

Hochschule Düsseldorf  
University of Applied Sciences



Fachbereich Medien  
Faculty of Media



# PointNet

## Funktionsweise und Anwendungsgebiete

**Marco Winter**  
Matrikel-Nr.: 740416

Wissenschaftliche Vertiefung im Studiengang

**Bachelor Medieninformatik**  
**17. Mai 2020**  
Version 1.1.0

**Betreuer:**  
Prof. Dr. rer. nat. Christian Geiger  
M. Sc. Marcel Tiator

## Zusammenfassung

Diese Arbeit beschäftigt sich mit PointNet, einem von Qi et al. [1] entworfenen neuronalen Netzwerk zur Analyse von 3D-Punktwolken. Nach einer kurzen Einführung in die zum Verständnis nötigen Grundlagen werden der Aufbau und die Funktionsweise des Netzwerks schrittweise erläutert. Ebenso wird die praktische Umsetzung von PointNet anhand einer eigenen Implementation mithilfe der Deep Learning API Keras besprochen. Hierfür wird ein interaktives Notebook zur Verfügung gestellt. Abschließend erfolgt ein kritischer Vergleich mit anderen Netzarchitekturen hinsichtlich der zugrundeliegenden Konzepte und deren Vor- und Nachteilen.

Mithilfe einer detaillierten Analyse soll diese Arbeit einen Einblick in den Themenbereich der Objekterkennung und -Segmentierung im dreidimensionalen Raum geben und dabei helfen, ein tiefergehendes Verständnis für die behandelte Netzarchitektur Pointnet zu erlangen.

**Kapitel 1** beschreibt die Grundlagen neuronaler Netze und soll einen thematischen Einstieg für das Verständnis dieser Arbeit bieten.

**Kapitel 2** erläutert den Aufbau und die theoretische Funktionsweise der von Qi et al. [1] entworfenen PointNet-Architektur.

**Kapitel 3** analysiert eine Beispiel-Implementation der PointNet-Architektur. Hierfür wird ein interaktives Jupyter Notebook bereitgestellt.

**Kapitel 4** zeigt Projekte, in denen PointNet erfolgreich angewendet wurde.

**Kapitel 5** soll anhand eines kritischen Vergleichs die Unterschiede sowie Vor- und Nachteile von PointNet gegenüber anderen Architekturen hervorheben.

**Kapitel 6** bietet einen kurzen Ausblick auf eine Weiterentwicklung der PointNet-Architektur und beschreibt die Unterschiede zwischen PointNet++ und PointNet

## Inhaltsverzeichnis

1.	Einleitung: Grundlagen neuronaler Netzwerke.....	1
1.1.	Topologie neuronaler Netzwerke .....	1
1.2.	Aktivierungsfunktionen .....	3
1.3.	Training eines KNN.....	5
1.4.	Deep Learning .....	6
2.	PointNet – Aufbau und Workflow .....	12
2.1.	Workflow der PointNet-Architektur: Klassifizierung .....	13
2.2.	Workflow der PointNet-Architektur: Segmentierung.....	22
2.3.	Workflow der PointNet-Architektur: Semantik .....	24
3.	PointNet in der Praxis – Implementation mit Keras .....	25
4.	Anwendungsgebiete der PointNet-Architektur .....	26
4.1.	Beispielprojekt 1 .....	26
4.2.	Beispielprojekt 2 .....	26
4.3.	Beispielprojekt 3 .....	27
5.	Unterschiede zu alternativen Netzarchitekturen.....	28
5.1.	Klassifizierung.....	28
5.2.	Segmentierung .....	30
5.3.	Semantik.....	32
6.	PointNet++ - eine Weiterentwicklung .....	33
7.	Fazit .....	36
8.	Quellenverzeichnis.....	37

# 1. Einleitung: Grundlagen neuronaler Netzwerke

Künstliche neuronale Netze (KNN) sind Berechnungsmodelle, deren Struktur stark vom Aufbau und der Funktionsweise biologischer Nervennetzwerke im Gehirn inspiriert ist. Sie bestehen aus einer Vielzahl untereinander verknüpfter Recheneinheiten, welche als künstliche Neuronen bezeichnet werden und gemeinsam ein hochkomplexes Kommunikationsnetzwerk bilden [2, p. 268].

Auf Basis vorgegebener Beispieldaten sind diese Systeme ohne spezifische Programmierung in der Lage, die Ausführung komplexer Aufgaben zu erlernen. Hierbei sind verschiedenste Problemstellungen denkbar. Von der Mustererkennung in beliebigen Datenmengen über die Realisierung autonomer Fahrzeuge [3] bis hin zur Erzeugung von virtuellen Gemälden [4] - künstliche neurale Netze werden in verschiedensten Bereichen eingesetzt.

Eine Anwendung für neuronale Netze, welche im Rahmen dieser Arbeit besondere Aufmerksamkeit erfährt, ist die Klassifizierung, Segmentierung und semantische Analyse dreidimensionaler Objekte. Hierbei lernt ein Netzwerk, anhand eines entsprechenden Datensatzes – bspw. das Signal eines Laserscanners in Form einer Punktwolke – Objekte und deren einzelne Bestandteile in einem gewissen Umfeld zu identifizieren.

In diesem Kapitel werden grundlegende Konzepte und Begriffe neuronaler Netze erläutert, welche für das Verständnis des Hauptteils dieser Arbeit vonnöten sind.

## 1.1. Topologie neuronaler Netzwerke

Ein KNN besteht in seinem Kern aus einem Netzwerk miteinander verbundener künstliche Neuronen. Jedes dieser Neuronen verfügt in der Regel über ein oder mehrere gewichtete *Eingangssignale* (engl. input), eine *Aktivierungsfunktion* (engl. activation function) und eine *Ausgabe* (engl. output). Der oder die Input-Werte werden mit ihren jeweiligen Gewichten multipliziert und anschließend als Summe an die Aktivierungsfunktion übergeben. Diese wiederum berechnet einen einzelnen Output-Wert für etwaige nachfolgende Neuronen oder die finale Ausgabe des Netzwerks [5, Ch. 1].

Die Neuronen eines KNN sind häufig in miteinander verknüpften Schichten angeordnet (wie in Abbildung 1.1 dargestellt). Mathematisch betrachtet stellen diese Schichten disjunkte Teilmengen dar, deren Neuronen über gewichtete Kanten miteinander in Beziehung stehen [2, p. 269]. Die erste und letzte Schicht eines Netzwerks werden hierbei als Eingabe- (engl. input layer) beziehungsweise Ausgabeschicht (engl. output layer) bezeichnet. Die Neuronen dieser Schichten verfügen über besondere Eigenschaften:

*Eingangsneuronen* (engl. input neurons) besitzen keinen gewichteten Eingang und keine Aktivierungsfunktion. Sie nehmen als Input-Schnittstelle die zu verarbeitenden Daten auf und geben diese über gewichtete Kanten an die zweite Schicht weiter.

*Ausgangsneuronen* (engl. output neurons) hingegen besitzen ausschließlich eingehende Kanten sowie ggf. eine Aktivierungsfunktion und stellen die Output-Schnittstelle dar.

Die Anzahl der Neuronen in diesen Schichten hängt von der Dimensionalität der zu verarbeitenden Daten bzw. des gewünschten Outputs ab [2, p. 269]. Ein Netzwerk, das z. B. den Verkaufswert eines Hauses anhand der Parameter *Grundfläche* und *Anzahl der Zimmer* ermittelt, verfügt über zwei Eingangs- und ein Ausgangsneuron.

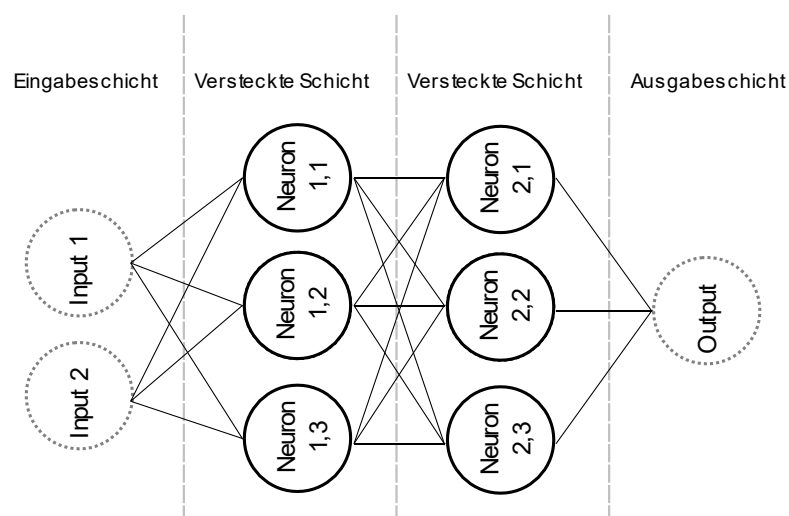


Abbildung 1.1: Aufbau eines vollständig verbundenen KNN

Zwischen der Eingangs- und Ausgangsschicht eines KNN liegen die sogenannten versteckten Schichten (engl. hidden layer). Die Neuronen dieser Schichten verarbeiten die Daten aus der Eingangsschicht oder einer vorherigen versteckten Schicht und übergeben das Ergebnis an die Ausgabeschicht oder eine nachfolgende versteckte Schicht. Sie helfen dem Netzwerk, die eingehenden Daten zu verstehen und erzeugen Schicht für Schicht die Ausgabe [5, Ch. 1].

Wie die Verknüpfung zwischen den Schichten eines neuronalen Netzes im Detail realisiert wird, ist ebenso wie die Anzahl der versteckten Schichten und deren Neuronen gänzlich von der gewählten Architektur und dem Zweck des KNN abhängig. Prinzipiell kann jedoch zwischen vorwärts gerichteten (engl. feed forward neural network) und rückgekoppelten Netzwerken (engl. recurrent / feedback neural network) unterschieden werden.

Bei Letzterem werden Verbindungen zwischen den Ausgängen einzelner Neuronen wahlweise mit deren eigenen Eingängen (direct feedback), den Eingängen von Neuronen in derselben Schicht (lateral feedback) oder den Eingängen von Neuronen in einer vorangegangenen Schicht (indirect feedback) verknüpft. Durch die Rückkopplung wird eine Art interner „Speicher“ in die Netzwerkstruktur eingebaut, welcher unter anderem die Verarbeitung von Zeitreihendaten [5, Ch. 13] oder Differentialgleichungen ermöglicht [6, p. 141].

## 1.2. Aktivierungsfunktionen

Wie im Abschnitt 1.1 beschrieben, besitzen die Neuronen eines KNN (mit Ausnahme der Eingabeneuronen) eine Aktivierungsfunktion. Diese Funktion ermöglicht es dem Neuron, aus der Summe seiner gewichteten Eingänge einen Ausgangswert zu berechnen.

Aktivierungsfunktionen können viele verschiedene Formen haben. Die Auswahl der korrekten Funktion stellt einen wichtigen Schritt beim Entwurf einer Netzarchitektur dar und hat einen starken Einfluss auf das benötigte Format der Eingabedaten sowie auf die Performanz und Genauigkeit des Netzwerks [5, Ch. 1].

In diesem Abschnitt werden zwei Aktivierungsfunktionen vorgestellt, welche für das Verständnis der Erläuterungen in Kapitel 2 und 3 vonnöten sind. Weitere

häufig vorkommende Aktivierungsfunktionen können in [5] und [6] nachgelesen werden.

### 1.2.1. Rectified Linear Unit

Eine Rectified Linear Unit (deutsch: gleichgerichtete Lineareinheit) – kurz ReLU genannt – ist ein Neuron, dessen Aktivierungsfunktion ein Gleichrichter (engl.: rectifier) ist. Die Funktionsgleichung eines Gleichrichters wird wie folgt beschrieben:

$$\Phi(z) = \max(0, z) \quad (1.1)$$

Eine ReLU setzt also sämtlichen negativen Input auf null und gibt positiven Input unverändert weiter. Der zugehörige Funktionsgraph wird in Abbildung 1.2 gezeigt. Obwohl es sich um eine simple Funktion handelt, liefert die ReLU aufgrund ihrer mathematischen Eigenschaften bessere Ergebnisse als andere Aktivierungsfunktionen und wird daher häufig als beste Alternative für die versteckten Schichten eines KNN empfohlen [5, Ch. 1].

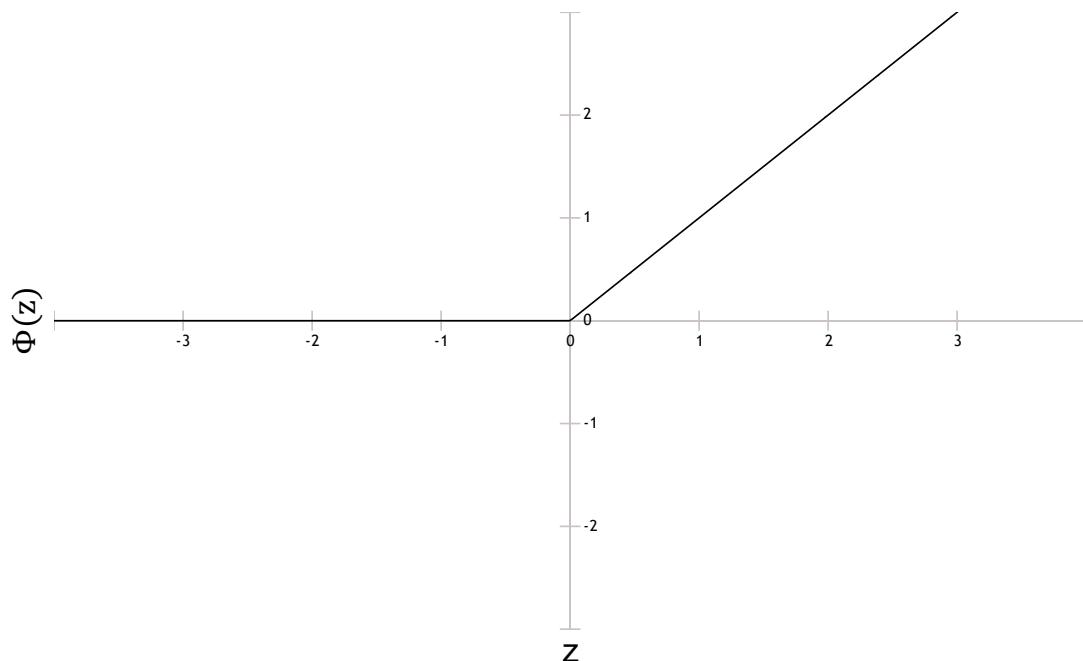


Abbildung 1.2: Graph der Gleichrichterfunktion  $\Phi(z) = \max(0, z)$

### 1.2.2. Softmax-Funktion

Bei der Softmax-Funktion handelt es sich um eine Aktivierungsfunktion, die vornehmlich in der Ausgabeschicht von klassifizierenden KNN eingesetzt wird [5, Ch. 1]. Hierbei wird jedem Neuron der Ausgabeschicht ein Wert zwischen null und eins zugewiesen. Dieser stellt die prozentuale Wahrscheinlichkeit dar, mit welcher ein Eingabewert der zum Ausgabeneuron gehörigen Klasse zugeordnet werden kann. Die Funktionsvorschrift der Softmax-Funktion wird wie folgt beschrieben:

$$\Phi_i = \frac{e^{z_i}}{\sum_{j \in group} e^{z_j}} \quad (1.2)$$

Hierbei bezeichnet  $z_i$  den Eingabewert eines Neurons, während die Summe die Werte aller  $j$  Eingabewerte für alle Neuronen der Ausgabeschicht aggregiert. Es handelt sich hierbei um eine normalisierte Exponentialfunktion und kann als Generalisierung der logistischen Sigmoid-Funktion betrachtet werden [7, p. 198]. Die Besonderheit der Softmax-Funktion ist die Tatsache, dass der Wert eines Neurons in Abhängigkeit zur Eingabe der benachbarten Neuronen gebildet wird. Dies führt dazu, dass die Summe aller Ausgabewerte einer Schicht nach Verwendung der Softmax-Funktion gleich eins ist. Als Aktivierungsfunktion der Ausgabeschicht eines KNN wird mithilfe der Softmax-Funktion diejenige Klasse gewählt, deren Neuron den höchsten Ausgabewert vorweist [5, Ch. 1].

### 1.3. Training eines KNN

Damit ein KNN korrekte Vorhersagen treffen, Daten klassifizieren oder anderweitige Probleme lösen kann, muss es trainiert werden. Beim Training eines KNN werden anhand von Beispieldatensätzen mittels eines Algorithmus die Verbindungsgewichte ggf. weitere Parameter schrittweise angepasst, bis ein bestimmtes Kriterium optimiert ist [6, p. 40]. Die Trainingsstrategie und der Algorithmus hängen hierbei von der Art der Testdaten und der Aufgabe des Netzwerks ab.

Im folgenden Abschnitt wird die Trainingsstrategie des überwachten Lernens (engl.: supervised learning) erläutert. Weitere Methoden und Algorithmen, welche



im Rahmen dieser Arbeit von geringerer Bedeutung sind, können in der Literatur von Bishop [7], Bibel et al. [6] oder Heaton [5] zur weiteren Lektüre nachgeschlagen werden.

### 1.3.1. Supervised Learning

Das Ziel des überwachten Lernens (engl.: supervised learning) ist die Modellierung einer bedingten Wahrscheinlichkeitsverteilung  $p(t | x)$ , mit deren Hilfe ein Ergebnis  $t$  für jeden möglichen Eingabewert  $x$  möglichst vorhergesagt werden kann. Hierfür werden Trainingsdaten in Form von Wertepaaren  $(t, x)$  benötigt, wobei  $t$  der korrekte Ausgangswert ist [7, pp. 137–138].

Um das Konzept an einem Beispiel zu verdeutlichen: Gegeben sei ein KNN, das E-Mails als *Spam* oder *non-Spam* klassifizieren soll. Um dieses Netzwerk mithilfe von überwachtem Training auf seine Aufgabe vorzubereiten, lässt man es wiederholt Datensätze aus Beispiel-Mails verarbeiten. Diese sind bereits als *Spam* oder *non-Spam* markiert. Das Ergebnis ist also bei einem Trainingsdatensatz für alle darin enthaltenen Beispiele bekannt.

Zu Beginn des Trainings sind die vom KNN vorhergesagten Werte aufgrund zufälliger Initialisierung der Gewichtungen oftmals falsch. Durch die Verwendung eines passenden Algorithmus wird die Differenz zwischen Vorhersage und vorgegebener Wahrheit (dargestellt durch eine Verlustfunktion) schrittweise reduziert. Auf diese Weise wird eine Annäherung der oben genannten bedingten Wahrscheinlichkeitsverteilung ermittelt, mit der Vorhersagen für neue Werte  $x$  getroffen werden können.

## 1.4. Deep Learning

Deep Learning ist ein untergeordneter Themenbereich des maschinellen Lernens, welcher vor allem für moderne KNN-Architekturen von großer Bedeutung ist. Deep Learning stellt eine neue Art des Lernens dar, deren Schwerpunkt in der Kombination aufeinanderfolgender Schichten zunehmend aussagekräftiger Darstellungen liegt [8, p. 8].

Verständlicher ausgedrückt handelt es sich bei einem „tiefen“ KNN um ein Netzwerk mit zwei oder mehr versteckten Schichten. In diesen Schichten werden die Eingabedaten sukzessive abstrahiert und in eine Darstellung überführt, mit welcher sich die Aufgabe des Netzwerkes leichter lösen lässt. Abbildung 1.3 zeigt diese zunehmende Abstraktion am Beispiel eines KNN zur Erkennung handschriftlicher Zahlen. Die Ausgaben der einzelnen Schichten werden zunehmend abstrahiert, bis schlussendlich anhand von 3x3-Gittern die Klassifizierung durchgeführt wird.

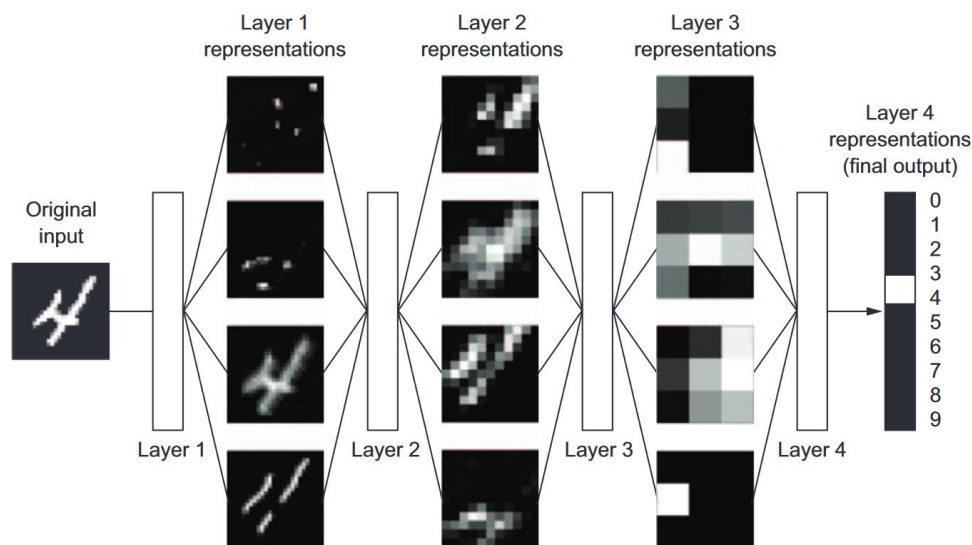


Abbildung 1.3: Beispiel eines "tiefen" neuronalen Netzwerks [8, p. 9].

Das Ziel eines „tiefen“ KNN ist also die (deterministische) Umwandlung der Eingabedaten in eine Form, die für die eigentliche Aufgabe des Netzwerkes (Klassifizierung, Segmentierung etc.) am besten geeignet ist. Oft wird dieser Schritt auch Feature Building genannt, da das Netzwerk die zur Verarbeitung nötigen Eigenschaften für einen bestimmten Input selbst erzeugt.

Im folgenden Abschnitt wird eine verbreitete Architektur für „tiefe“ neuronale Netze vorgestellt, um deren Eigenschaften näher zu beleuchten und zeitgleich wichtige Grundlagen für den Hauptteil dieser Arbeit zu vermitteln.

### 1.4.1. Convolutional Neural Network

Convolutional Neural Networks (deutsch: faltendes neuronales Netzwerk) sind eine spezialisierte Art von neuronalen Netzen zur Verarbeitung von Daten, die eine gitterartige Topologie besitzen. Beispiele sind mittels Sampling erzeugte Zeitreihendaten, die als 1-D-Gitter betrachtet werden können, sowie Bilddaten, die man sich als ein 2-D-Pixelgitter vorstellen kann. CNNs waren in den vergangenen Jahren enorm erfolgreich in der praktischen Anwendung. Der Name "Convolutional Neural Network" bedeutet, dass das Netzwerk eine mathematische Operation namens Faltung verwendet [9, p. 326].

CNNs sind Feed-Forward Netzwerke, besitzen gegenüber gewöhnlichen Netzwerken dieser Art jedoch mehrere Vorteile:

- Die von ihnen gelernten Muster sind invariant gegenüber Translationen [8, p. 123].
- CNNs können durch aufeinanderfolgende Faltungen räumliche Hierarchien lernen und somit immer komplexere Abstraktion durchführen [8, p. 123].
- Durch die Eigenschaften der Faltung (nur lokale Verbindungen, geteilte Gewichtungsparemeter) sowie durch das Max-Pooling ist die Anzahl der zu trainierenden Parameter z.T. mehrere Größenordnungen kleiner als bei gewöhnlichen, voll verbundenen Netzwerken.

Nachfolgend werden die typischen Bestandteile eines CNN kurz vorgestellt. Detailliertere Erläuterungen und Beispiele können in [9, Ch. 9], [8, Ch. 5] sowie [5, Ch. 10] nachgeschlagen werden.

#### **Faltungsschicht (engl.: Convolutional Layer):**

Der Hauptzweck dieser Schicht ist es, die Eingabedaten zu filtern und so lokale Merkmale zu erschließen, anhand derer die Aufgabe des Netzwerks erfüllt werden kann. Diese Merkmale können bspw. Kanten, Linien oder andere visuelle Eigenschaften eines Bildes sein. Die Anzahl der verwendeten Filter (auch Kernel genannt) korreliert hierbei direkt mit der Menge an Merkmalen [5, Ch. 10].

Für den Fall eines Bildes, dessen Inhalt klassifiziert werden soll, kann man sich die Filter der Faltungsebene als Pixel-Gitter vorstellen. Jeder „Pixel“ des Filters verfügt über einen trainierbaren Gewichtungsfaktor. Im Rahmen der Faltung wird das Bild schrittweise vom Filter-Gitter durchlaufen. Für jeden Schritt wird die Summe aller Produkte aus übereinanderliegenden Bildpixeln und Filterpixeln bzw. Gewichtungswerten gebildet und ausgegeben. Abbildung 1.4 stellt diesen Prozess schematisch dar. Der 2x2 Kernel durchläuft die Eingabe für alle gültigen Positionen und gibt die entsprechenden Produktsummen aus.

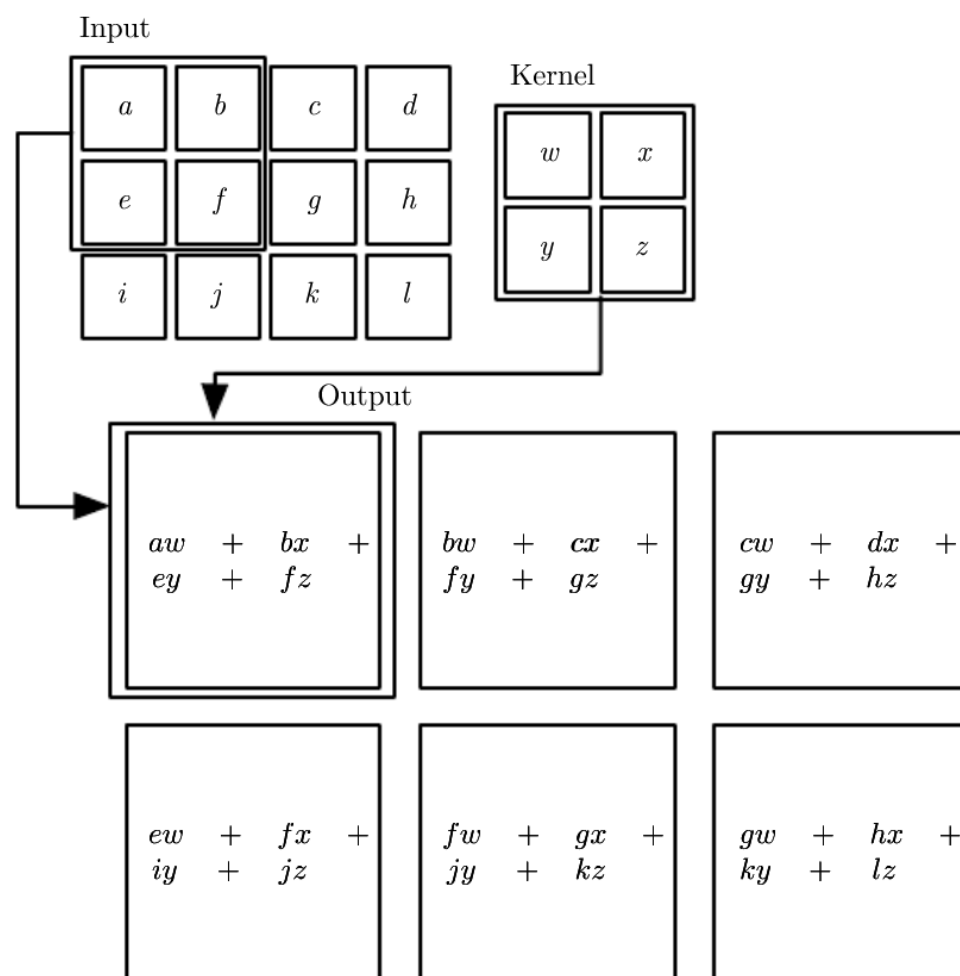


Abbildung 1.4: Schema der Faltungsoperation [9, p. 330].

Das Ergebnis der Faltungsoperation wird als Eingangssignal an die Neuronen der Faltungsschicht übergeben und mittels einer Aktivierungsfunktion, welche i.d.R. ReLU ist, verarbeitet.

**Max-Pooling:**

Beim Max-Pooling werden die Ausgangswerte der Faltungsschicht aggregiert. Hierbei durchläuft in Abhängigkeit mehrerer Parameter (Größe, Schrittgröße) ein Filter die Daten und übernimmt in jedem Schritt lediglich den maximalen Eingangswert in seinem Einflussbereich. Dieser Prozess wird in Abbildung 1.5 dargestellt. Hierbei wird das Gitter von einem 2x2-Filter mit der Schrittgröße 2 durchlaufen. Die farbig hinterlegten Quadrate stellen die einzelnen Filterschritte dar.

Zu den Zielen des Max-Pooling zählt die Reduktion der Gewichtungsparemeter zur Steigerung der Trainingsgeschwindigkeit [8, p. 128] sowie die Möglichkeit, nicht-uniforme Eingabedaten wie bspw. Bilder in verschiedenen Auflösungen mit einem einzigen Netzwerk zu verarbeiten [9, p. 339].

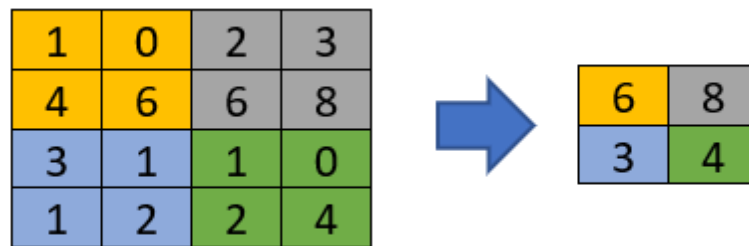


Abbildung 1.5: Beispiel für Max-Pooling mit einem 2x2 Filter und Schrittgröße 2

**Fully Connected Layer:**

Nachdem mithilfe von Faltungen und Pooling eine Darstellung der Daten erreicht ist, welche für die eigentliche Aufgabe des CNN optimal ist, folgt abschließend eine vollständig verknüpfte Schicht (engl. Fully-connected Layer), auch FC Layer oder Dense Layer genannt. Jedes Neuron der FC Layer ist mit jedem Neuron der vorangegangenen Schicht verbunden und hat somit die typische Struktur, welche bereits aus Feed-Forward-Netzwerken bekannt ist. Als Aktivierungsfunktionen werden hier typischerweise die ReLU oder im Falle der Ausgangsschicht die Softmax-Funktion verwendet.

Placeholder!

TODO: Zusammenfassung

- Welche Themen wurden in diesem Kapitel behandelt
- Welche Punkte sind von zentraler Bedeutung
- Hinweis auf den Hauptteil als Übergang: „Die nächsten Kapitel beschäftigen sich mit dem Hauptthema dieser Arbeit: Der Netzarchitektur PointNet. Deren Funktion wird komponentenweise erläutert und anhand einer beispielhaften Implementation veranschaulicht.“

## 2. PointNet – Aufbau und Workflow

Bei PointNet handelt es sich um eine KNN-Architektur, die 2016 von Qi et al. [1] entworfen wurde. Das Ziel bzw. die Aufgaben von PointNet sind die Klassifizierung, Segmentierung und die semantische Analyse von dreidimensionalen Objekten und Szenen.

PointNet verfolgt im Gegensatz zu vergleichbaren Netztypen (siehe Kapitel 5) eine neuartige Strategie: Objekte und Szenen werden in Form von Punktwolken direkt vom Netzwerk analysiert. Diese direkte Verarbeitung hat den Vorteil, dass die Daten nicht gesondert aufbereitet und in bestimmte Formate konvertiert werden müssen. Sie ist hocheffizient und im empirischen Vergleich dem bisherigen Stand der Technik nicht nur ebenbürtig, sondern zum Teil sogar überlegen [1, p. 1].

Allerdings birgt die Verarbeitung von Punktwolken auch einen entscheidenden Nachteil. Mathematisch betrachtet handelt es sich bei Punktwolken um ungeordnete Mengen. Die Reihenfolge, in welcher die Elemente einer Punktwolke an PointNet übergeben werden, ist also nicht fest, sondern variabel. Basierend auf dieser Eigenschaft muss das Netzwerk eine wichtige Voraussetzung erfüllen: Die Ausgabe für eine beliebige Punktwolke muss invariant gegenüber allen möglichen Permutationen der Punktwolken-Elemente sein [1, p. 1f.].

In den folgenden Abschnitten wird die in Abbildung 2.1 dargestellte PointNet-Architektur in separate Schritte aufgeteilt und in seiner Funktionsweise erläutert. Hierbei werden die Workflows zur Klassifizierung, Segmentierung und der semantischen Analyse von Szenen dargestellt.

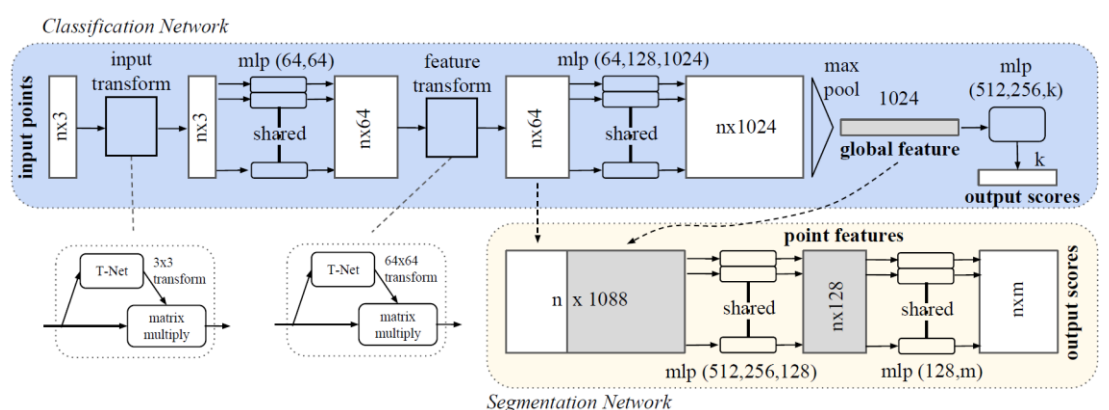


Abbildung 2.1: Aufbau der PointNet-Architektur [1, p. 3]

## 2.1. Workflow der PointNet-Architektur: Klassifizierung

Die PointNet-Architektur lässt sich grundlegend in zwei Bereiche unterteilen: Das Klassifizierungs- und das Segmentierungsnetzwerk. Im Klassifizierungsnetzwerk werden auf Basis der Eingabedaten für jedes Element der Punktwolke Eigenschaften (Features) ermittelt, welche lokale Informationen der Punktwolke enthalten. Dies können Kanten, Punktcluster oder Ähnliches sein. Anschließend wird mithilfe des Max-Pooling eine gegenüber dem Input invariante globale Signatur aggregiert, welche über mehrere vollständig verknüpfte Schichten (FC Layer) einer Objektklasse zugeordnet werden kann. Für die versteckten Schichten werden i.d.R. ReLU als Aktivierungsfunktion sowie batch normalization [10] verwendet. Für die Ausgabeschichten zur Klassifizierung und Segmentierung sowie die Ausgabeschichten der T-Nets findet die Softmax-Funktion Anwendung [1, p. 3].

Wie an Abbildung 2.2 zu erkennen ist, besteht das in der Abbildung markierte Klassifizierungsnetzwerk aus insgesamt sechs Teilschritten, welche nachfolgend erläutert werden.

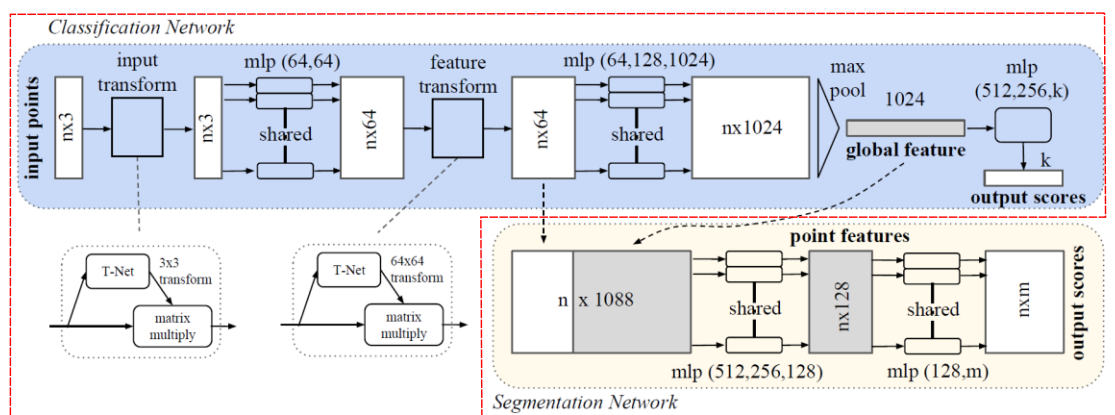


Abbildung 2.2: Klassifikationsnetzwerk der PointNet-Architektur [1, p. 3]



### 2.1.1. Input-Transformation

Wie bereits einleitend erwähnt ist die Invarianz gegenüber Permutationen des Inputs eine Grundvoraussetzung für die Funktion der PointNet-Architektur. Für die korrekte semantische Beschriftung einer Punktwolke wird zusätzlich eine weitere Form der Invarianz benötigt: Die Ausgabe darf nicht von geometrischen Transformationen wie Rotationen und Translationen beeinflusst werden. PointNet löst diese Problematik mithilfe eines „Mini-Netzwerks“, welches die Bezeichnung T-Net trägt. Das T-Net folgt in seinem Aufbau der PointNet-Architektur und hat das Ziel, eine affine Transformationsmatrix zu bestimmen. Diese Matrix wird auf die Elemente der Punktwolke angewendet und führt so zu einer Normalisierung des Inputs. Die Input-Transformation, welche in Abbildung 2.3 markiert ist, stellt somit den ersten Schritt im PointNet-Workflow dar.

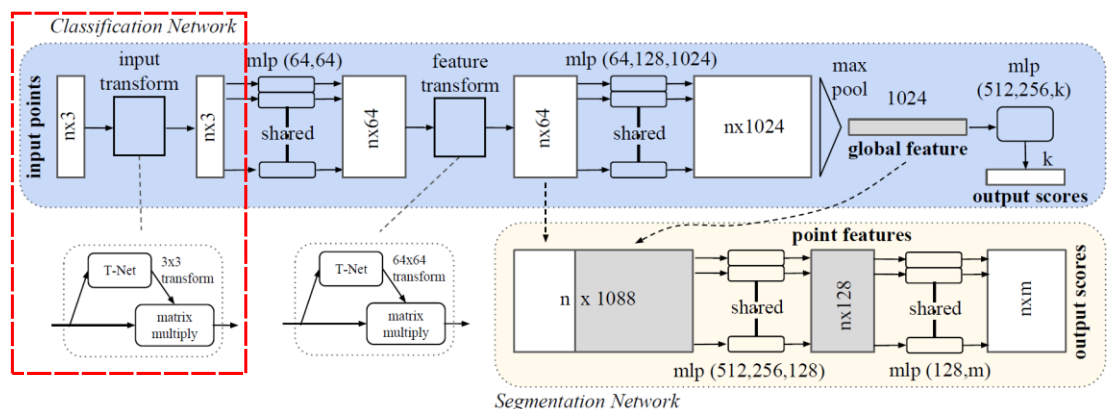


Abbildung 2.3: Erster Schritt - Input-Transformation mittels T-Net [1, p. 3]

Die Idee hinter dem T-Net basiert auf dem von Jaderberg et al. [11] vorgestellten Spatial Transformer Network (STN). Vereinfacht ausgedrückt ist es das Ziel eines STN, etwaige Verschiebungen oder Verzerrungen eines Inputs gegenüber eines definierten Basiszustands zu kompensieren, bevor eine weitere Verarbeitung stattfindet. Abbildung 2.4 zeigt das in [11, p. 2] vorgestellte Beispiel für die Funktionsweise eines STN. Hier wurden Bilder von MNIST-Zahlen zufälligen Transformationen unterzogen (a) und anschließend von einem STN verarbeitet. Die Darstellung zeigt, wie das Netzwerk zunächst die relevanten Regionen eines Bildes herausfiltert (b) und diese anschließend in einer normalisierten Pose ausgibt (c). Diese Ausgabe wird dann für die Klassifizierung genutzt (d).

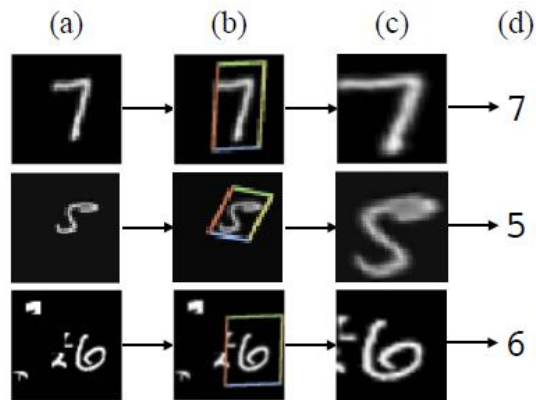


Abbildung 2.4: Beispiel für die Funktion eines STN [11, p. 2]

Auf PointNet übertragen bedeutet dies also, dass im Schritt der Input-Transformation die einzelnen Elemente der Punktwolke gewissermaßen in Position gerückt werden. Da die Punkte in Form von Vektoren an das Netzwerk übergeben werden und für die Bildung der Features voneinander unabhängig auf höherdimensionale Räume abgebildet werden (mehr dazu in Abschnitt 2.2.2), kann dies durch eine einfache Matrixmultiplikation erreicht werden [1, p. 4].

Abbildung 2.5 zeigt den schematischen Aufbau eines T-Net. Mithilfe eines mehrschichtigen Perzeptrons (multi layer perceptron), kurz MLP genannt, werden Features gebildet, welche in einer Pooling-Schicht zu einer Signatur aggregiert werden. Anhand dieser Signatur wird durch mehrere FC Layer die 3x3-Transformationsmatrix gebildet.

Da der Aufbau des T-Net im Wesentlichen dem des übergeordneten PointNet gleicht, werden die Details der einzelnen Bestandteile im nachfolgenden Abschnitt 2.2.2 näher erläutert.

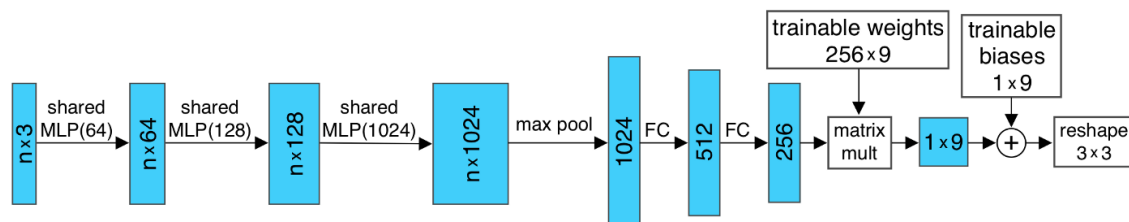


Abbildung 2.5: Aufbau T-Net (Placeholder! TODO: eigene Darstellung)

Placeholder!

TODO: Absatz über visuelle Darstellung (Input vs. Output nach Transformation)

Grafik 2.4: Beispiel-Bilder (vorher-nachher) Eventuell eigene Bilder (Plot der Punktwolke vor/nach input transform für mehrere Objekte)

## 2.1.2. Feature Building

Im zweiten Schritt des Klassifizierungsnetzwerks – dargestellt in Abbildung 2.6 – werden die normalisierten Koordinaten über einen MLP mit zwei versteckten Schichten und geteilten Gewichtungsparametern individuell auf einen höherdimensionalen Raum abgebildet. Umgangssprachlich ausgedrückt heißt dies, dass für jeden Punkt auf Basis seiner drei Raumkoordinaten insgesamt 64 Werte auf Basis der Gewichte berechnet werden.

Hierbei sind zwei Dinge von elementarer Bedeutung:

1. Die trainierbaren Gewichte der MLP-Schichten werden von allen Punkten geteilt. Jede Änderung der Gewichte im Trainingsprozess hat unmittelbare Folgen für alle Elemente der Punktwolke. Das Netzwerk muss also eine für alle Punkte passende Kombination von Gewichtungsparametern finden.
2. Alle Punkte werden individuell abgebildet. Abgesehen von den geteilten Gewichten gibt es keine Relation zwischen benachbarten Punkten. Dies bedeutet, dass die Bestimmung der Features invariant gegenüber der Reihenfolge der Punkte im Input ist.

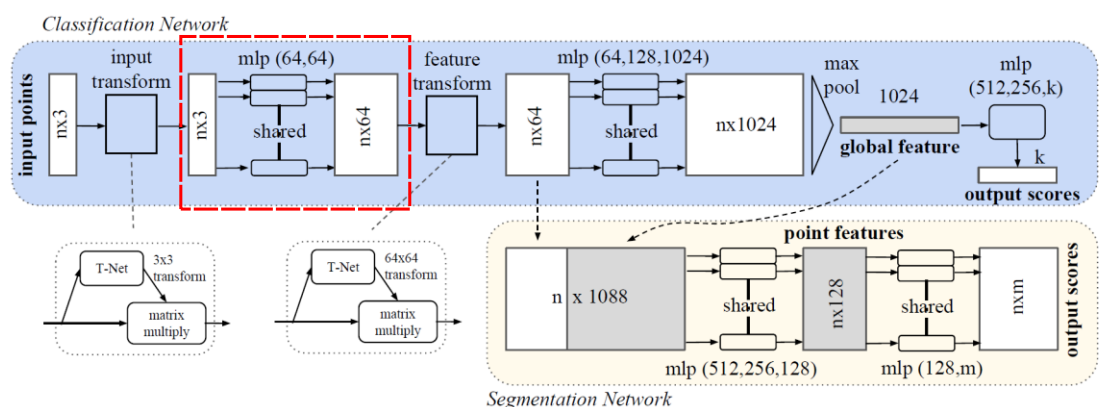


Abbildung 2.6: Zweiter Schritt - Feature Building

Die vom MLP berechneten 64 Werte sind für jeden Punkt einerseits unterschiedlich. Doch durch die Verwendung geteilter Gewichte und der daraus folgenden Verknüpfung aller Punkte enthalten diese Werte lokale Informationen über die Punktwolke. Räumlich nah beieinander liegende Punkte besitzen beispielsweise mathematisch ähnliche Feature-Vektoren.

Allgemein entspricht dieser Schritt der PointNet-Architektur der in Abschnitt 1.4 beschriebenen Abstraktion und Umwandlung des Inputs in eine für die weitere Verarbeitung geeignete Form.

Placeholder!

TODO: Visuelle Darstellung der Abstraktion via TSNE-Plot

Platzhalter Grafik 2.6: TSNE-Plot

- An dieser Stelle einen 3D-Plot der Punktwolke und einen 2D-Plot der Features nach <https://towardsdatascience.com/neural-network-embeddings-explained-4d028e6f0526>

- es werden drei Punkte aus der Wolke ausgewählt und dann in der 2D-Map hervorgehoben

### 2.1.3. Feature Transform

Die Feature-Transformation im PointNet – dargestellt in der Abbildung 2.7 – entspricht im Grunde dem unter 2.1.1 beschriebenen ersten Schritt des Netzwerks. Auch hier ist die individuelle Ausrichtung bzw. Normalisierung der Feature-Vektoren mithilfe einer Matrixmultiplikation das Ziel. Da die Dimensionen der Feature-Transformationsmatrix (64x64) um ein Vielfaches höher sind als bei der Input-Transformation (3x3), gestaltet sich die Optimierung wesentlich schwieriger. Um den Prozess zu stabilisieren und bessere Ergebnisse zu ermöglichen, wurde folgender Term zur Regularisierung der Verlustfunktion hinzugefügt [1, p. 4]:

$$L_{reg} = ||I - AA^T||^2 \quad (2.1)$$

Hierbei entspricht  $I$  der Einheitsmatrix und  $A$  der vom T-Net bestimmten Transformationsmatrix.

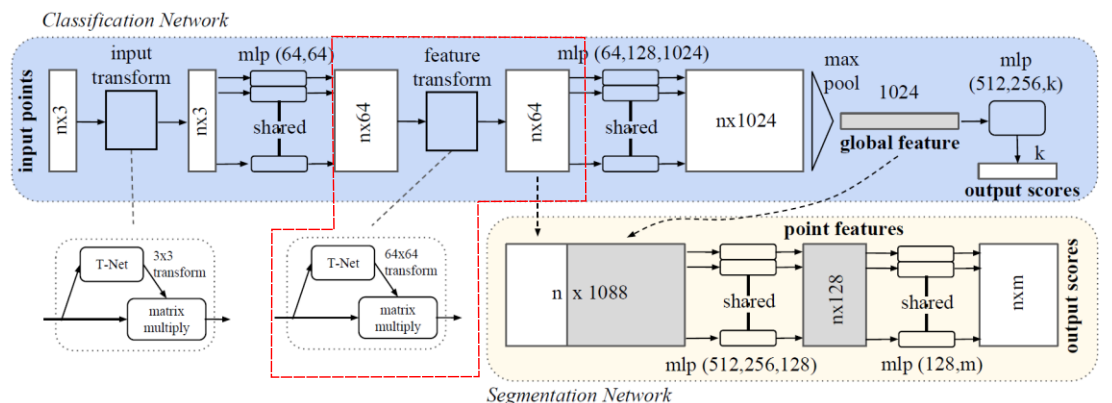


Abbildung 2.7: Dritter Schritt - Feature Transformation

### 2.1.4. Zwischenspeicher und erneutes Feature Building

Bevor der Input weiterverarbeitet wird, findet nach der Feature-Transformation eine Zwischenspeicherung der Daten statt. Dieser Schritt kann als Schnittstelle verstanden werden. Er ist streng genommen Teil des Segmentierungs-Netzwerks und wird in Abschnitt 2.2 erläutert. Im Anschluss an die Zwischenspeicherung werden die Feature-Vektoren der einzelnen Punkte erneut durch einen MLP

mit geteilten Gewichten auf einen höherdimensionalen Raum abgebildet. Dieser MLP wird in Abbildung 2.8 dargestellt und verfügt über drei Schichten mit 64, 128 und schließlich 1024 Elementen.

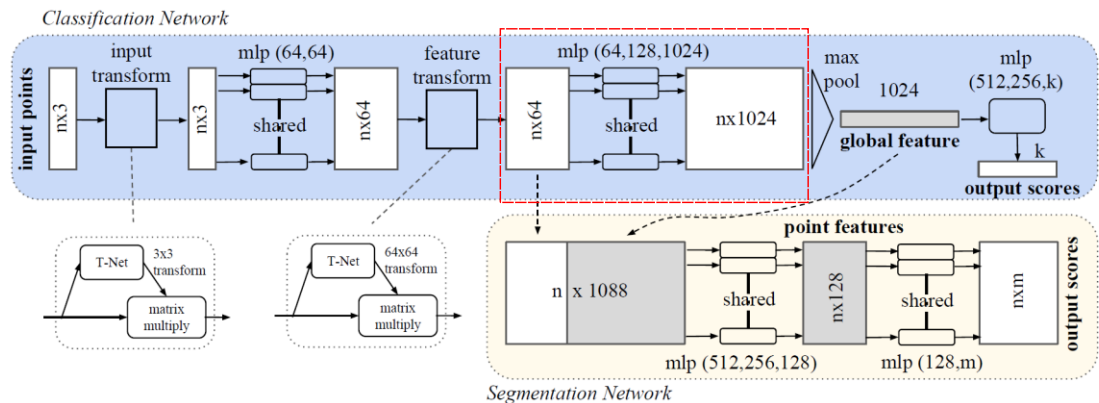


Abbildung 2.8: Vierter Schritt - erneutes Feature Building [1, p. 3]

Wie in der vorherigen Iteration des Feature Building ist auch in diesem Schritt die Erzeugung von Eigenschaftswerten zur weiteren Analyse das Ziel. Der Anstieg von 64 auf 1024 Werte pro Punkt bedeutet hierbei, dass lokale Eigenschaften und Nachbarschaftsbeziehungen wesentlich feingranularer dargestellt werden können.

### 2.1.5. Max-Pooling

In der Pooling-Schicht des PointNet werden die erzeugten Features zu einer globalen Signatur aggregiert. Hierbei wird die Max-Funktion genutzt, um die 1024 Feature-Werte aller Punkte Index für Index zu vergleichen und die jeweiligen größten Werte herauszufiltern. Abbildung 2.9 zeigt die Pooling-Schicht im PointNet-Workflow.

Da es sich bei der Max-Funktion um eine symmetrische Funktion handelt, ist die Reihenfolge der zu vergleichenden Werte irrelevant. In Verbindung mit den vorangegangenen Schichten bedeutet dies, dass im Rahmen des Max-Pooling für die Punktwolke eines Objektes unabhängig von der Reihenfolge ihrer Elemente und etwaiger Transformationen dieselbe globale Feature-Signatur ausgegeben wird.

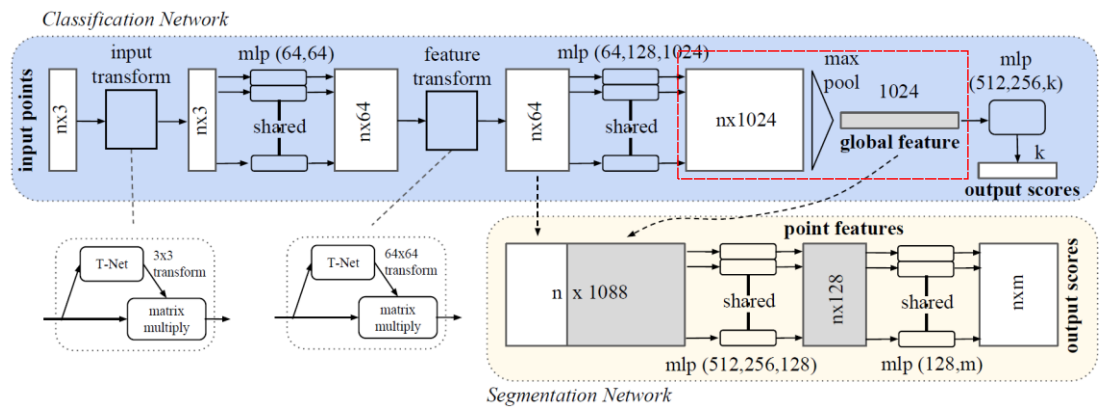


Abbildung 2.9: Fünfter Schritt - Max-Pooling

Ein weiterer interessanter Aspekt ist die Existenz von spezifischen Grenzmengen. Die erzeugte Feature-Signatur eines Objektes ist in einem gewissen Intervall der Vollständigkeit und Dichte seiner Punktwolke stets gleich. Die untere Grenze des Intervalls wird hierbei als kritische Punktmenge bezeichnet, die obere als obere Grenzform [1, p. 5]. Abbildung 2.10 zeigt eine Visualisierung dieser Grenzen anhand von vier Beispielen. Die Farbkodierung dient der Darstellung von Tiefeninformationen.

Anhand der Grafik wird deutlich, dass PointNet aufgrund der Existenz solcher Grenzmengen einen entscheidenden Vorteil besitzt: Das Netzwerk ist robust gegenüber unvollständigen Datensätzen. Diese Robustheit ist so stark, dass Qi et al. empirisch selbst bei Datensätzen mit 60 Prozent fehlenden Daten eine Genauigkeit von 80 Prozent nachweisen konnten [1, p. 10].

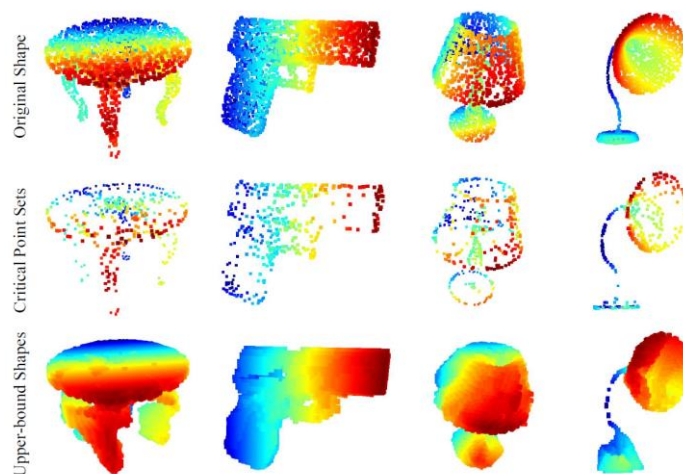


Abbildung 2.10: Kritische Punktmenge und obere Grenzform [1, p. 8]

## 2.1.6. Klassifizierung

Bevor in diesem Abschnitt auf den finalen Schritt des Klassifizierungsnetzwerks eingegangen wird, ist ein kurzer Rückblick auf die vorangegangenen Abschnitte sinnvoll.

Das Ziel des Netzwerks ist es, ein durch eine Punktwolke repräsentiertes Objekt invariant gegenüber Permutationen und Transformationen zu klassifizieren. Hierfür werden die 3D-Punktvektoren schrittweise räumlich normalisiert und zum Erzeugen von Features auf den  $\mathbb{R}^{1024}$  abgebildet. Aus den gewonnenen Features wird mittels Max-Pooling eine aus 1024 Werten bestehende globale Signatur erzeugt. Alle Schritte bis zu diesem Punkt dienen also lediglich dazu, eine beliebige Punktwolke in eine abstrahierte und standardisierte Form zu bringen, die den Anforderungen genügt.

Der eigentliche Klassifizierungsschritt, welcher in Abbildung 2.11 markiert ist, beinhaltet lediglich einen MLP. Dieser erzeugt über zwei FC-Layer mit ReLU und die Anwendung der Softmax-Funktion in der Ausgabeschicht aus den 1024 Werten der Feature-Signatur einen k-dimensionalen Vektor. Die Variable k steht hierbei für die Anzahl der Objektklassen.

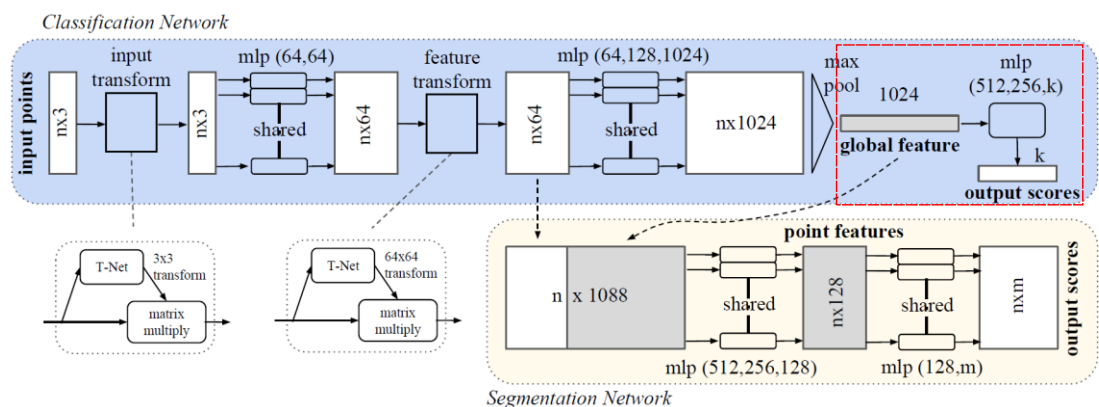


Abbildung 2.11: Sechster Schritt - Klassifizierung



## 2.2. Workflow der PointNet-Architektur: Segmentierung

Neben der simplen Klassifizierung von Objekten ist es für verschiedenste Aufgaben und Anwendungsbereiche von großer Bedeutung, die einzelnen Bestandteile eines Objektes erkennen zu können. Hierbei spricht man von der Segmentierung eines Objektes.

In diesem Abschnitt wird erläutert, wie PointNet durch eine Kombination lokaler und globaler Informationen die Segmentierung von Punktwolken ermöglicht. Der hierfür verwendete Teil des PointNet wird als Segmentierungsnetzwerk bezeichnet und in Abbildung 2.12 dargestellt.

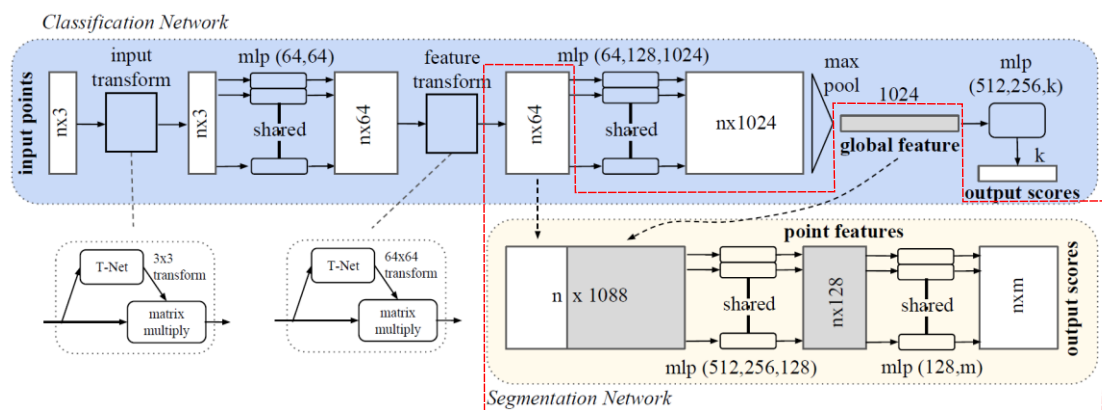


Abbildung 2.12: Segmentierungsnetzwerk

### 2.2.1. Feedback der globalen Signatur

Bei der Segmentierung muss ein KNN für jedes Element eines Objektes bestimmen, in welchem Teil des Objektes es sich befindet. Für PointNet heißt dies, dass jedem Element einer Punktwolke ein Wert bzw. eine Rolle für das entsprechende Segment zugeordnet werden muss. Für diese Zuordnung werden sowohl lokale als auch globale Informationen benötigt. Das Netzwerk muss die lokale Geometrie bzw. lokale Features kennen und benötigt die Daten des Gesamtobjekts als Kontext zu deren Einordnung.

PointNet löst diese Aufgabe über einen Feedback-Mechanismus. Wie in Abschnitt 2.1.4 beschrieben, werden die Feature-Vektoren im  $\mathbb{R}^{64}$  für die Segmentierung zwischengespeichert. Die im späteren Verlauf des Klassifizierungsnetzwerks erzeugte globale Feature-Signatur wird an jeden einzelnen Feature-Vektor angehängt. Diese Konkatination ergibt Vektoren mit 1088 Werten, die sowohl die zuvor gelernten lokalen Features als auch die globale Signatur des Objektes beinhalten. Somit ist die zuvor beschriebene Voraussetzung zur Segmentierung effizient und effektiv erfüllt [1, p. 4].

## 2.2.2. Erzeugung der Punktlables

Die konkatenierten Feature-Vektoren werden im Segmentierungsnetzwerk über einen MLP von 1088 Werten auf m-dimensionale Output-Vektoren reduziert, wobei m den möglichen Segmentklassen entspricht. Der MLP enthält insgesamt vier Schichten, die wie beim Feature Building des Klassifizierungsnetzwerks über geteilte Gewichtungsparemeter und keine Verknüpfungen zwischen benachbarten Punkten verfügen. Dargestellt wird dieser Teil des Workflows in Abbildung 2.13.

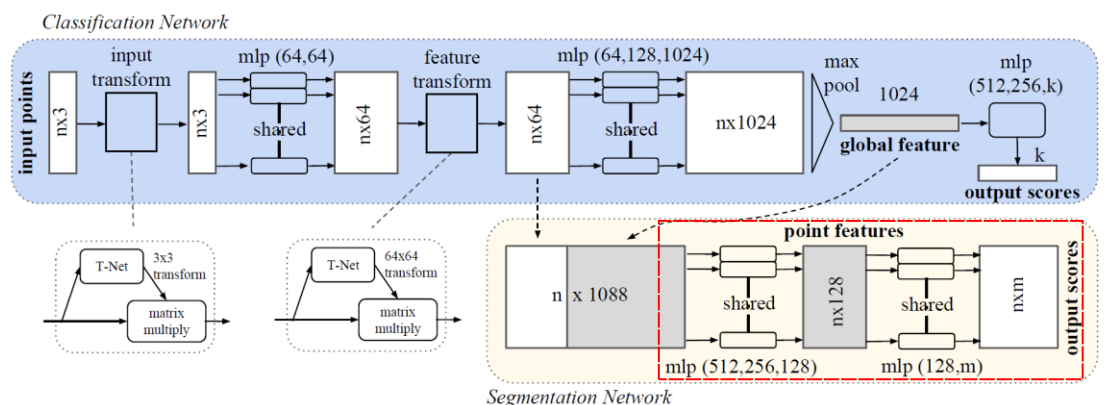


Abbildung 2.13: Erzeugung von Punktlables

Neben der Segmentierung erlaubt dieser Teil des Netzwerks zudem die Bestimmung weiterer Werte, welche sowohl von der lokalen Geometrie als auch der globalen Semantik abhängen. So zeigen Qi et al., dass mithilfe von PointNet auch die Normalen aller Elemente einer Punktwolke akkurat bestimmt werden können [1, p. 4].

## 2.3. Workflow der PointNet-Architektur: Semantik

Die semantische Analyse von 3D-Szenen ist zwar nicht per se Teil von PointNet. Allerdings kann das Segmentierungsnetzwerk ohne größere Umstände erweitert und für die Segmentierung von Szenen anpassen. Hierfür werden als Punktlables anstelle von Objektteilen ganze Objekte vergeben. Anders ausgedrückt: Anstatt eines Objektes wird beispielsweise die Punktwolke eines Zimmers als Gesamtheit betrachtet. Den Elementen der Punktwolke werden Objektklassen zugeordnet. Auf diese Weise ist PointNet in der Lage, die einzelnen Bestandteile einer Szene zu lokalisieren und zu klassifizieren [1, p. 6ff.].

Abbildung 2.14 zeigt die Visualisierung dreier Beispiele für die semantische Segmentierung von 3D-Szenen. Neben der Eingabe-Punktwolke werden das Ergebnis von PointNet (pred) und das vorgegebene Ergebnis (GT) im Vergleich dargestellt. Die einzelnen semantischen Klassen werden mit unterschiedlichen Farben dargestellt: Stühle in Rot, Tische in Lila, Sofas in Orange, Tafeln in Grau, Buchschränke in grün, Böden in Blau, Fenster in Violet, Säulen in Magenta, Türen in Khaki und nicht identifizierbare Objekte in Schwarz. Hierbei ist zu beachten, dass die Fehler der Eingabedaten (bspw. der Stuhl im zweiten Beispiel) absichtlich eingebaut wurden, um die Robustheit von PointNet gegenüber unvollständigen Punktwolken zu zeigen [1, p. 19].

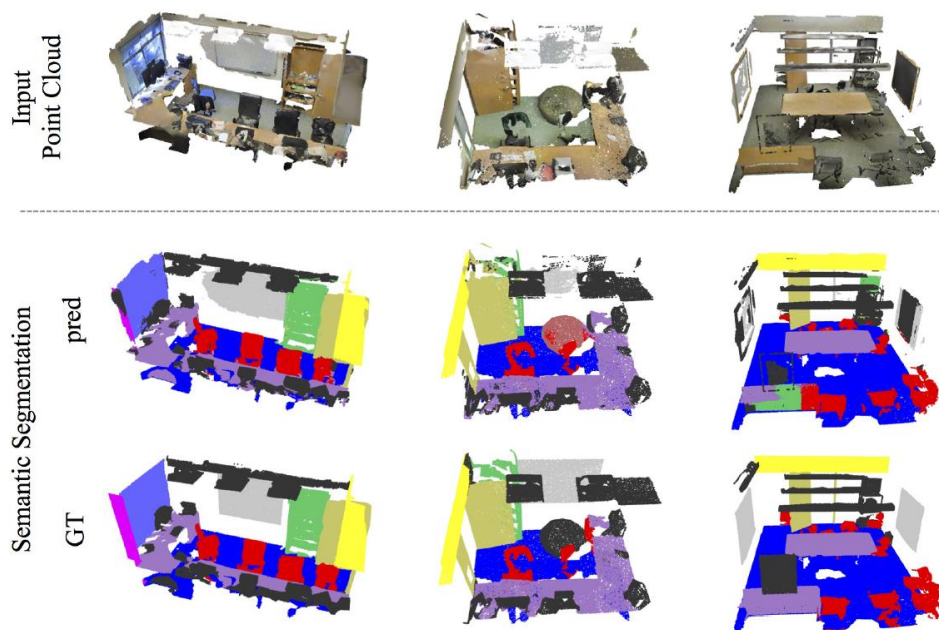


Abbildung 2.14: Beispiel für semantische Analyse von Szenen

### 3. PointNet in der Praxis – Implementation mit Keras

TODO: Text

- Original-Implementation der Autoren mit Tensorflow
- Nachimplementation in Keras auf Basis von <https://github.com/garyli1019/pointnet-keras>
- Grund für die Keras-Version: Kompakter, verständlicher
- Implementation wird in einem Jupyter Notebook nachgebaut (link zum Notebook)
- Anschließend statische Version des Notebooks

## 4. Anwendungsgebiete der PointNet-Architektur

TODO: Text

- PointNet existiert seit 2017
- Kurz Projekte beschreiben (falls existent), in denen PointNet genutzt wurde

### 4.1. Beispielprojekt 1

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

### 4.2. Beispielprojekt 2

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam

et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

### 4.3. Beispielprojekt 3

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.



### 5.1.2. 3D ShapeNet

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

### 5.1.3. VoxNet

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam



et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

## 5.2. Segmentierung

### 5.2.1. [27] aus Qi et al

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam





## 6. PointNet++ - eine Weiterentwicklung

TODO: Text

- Kurze Beschreibung, worin die Motivation für PointNet++ liegt (Problemstellung)
- Pipeline Beschreibung
- Unterschiede zu PointNet
- Anwendungsgebiete

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod  
 tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At  
 vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren,  
 no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit  
 amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut  
 labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam  
 et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata  
 sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur  
 sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore  
 magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo  
 dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lo-  
 rem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr,  
 sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam  
 erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.  
 Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit  
 amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy  
 eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam vo-  
 luptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd  
 gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum  
 dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invi-  
 dunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et  
 accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea  
 takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet,  
 consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et  
 dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo  
 duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est

## Platzhalter PointNet++ Pipeline

Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

## 7. Fazit

TODO: Text

Ziel der Arbeit nochmal erläutern. Vorgehensweise rekapitulieren.

Ergebnisse der Benchmarks nochmal hervorheben. Prognose für künftige Anwendungen bspw. im VR-Bereich => Überleitung BA-Thema

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

## 8. Quellenverzeichnis

- [1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," *CoRR*, vol. abs/1612.00593, 2016, [Online]. Available: <http://arxiv.org/abs/1612.00593>.
- [2] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
- [3] D. A. Pomerleau, "Efficient Training of Artificial Neural Networks for Autonomous Navigation," *Neural Computation*, vol. 3, no. 1, pp. 88–97, 1991, doi: 10.1162/neco.1991.3.1.88.
- [4] L. A. Gatys, A. S. Ecker, and M. Bethge, "A Neural Algorithm of Artistic Style," *CoRR*, vol. abs/1508.06576, 2015, [Online]. Available: <http://arxiv.org/abs/1508.06576>.
- [5] J. Heaton, *AIFH, Volume 3: Deep Learning and Neural Networks*. 2015.
- [6] W. Bibel, R. Kruse, and B. Nebel, "Computational Intelligence: Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze," 2015. [Online]. Available: <http://www.springer.com/series/12572>.
- [7] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.
- [8] F. Chollet, *Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek*. MITP-Verlags GmbH & Co. KG, 2018.
- [9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [10] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *CoRR*, vol. abs/1502.03167, 2015, [Online]. Available: <http://arxiv.org/abs/1502.03167>.



- [11] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial Transformer Networks," *CoRR*, vol. abs/1506.02025, 2015, [Online]. Available: <http://arxiv.org/abs/1506.02025>.

## Abbildungsverzeichnis

Abbildung 1.1: Aufbau eines vollständig verbundenen KNN .....	2
Abbildung 1.2: Graph der Gleichrichterfunktion $\Phi(z) = \max(0, z)$ .....	4
Abbildung 1.3: Beispiel eines "tiefen" neuronalen Netzwerks [8, p. 9]. .....	7
Abbildung 1.4: Schema der Faltungsoperation [9, p. 330]. .....	9
Abbildung 1.5: Beispiel für Max-Pooling mit einem 2x2 Filter und Schrittgröße 2 .....	10
Abbildung 2.1: Aufbau der PointNet-Architektur [1, p. 3].....	12
Abbildung 2.2: Klassifikationsnetzwerk der PointNet-Architektur [1, p. 3].....	13
Abbildung 2.3: Erster Schritt - Input-Transformation mittels T-Net [1, p. 3] .....	14
Abbildung 2.4: Beispiel für die Funktion eines STN [11, p. 2] .....	15
Abbildung 2.5: Aufbau T-Net (Placeholder! TODO: eigene Darstellung).....	15
Abbildung 2.6: Zweiter Schritt - Feature Building .....	16
Abbildung 2.7: Dritter Schritt - Feature Transformation .....	18
Abbildung 2.8: Vierter Schritt - erneutes Feature Building [1, p. 3].....	19
Abbildung 2.9: Fünfter Schritt - Max-Pooling .....	20
Abbildung 2.10: Kritische Punktmenge und obere Grenzform [1, p. 8].....	20
Abbildung 2.11: Sechster Schritt - Klassifizierung.....	21
Abbildung 2.12: Segmentierungsnetzwerk .....	22
Abbildung 2.13: Erzeugung von Punktlables .....	23
Abbildung 2.14: Beispiel für semantische Analyse von Szenen .....	24

## Anhang