

Untersuchung von Störstellen in Quasikristallen unter Zuhilfenahme von neuronalen Netzen

Bachelorarbeit aus der Physik

Vorgelegt von

Ali Döner

12.09.2019

Institut für Theoretische Physik I
Friedrich-Alexander-Universität Erlangen-Nürnberg



Betreuer: Prof. Dr. M. Schmiedeberg

Zusammenfassung

Quasikristalle verfügen im Gegensatz zu periodischen Kristallen über keine Einheitszelle, dennoch weisen sie langreichweitig eine geordnete Struktur auf. Diese Ordnungsstruktur kann in Quasikristallen jede diskrete Rotationssymmetrie annehmen. Während Defekte in periodischen Kristallen für gewöhnlich mit dem bloßen Auge zu identifizieren sind, ist die visuelle Sinneswahrnehmung bei der Lokalisierung von Defekten in quasikristallinen Strukturen oft überfordert.

Ziel dieser Arbeit ist es, mit dem maschinellen Lernen einen Rahmen zur Untersuchung von Störstellen in Quasikristallen zu schaffen. Dafür werden Dislokationen, welche einen Defekt darstellen, in dodekagonalen Quasikristallen mit Hilfe von neuronalen Netzwerken analysiert, um sie nach ihrer Art und ihrer Position zu klassifizieren. Im ersten Teil der Arbeit wird mit der Einführung des theoretischen Hintergrunds darauf hingearbeitet, dodekagonale Quasikristalle als Rohdaten für das neuronale Netzwerk zu erstellen und aufzubereiten. Im zweiten Teil erfolgt eine Vorstellung des Aufbaus und des prinzipiellen Arbeitsablaufs von neuronalen Netzen als Teilgebiet des Deep Learnings. Im Anschluss werden unter Zuhilfenahme geeigneter Methoden neuronale Netze als Modelle instanziiert, welche die Dislokationsrichtung und die Position der Dislokation in Trainingsdaten erkennen, um aussagekräftige Vorhersagen für neue Daten zu treffen. Die Klassifikation nach der Art der Dislokation erfolgt mit einer Erfolgsrate von 100 % fehlerfrei. Um die Vorhersage der Position einer Dislokation weiterführend zu bewerten, wird neben der Korrektklassifizierungsrate eine Metrik ausgegeben, die den mittleren euklidischen Abstand zwischen tatsächlicher Position und der Ausgabe des Modells wiedergibt. Das trainierte neuronale Netz erfüllt mit einer Erfolgsrate von über 87 % und einem mittleren euklidischen Abstand von 0.13 ± 0.11 Längeneinheiten zwischen Zielwert und Vorhersage auch diese Problemstellung sehr gut.

Inhaltsverzeichnis

1	Quasikristalle	3
2	Störstellen und der Burgers-Vektor	4
3	Erstellen von Quasikristallen mit wohldefinierten Störstellen	6
3.1	Das Laserpotential zur Erzeugung von Quasikristallen	6
3.2	Einbau von Störstellen durch den Phasenanteil	7
3.3	Teilchenpositionen	9
4	Neuronale Netzwerke	11
4.1	Machine Learning, Deep Learning und klassische Programmierung	11
4.2	Einführung und Aufbau des neuronalen Netzes	12
5	Quasikristalle als Datensätze für das neuronale Netz	14
6	Klassifikation der Art des Burgers-Vektors	16
6.1	Vorbereitung der Rohdaten	16
6.2	Initialisierung und Training des neuronalen Netzes	17
6.3	Evaluation des trainierten neuronalen Netzes	18
7	Klassifikation der Position der Dislokation	20
7.1	Vorbereitung der Rohdaten	20
7.2	Training und Evaluation des neuronalen Netzes	21
7.3	Vorhersagen des Modells für Eingabedateien mit zwei Dislokationen	25
8	Ausblick: Klassifikation von weiteren Defekten	26
	Literaturverzeichnis	27
	Eidesstattliche Erklärung	28

1 Quasikristalle

Die Entdeckung von quasikristallinen Strukturen im Jahre 1982 stellt einen Wendepunkt in der wissenschaftlichen Auseinandersetzung mit geordneten Strukturen dar [1][2]. Shechtmans Beobachtungen an einer Aluminiumlegierung bilden damit einen Meilenstein in der Kristallografie und wurden 2011 mit einem Nobelpreis honoriert. Dieser Meilenstein erweist sich heute als Ausgangspunkt jahrzehntelanger intensiver experimenteller und theoretischer Grundlagenforschung [3]. Quasikristalle finden aufgrund ihrer Symmetrie-Eigenschaften in interdisziplinärer Vielfalt Verwendung - ob in der Festkörperphysik, in der Materialforschung oder in der Mathematik. Um einen Einblick in die Besonderheiten von Quasikristallen zu gewinnen, werden diese in diesem Abschnitt im Kontrast zu periodischen Kristallen eingeführt.

Quasikristalle werden als solche bezeichnet, da sie zwar in Übereinstimmung mit periodischen Kristallen über eine Fernordnung und eine Symmetrie verfügen, in Abgrenzung zu periodischen Kristallen aber nicht aus sich wiederholenden Einheitszellen bestehen. Ein Quasikristall besitzt also keine einfache Translationssymmetrie, birgt aber dennoch langreichweitige, geordnete Strukturen [4]. Tatsächlich ist die fehlende Existenz von Einheitszellen mit erstaunlichen Rotationssymmetrien verbunden: Quasikristalle können jede diskrete Rotationssymmetrie annehmen und sind nicht wie periodische Kristalle auf maximal sechszählige Drehachsen beschränkt [5]. In dieser Arbeit steht allerdings nicht die Symmetrie der Kristalle im Vordergrund, sondern die Existenz und Lokalisierung sogenannter Störstellen, die die symmetrische Ordnung brechen können.

2 Störstellen und der Burgers-Vektor

In der Natur kommen Störstellen in Kristallen sehr häufig vor und beeinflussen die Materialeigenschaften und die Stabilität des Festkörpers weitgehend. Sie sind im Sprachgebrauch daher auch als Unreinheiten etabliert. Einen mathematischen Ansatz zur Beschreibung von Art und Ausmaß einer Dislokation liefert die Einführung von zusätzlichen Freiheitsgraden für Quasikristalle [6]. Dieser Ansatz folgt aus der Einteilung von Quasikristallen in Kristallklassen (lokale Isomorphismen) und beschreibt die Existenz von Anregungen phasonischer und phononischer Natur [4]. Phononen lassen sich sehr gut als Anregungen des Systems in Form von Gitterschwingungen veranschaulichen. Sie nehmen Einfluss auf die Elastizität des Quasikristalls und können beispielsweise in Folge von thermischer Anregung entstehen. Phasonen hingegen lassen sich nur schwierig anschaulich darstellen und werden durch Translationen in einem höherdimensionalen Hyperraum erklärt [6]. Phasonische Anregungen äußern sich durch sogenannte phasonische Flips, in denen einzelne Gitteratome einer Umordnung unterliegen und ihre Position ändern [7].

Beschrieben wird die Topologie einer Dislokation durch den Burgers-Vektor. Im Realraum erhält man diesen Vektor, indem man zunächst den Bereich um eine Störstelle mit einer geschlossenen Kurve umrundet (vgl. Abbildung 1 (a)). Im Anschluss werden die identischen Schritte an einem Kristall ohne Dislokation durchgeführt (vgl. Abbildung 1 (b)). Aus der Differenz von Startpunkt und Endpunkt errechnet sich dann der Burgers-Vektor [4].

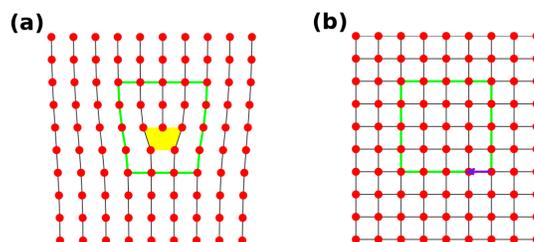


Abbildung 1: Geometrischer Ansatz zur Bestimmung des Burgers-Vektors, Bilder aus [4].

Mathematisch lässt sich das Abtasten des Randbereichs einer Störstelle durch ein geschlossenes Integral beschreiben. Betrachtet man ein phononisches Verschiebungsfeld $\vec{u}(\vec{r})$, so ergibt sich die resultierende totale Verschiebung zu

$$\Delta\vec{u} = \oint d\vec{u}(\vec{r}). \quad (1)$$

Die Folgen dieser Verschiebung sind in Abbildung 2 stellvertretend an einem oktagonalen Quasikristall veranschaulicht: Wird eine Störstelle lediglich durch ein von Phononen erzeugtes Verschiebungsfeld beschrieben, so entsteht neben der eigentlichen, gelb markierten Fehlstelle, eine Lücke in der topologischen Komposition des Quasikristalls (grün). Diese Lücke ist der fehlenden Translationssymmetrie des Quasikristalls geschuldet und es gibt in der Folge keine mögliche Parkettierung, die die Symmetrie erhält und die Lücke erschließt (d). Durch die zusätzlichen phasonischen Freiheitsgrade kann diese Lücke jedoch trotzdem erschlossen werden. Ein zweites, phasonisches Verschiebungsfeld $\vec{w}(\vec{r})$ kann also in Kombination mit dem phononischen Anteil $\vec{u}(\vec{r})$ die Stetigkeit des Dichtefeldes um die Störstelle erhalten und so für eine erschlossene Parkettierung sorgen (e). Die totale Verschiebung $\Delta\vec{w}$ erhält man analog zu Gleichung (1) durch ein geschlossenes In-

tegral um die Dislokation. Phononische und phasonische Anteile an einer Störstelle werden beide im Burgers-Vektor repräsentiert. Er ergibt sich folglich zu $\vec{B} = (\Delta\vec{u}, \Delta\vec{w})$ [8][4][9].

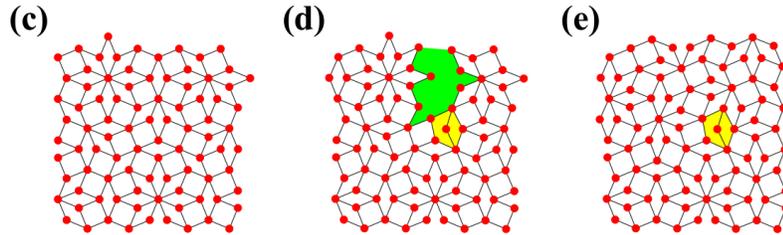


Abbildung 2: Veranschaulichung des phononischen und phasonischen Anteils einer Störstelle im Quasikristall, Bilder aus [4].

Die Schwierigkeit in der Berechnung des Burgers-Vektors besteht vor allem darin, eine geeignete Parametrisierung für die möglichen Symmetrien der Quasikristalle zu finden, welche eine kontinuierliche Parkettierung des umliegenden Bereichs sicherstellt. Die Komponenten des Burgers-Vektors variieren in Abhängigkeit der vorliegenden Kristallsymmetrie.

3 Erstellen von Quasikristallen mit wohldefinierten Störstellen

Zielsetzung dieser Arbeit ist es, Modelle zu entwickeln, welche in der Lage sind zu jeder Eingabe eines Quasikristalls eine Ausgabe erzeugen, die die Position der Dislokation oder die Komponenten des Burgers-Vektors angibt. In den nachfolgenden Abschnitten wird vor diesem Hintergrund das Erstellen der Quasikristalle als Rohdaten für das Modell aufgearbeitet.

3.1 Das Laserpotential zur Erzeugung von Quasikristallen

Zur grafischen Darstellung einer quasikristallinen Struktur wird ein Potentialfeld in einer Ebene verwendet. Bei dem Potential handelt es sich um ein Laserpotential, resultierend aus der Interferenz von N gleich starken, symmetrisch positionierten Laserstrahlen [4][10]. Das Potential am Ort $\vec{r} = (x, y)$ errechnet sich zu

$$V(\vec{r}) = -\frac{V_0}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \cos[(\vec{G}_i - \vec{G}_j) \cdot \vec{r} + \varphi_i(\vec{r}) - \varphi_j(\vec{r})]. \quad (2)$$

Die Komponenten von φ repräsentieren dabei den Phasenanteil und bei \vec{G} handelt es sich um die reziproken Gittervektoren. Die Parametrisierung der Gittervektoren erfolgt sternförmig und symmetrisch, sodass $\vec{G}_i = G(\cos[2\pi i/N], \sin[2\pi i/N])$.

Der Phasenanteil φ wird zunächst vernachlässigt und mit Null gleichgesetzt. Er ist ausschlaggebend für das Platzieren von Störstellen und wird daher im nächsten Abschnitt genauer analysiert. Die Symmetrie des Quasikristalls lässt sich nach Gleichung (2) direkt über die Anzahl N an Laserstrahlen festlegen. In Abbildung 3 sind verschiedene Realisierungen von Quasikristallen dargestellt. Im weiteren Verlauf werden dodekagonale Quasikristalle mit $N = 12$ verwendet.

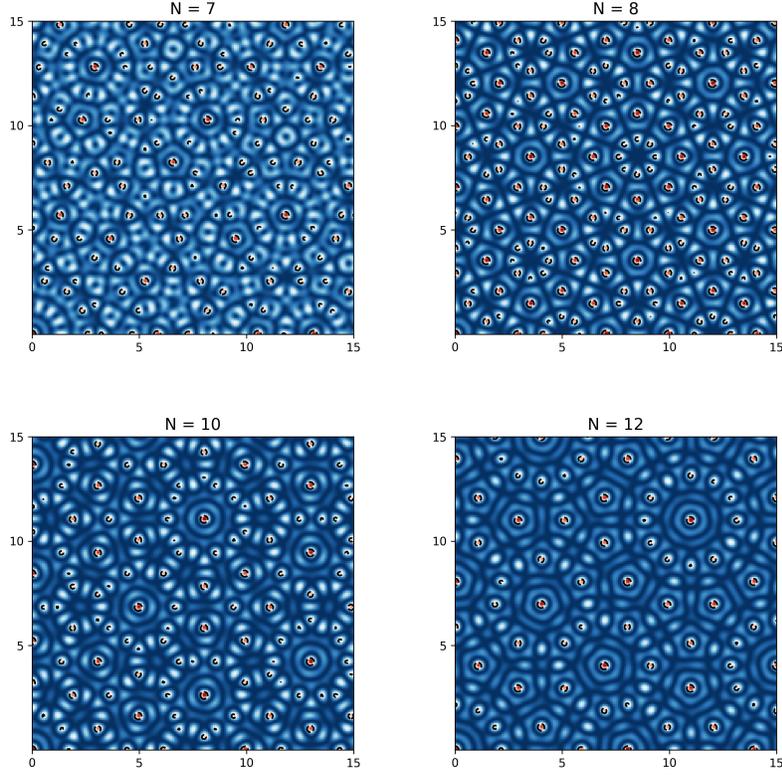


Abbildung 3: Quasikristalle in Abhängigkeit von der Anzahl N an Laserstrahlen aus dem Laserpotential aus Gleichung (2).

3.2 Einbau von Störstellen durch den Phasenanteil

In diesem Abschnitt wird auf die Platzierung einer Störstelle in dem Symmetrie-Muster des Quasikristalls eingegangen. Wie im vorherigen Abschnitt eingeführt, ist hierfür der Phasenanteil φ ausschlaggebend. Dieser ergibt sich nach [9][4] zu

$$\varphi_j(\vec{r}) = \vec{G}_j \vec{u} + \vec{G}_{kj \bmod N} \vec{w} + C. \quad (3)$$

Die Vektoren \vec{G}_j beschreiben weiterhin die reziproken Gittervektoren. Die Vektoren $\vec{u} = (u_x, u_y)$ und $\vec{w} = (w_x, w_y)$ bilden den phononischen und phasonischen Anteil der Verschiebung und bei k handelt es sich um eine Konstante. Im Allgemeinen ist die Berechnung der Verschiebungsfelder $\vec{u}(\vec{r})$ und $\vec{w}(\vec{r})$ nicht trivial. Für die explizite Darstellung und Herleitung der Komponenten von $\vec{u}(\vec{r})$ und $\vec{w}(\vec{r})$ wird daher auf die Arbeit von Piali De und Robert Pelcovits verwiesen [8]. Zur Parametrisierung von $\vec{u}(\vec{r})$ und $\vec{w}(\vec{r})$ kann aber ausgenutzt werden, dass die Komponenten $\Delta\vec{u}$ und $\Delta\vec{w}$ des Burgers-Vektors wohldefiniert sind. Dafür wird an dieser Stelle ein Verschiebungsfeld $\vec{D}(\vec{r})$ eingeführt, welches aus den beiden Vektorfeldern $\vec{u}(\vec{r})$ und $\vec{w}(\vec{r})$ zusammengesetzt ist, sodass $\vec{D}(\vec{r}) = (\vec{u}(\vec{r}), \vec{w}(\vec{r}))$. Zur Bestimmung der Komponenten von $\vec{D}(\vec{r})$ wird noch eine geometrische Korrelation zwischen dem Burgers-Vektor \vec{B} und dem Verschiebungsfeld \vec{D} benötigt. Diese liefert beispielsweise ein Arkustangens-Term, konkret $\arctan(\frac{y}{x})$. Dieser ist für $(x, y) \neq (0, 0)$ zwar als

Verkettung stetiger Funktionen stetig, allerdings existiert keine stetige Fortsetzung auf ganz \mathbb{R}^2 . Tatsächlich findet man für jedes $c \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ eine Nullfolge $(x_n, y_n) \xrightarrow{n \rightarrow \infty} (0, 0)$, sodass $\arctan(\frac{y_0}{x_0})$ für $n \rightarrow \infty$ gegen c konvergiert. In Abhängigkeit des Vorzeichens von y_0 äußert sich die Mehrdeutigkeit des Arkustangens. Diese wird den Anforderungen an die topologische Struktur einer Dislokation gerecht [8]. Es ergibt sich nach [4]:

$$2\pi D_j(x, y) = B_j \arctan\left(\frac{y}{x}\right) + C. \quad (4)$$

Ein Burgers-Vektor $\vec{B} = (\vec{b}_j^{(u)}, \vec{b}_j^{(w)})$ ist für $N = 12$ vierkomponentig und errechnet sich analog zu den Gittervektoren sternförmig zu

$$\begin{aligned} \vec{b}_j^{(u)} &= (u_r \cos[\pi(2j-1)/N], u_r \sin[\pi(2j-1)/N]), \\ \vec{b}_j^{(w)} &= (w_r \cos[k\pi(2j-1)/N], w_r \sin[k\pi(2j-1)/N]), \end{aligned} \quad (5)$$

wobei $j = 1, \dots, N$, $u_r = \sqrt{2}/(3 - \sqrt{3})$ und $w_r = -u_r/(1 + \sqrt{2})$ [4]. Um das Verschiebungszentrum an einer bestimmten Position $P(x, y)$ zu platzieren, werden die Komponenten des Arkustangens jeweils um x beziehungsweise um y verschoben. Im Allgemeinen kommt es wegen der periodischen Randbedingungen in Simulationen zu Problemen, da das aus der Dislokation resultierende Verschiebungsfeld zum Rand der Box divergiert. Stellt man sich den Verschiebungskern in Anlehnung an die elektrische Ladung von Teilchen als topologische Ladung vor, so kann diese durch einen zweiten, antiparallelen Burgers-Vektor, welcher symmetrisch zum ersten platziert wird, auf einen lokalen Bereich begrenzt werden (vgl. Abbildung 4 links).

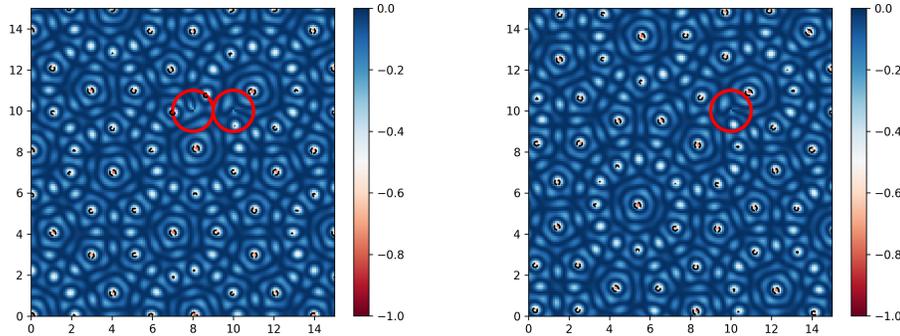


Abbildung 4: Dodekagonaler Quasikristall mit zwei antiparallelen Burgers-Vektoren (links) und einer einzelnen Störstelle (rechts).

Zur späteren Klassifikation ist allerdings nicht von Relevanz, ob das Zentrum der Störstelle ein Verschiebungsfeld induziert, das sich auch zum Rand des Bildes fortsetzt. Exemplarisch zeigt Abbildung 4 zusätzlich die Auswirkungen einer einzelnen Störstelle auf die Struktur des Kristalls.

3.3 Teilchenpositionen

Da die Quasikristalle durch das Laserpotential aus Gleichung (2) gegeben sind, bewegt sich der Wertebereich des Potentialfeldes lediglich im Intervall $[-1, 0]$. Das Potentialfeld kann auch als Potential-Landschaft mit ausgeprägten Minima und Maxima aufgefasst werden. Die Positionen der lokalen Extremalstellen können durch die Implementierung einer Suchfunktion ermittelt werden. Im Allgemeinen folgt die Verteilung der lokalen Minima größtenteils der Symmetrie des Quasikristalls, sodass sich die erfassten Tupel als Teilchenpositionen der Gitteratome identifizieren lassen. In Abbildung 5 ist ein dodekagonaler Quasikristall mit Teilchen an den lokalen Minima dargestellt. Allerdings können nicht alle durch die Suchfunktion erfassten lokalen Minima als Teilchen verifiziert werden, da es auch einige, in der Regel flachere, Extremalstellen zwischen den Symmetriestrukturen gibt, welche bei niedrigen Dichten eventuell keine Gitteratome darstellen. Teilchen an diesen Positionen sind in Abbildung 5 gelb dargestellt und werden durch eine Hilfsfunktion herausgefiltert.

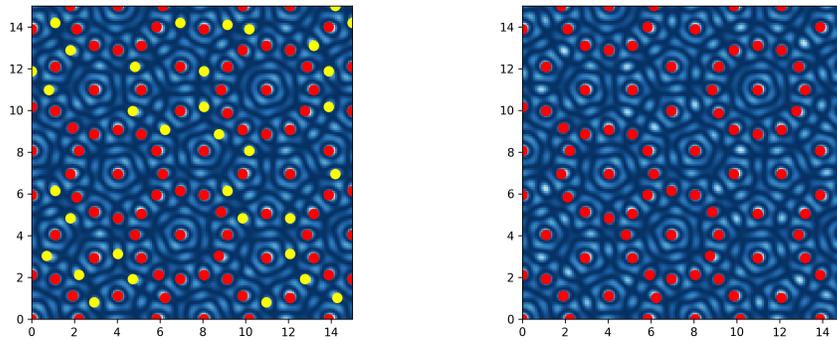


Abbildung 5: Bestimmung der Teilchenpositionen im dodekagonalen Quasikristall durch die lokalen Minima (links) und mit Berücksichtigung der Symmetrie (rechts).

Die Kennzeichnung der Teilchenpositionen birgt den Vorteil, das Verschiebungsfeld, welches von einer Dislokation hervorgerufen wird, sichtbar zu machen. In Abhängigkeit der Position und Art der Störstelle verschieben sich die umliegenden Teilchen durch den Einfluss der topologischen Ladung. Diese Verschiebung ist in Abbildung 6 dargestellt: Die blauen Teilchen stellen die Positionen im fehlerfreien dodekagonalen Kristall dar. Die roten Punkte kennzeichnen die Teilchen in Anwesenheit eines Verschiebungszentrums. Hier zeigt sich die Auswirkung von zwei Störstellen mit entgegengesetztem Burgers-Vektor noch einmal konkreter. Während die Kristallsymmetrie auf dem linken Bild durch den Einbau von zwei Dislokationen weitestgehend erhalten bleibt, verschieben sich die Teilchen auf dem rechten Bild unter dem Einfluss einer einzelnen Störstelle deutlicher.

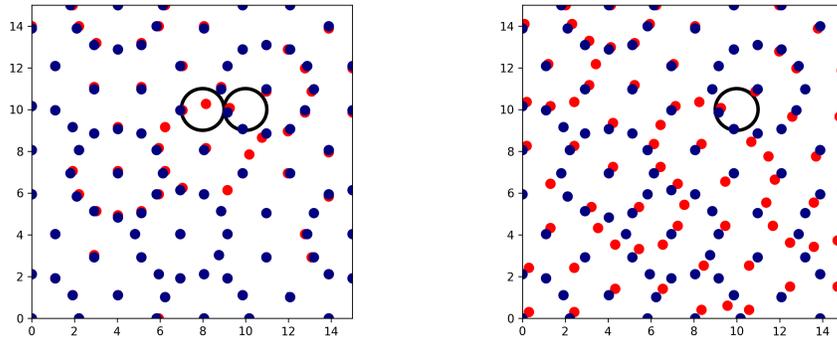


Abbildung 6: Verschiebung (rot) der Teilchenpositionen im dodekagonalen Quasikristall mit zwei antiparallelen Burgers-Vektoren (links) und einer einzelnen Störstelle (rechts).

4 Neuronale Netzwerke

Es erweist sich als äußerst schwierig, Störstellen in einem Quasikristall mit dem bloßen Auge zu erkennen. Sie zusätzlich nach ihrem Burgers-Vektor zu klassifizieren, ist ohne Anwendung der Fourier-Transformation nicht möglich. An dieser Stelle schafft die Anwendung von maschinellem Sehen Abfindung. Es ist sinnvoll, die Merkmalerkennung in der Kristallstruktur durch Deep Learning Modelle vorzunehmen. Nachdem die Rohdatenerzeugung vollständig vorgestellt worden ist, werden im Folgenden neuronale Netzwerke eingeführt und im Anschluss anhand der Funktion ihrer Bestandteile genauer vorgestellt.

4.1 Machine Learning, Deep Learning und klassische Programmierung

Um einen Einblick in den Arbeitsablauf und ein Gefühl für die Struktur der in dieser Arbeit genutzten Bilderkennungsmechanismen zu gewinnen, beschäftigt sich dieser Abschnitt mit den theoretischen Grundlagen des Deep Learnings. Dafür wollen wir das Deep Learning zunächst als Fachgebiet des Machine Learnings einführen und einen systematischen Kontrast zur klassischen Programmierung ziehen: Konventionell werden die Herangehensweisen und Regeln zum Lösen einer Aufgabe von dem Programmierer wohlbestimmt und im Anschluss in Form eines festen Regelsatzes in ein Programm implementiert. Das Programm liefert dann auf Grundlage der Daten und der Regeln die Antworten. Beim Machine Learning hingegen werden die Antworten gemeinsam mit den Rohdaten als Eingabe in das Programm gegeben, das wiederum Regeln als Ausgabe formuliert. Schematisch ist die Klassifikation in Abbildung 7 dargestellt. Als Anforderung an die resultieren-

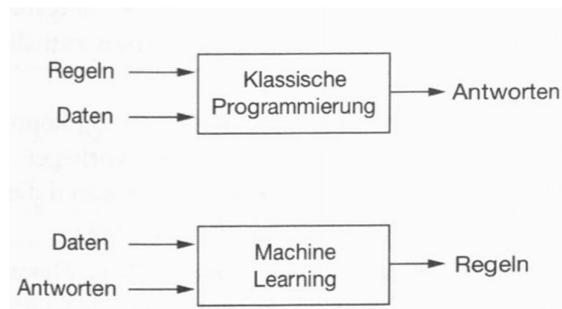


Abbildung 7: Systematik und Zielsetzung von Machine Learning im Kontrast zu der klassischen Programmierung, Bild aus [11].

den Regeln stellt man die Anwendbarkeit auf neue, dem Programm bisher unbekannte Daten und als Ergebnis erwartet man eigenständige Antworten. Da die von dem Machine-Learning-System bestimmten Regeln zunächst allein die gefundenen Korrelationen zwischen den eingegebenen Rohdaten und den Antworten repräsentieren, ist es sinnvoll, der Größenordnung und der Vielseitigkeit der Rohdaten einen besonderen Wert zuzusprechen. Ziel ist es also, dem System möglichst relevante Beispiele bereitzustellen, die zur Lösung der Aufgabenstellung hinreichend sind. Diese Beispiele werden von dem System zur Automatisierung genutzt, in dem nach einer statistischen Struktur gesucht wird [11]. Wie diese statistischen Strukturen gefunden werden und welchen Kontrollparametern sie unterliegen, wird im Folgenden Abschnitt vorgestellt.

4.2 Einführung und Aufbau des neuronalen Netzes

Im Rahmen des Deep Learnings werden Eingabedaten in sinnvolle Ausgaben transformiert. Deep-Learning-Algorithmen setzen dabei das Erlernen aufeinanderfolgender Repräsentations-Schichten als Schwerpunkt [11]. Diese Schichten (Layer genannt) beinhalten zunehmend sinnvollere Repräsentationen der Eingabedaten und werden insbesondere gleichzeitig von einem Modell erlernt, welches man als neuronales Netz bezeichnet. Ein neuronales Netzwerk stellt also eine Abfolge von aneinandergereihten Layern dar, welche jeweils eine einfache Transformation der ursprünglichen Eingabedaten vollziehen. Die übermittelte Information wird durch diesen Prozess gefiltert und immer aussagekräftiger im Bezug auf die Lösung des gestellten Problems. Die Anzahl der beitragenden Layer wird als Tiefe des Modells bezeichnet.

In diesem Abschnitt wird qualitativ dargelegt, wie das neuronale Netz Repräsentationen von Daten erlernt und wie dieser Lernvorgang kontrolliert und validiert werden kann. Den Ausgangspunkt stellt dabei Abbildung 8 dar: Wir betrachten eine Eingabe X , welche verschiedene Layer durchläuft und dabei transformiert wird. Nach der konsekutiven Datentransformation wird dann eine Vorhersage Y' getroffen. Steuerparameter für diese Abfolge sind die Gewichtungen der Layer. Die Parametrisierung eines neuronalen Netzes erfolgt also durch seine Gewichtungen [11]. Bei der Eingabe X handelt es sich um ein Sample - bei der zugehörigen Klasse Y um das Label.

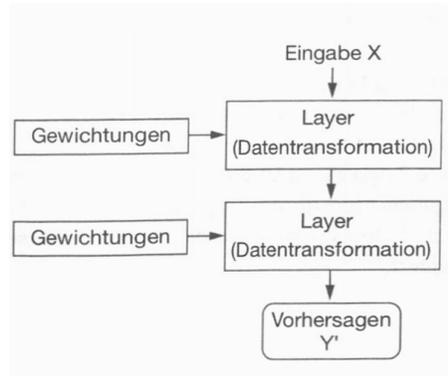


Abbildung 8: Schematische Darstellung der Parametrisierung eines neuronalen Netzes, Bild aus [11].

Die Zielsetzung ist in der Folge die Gewichtungen so zu modifizieren, dass das Modell eine sehr gute Vorhersage trifft.

Um das Modell zu bewerten und um die Gewichtungen zu optimieren, ist ein Maß für den Erfolg des Algorithmus notwendig. Dieses Maß wird durch die Verlust- bzw. Fehlerfunktion bestimmt. Sie berechnet, wie sehr die vorhergesagten Ausgaben des Modells von den tatsächlichen Zielwerten abweichen und speichert das Ergebnis in dem Verlustscore. Dieser Verlustscore kann nun für jeden Durchlauf ausgegeben werden und darüber hinaus als Feedback-Signal zur Anpassung der Gewichtungen zur Verfügung gestellt werden (vgl. Abbildung 9) [11].

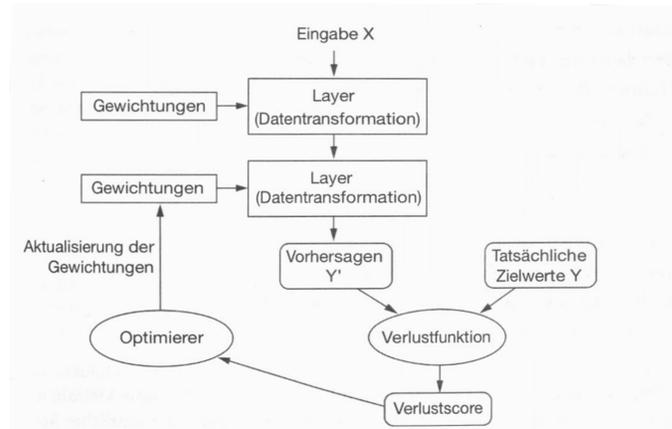


Abbildung 9: Prinzipieller Arbeitsablauf zur Anpassung der Gewichtungen des neuronalen Netzes.

Das Feedback-Signal hat die Aufgabe, die Werte der Gewichtungen so anzupassen, dass der Verlustscore für die aktuelle Eingabe abnimmt [11]. Die technische Umsetzung erfordert für das Optimieren des Verlustscores einen Algorithmus - den sogenannten Optimierer. Dieser bedient sich bei der Optimierung des Verlustscores dem sogenannten Gradientenabstiegsverfahren. Als Backpropagation-Algorithmus wird das Minimum der Fehlerfunktion durch Anpassung der Gewichtungen des neuronalen Netzes entlang des absteigenden Gradienten berechnet [12]. Das neuronale Netz wird nun in einer Trainingsschleife mit jedem erfassten Beispiel so angepasst, dass der Verlustscore minimal wird. Nach Durchlaufen der Trainingsschleife minimieren also die aus den Gewichtungen gewonnenen Parameter die Verlustfunktion und man spricht von einem trainierten neuronalen Netzwerk. Die Vorhersagen Y' stimmen jetzt so gut wie möglich mit den tatsächlichen Zielwerten Y überein. In Analogie zu dem Maß für den Optimierungsprozess dient die Korrekt-klassifizierungsrate einem Maß dafür, in welchem Verhältnis das neuronale Netz der Eingabe X den tatsächlichen Wert Y zuordnet. Sie wird in Prozent angegeben und liefert je nach Wahl der Komponenten des neuronalen Netzes einen anderen Wert.

An dieser Stelle äußert sich, warum Machine Learning im Sprachgebrauch als Ingenieurwissenschaft etabliert ist: Die Korrekt-klassifizierungsrate ist abhängig von zahlreichen Parametern, die allesamt keine überschaubare Struktur aufweisen. Zu den Parametern zählen unter anderem die Aufbereitung der Rohdaten, die Tiefe des Modells (Art und Anzahl der Layer), die Wahl des Trainingsprozesses, die Wahl der Verlustfunktion und die Wahl des Optimierers. Die Keras-Bibliothek bietet in ihrer Funktion als Modell-Bibliothek die Möglichkeit einzelne Komponenten eines neuronalen Netzes (wie Layer oder Optimierer) als Bausteine zu verwenden. Keras wird in der Forschung vor allem für Problemstellungen genutzt, welche in den Bereich der maschinellen Sinneswahrnehmung fallen. Die in der Keras-Bibliothek implementierten Algorithmen eignen sich daher gut zur Klassifikation der Störstellen in den vorgestellten Quasikristallen. Als High-Level Programmierschnittstelle bietet Keras keine Low-Level-Operationen, die für den mathematischen Umgang mit den Eingabedaten notwendig sind. Hierfür wird das Backend TensorFlow (<https://www.tensorflow.org/>) verwendet.

5 Quasikristalle als Datensätze für das neuronale Netz

In Fortsetzung an die in Kapitel 3 eingeführte Vorgehensweise zum kontrollierten Platzieren von Störstellen in Quasikristallen wird in diesem Abschnitt veranschaulicht, welche Datensätze für das neuronale Netz Verwendung finden und wie diese Datensätze aufgebaut sind. Das aus der Potentialfunktion (2) resultierende Kristallgitter bildet als Sample eine geeignete Eingabedatei für das Modell. Dieses Gitter hat eine Höhe und Breite von 15 Längeneinheiten und ist in der Shape (150, 150) gegeben. Die Shape des Potentialgitters entspricht der Pixelzahl des Bildes und ist in einem Tensor *Images* gespeichert. Dieser Tensor ist im Gegensatz zu dem Potentialgitter dreidimensional, da er im ersten Eintrag die Samplenummer speichert. Ein Datensatz mit N Bildern wird in der Shape (N , 150, 150) gespeichert. Zusätzlich ist jeder Kristall mit einer Dislokation ausgezeichnet. Das Zentrum der Dislokation wird dabei bei der Erstellung jedes einzelnen Samples zufällig generiert, zur Berechnung des Potentialgitters verwendet und im Anschluss in dem Datensatz *Burgers_pos* gespeichert. Es wird im Vorfeld allerdings ausgeschlossen, dass sich das Verschiebungszentrum am Randbereich der Box befindet. Da die Position zwei Komponenten hat, hat der Tensor *Burgers_pos* für den eingeführten Eingabedatensatz eine Shape von (N , 2). Ebenso variiert die Art bzw. die Form der Störstelle in jedem Sample. Insgesamt werden sechs verschiedene Dislokationsrichtungen durch sechs einfache Burgers-Vektoren unterschieden. Die Parametrisierung dieser Vektoren ist durch Gleichung (5) gegeben. Auch hier erfolgt die Zuweisung für jede Eingabedatei zufällig, sodass jedes Sample ein Verschiebungszentrum an einer willkürlichen Position mit willkürlichem Burgers-Vektor besitzt. Die Art des Burgers-Vektors wird als Integer von 0 bis 5 im Datensatz *Burgers_type* festgehalten. Dieser Tensor hat in der Folge eine Shape von (N , 1). Zur Veranschaulichung sind die verwendeten Datensätze in Abbildung 10 dargestellt.

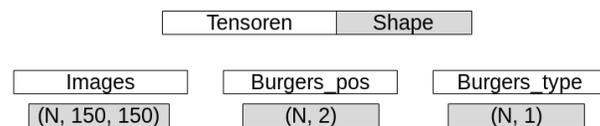


Abbildung 10: Veranschaulichung der genutzten Datensätze für N Bilder.

Zuletzt werden die Datenpakete noch in Bezug auf ihre Funktionalität in drei Teilmengen unterteilt: in eine Trainingsdatenmenge, eine Validierungsdatenmenge und eine Testdatenmenge. Die Trainingsdatenmenge wird als Eingabe genutzt, um das neuronale Netz mit Beispielen zu trainieren. Die Funktionalität der Validierungsdatenmenge besteht darin, den Fortschritt des neuronalen Netzes nach jedem Trainingsdurchlauf zu bewerten. Abschließend wird der Trainingsprozess des neuronalen Netzes anhand der noch unbekanntesten Testdatenmenge ausgewertet. Die Einteilung erfolgt jeweils durch die Kennzeichnungen „Train“, „Val“ und „Test“ und ist in Abbildung 11 illustriert.

Um ein statistisch repräsentatives Ergebnis zu erzielen, muss beim Aufbereiten der Datensätze darauf geachtet werden, dass die Trainings- und die Validierungsdatenmenge disjunkt sind, da das neuronale Netz ansonsten bereits Lösungen zu Samples erhält, ehe es vollständig trainiert

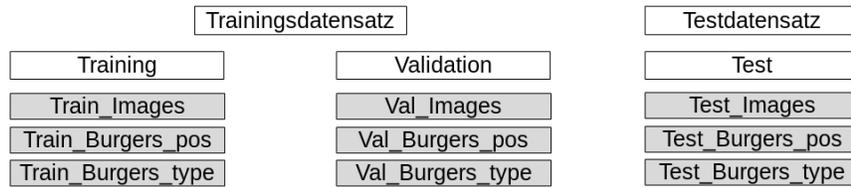


Abbildung 11: Einteilung der Datenpakete nach ihrer Funktionalität in Trainings- und Testdaten.

worden ist. Die Aussagekraft der Rohdaten ist für ein erfolgreich trainiertes Modell von zentraler Bedeutung. Sie beeinflusst die Qualität der Entscheidungsregeln, die zu einer Klassifizierung führen und damit auch die Qualität der gesammelten Ergebnisse. Sind die Entscheidungsregeln aussagekräftiger, so lassen sich die Ergebnisse auch besser auf unbekannte Daten verallgemeinern. Daher ist es von besonderem Interesse, möglichst viele voneinander unterscheidbare Samples zu generieren. Das gelingt durch einfache Kombinatorik: Wählt man für ein Quadrat mit 14 Längeneinheiten in x- und y-Richtung ganze Zahlen für die Initialisierung der Position des Verschiebungskerns, so ergeben sich zunächst insgesamt 196 Möglichkeiten für die Beschreibung des Quasikristalls. Wird zusätzlich berücksichtigt, dass es zu jedem Verschiebungskern genau sechs Möglichkeiten für einen Burgers-Vektor gibt, so ergeben sich insgesamt 1176 Möglichkeiten. Das sind deutlich zu wenige Realisierungen, um große Trainingsdatensätze zu erzeugen. Ist für ein Sample sowohl die Art der Dislokation, als auch seine Lokalisation identisch, so ist auch das errechnete Potentialfeld deckungsgleich.

Die Kongruenz der Bilder wird präventiv verhindert, indem die Initialisierung der Position der Störstelle als Fließkommazahl erfolgt und im Anschluss gerundet gespeichert wird. Für die Störstelle werden in Analogie zur Form des Potentialfeldes jeweils 140 Möglichkeiten in x- und in y-Richtung gewählt. Dadurch ver Hundertfacht sich die Zahl der voneinander unterscheidbaren Möglichkeiten für die Realisierung disjunkter Samples von 1176 auf 117600.

Nachdem alle technischen und theoretischen Grundlagen eingeführt worden sind, wird der vorgestellte Arbeitsablauf durch die Klassifikation der Dislokationen durchgeführt.

6 Klassifikation der Art des Burgers-Vektors

Als Erstes soll das neuronale Netz Quasikristalle, welche mit Störstellen ausgezeichnet sind, der Art ihrer Störstelle zuordnen. Es handelt sich hierbei um eine Single-Label-Mehrfachklassifikation, da jeder Eingabedatei genau eine Klasse zugeordnet wird. Insgesamt gibt es sechs verschiedene Klassen - jede ist stellvertretend für einen anderen Burgers-Vektor. Die Daten werden in eine Trainings-, eine Validierungs- und eine Testdatenmenge aufgeteilt und im Folgenden für das neuronale Netz aufbereitet.

6.1 Vorbereitung der Rohdaten

In der Trainingsdatenmenge sind 18300 Quasikristalle als Bilder mit einer Größe von 150 x 150 Pixeln im Tensor *Train_Images* gespeichert. Zusätzlich ist für jedes Sample in einem zweiten Tensor *Train_Burgers_type* die Initialisierung des Burgers-Vektors stellvertretend als Integer von 0 bis 5 gespeichert. Unter Einsatz der sogenannten Holdout-Methode werden die letzten 1000 Samples der Trainingsdatenmenge entnommen, um daraus die Validierungsdatenmenge *Val_Images* und den zugehörigen Datensatz *Val_Burgers_type* zu generieren. Die Validierungsdatenmenge ist notwendig, um den Fortschritt der Modifikation der Parameter des neuronalen Netzes zu kontrollieren, ehe das trainierte Modell mit dem unbekanntem Datensatz *Test_Images* evaluiert wird.

Der Tensor *Test_Images* der Testdatenmenge besteht aus weiteren 9500 Bildern. Die Vorhersagen des neuronalen Netzwerks werden dann mit den zugehörigen Labeln aus *Test_Burgers_type* verglichen. Bevor die Daten allerdings für den Trainingsprozess genutzt werden können, müssen sie in eine mit dem neuronalen Netz kompatible Form transformiert werden. Konkret werden die Einträge aus *Train_Burgers_type* und *Test_Burgers_type* dafür in einen One-Hot-Code umgewandelt. Die Listen enthalten zunächst einmal für jedes Bild eine Ziffer, welche die Art der Störstelle repräsentiert. Diese Information bleibt auch in der One-Hot-Kodierung erhalten, allerdings wird der Inhalt jedes Labels in einen Vektor in Binärform umgeschrieben. Die One-Hot-Kodierung wird auch als 1-aus-n-Code bezeichnet, da die Dimension des Vektors durch die Anzahl der verschiedenen Klassen n bestimmt ist und die Information des Labels durch eine 1 an der entsprechenden Stelle gekennzeichnet wird.

Zur Verdeutlichung wird der erste Eintrag von *Train_Burgers_type* in Abbildung 12 vor und nach der One-Hot-Kodierung aufgerufen. Vor dem Aufruf beinhaltet der erste Eintrag des Arrays den

```
Train_Burgers_type = Train_Burgers_type.astype(int)
Train_Burgers_type[0]
array([2])
Train_Burgers_type = to_categorical(Train_Burgers_type)
Train_Burgers_type[0]
array([0, 0, 1, 0, 0, 0])
```

Abbildung 12: Inhalt des ersten Labels der Trainingsdatenmenge *Train_Burgers_type* vor und nach der Transformation in One-hot-Kodierung.

Wert 2 - das erste Sample ist also mit dem dritten Burgers-Vektor initialisiert. Diese Information

bleibt auch nach der Umwandlung erhalten. An der zweiten Stelle ist der Vektor mit einer 1 gegeben, wohingegen alle anderen Einträge den Wert 0 aufweisen.

Zuletzt wird noch geprüft, ob Trainings- und Validierungsdatenmenge disjunkt sind.

6.2 Initialisierung und Training des neuronalen Netzes

Zur Durchführung des Trainingsprozesses wird sowohl die Tiefe des Modells, als auch die Struktur der einzelnen Layer festgehalten. Dies erfolgt unter Benutzung der Sequential-Methode der Keras-Bibliothek. Als Schichten des Netzes werden zwei Dense Layer verwendet, welche mit der Relu-Funktion aktiviert werden. Die Dense-Layer wenden eine Matrixmultiplikation und eine Addition auf die Eingabedatei an, sodass die Ausgabe durch

$$output = dot(W, input) + b \tag{6}$$

gegeben ist [11]. Bei diesen Transformationen handelt es sich um lineare Operationen. Um den Lösungsraum um eine Nichtlinearität zu erweitern, wird die Relu-Funktion als Aktivierungsfunktion gewählt [13]. Die Dimension des Lösungsraums beider Dense-Layer wird auf 16 festgelegt.

Ein höherdimensionaler Lösungsraum beschleunigt zwar den Fortschritt in der Merkmalerkennung des Modells, birgt aber gleichzeitig die Gefahr, dass eine sogenannte Überanpassung an die Trainingsdaten erfolgt. Ist das Modell zu gut an die Trainingsdaten angepasst, so erkennt es Merkmale, die spezifisch den Trainingsdaten zuzuordnen sind, sich aber nicht verallgemeinern lassen. Das letzte Layer wird mit einer Softmax-Funktion aktiviert, welche eine Ausgabe der Vorhersage in One-Hot-Kodierung ermöglicht. Bei der Softmax-Funktion handelt es sich um die normalisierte Exponentialfunktion und die Ausgabe gibt eine Wahrscheinlichkeitsverteilung über alle möglichen Burgers-Vektoren an. Die Auswahl der Ausgabe als Wahrscheinlichkeitsverteilung ermöglicht darüber hinaus die Wahl der kategorialen Kreuzentropie als Verlustfunktion. Diese generiert im Kontrast zur Methode der kleinsten Quadrate als Minimierungsfunktion deutlich genauere Ergebnisse [14]. Das Training erfolgt nun in mehreren Trainingsepochen, nach denen jeweils eine Modifikation der Parameter des neuronalen Netzes erfolgt. Um das Training beurteilen zu können, gibt das Modell nach jeder Epoche den Wert der Verlustfunktion und der Korrektklassifizierungsrate für die Trainings- und die Validierungsdatenmenge an. Eine mögliche Überanpassung lässt sich gut erkennen, wenn die Fehlerfunktion und das Erfolgskriterium für Trainings- und Validierungsdatenmenge im Trainingsprozess aufgezeichnet werden.

6.3 Evaluation des trainierten neuronalen Netzes

Zur Evaluation des Trainingsprozesses ist in Abbildung 13 die Korrektklassifizierungsrate in Abhängigkeit der Trainingsepoche aufgetragen.

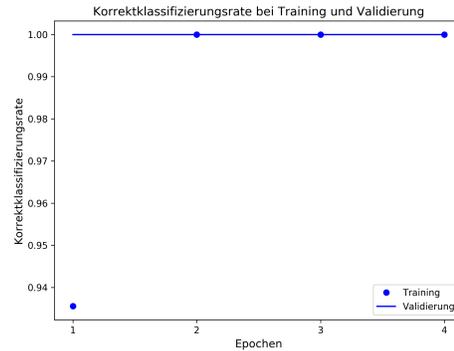


Abbildung 13: Korrektklassifizierungsrate in Abhängigkeit der Trainingsepoche für die Zuweisung des Burgers-Vektors.

Das neuronale Netz weist nach einer Trainingsepoche bereits eine Genauigkeit von 1.00 auf - es ordnet also nicht nur dem Trainingsatz *Train.Images* fehlerfrei die richtigen Labels zu, sondern auch der Validierungsmenge *Val.Images*.

Eine Anwendung auf den noch unbekanntem Testdatensatz *Test.Images* ergibt ebenfalls eine Treffgenauigkeit von 100 %. Der Wert der Verlustfunktion liegt bei $1.2 \cdot 10^{-7}$. Trifft man mit dem Modell nun eine Vorhersage für eine unbekanntem Eingabedatei, so eignet sich die ausgegebene Wahrscheinlichkeitsverteilung sehr gut, um jeder Bilddatei ein Säulendiagramm zuzuweisen. Wenn sich die Zuweisung des neuronalen Netzes mit dem tatsächlichen Wert aus *Test.Burgers.type* deckt, so wird das Diagramm in der Farbe blau abgebildet. Ist die Vorhersage fehlerhaft, so wird die Säule rot eingefärbt. Abbildung 14 stellt beide Möglichkeiten gegenüber. Die fehlerhafte Ausgabe entstammt einem mangelhaft trainierten Modell, das vierdimensionale Hypothesenräume für die Transformation der Eingabedaten nutzt. Die in den Repräsentationen enthaltene Information ist damit zu beschränkt, um eine fehlerfreie Zuordnung in eine von sechs verschiedenen Klassen zu gewährleisten.

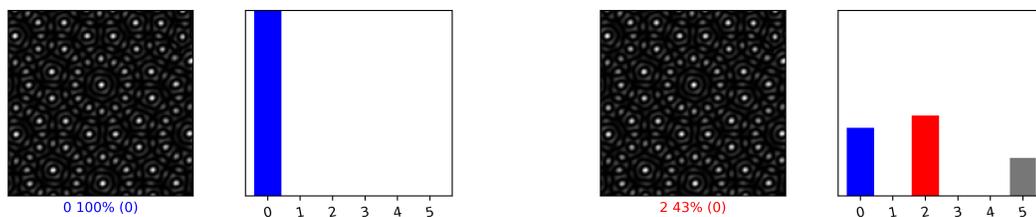


Abbildung 14: Visualisierte Ausgabe des Modells für eine richtige (links) und eine falsche Zuordnung (rechts).

Die Anzahl der verdeckten Einheiten pro Layer ist im Vergleich zum hinreichend trainierten neu-

ronalen Netz von 16 auf 4 gesenkt worden. Am Säulendiagramm der fehlerhaften Klassifikation erkennt man gut, warum sich eine One-Hot-Kodierung als Ausgabe eignet: Das neuronale Netz erkennt Merkmale und errechnet eine Korrelation des Eingabebildes mit den einzelnen Ausgabe-Kanälen. Durch die Wahl der Softmax-Aktivierungsfunktion wird diese Merkmalerkennung in Prozent angegeben und die Summe aller Kanäle ergibt 1. Das neuronale Netz entscheidet letztendlich anhand der errechneten Wahrscheinlichkeitsverteilung, um welche Art von Störstelle es sich bei der Eingabe handelt und gibt den Kanal mit der größten Korrelation als Ergebnis aus. Zum Abschluss dieser Klassifikation sind 15 Zuordnungen aus der Testdatenmenge *Test.Images* abgebildet (vgl. Abbildung 15). Sie verdeutlichen, wie gut das trainierte neuronale Netz auf die Klassifikationsaufgabe abgestimmt ist.

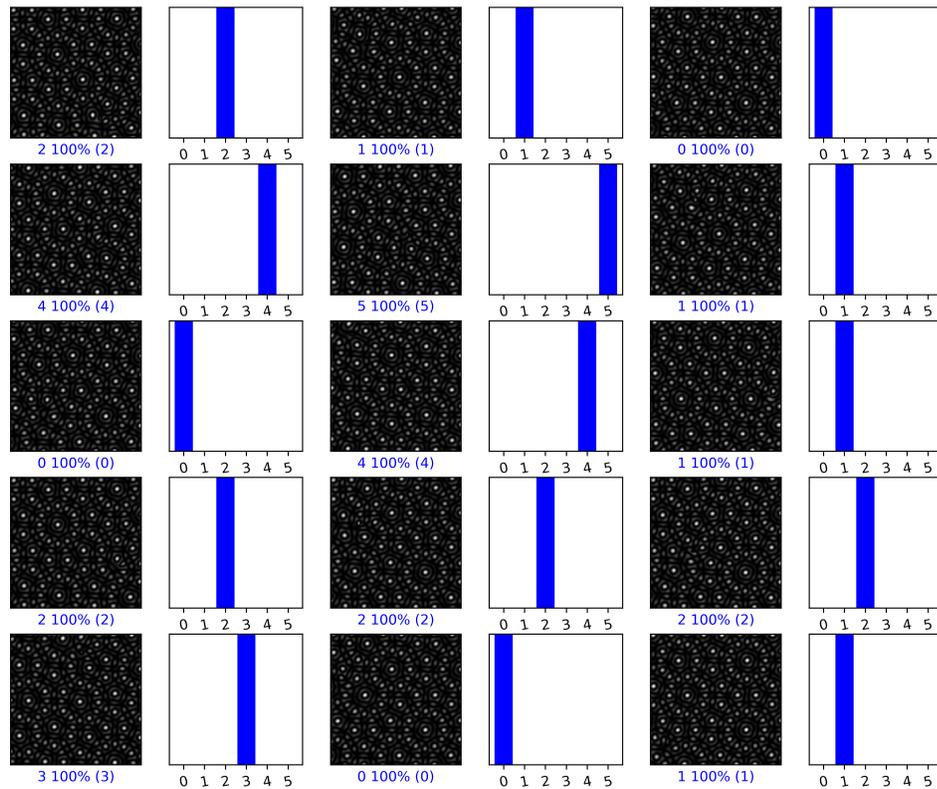


Abbildung 15: Vorhersagen des neuronalen Netzes für die ersten 15 Testdaten aus *Test.Images*.

7.2 Training und Evaluation des neuronalen Netzes

Das Modell wird erneut mit mehreren Dense-Schichten und der Relu-Funktion aktiviert. Der wesentliche Unterschied zur Klassifikation der Art des Defektes besteht in den verdeckten Einheiten jeder Schicht. Anstelle von 16 verdeckten Einheiten finden nun 512 Einheiten pro Schicht Verwendung. Der Hypothesenraum bietet dadurch erheblich umfassendere Ressourcen zur Merkmalerkennung. Das Training erfolgt in 80 Trainingsepochen, wobei die Korrektklassifizierungsrate und die Verlustfunktion weiterhin nach jeder Epoche festgehalten werden. Die Optimierungsparameter sind in Abbildung 17 gegen die Trainingsepochen aufgetragen.

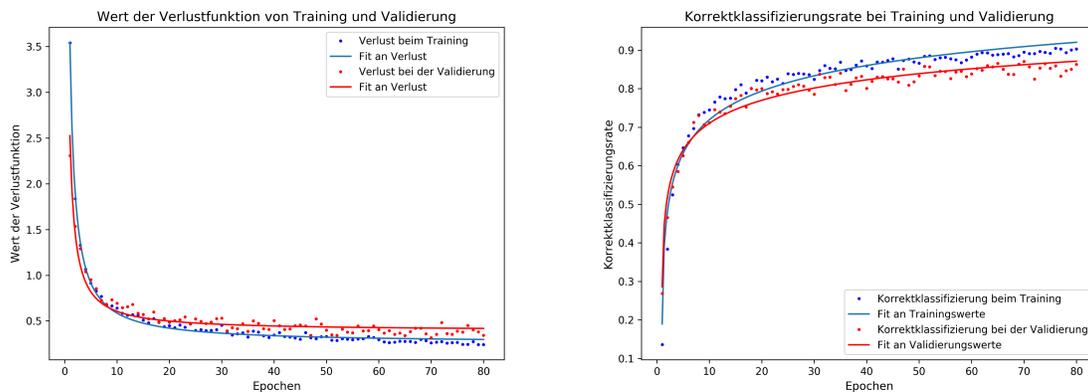


Abbildung 17: Korrektklassifizierungsrate und Verlustfunktion für die Position der Dislokation in Abhängigkeit der Trainingsepochen.

Qualitativ sinkt der Wert der Verlustfunktion mit zunehmender Epochenzahl, wohingegen der Wert der Korrektklassifikation zunimmt. Das stimmt mit den Erwartungen überein. Das Modell weist nach 80 Epochen eine Genauigkeit von 86 % auf und der Wert der Verlustfunktion sinkt bis auf den Wert 0.34 ab. Setzt man sich näher mit den Plots auseinander, so ist ebenfalls ersichtlich, dass der Fortschritt nicht monoton wächst. Das liegt daran, dass die Gewichtungen des neuronalen Netzes sehr vielen Parametern unterliegen, die allesamt und gleichzeitig an die Trainingsdaten angepasst werden. Eine Modifikation dieser Parameter hat in der Folge nicht instantan eine progressive Auswirkung auf die Validierung des Modells nach jeder Epoche und unterliegt einer Schwankung. Der Wert der Verlustfunktion nimmt reziprok mit der Epochenzahl ab. Diese Abhängigkeit wird durch die hyperbolische Fit-Funktion für die Trainings- und Validierungsdatenmenge bestätigt.

Um den Trainingsprozess weiterführend zu quantifizieren, wird zusätzlich eine Metrik definiert, welche in jeder Epoche den mittleren euklidischen Abstand zwischen Vorhersage und Zielwert ausgibt. Sie ist in Abbildung 18 für die Trainings- und Validierungsdatenmenge gegen die Trainingsepochen aufgetragen.

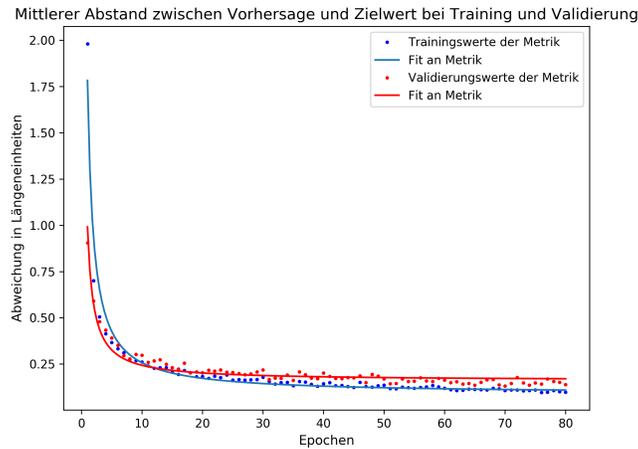


Abbildung 18: Mittlerer euklidischer Abstand zwischen den Zielwerten und den Ausgaben des neuronalen Netzes für die Position der Dislokation in Abhängigkeit der Trainingsepoche.

Der Wert für die mittlere euklidische Norm beträgt für die Trainingsdaten nach Durchlauf der 80 Epochen 0.10 Längeneinheiten. Für die Validierungsdatenmenge ergibt sich ein Wert von 0.14. Der geringe Wert der mittleren euklidischen Norm deckt sich mit der Erwartung aus der hohen Korrekturklassifizierungsrate und bestätigt sich auch bei einer Erweiterung auf die Testdatenmenge. Im Mittel ergibt sich für die Vorhersagen des neuronalen Netzes anhand der 9500 Testbilder eine geometrische Abweichung von 0.13 ± 0.11 Längeneinheiten. Insgesamt lassen sich die Ausgaben des Modells im Bezug auf die eingeführte Metrik in drei Abstände einteilen. Zunächst ist der Abstand Null, wenn das neuronale Netz dem Sample das richtige Label zuweist. Mit einer absoluten Häufigkeit von 8320 Ereignissen tritt die richtige Zuweisung in 87.6% der Fälle ein. Eine richtige Ausgabe des neuronalen Netzes ist gemeinsam mit der ermittelten Wahrscheinlichkeitsverteilung in Abbildung 19 aufgetragen. Die Zuweisung wird rot auf dem zugehörigen Potentialfeld des Rohbildes aus *Test_Images* visualisiert.

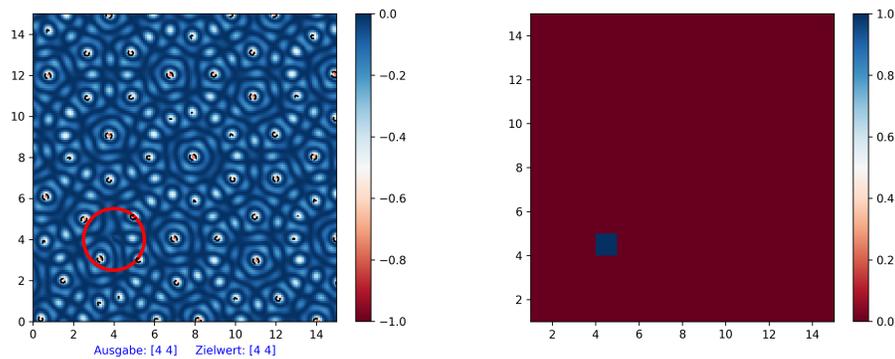


Abbildung 19: Ausgabe der Position der Dislokation im Potentialfeld des Quasikristalls (links) und als Wahrscheinlichkeitsverteilung (rechts) des neuronalen Netzes.

Die Dislokation befindet sich an dem Punkt $P = (4, 4)$. Die Wahrscheinlichkeitsverteilung wird im Wesentlichen durch den Wert bei $(4, 4)$ dominiert. Nachdem diese Position das Maximum der Verteilung bildet, wird er von dem neuronalen Netz - übereinstimmend mit dem Label - als Ausgabe festgelegt und markiert.

Mit einer relativen Häufigkeit von 12 % gibt das neuronale Netzwerk in 1136 Ereignissen die Position in einer Dimension richtig aus und liegt in der zweiten Dimension um einen Wert daneben. Die Metrik für den euklidischen Abstand beträgt dementsprechend 1. Ein Beispiel für diese fehlerhafte Zuweisung ist in Abbildung 20 aufgetragen. Ist die Zuweisung nicht richtig, so wird neben der Ausgabe in rot auch der Zielwert in grün markiert.

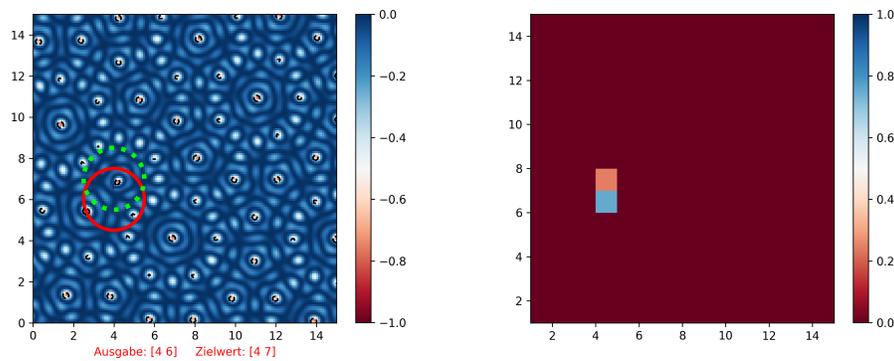


Abbildung 20: fehlerhafte Ausgabe (rot) und Zielwert (grün) der Position der Dislokation im Quasikristall (links). Die zugehörige Wahrscheinlichkeitsverteilung (rechts).

Die größte geometrische Abweichung vom tatsächlichen Zielwert beträgt $\sqrt{2}$ und ist genau dann erfüllt, wenn sowohl die Ausgabe in x-Richtung als auch die in y-Richtung unmittelbar an ihren Zielwert angrenzen. Diese Möglichkeit tritt mit einer relativen Häufigkeit von unter einem Prozent selten ein. Ein Beispiel ist in Abbildung 21 aufgetragen.

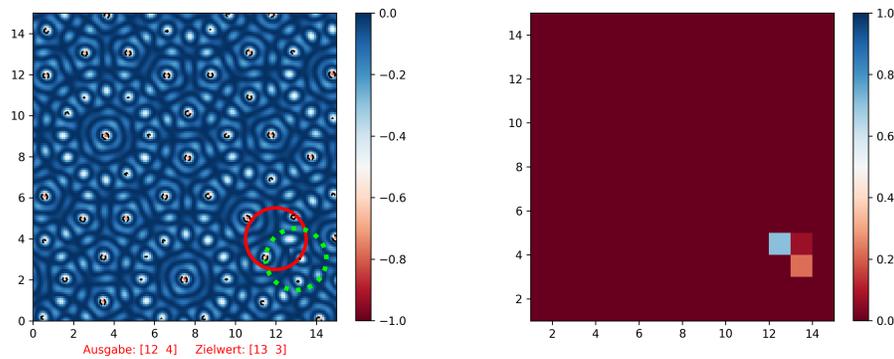


Abbildung 21: fehlerhafte Ausgabe (rot) und Zielwert (grün) der Position der Dislokation im Quasikristall (links). Die zugehörige Wahrscheinlichkeitsverteilung (rechts).

Die hohe Korrektklassifizierungsrate und die Tatsache, dass das neuronale Netzwerk nie mehr als eine Position von dem Zielwert abweicht, zeigen auf, wie erfolgreich sich die Mustererkennung auf die Testdaten erweitern lässt. Diese Beobachtung deckt sich sehr gut mit den visualisierten Wahrscheinlichkeitsverteilungen. Die Einteilung einer Ausgabe in fehlerhaft oder richtig bezieht sich durchgehend auf die gerundete Position der Störstelle. Betrachtet man die Position der Dislokation einer fehlerhaften Ausgabe vor dem Runden, so ist diese immer in der Nähe des Grenzwertes zwischen Auf- und Abrundung.

Zum Abschluss der Klassifikation sind in Abbildung 22 vier Bilder aus *Test_Images* mit den Zuweisungen des neuronalen Netzes aufgetragen. In allen Fällen stimmt die Vorhersage des Modells mit den tatsächlichen Positionen überein.

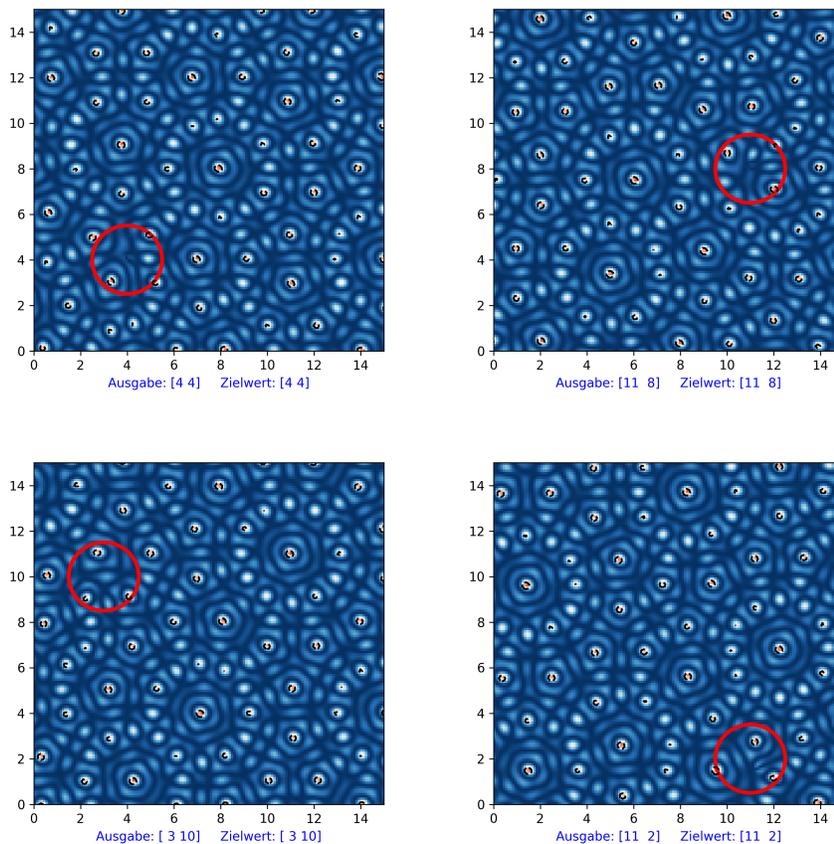


Abbildung 22: exemplarische Zuweisungen der Position der Dislokation durch das neuronale Netz.

8 Ausblick: Klassifikation von weiteren Defekten

Die in dieser Arbeit vorgestellten Modelle eignen sich beide sehr gut für die Klassifikation einer einzelnen Dislokation in einem dodekagonalen Quasikristall. Als Ausblick eignet sich die Erweiterung der Modelle auf eine variable Anzahl an Störstellen in einer beliebigen quasikristallinen Struktur. Ein solches Modell hat das Potential weitreichende praktische Anwendung in der Festkörperphysik zu finden. In Aussicht gestellt wird darüber hinaus die Einführung und Klassifikation von phasennischen Flips als Defekte in der Kristallstruktur. Diese Flips forcieren eine Umordnung einzelner Teilchen in der Ordnungsstruktur, weshalb eine Aufbereitung der Teilchenmuster aus Abschnitt 3.3 als Rohdaten für ein neues neuronales Netzwerk aussichtsreich erscheint.

Literaturverzeichnis

- [1] OHASHI, WATARU UND SPAEPEN, FRANS: *Stable Ga–Mg–Zn quasi-periodic crystals with pentagonal dodecahedral solidification morphology*. *Nature*, 330:555, 1987.
- [2] SHECHTMAN, DAN UND BLECH, ILAN UND GRATIAS DENIS UND CAHN JOHN W.: *Metallic phase with long-range orientational order and no translational symmetry*. *Physical review letters*, 53:1951, 1984.
- [3] DUBOIS, JEAN-MARIE: *Quasicrystals*. *Journal of Physics: Condensed Matter*, 13:7753, 2001.
- [4] SANDBRINK, MATTHIAS: *Tailored Colloidal Quasicrystals*. Doktorarbeit, 2015.
- [5] MARTINSONS, MIRIAM: *phasonic Degrees of Freedom in Quasicrystals*. Doktorarbeit, 2019 (eingereicht).
- [6] BROMMER, PETER: *Entwicklung und Test von Wechselwirkungspotenzialen in Quasikristallen*. Diplomarbeit, 2003.
- [7] KROMER, JUSTUS A. UND SCHMIEDEBERG, MICHAEL UND ROTH JOHANNES UND STARK HOLGER: *What phasons look like: Particle trajectories in a quasicrystalline potential*. *Physical review letters*, 108:218301, 2012.
- [8] DE, PIALI UND PELCOVITS, ROBERT A.: *Linear elasticity theory of pentagonal quasicrystals*. *Physical Review B*, 35:8609, 1987.
- [9] SOCOLAR, JOSHUA E.S. UND LUBENSKY, T.C. UND STEINHARDT PAUL J.: *Phonons, phasons, and dislocations in quasicrystals*. *Physical Review B*, 34:3345, 1986.
- [10] RÜHLE, FELIX UND SANDBRINK, MATTHIAS UND STARK HOLGER UND SCHMIEDEBERG MICHAEL: *Effective substrate potentials with quasicrystalline symmetry depend on the size of the adsorbed particles*. *The European Physical Journal E*, 38:54, 2015.
- [11] CHOLLET, FRANCOIS: *Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek*. MITP-Verlags GmbH & Co. KG, 2018.
- [12] ROJAS, RAUL: *Theorie der neuronalen Netze: eine systematische Einführung*. Springer-Verlag, 2013.
- [13] GULLI, ANTONIO UND PAL, SUJIT: *Deep Learning with Keras*. Packt Publishing Ltd, 2017.
- [14] DUNNE, ROB A. UND CAMPBELL, NORM A.: *On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function*. In: *Proc. 8th Aust. Conf. on the Neural Networks, Melbourne*, 1997.

Eidesstattliche Erklärung

Ich versichere, dass ich die Bachelorarbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat. Alle Ausführungen der Arbeit, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Erlangen, den

ALI DÖNER