

TECHNISCHE UNIVERSITÄT DRESDEN

FAKULTÄT INFORMATIK

INSTITUT FÜR SOFTWARE- UND MULTIMEDIATECHNIK

PROFESSUR FÜR COMPUTERGRAPHIK UND VISUALISIERUNG

PROF. DR. STEFAN GUMHOLD

## Großer Beleg

# Parameterrekonstruktion für Röntgen-Kleinwinkelstreuung mit Deep Learning

Patrick Stiller

(Mat.-Nr.: 3951290)

Betreuer: Dr. Dmitrij Schlesinger, Dr. Heide Meißner, Dr. Michael Bussmann

Dresden, 15. Januar 2019

---

## Aufgabenstellung

aufgabenstellung

---

# Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die von mir am heutigen Tag dem Prüfungsausschuss der Fakultät Informatik eingereichte Arbeit zum Thema:

*Parameterrekonstruktion für Röntgen-Kleinwinkelstreuung mit Deep Learning*

vollkommen selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Dresden, den 15. Januar 2019

Patrick Stiller

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Grundlagen</b>	<b>4</b>
2.1	Physikalischer Hintergrund . . . . .	4
2.1.1	Kleinwinkelstreuung . . . . .	4
2.1.2	Experimentbeschreibung . . . . .	5
2.2	Problemidentifikation . . . . .	7
2.3	Fouriertransformation . . . . .	7
2.4	Filter . . . . .	8
2.5	Neuronale Netze . . . . .	9
2.5.1	Aufbau . . . . .	9
2.5.2	Feed-Forward-Neuronal Networks . . . . .	10
2.5.3	Lernprozess . . . . .	11
<b>3</b>	<b>Simulation</b>	<b>13</b>
3.1	Motivation . . . . .	13
3.2	Simulationsbeschreibung . . . . .	13
3.2.1	Design der Gitterstruktur und Simulation des Hochenergielasereinflusses . . . . .	13
3.2.2	Effekt der Startparameter auf die Gitterstruktur . . . . .	14
3.2.3	Simulation der Streuung . . . . .	15
3.2.4	Auswirkung der Startparameter auf das Simulationsergebnis . . . . .	16
<b>4</b>	<b>Datenbasis</b>	<b>18</b>
4.1	Generator . . . . .	18
4.2	Wahl der Eingangsparameter . . . . .	19
4.3	Training-, Validation- und Testdatensatz . . . . .	19
<b>5</b>	<b>Neuronales Netz</b>	<b>20</b>
5.1	Architektur . . . . .	20

5.2	Lossfunction . . . . .	20
5.3	Optimizer . . . . .	20
<b>6</b>	<b>Ergebnisse und Diskussion</b>	<b>21</b>
6.1	Lernprozess . . . . .	21
6.2	Ergebnisse und statistische Auswertung . . . . .	21
6.3	Fazit . . . . .	21
	<b>Literaturverzeichnis</b>	<b>22</b>

# 1 Einleitung

## 2 Grundlagen

### 2.1 Physikalischer Hintergrund

#### 2.1.1 Kleinwinkelstreuung

Die Kleinwinkelstreuung in Englisch Small Angle X-Ray Scattering (SAXS) ist eine universelle Technik zur Untersuchung von Feststoffen. Dabei kann SAXS Informationen über die kristalline Struktur, chemische Komposition und die physikalische Eigenschaften eines untersuchten Feststoffes liefern [Cro]. Die Untersuchung von Feststoffen unter Einfluss von Kurzpulslasern ist eine wichtige Aufgabe der heutigen Physik. Wissen über das Verhalten von Feststoffen unter Einfluss von Kurzpulslasern kann offene Fragen in der Krebsforschung und in der Astrophysik beantworten. Dieser Beleg bezieht sich auf die Publikation von Thomas Kluge, welches 2018 mit dem Titel: “Observation of ultrafast solid-density plasma dynamics using femtosecond X-ray pulses from a free-electron laser” veröffentlicht wurde. In der angesprochenen Publikation wurde SAXS für die Untersuchung von Plasma eingesetzt [KRM<sup>+</sup>18].

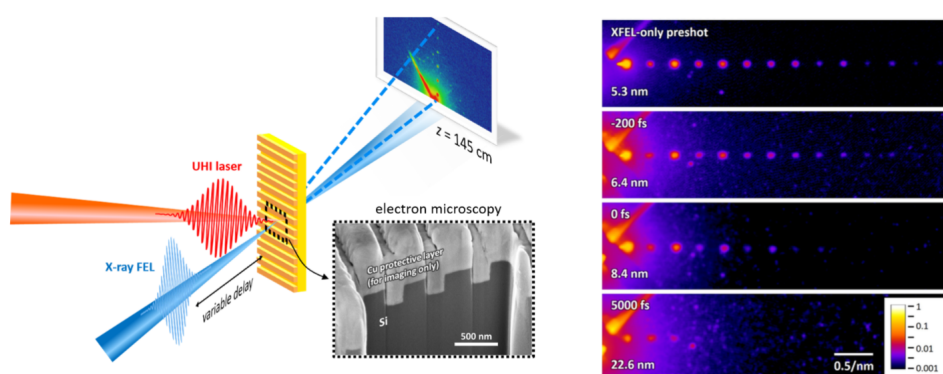


Abbildung 2.1: Schematischer Aufbau des Experimentaufbaus. Im rechten Teil der Abbildung sind Detektorbilder in Abhängigkeit des Einsetzens des Röntgenlaserpulses dargestellt. Bildquelle: [KRM<sup>+</sup>18]

### 2.1.2 Experimentbeschreibung

Das in [KRM<sup>+</sup>18] beschriebene Experiment besteht aus vier Hauptkomponenten (siehe Abbildung 2.1 links). Dem Target, dem Hochintensitätslaser (UHI), dem Röntgen-Freie-Elektronen-Laser(XFEL) und dem Detektor. Beim Target handelt es sich um ein Kupfertarget. In das Kupfertarget wurde eine Gitterstruktur eingraviert, um eine analytische Beschreibung des Experimentausgangs zu ermöglichen. Zusätzlich wird das Target mit einer 2  $\mu\text{m}$  breiten Siliziumschicht überzogen. Durch die eingravierte Gitterstruktur besitzt das Target nur noch einen eindimensionalen Informationsgehalt und kann somit durch drei Parameter vollständig beschrieben werden. Die drei Parameter sind Pitch, Feature-Size und die Aufweichungsbreite  $\sigma$ . Dabei beschreiben Pitch und Feature-Size die Struktur des Targets und Sigma den Aufweichungseffekt des Hochintensitätslasers (siehe Abbildung 2.2).

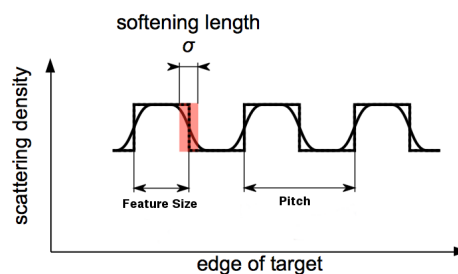


Abbildung 2.2: Analytische Beschreibung des Querschnittes des Targets. Dabei ist das Grating zu erkennen und dass es durch drei Parameter: Pitch, Feature-Size und Sigma beschrieben werden kann. Bildquelle: [Zac17]

Während des Experiments wird das Target durch einen Hochintensitätslaser in einem Winkel von  $90^\circ$  beschossen. Aufgrund des Elektrischen Feldes des Lasers entsteht Plasma. Bei Plasma handelt es sich um ein Gemisch aus freien Elektronen, positiven Ionen und neutralen Teilchen, welche unter ständiger Wechselwirkung untereinander und mit Photonen stehen. Dadurch kann es zu unterschiedlichen Energie- bzw. Anregungszuständen kommen. Der Plasmazustand eines Stoffes wird auch als vierter Aggregatzustand bezeichnet [Wis]. Während des Beschusses durch den Hochintensitätslaser ist außerdem ein Schmelzprozess und somit eine Aufweichung der Gitterstruktur des Targets wahrzunehmen (siehe Abbildung 2.3).

Das durch den Hochintensitätslaser erzeugte Plasma soll es auf seine Struktur und Elektrodynamik untersucht werden. Dafür wird leicht zeitversetzt ein zweiter Laser benutzt. Dabei handelt es sich um einen Röntgen-Freie-Elektronen-Laser, welcher in einem Winkel von  $45^\circ$  mit einer Pulsdauer von 40 Femtosekunden auf das Target schießt. Bei diesem Vorgang kommt es zu einem Streuvorgang des Lichts des Röntgenlasers an den Elektronen des Targets. Die Intensitäten des gestreuten Röntgenlasers wer-



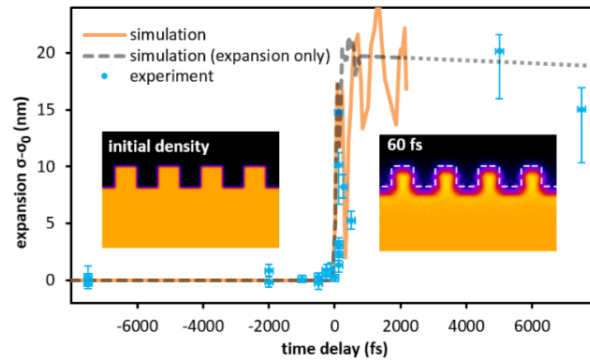


Abbildung 2.3: Veränderung des Gratings nach einer Bestrahlungsdauer von 60 Femtosekunden Bildquelle: [KRM<sup>+</sup>18]

den durch einen Detektor, welches hinter dem Target platziert ist, gemessen (Abbildung 2.1 links). Beim Detektor handelt es sich um ein Raster von Lichtdetektoren, welche die ankommenden Lichtintensitäten messen. Dabei ist der Lichtdetektor Zeit integrierend, das heißt, das ankommende Lichtintensitäten über die Zeit summiert werden. Diesen Effekt ist in der Abbildung 2.4 zu erkennen, welche einen beispielhaften Streuprozess an zwei Elektronen zeigt.

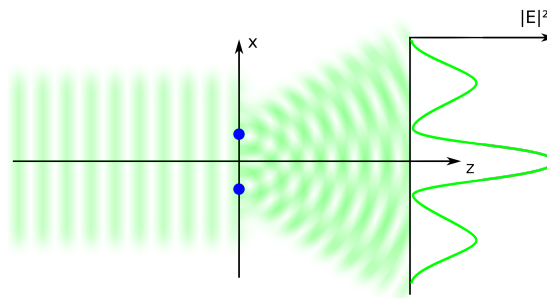


Abbildung 2.4: Beispielhafte Darstellung einer Streuung an zwei Elektronen. Die Röntgenlaserpulse treffen geradlinig auf die Elektronen und werden dann bei den Elektronen gestreut. Der Detektor dahinter misst die ankommenden Intensitäten der kreisförmigen Wellen und summiert diese über die Zeit. Bildquelle: [Zac17]

Durch die Zeitintegrität des Detektors, gehen die zeitlichen Abstände der Wellen verloren (Phase). Das entstandene Detektorsignal ist äquivalent zum Betragsquadrat der Fouriertransformation Gitterstruktur  $\eta$  (siehe Gleichung (2.1)) [KRM<sup>+</sup>18].

$$\Phi \approx |r_0 \cdot \int \eta(\vec{r}) \cdot e^{i\vec{q}\vec{r}} d\vec{r}|^2 \quad (2.1)$$

## 2.2 Problemidentifikation

Einer der Herausforderungen des beschriebenen Experiment ist die Zeitintegrität des Detektors. Diese Detektoreigenschaft ruft den Verlust der Phase hervor, sodass eine Rekonstruktion der Elektronendichteverteilung  $\eta$  mit Hilfe einer inversen Fouriertransformation (IFT) nicht möglich ist. Um dieses Problem zu lösen werden iterative Algorithmen, sogenannte Phaseretrieval-Algorithmen verwendet. Beispiele für solche Algorithmen sind [Fie82]: Error-Reduction Algorithm, Gradient Search Methods und der Input Output Algorithmus. Probleme bei diesen Algorithmen sind, dass erstens bei allen Algorithmen keine Konvergenz garantiert ist und zweitens sehr viele Iterationen notwendig sind um eine Optimierung der errechneten Phase zu erzeugen. Diese Eigenschaften führen dazu, dass eine hochfrequente Verarbeitung von Experiment Ergebnissen verwehrt bleibt. Deswegen soll mit Hilfe von Deep Learning die Phase bzw. die Eigenschaften (drei Parameter zur Beschreibung des Gratings) der Elektronendichteverteilung  $\eta$  rekonstruiert werden. Der gewählte Deep Learning Ansatz ist ein neuronales Netz. Ein neuronales Netz hat den Vorteil, dass es sich im Gegensatz zu iterativen Verfahren durch GPU parallelisieren lässt und es die Phaseninformation bzw. die Parameter der Elektronendichte nicht-iterativ bestimmt(Einschrittverfahren).

## 2.3 Fouriertransformation

Die Fouriertransformation (benannt nach Jean Baptiste Joseph Fourier) ist eine Transformation, welche zeitbezogene Welle im Ortsraum in ihre frequenzmäßigen Spektralanteile zerlegt(Abbildung 2.5). Die Fouriertransformation wird auch als Transformation vom Orts- in den Frequenzraum bezeichnet. Die Fouriertransformation ist folgendermaßen definiert:

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x u} dx \quad (2.2)$$

Die Spektralanteile  $F(u)$  werden wie bei einem Basiswechsel durch ein Integral bestimmt. Dabei ist das Ergebnis des Integrals die Länge der Projektion in Richtung der durch  $u$  kodierten Frequenz. Im endlich-diskreten Fall wird das Integral durch eine Summe bis zur Anzahl der zu verwendeten Wellen ersetzt. Jede Stelle  $u$  der Fouriertransformation kodiert eine Welle  $e^{-2\pi i x u}$ , welche über die Euler Identität  $e^{ikx} = \cos(kx) + i \cdot \sin(kx)$  auch in ihre Kosinus- und Sinusanteilen beschrieben werden kann. Der

Funktionswert der Fouriertransformation  $F(u)$  kodiert mit welchem Anteil die durch  $u$  kodierte Frequenz verwendet wird. Das Ergebnis der Fouriertransformation kann man auch durch das Amplituden- und durch das Phasenspektrum beschreiben. Dabei bestimmt die Amplitude das Maxima und das Minima der zugehörigen Welle und die Phase die Verschiebung der Welle. Das Amplitudenspektrum  $|F(u)|$  und das Phasenspektrum  $\phi(u)$  ergeben sich durch die folgende Gleichungen [Sch]:

$$|F(u)| = \sqrt{R^2(u) + I^2(u)} \quad (2.3)$$

$$\phi(u) = \tan^{-1} \frac{I(u)}{R(u)} \quad (2.4)$$

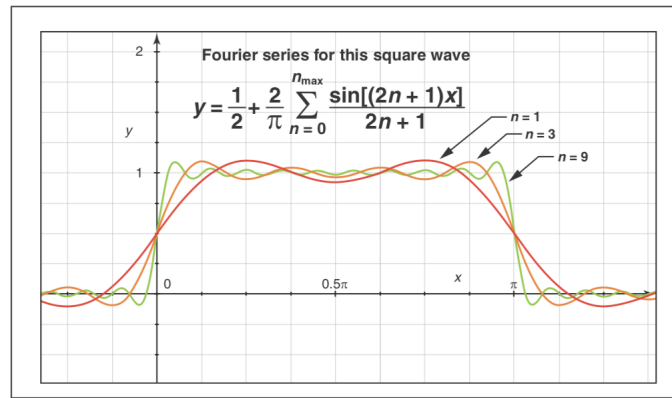


Abbildung 2.5: Überlagerung mehrerer Wellen um eine Rechteckfunktion zu approximieren. Bildquelle: [GNHB08]

## 2.4 Filter

Ein Filter ist eine Operation, welche auf einem Signal verwendet wird um Signale zu glätten, Signalstörungen zu vermeiden oder um Rauschen zu verhindern. In den meisten Fällen wird die Filteroperation mit Hilfe von Faltung realisiert. Dabei wird die Faltung zweier Funktionen ( $f * g$ ) durch den funktionalen Zusammenhang in Formel (2.5) beschrieben. Dabei ist  $f$  die Funktion, welche das Signal beschreibt und  $g$  die Funktion, welche den Filter beschreibt. Im diskreten Fall wird das Integral durch eine Summe bis zur entsprechenden Filtergröße ersetzt [Kla13].

$$(f * g)(x) := \int_{\mathbb{R}^n} f(\tau)g(x - \tau)d\tau \quad (2.5)$$

## 2.5 Neuronale Netze

### 2.5.1 Aufbau

Neuronale Netze sind eine mathematische Adaption des realen menschlichen Gehirns. Ein neuronales Netz besteht aus vielen kleinen Komponenten, Neuronen, welche durch gerichtete und gewichtete Verbindungen verbunden sind. Mathematisch definiert ist ein neuronales Netz ein Tripel  $(N, V, w)$  mit den Beiden Mengen  $N$  und  $V$  und der Funktion  $w$ .  $N$  ist die Menge aller Neuronen und  $V = \{(i, j) | i, j \in N\}$  die Menge der Verbindungen zwischen Neuron  $i$  und Neuron  $j$ . Die Funktion  $w : V \rightarrow \mathbb{R}$  beschreibt die Gewichte des Neuronalen Netzes. Wobei  $w(i, j)$  das Gewicht zwischen dem Neuron  $i$  und Neuron  $j$  beschreibt. Im Allgemeinen wird anstatt der Funktionsnotation die Notation  $w_{i,j}$  für die Gewichte zwischen zwei Neuronen verwendet. [Kri07] Die nächste wichtige Komponente ist die Propagierungsfunktion  $net_j$  eines Neurons, welche einen wichtigen Teil des Informationsflusses in einem Neuronalen Netz definiert. Dabei wird der Input des Neurons  $j$  durch dessen Propagierungsfunktion  $net_j$  bestimmt. Die Propagierungsfunktion  $net_j$  nimmt den Output aller Neuronen welche eine ausgehende Verbindung zum Neuron  $j$  besitzen als Input. So wird die Propagierungsfunktion  $net_j$  durch folgenden funktionalen Zusammenhang beschrieben :

$$net_j = \sum_{i \in I_j} (o_i \cdot w_{i,j}) \text{ mit } I_j = \{i \in N | (i, j) \in V\} \quad (2.6)$$

Der Output  $o_j$  eines Neuronen  $j$  wird mit Hilfe der Aktivierungsfunktion  $a_j$  und der der Propagierungsfunktion berechnet. Dazu wird noch der Schwellwert  $\theta_j$  zur Hemmung des Aktivierungszustand des Neurons zur Hilfe genommen. Somit ergibt sich für den Output des Neurons folgender funktionaler Zusammenhang :

$$o_j = a_j(net_j - \theta_j) = f(x) \quad (2.7)$$

Die Wahl der Aktivierungsfunktion ist von Anwendungsfall von Anwendungsfall unterschiedlich. In den meisten Fällen werden differenzierbare Aktivierungsfunktionen benutzt, da sie den Lernprozess, des neuronalen Netzes erleichtern. Das erste Beispiel für eine Aktivierungsfunktion ist die Heaviside-Funktion(Step-Funktion), welche auf 0 oder 1 abbildet(Abbildung ??a). Diese Funktion ist an der Stelle 0 nicht differenzierbar und im generellen für moderne Optimierungsansätze für neuronale Netze nicht geeignet, da die Ableitung der Step-Funktion an allen Stellen 0 ist. Beispiel für gängige und differenzierbare Funktionen sind die Sigmoid-(Formel (2.8)), Tangens Hyperbolicus-(Formel (2.9)) und die Rectified Linear Units- Aktivierungsfunktion(Formel (2.10))

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.8)$$

$$f(x) = \tanh(x) = 1 - \frac{2}{e^{2x} + 1} \quad (2.9)$$

$$f(x) = \max(0, x) \quad (2.10)$$

Jeder der genannten Aktivierungsfunktion hat auf Hinsicht seiner Berechnungsdauer, Wertebereich und Ableitung Vor- und Nachteile. Die Sigmoid und die Tangens Hyperbolicus Aktivierungsfunktion haben einen eingeschränkten Wertebereich, dadurch kann es mit diesen Aktivierungsfunktionen zu keinen zu hohen Werten im Neuronalen Netz kommen und negative Werte werden im Gegensatz zu ReLu noch berücksichtigt. Jedoch sind beide langsamer zu berechnen als die Rectified Linear Units und bieten für den Lernprozess kleinere Gradienten. Rectified Linear Units(ReLu) ist im Vergleich zu Sigmoid und Tangens Hyperbolicus schneller zu berechnen und bietet größere Gradienten. Jedoch können Neuronen, welche einen negativen Input bekommen nur noch 0 Ausgeben. Man spricht in diesem Zusammenhang von einem totem Neuron. Ein weiterer Nachteil von ReLu ist der Wertebereich, denn dieser ist nicht eingeschränkt. Somit können im Neuronalen Netz sehr große Werte entstehen, welche den Wertebereich von Zahlenstandards wie zum Beispiel 32 Bit Float überschreiten und somit kein valider Datenfluss im Neuronalen Netz gegeben ist.

### 2.5.2 Feed-Forward-Neuronal Networks

Für ein Neuronales Netzwerk werden verschiedene Netzwerktopologien verwendet. Ein Beispiel dafür sind Feed Forward Neuronal Networks (Abbildung 2.6) Bei einem Feed Forward Neuronal Network werden die Neuronen als Schichten angeordnet. Diese Schichten sind miteinander verbunden. Dabei wird die Erste Schicht, welche die Input Daten bekommt, als Input-Layer bezeichnet. Und die letzte Schicht, welche die Netzberechnung ausgibt, als Output-Layer bezeichnet. Schichten, welche sich zwischen Input- und Output-Layer befinden und somit keinen Kontakt nach außen haben, werden als Hidden Layer bezeichnet. Ein wichtiges Merkmal von Feed Forward Networks ist, dass der Datenfluss geradlinig ohne Rückkopplung vom Input-Layer über die Hidden-Layer zum Output-Layer verläuft.

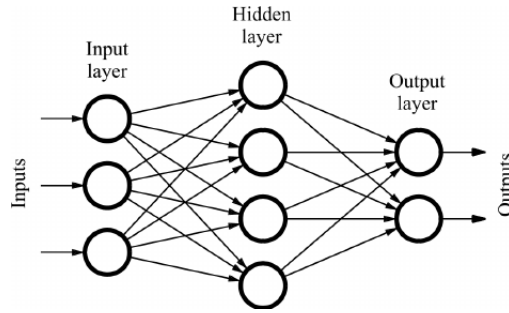


Abbildung 2.6: Beispielhafte Darstellung für ein Feed-Forward-Neuronal-Network. Bildquelle: [QR11]

### 2.5.3 Lernprozess

Das Anlernen von Neuronalen Netzen wird in den meisten Fällen durch überwachtes Lernen (supervised Learning) realisiert. Im speziellen wird der Lernprozess durch den Backpropagation Algorithmus und Gradientenabstiegsverfahren durchgeführt. Bei einem überwachten Lernansatz besteht der Datensatz zum Trainieren des Neuronalen Netzen aus zwei Teilen. Zu jedem Netinput  $x_i$  gibt es ein zugehöriges Label  $y_i$ , welches den gewünschten Netzoutput definiert. Der Lernprozess eines neuronalen Netzes kann in drei Phasen aufgeteilt werden. Der Forward-Pass, Loss-Calculation und Backward Pass [Bec]. Zu Beginn des Trainingsprozesses werden die Gewichte des Neuronalen Netzes mit Zufallszahlen initialisiert. Beim Forward Pass werden die Input Daten zur Kalkulation des derzeitigen Net-Outputs zum Input-Layer des Neuronalen Netzes gegeben. Das Neuronale Netz bestimmt dann durch die Rechenvorschriften des Neuronalen Netzes den Net-Output  $\hat{y}_i$ . Im zweiten Schritt wird dann die Qualität des Netouput  $\hat{y}_i$  bestimmt. Dazu wird ein Fehlermaß benutzt, das den Grad des Unterschiedes zwischen  $\hat{y}_i$  und  $y_i$  bestimmt. Eine beispielhafte Errorfunktion ist Mean-Squared Error (2.11).

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (2.11)$$

Im dritten Schritt, dem Backward-Pass, kommt es zur Optimierung des Neuronalen Netzes. Mithilfe des Fehlers, welcher bis zur Eingabeschicht zurück propagiert, werden die Gewichte, je nach ihrem Einfluss auf den Net-Output angepasst. Dazu wird ein Gradientenabstiegsverfahren verwendet. Dafür wird die partielle Ableitung des Fehlerterms benötigt. Somit ergibt sich für die Gewichtsveränderung  $\Delta w_{ij}$  des Gewichts zwischen Neuron i und Neuron j durch folgenden Funktionalen Zusammenhang :

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = -\eta \delta_j o_i \quad (2.12)$$

Dabei ist  $E$  die Errorfunktion,  $\delta_j$  der Gradient Neurons j und  $o_i$  die Ausgabe des Neuron i. Der letzte wichtige Parameter ist  $\eta$  welche die Learning-Rate des Gradientenabstiegsverfahren bestimmt. Schluss-

sendlich fehlt die Definition des Gradienten. Dieser ist davon abhängig, wie stark das Neuron den Output des Neuronalen Netz beeinträchtigt. Deswegen wird eine Unterscheidung getroffen, ob das Neuron sich im Output-Layer oder darunter befindet. In der folgenden Gleichung (2.13) ist die Definition des Gradienten und die zugehörige Propagierung des Fehlers.

$$\delta_j = \begin{cases} a_j(\text{net}_j)(o_j - t_j) & \text{falls } j \text{ Ausgabeneuron ist} \\ a_j(\text{net}_j) \sum_k \delta_k w_{jk} & \text{falls } j \text{ verdecktes Neuron ist.} \end{cases} \quad (2.13)$$

Dabei ist die Gleichung (2.13) in Abhängigkeit von den Variablen  $o_j$  dem Output des Neurons  $j$ ,  $t_j$  die Soll-Ausgabe des Neurons  $j$  und der Aktivierungsfunktion  $a_j$  des Neuron  $j$  angegeben. Und somit ergibt sich das neue Gewicht durch eine Addition des alten Gewichts mit der errechneten Gewichtsveränderung.

$$w_{ij}^{\text{neu}} = w_{ij}^{\text{alt}} + \Delta w_{ij} \quad (2.14)$$

Der Backpropagation wird iterativ solange angewandt, bis eine bestimmte Anzahl von Iterationen erreicht ist, oder andere Kriterien erfüllt sind. [Wik18]

## 3 Simulation

### 3.1 Motivation

Um das Ergebniss, des SAXS-Experimentes bestmöglich zu verstehen und aufgrund einer mangelnder Anzahl von Experimentdaten, wurde im Rahmen der Master Arbeit von Malte Zacharias mit dem Titel: “Model-Driven Parameter Reconstruction from Small Angle X-Ray Scattering Images “eine Modellierung entwickelt. Die Modellierung beinhaltet das Design der Gitterstruktur (Elektronendichteverteilung), die Modellierung des Hochenergielasereinflusses, die Modellierung des Streuvorgangs und die Modellierung des Detektorbildes. Das ganze Kapitel bezieht sich auf die Master Arbeit von Malte Zacharias [Zac17].

### 3.2 Simulationsbeschreibung

#### 3.2.1 Design der Gitterstruktur und Simulation des Hochenergielasereinflusses

Wie in Kapitel 2 beschrieben sind die Targets des SAXS-Experiments als Gitter strukturiert, um eine analytische Beschreibung des Targets zu ermöglichen. Der erste Simulationsschritt ist die Bestimmung der Gitterstruktur. Die Breite des Gratings ist auf N Pixel beschränkt. Die Struktur des Gratings ist über die drei Startparameter Pitch, Feature-Size und Sigma definiert. Dabei ist der Parameter Feature-Size für die Breite eines Features zuständig und der Parameter Pitch für die Periodizität, somit sind beide Parameter für die Struktur des Gratings ohne Einfluss des Hochenergielasers verantwortlich. Der Parameter Sigma bestimmt die Aufweichungsbreite des Gratings und somit den Einfluss des Hochenergielasers. Zunächst wird anhand der Parameter Feature-Size und Sigma, die Struktur eines Features bestimmt. Dazu wird eine Rechteckfunktion ( $1_{[0, \text{Feature-Size}]}$ ), welche die Breite eines Features bestimmt, mit einer Gaussverteilung ( $\exp(-x^2/2\sigma^2)$ ) gefaltet [KRM<sup>+</sup>18], um den Hochenergielasereinfluss zu simulieren. Das Ergebnis der Faltung wird mit Hilfe der Errorfunktion (Gleichung (3.1)) dargestellt.

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-\xi^2} d\xi \quad (3.1)$$



Das Ergebnis der Faltung des Rechteckimpulses und der Gauss-Verteilung ist die Modellierung eines Features:

$$\tilde{\eta} = \frac{\sqrt{\pi}\sigma}{2} \left( \operatorname{erf}\left(\frac{x}{\sqrt{2}\sigma}\right) - \operatorname{erf}\left(\frac{x - \text{fsize}}{\sqrt{2}\sigma}\right) \right) \quad (3.2)$$

Das modellierte Feature wird durch eine weitere Faltung mit mehreren um Pitch verschobene Dirac-Impulsen [Beu11] periodisch fortgesetzt wird, um die Grating-Struktur zu komplettieren.

### 3.2.2 Effekt der Startparameter auf die Gitterstruktur

Die drei Startparameter Sigma ( $\sigma$ ), Pitch und Feature-Size(fsize), welche die Struktur des Gratings beschreiben, haben unterschiedliche Effekte auf die Gratingstruktur. Sigma ist der Parameter, welcher, wie bereits beschrieben, den Einfluss des Hochenergielasers auf ein Feature modelliert. Desto höher Sigma gewählt wird, desto mehr wird die Kantenstruktur des Targets aufgeweicht. In Abbildung 3.1 ist dieser Effekt dargestellt.

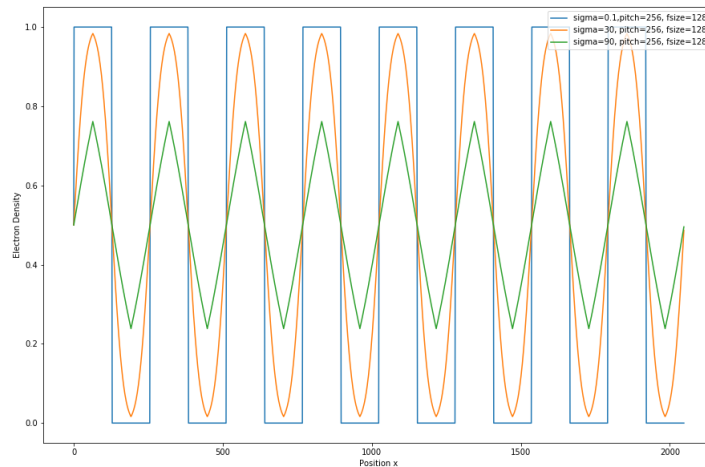


Abbildung 3.1: Vergleich der Kantenstruktur unter variierenden Sigma-Werten bei gleichbleibenden Pitch- und Feature-Size- Werten

Der Parameter Pitch(p) bestimmt die Periodizität der Kantenstruktur. Daraus folgt, dass mit steigendem Pitch-Wert die Anzahl der Features sinkt. Dieser Effekt ist in der Abbildung 3.2 in der rechten Spalte dargestellt. Der dritte Startparameter Feature-Size(f) bestimmt die Größe der Features. Dabei kann die Feature-Size nicht größer als Pitch gewählt werden, da sonst keine Kantenstruktur mehr erkennbar wäre. Der Effekt der Feature-Size wird in Abbildung 3.2 in der linken Spalte dargestellt.

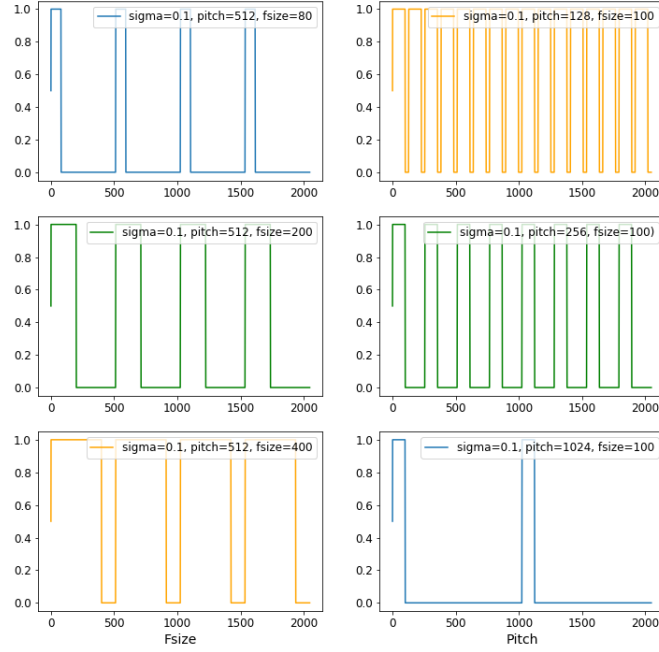


Abbildung 3.2: Vergleich der Kantenstruktur unter variierenden Pitch-Werten(rechte Spalte) und variierenden Feature-Size-Werten(linke Spalte)

### 3.2.3 Simulation der Streuung

Wie im Kapitel zur Experimentbeschreibung beschrieben, ist das aufgenommene Detektorbild äquivalent zum Betragsquadrat der Fouriertransformation der Elektronendichte  $\eta$ . Im Fall der Simulation ist die Elektronendichte eine diskrete eindimensionale Elektronendichteverteilung. Deswegen wird zur Bildung des Detektorbildes eine eindimensionale diskrete Fouriertransformation (3.3) verwendet.

$$\hat{a}_k = \sum_{j=0}^{N-1} e^{-2\pi i \cdot \frac{j \cdot k}{N}} \cdot a_j \text{ für } k = 0, \dots, N-1 \quad (3.3)$$

Das entstandene Produkt der Fouriertransformation ist im Raum der Komplexen Zahlen und weißt den gleichen Informationsgehalt wie die Elektronendichteverteilung auf. Das heißt Amplituden- und Phaseninformationen sind nach wie vor vorhanden. Wie in Kapitel 1 beschrieben, ist der Detektor zeitintegrierend und die Phase geht verloren, deswegen wird die Phaseninformation im weiteren Simulationsverlauf nicht weiter verwendet. Um schlussendlich die aufgenommenen Intensitäten des Detektors (siehe Abbildung 3.3) zu erhalten, wird das Betragsquadrat des Amplitudenspektrums gebildet.

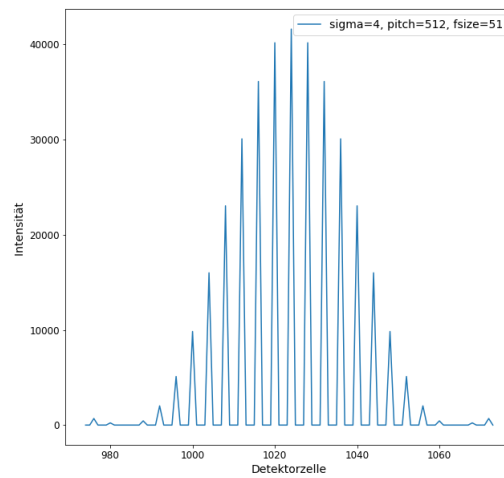


Abbildung 3.3: Resultierendes Detektorbild als Endprodukt der Simulation

### 3.2.4 Auswirkung der Startparameter auf das Simulationsergebnis

Die Startparameter der Simulation haben ebenfalls Auswirkung auf das Endprodukt der Simulation. Betrachtet man zunächst die Auswirkungen des Startparameters Pitch (Abbildung 3.4), kann man erkennen, dass mit steigendem Pitch sich das Intensitätsmaximum verringert. Zusätzlich kommen noch mehr Peaks hinzu. Das hängt damit zusammen, dass mit steigendem Pitch sich die Anzahl der Features verringert und somit mehr Frequenzen benötigt werden, um die Gitterstruktur zu modellieren.

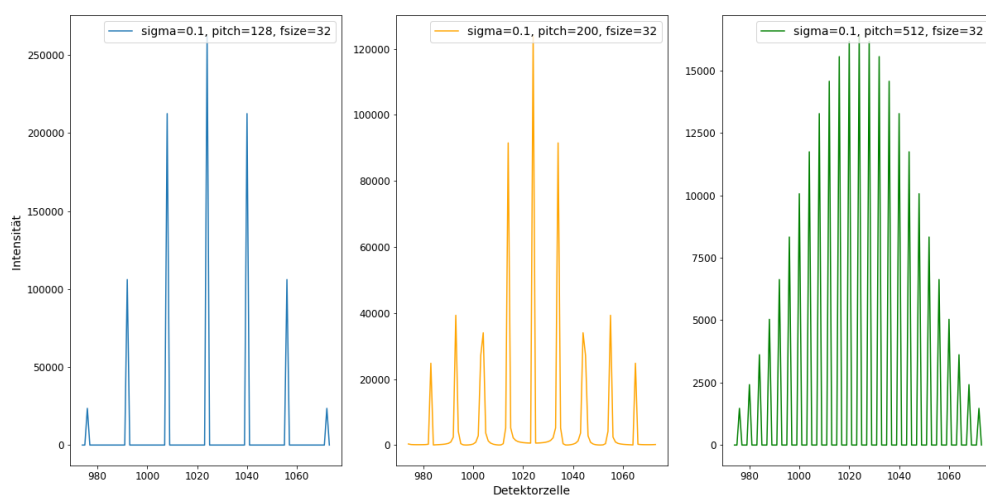


Abbildung 3.4: Verhalten der Detektoraufnahme in Abhängigkeit des Parameters Pitch

Der Startparameter Feature-Size verhält sich invers zum Parameter Pitch. Mit steigendem Feature-Size Wert, kommt es zu einer Erhöhung des Intensitätsmaximums und es werden weniger Frequenzen benötigt um die Gitterstruktur zu modellieren, somit sind auch weniger Peaks im Detektorsignal vorhanden (Abbildung 3.5).

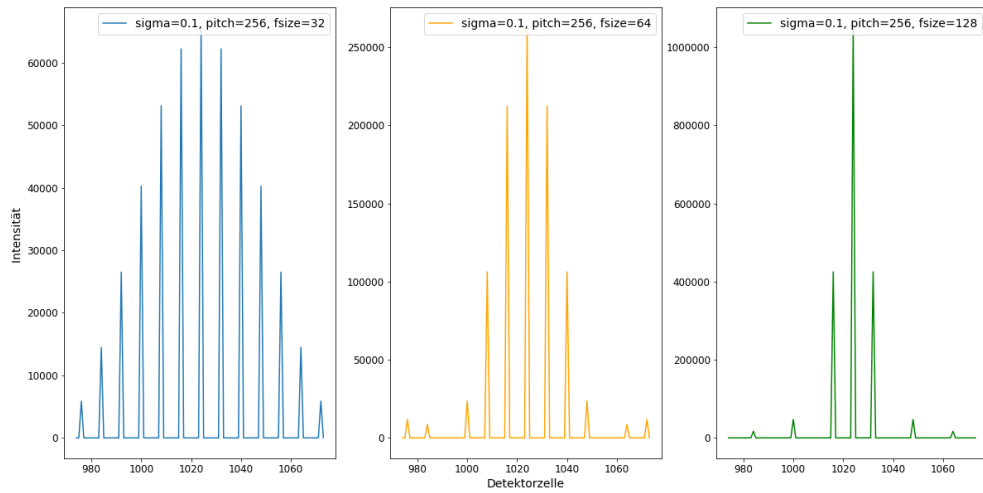


Abbildung 3.5: Verhalten der Detektoraufnahme in Abhängigkeit des Parameters Fsize

Der letzte Startparameter Sigma hat ansteigenden Wert ähnliche Auswirkungen auf die Detektoraufnahmen wie Fsize. Bei steigendem Sigma-Wert steigt das Intensitätsmaximum und die Anzahl der Peaks verringert sich (Abbildung Jedoch skaliert der Parameter Sigma anders als Fsize. Kleinere Unterschiede im Sigma Wert, kann größere Auswirkungen auf die Anzahl der Peaks haben. Auch der Anstieg des Intensitätsmaximums in Abhängigkeit zu Sigma ist geringer im Vergleich zur Abhängigkeit zu Fsize.

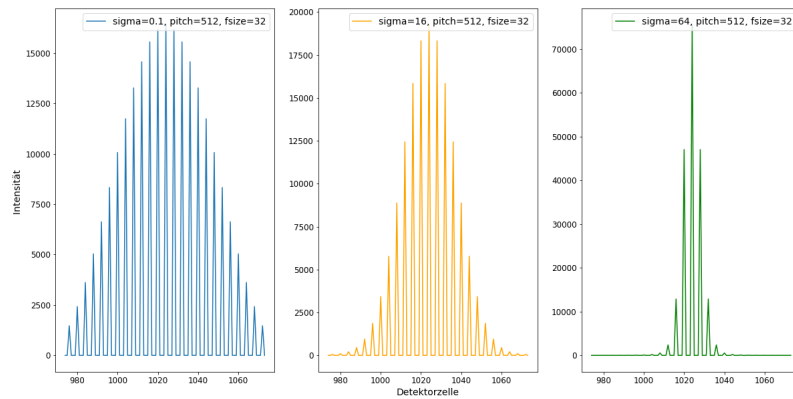


Abbildung 3.6: Verhalten der Detektoraufnahme in Abhängigkeit des Parameters Sigma

## 4 Datenbasis

### 4.1 Generator

Um einen Machine Learning Ansatz auszuführen, muss eine ausreichend große Datenbasis existieren. Dazu wurde im Rahmen dieser Arbeit ein Generator entwickelt, welcher mit Hilfe der vorher beschriebenen Simulation ausreichend viele Trainingsdaten produzieren kann. Bevor der Generator beginnen kann, muss eine Population  $P$  von Startparametern festgelegt werden. Um die Berechnung der Simulationsdaten, welche aus  $P$  resultieren, zu beschleunigen wurde die Simulation mit Hilfe von MPI (Message Parsing Interface) parallisiert. MPI dubliziert einen Algorithmus auf  $N$  beliebige Prozesse, welche dann mit Hilfe vordefinierter Routinen miteinander kommunizieren können [GL94]. Mit Hilfe von MPI-Routinen wurde die Population auf  $N$  MPI generierte Prozesse aufgeteilt. Jeder Prozess berechnet dann für seine kleinere Population von Startparametern die Simulationsergebnisse, welche dann anschließend als Dateien in den Speicher geladen werden. Zusätzlich zum Simulationsergebnis wird die Population der Startparameter und die daraus generierten Gitterstrukturen gespeichert, welche als Labels für das Training des neuronalen Netzes vorgesehen sind.

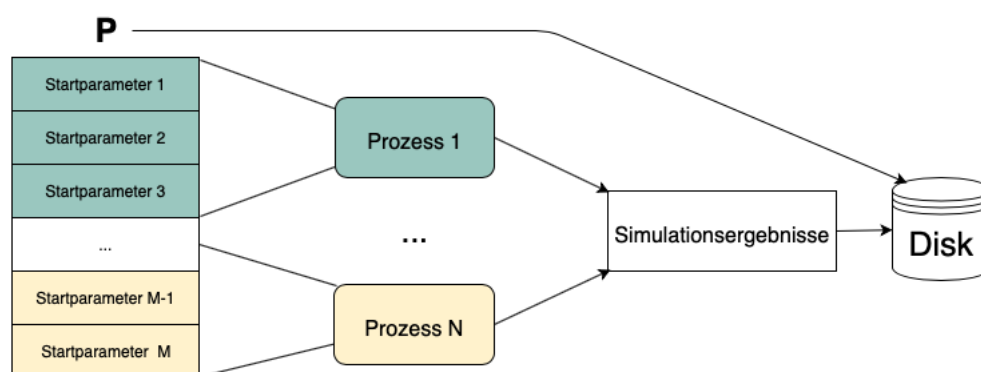


Abbildung 4.1: Schematischer Aufbau des Generators

## **4.2 Wahl der Eingangsparameter**

## **4.3 Training-, Validation- und Testdatensatz**

## **5 Neuronales Netz**

### **5.1 Architektur**

### **5.2 Lossfunction**

### **5.3 Optimizer**

## **6 Ergebnisse und Diskussion**

### **6.1 Lernprozess**

### **6.2 Ergebnisse und statistische Auswertung**

### **6.3 Fazit**



## Literaturverzeichnis

- [Bec] BECK, Fabian: *Backpropagation*. <http://www.neuronalesnetz.de/backpropagation1.html>
- [Beu11] BEUCHER, Ottmar: *Übungsbuch Signale und Systeme Lösungsband zum Lehrbuch Signale und Systeme - Theorie, Simulation, Anwendung*. 2011
- [Cro] CROCE, Gianluca: *The Small Angle X-ray Scattering Technique: An Overview*. – <http://people.unipmn.it/gcroce/download/theory.pdf>
- [Fie82] FIENUP, J. R.: Phase retrieval algorithms: a comparison. In: *Appl. Opt.* 21 (1982), Aug, Nr. 15, 2758–2769. <http://ao.osa.org/abstract.cfm?URI=ao-21-15-2758>
- [GL94] GROPP, William ; LUSK, Ewing: *An Introduction to MPI - Parallel Programming with the Message Passing Interface*. (1994)
- [GNHB08] GALLAGHER, Thomas A. ; NEMETH, A J. ; HACEIN-BEY, Lotfi: An introduction to the Fourier transform: relationship to MRI. In: *AJR. American journal of roentgenology* 190 5 (2008), S. 1396–405
- [Kla13] KLAKOW, Dietrich: *Grundlagen der Signalverarbeitung*. (2013)
- [Kri07] KRIESEL, David: *Ein kleiner Überblick über Neuronale Netze*. 2007 <http://www.dkriesel.com>
- [KRM<sup>+</sup>18] KLUGE, Thomas ; RÖDEL, Melanie ; METZKES, Josefine ; PELKA, Alexander ; GARCIA, Alejandro L. ; PRENCIPE, Irene ; REHWALD, Martin ; NAKATSUTSUMI, Motoaki ; MCBRIDE, Emma E. ; SCHÖNHERR, Tommy ; GARTEN, Marco ; HARTLEY, Nicholas J. ; ZACHARIAS, Malte ; ERBE, Arthur ; GEORGIEV, Yordan M. ; GALTIER, Eric ; NAM, Inhyuk ; LEE, Hae J. ; GLENZER, Siegfried ; BUSSMANN, Michael ; GUTT, Christian ; ZEIL, Karl ; RÖDEL, Christian ; HÜBNER, Uwe ; SCHRAMM, Ulrich ; COWAN, Thomas E.: *Observation of ultrafast solid-density plasma dynamics using femtosecond X-ray pulses from a free-electron laser*. 2018
- [QR11] QUIZA R., Davim J.: *Computational Methods and Optimization. Machining of Hard Mate-*

rials. 190 5 (2011), S. 1396–405

[Sch] SCHLESINGER, Dmitrij: *Bildverarbeitung: Fourier-Transformation*

[Wik18] WIKIPEDIA: *Backpropagation* — *Wikipedia, The Free Encyclopedia*. <http://de.wikipedia.org/w/index.php?title=Backpropagation&oldid=181456626>, 2018. – [Online; accessed 04-October-2018]

[Wis] WISSENSCHAFT, Spektrum der: *Plasma*

[Zac17] ZACHARIAS, Malte: *MODEL-DRIVEN PARAMETER RECONSTRUCTIONS FROM SMALL ANGLE X-RAY SCATTERING IMAGES*. <http://dx.doi.org/10.5281/zenodo.1208410>. Version: November 2017

## **Danksagung**

Die Danksagung...

## **Erklärungen zum Urheberrecht**

Hier soll jeder Autor die von ihm eingeholten Zustimmungen der Copyright-Besitzer angeben bzw. die in Web Press Rooms angegebenen generellen Konditionen seiner Text- und Bildübernahmen zitieren.