutinen genom att korrigera dess resultat.

Till sist behöver ett konvergensvillkor utarbetas för att på ett rättvist sätt kunna jämföra olika nätverk. Detta konvergensvillkor behöver ta i beaktande olika arkitekturers tröghet under träningsprocessen.

Referenser

[1] R3B-kollaborationen. *Technical Report for the Design, Construction and Commissioning of The CALIFA Barrel: The R3B CALorimeter for In Flight detection of γ rays and high energy charged pArticles*. Tekn. rapport. FAIR, 2011. URL: https://fair-center.eu/fileadmin/fair/publications_exp/CALIFA_BARREL_TDR_web.pdf.

[2] FAIR. *FAIR - An International Facility for Antiproton and Ion Research*. URL: http://www.fair-center.eu/. (Hämtad 2018-04-24).

[3] R3B-kollaborationen. *A next generation experimental setup for studies of Reactions with Relativistic Radioactive Beams*. URL: https://www.gsi.de/work/forschung/nustarenna/nustarenna_divisions/kernreaktionen/activities/r3b.htm. (Hämtad 2018-04-24).

[4] D. Cortina-Gil m. fl. "CALIFA, a Dedicated Calorimeter for the R3B/FAIR". I: *Nuclear Data Sheets* 120 (2014), s. 99 –101. ISSN: 0090-3752. DOI: 10.1016/j.nds.2014.07.017.

[5] S. Lindberg. "Optimised Use of Detector Systems for Relativistic Radioactive Beams". Examensarb. Chalmers tekniska högskola, 2013.

[6] M. A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.

[7] P. Adrich. *The Crystall Ball γ-Detector Setup and Calibration*. Tekn. rapport. GSI, 2003. URL: http://www-land.gsi.de/a_new_land/__public/documentation/detectors/crystal_ball/setup_and_calibration/cb_docu.pdf.

[8] S. Takeuchi m. fl. "DALI2: A NaI(Tl) detector array for measurements of γ rays from fast nuclei". I: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 763 (2014), s. 596 –603. ISSN: 0168-9002. DOI: 10.1016/j.nima.2014.06.087.

[9] W. R. Leo. *Techniques for nuclear and particle physics experiments: a how-to approach*. English. New York;Berlin; Springer, 1987. ISBN: 3540-1738.

[10] O. Klein och Y. Nishina. "Über die Streuung von Strahlung durch freie Elektronen nach der neuen relativistischen Quantendynamik von Dirac". I: *Zeitschrift für Physik* 52.11 (1929), s. 853–868. ISSN: 0044-3328. DOI: 10.1007/BF01366453.

[11] H. A. Bethe och J. Ashkin. "Passage of Radiation through Matter". I: *Experimental Nuclear Physics* 1 (1953), s. 325–331.

[12] S. Tazzari och M. Ferrario. "Trends in high energy particle accelerators". I: *Reports on Progress in Physics* 66.6 (2003), s. 1045.

[13] S. Agostinelli m. fl. "Geant4—a simulation toolkit". I: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 506.3 (2003), s. 250–303. ISSN: 0168-9002. DOI: 10.1016/S0168-9002(03)01368-8.

[14] CERN. *GEANT4*. 2018. URL: http://geant4.web.cern.ch/. (Hämtad 2018-04-22).

[15] L.L. Carter och E.D. Cashwell. *Particle-transport simulation with the Monte Carlo method*. Tekn. rapport. Los Alamos Scientific Lab., 1975. DOI: 10.2172/4167844.

[16] P. Doornenbal. "In-beam gamma-ray spectroscopy at the RIBF". I: *Progress of Theoretical and Experimental Physics* 2012.1 (2012), s. 03C004. DOI: 10.1093/ptep/pts076.

[17] W. Rindler. *Relativity Special General and Cosmological*. Oxford University Press, 2006.

[18] L. Atar m. fl. "Quasifree (p, $2p$) Reactions on Oxygen Isotopes: Observation of Isospin Independence of the Reduced Single-Particle Strength". I: *Phys. Rev. Lett.* 120 (5 2018), s. 052501. DOI: 10.1103/PhysRevLett.120.052501.

[19] T. M. Mitchell. *Machine Learning*. McGraw-Hill Science/Engineering/Math, 1997. ISBN: 0070428077.

[20] I. Goodfellow, Y. Bengio och A. Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[21] D. P. Kingma och J. Ba. "Adam: A Method for Stochastic Optimization". I: *CoRR* abs/1412.6980 (2014). arXiv: 1412.6980.

[22] M. Hutson. "Has artificial intelligence become alchemy?" I: *Science (New York, N.Y.)* 360.6388 (2018), s. 478. ISSN: 1095-9203. DOI: 10.1126/science.360.6388.478.

[23] H. T. Johansson. *ggland*. URL: http://fy.chalmers.se/~f96hajo/ggland/. (Hämtad 2018-04-22).

[24] CERN. *About ROOT*. URL: https://root.cern.ch/about-root. (Hämtad 2018-04-18).

[25] CERN. *ROOT files*. URL: https://root.cern.ch/root-files. (Hämtad 2018-04-18).

[26] HPC2N (bbrydsoe). *Kebnekaise*. 2016. URL: https://www.hpc2n.umu.se/resources/hardware/kebnekaise. (Hämtad 2018-04-20).

[27] Tensorflow. *Tensorflow - Programmers guide*. URL: https://www.tensorflow.org/programmers_guide/. (Hämtad 2018-04-17).

[28] Tensorflow. *Tensorflow - Tutorials*. URL: https://www.tensorflow.org/tutorials/. (Hämtad 2018-04-18).

[29] Tomruen. *Hexagonal lattice*. 2015. URL: https://commons.wikimedia.org/wiki/File:Equilateral_Triangle_Lattice_rotated.svg.

A

Detaljer kring faltningsmetoder

Detta appendix behandlar metoder och träningstider, relaterat till faltning och nätverket för rekonstruktion av multiplicitet.

A.1 Metoder

I detta avsnitt beskrivs hur de olika tillvägagångssätten för faltning implementerades. Även information kring klassificeringsnätverket för multiplicitetrekonstruktion och hur dess resultat utvärderades, presenteras.

A.1.1 Klassificeringsnätverk för multiplicitetsbedömning

Nätverket för multiplicitetsrekonstruktion tog in 162 kristallenergivärden och resulterade i $m + 2$ utnoder, där m är det maxiala antalet partiklar som undersöks. De två extra utnoderna representerar händelser utan deponerad energi och händelser med högre multiplicitet än den maximala multipiciten som vill undersökas. Detta eftersom nätverket tränades på ett större interval än vad som evaluerades på. De resultat som presenteras i rapporten från multiplicitetnätverket har evaluerats på händelser med en multiplicitet på 1 till 6, och träningen har skett på händelser med en multiplicitet på 0 till 7. Anledningen till denna buffert runt det undersökta intervallet var att risken att nätverket rekonstruerar rätt, bara för att inget högre eller lägre alternativ finns att tillgå, ville reduceras.

Den data som användes innehöll precis som den för programmet för energirekonstruktion, endast händelser där minst 90 % av den totala energin hos varje partikel deponerats. Energin hos γ-partiklarna varierades likformigt från intervallet 0,1 MeV till 10 MeV. För att representera rätt svar användes one-hot-representation. Den data som tränades på var helt separat från den som evaluerades på, och totalt sett användes ungefär 10^6 händelser per antal partikel. Träningsdatamängden bestod av 90 % av den totala datamängden och resterande del användes som evalueringsdata.

Nätverksstrukturen, förutom in- och utlagret, bestod av varierande antal lager, och om faltningslager användes placerades dessa alltid före de fullt kopplade lagren. I de fall där dropout tillämpades, var det bara på de fullt kopplade lagren. På utlagret tillämpades softmax och sedan användes cross-entropy-kostnadsfunktionen på dessa och one-hot-vektorn, se avsnitt 2.2.2 för mer information kring upplägget för kostnadsfunktionen.

Pricksäkerheten för varje enskild händelse i gruppen utvärderades genom att jämföra indexet för det maximala värdet bland utnoderna med motsvarande index för one-hot-vektorn. För att avgöra den slutgiltiga pricksäkerhet som en nätverksarkitektur kunde uppnå tränades nätverket $2 \cdot 10^6$ träningsiterationer. Vart hundrade träningssteg sparades

medelvärdet av pricksäkerheterna för en grupp på 300 händelser från evalueringsdatamängden. Sedan delades de sparade pricksäkerheterna upp i intervall där varje intervall motsvarade 50 000 iterationer, och den slutgiltiga pricksäkerheten för nätverket togs fram genom att beräkna medelvärdet för varje sådant intervall, och ta det maximala av dessa medelvärden. Med denna metod för att evaluera pricksäkerheten, gör det inget om nätverkets prestation skulle försämras på grund av överträning, det är ändå den maximala pricksäkerheten som undersöks. Anledningen till varför totalt $2 \cdot 10^6$ träningsiterationer valdes, grundade sig i att de flesta undersökta nätverk tycktes konvergera för betydligt färre iterationer än så. Eftersom inget konsekvent konvergensvillkor utvecklats så valdes detta stora antal iterationer, för att garantera att nätverket konvergerat. En graf över hur pricksäkerheten förändrades som funktion av antalet träningssteg studerades dock alltid manuellt för att försäkra att nätverket verkade vara konvergerat.

Ytterligare information kring multiplicitetsnätverket var att optimeringsalgoritmen som användes var Adam (se avsnitt 2.2.5), att aktiveringsfunktionen för de dolda lagren var ReLu (se avsnitt 2.2.1) och att gruppstorleken för träningsdatamängden var 100.

A.1.2 Ett faltningslager med närmsta grannar (NG)

Ett faltningslager för närmsta grannar implementerades genom att vektorn med kristallenergierna, först konverterades så att varje kristallenergi placerades i grupper med sina grannar. Alltså konverterades den 162 långa vektorn med kristallenergier till en $162 \cdot 7$ lång vektor, då det maximala antalet grannar för en kristall var sex. Denna konvertering illustreras i figur A.1 och implementerades i TensorFlow med hjälp av matrismultiplikation, eftersom gpu:er är optimerade för denna operation. Matrisen som den 162 långa vektorn multiplicerades med bestod nästan bara av nollor, och hastigheten på operationen förbättrades avsevärt genom att spara denna matris i TensorFlows format för glesa matriser: SparseTensor. För de kristaller som endast hade fem grannar fylldes det sista elementet ut med 0. Det undersöktes att fylla ut med värdet -1 för att tydligt markera för filtren att det endast var fem grannar som filtret täckte, men detta gav försämrat resultat i de undersökta fallen.

Den första grannenergin kom alltid från den granne som var närmast kristallen med index 81 (strålutgångkristallen), och övriga följde i ordning. Detta gjordes för att ge effekten av att faltningsfiltren translaterades med konstant orientering över kristallerna. Faltningsfilter av storleken 1×7 translaterades sedan över den $162 \cdot 7$ radvektorn, med steglängd 7, vilket gav effekten att varje filter translaterades över varje kristall med tillhörande grannar. Resultatet från detta faltningslager kan alltså representeras som en $162 \times$ (antal filter) matris, som sedan plattades ut för att användas som indata för de fullt kopplade lagren.

A.1.3 Faltning med närmsta grannar två gånger i följd (NG-NG)

NG-NG-faltning är två stycken NG-faltningslager i följd. Dock fås en $162 \times$ (antal filter) stor matris som utdata från det första faltningslagret, och konverteringen som illustreras i figur A.1 behöver göras för varje kolonn. Detta implementerades genom att platta ut den $162 \times$ (antal filter) stora matrisen, göra matrismultiplikation med en gles matris, och sedan omforma resultatet till en $162 \cdot 32 \times$ (antal filter) matris med funktionen reshape i TensorFlow.

$$[E_1 \ \dots \ E_{162}]\,\boldsymbol{A} \implies [E_{1,1}, E_{1,2}, E_{1,3}, E_{1,4}, E_{1,5}, E_{1,6}, E_1 \dots E_{162,1}, E_{162,2}, E_{162,3}, E_{162,4}, E_{162,5}, E_{162,6}, E_{162}]$$

Figur A.1: *Presentation av konverteringen av indata för att kunna implementera ett faltningslager med närmsta grannar. Var och en av energierna i den 162 långa vektorn med kristallernas energier parades ihop med dess grannar till en 7 element lång vektor och tillsammans bildade dessa en $7 \cdot 162$ lång vektor. Detta gjordes genom att multiplicera den med matrisen $\boldsymbol{A}$. Varje kristall har antingen fem eller 6 grannar och för fallen med 5 grannar fylldes det tomma elementet ut med en nolla.*

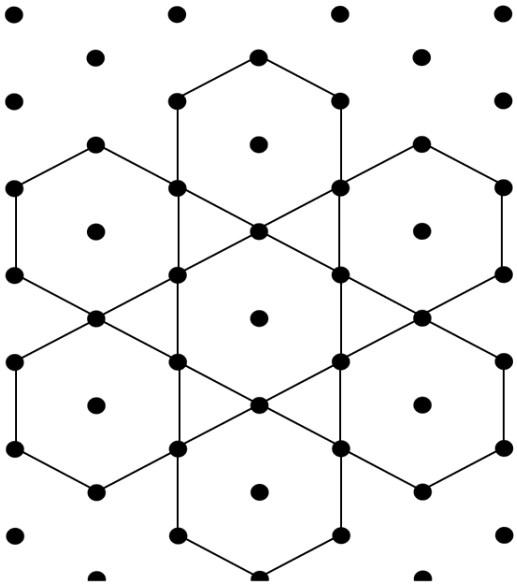
A.1.4 Två faltningslager med medelvärdes-pooling (NG-pool-NG-pool)

Hur pooling-filtret var utformat och hur det translaterades, illustreras i figur A.2. Det första medelvärdes-pooling-lagret implementerades genom att först bestämma vilka kristaller som pooling-filtren skulle placeras över. Anledningen till varför det fungerar att benämna utnoderna från faltnings- och pooling-lager som kristaller är att alla filter är centrerade runt en kristall. Pooling-kristallerna bestämdes genom att placera ut ett pooling-filter på stålutgångskristallen och resterande filter placerades utifrån denna. Detta tillvägagångssätt resulterade i 47 stycken pooling-kristaller. Sedan konverterades den $162 \times$(antal filter)-matrisen som var resultatet från det första faltningslagret, till en $162 \cdot 7 \times$ (antal filter)-matris, där varje rad konverterats på samma sätt som indatalagret för det första faltningslagret. Därefter behölls endast de $7 \times$ (antal filter) element som motsvarande de kristaller som pooling ville tillämpas på. Detta gjordes i praktiken genom att matrisen som gjorde konverteringen, också placerade pooling-kristall-grupperna först, och de extraherades genom att plocka ut de $47 \cdot 7$-första elementen. Sedan tillämpades medelvärdes-pooling på denna nya matris med en filterstorlek på 1×7 med steglängd 7. Som utdata gav det första poolinglagret alltså en $47 \times$ (antal filter)-matris.

Samma procedur som ovan gällde även för det andra faltningslagret och pooling-lagret. Det vill säga faltningsfiltrena translaterades över de 47-kristallerna och täckte de närmsta grannarna. Pooling-filtret verkade också enligt figur A.2. Som utdata gav det första poolinglagret en $14 \times$ (antal filter)-matris, som plattades ut för att användas som indata för de fullt kopplade lagren.

A.1.5 Två faltningslager utan pooling (NG-NNG)

Denna strategi bestod av att placera filter som täckte de närmsta grannarna, de näst närmsta grannarna och centerkristallen, på de 14 kristaller som var resultatet från metoden NG-pool-NG-pool. Precis som för NG-NG-faltningen, plattades resultatet från NG-lagret ut och multiplicerades med en matris. I detta fall resulterade dock denna matrismultiplikation, efter omformingen med rehape funktionen, i en matris på $19 \cdot 14 \times$ (antal filter), där siffran 19 kommer från att en kristall som mest kan ha 18 grannar plus nästnärmsta grannar och siffran 14 är de kristaller som blev kvar efter NG-pool-NG-pool metoden (se avsnitt A.1.4). I de fall när dessa kristaller inte var 18 stycken, fylldes resterande element ut med nollor. Efter att de 1×19 stora faltningsfiltren tillåtits att translatera över varje kolonn i $19 \cdot 14 \times$ (antal filter)-matrisen med steglängd 19, kvarstod en $14 \times$ (antal filter) stor matris. Denna plattades sedan ut för att användas som indata för de fullt kopplade lagren.



Figur A.2: *Illustrering av hur de hexagonala pooling-filtren translaterades över kristallerna i Crystal Ball, som motsvaras av noderna i figuren. I denna figur är sju filter placerade över sju kristaller, men i praktiken translaterades de över hela detektorn. Det hexagonala gittret är från Wikimedia Commons [29].*

A.2 Träningstider

I detta avsnitt behandlas träningstider för multiplicitetsrekonstruktion med olika faltningsmetoder. Träningstiderna presenteras i tabell A.1. Alla nätverk förutom det utan faltning, hade 2 lager med 2048 noder per lager. Nätverket utan faltning var ett av de nätverk som presterat bäst när ingen faltning var tillämpat, det med 5 lager och 2048 noder per lager. Träningstiden till nätverken konvergerat definierades som den tid som krävdes för att nå 99% av den maximala pricksäkerheten. Alla mätningar gjordes med en dropout-sannolikhet på 0. Dock noterades att träningstiden till konvergens förändrades när olika dropout-sannolikheter användes, både till det bättre och till det sämre, så tabellen ger mest en indikation till hur tideffektiva de olika metoderna är.

Tabell A.1: *Träningstider för de olika faltningmetoderna. Alla nätverk förutom det utan faltning hade 2 lager med 2048 noder per lager. Nätverket utan faltning var ett av de nätverk som presterat bäst när ingen faltning var tillämpat, det med 5 lager och 2048 noder per lager. Träningstiden till nätverken konvergerat definierades som den tid som krävdes för att nå 99% av den maximala pricksäkerheten. Alla mätningar gjordes med en dropout-sannolikhet på 0.*

Faltningsmetod	Träningstid för $2 \cdot 10^6$ iterationer (h)	Träningstid tills konvergens (h)
Utan faltning	8,9	3,8
NG	7,8	3,7
NG-NG	10,5	2,1
NG-pool-NG-pool	8,2	5,5
NG-NNG	5,8	2,7

B

Nätverkets struktur

B.1 Konvergens av nätverk

För en rättvis jämförelse nätverk sinsemellan behöver ett konsekvent konvergensvillkor hittas. Detta för att effekter från träningen inte ska ha någon inverkan på nätverkets prestation och på så sätt ge ett missvisande resultat. I det här arbetet har två varianter primärt testas: en typ av ”derivatametod” och en *early stopping*-metod.

”Derivatametoden” jämförde differenser i kostnadsfunktionen mellan olika träningsiterationer och när kostnadsfunktionens förändringstakt blivit mindre än ett bestämt tröskelvärde stoppades träningen. Konvergensvillkoret var dock problematiskt för större nätverk eftersom de har större tröghet och konvergerar långsammare. Det gjorde att de större nätverken avslutades i ett tidigare stadie i träningsprocessen och därför skulle denna typ av konvergensvilkor missgynna större nätverk.

Early stopping-metoden testades på två olika sett. Dels applicerades den på kostnadsfunktionen och träningen stoppades när denna inte minskat efter 1000 försök att uppdateras. Eftersom kostnadsfunktionens värde var så pass brusigt så avbröts träningen mer eller mindre slumpmässigt och det optimala värdet motsvarade ett väldigt lokalt minima. Early stopping applicerades även på den anpassade fördelningen av felet för $\cos\theta$ men eftersom anpassningen misslyckades tidsvis så gav även denna metod en missvisande konvergens. Tillvägagångssättet att studera de fysikaliska parametrarna istället för kostnadsfunktionen är troligen närmare ett tillfredställande konvergensvilkor eftersom det är dessa storheter som är intressanta vid ett experiment. Metoden går att applicera exempelvis på nätverk med olika viktningsfaktorer λ mellan termerna i kostnadsfunktionen eftersom konvergensen inte skulle bero på kostnadsfunktionens värde (fortfarande problem med AD och RD då dessa får skeva fördelningar).

Konvergensvillkoret är fortfarande en öppen fråga och något som är viktigt för framtida jämförelser av olika nätverksarkitekturer. Ingen större effekt av överträning har dock uppkommit för de nätverk som använts under projektets gång och det kan troligtvis vara en orsak till att konvergensmetoderna inte fungerat. Dessa metoder innehåller också parametrar som kan optimeras, till exempel antal uppdateringar av den minsta kostnadsfunktionen innan träningen stoppas eller tröskelvärdet för förändringstakten. Vidare studier behövs för att finna ett tillfredsställande konvergensvillkor.

B.2 Variation av nätverksarkitektur

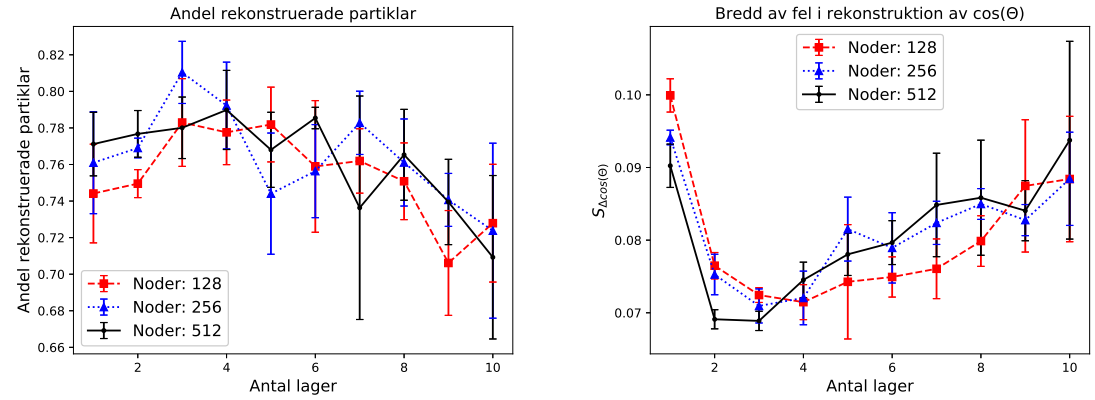
Att jämföra nätverk sinsemellan kan exempelvis göras genom att studera storheter som standardfel i rekonstruktionen, men för att detta ska vara en rättvis jämförelse krävs det

att nätverken gör fel på samma sätt. En annan metod för jämförelser mellan nätverk presenteras i kapitel 4 där nätverkens prestation i ett tänkt experiment studeras. Detta anses vara ett mer robust sätt att utvärdera nätverkets arkitektur, speciellt eftersom det sammanfattar de olika aspekterna av rekonstruktionen och viktar de på ett sätt som är relevant vid användning av algoritmen.

I detta avsnitt studeras nätverk där endast arkitekturen varierats. Det gjordes genom att antalet noder och lager varieras på ett sådant sätt att antalet noder i alla lager var samma. Eftersom samma kostnadsfunktion användes i alla de studerade fallen anses felen vara så lika att en jämförelse mellan fördelningen av felet i rekonstruktionen kan ses som ett mått på nätverkets prestation. Till fördelningen av felen anpassas är en t2-fördelning, se appendix B.3.

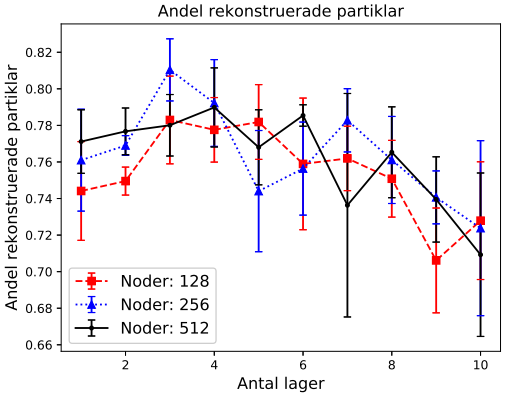
I figur B.1 visas en jämförelse mellan olika arkitekturer. På grund av avsaknaden av konvergensvillkor så har nätverken tränats i $5 \cdot 10^5$ iterationer, en träningslängd som av erfarenhet sätt ge konvergerade nätverk. Kostnadsfunktionen som använts är en relativ kostnadsfunktion med $\lambda = 1$. Framförallt visas att vinkelrekonstruktionen förbättras med flera lager till att börja med, men även att ökat antal noder i varje lager ger en förbättring. Att tänka på är att antalet noder ger än större ökning av antal parametrar i modellen (stora nätverk: #parametrar $\approx$ #lager $\cdot$ (#noder)^2). Rekonstruktionen verkar sedan försämras vilket kan bero på att de större nätverken inte har konvergerat. Troligen beror det på en diskripans mellan hur nätverkens prestation bedöms eftersom denna försämring inte samma grad har uppkommit vid andra utvärderingar av nätverkets prestation. Se exempelvis kapitel 4.

För energirekonstruktionen, figur B.1b och B.1c, ses inga drastiska förändringar för de olika nätverkarkitekturerna. Detta skulle betyda att rekonstruktion av energin är ett så pass enkelt problem att alla dessa nätverk är tillräckligt stora för att genomföra rekonstruktionen. Att ha 10 lager och 256 noder som standard för arkitekturen var ett val som visat sig fungera för de problem som skulle lösas. Att ha ändrat den arkitekturen till följd av denna undersökning anses inte motiverat på grund av de små skillnaderna. Det bör inte ha någon större inverkan på de resultat från händelserekonstruktionen som tagits fram.



(a)

(b)



(c)

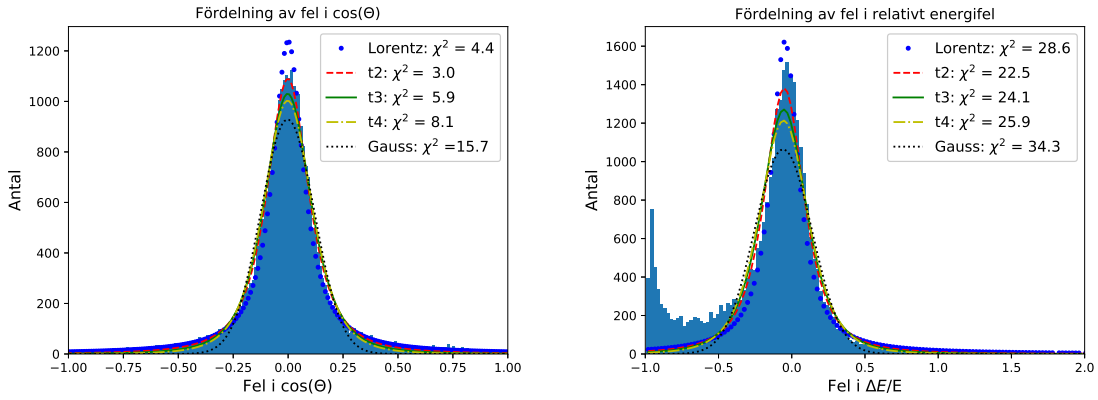
Figur B.1: *Parametrar som beskriver fördelningen av felen i rekonstruktionen för olika antal lager och noder i varje lager. Nätverket är tränat med en relativ kostnadsfunktion i* $5 \cdot 10^5$ *iterationer. (a) Bredden av t2-fördelningen som beskriver felet i* $\cos\theta$*. (b) Bredden av t2-fördelningen för det relativa felet i energirekonstruktionen. (c) Andelen partiklar i energirekonstruktionen som rekonstrueras.*

B.3 Fördelning av felen i rekonstruktionen

Vid en närmare studie av fördelningen av felet i nätverkets rekonstruktion upptäcktes att dessa inte följer en Gaussisk fördelning, framförallt eftersom fördelningen inte dör av tillräckligt snabbt. Ett exempel ses i figur B.2 där fördelningen av felen i rekonstruktionen ses där kostnadsfunktionen RD använts.

Tillsammans med fördelningen i figur B.2 ses olika exempel på fördelningsfunktioner som har anpassats till fördelningen. Standardmässigt för anpassningar av histogram antogs mätosäkerheten i varje stapel vara roten ur stapelhöjden. Den Gaussiska fördelningen dör av för snabbt och beskriver inte ytterligheterna av felen i rekonstruktionen. Vinkelrekonstruktionen, figur 3.6, beskrivs bäst av en t2-fördelning eftersom denna får bäst χ^2. De relativa felen i energirekonstruktionen, figur B.2b, har två toppar. Den första vid den korrekta rekonstruktionen med även en vid 100 %-fel vilket motsvarar de partiklar som inte rekonstruerats och placerats i ”mattan”. Ingen av fördelningarna som testas beskriver båda

topparna men av de som studeras beskriver även här t2-fördelningen rekonstruktionens fel bäst, eftersom den fångar huvudtoppens karaktär bäst.



(a) **(b)**

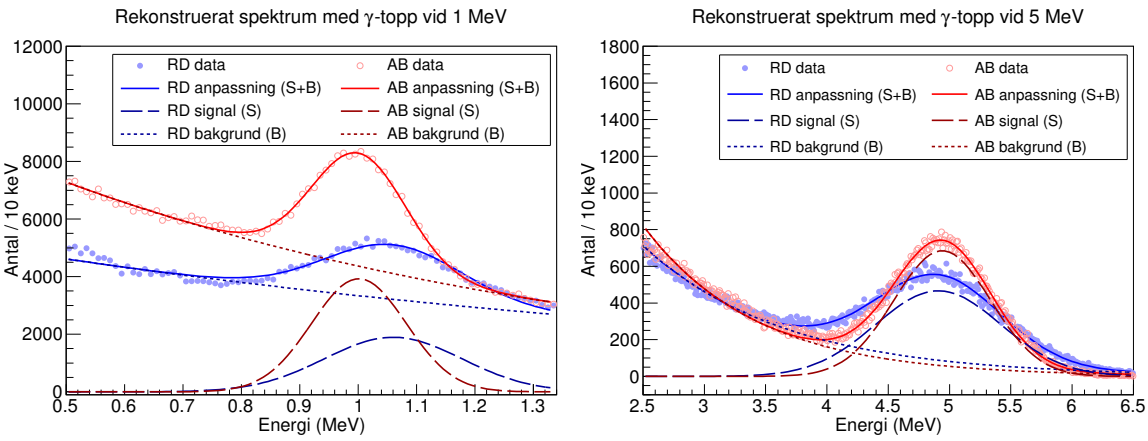
Figur B.2: *Fördelningar av fel i rekonstruktionen för ett nätverk med kostnadsfunktion RD. (a) För vinken ses att fördelningen inte är Gaussisk eftersom fördelningen inte dör av tillräckligt snabbt. Bästa anpassningen fås av en t2 fördelning eftersom denna får bäst χ^2. (b) Fördelning av det relativa felet. Det finns två toppar där toppen centrerad kring origo motsvarar korrekta rekonstruktioner medan den andra toppen vid 100 % fel motsvarar rekonstruktioner i mattan. Även här ses att en t2-fördelning verkar bäst, även om den inte fångar den andra toppen.*

Eftersom t2-fördelningen beskriver felen bäst är det denna fördelningen som använts för att jämföra olika nätverk med varandra. Normalt normerades fördelningarna innan anpassning gjordes men fortfarande så anpassades en amplitud. Amplituden ses nu som ett mått för hur stor andel av de rekonstruerade partiklarna som anpassningen har fångat. I fallet med energin visar det även hur stor andel av de rekonstruerade partiklarna som har placerats i den korrekt rekonstruerade toppen och inte placerats i mattan.

C

Anpassningar av rekonstruerade spektrum för jämförelse av realistisk data

I figur C.1 visas två ytterligare exempel på anpassningar av rekonstruerade energispektrum som använts för att beräkna signal-bakgrund-förhållanden i figur 4.2. Likt avsnitt 4.1 jämförs addback-rutinen (AB) med ett nätverk tränat med kostnadsfunktionen Relativ Doppler (RD). I figur C.1a rekonstrueras data med γ-topp vid 1 MeV och i figur C.1b rekonstrueras data med γ-topp vid 5 MeV. Anpassningarna är gjorda på samma sätt som i avsnitt 4.1. Skillnaden mellan spektrumen är dels olika bakgrundsnivåer kring signalerna, dels signalernas olika amplitud och bredd. Notera de olika skalorna.



(a) **(b)**

Figur C.1: *I (a) och (b) visas rekonstruerade spektrum från data med γ-toppar vid* 1 MeV *respektive* 5 MeV*. Anpassningarna användes för att beräkna S/B-värden presenterade i figur 4.2.*