



**BERGISCHE  
UNIVERSITÄT  
WUPPERTAL**

Fakultät für Elektrotechnik  
Informationstechnik und Medientechnik

---

Lehrstuhl für Allgemeine Elektrotechnik und Theoretische Nachrichtentechnik

# Master-Thesis

**Untersuchung von Deep Learning Methoden zur  
Rauschunterdrückung bei gestörten Sprachsignalen**

Felix Schürmann

1110259

Elektrotechnik

Informations- und Kommunikationstechnik

Wuppertal, den 03. August 2020

Erstgutachter:

Prof. Dr. Ing. Anton Kummert

Zweitgutachter:

Prof. Dr. rer. nat. U. Pfeiffer



## **MASTER-STUDIENGANG ELEKTROTECHNIK**

### **THEMA FÜR DIE MASTER-THESIS**

Kandidat: **Felix Schürmann**

Matrikelnummer: **1110259**

Betreuer: **Prof. Dr.-Ing. A. Kummert**

**Thema: Untersuchung von Deep Learning Methoden zur Rauschunterdrückung bei gestörten Sprachsignalen**

#### **Erläuterungen:**

In vielen Bereichen gewinnen Machine Learning bzw. Deep Learning Verfahren zunehmend an Bedeutung. Auch im Bereich der Rauschunterdrückung, einem häufig untersuchten Thema im Bereich der Signalverarbeitung, kann die Anwendung von Künstlicher Intelligenz Erfolge erzielen.

Im Rahmen der Arbeit soll der Kandidat die Möglichkeit der Rauschunterdrückung bei gestörten Sprachsignalen mittels künstlicher neuronaler Netze untersuchen. Hierzu soll zunächst der Stand der Technik dokumentiert und darauf aufbauend eine Evaluation der Deep Learning Architekturen an Hand von Sprachdatensätzen, die verschiedene Rauschquellen enthalten, durchgeführt werden. Die Ergebnisse sind mit verschiedenen bestehenden Methoden zu vergleichen und zu evaluieren. Die Ergebnisse sind in geeigneter Form zu dokumentieren und zu diskutieren.

Wuppertal, 01. Februar 2020

(Unterschrift)

Erstgutachter: **Prof. Dr.-Ing. A. Kummert**

Zweitgutachter: **Prof. Dr. rer. nat. U. Pfeiffer**

---

#### **Prüfungsamt:**

Kennziffer: **201 THM ET 5 KÜ**

Ausgabedatum: **1.2.2020**

Abgabedatum: **3.8.2020**

---

(Unterschrift)

## **Eidesstattliche Erklärung**

Hiermit versichere ich, dass ich die Arbeit selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Wuppertal, den 03. August 2020

---

(Unterschrift)

## **Einverständniserklärung**

Ich bin damit einverstanden, dass meine Abschlussarbeit wissenschaftlich interessierten Personen oder Institutionen zur Verfügung gestellt werden kann. Korrektur- oder Bewertungshinweise in meiner Arbeit dürfen nicht zitiert werden.

Wuppertal, den 03. August 2020

---

(Unterschrift)

## Kurzfassung

Sprachverbesserung ist aufgrund des steigenden Bedarfs und Anspruchs in modernen Kommunikationssystemen sowie Systemen zur automatischen Spracherkennung (*ASR*) weiterhin ein aktives Forschungsfeld. Diese Systeme sollen in vielseitigen akustischen Umgebungen eingesetzt werden können. Auf Deep Learning basierende Methoden bieten Lösungen für die Nachteile, die aus klassischen statistisch-basierten Algorithmen herrühren. In dieser Thesis wird daher die historische Entwicklung der Sprachverbesserung, angefangen mit spektraler Subtraktion und den Algorithmen nach Eprahim und Malah, sowie von state-of-the-art Systemen wie DeepXi und SEGAN, aufgezeigt. Weiter werden die Grundlagen moderner Sprachverbesserung basierend auf Maskierungsmethoden wie der Ideal Ratio/Binary Mask (*IRM/IBM*), sowie häufig eingesetzte Deep Learning Techniken, hergeleitet. Hierzu zählen unter anderem *ResNets*, *LSTMs* und *dilatierte CNNs*. Danach werden, darauf aufbauend, neuartige Architekturen vorgestellt und anhand von objektiven Metriken wie *PESQ*, *STOI* und *SDR* evaluiert. Hierbei wird gezeigt, dass eine Subdatensatz Trainingsstrategie mit Lernratendämpfung *PESQ* Metriken um 0.25 Punkte im Vergleich zu Standard Trainingsstrategien verbessern kann. Weiterhin werden angepasste Fehlerfunktionen, wie der *PMSQE*, in einer Fahrzeugumgebung mit hoher spektraler Varianz angewendet. Hier wird eine Adaption der Fehlerfunktion für die Nutzung in maskierungsbasierten Netzen vorgestellt, welche für die genannte Umgebung, in der MSE Fehlerfunktionen scheitern können, eine Verbesserung der Sprachqualität erreicht. Weiter wird eine Änderung des Trainingsziels zum a-priori SNR und Post-Processing Methoden wie der *Global Variance Equalization* untersucht. Abschließend werden die Ergebnisse zusammengefasst und mit state-of-the-art Systemen verglichen, sowie ein Ausblick in Zukunftsperspektiven zur Sprachverbesserung gegeben.

## Abstract

Speech enhancement remains an active research topic as its demand increased for modern communication systems as well as automatic speech recognition (ASR) in ever more versatile acoustic environments. Deep learning based artificial neural networks offer solutions to the drawbacks of classic statistical based algorithms. This thesis herefore illustrates the historic development of speech enhancement starting with spectral subtraction and the algorithms of Eprahim and Malah and going further to state-of- the-art DL systems like DeepXi and SEGAN. Furthermore the fundamentals of modern speech enhancement, such as Ideal Ratio/Binary Masks (IRM/IBM) are derived aswell as deep learning techniques which are commonly used in speech enhancement naming ResNets, LSTMs and dilated CNNs among others. Hereafter novel architectures based on these are proposed and evaluated with objective metrics such as PESQ, STOI and SDR. Here it is shown that a training strategy based on sub-datasets with learning rate decay can improve PESQ metrics up to 0.25 in contrast to standard training strategies. Furthermore adjusted loss functions such as PMSQE are evaluated in high spectral variance vehicular environments. Accordingly an adaptation of the loss function for use in masking-based systems is proposed for the mentioned environment in which MSE loss functions may fail. Furthermore a formulation of the a-priori SNR as training target and post processing methods such as the global variance equalization are shown and evaluated. Finally the findings are summarized and compared to state-of-the-art systems. Moreover future prospects for speech enhancement are given.

# Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>1</b>
<b>2 Grundlagen</b>	<b>2</b>
2.1 Rauschunterdrückung mit klassischen Methoden der Signalverarbeitung	2
2.1.1 Spektrale Subtraktion . . . . .	2
2.1.2 Algorithmen nach Ephraim und Malah . . . . .	4
2.2 Künstliche neuronale Netze . . . . .	6
2.2.1 Aufbau . . . . .	6
2.2.2 Funktionsweise . . . . .	8
2.2.3 Netze für die Zeitreihenanalyse . . . . .	10
2.3 Rauschunterdrückung mit künstlichen neuronalen Netzen . . . . .	17
2.3.1 Methoden . . . . .	17
2.3.2 Evaluierungsmetriken . . . . .	20
2.3.3 Stand der Technik . . . . .	22
<b>3 Datenvorverarbeitung</b>	<b>28</b>
3.1 Kontextfenster . . . . .	29
3.2 Skalierung . . . . .	30
3.3 Datensatz . . . . .	30
<b>4 Untersuchung von Ideal Binary Masks mit LSTM Ansatz</b>	<b>31</b>
4.1 Gewichtung der Klassen . . . . .	32
4.2 Subband Klassifikation . . . . .	33
4.3 Diskussion . . . . .	35
<b>5 Ideal Ratio Mask Störunterdrückung mit Residual CNN</b>	<b>37</b>
5.1 Architektur . . . . .	37
5.2 Trainingsstrategie . . . . .	39
5.3 Analyse und Ergebnisse . . . . .	41
5.4 Anpassung der Architektur . . . . .	47
<b>6 Ideal Ratio Mask Störunterdrückung mit Dilated CNN+LSTM</b>	<b>48</b>
6.1 Architektur . . . . .	48
6.2 Analyse und Ergebnisse . . . . .	49
6.3 Convolution auf der Frequenzachse . . . . .	52

6.4	Dilated-Residual-CNN-LSTM . . . . .	53
<b>7</b>	<b>Ideal Ratio Mask Störunterdrückung mit DenseNet Implementierung</b>	<b>54</b>
7.1	DenseRNet . . . . .	55
<b>8</b>	<b>Anpassen der Fehlerfunktion</b>	<b>56</b>
8.1	PMSQE . . . . .	57
<b>9</b>	<b>Störunterdrückung mit A-priori SNR als Trainingsziel</b>	<b>60</b>
9.1	Parametrisierter Wiener Filter . . . . .	61
<b>10</b>	<b>Post-Processing</b>	<b>62</b>
10.1	Global Variance Equalization . . . . .	62
10.1.1	Untersuchung der Varianz . . . . .	63
10.1.2	Anwendung und Ergebnisse . . . . .	65
10.1.3	Diskussion . . . . .	67
10.2	Post-Gain . . . . .	68
<b>11</b>	<b>Auswertung und Vergleich</b>	<b>69</b>
<b>12</b>	<b>Zusammenfassung und Ausblick</b>	<b>73</b>
<b>A</b>	<b>Literaturverzeichnis</b>	<b>80</b>
<b>B</b>	<b>Quellenangaben</b>	<b>85</b>

# 1 Einleitung

Moderne Kommunikationstechnologie hat Einzug in unser alltägliches Leben gefunden. Nicht nur in der Freizeit, sondern auch im Berufsleben werden wir von Geräten begleitet, die eine ständige Kommunikation mit einer Vielzahl an Personen in Echtzeit ermöglichen. Gerade in der Arbeitswelt erleben wir eine ergonomische Transformation. Möglichkeiten zur Nutzung digitaler Strukturen eröffnen die Option eines Arbeitens von zu Hause, oder jedem anderen möglichen Ort. Besonders hier muss eine Kommunikation geschaffen werden, die stabil, frei von Störungen und damit frei von Ermüdung, ein effizientes Arbeiten ermöglicht. Bislang erleben wir Beeinträchtigungen dieser Kommunikation, geschaffen durch Hintergrundstörungen, welche die Verständlichkeit übertragener Sprache reduzieren, oder durch fortwährenden Einfluss zur Ermüdung des Gesprächspartners führen.

Nicht nur bei der Mensch-zu-Mensch Kommunikation beobachten wir die Notwendigkeit ungestörter Übertragung von Information. Der Einfluss von Maschinen, die in der Lage sind über eine Spracherkennung Befehle entgegen zu nehmen, nimmt ebenfalls immer mehr Einfluss auf unser Leben. So kann es an dieser Stelle auch zu sicherheitskritischen Anwendungen kommen, wie in einer lauten Industrieumgebung, beim Einsatz von Rettungskräften, oder im Automobil. An diesen Stellen ist eine Aufbereitung und Enstörung von Sprache notwendig, um die Automatisierung voranzutreiben. Zu weiteren Anwendungsfeldern von Sprachverbesserungssystemen zählen außerdem Hörgeräte und biometrische Erkennungssysteme.

Die Reduktion von Rauschen und Störungen in monauralen Signalen gehören zu den Feldern der Informationstechnologie mit langer Forschungshistorie. Bereits seit den 1970er Jahren existieren Systeme zur Störreduktion, häufig basierend auf statistischen Methoden der Signalverarbeitung [1]. Diese Methoden bieten jedoch keine gute Performanz bei nicht stationären Störungen, also Störungen die sich mit der Zeit ändern. Für diese Problematik können tiefe neuronale Netze eine deutliche Verbesserung der Entstörleistung bewirken, im Gegensatz zu klassisch statistischen Methoden [2]. Besonders die gestiegene Rechenkapazität befähigt moderne Computer zum Trainieren komplexer Netze, sowie zu deren Inferenz in Echtzeit. Der Bereich des Deep Learning hat in jüngster Zeit viele Durchbrüche erlebt, vor allem im Bereich der Bildverarbeitung und Klassifizierung. Für diese Arbeit wird erörtert, welche Deep Learning Methoden eine Sprachverbesserung erreichen können. Hierfür wird der aktuelle Stand der Technik dokumentiert, sowie verschiedene eigene Ansätze bzw. Abstraktionen vorhandener Netze implementiert und evaluiert.

## 2 Grundlagen

In diesem Kapitel werden die Grundlagen der Störunterdrückung mit klassisch statistischen Methoden, sowie die Grundlagen der Störunterdrückung mit neuronalen Netzen erläutert. Weiterhin werden state-of-the-art Systeme vorgestellt.

### 2.1 Rauschunterdrückung mit klassischen Methoden der Signalverarbeitung

Störunterdrückung mit statistischen Methoden ist ein Forschungsbereich, der über die letzten Dekaden ausführlich untersucht wurde. Die grundlegenden Methoden hierfür werden im Folgenden erläutert.

#### 2.1.1 Spektrale Subtraktion

Die Grundlagen für Rauschunterdrückung in der Sprachsignalverarbeitung bilden die Methoden, die sich unter dem Begriff *spektrale Subtraktion* wiederfinden. Hierbei wird das geschätzte rauschreduzierte Spektrum  $\hat{X}(j\omega)$  aus dem Spektrum des Ausgangssignals  $Y(j\omega)$  und einer frequenzabhängigen Gain Funktion  $G(\omega)$  gebildet [1].

$$\hat{X}(j\omega) = G(\omega)Y(j\omega) \quad (2.1)$$

Tabelle 2.1 zeigt die gängigsten Gain-Funktionen aus dem Bereich der statistischen Signalverarbeitung.  $|\hat{D}(j\omega)|^2$  beschreibt hierbei die geschätzte mittlere Rauschleistung. Der Parameter  $\alpha$  kann zur Abstimmung verwendet werden.

---

$G(j\omega) = 1 - \alpha \frac{ \hat{D}(j\omega) }{ Y(j\omega) }$	$G(j\omega) = \sqrt{1 - \alpha \frac{ \hat{D}(j\omega) ^2}{ Y(j\omega) ^2}}$
<b>Amplitudensubtraktion</b>	<b>Spektralsubtraktion</b>
$G(j\omega) = 1 - \alpha \frac{ \hat{D}(j\omega) ^2}{ Y(j\omega) ^2}$	$G(j\omega) = f(SNR_{prio}, SNR_{post})$
<b>Wienerfilter</b>	<b>Ephraim-Malah-Filter</b>

---

Tabelle 2.1: Beispiele Übertragungsfunktionen

Abbildung 2.1 zeigt das Prinzip der Spektralen Subtraktion. Das Eingangssignal  $y(t)$  wird mit einer schnellen Fourier Transformation in den Frequenzbereich transformiert und anschließend wird das Betragsquadrat, welches die spektrale Leistungsdichte erzeugt, gebildet. Danach wird eine Schätzung der Leistung des Rauschspektrums von der Leistung des transformierten Eingangssignals subtrahiert. Weiter stehen Methoden zur Nachbearbeitung zur Verfügung, um zuletzt das Spektrum mittels Inverser Fourier Transformation wieder in den Zeitbereich zu wandeln. Hierfür werden die Phaseninformationen aus dem eingangs bezogenen gestörten Signal verwendet.

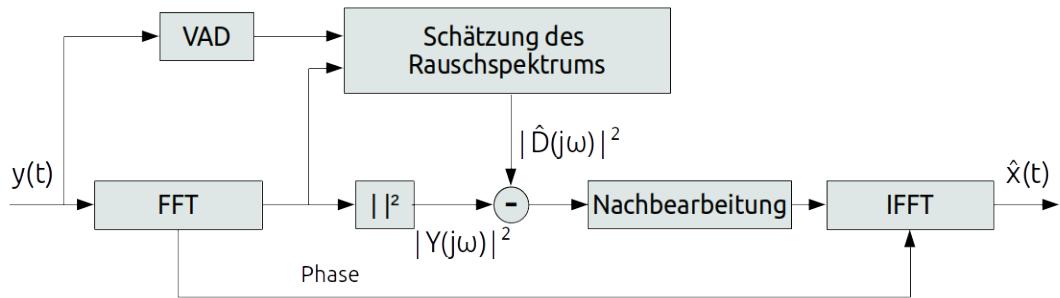


Abbildung 2.1: Spektrale Subtraktion

Optional findet sich in solchen Systemen häufig ein *Voice Activity Detector (VAD)*, der erkennen soll, zu welchen Zeitpunkten des Signals eine Sprachübertragung stattfindet. Dies wird verwendet um statistische Aussagen über die Verteilung der Rauschenergie während Sprachpausen machen zu können und um die Ausgabe in Sprachpausen herunter zu regeln. Häufig wird die Leistung des Rauschsignals über den Mittelwert des Quadrates der Rauschamplitude geschätzt [1]. Beim Einsatz der spektralen Subtraktion kommt es zur Bildung von Artefakten, die als *musical noise* bezeichnet werden. Diese entstehen dadurch, dass unter realen Bedingungen das Rauschspektrum nicht stationär ist. Dieses verändert sich zeitlich im Vergleich zur Referenzaufnahme, die während einer Sprachpause gemacht wurde. Die nach der Subtraktion auftretenden Differenzen zwischen geschätztem und wahrem Spektrum führen zu steilen Amplitudensprüngen im Bereich einzelner Frequenzen. Diese Artefakte werden als so störend empfunden, so dass es hierdurch zu keinem zufriedenstellenden Ergebnis kommt. Auch wenn das Rauschen reduziert wird, kann die Verständlichkeit der Sprache nicht verbessert werden. Über die Jahre wurden verschiedene Methoden der Nachbearbeitung vorgestellt, die musical noise zu reduzieren versuchen, in dem z.B. die Eigenschaften des menschlichen Hörens mit einbezogen werden, oder die Spektrale Subtraktion mit Hilfe von Wavelet-Paketen durchgeführt wird [3].

### 2.1.2 Algorithmen nach Ephraim und Malah

Im Gegensatz zu anderen vorgestellten Methoden, kann mit den Systemen von Ephraim und Malah, aus den Jahren 1983 bis 1985, die Entstehung von *musical noise* weitgehend vermieden werden. Grundstein dieser Systeme bildet ein *Minimum-Mean-Squared-Error (MMSE)*-Schätzer (dt. Methode der kleinsten Fehlerquadrate). Dieser schätzt die kurzzeitige spektrale Amplitude (engl. *Short-Time-Spectral-Amplitude (STSA)*), basierend auf der Annahme, dass sich die Spektralkomponenten von Sprache und Rauschen wie statistisch unabhängige gaußsche Variablen verhalten [4]. Die Gain Funktion wird hierbei durch das *a priori* und *a posteriori* Signal-Rausch-Verhältnis bestimmt. Diese beiden Parameter müssen für jeden zeitlichen Frame  $t$  und jede spektrale Komponente  $f$  berechnet werden. Diese werden im weiteren Verlauf der Arbeit als TF-Slots bezeichnet.

$$SNR_{post}(t, f) = \frac{|X(t, f)|^2}{|D(t, f)|^2} \quad (2.2)$$

Formel 2.2 beschreibt das *a posteriori* Signal-Rausch-Verhältnis und ist eine Schätzung aus den Daten des aktuellen Frames [3].

$$SNR_{prio}(t, f) = (1 - \alpha)P[SNR_{post}(t, f)] + \alpha \frac{|G(t - 1, f)X(t - 1, f)|^2}{|D(t, f)|^2} \quad (2.3)$$

Das a priori SNR ist in Formel 2.3 beschrieben. Hier wird der aktuelle Frame mit einer Gewichtung von  $1 - \alpha$  und der letzte rauschreduzierte Frame mit der Gewichtung  $\alpha$  zum Bezug genommen.  $P[X] = x$  wenn  $x \geq 0$  und  $P[x] = 0$  sonst. Ephraim und Malah sowie Cappé setzen den Parameter  $\alpha$  zu 0.98. Es ergibt sich eine hohe Dämpfung bei niedrigem  $SNR_{prio}$  und eine niedrige Dämpfung bei hohem  $SNR_{prio}$ .  $SNR_{post}$  zeigt sich hier als korrigierender Parameter vor allem bei niedrigem a priori SNR [5].

In der Notation von Cappé [5] ist die Gain Funktion für den MMSE-STSA Schätzer gegeben als:

$$G = \frac{\sqrt{\pi}}{2} \sqrt{\left(\frac{1}{1 + SNR_{post}}\right) \left(\frac{SNR_{prio}}{1 + SNR_{prio}}\right)} \cdot M[(1 + SNR_{post}) \left(\frac{SNR_{prio}}{1 + SNR_{prio}}\right)] \quad (2.4)$$

Die Gain Funktion  $G(t, f)$  wird hierbei mit jedem Zeit-Frequenzslot des Eingangsspektrums  $X(t, f)$  multipliziert.

$$M[\Theta] = \exp\left(-\frac{\Theta}{2}\right) \left[ (1 + \Theta) I_0\left(\frac{\Theta}{2}\right) + \Theta I_1\left(\frac{\Theta}{2}\right) \right] \quad (2.5)$$

Die  $M[\theta]$  Funktion aus Formel 2.4 ist in Formel 2.5 beschrieben, wobei  $I_0$  und  $I_1$  für die Besselfunktionen nullter und erster Ordnung stehen [5].

Hierauf aufbauend wurden die Algorithmen nach Ephraim und Malah um einen Parameter erweitert, der die Wahrscheinlichkeit für die Abwesenheit von Sprache in einem jeweiligen Zeit-Frequenz Slot in Betracht zieht. Außerdem wurde der Algorithmus im Weiteren so angepasst, dass dieser den mittleren quadratischen Fehler des logarithmierten Spektrums minimiert, da dieses für das subjektive Empfinden der entrauschten Sprache bedeutungsvoller ist [3]. Zu erklären ist dies damit, dass das menschliche Lautstärkeempfinden auch einer logarithmischen Funktion in Abhängigkeit des Schalldrucks folgt. In [6] beschreiben Ephraim und Malah, dass die Qualität des Entrauschens bei Verwendung des logarithmierten Spektrums signifikant verbessert wird.

Im weiteren Verlauf der Arbeit wird gezeigt, dass die vorgestellten Gain Funktionen auch heute zum Einsatz kommen. So werden z.B. in Nicolson's DeepXi [7] die a priori und a posteriori SNR mit Hilfe eines neuronalen Netzes gebildet, um anschließend verschiedene Gain Funktionen, wie den logarithmischen spektralen Amplituden Schätzer MMSE-LSA, oder einen Wiener Filter anzuwenden.

## 2.2 Künstliche neuronale Netze

Künstliche neuronale Netze (**KNN**) sind vor allem in den letzten Jahren in besonderen Fokus der Forschung gerückt. Auch wenn es bereits erste kommerzielle Anwendungen in den 1970er Jahren gab, erlangte das Forschungsfeld erst über das letzte Jahrzehnt seine heutige Bedeutung. Die Gründe hierfür sind die deutlich gestiegene Rechenkapazität, die die Nutzung großer Netze ermöglichte, sowie die Durchbrüche im Bereich der Bildverarbeitung. Einige Begrifflichkeiten aus dem Bereich der neuronalen Netze werden im Folgenden zu ihrer Verständlichkeit in der englischen Originalbezeichnung belassen.

### 2.2.1 Aufbau

Es existieren eine Vielzahl von Varianten für künstliche neuronale Netze. Für den Anwendungsfall werden vor allem vorwärtsgerichtete und rekurrente Netze verwendet.

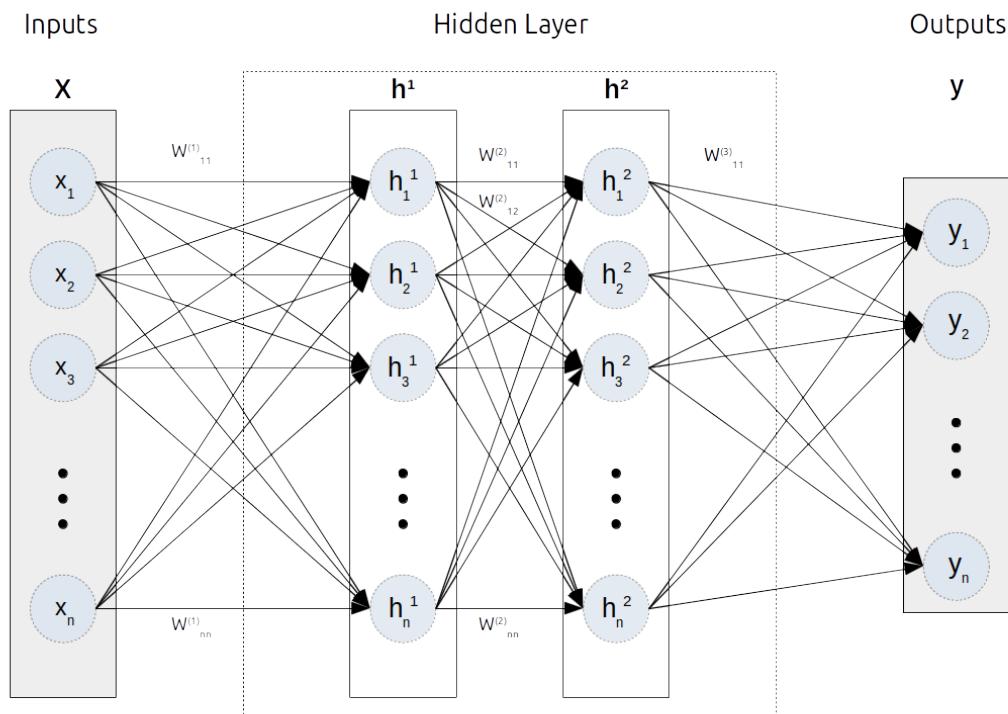


Abbildung 2.2: vorwärtsgerichtetes neuronales Netz

Abbildung 2.2 zeigt ein mehrlagiges Perzepron (engl. *multilayer perceptron (MLP)*). Ein vorwärtsgerichtetes Netz, bei dem die Informationen von der Eingangsschicht bis hin zur Ausgangsschicht in einer Richtung weitergeleitet werden. Neuronale Netze können mehrere verborgene Schichten (engl. *hidden layer*) beinhalten.

Die einzelnen Schichten bestehen aus einer gewissen Anzahl von Neuronen. Die Funktionsweise der Neuronen ist in Abbildung 2.3 verdeutlicht. Die Eingangssignale  $x_1$  bis  $x_n$  werden jeweils mit einem Gewicht  $w_1$  bis  $w_n$  multipliziert und anschließend mit der Propagierungsfunktion zusammengefasst. An dieser Stelle kann der Funktion ein *bias* Wert hinzugefügt werden. Dieser wird genutzt, um den Ausgang des Neurons anzupassen. Hiermit lässt sich die Aktivierungsfunktion verschieben bzw. vermeiden, dass Neuronen einen Nullwert behalten. Die Flexibilität des Netzes, sich den Eingangsdaten anzupassen, wird erhöht. An nächster Stelle folgt die Aktivierungsfunktion, die den Ausgang des Neurons bildet. Je nach Anwendungsfall werden verschiedene Aktivierungsfunktionen verwendet. Um das Verfahren zur Fehlerrückführung nutzen zu können (engl. *backpropagation*), ist es notwendig, dass die Aktivierungsfunktion überall differenzierbar ist [8].

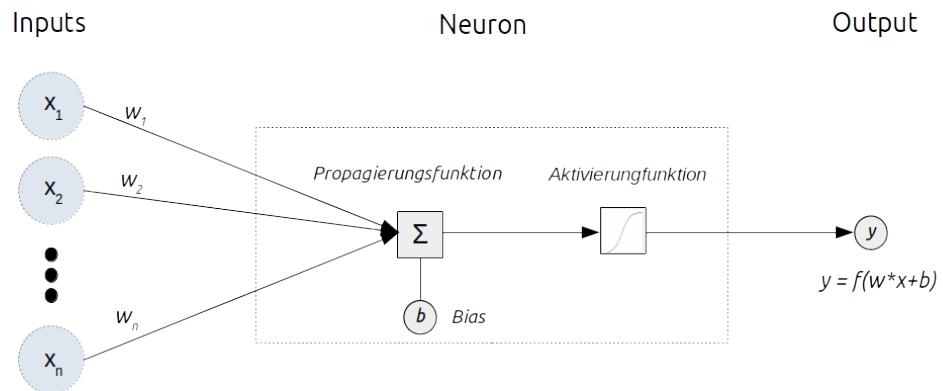


Abbildung 2.3: Neuron

Aktivierungsfunktionen, die in der Praxis gängig verwendet werden, sind in Abbildung 2.4 zu sehen. Tabelle 2.2 zeigt ihre Formeln. Der Vorteil der ReLu Funktion besteht darin, dass sich Netze hiermit schneller trainieren lassen und es bei Convolutional Neural Nets nicht zu schwindenden Gradienten kommt [28] (siehe Sektion 2.2.3). Tangenshyperbolikus als Aktivierungsfunktion sorgt dafür, dass negative Eingänge auch stark negativ abgebildet werden und Eingänge nahe Null ebenso bei Null abgebildet werden. Die Sigmoid Funktion wird häufig im Zusammenhang mit Wahrscheinlichkeiten verwendet, da sie Ausgaben zwischen 0 und 1 generiert.

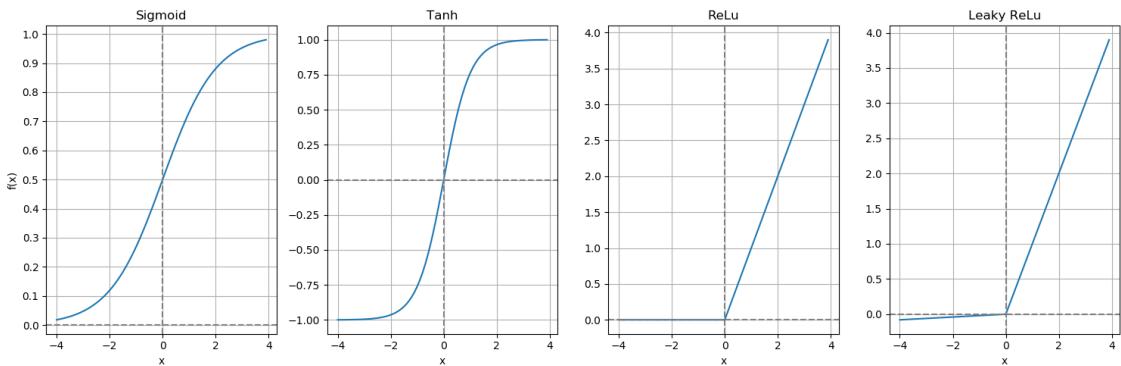


Abbildung 2.4: Aktivierungsfunktionen

Funktion	Sigmoid	Tanh	ReLU	Leaky ReLu
$f(x)$	$\frac{1}{1+e^{-x}}$	$\frac{e^x - e^{-x}}{e^x + e^{-x}}$	$\max(0, x)$	$\max(\epsilon x, x)$ mit $\epsilon \ll 1$

Tabelle 2.2: Aktivierungsfunktionen

## 2.2.2 Funktionsweise

Im Folgenden wird die Funktionsweise neuronaler Netze beschrieben. Hierbei werden die Fehlerrückführung und die Funktionsweise des Optimierers angesprochen.

### Fehlerrückführung

Künstliche neuronale Netze werden mit der bereits angesprochenen Fehlerrückführung trainiert, sofern es sich, wie im Anwendungsfall, um ein *überwachtes Lernverfahren* handelt. Hierbei werden die im Netz vorhandenen Gewichte angepasst, um die Eingangswerte möglichst nah an die vorgegebenen Ausgangswerte anzunähern. Der Vergleich mit den bekannten und richtigen Ergebnissen beschreibt dabei die „Überwachung“. Die Fehlerrückführung basiert häufig auf dem mittleren quadratischen Fehler und stellt einen Spezialfall des Gradientenverfahrens dar [A1].

Zur Beschreibung des Fehlers können verschiedene Funktionen herangezogen werden. Für den mittleren quadratischen Fehler gilt:

$$E = \frac{1}{2} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.6)$$

Um den Fehler zu minimieren wird der Gradient mit einem iterativen Prozess berechnet.

$$\nabla E = \left( \frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \dots, \frac{\partial E}{\partial w_n} \right) \quad (2.7)$$

Die Fehlerfunktion stellt sich also als eine Verkettung von Funktionen, die über die Verbindungen der Neuronen mit ihren Gewichten und ihren Aktivierungsfunktionen gebildet wird, dar. Jedes im Netz befindliche Gewicht wird entsprechend der Formel 2.8 angepasst:

$$\Delta w_i = -\gamma \frac{\partial E}{\partial w_i} \quad \text{mit } i = 1, \dots, n \quad (2.8)$$

$\gamma$  beschreibt dabei die Lernrate [9]. Diese stellt einen wichtigen einstellbaren Hyperparameter dar. Der Algorithmus wird beendet, wenn der Fehler ausreichend klein geworden ist. Zum erfolgreichen Training ist es also notwendig, Minima in der Fehlerfunktion zu finden.

KNN werden auch bei *unüberwachten Lernverfahren* genutzt. Hierbei werden Muster ausschließlich aus den Eingangsdaten gebildet. Die Muster werden dabei klassifiziert und anhand von Features zusammengefasst. Dies wird vor allem zur Segmentierung und Komprimierung von Daten eingesetzt [A10].

## Optimierer

Die richtige Wahl des Lernparameters entscheidet, ob lokale Minima gefunden werden können, oder ob der Optimierer wegen zu großer Lernrate wieder von diesen divergiert. Für die Wahl des Optimierers stehen verschiedene Varianten zur Verfügung. Der stochastische Gradientenabstieg (engl. *stochastic gradient descent (SGD)*) berechnet sich iterativ über zufällig selektierte Untermengen eines Datensatzes. Somit wird die Rechenlast deutlich reduziert, da ansonsten der Gradient über den gesamten Datensatz berechnet werden müsste [A2]. Die *Root Mean Square Propagation (RMSProp)* stellt eine Erweiterung des SGD dar, in welcher die Lernrate für jeden Parameter angepasst wird. Die Lernrate für ein Gewicht wird hier durch einen fortlaufenden Mittelwert der letzten Gradienten dividiert. Der *Adaptive Moment Estimation (Adam)* Optimierer erweitert RMSProp damit, dass die Momente zweiter Ordnung der Gradienten mit in die Berechnung einfließen [A2]. Empirisch wurde gezeigt, dass der Adam Optimierer den anderen gängigen Methoden überlegen ist [10].

### Finale Aktivierungsfunktion

Für den Ausgang des Netzes werden weitere Aktivierungsfunktionen, je nach Art des Problems, herangezogen. Ein Klassifizierungsproblem kennzeichnet sich dadurch, dass am Ausgang unterschieden wird, ob die Eingänge des Systems einer bestimmten Klasse zugeordnet werden können. So kann beispielsweise erkannt werden, ob sich ein bestimmter Gegenstand in einem Bild befindet oder nicht (binäres Klassifizierungsproblem). Auch kann unterschieden werden, um welche Art von Gegenstand es sich handelt (Multiklassen-Klassifizierungsproblem). Weiterhin kann unterschieden werden, ob, wie im oben genannten Beispiel, ein oder mehrere Klassenlabel für das Bild zutreffen können. (Es kann nur ein bestimmter Gegenstand im Bild vorkommen oder es können mehrere verschiedene Gegenstände im Bild vorhanden sein.) Hierbei spiegeln die Ausgangswerte die Wahrscheinlichkeiten für das Vorkommen eines Labels wieder. Tabelle 2.3 gibt eine Übersicht darüber, welche finalen Aktivierungsfunktionen und Fehlerfunktionen für ein jeweiliges Problem genutzt werden können.

Problem	Ausgangstyp	Aktivierungsfunktion	Fehlerfunktion
Regression	numerischer Wert	Linear	Mean Squared Error
Klassifizierung	binärer Wert	Sigmoid	Binary Cross Entropy
Klassifizierung	ein Label, Multiklassen	Softmax	Cross Entropy
Klassifizierung	Multilabel, Multiklassen	Sigmoid	Binary Cross Entropy

Tabelle 2.3: Übersicht Aktivierungs und Fehlerfunktion

Die Entstörung von Sprache kann sowohl als Regressions-, als auch als Klassifizierungsproblem beschrieben werden und wird in Kapitel 2.3 erläutert.

### 2.2.3 Netze für die Zeitreihenanalyse

Für den Anwendungsfall können die Eingangsdaten der Netze als Zeitreihe dargestellt werden. Für die Analyse von Zeitreihen werden daher verschiedene Techniken vorgestellt.

#### Recurrent Neural Network (RNN)

Im Gegensatz zum mehrlagigen Perzeptron, welches in Kapitel 2.2 vorgestellt wurde, nehmen rekurrente neuronale Netze Bezug auf Daten, die zu vorherigen Zeitpunkten gesehen wurden. Außerdem besitzen RNNs gemeinsame geteilte Gewichte. Diese

ermöglichen eine Generalisierung für ungesehene Sequenzen variabler Länge und beschleunigen das Lernen bei ähnlichen Eingangsdaten.

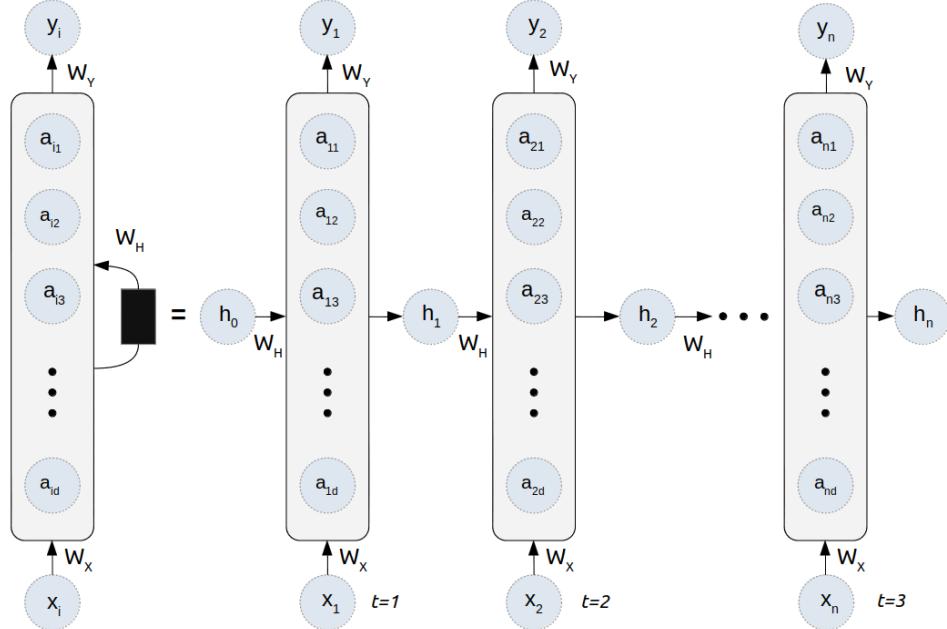


Abbildung 2.5: Rekurrentes Netz

In Abbildung 2.5 ist die Funktionsweise eines rekurrenten Netzes dargestellt. Hierbei stellt  $h$  den Ausgang eines *hidden state*, der hier als Block dargestellt ist, dar. Die Anzahl der Knoten in einem Block ist ein einstellbarer Hyperparameter. Die Matrizen  $W_x, W_y, W_z$  sind die geteilten Gewichte.  $x_i$  ist der Eingangsvektor für jeden Zeitschlitz  $i = 1, 2, \dots, n$  [A3].

Der Nachteil der RNNs liegt im Problem der verschwindenden Gradienten (engl. *vanning gradients*) bzw. der explodierenden Gradienten (engl. *exploding gradients*), da diese die Fehlerrückführung über eine lange Sequenz durchführen müssen. Dies führt zum Absinken bzw. enormer Erhöhung des Gradienten mit steigender Anzahl an Schichten und dazu, dass die weiter entfernten hidden states keine Rolle mehr bei der Berechnung des aktuellen Schrittes spielen bzw. es wie im Fall der exploding gradients, zu instabilen Netzen führt, die keine Abbildungen mehr lernen können [11].

## Long-Short-Term-Memory (LSTM)

Die in Sektion 2.2.3 beschriebenen Unzulänglichkeiten können durch den Einsatz von Long-Short-Term-Memory (**LSTM**) Zellen vermieden werden. Diese wurden 1997 von Hochreiter und Schmidhuber vorgestellt und vermeiden explizit das Problem der langzeitigen Abhängigkeiten [12]. Ähnlich wie die Kettenstruktur der in Abbildung 2.5 gezeigten Zellen, besitzen die LSTMs eine wiederkehrende Aneinanderreichung von Zellen. Eine Zelle besteht dabei aus vier interagierenden Schichten. Die obe-

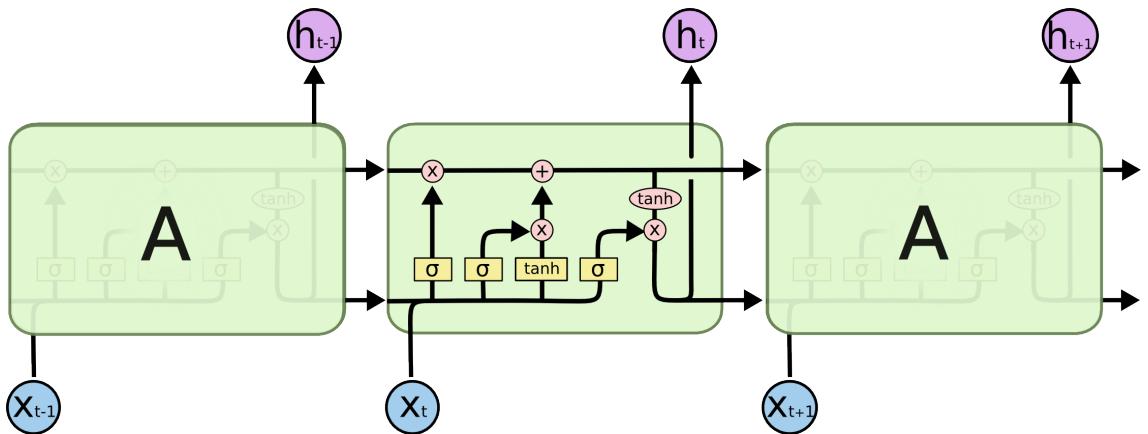


Abbildung 2.6: Aufbau einer LSTM Zelle [A4]

re horizontale Linie in Abbildung 2.6 beschreibt den Zustand der Zelle. Hierüber können Informationen zur nächsten Zelle weitergegeben werden. Das LSTM kann Informationen hinzufügen oder entfernen. Der Informationsfluss innerhalb der Zelle wird von den *Gates* gesteuert. Diese bestehen aus einer Sigmoid-Aktivierungsschicht und einer Punktweisen Multiplikation. Die Sigmoid Funktion gibt Werte zwischen 0 und 1 aus. Somit wird quasi ein Maß an Information bestimmt, die durch die Gates fließen [A4].

Das Gate links wird als *forget Gate* bezeichnet und entscheidet wie viel an Information aus dem Zustand der Zelle entfernt werden soll. Das nächste Gate besteht aus zwei Teilen und entscheidet, wieviel neue Information an den Zustand der Zelle übertragen werden soll. Abschließend wird der Ausgangsvektor  $h_t$  durch Multiplikation des Zellzustands mit den Eingangsdaten und der Tangenshyperbolikus Aktivierungsfunktion gebildet.

Es existieren verschiedene Abstraktionen der LSTM Zellen, wie z.B. die *Gated Recurrent Units*, die eine Alternative mit weniger Parametern darstellen, oder die *Grid LSTMs*, bei der die Zellen nicht nur zwischen den Schichten des Netzes, sondern auch über die räumlich-zeitlichen Dimensionen der Eingangsdaten verbunden sind [13]. In den letzten Jahren ist zudem der Mechanismus der *Attention* als Erweiterung von RNNs und LSTMs in den Fokus gerückt. Hierbei wird ein besondere Fokus auf

bestimmte Eingangsdaten gelegt, welche für die Schätzung der Ausgangssequenz relevant sind. Im Bereich der Sprachverbesserung werden häufig bidirektionale LSTMs eingesetzt. Hierbei werden im Unterschied zu unidirektionalen LSTMs, welche nur Information aus den letzten Eingängen in hidden states speichert, die Eingangsdaten zusätzlich rückwärts durchlaufen um vergangene und zukünftige Information zu konservieren.

### **Convolutional Neural Network (CNN)**

Besonders in der Verarbeitung von Bilddaten haben die Convolutional Neural Networks in den letzten Jahren Einzug gehalten. So wurde die kleinste Fehlerquote bei der Bildklassifizierung mit der MNIST Datenbank mit Hilfe von CNNs erreicht [A5]. Weitere Einsatzgebiete finden sich aber auch in der Audioverarbeitung, wie z.B. maschineller Übersetzung oder Satzmodellierung und -klassifizierung. Der wesentliche Vorteil von CNNs besteht darin, relevante *Features* aus den Eingangsdaten selbstständig erkennen zu können. Features sind charakteristische Merkmale, die beispielsweise einen Hund von einer Katze unterscheiden [A6]. Ein weiterer Vorteil ergibt sich aus der besonderen Effizienz bei der Berechnung des Netzes, da die wesentlichen Operationen aus Faltungen und *Pooling* bestehen.

The diagram shows a 5x5 input matrix labeled "Image" and a 3x3 filter matrix labeled "Filter / Kernel". The result is a 3x3 feature map labeled "Feature".

2	4	9	1	4
2	1	4	4	6
1	1	2	9	2
7	3	5	1	3
2	3	4	8	5

X

1	2	3
-4	7	4
2	-5	1

=

51		

Feature

Image

Filter / Kernel

Abbildung 2.7: Faltung [A6]

Bei der Faltung wird, wie in Abbildung 2.7 zu sehen, ein Filter über die Eingangsdaten geschoben und eine punktweise Multiplikation der überlappendenden Felder durchgeführt. Typischerweise wird der Filter von links nach rechts und anschließend eine Reihe nach unten geschoben. Die Schrittweite hierbei wird auch als *Stride* bezeichnet. Wie auch die Strides bildet die Anzahl der Filter einen festzulegenden Hyperparameter. Mit jedem Filter wird eine neue *Feature Map* generiert, welche zusammen konkateniert werden.

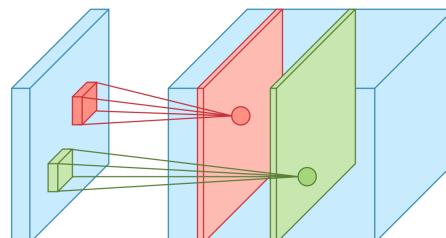


Abbildung 2.8: Konkatenierung der Feature Maps [A6]

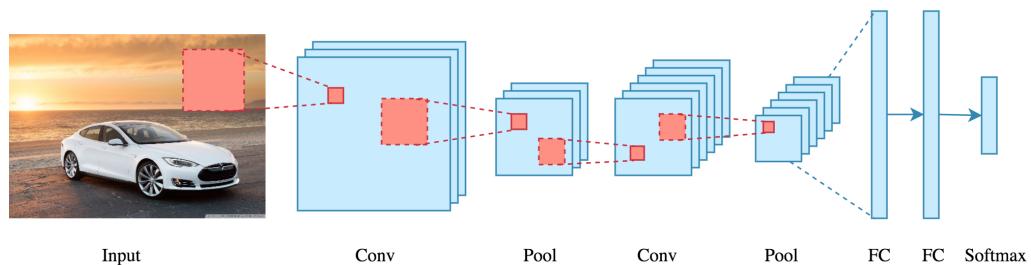


Abbildung 2.9: Aufbau eines CNN [A6]

Meist folgt auf einen Convolutional Layer ein Pooling Layer. Hier werden Informationen quasi komprimiert, indem Durchschnitts oder Maximalwerte gebildet werden.

Hierfür wird ein Fenster variabler Größe definiert, welches die Eingänge für eine Pooling Operation bildet. Die Schrittweite der Verschiebung des Pooling-Fensters wird als (engl. *stride*) bezeichnet. Die Pooling Operation reduziert somit standardmäßig die Dimensionen einer Feature Map. Ein Auffüllen mit Nullen, auch bekannt als Zero-Padding, kann dies jedoch verhindern. Abbildung 2.9 zeigt einen beispielhaften Aufbau eines Netzes, welches CNNs integriert.

## 1D Convolution

Speziell für Zeitreihen können eindimensionale Faltungen vorgenommen werden, die Features aus dem Verlauf erkennen können. Abbildung 2.10 zeigt schematisch die

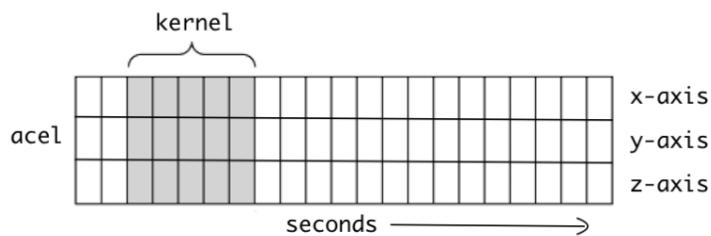


Abbildung 2.10: 1D-Convolution für Beschleunigungssensordaten [A7]

Funktionsweise von 1D-CNNs bei Daten aus einem Beschleunigungssensor. Der Kernel wird in der Zeitachse verschoben und kann so das Auftreten von, wie im gegebenen Beispiel, speziellen Bewegungen, wie dem Laufen, Springen usw. erkennen [A7]. Im weiteren Verlauf dieser Arbeit wird nur noch der Begriff der Convolution verwendet, um eine klare Abgrenzung zur mathematischen Faltungsoperation zu finden.

## ResNet

Das *Residual Neural Network (ResNet)*, vorgestellt von Kaiming He et al. [14], nutzt sogenannte *Skip-Connections*. Wie in Abbildung 2.11 zu sehen, wird eine Iden-

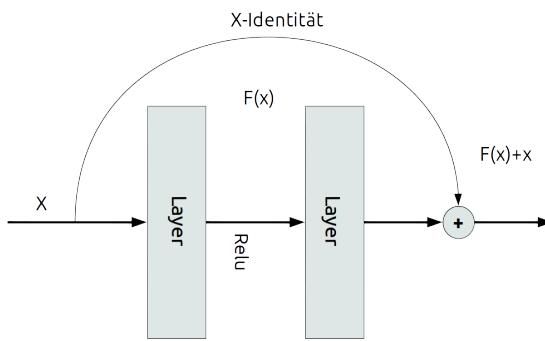


Abbildung 2.11: Skip Connections

tät des Eingangs auf das Ergebnis von meist zwei Netzsichten addiert. Diese Architektur vermindert das Problem der Vanishing Gradients und ermöglicht somit die Konstruktion besonders tiefer Netze, die den flachen Netzen (engl. *shallow networks*) meist überlegen sind.

## Dilated Convolutions

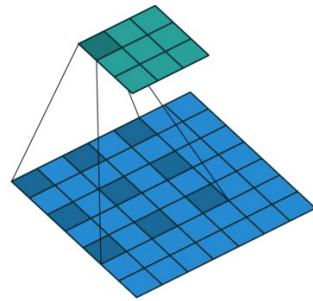


Abbildung 2.12

Dilated Convolutions expandieren das rezeptive Feld um längerfristige Abhängigkeiten abilden zu können, siehe Abbildung 2.12. Hierbei expandiert das rezeptive Feld mit der Anzahl  $n - 1$  Pixel für eine Dilatation von  $n$ . Die Anzahl der Elemente des Kernels bleiben dabei bestehen.

## 2.3 Rauschunterdrückung mit künstlichen neuronalen Netzen

Im Folgenden werden die Konzepte für eine Rauschunterdrückung in Sprachsignalen mit neuronalen Netzen beschrieben.

### 2.3.1 Methoden

Verschiedene Methoden werden verwendet, um eine Entstörung von Sprache mit Hilfe von neuronalen Netzen zu erzielen. Wie bereits angesprochen, können die Trainingsziele als Regressions- oder Klassifikationsproblem beschrieben werden. Für die Forschung bedeutend war in den letzten Jahren vor allem das Erzeugen von Masken, die im Grunde einen Supression-Gain, wie in Kapitel 2.1 beschrieben, berechnen. Dieser wird dabei auf jeden Zeit-Frequenzslot angewendet. Zudem sind generative Methoden nutzbar, die wie bei WaveNet, auch im Zeitbereich angewendet werden.

#### Ideal Binary Mask

Die Ideal Binary Mask erzeugt eine binäre Gain Funktion, welche anhand eines Schwellwertes TF-Slots kategorisiert. Hierfür wird der Signal-Rauschabstand in [15] mit folgender Formel berechnet:

$$SNR_{db}(t, f) = 20 \cdot \log_{10} \left( \frac{S(t, f)}{X(t, f) - S(t, f)} \right) \quad (2.9)$$

Hierbei setzt sich das gestörte Signal  $X(t, f)$  zusammen aus dem korrespondierenden ungestörten Signal  $S(t, f)$  und der Störung  $D(t, f)$ .

$$X(t, f) = S(t, f) + D(t, f) \quad (2.10)$$

Im Folgenden wird die binäre Maske erzeugt durch:

$$IBM(t, f) = \begin{cases} 1 & \text{wenn } SNR_{db}(t, f) > \theta \\ 0 & \text{sonst} \end{cases} \quad (2.11)$$

Der Parameter  $\theta$  wird in [15] mit 0 gewählt. Allgemein können verschiedene Methoden zur binären Maskierung verwendet werden.

Abbildung 2.13 zeigt die Spektrogramme eines Sprachsignals. Im obersten Plot wurde die ideale Binärmaske mit Formel 2.11 für den Signal-Rauschabstand berechnet. Der mittlere Plot zeigt das Sprachsignal, welches mit einer Störung überlagert ist.

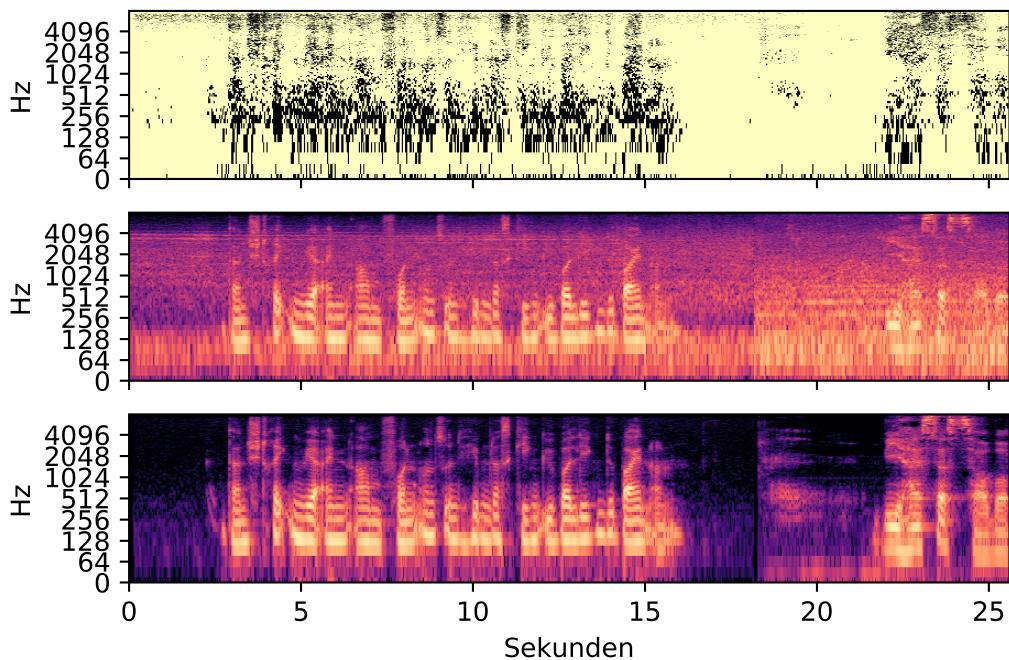


Abbildung 2.13: Spektrogramm für die IBM und Referenz mit und ohne Störung

Der untere Plot zeigt das Spektrogramm der ungestörten Sprache.

Zu den Anwendungsfeldern der IBMs zählen u.A. die Signalabtrennung von einem bestimmten Sprecher, welches sich als Aufgabe im *Cocktail-Party* Phänomen zeigt, sowie dem Entstören von Sprache allgemein. Außerdem können binäre Masken als Front-End für automatische Spracherkennungssysteme verwendet werden. IBMs werden auch in der binauralen Sprachverbesserung eingesetzt. In den letzten Jahren wurden verschiedene Methoden der *Computational Auditory Scene Analysis* (CASA) vorgestellt, die die Performanz von IBM System zu verbessern versuchen. Diese nutzen oftmals Methoden, um gewisse Segmente zu gruppieren, wie eine Analyse der Kontinuität von Tonhöhen (engl. *pitch continuity*) oder einer Silbenansatzdetektion (engl. *onset detection*) [16]. Im allgemeinen zeigt sich die Verwendung der Ideal Ratio Mask der Ideal Binary Mask als Trainingsziel überlegen, da diese die gleichen wichtigen Eigenschaften zum Lösen der angesprochenen Aufgaben erfüllt, aber mit den psychophysikalischen Eigenschaften der menschlichen Wahrnehmung eher korrespondiert [17].

### Ideal Ratio Mask

Für die Ideal Ratio Mask als Trainingsziel wird in [18] folgende Formel angegeben:

$$IRM(t, f) = \left( \frac{|S(t, f)|^2}{|S(t, f)|^2 + |N(t, f)|^2} \right)^\beta \quad (2.12)$$

Hier wird der einstellbare Parameter  $\beta$  zu 0.5 gewählt. Es wird also ein Verhältnis zwischen dem ungestörten Signal zu dem Signal mit aufaddierter Störung gebildet, welches immer im Intervall [0;1] liegt.  $\beta > 1$  sorgt dementsprechend für eine größere Dämpfung im Trainingsziel und vice versa.

Die Vorteile der IRM liegen in der deutlich verringerten Artefaktbildung im Vergleich zur IBM. Allgemein erzielt die IRM deutlich bessere Werte bei der Evaluierung mit dem PESQ (*Perceptual Evaluation of Speech Quality*) Standard, als binäre Maskierungsmethoden [17]. Für die IRM wurden in den letzten Jahren verschiedene Abstraktionen vorgestellt. So wird in [19] die IRM auf eine Sigmoid-Funktion abgebildet, um das aktuelle Signal-Rauschverhältnis zu schätzen. In [18] wird die *Optimal Ratio Mask* als Trainingsziel unter Einbezug von Phaseninformationen hergeleitet.

### Generative Adversarial Network (GAN)

Generative Adversarial Networks wurden 2014 von Goodfellow et al. vorgestellt [20]. Sie eröffneten die Möglichkeit der Generierung von Daten, die anhand ihrer statistischen Eigenschaften denen aus einem Trainingssatz ähneln. Grundlage hierfür bilden zwei neuronale Netze, die ein Nullsummenspiel miteinander spielen. Ein *generatives* Netz erzeugt Daten, welche vom *diskriminativen* Netz auf ihre Echtheit geprüft werden, also ob es sich um erzeugte Daten oder um Daten aus dem Trainingssatz handelt. Das Trainingsziel des generativen Netzwerks ist hierbei die Erhöhung der Fehlerrate des diskriminativen Netzes [A8].

In der Sprachverbesserung werden GANs eingesetzt, um eine direkte Abbildung der Wellenform im Zeitbereich vom gestörten zum ungestörten Signal zu finden. Hierbei wird der Nachteil der Methoden, die auf dem Leistungsspektrum basieren und damit keine Phaseninformationen beinhalten, umgangen. Der Trainingsprozess für GANs gestaltet sich hierbei jedoch oft als schwierig [21].

### 2.3.2 Evaluierungsmetriken

Um die in dieser Arbeit untersuchten Netze zu evaluieren und mit anderen Forschungsergebnissen vergleichen zu können, wurden Metriken basierend auf objektiven Standards herangezogen. Diese stehen im Kontrast zu subjektiven Werten, wie dem MOS, wenn dieser mit Probanden durchgeführt wurde. Nachfolgend sind die Metriken erläutert.

#### PESQ

*Perceptual Evaluation of Speech Quality* ist ein weltweit eingesetzter Standard zur Bewertung der Qualität von Sprachübertragungen. Häufig wird dieser von Netzbetreibern und Telefonherstellern zur Überprüfung von Übertragungsstrecken o.Ä. eingesetzt. Hierbei wird ein aufgenommenes Signal mit seiner Referenz vor dem Durchlaufen eines Systems verglichen. Die Bewertung erfolgt hierbei auf der MOS Skala, welche bereits in Sektion 2.3.3 erläutert wurde [34]. Abbildung 2.14 zeigt Komponenten und Aufbau des Evaluierungsmodells.

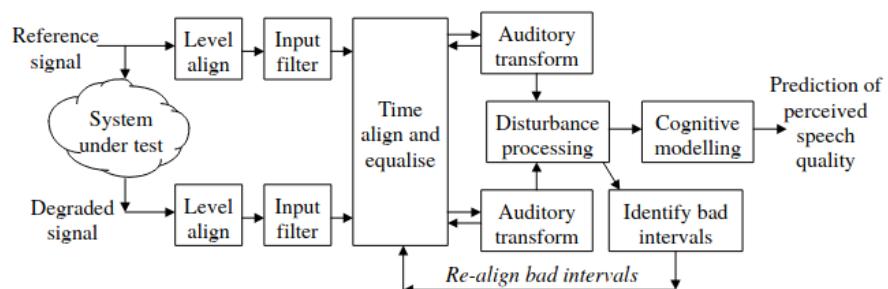


Abbildung 2.14: Struktur des PESQ Modell [34]

#### STOI

Eine weitere in der Literatur häufig verwendete Metrik ist die *short-time objective intelligibility*, welche eine objektive Bewertung für die Verständlichkeit der untersuchten Sprache liefern soll. Hierfür werden Gebiete einer ungefähren Länge von 400ms aus TF-Slots betrachtet. Diese werden in 1/3 Oktavenabstände unterteilt um Korrelationskoeffizienten zu berechnen [35].

## SDR

Die *Signal to Distortion Ratio* bzw. *Source to Distortion Ratio* bietet eine weitere Metrik, die in der Literatur häufig eine Anwendung bei Separierungsaufgaben findet. In [39] wird die SDR angegeben als:

$$SDR = 10\log_{10} \frac{|Y|^2}{|\hat{Y} - Y|^2} \quad (2.13)$$

Verglichen wird hier die Energie des Nutzsignals ohne Störung  $Y$ , zur übrig gebliebenen Störung, welche sich nach Subtraktion von  $Y$  vom geschätzten entstörten Signal  $\hat{Y}$  ergibt. Mit dieser Metrik kann abgeschätzt werden, inwieweit die additive Störung des Signals durch das System entfernt werden konnte. Beachtet werden muss, dass eine Degradation des Sprachsignals ebenfalls zu einer Erhöhung des SDR führt.

### 2.3.3 Stand der Technik

In dieser Sektion werden aktuell wichtige Architekturen und Frameworks aus der Forschung dargestellt.

#### EHNet

Tashev et al. stellten 2018 das EHNet vor. Ein datengetriebenes Ende-zu-Ende Modell, welches aus einer Kombination aus Convolutional und Rekurrenten Netzen besteht. Es kommt ohne Annahmen über die Stationarität oder die Arten von Störungen aus. Mit Hilfe der Convolution Filter werden zunächst Strukturen aus dem Zeitbereich und Frequenzbereich extrahiert, welche im Anschluss von einem Bidirektionalen LSTM verarbeitet werden. Berichtet wird hier von einer PESQ Verbesserung von 0.64 auf vom Netz ungesehenen Rauschtypen [22].

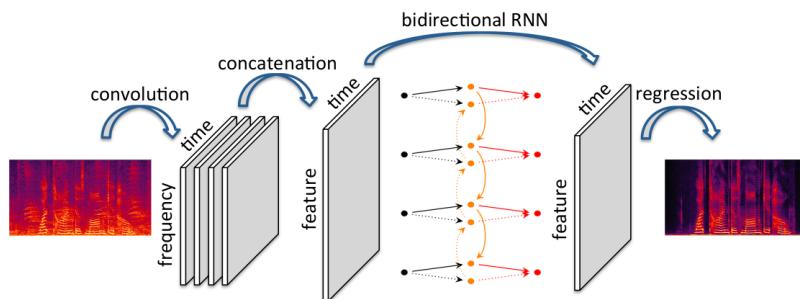


Abbildung 2.15: EHNet Model Architektur [22]

Als Trainingsziel wird die Minimierung des quadratischen Fehlers zwischen dem wahren Spektrum und dem geschätzten Spektrum angesetzt, welches aus einem multiplikativen Faktor aus dem Eingangssignal geschätzt wird. Dies entspricht einer Ratio Mask, welche in Sektion 2.3.1 beschrieben ist. Abbildung 2.15 verdeutlicht hierbei das Modell. Ein 500 Frame langes Fenster mit 256 Frequenz Bins wird hier mit 256 Kerneln der Größe 32x11 und einer Schrittweite von 16x1 in der Frequenz und Zeitachse gefaltet. Diese werden anschließend vertikal entlang der Feature Dimension konkateniert um eine 2D Feature Map als Input für ein bidirektionales Netz zu erzeugen. Das bidirektionale Netz kann hierfür zeitliche Abhängigkeiten in beiden Richtungen erfassen. Die Dimension der Feature Map wird zudem bei den Faltungen mittels Padding gleich belassen. Als Aktivierungsfunktion zwischen den Schichten wird ReLu genutzt. Abschließend wird ein Fully-Connected Layer eingesetzt, der eine lineare Regression zu den entsprechenden Gain Parametern durchführt.

## DeepXi

DeepXi wurde 2019 von Nicolson und Paliwal vorgestellt [7]. Es handelt sich hierbei um ein Framework mit MMSE-Schätzer Ansatz, das die Brücke zu den modernen Techniken der Sprachverbesserung mit Hilfe von Deep Learning Methoden schlagen soll. Die Performanz eines MMSE Schätzers hängt von der Genauigkeit der *a priori* und *a posteriori* SNR Schätzungen ab (siehe Sektion 2.1.2). Hierfür wurden die SNR Schätzungen zunächst von einem neuronalen Netz, welches aus Residual Long-Short-Term-Memorys (**ResLSTM**) besteht, berechnet. Eine Beschreibung hierfür findet sich in [7]. Aktuell wurden die LSTMs allerdings durch Residual Bottleneck Blöcke mit 1D Convolution und Dilatationen ersetzt [A9].

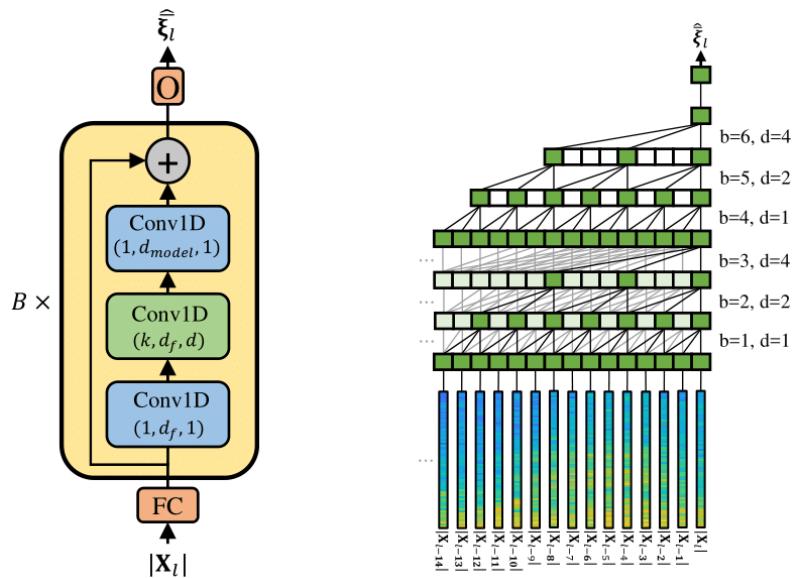


Abbildung 2.16: Temporal Convolution Network [A9]

Die vorgestellte Architektur ist in Abbildung 2.16 (Kernelgröße  $k$ , Dilatation  $d$ , Bottleneck Größe  $d_f$ , output Größe  $d_{model}$ ) zu sehen und beschreibt ein Temporal Convolution Network (**TCN**) [23]. Den Ausgang des Netzes bildet der Parameter  $\xi$ , der für das *a priori* SNR steht. Es werden 40 Bottleneck Blöcke, 2 Millionen Parameter und eine maximale Dilatation von 16 genutzt. DeepXi kann zur Sprachverbesserung, Rauschschätzung und als Front-End für automatische Spracherkennungssysteme eingesetzt werden. Als mögliche Methoden stehen MMSE-STSA (*short-time spectral amplitude*) oder MMSE-LSA (*log-spectral amplitude*) Schätzer, sowie Wiener Filter und IRM und IBM zur Verfügung. Berichtet wird eine durchschnittliche Verbesserung des PESQ Wertes um 0.85 bei Nutzung des LSA Schätzers.

# WaveNet for Speech Enhancement

Anders als die bisher vorgestellten Methoden arbeitet das 2017 vorgestellte *Wavenet for Speech Enhancement* im Zeitbereich des Signals. Hiermit wird, wie bereits ange- sprochen, im Gegensatz zu den meisten Methoden, die auf der spektralen Amplitude basieren, die Phaseninformation mit einbezogen. Hierfür nutzt das Netz nichtkau- sale Convolutions mit Dilatation, welche in Abbildung 2.17 dargestellt sind. In [24] beschreiben die Autoren die Vorteile der Verarbeitung im Zeitbereich damit, dass es Zugriff auf Strukturen gibt, die bei anderen Methoden nicht in gleichem Maße zur Verfügung stehen. Hierzu zählen die Klangfarbe und Phoneme. Wavenet adaptiert hierbei das PixelCNN, ein generatives Modell für Bilder [25]. Das *Wavenet for Speech Enhancement* ist quasi eine Erweiterung des Wavenets, welches zur Synthesierung von Sprache eingesetzt wird. Im ursprünglichen Wavenet wird eine Wahrscheinlich- keitsdichtefunktion über einen autoregressiven Prozess geschätzt. In der Abstraktion zur Sprachverbesserung wird eine Regression als Trainingsziel gesetzt, die direkt die Amplitude im Zeitbereich schätzt. Die nichtkausalen Convolutions werden in ei-

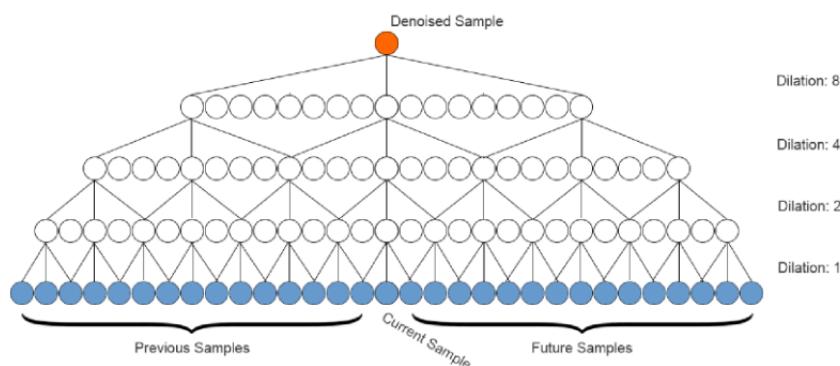


Abbildung 2.17: Nichtkausale Convolution mit Dilatation [24]

nem Residual Layer durchgeführt. Das Modell berechnet, anders als in Abbildung 2.17, nicht nur ein Sample am Output, sondern eine Reihe an Samples. Da sich in dieser Architektur viele Berechnungen überlappen, kann mit dieser Methode Redundanz vermieden werden. Auch bei einer Echtzeitübertragung kann eine kleine Latenz in Kauf genommen werden, was das symmetrische Kontextfeld, wie in Abbildung 2.17, ermöglicht. Weiter wird eine Funktion namens *Conditioning* eingesetzt, die für den dort gewählten Datensatz einen binär codierten Skalar als Bias Wert für unterschiedliche Sprecher in jede Convolution Operation hinzufügt. Abbildung 2.18 verdeutlicht die Funktionsweise der Architektur. Jeder Layer verdoppelt die Dilatation bis zu einem Wert von 512. Das gestörte Sprachfragment wird, ausgehend von

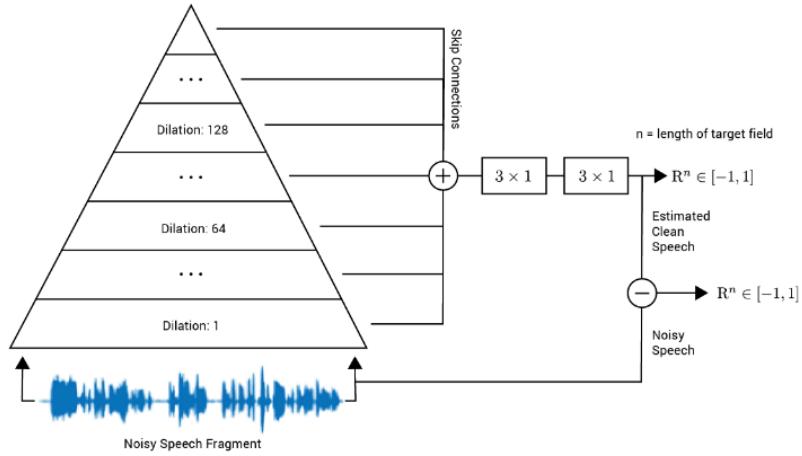


Abbildung 2.18: Wavenet zur Sprachentstörung [24]

einem Kanal auf 128 Kanäle (Feature Maps) mit einem 3x1 Kernel ausgedehnt. Die Skip-Connections sind damit 128 1x1 Kernel, die aufaddiert werden um den Output durch eine ReLu Funktion zu bilden. Anschließend werden zwei weitere 3x1 Kernels genutzt, um den Output für das Zielfeld der Größe  $n$  zu bilden. Das Rauschsignal kann durch die Subtraktion von gestörtem und geschätztem entstörten Signal gebildet werden.

Zur Evaluierung wurden **SIG**, **BAK**, **OVL** [26] Metriken, sowie eine subjektive Bewertung durch Probanden genutzt. Diese wurde ebenfalls auf einer Skala von 1-5 durchgeführt, wobei 1 für eine sehr degradierte Sprache mit aufdringlichem Störgeräusch steht und 5 für eine nicht degradierte Sprache ohne Störgeräusche. Verglichen wurde hier mit einer Wiener Filter Methode, welche einen MOS(*Mean Opinion Score*) Wert von 2.92 erreicht. Im Vergleich erreicht Wavenet einen Wert von 3.60 .

## SEGAN

SEGAN ist ein generatives Netz, welches 2017 vorgestellt wurde [27]. Als Trainingsziel wird hier das Mapping von einer Verteilung von Amplitudenwerten der ursprünglichen Samples hin zur neuen (entstörten) Verteilung gesetzt. Das Mapping wird vom Generator durchgeführt, welcher in Abbildung 2.19 dargestellt ist. Der Generator besteht aus einer Encoder-Decoder Architektur und ist mit Convolutional Layern besetzt. Dies befähigt den Generator zum Erkennen von zeitlich nah beieinander liegenden Korrelationen und reduziert den Rechenaufwand und die benötigten Parameter. Die Skip-Connections vom Encoder zum Decoder verhindern das Einbüßen von Informationen durch den Bottleneck. Diese verbinden jeden Encod-

der Layer mit seinem korrespondierenden Decoder Layer. Hier wird das Eingangssignal über Convolutional Layer hin zu einer komprimierten Darstellung dezimiert, die als *thought vector*  $\mathbf{c}$  bezeichnet wird. Dieser wird mit dem *latent vector*  $\mathbf{z}$  konateniert.

Pascual et al. beschrieben in, dass für Fehlerfunktionen, wie den mittleren quadratischen oder den absoluten Fehler, inherente Annahmen über die Verteilung der Ausgangswerte angenommen werden, die z.B. die Verteilung hin zum einem Mittelwert treiben. Dies wird damit umgangen, indem das Trainingsziel des Diskriminators darin besteht, den Generator zu real aussehenden Schätzungen zu treiben. Zusätzlich wird eine  $L_1$  Regularisierung verwendet.

In der Evaluierung wird eine durchschnittliche Verbesserung des PESQ Wertes von 0.19 aufgezeigt. In diesem Fall erreicht der Wiener Filter mit einer Verbesserung von 0.25 einen besseren Wert. Jedoch zeigt sich das SEGAN Netz bei anderen Performanzindikatoren wie dem CSIG oder CBAK überlegen. Eine subjektive Evaluierung zeigt einen MOS (siehe Sektion 2.3.3) von 3.18. Im Vergleich dazu wird ein Wiener Filter mit 2.70 und das gestörte Signal mit 2.09 angegeben.

SEGAN arbeitet, wie auch Wavenet, im Zeitbereich des Signals. Auch hier wird ein Ende-zu-Ende Modell mit 28 verschiedenen Sprechern und 40 Rauschtypen trainiert. In [27] werden als Hauptvorteile der schnelle Berechnungsprozess aufgrund fehlender rekurrenter Operationen, die Nichtkausalität, das Ende-zu-Ende Training und die geteilte Parametrisierung für verschiedene Sprecher und Rauschtypen, die dem System zur Generalisierung verhelfen, genannt.

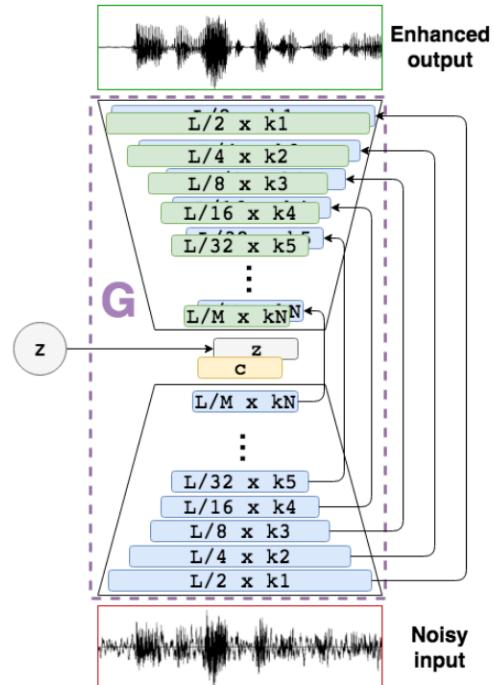


Abbildung 2.19: SEGAN Generator [27]

## Industrie

Während der Erstellungszeit dieser Arbeit wurden Beta Versionen der Sprachverbesserungssysteme von NVIDIA (RTX Voice) und Krisp veröffentlicht. Für die Berechnung in Echtzeit nutzt NVIDIA die CUDA Kerne moderner Grafikkarten. Hierbei ist neben der Tatsache, dass neuronale Netze verwendet werden, wenig über die Funktionsweise dieser Systeme bekannt. Babblelabs bietet ebenfalls Sprachverbesserungssysteme an. Bekannt ist hier, dass das Trainieren des Netzes einen Umfang von 6-8 Wochen auf 8 NVIDIA V100 Karten beansprucht. Hierbei werden die Trainingsziele während des Prozesses mehrmals angepasst. Der Trainingsdatensatz besteht dabei aus 40000 Stunden Sprache, sowie jeweils 15000 Stunden an Störgeräuschen und Musik. Außerdem werden 100000 raumakustische Modelle einbezogen um Störungen durch Nachhall usw. zu berücksichtigen [A11].

### 3 Datenvorverarbeitung

Die in dieser Arbeit vorgestellten Netze wurden mit dem Framework Tensorflow (Version 1.14) und der dazugehörigen Keras Bibliothek berechnet. Genutzt wurde hierfür eine NVIDIA GTX 1080 Grafikkarte in einer Konfiguration mit einem Intel i7 4770k bei 16 GB Arbeitsspeicher.

In dieser Arbeit wurden verschiedene Daten und deren Darstellungsformen als Eingänge für die Berechnungen der neuronalen Netze verwendet. Für die spektralen Amplituden der Fourier Transformation als Eingangsdaten wurde die Python Bibliothek Librosa genutzt [R1]. Zur Erstellung des Datensatzes wurden Sprachaufnahmen mit einer Länge von ca. vier bis acht Sekunden mit verschiedenen Störungen überlagert. Zunächst wurden sechs verschiedene Störungen verwendet:

*Martinshorn, Straßenlärm, Fahrgeräusche aus einem Autoinnenraum, Waschmaschine, Flugzeuglandung* und das Durcheinanderreden vieler verschiedener Sprecher bei einer Party, auch bekannt als (engl.) *babble noise*.

Um die Datensätze zu erzeugen, wurden die Sprach- und Störsignale mit Hilfe der schnellen Fouriertransformation (**STFT**) in den Frequenzbereich gewandelt. Hierfür wurde die Hann Fensterfunktion genutzt. Die Fensterlänge wurde zu 512 gewählt, welches entsprechend 257 Frequenzbins des einseitigen FFT Spektrums liefert. Eine Fensterlänge entspricht demnach 31.25 ms bei einer Abtastrate von 16000 Hz. Als Überlappungsfaktor wurde  $\frac{1}{4}$  der Fensterlänge gewählt.

Die Sprachaufnahmen wurden dem deutschen Teil des "Common Voice" Datensatzes von Mozilla entnommen und beinhalten über 8000 Äußerungen [R3]. Diese wurden vom *mp3* Format zum *wav* Format mit einer Bittiefe 16 und PCM Codierung gewandelt.

Um die Überlagerung von Sprache und Störung zu bilden, wurde ein Script, das zur Überlagerung von *wav* Dateien bei einem bestimmten SNR verwendet werden kann, dementsprechend angepasst, um einen Datensatz zu generieren [R2]. Für die Überlagerung wurden SNR Werte von -5 dB bis +20 dB, mit einem Abstand von jeweils 5dB, verwendet. Das Skript zur Erzeugung des Datensatzes wählt hierfür eine zufällige Sprachdatei und überlagert diese mit einer zufällig gewählten Störung bei einem zufälligen SNR Wert. Anschließend erfolgt die Wandlung der Überlagerung, sowie deren dazugehöriger (engl.) *ground truth*, also des Sprachsignals ohne Störung,

in den Frequenzbereich. Das Frequenzspektrum wird im Weiteren logarithmiert und getrennt von den dazugehörigen Phaseninformationen im *pickle* Format gespeichert. Die Verarbeitung von Sprachdatensätzen führt zu einem hohen Speicher- und Rechenaufwand. Dementsprechend wurden mehrere Teildatensätze erzeugt, die aus jeweils 1500 Äußerungen bestehen, um eine schnelle und sichere Verarbeitung zu gewährleisten.

### 3.1 Kontextfenster

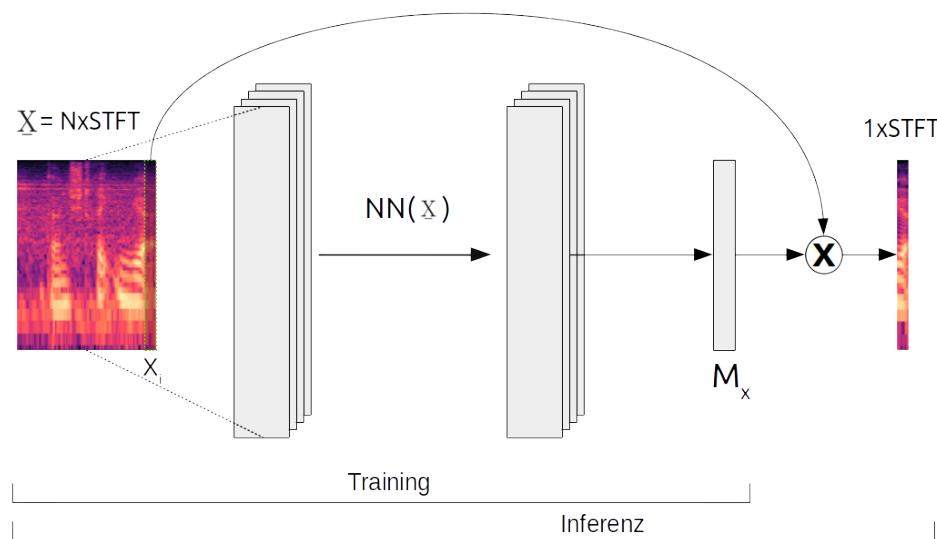


Abbildung 3.1: Neuronales Netz zur Maskierung

In Sektion 2.3.3 wurde bereits die Möglichkeit dargestellt, Berechnungen zur Sprachverbesserung mit Hilfe von *Kontextfenstern* zu realisieren. Abbildung 3.1 zeigt die in dieser Arbeit hauptsächlich gewählte Architektur. Links im Bild zu sehen ist die Eingangsmatrix  $\underline{X}$ , bei der der Vektor  $X_i$  mit  $i \in (1, N)$  für die STFT des aktuellen Zeitpunktes steht. An diesen angestellt sind die Vektoren der letzten  $N$  Zeitpunkte, wodurch die Kontextmatrix gebildet wird. Der Lernprozess besteht also darin, aus dem Kontextfenster eine Maske für entsprechend 257 Frequenz Bins des aktuellen Zeitpunktes zu bilden. Zur Inferenz wird, wie in Abbildung 3.1 zu sehen, der Vektor  $X_i$  mit dem Maskenvektor  $M_{x_i}$  multipliziert. Bei der Inferenz werden die Phaseninformationen des gestörten Signals übernommen.

## 3.2 Skalierung

Um die Performance des Lernvorgangs zu verbessern, wurden die Eingangsdaten skaliert. Allgemein gebräuchlich ist hier die Standardisierung, bei der die Werte zum Mittelwert 0 mit Einheitsvarianz abgebildet werden. Dies hilft dem Optimierer bei der Konvergenz zu den Minima. Formel 3.1 zeigt diese Abbildung.

$$Z = \frac{x - \mu}{\sigma} \quad (3.1)$$

Hierbei steht  $x$  für die Eingangswerte.  $\mu$  und  $\sigma$  bilden den Mittelwert und die Varianz pro jeweiliger Zeile der Eingangsmatrix [R4].

## 3.3 Datensatz

Zusammengefasst bedeutet dies, dass ein Datensatz mit folgenden Eigenschaften erstellt wurde, der als Grundlage für die meisten Versuche diente.

<b>Äußerungen</b>	<b>Störtypen</b>	<b>SNR</b>
1500	6	[-5, 0, 5, 10, 15 , 20]
<b>STFT Dimensionen</b>	<i>Training (80%)</i>	<i>Test (20%)</i>
	[257, 485338]	[257, 157307]
<b>IBM Klassenverteilung</b>	<i>1</i>	<i>0</i>
	ca. 25 %	ca. 75 %

Als Evaluierungsdatensatz wurden 30 Äußerungen für jeweils sechs Störtypen, bei entsprechend jeweils 6 Störabständen, verwendet. Dies bedeutet 36 Datensätze mit jeweils 30 Äußerungen.

## 4 Untersuchung von Ideal Binary Masks mit LSTM Ansatz

Für diese Arbeit wurden eine Reihe verschiedener Netzarchitekturen für den jeweiligen Anwendungsfall experimentell hinsichtlich ihrer Performanz zur Sprachverbesserung untersucht. Folgend werden diese Architekturen vorgestellt, wobei Beobachtungen, Trainingsstrategien und Evaluierungskriterien erläutert werden.

In Sektion 2.3.1 wurde bereits die IBM als Ziel der CASA beschrieben. Im Rahmen dieser Arbeit wurde untersucht, ob sich binäre Masken als zuträglich für die Sprachverbesserung erweisen können. Wie bereits angesprochen, findet sich die Verbreitung von IBM vornehmlich im Bereich von Separierungsproblemen, wie dem Auseinanderhalten verschiedener Sprecher [29], dem Isolieren der Stimme aus Musikstücken [30] und allgemein als Perfomanzindikator in einer Vielzahl von Abhandlungen. Zur Berechnung der Masken werden häufig zusätzliche Features zu den spekralen Amplituden hinzugezogen. Ein Beispiel unter vielen sind die Mel-Frequenz-Cepstrum-Koeffizienten, welche eine Darstellung basierend auf einer diskreten Cosinustransformation auf der Mel Skala darstellen. Diese orientiert sich an der für Menschen wahrgenommenen Tonhöhe [A12].

In [15] wird von einem Ende zu Ende System berichtet, dass in der Lage sein soll, unter Zuhilfenahme von Subbandenergie Komponenten, die Binärmaske zu schätzen.

Abbildung 4.1 zeigt die Netzarchitektur aus [15]. Zu sehen sind drei aneinanderge-reihte Bidirektionale LSTM Schichten mit jeweils 1024 Zellen. Der Ausgangsvektor  $\hat{y}(n)$  wird mit ei-ner Sigmoid Aktivierungsfunktion gebil-det. In [15] wurde ebenfalls eine DFT mit Breite 257 genutzt. Trainiert wur-de das Netz mit Adam als Optimie-rer und *binary crossentropy* (dt. Kreuz-tropie) als Fehlerfunktion. Die be-schriebene Architektur wurde unter Be-trachtung verschiedener Parameter un-tersucht. Zunächst wurde versucht aus-schließlich unter Zuhilfenahme der spek-tralen Amplituden der STFT eine Mas-kenschätzung zu erreichen. Hierfür wur-de der Ausgangsvektor ebenfalls zu ei-ner Länge von 257 gesetzt. Es zeigt sich, dass das Netz nach dem Training nicht in der Lage ist, eine entsprechen-de Abbildung zu finden. Die Ausgangsmatrix der Schätzung bestand hier in allen Ele-menten aus dem selben Wert, der nahe-legt, eine Abbildung des Durchschnitts-werts aller im Training gesehenen Werte zu sein. Weitere Versuche unter Zuhilfenahme von Mel-Komponenten, welche an die STFT Vektoren konkateniert wurden, zeigten ebenfalls keine Verbesserung für diese Konfiguration.

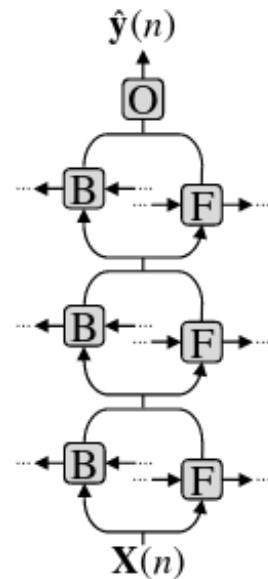


Abbildung 4.1: Bidirektionales LSTM [15]

## 4.1 Gewichtung der Klassen

Allgemein muss bei einem Klassifizierungsproblem die Häufigkeit des Vorkommens der einzelnen Klassen des Datensatzes berücksichtigt werden. Die Zielmaske der IBM zeigte sich entsprechend durch die Erfüllung des Kriteriums aus Formel 2.11 mit einer Häufigkeit von 75 % vorkommenden Nullen, welche Zeit-Frequenzslots (TF-Slots) mit niedrigem SNR darstellen und 25 % vorkommenden Einsen, welche für die verlässlichen TF Slots mit hohem SNR stehen. Der Optimierer sieht eine generelle Schätzung des Nullwertes daher als günstig an, da für diesen so bereits 75 % Genauigkeit erreicht ist.

In diesem Zuge wurde für das weitere Training die *weighted binary crossentropy* Fehlerfunktion eingeführt. Diese gewichtet die Komponenten der einzelnen Klassen

und deren geschätzte Wahrscheinlichkeit mit einem multiplikativen Faktor, um so den Fehlerwert anzupassen. Dies treibt den Optimierer dazu, die Gewichte des Netzes weiter anzupassen, um nicht, wie im oben beschrieben Szenario, aufgrund des ungleichmäßigen Datensatzes in einem Minimum stecken zu bleiben. Damit ergibt sich die neue Fehlerfunktion zu Formel

$$H_p(q) = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(p(y_i)) \cdot \alpha_i + (1 - y_i) \cdot \log(1 - p(y_i)) \quad (4.1)$$

$y$  beschreibt hier die Klasse und  $p(y)$  die vorhergesagte Wahrscheinlichkeit der Zugehörigkeit zu einer Klasse.  $\alpha_i$  ist der klassenabhängige Gewichtungsfaktor. Auch mit angepassten Gewichten kann das gewählte Netz nicht konvergieren. Es zeigt sich eine Mittelwertbildung ähnlich wie im ersten Versuch, diesmal allerdings beschränkt auf einen Frequenzbin. Andere Methoden zum Ausgleich des Datensatzes bestehen darin, die Trainingsdaten neu aufzubauen, indem Teile entfernt oder hinzugefügt werden und somit die Verteilung der Klassen anzupassen. Das gegebene Netz wurde im Anschluss hinsichtlich einer Subband-Klassifikation untersucht.

## 4.2 Subband Klassifikation

In der Literatur wurden eine Vielzahl verschiedener Subband Klassifikationen untersucht. In [31] wird ein 64 Band Gammoton Filter eingesetzt. In [32] werden 26 Subbänder über Mel Filter Komponenten und Methoden wie etwa Subband-Centroide gebildet.

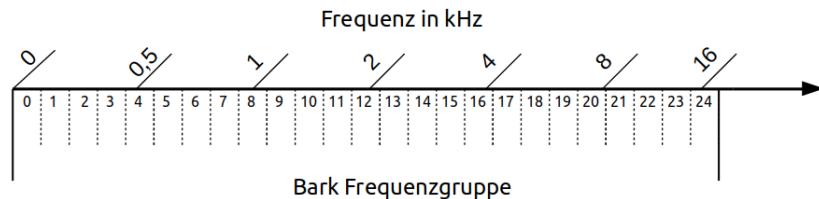


Abbildung 4.2: Bark Skala

Im Rahmen dieser Arbeit wurde daher als Alternative ein Ansatz über die Bark Skala gewählt. Diese ist eine psychoakustische Skala, welche die wahrgenommene Tonhöhe wiederspiegelt. Entsprechend zeigt sich eine Verdopplung des Bark-Wertes als wahrgenommene Verdopplung der Tonhöhe [A13]. Abbildung 4.2 verdeutlicht den Zusammenhang zwischen den Bark-Frequenzgruppen und der Frequenz in kHz.

Aufgrund der Verarbeitung von Signalen mit einer Abtastrate von 16 kHz und damit maximal auftretender Frequenz von 8 kHz, wurde eine Darstellung mit 20 Subbändern gewählt. Um die Subbänder zu generieren wurde eine Funktion erstellt, die die Matrix der STFT anhand der Trennfrequenzen der Bark Skala aufteilt. Da das menschliche Ohr Signale in Frequenzgruppen der Bark-Skala verarbeitet, ergeben sich möglicherweise nützliche Effekte für die Sprachverbesserung, wenn eine Maskierung anhand dieser Referenz durchgeführt wird. Um den binären Maskenwert als Trainingsziel zu formulieren, wurden die STFT Amplitudenwerte in das Leistungsspektrum gewandelt und je Subband in Relation zur Bandbreite gesetzt. Entsprechend wurde also die Energiedichte pro Hertz berechnet. Anschließend wurde mit Formel 2.9 die Relation aus Signalenergiedichte zu Rauschenergiedichte gebildet, was zu einer dimensionslosen Darstellung führt, die anhand des Kriteriums aus Formel 2.11 in die binäre Zielmaske gewandelt wurde.

Implementiert wurde zunächst ein Netz aus 20 parallel angeordneten LSTMs mit jeweils 64 Zellen. Jedes LSTM verarbeitet dabei ausschließlich das aus der STFT gewonnene Subband im Bereich der Bark Bänder. Das Ergebnis hieraus zeigte erneut keine ausreichende Konvergenz. Daher wurde im nächsten Versuch die Anzahl der Subbänder auf 60 erhöht. Hierfür wurden die Subbänder weiter in jeweils drei neue Subbänder gleichen Abstands unterteilt. Das Training wurde in 10 Epochen mit jeweils 2000 Mini Batches je Epoche durchgeführt. Die Länge des Kontextfensters betrug 16. Die effektive Trainingszeit betrug dabei ca. 12 Stunden auf der gewählten Konfiguration. Weiterhin wurde die Fehlerfunktion aus Formel 4.1 und Adam als Optimierer verwendet.

Die erzielte Genauigkeit beträgt dabei im Mittel auf dem Testdatensatz ca. 83 %. Jedoch zeigt sich, dass die allgemeine Genauigkeit aufgrund der unterschiedlichen Auftrittswahrscheinlichkeit der beiden Klassen kein ausreichendes Bewertungsmaß darstellt.

Die Schätzung des Netzes gibt die Auftrittswahrscheinlichkeit der beiden Klassen wieder. Standardmäßig werden Werte  $> 0.5$  als Klasse 1 abgebildet. Klasse 1 steht dabei für den Wert 1 als Ergebnis der IBM. In Abbildung 4.3 wurden die erreichten Trefferraten, sowie die falsch positiv erkannten TF Slots des Testdatensatzes in Bezug zum gewählten Schwellwert gesetzt. Es zeigt sich, für den standardmäßigen Schwellwert, eine Trefferrate von 20 % für die Klasse 1. 5 % des Datensatzes wurde dabei fälschlicherweise der Klasse 1 zugeschrieben. Entsprechend der Literaturkonvention werden diese Werte als HIT (Treffer) und FA (*engl. false alarm*) bezeichnet. Für einen Schwellwert von 0,1 können 80 % der Klasse 1 erkannt werden, jedoch werden auch 35 % des Datensatzes falsch zugeordnet.

Abbildung 4.4 vergleicht die geschätzte IBM (c) zur *ground truth* IBM (a) in einem Ausschnitt des Testdatensatzes. In (b) ist die Ausgabe des neuronalen Netzes zu sehen. Durch Anwendung des Schwellwertkriteriums entsteht die IBM in (c). Der erste Teil des Ausschnitts ist dabei mit der Martinshorn Störung überlagert, erkenn-

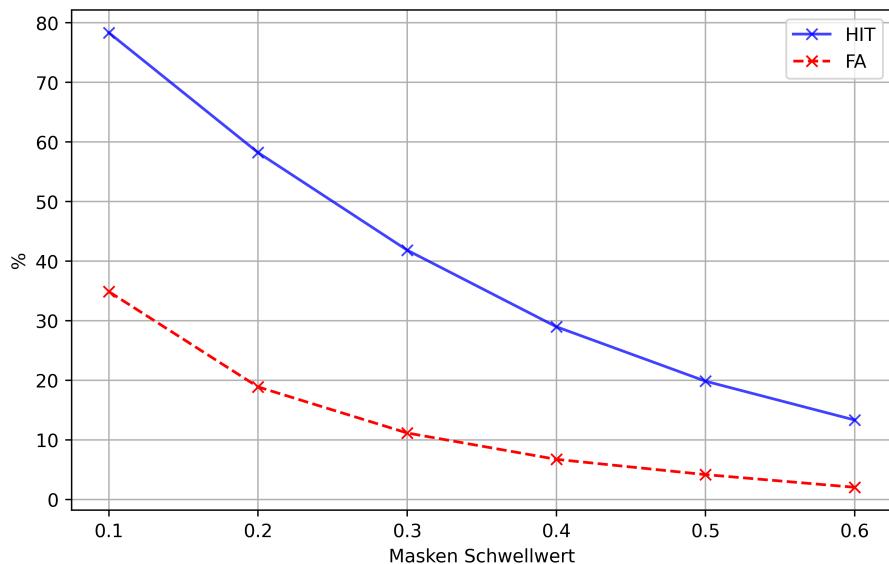


Abbildung 4.3: Trefferrate und falsch positiv klassifizierte TF-Slots

bar an den auffälligen horizontalen Linien, die aus den singulären Frequenzen samt Obertönen des Horns entstehen. An der Schätzung ist zu erkennen, dass das Netz bestimmte Gruppen an TF-Slots zusammengefasst hat und somit die Granularität der Klassifikation wie in der ground truth abgenommen hat. Auch zu erkennen ist, dass größere Bereiche nicht der entsprechenden Klasse zugeordnet werden konnten, wie im hinteren Teil, der durchgehend als Klasse 0 geschätzt wurde.

### 4.3 Diskussion

Wenn die TF-Slots mit hoher Genauigkeit klassifiziert werden können, ist eine IBM in der Lage, additive Störungen deutlich zu mindern und somit die Verständlichkeit der Sprache zu erhöhen. Jedoch leidet die Qualität der Sprache hierbei enorm. In [36] wird daher eine adaptive Maske als Kombination aus IRM und IBM verwendet. Im Rahmen dieser Arbeit wurde gezeigt, dass ein Subband Klassifikator aus parallelen LSTMs in der Lage ist, eine funktionsfähige binäre Maskierung zu erreichen. Jedoch bleibt dieser in der Leistungsfähigkeit noch hinter den in bspw. [37] vorgestellten Methoden zurück, bei denen eine Kombination aus vortrainiertem KNN und Support Vector Machines Subbänder klassifiziert. In [31] wird von Trefferraten um 60 % berichtet, wenn Spektralamplituden als Eingangsfeatures verwendet werden. Aus der Literatur zeigt sich zudem, dass das Einbinden weiterer Features, wie den angesprochenen MFCCs und anderen, besonders zuträglich für eine korrekte Klassifizierung sind.

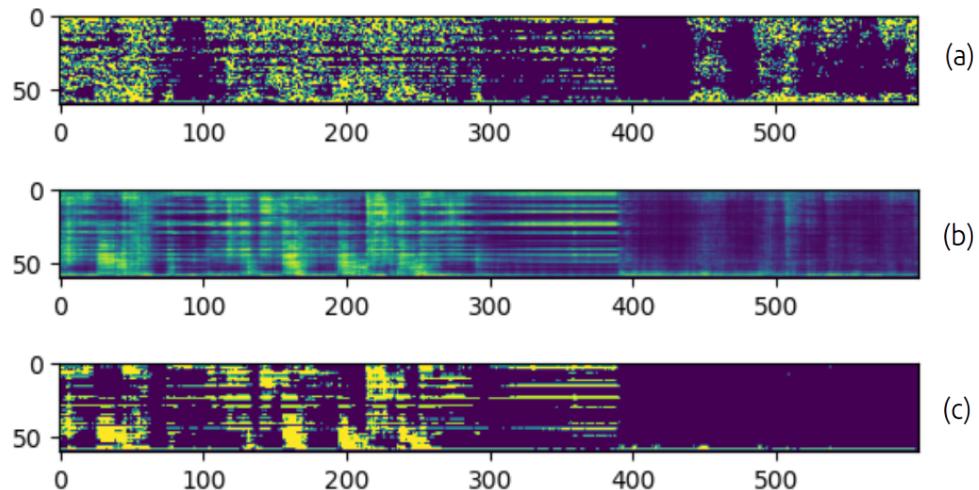


Abbildung 4.4: Vergleich von Schätzung und Ground Truth

Um die vorgestellte Architektur weiter zu verbessern, sollte daher die Zahl der Eingangsfeatures erweitert werden. Zudem empfiehlt es sich, die Dimensionen der LSTM zu erweitern sowie mehrere LSTMs aneinander zu reihen. Die Fehlerfunktion sollte weiterhin mit einem Gewichtungsparameter angepasst sein, um so die Disbalance der Klassen auszugleichen. Möglicherweise bietet auch der sogenannte *Focal Loss* [38] Raum für weitere Untersuchungen, da mit diesem die Aufmerksamkeit auf schwer- und falsch klassifizierte Samples gerichtet wird. Die IBM kann daher allgemein, neben den Anwendungsfeldern der Separierungsprobleme, wie etwa dem Trennen von verschiedenen Sprechern, als unterstützende Maßnahme für Berechnungen mit Hilfe der IRM verwendet werden. In [40] wird hierfür die IBM als Post-Processing Methode angewendet, in dem die Rauschunterdrückung in einzelnen TF-Slots anhand der Ergebnisse aus der IBM angepasst wird. Aufgrund der Tatsache, dass ein Ansatz zur direkten Sprachverbesserung über die IRM ohne weitere Umwege realisierbar sein kann, wird der Fokus im Rahmen dieser Arbeit im Weiteren auf Methoden der IRM liegen.

# 5 Ideal Ratio Mask

## Störunterdrückung mit Residual CNN

Wie bereits diskutiert, eignet sich die IBM nur sehr bedingt zur Sprachverbesserung und sollte in Anwendungsfällen bei denen das Resultat vom Menschen wahrgenommen wird, nicht als Mittel der Wahl gelten [33].

Berechnungsmethoden mit einer IRM bieten zudem den Vorteil, dass eine IBM aus dieser abgeleitet werden kann. Im Rahmen dieser Arbeit wurden eine Vielzahl von Netzkonfigurationen untersucht. Allgemein zeigte sich, dass die Fähigkeit zur Konvergenz eines Netzes von einer Fülle an Parametern abhängt. So können bereits kleinste Änderungen z.B. der Lernrate über den Erfolg des Trainings bestimmen. Aufgrund dieser Tatsache muss beim Experimentieren an dieser Problemstellung Klarheit über die kritischen Parameter herrschen. Dementsprechend werden diese, soweit aufgetaucht, in der Beschreibung der Netze angesprochen. Das Auffinden von funktionalen Netzen zeigte sich insgesamt als nicht trivial. Neben einer Vielzahl an Publikationen in diesem Bereich ist die Bereitschaft zur Veröffentlichung des dazugehörigen Quellcodes in der Forschergemeinschaft eher gering. Aufnahmen bilden hier die bereits angesprochenen Wavenet, SEGAN und das DeepXi Framework. Von diesen bietet, von der Funktionsweise, allerdings nur das DeepXi eine gewisse Nähe zum hier gewählten Ansatz über eine IRM, da dort ebenfalls ein Maskenwert zwischen 0 und 1 geschätzt wird, der eine Abbildung für das instantane a priori SNR liefert. Es wurden für diese Arbeit insgesamt ca. 60 verschiedene Netzkonfigurationen entworfen und generell erprobt. Diese können im beiliegenden Programmcode eingesehen werden. Im Folgenden werden nur die vielversprechendsten dieser Architekturen vorgestellt und evaluiert.

### 5.1 Architektur

Verschiedene Studien haben die Fähigkeit von Residual Blöcken zur Sprachverbesserung demonstriert. In [7] werden ResLSTM beschrieben, in denen Skip-Connections zwischen LSTM Schichten verwendet werden, um die a priori SNR aus spektralen

Amplituden zu schätzen. In [41] werden Dilated Convolutions mit Residual Blöcken kombiniert.

Layer Name	Eingangsdim.	Layer Hyperparameter	Ausgangsdim.
Conv2D	257 x T x 1	3x3, (1,1), 32	255, T-2, 32
BatchNorm	-	-	-
Conv2D	253 x T x 32	3x3, (1,1), 64	253, T-4, 64
BatchNorm	-	-	-
MaxPooling2D	253 x T-4 x 64	(3,3)	84 x T-12 x 64
N x ResNet Block (2xConv2D+BN+Add)	84 x T-12 x 64	3x3, (1,1), 64	84 x T-12 x 64
Conv2d	84 x T-12 x 64	3x3, (1,1), 64	82 x T-14 x 64
AveragePooling2D	82 x T-14 x 64	-	64
Dense	64	-	257
Dropout	257	(0.2)	257
Dense	257	-	257

Tabelle 5.1: Residual Architektur 1

Um die Leistungsfähigkeit von Residual Blöcken zu testen, wurden verschiedene Netzkonfigurationen aus CNNs mit Skip-Connections erprobt.

Eine Konfiguration, die sich als konvergent gezeigt hat, ist in Tabelle 5.1 aufgeführt. Hierbei steht der Parameter T für die Länge des Kontextfensters, also die Anzahl der zurückliegenden zeitlichen Frames. Dieser wurde zunächst zu 16 gewählt. Es ergeben sich 3,3 Millionen trainierbare Parameter für diese Architektur. Die beiden 2D Convolutional Layer, auf die ein Max Pooling folgt, sollen grobe Strukturen aus der STFT Matrix extrahieren. Der Pooling Layer unterstützt die Convolutional Layer bei der Erkennung von Kanten und überführt die Darstellung in eine abstraktere

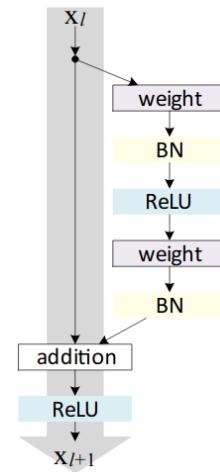


Abbildung 5.1: Residual Block [42]

Form, die eine Überanpassung (engl. *overfitting*) des Netzes verhindern und die Rechenlast verringern soll. Batchnormalisierung als Layer soll die Geschwindigkeit, Stabilität und Performanz der Architektur verbessern. Die Block Architektur ist dabei die in [14] vorgestellte originale Variante, welche in Abbildung 5.1 zu sehen ist. In [42] werden weitere Varianten vorgestellt, bei denen Änderungen der Aktivierungsschichten die Leistungsfähigkeit des Netzes verbessern. Angelehnt an die originale Architektur wurden die Filtergrößen im Residual Block zu 3 und die Anzahl an Filter zu 64 gewählt.

Zum Abschluss folgen ein Average Pooling Layer, sowie ein Dropout, der die Fähigkeit zur Generalisierung verbessern soll. Beim Dropout werden eine gewisse Anzahl an Neuronen während des Trainingsvorgangs abgeschaltet um eine Überanpassung des Netzes an die Eingangsdaten zu mindern.

## 5.2 Trainingsstrategie

Beim Trainieren von neuronalen Netzen stellt sich die Frage, wie das System effektiv aus den bereitgestellten Daten lernen kann. Oftmals werden die Daten im Vorhinein durchmischt, um die statistischen Eigenschaften von Trainings und Testdatensätzen auszugleichen. Der Optimierer muss in der Lage sein, lokale Minima der Fehlerfunktion zu finden, darf aber gleichzeitig möglichst nicht in suboptimalen Minima verweilen. Hierüber entscheiden die Wahl des Optimierers, sowie dessen Lernrate. Für diese Arbeit wurden daher zwei verschiedene Trainingsstrategien eingesetzt. Ein gängiges Mittel der Wahl besteht aus einer Iteration des Optimierers mit festgesetzter Lernrate über den gesamten Datensatz. Diese vollständige Iteration wird als eine Epoche deklariert. Im Anschluss wird der Trainingsdatensatz erneut durchlaufen. Diesmal wird die Lernrate reduziert (engl. *learning rate decay*), was zu geringeren Anpassungen der Gewichte im Netz führt. Dieser Vorgang wird bis zur ausreichend kleiner Fehlerrate wiederholt.

Nun ist im Anwendungsfall die Wahl der Größe des Trainingsdatensatzes gewisserweise arbiträr, da die Anzahl der durchlaufenen Äußerungen beliebig gewählt werden kann. Allgemein wird angenommen, dass Netze mit mehr Informationen bessere Fähigkeiten zur Generalisierung bilden. Andererseits besteht die Möglichkeit, dass die Gewichtsanpassung suboptimal verläuft, wenn der Optimierer die selben Daten erst nach geraumer Zeit erneut sieht, wenn sich in der Zwischenzeit große Änderungen der Gewichte ergeben haben. Vor allem bei Eingangsdaten, bei denen sich die zu erkennenden Features weniger klar herausbilden können als bspw. Kanten in der Bildverarbeitung, könnte sich solch ein Effekt verstärkt bemerkbar machen.

Die Trainingsstrategien wurden daher wie folgt gewählt:

**1. Generatorfunktion**, die kontinuierlich eine Anzahl von Mini-Batches zum Optimierer (Adam) liefert. Der gesamte Datensatz wird durchlaufen, bevor die Lernrate angepasst wird. Trainiert wird über 10 Epochen.

**2. Subdatensätze**, welche aus 200 Batches erzeugt werden. Eine Epochenlänge wird dabei mit einer Iteration durch den Subdatensatz definiert. Trainiert wird ebenfalls über 10 Epochen pro Subdatensatz. Im Anschluss wird der nächste Subdatensatz geladen. Als Optimierer wird Adam verwendet.

Als Trainingsziel wird die IRM im Intervall  $[0;1]$  formuliert. Diese wurde direkt aus dem logarithmierten Spektrum gebildet. Das logarithmierte Spektrum der STFT liegt zwischen 0 dB (Referenz: Vollaussteuerung des Signals im Zeitbereich = 1) und -80 dB. Es wurde also die Formel 2.12 verwendet, jedoch im logarithmierten Spektrum und ohne Parameter  $\beta$ .

Um die gewählten Trainingsstrategien vergleichbar zu machen, wurde ein Lernratenplaner eingefügt. Der Adam Optimierer reduziert seine Lernrate mit jeder neuen Epoche. Es zeigte sich, dass das Durchlaufen des Subdatensatzes als Epoche für den Optimierer gewertet wird, welches bei der gewählten Strategie schnell zu verschwindend kleiner Lernrate führte. Bei Adam ist die Lernrate weiterhin nicht zwingend festgelegt, da sich die Rate für verschiedene Gewichte unterscheiden kann. Der angegebene Wert entspricht daher der *initialen* und damit maximalen Lernrate [10].

$$Lernrate = \begin{cases} 0.001 & Epoche \leq 2 \\ 0.001 \cdot 0.1 \cdot (10 - Epoche) & \text{sonst} \end{cases} \quad (5.1)$$

Eine auf einer Treppenfunktion basierende Lernratenstrategie zeigte sich bei Versuchen als praktikabel. Das Experiment wurde daher mit den Parametern aus 5.1 für beide Strategien durchgeführt. Zudem wurde die Strategie mit Generator und Default Lernrate durchgeführt.

### 5.3 Analyse und Ergebnisse

Nachfolgend werden die Ergebnisse der beiden Strategien zunächst grafisch dargestellt. Anschließend werden diese anhand der Metriken aus Sektion 2.3.2 verglichen.

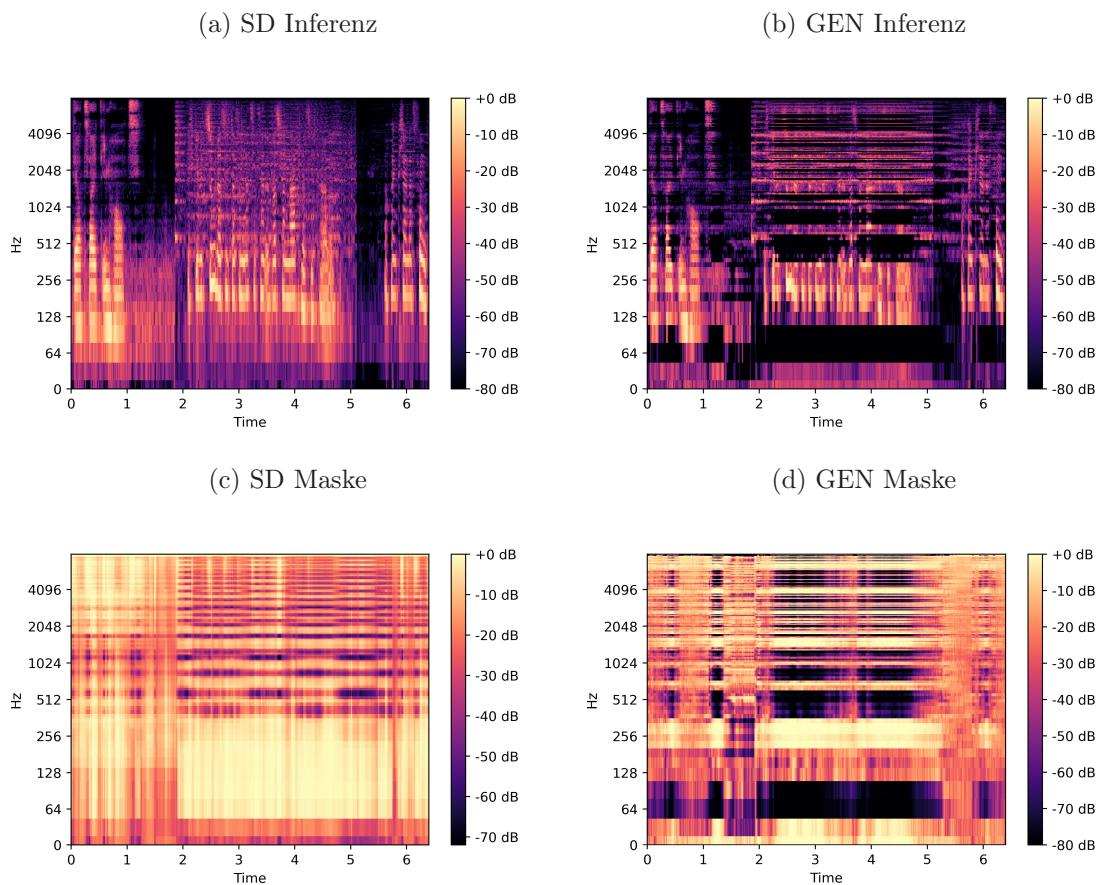


Abbildung 5.2: Vergleich Trainingsstrategien, Kontextfensterlänge: 32

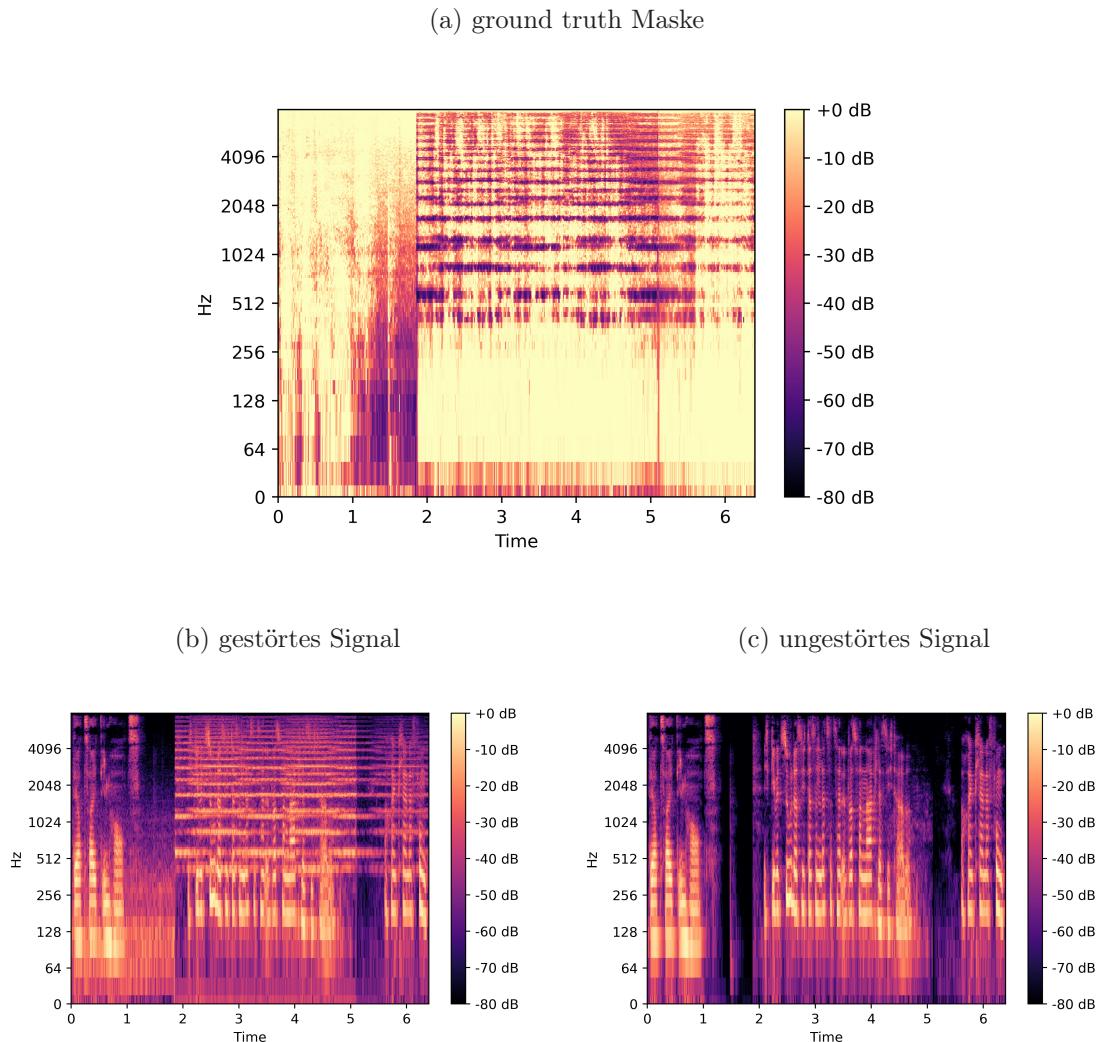


Abbildung 5.3: Signal mit und ohne Störung

Abbildung 5.2 zeigt das Spektrogramm eines Ausschnitts des Datensatzes. Der erste Teil des zu sehenden Signals in (a) besteht aus einer Überlagerung mit Fahrgeräuschen aus einem Autoinnenraum. Ab ca. 2 Sekunden ist das Sprachsignal mit einem Martinshorn überlagert. In (b) ist das Signal ohne Störung zu sehen. Die *ground truth* Maske wird in (c) dargestellt. Das Intervall  $[0,1]$  wurde in eine Darstellung in Dezibel überführt, um die durch die Maske erzeugte Dämpfung wiederzugeben. Die Ergebnisse für die gewählten Trainingsstrategien sind in Abbildung 5.3 zu sehen. Für die Generatorstrategie wurden eine Anzahl von 20 Batches pro Verarbeitungsschritt gewählt. Die Ergebnisse des Generators sind mit GEN, die des Subdatensatzes mit SD bezeichnet. Wie aus den Abbildungen ersichtlich ist, gibt es bei der Generatorstrategie größere zusammenhängende Flächen des Spektrogramms, wel-

che extrem gedämpft sind. Diese kommen auch in Regionen vor, welche gar keine Dämpfung erhalten sollten (Vergleich (c) und (d)). Allgemein zeigen sich bei der Generatorstrategie schärfere Übergänge zwischen Regionen mit niedriger und starker Dämpfung.

Vergleicht man Abbildung 5.3 (a) mit Abbildung 5.2 (c), so wird deutlich, dass die grobe Struktur der Maske Ähnlichkeiten zu der ground truth Maske besitzt. Die SD Maske erzeugt scharf abgetrennte Linien, die die einzelnen Frequenzen des Hornes dämpfen. Im Bereich der Fahrgeräusche sind vermehrt Dämpfungen im Bereich der ground truth Maske zu sehen, jedoch zeigen sich auch Dämpfungen in Bereichen, die diese nicht erhalten sollten.

Aus dem Spektrogramm wird bereits die Überlegenheit der Subdatensatzstrategie deutlich. Die Anzahl der gewählten Batches in der Generatorstrategie muss jedoch auch berücksichtigt werden, da sich für eine höhere Anzahl sowohl Vor- als auch Nachteile beim Update der Gewichte ergeben können. In jedem Fall bleibt bereits festzuhalten, dass die SD Strategie zudem bei niedriger GPU RAM Speicherkapazität eingesetzt werden kann, wenn keine hohen Batchgrößen in diesen geladen werden können um ein funktionales Trainieren zu ermöglichen. Weiterhin muss beim Einsatz dieser Strategie die Lernrate und Anzahl der Epochen austariert werden, da es sonst zu einer Überanpassung kommen kann, welche sich als latent vorkommende Dämpfung über alle Störtypen zeigt. Im Weiteren werden die Ergebnisse der Evaluierungsmetriken in Tabelle 5.2 beispielhaft für einen Rauschtyp gezeigt. Hierbei stehen die in den Reitern verwendeten Kürzel -N für das gestörte Signal und -I für das inferierte Signal

SNR	PESQ-N	STOI-N	SDR-N	PESQ-I	STOI-I	SDR-I	PESQ $\pm$	STOI $\pm$	SDR $\pm$
20	1.9	0.91	13.71	3.14	0.93	7.37	1.24	0.02	-6.34
15	1.57	0.84	6.39	2.86	0.9	7.27	1.29	0.06	0.88
10	1.29	0.81	0.47	2.4	0.9	7.15	1.11	0.09	6.68
5	1.19	0.74	-5.43	1.95	0.86	5.71	0.76	0.12	11.14
0	1.15	0.66	-11.02	1.56	0.79	2.17	0.41	0.13	13.19
-5	1.21	0.6	-16.16	1.26	0.71	-2.78	0.05	0.11	13.38

Tabelle 5.2: Ergebnisse SD Residual CNN, Rauschtyp: Martinshorn

Repräsentativ wird ein Ausschnitt des Ergebnisse mit Generatorstrategie gezeigt.

SNR	PESQ-N	STOI-N	SDR-N	PESQ-I	STOI-I	SDR-I	PESQ $\pm$	STOI $\pm$	SDR $\pm$
15	1.57	0.84	6.39	1.35	0.76	3.09	-0.22	-0.08	-3.3

Tabelle 5.3: Generator Residual CNN, Rauschtyp: Martinshorn

Die Berechnung der Metriken erfolgte auf jeweils 30 Äußerungen mit 6 Sekunden Länge je SNR Wert. Um den SDR Wert zu erzeugen wurde der Mittelwert der Ergebnisse aus einer Berechnung über die STFT Amplituden je Äußerung herangezogen.

Wie in Tabelle 5.3 zu sehen, führt die Generatorstrategie zu keinem nutzbaren Ergebnis für die vorgestellte Architektur. Alle Evaluierungsmetriken bestätigen eine Abnahme der Metriken im Vergleich zum gestörten Signal.

Für die SD-Strategie zeigen sich in Tabelle 5.2 Verbesserungen des PESQ Wertes von bis zu 1.29 für die Störung mit Martinshorn. Für ein SNR von 0 beträgt die Verbesserung des PESQ Wertes lediglich 0,41 Punkte, jedoch kann der STOI Wert um 0,13 gesteigert werden. Die Verständlichkeit der Sprache wird somit erhöht. Weiter ist erkennbar, dass im Bereich mit niedrigem SNR die Störungen, repräsentiert durch die SDR Metrik, um bis zu 13.38 dB reduziert werden können.

Die gewählte Architektur scheint besonders gut in der Lage zu sein, die Störung mit Martinshorn zu entfernen. Die scharf abgetrennten Linien im Spektrogramm können mit Methoden der CNN dementsprechend gut erkannt werden. Die Fortschritte aus dem Bereich der Bildverarbeitung, in denen Objekte aus Bildern mit Hilfe von CNNs segmentiert werden, spricht dafür, dass derartige Strukturen verlässlich erkannt werden können.

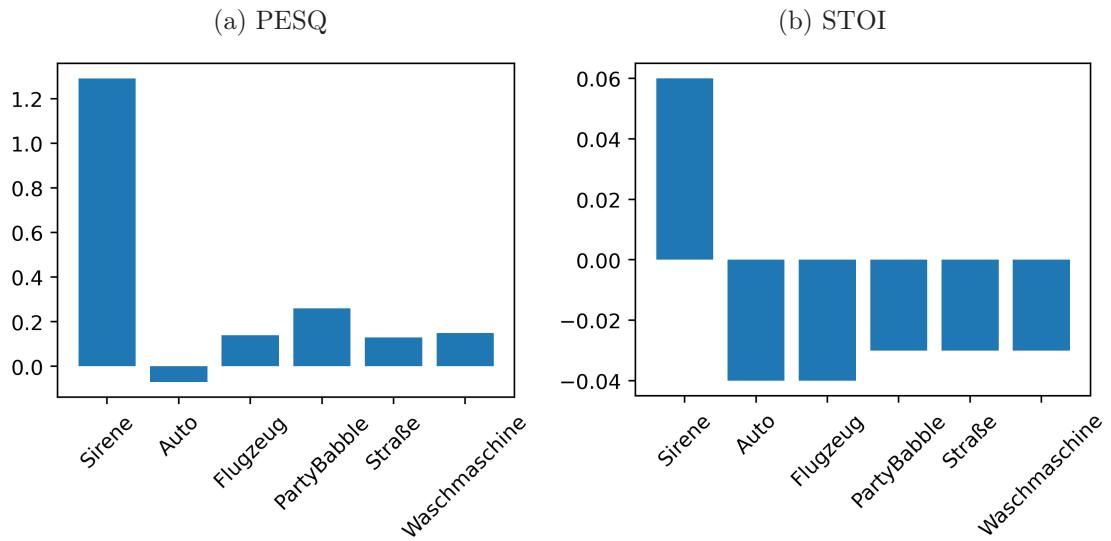


Abbildung 5.4: SD Metrik Änderungen bei 15 db SNR

Abbildung 5.4 zeigt die Ergebnisse für PESQ und STOI mit den gewählten Störgeräuschen bei 15 dB SNR. Es wird deutlich, dass das System für andere Störquellen

als die Sirene eine deutlich schlechtere Performanz liefert. Hier bewegen sich die PESQ Verbesserungen im Bereich von 0,15 bis 0,3 Punkten. Beim Blick auf die STOI Metrik zeigt sich, dass die Verständlichkeit der Sprache durch das System sogar noch leicht gemindert wird.

Abbildung 5.5 zeigt die Änderung des SDR Wertes im inferierten Signal.

Werte über 0 bedeuten eine, im Vergleich zum Signal mit Störung, Verbesserung hinsichtlich der Störabstände zwischen der Amplitude der Leistung aus inferiertem und Referenzsignal. Wie zu erkennen, erhöht sich die SDR also für die meisten Störungen unter einem SNR von ca. 10 db. Aufnahme bildet hier die Störung durch Fahrgeräusche im Auto, bei der bei jedem SNR der Anteil der Störungen verstärkt wird. Somit werden im Bereich von hohem SNR weitere Störungen durch das Sprachverbesserungssystem hinzugefügt, deren Signifikanz von unterschiedlicher Bedeutung sein kann.

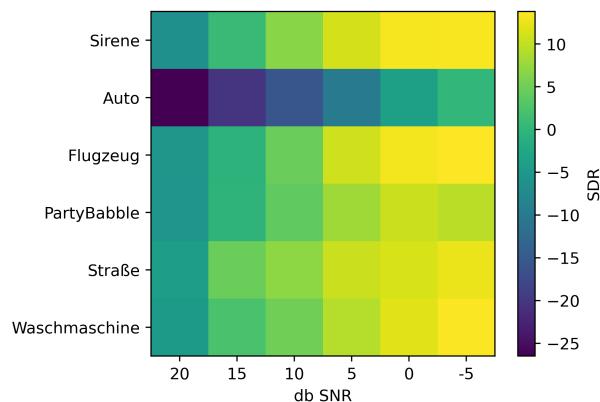


Abbildung 5.5: SDR Änderung

Um zu verstehen, warum gerade die Störgeräusche eines Autoinnenraums besonders schlechte Ergebnisse liefern, wurden die Spektrogramme und Leistungsdichtespektren (LDS) einiger Störungen betrachtet.

Außerdem wurde das gemittelte LDS aus 1000 Auflösungen von ungestörten Sprachsignalen in Abbildung 5.6 gebildet. Abbildung 5.7 zeigt die LDS von Störungen im Auto und durch ein Flugzeug. Für die Erzeugung des LDS wurde die Welch-Methode verwendet. Wie in Abbildung 5.7 (c) zu sehen ist,

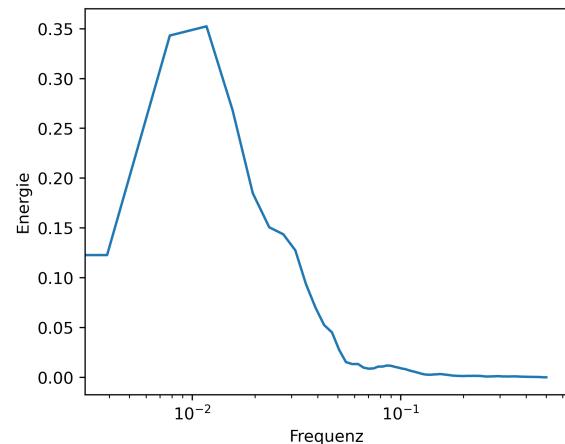


Abbildung 5.6: LDS Sprache

flacht das LDS der Autoinnenraumstörung bei einer normierten Frequenz von ca.  $10^{-2}$  stetig und schnell ab. Dies wird auch aus der Betrachtung des Spektrogramms deutlich. Vergleicht man das LDS zwischen dieser Störung und der aus den Sprachaufnahmen, zeigt sich eine gewisse Überlappung der spektralen Komponenten, allerdings gibt es auch Bereiche in denen die Energie der Sprache häufiger präsent ist.

Große Überlappungen der Komponenten erschweren eine erfolgreiche Sprachverbesserung. Bei der Betrachtung des LDS und des Spektrogrammes für die Störung im Auto kann die Vermutung entstehen, dass CNN Schwierigkeiten haben könnten, die stationäre Charakteristik dieser Störung zu erfassen. Dies könnte im Zusammenhang mit den Fähigkeiten von CNNs zur Erkennung klarer Strukturen stehen.

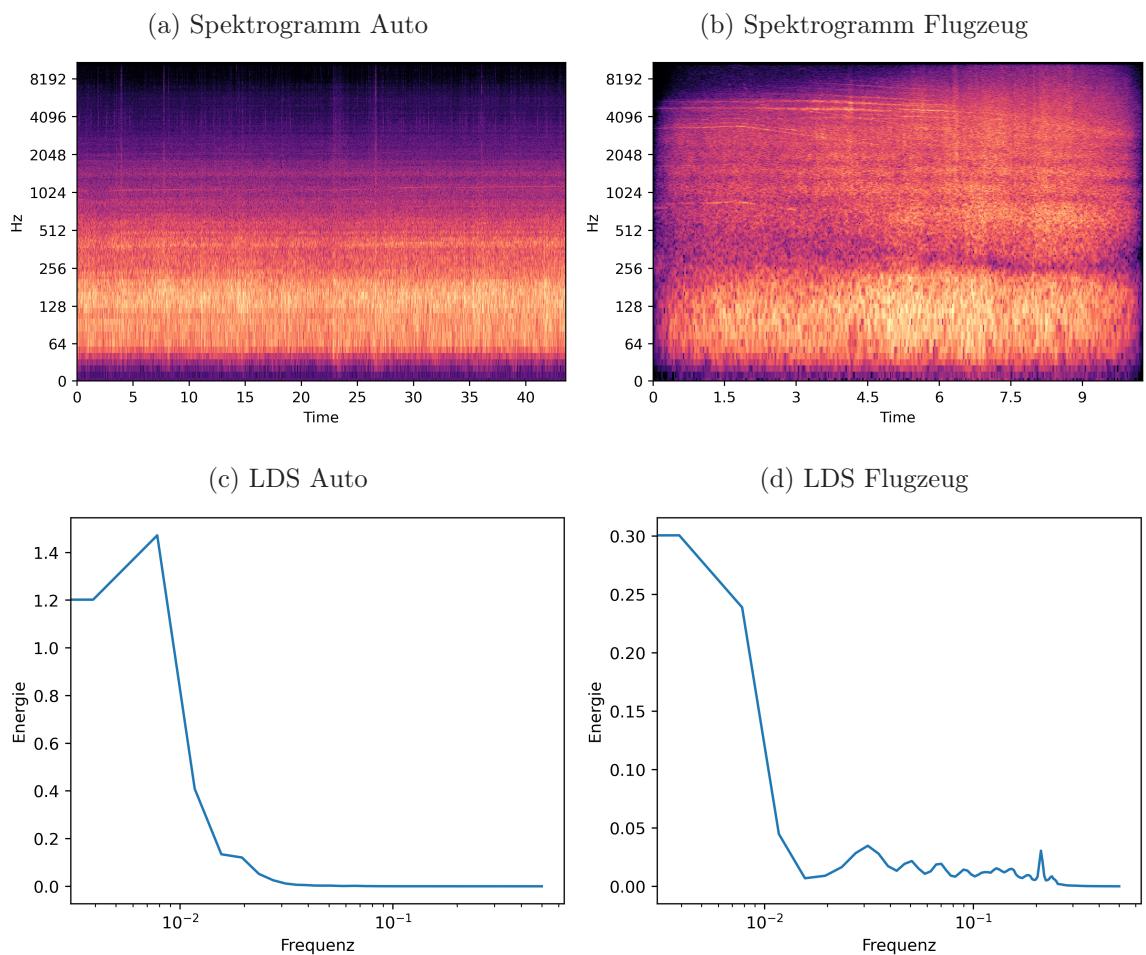


Abbildung 5.7: Spektrale Leistungsdichte und Spektrogramm

Bei der Analyse von CNN Filtern, welche auf einem MNIST Datensatz erzeugt wurden, kann gesehen werden, dass Kanten, Kurven und Winkel als Features erkannt werden [A14]. Ein Erkennen von Struktur muss daher für eine Umgebung mit hoher spektraler Varianz, wie dem Störgeräusch im Auto, für die gewählten CNN Filter schwierig sein. Durch ständig wechselnde Muster der Amplituden innerhalb eines Filterfensters, welche sich im Bereich ihres Auftretens scheinbar zufällig ergeben, kann der Optimierer nicht zu einem Funktionsminimum konvergieren um somit eine eindeutige Feature Map zu definieren. Möglicherweise müsste daher für diesen Frequenzbereich eine andere Konfiguration, mit bspw. erweiterter Filtergröße gefunden werden, welche die Struktur des ungestörten Signals besser separieren kann.

## 5.4 Anpassung der Architektur

Es wurde gezeigt, dass die gewählte Architektur in Kombination mit der SD Trainingsstrategie grundlegend in der Lage ist eine Sprachverbesserung zu erreichen. Jedoch unterscheidet sich die Perfomanz auf unterschiedlichen Störgeräuschen enorm. Es kann daher angedacht werden, diese Architektur weiter zu optimieren. Möglich wären hier verschiedene Längen des Kontextfensters. Auch die Anzahl der Residual-Schichten sollte variiert werden. Der Adam Optimierer stellt für die Charakteristik der Eingangsdaten möglicherweise nicht die beste Wahl dar. Diese sind dünn besetzt. (engl. *sparse data*). In der Literatur werden für diese Eigenschaft AdaGrad und AdaDelta als Alternative häufig aufgeführt. Weiterhin muss überprüft werden, ob die gewählte Anzahl an Parametern ausreichend ist, oder ob mit größerer Anzahl eine Verbesserung erreicht werden kann, ohne dass das Netz in Überanpassung geht. Hierfür werden üblicherweise die Differenzen der Fehlergrößen zwischen Trainings und Testdatensatz herangezogen. Auch müsste, bei gewählter SD Trainingsstrategie austariert werden, in wie weit die Anzahl der Trainingsepochen erhöht werden kann, bis es auch hier zu einer Überanpassung an die Daten kommt. Es wurden einige weitere Versuche durchgeführt, die aber zu keiner bedeutenden Verbesserung des Systems beitrugen. So konnte bspw. jedoch gesehen werden, dass bei einem Training mit 20 Epochen und SD Strategie eine Überanpassung erfolgt, so dass Regressionskomponenten aus der Störung mit Martinshorn auch latent in den Masken für andere Störarten vorkommen.

# 6 Ideal Ratio Mask

## Störunterdrückung mit Dilated CNN+LSTM

Aufgrund des gezeigten Unterschieds in der Entstörungsleistung, in Konklusion mit der Beschaffenheit der Störtypen, wurde eine weitere Architektur formuliert, in welcher die Vorteile der Strukturerkennung von CNN mit den Fähigkeiten zur Zeitreihenanalyse von LSTMs kombiniert werden.

### 6.1 Architektur

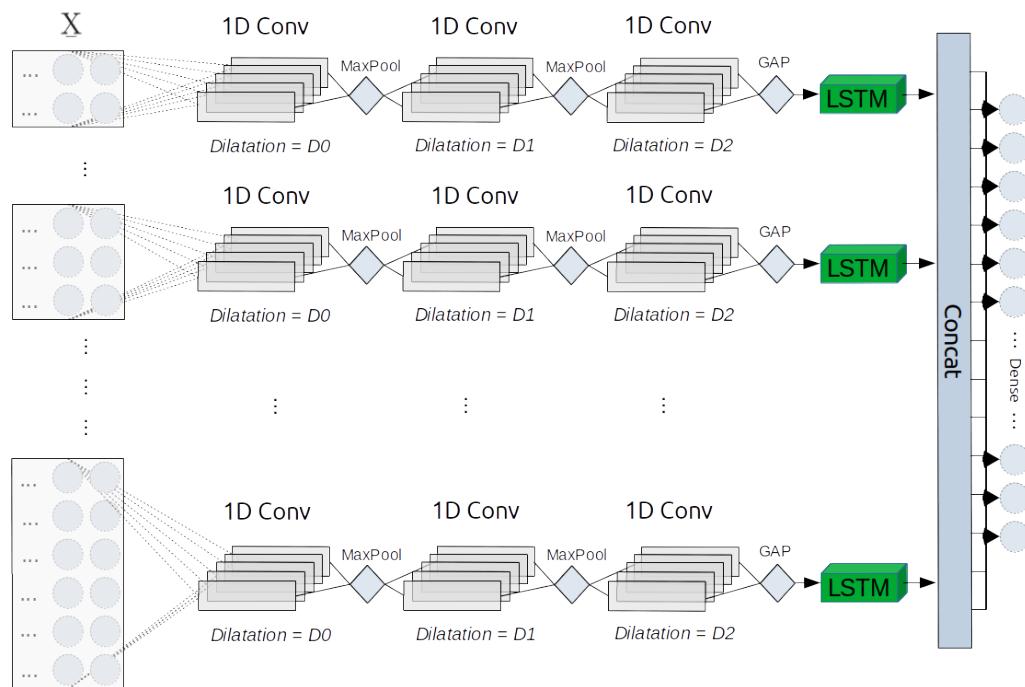


Abbildung 6.1: Dilated 1D-CNN - LSTM Architektur

Abbildung 6.1 zeigt den Aufbau der Architektur. Eine Aneinanderreihung von CNN und LSTM wurde im EHnet, gezeigt in Sektion 2.3.3, verwendet. Im Rahmen dieser Arbeit wird eine neuartige Architektur vorgestellt, die zur Maskenberechnung drei eindimensionale CNN mit Dilatation, gefolgt von einer LSTM Schicht und anschließender Konkatenierung, nutzt. Diese Architektur ist dabei das Ergebnis von ausgiebiger Testung verschiedener Anordnung der gewählten Techniken und Funktionen. Die Dilatation der CNN wird hierbei inkrementell mit der Tiefe des Netzes gebildet. Hierdurch sollen längerfristige Abhängigkeiten erkannt werden. Die Struktur einer inkrementellen Dilatation ist dabei an verschiedene Ergebnisse aus der Literatur angelehnt. Die Eingangsdaten werden in eine Anzahl B von Subbändern unterteilt, die jeweils parallel verarbeitet werden. Die Einteilung richtet sich nach der Darstellung der Bark-Skala, deren Funktionsweise bereits in Sektion 4.2 erörtert wurde. Dies erfolgt in der Annahme, dass die Signalverarbeitung im menschlichen Ohr ebenfalls in Bark-Frequenzbändern geschieht. Features sollen hier auf dieser Grundlage erkannt werden. Zwischen den dilatierten CNN Schichten erfolgt ein 1D MaxPooling, welches sich durch Testung als notwendig zur Funktionalität erwies. Zwischen CNN und LSTM erfolgt ein globales Mittelwert-Pooling (engl. *global average pooling*), welche als Grundlage für den LSTM Feature Vektor dienen soll. Nach der Konkatenierung der LSTM Ausgänge folgen optional ein- oder mehrere Dense Layer mit oder ohne Dropout, die den Ausgang des Systems bilden.

## 6.2 Analyse und Ergebnisse

Für die Analyse der Architektur wurde diese wie in Tabelle 6.1 zu sehen konfiguriert.

	Dilatation	Filter	Fenstergröße
<i>60 x CNN_1D</i>			
cnn_1	1	64	2
cnn_2	2	64	4
cnn_3	4	64	8
MaxPool	-	-	2
<i>60 x BLSTM</i>		32	-

Tabelle 6.1: Netzkonfiguration mit 60 CNN

Um herauszufinden, welche Trainingsstrategie sich für dieses Netz eignet, wurde dieses erneut mit den in Sektion 5.2 vorgestellten Strategien trainiert. Auch sollte sich an dieser Stelle zeigen, ob die gewählte Subdatensatz Strategie aufgrund der Beschaffenheit der Daten besser funktioniert, bzw. schneller konvergiert, oder ob die Netzarchitektur selbst der ausschlaggebende Grund ist.

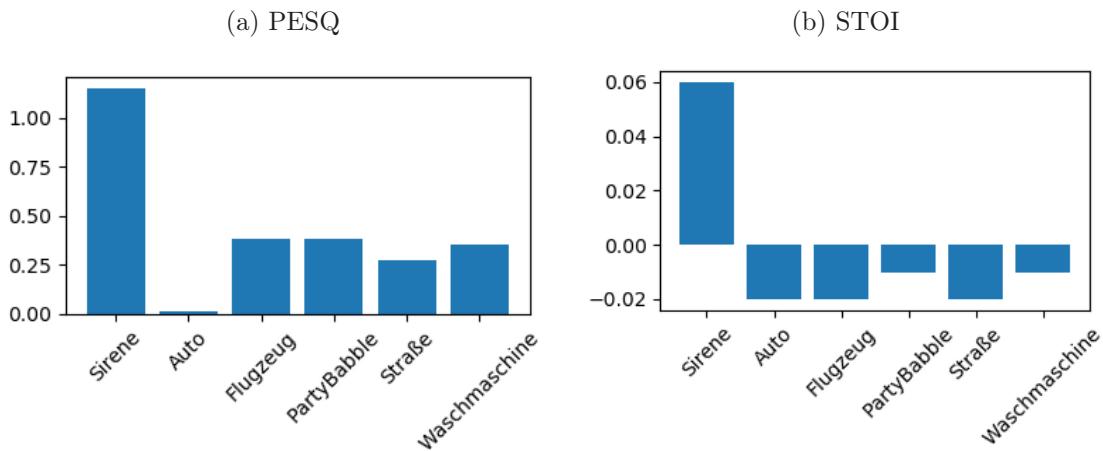


Abbildung 6.2: SD Metrik Änderungen bei 15 db SNR

Hierfür wurde erneut der Trainingsdatensatz mit jeweils 1500 Äußerungen genutzt, welcher in 10 Epochen mit beiden Strategien durchlaufen wird. Abbildung 6.2 zeigt die Änderungen der PESQ und STOI Metrik durch Training mit SD Strategie für ein SNR mit 15 dB. Beim Vergleich mit den Ergebnissen aus Sektion 5.3 wird deutlich, dass die D-CNN+LSTM Architektur bei allen Störtypen, bis auf die Sirenenstörung, bessere Werte erreicht. So verbessert sich bspw. die Erhöhung des PESQ Wertes von 0.18 auf 0.4 bei einer Störung mit Flugzeuggeräuschen. Bei der ResNet Architektur verringerte sich zuvor der STOI Wert bei den Störungen im Auto und durch ein Flugzeug um 0.04. Hier sind es nur noch 0.02. Die Werte für die Störungen mit der Sirene bleiben bei beiden Architekturen ungefähr gleich. Allgemein zeigt sich also, dass die gewählte Architektur besser in der Lage ist, die von der Art her in der STFT „strukturloseren“ Störungen besser zu unterdrücken und damit die Qualität der Sprache zu erhöhen. Hierbei werden, im Vergleich zur ResNet Architektur, weniger Störungen in die Sprache induziert, welche die Verständlichkeit der Sprache beeinträchtigen. Ein Absinken des STOI Wertes um 0.02 zeigt sich zudem als eine eher geringe Beeinträchtigung.

Weiter werden die Ergebnisse für die PESQ Metrik über alle SNRs in Abbildung 6.3 gezeigt.

Es wird deutlich, dass die Verläufe für die Störungen „Flugzeug“, „PartyBabble“, „Waschmaschine“ und „Straße“ ähnlich sind. Die Performanz des Systems nimmt mit abnehmendem SNR Wert ab. So befinden sich die Verbesserungen des PESQ Wertes für ein SNR von 0 im Bereich von 0.1 für die meisten Störungen. Schlechte Perfomanz von Sprachverbesserungssystemen für niedrige SNR ist eine in der Literatur bekannte Problematik. So werden z.B. in [46] Phaseninformationen als Gegenmaßnahme mit in eine KNN Regression einbezogen.

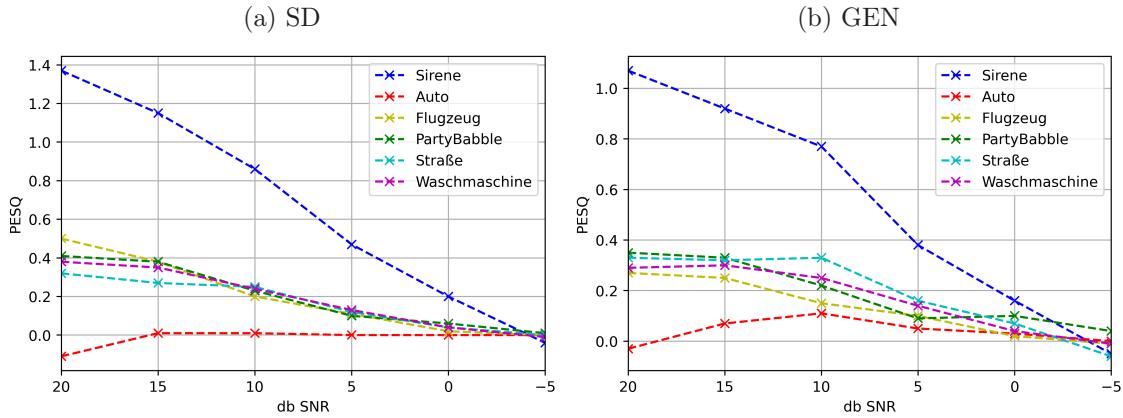


Abbildung 6.3: PESQ Inkrement in Abhängigkeit des SNR für SD und GEN Strategie

Im Vergleich der beiden Trainingsstrategien fallen Unterschiede in der Performanz zwischen Generator und Subdatensatz auf, jedoch sind diese nicht so ausgeprägt wie in der ResNet Architektur. Allgemein zeigen sich Wertunterschiede von 0.1 bis 0.3 auf der PESQ Metrik. Auffällig ist jedoch, dass einzig die Störung aus dem Autoinnenraum mit der Generatorstrategie eine Verbesserung erfährt, auch wenn diese nur minimal ist.

Es besteht die Möglichkeit, dass die Generatorstrategie mit längerem Training ebenfalls die Werte der Subdatensatzstrategie erreichen kann. Festzuhalten bleibt daher, dass die Architektur eine nicht unerhebliche Rolle bei der Wahl der Trainingsstrategie spielen muss. Aufgrund der schnelleren Konvergenz sollte daher die SD-Strategie bei Architekturen, welche STFT Daten verarbeiten, auch (zusätzlich) angewendet werden. Der Versuch mit Generatorstrategie wurde mit einer Batchgröße von 200 wiederholt um eine Referenz zur SD-Strategie zu bilden, bei der die Größe des SD ebenfalls aus 200 Batches besteht. Hierbei zeigt sich allgemein eine geringere Performanz als im Versuch mit 20 Batches und Generatorstrategie mit Werten um 0.6 Inkrement in PESQ für das Martinshorn und 0.15 für andere Störungen. Dies bestätigt weiter die Effizienz der vorgestellten SD-Trainingsstrategie.

Die gezeigte Architektur wurde zudem in Teilen angepasst. Zunächst wurde die Anzahl der Dilated CNN von 60 auf 23 reduziert. Hierbei zeigt sich eine Verschlechterung der PESQ Metrik um durchschnittlich 0.2 Punkte im Vergleich zur Variante mit 60 D-CNN. Folgend wurde die Anzahl der D-CNN auf 128 erhöht, so dass zwei bzw. drei Frequenzbänder der STFT zusammen verarbeitet werden. Für diese Konfiguration zeigt sich keine Konvergenz. Es scheint daher, als seien aus nur zwei Frequenzbändern zuwenig Information bzw. Features mit einer 1D-Convolution über die Zeitachse erkennbar.

### 6.3 Convolution auf der Frequenzachse

In [47] wird eine Encoder-Decoder Struktur genutzt, bei der die Convolutions nur über die Frequenzachse durchgeführt werden. Die vorgestellte D-CNN-LSTM Architektur wurde daher angepasst, so dass aus einem Kontextfenster der Länge 32 jeweils zwei zeitliche Frames mit einer 1D-Convolution über die Frequenzachse verarbeitet werden. Das Netz wurde über 25 Epochen mit Generatorstrategie trainiert, die Parameter für Dilatation und Fenstergröße wurden, wie in Tabelle 6.1 gezeigt, übernommen, jedoch mit einer entsprechenden Anzahl von 16 parallelen CNN.

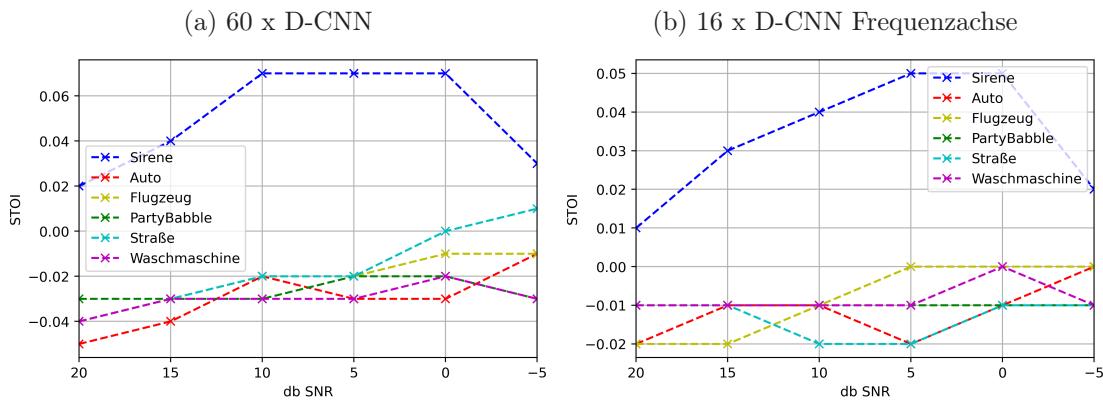


Abbildung 6.4: STOI Vergleich zwischen Convolution auf Zeit/Frequenzachse

Ein Vorteil dieser Struktur liegt in der dadurch geringeren Anzahl an Parametern. Im direkten Vergleich zum vorhergegangen Test zeigen sich jedoch nur minimale Unterschiede bei der Auswertung der PESQ Metrik. Für die STOI Metrik zeigen sich minimal geringere schädliche Auswirkungen durch die Inferenz, welche in Abbildung 6.4 gezeigt werden. Aufgrund der deutlich verringerten Anzahl an Parametern bei ähnlicher Performanz auf den Evaluierungsmetriken, sollte dieser Versuch mit einer gesteigerten Anzahl an Filtern im D-CNN wiederholt werden. Ein weiterer Versuch mit 32 parallelen D-CNN-LSTM, sodass jeder zeitliche Frame parallel verarbeitet wird, zeigen sich nach 10 Epochen mit Diskrepanz zwischen Trainings- und Testfehler. Dies spricht für fehlende Fähigkeit zur Generalisierung.

## 6.4 Dilated-Residual-CNN-LSTM

Wie in den vorangegangenen Kapiteln beschrieben, können Residual Verbindungen den Informationsfluss durch das Netz verbessern und tiefe Netze ermöglichen, indem das Problem der vanishing gradients abgeschwächt wird. Aufgrund der im D-CNN-LSTM hintereinandergeschalteten 1D Convolutions mit variabler Fenstergröße und Dilatation besteht die potenzielle Option im Schwinden von relevanter Information zwischen den CNN-Schichten. Daher wurden die 1D-Convolution Layer durch Residual Layer mit der in Abbildung 5.1 gezeigten Architektur ersetzt. Entsprechend wurden die 2D Convolutions im Residual Block durch 1D Convolutions ersetzt. In einem ersten Versuch zeigte sich bei der Trainingsevaluierung erneut eine größere Diskrepanz zwischen Trainings und Testdatensatz, sodass nach zehn Epochen noch eine durchschnittliche mittlere Abweichung von 17 % bei Schätzung der Maske auf dem Testdatensatz bestand. Da Residual Verbindungen besonders tiefe Netze ermöglichen, wurde im Folgenden die Res-D-CNN Struktur erweitert, sodass bei drei Strukturen mit festgesetzter Dilatation und Fenstergröße jeweils acht der beschriebenen Residual Architekturen hintereinander verwendet werden. Nachteilig für diese Architektur ist eine besonders lange Trainingszeit. Bei 11 Millionen Parametern benötigt eine Epoche ca 4,5 Stunden. Es zeigt sich, dass das Netz nach drei Epochen nur eine geringfügige Abnahme der Fehlerfunktion auf den Trainingsdaten vorweisen konnte. Dies bedeutet, dass an dieser Stelle eine Erweiterung des Netzes mit Residual Verbindungen keine weitere Befähigung zur Extraktion relevanter Features liefert. Ein möglicher deutlicher Rückgang der Fehlerfunktion bleibt für ein intensiveres Training mit einer Vielzahl an Epochen jedoch nicht endgültig ausgeschlossen.

# 7 Ideal Ratio Mask Störunterdrückung mit DenseNet Implementierung

Als weitere Alternative wurde die 2016 vorgestellte DenseNet [43] Architektur, in Bezug auf die Fähigkeit zur Sprachverbesserung, untersucht. In einer aktuelleren Veröffentlichung werden die Dense-Block Strukturen aus DenseNet mit Residual Connections kombiniert, um eine Überallokierung von Parametern zur Wiederverwendung von Features, wie sie im DenseNet entsteht, zu reduzieren [44].

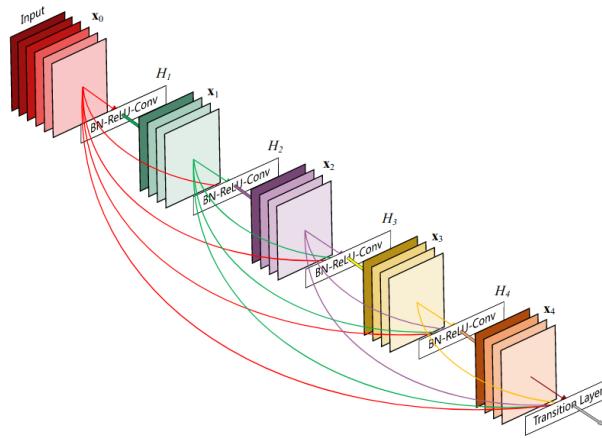


Abbildung 7.1: DenseNet Block [43]

In DenseNet wird jeder Convolutional Layer vorwärtsgerichtet mit jedem weiteren Layer verbunden. Dies steht im Kontrast zu den subsequenten Verbindungen aufeinanderfolgender Standard CNNs. Abbildung 7.1 zeigt hier die Struktur eines Dense Blockes. Die Vorteile der DenseNet Architektur liegen in der Reduzierung des *vanning gradients* Problem und der Propagierung von Features durch das Netz bei Reduzierung der Parameter im Vergleich zu anderen Methoden.

In Abbilung 7.2 wird die Architektur des DenseNet für drei Blöcke gezeigt. Zwischen den Blöcken bestehen Schichten aus Convolution und Pooling. Diese werden als *Transition Layer* bezeichnet und sorgen für eine Änderung der Feature Map Größen.

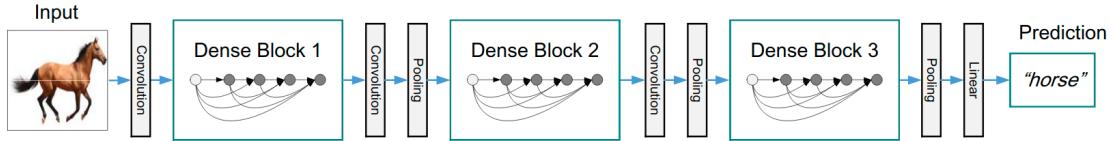


Abbildung 7.2: DenseNet Architektur [43]

In [43] werden vier verschiedene Architekturen mit unterschiedlicher Anzahl an Dense Blöcken vorgestellt.

Für diese Arbeit wurde die DenseNet-121 und DenseNet-169 Architektur ausprobiert. Dafür wurde die letzte Schicht an die 257 FFT Bins angepasst und die Fehlerfunktion zu MSE gewählt. Es zeigt sich, dass die DenseNet-169 Architektur mit der gewählten Generatorstrategie und 10 Epochen nicht konvergiert. Für Densenet-121 zeigten sich die Ergebnisse für die PESQ Metrik mit Generatorstrategie in Abbildung 7.3. Ein Trainingsversuch mit SD Strategie zeigt hier keine Konvergenz. Außerdem wurde ein weiterer Versuch mit von 0.001 zu 0.002 geänderter Lernrate und Generatorstrategie durchgeführt, für den sich auch keine Konvergenz mehr zeigt.

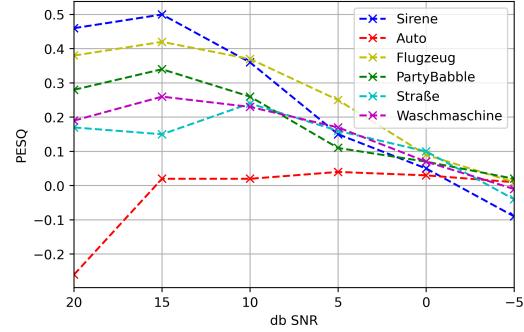


Abbildung 7.3: Densenet PESQ

## 7.1 DenseRNet

Potenziell kann eine Densenet Architektur relevante Feature zur Sprachverbesserung erkennen. Jedoch zeigt sich, dass bei der originalen Struktur Konvergenz nur in einem sehr kleinen Bereich der gewählten Hyperparameter stattfindet. In [48] wird ein akustisches Modell zur Spracherkennung vorgestellt. Hier wurde die Convolution innerhalb der Dense Blöcke durch Residual Blöcke ersetzt. Bei einem Versuch mit entsprechend angepassten Dense Blöcken, zeigt sich jedoch erneut eine fehlende Fähigkeit zur Generalisierung für die Aufgabe der Sprachverbesserung mit großer Diskrepanz der Fehlerfunktion zwischen Trainings und Testdaten. Für weitere Untersuchungen sollte daher die Architektur aus [44] betrachtet werden, bei der ebenfalls Residual- mit Dense Blöcken kombiniert werden.

## 8 Anpassen der Fehlerfunktion

Systeme zur Sprachverbesserung sind meist in der Lage, die Qualität der inferierten Sprache zu verbessern, jedoch wird hierbei häufig die Verständlichkeit reduziert. Dies ist ein in der Literatur bekanntes Problem [49]. Für stationäres Rauschen kann mit Hilfe von spektraler Subtraktion eine minimale Verbesserung der Verständlichkeit erreicht werden. Die Gründe für die fehlende Fähigkeit liegen u.A. in der angewendeten Fehlerfunktion, deren Minimierung nicht zwingend mit einer Verbesserung der Verständlichkeit korreliert. Bei Approximation mit dem MSE kommt es zu zwei verschiedenen Störungen, da es für die Funktion keinen Unterschied macht, ob die Differenz zwischen wahrem und geschätztem Signal positiv oder negativ ist. Die dadurch auftretenden Störungen müssen zudem als unterschiedlich prekär für die Verbesserung der Sprachverständlichkeit erachtet werden.

Eine mögliche Abhilfe kann durch das Einbinden der objektiven Evaluierungsmetriken wie PESQ, STOI oder SDR, welche in Sektion 2.3.2 beschrieben sind, in die Fehlerfunktion, erreicht werden.

Jedoch bereitet gerade der in dieser Arbeit gewählte Ansatz mittels Maskierung hierfür eine Schwierigkeit. Das KNN beschränkt sich auf die Ausgabe eines 257 Bin langen Vektors aus den vorangegangenen Frames variabler Länge. Dieser Vektor beeinhaltet die geschätzten Maskenwerte der IRM und nicht die Amplituden im Zeitbereich des Sprachsignals, welche zur Berechnung der Metriken verwendet werden. Eine ideale Lösung des Problems bestünde im Aufbau eines Netzgraphen, welcher die vorangegangenen Eingangswerte speichern sowie eine Umwandlung und Berechnung durchführen kann. Zudem bräuchte es einen Generator, der den Datensatz in entsprechend parallelisierbare Eingangsdaten aufteilt.

In Versuchen wurde festgestellt, dass für eine Standard Berechnung des PESQ ein Zeitabschnitt bestehend aus 500 Frames der STFT benötigt wird, damit es zu keinem Abbruch durch Fehler kommt. Bei der Berechnung eines einzelnen zeitlichen Framevektors wäre eine Aussage über den PESQ Wert der letzten 12 Sekunden höchstwahrscheinlich auch nicht mehr zuträglich. Die STOI Metrik benötigt zur Berechnung 400 ms lange Abschnitte.

## 8.1 PMSQE

In [50] wurde für dieses Problem eine Fehlerfunktion vorgestellt, welche eine Berechnung für einzelne zeitliche Frames, basierend auf der PESQ Metrik, ermöglicht. Hierbei werden die im PESQ Standard vorkommenden Störungsterme als differenzierbare Funktion gebildet. Diese bestehen aus einem *symmetrischen* und einem *asymmetrischen* Term. Der symmetrische Term beschreibt die absoluten Unterschiede der Leistungsspektren, wenn diese auf einer wahrgenommenen Lautstärkeeskala betrachtet werden. Der asymmetrische Term wird aus dem symmetrischen Term berechnet, jedoch werden positive spektrale Differenzen anders als negative gewertet. Dies geschieht aufgrund der unterschiedlichen menschlichen Wahrnehmung von zusätzlich induzierter Störung, oder zu großer Abschwächung der Amplituden.

Aufgrund des gewählten Maskierungsansatzes wurde die Berechnung zunächst außerhalb des Tensorflow Graphen durchgeführt, was zu einer deutlich erhöhten Verarbeitungszeit führt. Mögliche Abhilfen können durch eine Verarbeitung der Signale im Zeitbereich erreicht werden, wenn die einzelnen Batches der Mindestlänge zur Berechnung entsprechen. Ebenfalls möglich ist eine direkte Schätzung des Spektrums im Frequenzbereich, sowohl für einzelne Frames als auch größere Fenster, wie im EHNet (Sektion 2.3.3). Für Schätzungen im Zeitbereich werden im Rahmen einer aktuellen Veröffentlichung in [51] Fehlerfunktionen zur Berechnung von PESQ, STOI und ESTOI bereitgestellt. Diese könnten leicht auf Berechnungen mit Daten der STFT angepasst werden.

Um die beschriebene Problematik zu umgehen, wurde das Netz um einen zweiten Ausgang mit PMSQE Fehlerfunktion erweitert. Die Funktionsweise ist in Abbildung 8.1 gezeigt.

Hier wird nun die Berechnung des Leistungsspektrums innerhalb des Tensorflow Graphen vorgenommen, indem die Maske des Output1 mit den entlogarithmierten Eingangsdaten multipliziert wird. Output1 wird dabei weiterhin mit der MSE Fehlerfunktion optimiert. Auf Output2 wird der beschriebene PMSQE Loss angewendet. Die Gesamtfehlerfunktion wird gebildet durch:

$$E = E_{MSE} + \alpha E_{PMSQE} \quad (8.1)$$

Hierbei wurde nach Sicht auf einen Trainingsvorgang der Parameter  $\alpha = 0.01$  gewählt, so dass die Gewichtung der Fehlerfunktion des PMSQE effektiv für ca. 25 % der in der Fehlerfunktion auftretenden Werte verantwortlich ist.

Das in Abbildung 8.1 gezeigte Netz wurde mit einer Generatorstrategie für 10 Epochen trainiert. Die Ergebnisse des Trainings sind in Abbildung 8.2 zu sehen. Auffällig ist hier, dass sich die PESQ Metrik Änderungen im Vergleich zum Referenzversuch

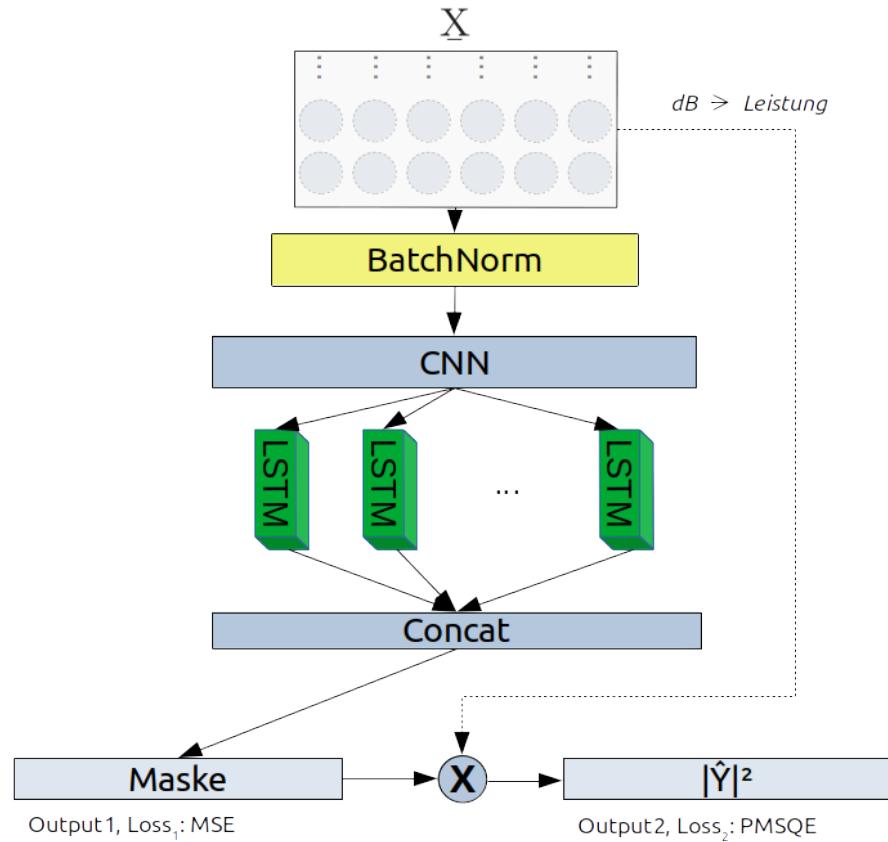


Abbildung 8.1: Architektur mit zwei Ausgängen und PMSQE Fehlerfunktion

aus Sektion 6 für alle Störtypen, mit Ausnahme der Störung durch Fahrgeräusche im Autoinnenraum, verschlechtert haben. Für die Störung durch Straßenlärm kann nahezu keine Verbesserung des PESQ Wertes gesehen werden. Die Störung im Auto ist hier von besonderer Bedeutung, da in allen vorangegangenen Versuchen für diesen Störtyp keine Verbesserung bzw. teils sogar eine Abnahme der Metrik zu sehen ist. Die PMSQE Fehlerfunktion scheint besser in der Lage ist, diese Störung abzubilden. Eine mögliche Erklärung könnte die granulare Struktur sein, welche durch ständigen Wechsel der Amplituden dem MSE Optimierer nicht ermöglicht die Funktionsminima zu finden. In der angesprochenen Veröffentlichung der Fehlerfunktion werden MSE und PMSQE mit gleicher Gewichtung angesetzt. Dies führte in einem ersten Versuch jedoch zu keiner schnellen Konvergenz mit der vorgestellten SB-D-

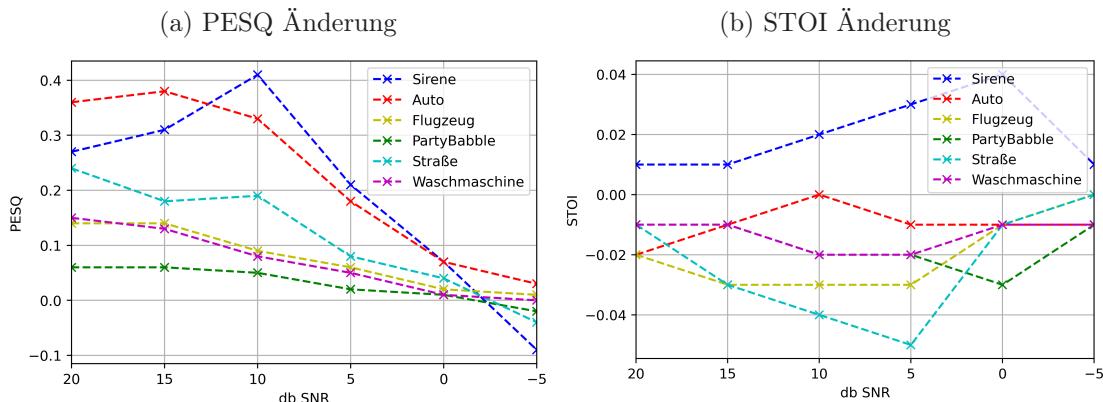


Abbildung 8.2: Architektur mit MSE+PMSQE Fehlerfunktion

CNN-LSTM Architektur. Die Ursachen können in der initialen Lernrate liegen bzw. der hier geänderten Konfiguration mit Bildung des Wertes auf einzelnen Frames. Daher müsste der Versuch mit Änderungen des Parameters  $\alpha$  der Fehlerfunktion, sowie weiteren Lernraten, wiederholt werden, um Klarheit über die Möglichkeiten von angepassten Fehlerfunktionen zu erlangen.

## 9 Störunterdrückung mit A-priori SNR als Trainingsziel

Bislang wurde im Rahmen dieser Arbeit die IRM (Formel 2.12) als Trainingsziel zur Sprachverbesserung untersucht. Wie bereits angesprochen, werden bei Systemen wie dem DeepXi a priori SNR der TF-Slots geschätzt, um anschließend Gain-Funktionen zu berechnen. Zu den aus der klassischen statistischen Signalverarbeitung gehörenden Gain Funktionen gehört u.A. auch der Wiener Filter Ansatz. Dieser ist stark mit der IRM verwandt. Wenn die Wurzelfunktion auf den Wiener Ansatz angewendet wird, ergibt sich die IRM aus Formel 2.12 mit  $\beta = 0.5$  [52]. Daher wurde untersucht, ob die bislang vorgestellten Netzarchitekturen neben der IRM auch in der Lage sind, geeignete Schätzungen für das a priori SNR zu leisten.

Die Formel zur Berechnung des a priori SNR für einzelne TF-Slots ist in Formel 2.9 gegeben. Aus [7] ist bekannt, dass neuronale Netze schneller konvergieren, wenn die Zielfunktion im Interval  $[0;1]$  abgebildet ist. Zur Erstellung der Abbildungsfunktion nutzten die Autoren die Wahrscheinlichkeitsdichtefunktion der auftretenden SNR. An dieser Stelle wird eine Sigmoid Abbildungsfunktion, welche in Abbildung 9.1 (a) zu sehen ist, genutzt. Die Zielmaske wurde mit Mittelwert 0 normalisiert.

$$Maske(t, f) = \frac{1}{1 + e^{-0.1 \cdot \hat{SNR}_{prio}(t, f)}} \quad (9.1)$$

Die Gain Funktion des Wiener Filters ist in Abbildung 9.2 zu sehen und wird bestimmt durch:

$$Gain(t, f) = \frac{\hat{SNR}_{prio}(t, f)}{1 + \hat{SNR}_{prio}(t, f)} \quad (9.2)$$

Die Vorteile einer Architektur, die zuverlässig a priori SNR Werte schätzen kann,

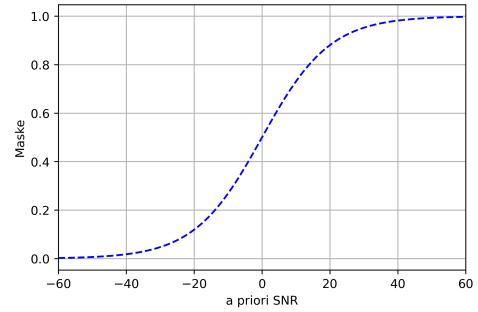


Abbildung 9.1: Abbildungsfunktion a-priori SNR

liegen darin, dass verschiedene Gain-Funktionen auf diese angewendet werden können, sowie darin, dass ein Fundus an Forschungsergebnissen der letzten Dekaden existiert, der auf statistischen Methoden basiert, mit dem die Funktionalität weiter abstrahiert werden kann.

## 9.1 Parametrisierter Wiener Filter

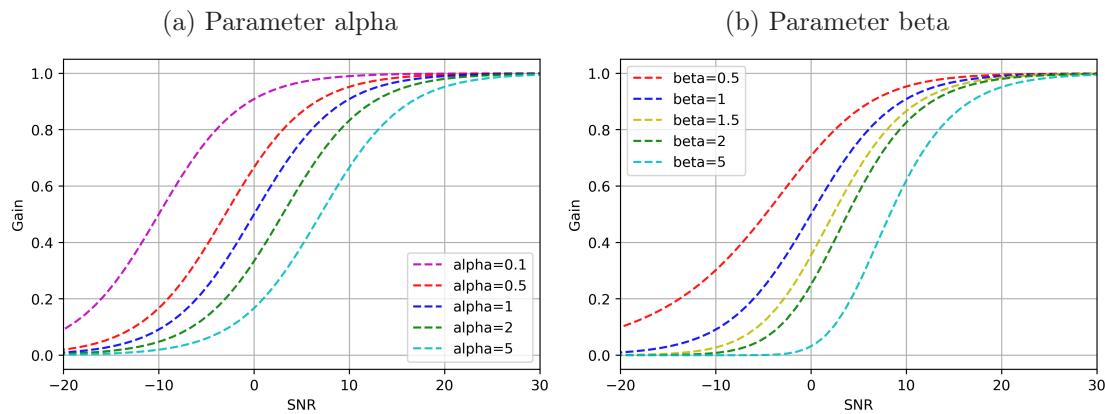


Abbildung 9.2: Gain Kurven des parametrisierten Wiener Filter

Die Formel für den parametrisierten Wiener-Filter wird gegeben durch:

$$Gain(t, f) = \left( \frac{\hat{SNR}_{prio}(t, f)}{\alpha + \hat{SNR}_{prio}(t, f)} \right)^\beta \quad (9.3)$$

Bislang bewies das im Rahmen dieser Arbeit entwickelte Subband D-CNN-LSTM Funktionalität bei der Berechnung der IRM. Daher wurde dieses Netz für die Aufgabe der a-priori SNR Schätzung evaluiert. Hierfür wurde ein parametrisierter Wiener Filter mit  $\alpha = 0.2$  und  $\beta = 0.2$  genutzt. Die Parameter wurden durch Evaluierung mit den objektiven Metriken einer 1,5 Minuten langen Sequenz verschiedener Störungen ermittelt.

In Versuchen mit parametrisiertem und unparametrisiertem Wiener Filter zeigen sich keine Verbesserungen der Evaluierungsmetriken. Daher wird an dieser Stelle auf die Grafiken der Ergebnisse verzichtet. Das Netz vermag nicht, nach Sichtung der inferierten Zielmaske, eine hinreichende Abbildungsfunktion zu finden. Für weitere Versuche muss die Architektur angepasst werden.

# 10 Post-Processing

Abbildung 2.1 zeigt ein klassisches Sprachverbesserungssystem mit spektraler Subtraktion. Teil dieses Systems ist eine Nachbearbeitung (engl. *post processing*) des Signals, bevor dieses zurück in den Zeitbereich gewandelt wird. In einer Nachbearbeitung können bspw. Störungen, die durch systeminherente Funktionalität entstehen, gemindert werden. Auch für Sprachverbesserungssysteme auf Basis von neuronalen Netzen wurden Nachbearbeitungsmöglichkeiten vorgestellt. Möglich sind bspw. ein Nachtrainieren des Netzes mit angepasster Fehlerfunktion, oder eine direkte Anpassung der Ausgangsdaten anhand statistischer Verfahren.

## 10.1 Global Variance Equalization

Die *Global Variance Equalization* (dt. globaler Varianzausgleich) Methode wurde 2014 von Xu et. al vorgestellt [45]. In ihrer Veröffentlichung berichten die Autoren von einer Überglättung (engl. *oversmoothing*) des Histogramms der inferierten Features in regressionsbasierten Netzen. Um diese Überglättung zu mindern, sollen die Varianzen der inferierten Features dementsprechend angepasst werden, so dass die auftretenden Features näher zu ihrer wahren Verteilungsdichte rücken.

Im Weiteren soll überprüft werden, ob sich die beschriebene Problematik in den vorgestellten Architekturen dieser Arbeit zeigt. Außerdem wird das Verfahren angewendet, um anhand der Evaluierungsmaßnahmen die Performanz der Nachbearbeitung zu bewerten.

Die globale Varianz (**GV**) der Feature Vektoren wird in Formel 10.1 definiert.

$$GV(f) = \frac{1}{M} \sum_{t=1}^M \left( \hat{X}_t^f - \frac{1}{M} \sum_{t=1}^M \hat{X}_t^f \right)^2 \quad (10.1)$$

$\hat{X}_t^f$  steht für die Feature Komponente f, zum Zeitpunkt t, bei einer Matrix bestehend aus M Zeit-Frames.

Ebenfalls wird die dimensionslose Formel für die globale Varianz in Formel 10.2 vorgestellt.

$$GV = \frac{1}{M \cdot D} \sum_{t=1}^M \sum_{f=1}^D \left( \hat{X}_t^f - \frac{1}{M \cdot D} \sum_{t=1}^M \sum_{d=1}^D \hat{X}_t^f \right)^2 \quad (10.2)$$

### 10.1.1 Untersuchung der Varianz

Um das beschriebene Problem der Überglättung zu untersuchen, wurden zunächst Histogramme der Gain-Parameter betrachtet.

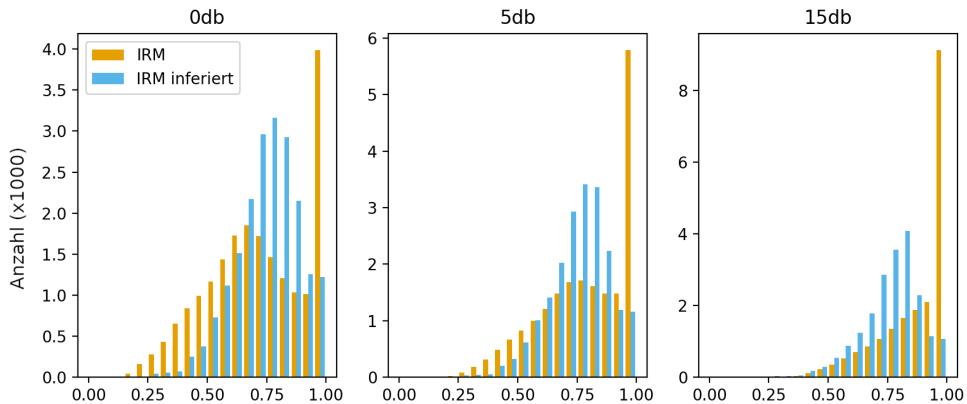


Abbildung 10.1: Gain-Histogramm, Störung: Sirene

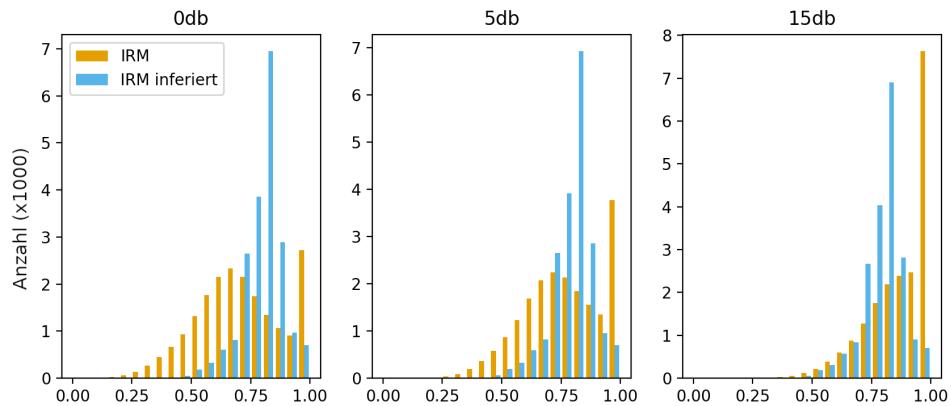


Abbildung 10.2: Gain-Histogramm, Störung: Straße

Abbildung 10.1 und 10.2 zeigen hier die ground truth Gain-Parameter, sowie die Parameter nach Inferenz mit dem vorgestellten Subband D-CNN-LSTM. Es wurden drei verschiedene SNR betrachtet. Ein Blick auf die Abbildung mit 15 dB SNR bei beiden Störungen macht deutlich, dass es vor allem eine besondere Glättung des Gain Parameters gibt, wenn dieser im Bereich von 1 in der ground truth liegt. In vielen Bereichen kommt es zu einer zu großen Dämpfung einzelner TF-Slots. TF-Slots ohne Störung werden vom Netz als solche mit Störung betrachtet. Allgemein

zeigt sich, dass die Verteilungsfunktion der inferierten IRM im Mittelwert verschoben ist. Zudem hat sich die Form der Verteilung geändert, wie in Abbildung 10.2 zu erkennen, mit einer Stauchung bzw. verringelter Varianz.

Im Folgenden sind die Ergebnisse aus der Anwendung der Formeln 10.1 und 10.2 gezeigt.

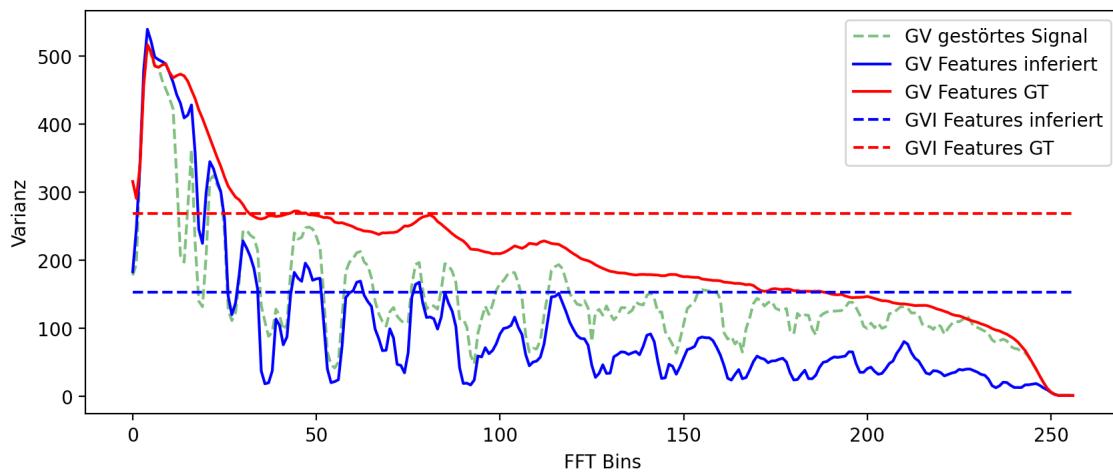


Abbildung 10.3: Globale Varianz mit Störung „Sirene“ bei 10 dB SNR

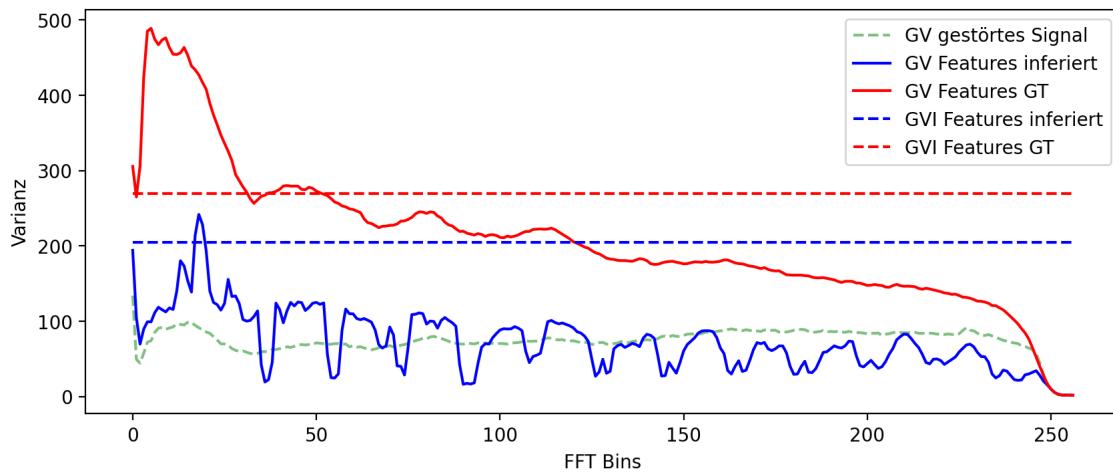


Abbildung 10.4: Globale Varianz mit Störung „Straße“ bei 10 dB SNR

Gezeigt wird die GV für das gestörte Signal, sowie für das Signal ohne Störung und das Signal nach Inferenz durch das KNN. Außerdem wird die dimensionslose GV für

das ungestörte und inferierte Signal dargestellt. Zunächst fällt hier die hohe Varianz der STFT Features im Bereich niedriger Frequenzen auf. Dies ist auf die Verteilung der Energie in Sprache zurückzuführen, wie bereits in Abbildung 5.6 gesehen wurde. Betrachtet man die Funktion des gestörten Signals, welche in grün dargestellt ist, wird deutlich, dass sich die Varianz im Vergleich zum ungestörten Signal über nahezu alle Frequenzen gemindert hat. Durch das Einbringen der Störung mindern sich die Abstände der STFT Amplituden und damit die Varianz.

Die in blau dargestellte Kurve beschreibt die GV nach Inferenz durch das KNN. Im Idealfall müsste sie mit der roten Kurve identisch sein. Die beiden gestrichelten Linien sind das Ergebnis aus Formel 10.2 als dimensionsunabhängige GV.

### 10.1.2 Anwendung und Ergebnisse

In [45] werden verschiedene Methoden vorgestellt, um die Varianz des inferierten Signals anzuheben. Hierfür kann der Parameter  $\beta$  berechnet werden, der das Verhältnis der Varianzen darstellt. Mit dimensionsunabhängiger GV berechnet sich dieser durch:

$$\beta = \sqrt{\frac{GV_{ref}}{GV_{est}}} \quad (10.3)$$

Der dimensionsabhängige Parameter kann berechnet werden durch:

$$\alpha(d) = \sqrt{\frac{GV_{ref}(d)}{GV_{est}(d)}} \quad (10.4)$$

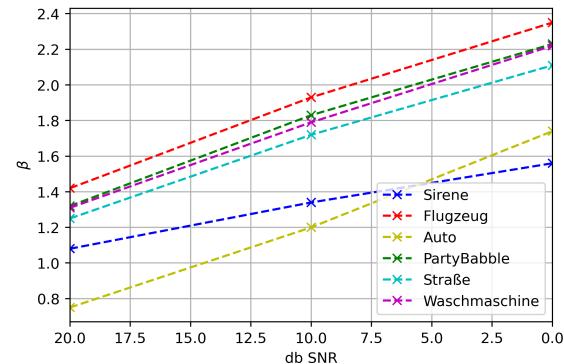


Abbildung 10.5:  $\beta$  für verschiedene Störungen

Laut den Autoren ist der dimensionsunabhängige Parameter zu bevorzugen, da dieser größere Verbesserungen der Evaluierungsmaßen zeigt, welches auf die Instabilität des Ausgleichsfaktors über die Frequenz Bins zurückgeführt wird.

Aus diesem Grund sind die Ergebnisse der Berechnung des Parameters  $\beta$  anhand verschiedener Störungen bei unterschiedlichen SNR mit dem dimensionsunabhängigen Parameter untersucht. In Abbildung 10.5 wurden die Ergebnisse aus Formel 10.3 aufgetragen. Hier kann gesehen werden, dass die Steigungen von fünf der sechs Störungen ähnlich sind, sowie dass die Störungen „Flugzeug“, „Straße“, „Waschmaschine“ und „Partybabble“ eng beieinander liegen. Der Anstieg des Varianzunterschieds nimmt also linear mit abnehmendem SNR zu.

Um die Varianz zu erhöhen, wurde eine Formel für Systeme basierend auf Maskierungsmethoden aus [53] direkt auf die Gain-Parameter angewendet:

$$\hat{Gain}_{GVE}(t, f) = (\hat{Gain}(t, f) - m_{GT}) \cdot \sqrt{\frac{GV_{ref}}{GV_{est}}} + m_{GT} \quad (10.5)$$

$m_{GT}$  steht für den Mittelwert der Gain-Werte aus der ground truth Maske. In Tabelle 10.1 sind die Ergebnisse der Evaluierungsmaßen PESQ und STOI mit der Anwendung der Global Variance Equalization als Post Processing Methode für das vorgestellte Subband D-CNN-LSTM dargestellt.

	SNR	gestört	inferiert	inferiert+GVE	STOI ±	$\beta$
Sirene	20	1.89	3.26	<b>3.34</b>	0.0	1.08
	10	1.28	2.14	<b>2.42</b>	<b>0.09</b>	1.34
	0	1.15	1.35	<b>1.51</b>	<b>0.07</b>	1.56
Flugzeug	20	2.11	<b>2.6</b>	<b>2.6</b>	-0.01	1.42
	10	1.31	1.52	<b>1.53</b>	-0.02	1.93
	0	1.09	<b>1.11</b>	1.1	-0.01	2.35
Auto	20	2.7	2.6	<b>2.65</b>	0.01	0.75
	10	1.66	<b>1.68</b>	1.66	-0.01	1.2
	0	1.19	<b>1.2</b>	1.18	-0.02	1.74
Partybabble	20	2.09	2.51	<b>2.52</b>	0.0	1.32
	10	1.39	1.62	<b>1.64</b>	-0.02	1.83
	0	1.18	<b>1.24</b>	1.21	-0.01	2.23
Straße	20	2.25	<b>2.58</b>	2.56	-0.01	1.25
	10	1.44	<b>1.69</b>	1.65	-0.02	1.72
	0	1.11	<b>1.15</b>	1.13	-0.02	2.11
Waschmaschine	20	2.12	<b>2.49</b>	2.48	-0.01	1.31
	10	1.41	<b>1.65</b>	1.63	-0.02	1.79
	0	1.1	<b>1.14</b>	1.12	-0.01	2.22

Tabelle 10.1: Evaluierungsmaßen für Subband D-CNN-LSTM mit GVE

Bei Störung mit Sirene bzw. Martinshorn wird die größte Verbesserung der Metriken bei 10 db SNR erreicht. Hier kann die PESQ Metrik um 0.28 Punkte zusätzlich erhöht werden. Die STOI Metrik wird gleichzeitig um weitere 0.09 Punkte erhöht. Für andere Störtypen zeigte sich entweder keine oder eine geringfügige Verbesserung der Metriken.

### 10.1.3 Diskussion

Wie aus den Ergebnissen zu entnehmen ist, unterscheiden sich die erreichten Metriken stark bei der Betrachtung einzelner Störtypen. Im Rahmen dieser Arbeit wurde die GVE als Nachbearbeitungsmethode angewendet. In [45] wird zusätzlich das Netz mit einer an die GVE angepassten Fehlerfunktion nachtrainiert. Diese erreicht eine weitere Verbesserung des PESQ Wertes um 0.01 bis 0.02 Punkte, welche als eher geringe Verbesserung gewertet werden können. Über eine GVE mit dem dimensionsunabhängigen Parameter  $\beta$  wird in der angesprochenen Veröffentlichung eine Verbesserung von 0.12 PESQ Punkten bei 20 db SNR erreicht. Dies wurde für eine Evaluierung auf vier Störtypen vermeldet. In den hier durchgeföhrten Untersuchungen zeigten sich die Verbesserungen allgemein als eher marginal, mit Ausnahme der Störung durch die Sirene. Daher kann vermutet werden, dass die Charakteristik der Störung darüber entscheidet, inwiefern eine Nachbearbeitung mit der GVE als sinnvoll zu betrachten ist, da es hierdurch möglicherweise zudem zu minimalen Verschlechterungen oder einem Gleichbleiben der STOI Metrik kommen kann. Diese Ergebnisse sind in etwaiger Übereinkunft mit denen aus [53]. In dieser Publikation wird eine iteratives Verfahren zur GVE angewendet, welches die STOI Metrik um weitere 0.01 bis 0.02 Punkte verbessern soll. Im Anwendungsfall eines vollständigen Sprachverbesserungssystems sollte daher zunächst betrachtet werden, in welcher Relation die beschriebene Überglättung durch das neuronale Netz vorhanden ist, um dann je nach Störtyp diese Nachbearbeitungsmethode bedarfsweise einzusetzen. Abschließend werden Spektrogramme zweier inferierter Äußerungen mit und ohne GVE in Abbildung 10.6 gezeigt. Wie aus der Bildverarbeitung bekannt, bildet die GVE eine Form des Histogrammausgleichs und erhöht den Kontrast des Bildes. Im Falle des STFT Spektrogramms ist dies an den leicht vergrößerten Schwarzbereichen zu erkennen.

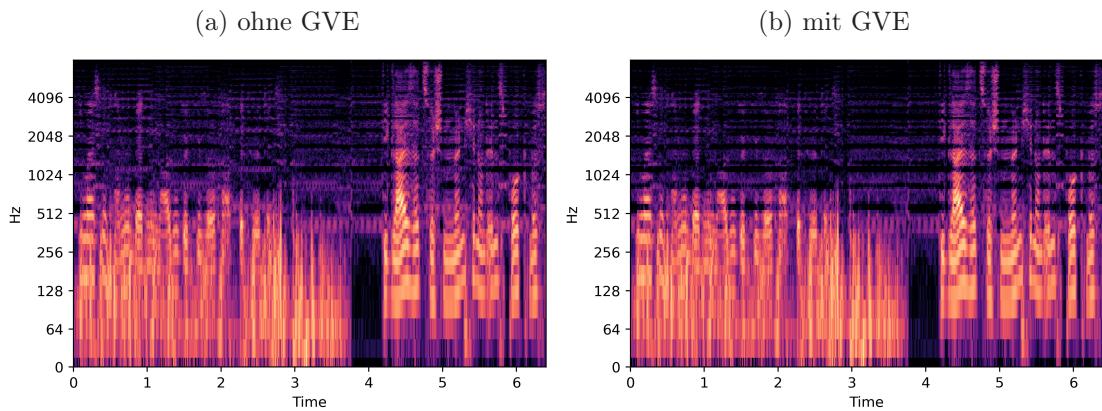


Abbildung 10.6: Spektrogramme mit und ohne GVE

## 10.2 Post-Gain

Wie bereits bekannt, erfolgt die Schätzung der IRM als Gain-Funktion auf dem Intervall  $[0;1]$ . Eine einfache Möglichkeit zur Abstimmung zwischen dem Maß an Dämpfung und der Qualität der Sprache besteht in der Einführung eines zusätzlichen Parameters  $\gamma$ . Die IRM wurde angepasst zu:

$$Gain(t, f) = (IRM_{KNN}(t, f))^\gamma \quad (10.6)$$

Um zu Betrachten, inwieweit ein nachträgliches Anpassen des Parameters  $\gamma$  die objektiven Evaluierungsmaßen ändern kann, wurde beispielhaft für die Störung im Autoinnenraum, mit der Architektur aus Sektion 8.1, eine Untersuchung durchgeführt. Die genutzten Werte des Parameters sind im Diagramm zu sehen.

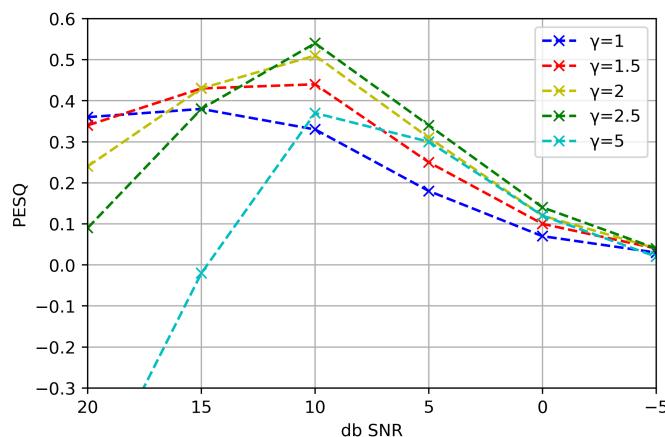


Abbildung 10.7: Post-Gain PESQ Änderung, Störung Auto

Wie zu sehen wird der beste Wert, eine Steigerung der PESQ Metrik um ca. 0.55, bei einem SNR von 10 dB und  $\gamma = 2.5$  erreicht. Jedoch fällt die dazugehörige Kurve bei höherem SNR stark ab, sodass sie hier ein schlechteres Ergebnis liefert, als die Ausgangskurve mit  $\gamma = 1$ . Für ein bestmögliches Ergebnis der PESQ Metrik sollte in diesem Fall  $\gamma = 1.5$  gewählt werden, da sich für diesen Wert eine kontinuierliche Steigerung über alle SNR gegebenüber dem Referenzwert ergibt.

Bei einer Anpassung des Wertes auf  $\gamma = 1.5$  ergibt sich eine minimale Verschlechterung der STOI Metrik, welche ca. 0.01 unter dem Referenzwertes liegt. Weitere Ergebnisse hierzu werden in Kapitel 11 vorgestellt.

# 11 Auswertung und Vergleich

Nachfolgend wird ein Blick auf die zusammengefassten Ergebnisse dieser Arbeit geworfen, um diese in Relation zu den Ergebnissen aus state-of-the-art Systemen zu bewerten. Hierfür dient die Tabelle 11.1. Diese zeigt Ergebnisse der objektiven Evaluierungsmetriken, welche in Sektion 2.3.2 vorgestellt wurden. Beispielhaft wird die Tabelle für einen Signal-Rauschabstand von 15 dB gezeigt. Dieser Wert wurde gewählt, weil hier die allgemeinen Unterschiede zwischen den Netzen am besten in Erscheinung treten. Die in Tabelle gezeigten Werte entstehen durch eine Evaluierung mit 30 Äußerungen, die eine Länge von ca. 6 Sekunden aufweisen. Die höchsten erreichten Werte sind in Fettschrift markiert.

Die in der Tabelle 11.1 oben aufgeführten Werte für die Signale mit Störung dienen als Referenz. Betrachtet man hier die Werte der PESQ Metrik, so wird deutlich, dass verschiedene Störtypen unterschiedliche Auswirkungen auf den erreichten Wert haben, auch wenn die Überlagerung beim gleichen SNR stattgefunden hat. Auffällig ist, dass für fünf der sechs Störungen der höchste Wert, der die Verständlichkeit der Sprache beschreiben soll (STOI), beim gestörten Signal liegt. Die Problematik der Verständlichkeitsreduktion durch MSE Fehlerfunktionen wurde bereits in 8 angeprochen.

Für drei Störtypen konnten Verbesserungen der Signal-to-Distortion Ratio erreicht werden.

Zuerst wird das vorgestellte Residual CNN, welches mit Subdatensatzstrategie trainiert wurde, betrachtet. Dieses erreicht für die Störung durch Sirene in allen drei Kategorien die größten Verbesserungen. Zurückzuführen ist dies auf die konsequente Nutzung von CNN, welche über die Gesamtheit des STFT Kontextfensters angewendet wurde. Für andere Störungen liegen die Werte hinter denen der anderen vorgestellten Architekturen zurück. Die beste Performanz allgemein gesehen lieferte das im Rahmen der Arbeit entwickelte Subband D-CNN-LSTM. Es werden für drei der sechs Störungen die höchsten Werte erzielt, ohne dass es zu großen Einbußen in der Verständlichkeit der Sprache kommt. Für die Sirenenstörung wird der gleiche STOI Wert wie beim Residual CNN erreicht.

gestörtes Signal		PESQ	STOI	SDR
Sirene	1.57	0.84	6.39	
Auto	2.19	<b>0.96</b>	<b>29.2</b>	
Flugzeug	1.65	<b>0.91</b>	<b>6.41</b>	
PartyBabble	1.68	<b>0.87</b>	<b>7.14</b>	
Straße	1.86	<b>0.91</b>	2.36	
Waschmaschine	1.73	<b>0.88</b>	4.58	
<hr/>				
Residual CNN				
Sirene	<b>2.86</b>	<b>0.9</b>	<b>7.27</b>	
Auto	2.12	0.92	8.74	
Flugzeug	1.79	0.87	5.97	
PartyBabble	1.94	0.84	6.89	
Straße	1.99	0.88	7.15	
Waschmaschine	1.97	0.86	6.22	
<hr/>				
Subband D-CNN-LSTM				
Sirene	2.72	<b>0.9</b>	5.29	
Auto	2.2	0.94	3.04	
Flugzeug	2.03	0.89	2.77	
PartyBabble	<b>2.06</b>	0.86	3.73	
Straße	<b>2.13</b>	0.89	4.35	
Waschmaschine	<b>2.08</b>	0.87	3.78	
<hr/>				
DenseNet-121				
Sirene	2.07	0.82	5.61	
Auto	2.21	0.9	9.91	
Flugzeug	<b>2.07</b>	0.86	6.37	
PartyBabble	2.02	0.83	6.53	
Straße	2.01	0.86	<b>7.37</b>	
Waschmaschine	1.99	0.84	<b>6.75</b>	
<hr/>				
PMSQE-Loss SB-D-CNN-LSTM				
Sirene	1.88	0.85	6.97	
Auto	<b>2.57</b>	0.95	13.87	
Flugzeug	1.79	0.88	5.95	
PartyBabble	1.74	0.86	6.58	
Straße	2.04	0.88	6.08	
Waschmaschine	1.8	0.84	6.34	
<hr/>				
DeepXi - MMSE-LSA				
Sirene	2.02	0.87		
Auto	2.29	0.93		
Flugzeug	1.98	0.89		
PartyBabble	2.05	<b>0.88</b>		
Straße	1.94	0.88		
Waschmaschine	1.86	0.86		

Tabelle 11.1: Auswertung der vorgestellten Netze bei 15 dB SNR

Für die Implementierung des DenseNet-121 werden leichte Verbesserungen im Vergleich zum Residual CNN gesehen, jedoch kommt es zu stärkeren Einbußen in der Verständlichkeit, als in anderen Netzen. Einzig für die Störung des Flugzeugs zeigt sich hier der beste PESQ Wert. Für Störungen durch eine Waschmaschine und durch Straßenlärm, erreichen die SDR die höchsten Werte. Dies muss bedeuten, dass wenig zusätzliche Störungen durch das Netz hinzugefügt wurden, jedoch sind diese aufgrund der schlechteren Werte für PESQ und STOI weniger von Bedeutung.

Die vorgestellte Subband D-CNN-LSTM Architektur wurde mit zusätzlicher PMS-QE Fehlerfunktion trainiert. Auffällig ist hier, dass die Störung im Autoinnenraum ihre größte Verbesserung erfährt. Diese konnte in anderen Netzen nicht, oder nur marginal verbessert werden. Der SDR Wert verdeutlicht, dass an dieser Stelle besonders wenig zusätzliche Störungen durch das Netz erzeugt wurden.

Zuletzt wurde eine Auswertung mit dem DeepXi Framework vorgenommen. Leider konnte dieses, zu seinem Zeitpunkt der Kompatibilität mit der hier genutzten TensorFlow Version, nicht mit den gewählten Störungen nachtrainiert werden. Jedoch ist das Netz bereits auf die gleichen Störarten trainiert. Es zeigen sich für alle Störtypen leichte Verbesserung der PESQ Metrik, sowie für die Störung durch BabbleNoise eine geringe Verbesserung des STOI Wertes. An dieser Stelle stellt sich die Frage nach der Leistungsfähigkeit der gewählten Evaluierungsmetriken. Eine subjektive Begutachtung der Ergebnisse aus der DeepXi Inferenz zeigte eine solide Störunterdrückung bei gleichzeitig guter Qualität der Sprache, welche sich möglicherweise nicht vollständig in den Ergebnissen der Evaluierungsmetriken widerspiegeln. Wie bereits in Kapitel 2.3.3 beschrieben wurde, werden auch andere Evaluierungsmetriken herangezogen.

Ein selbiges Phänomen lässt sich nach subjektiver Testung von SEGAN Ergebnissen vermuten, in deren Publikation nur von geringen PESQ Verbesserungen berichtet wird. Hier werden auch weitere Metriken angewendet, die möglicherweise besser die Entstörleistung bewerten. Im EHNet wird ebenfalls die PESQ Metrik herangezogen, daher werden in Tabelle 11.2 die durchschnittlichen Verbesserungen durch das System gezeigt. Hier wird ebenfalls der zum jetzigen Zeitpunkt aktuellste Wert der maximalen Verbesserung im DeepXi Framework gezeigt.

		gestört	inferiert
PESQ	EHNet	2.26	2.86
	SEGAN	1.97	2.16
	DeepXi	1.97	3.03

Tabelle 11.2: PESQ SOTA

	Sirene	Auto	Straße	PartyBabble	Flugzeug	Waschmaschine
<b>PESQ</b>						
<i>gestört</i>	1.57	2.2	1.86	1.68	1.65	1.73
( $\gamma = 1.5$ ) PG	<b>3.0</b>	2.15	<b>2.15</b>	<b>2.16</b>	<b>2.1</b>	<b>2.12</b>
PG+GVE	2.97	<b>2.21</b>	2.13	2.15	2.1	2.11
<b>STOI ±</b>						
PG	-0.35	<b>0.04</b>	<b>0.03</b>	0.00	-0.01	-0.01
PG+GVE	-0.35	<b>0.06</b>	0.02	-0.01	-0.02	0.00

Tabelle 11.3: SB-D-CNN-LSTM Ergebnisse mit Post-Processing, 15 db SNR

In Tabelle 11.3 werden abschließend Ergebnisse aus der Anwendung der in Kapitel 10 vorgestellten Nachbearbeitungsmethoden gezeigt. Hier steht PG für die Anwendung des Post-Gains, sowie PG+GVE für die Anwendung der Global Variance Equalization auf die veränderten Post-Gain-Parameter mit  $\gamma = 1.5$ . Wie zu erkennen, können durch das Einbringen dieses Parameters bei allen PESQ Werten weitere Verbesserungen erzielt werden, wenn man diese mit den Ergebnissen aus Tabelle 11.1 vergleicht.

Die größte Verbesserung entsteht erneut bei der Störung mit Sirene, welches mit der geringen spektralen Überschneidung der Sprach- zu Störenergien, sowie der klaren Struktur der Störung begründet werden kann. Hier können zusätzliche 0.28 Punkte erreicht werden, jedoch nimmt die Verständlichkeit deutlich ab. Dies kann möglicherweise durch eine nicht optimale Wahl des Parameters  $\gamma$  begründet werden.

Alle weiteren Störungen können bei diesem SNR ebenfalls verbessert werden, wobei die Störung im Auto um weitere 0.07 Punkte durch die Anwendung der GVE profitiert. Gleichzeitig wird hier der STOI Wert um 0.02 Punkte verbessert, nachdem dieser durch den PG um 0.04 Punkte erhöht wurde. Die STOI Metrik erfährt mit Anwendung der GVE für andere Störungen nur leichte Einbußen. Die Einbußen durch den PG können mit der GVE, wie bei der Störung durch eine Waschmaschine, wieder ausgeglichen werden. Eine Anwendung der Nachbearbeitungsmethoden kann daher als Möglichkeit des Trade-offs zwischen Qualität und Verständlichkeit gesehen werden.

## 12 Zusammenfassung und Ausblick

Das Ziel dieser Arbeit bestand in der Untersuchung von Möglichkeiten zur Störunterdrückung mit Deep Learning Techniken in monauralen Sprachsignalen. Hierfür wurden zunächst die historisch wichtigsten Grundlagen aus der Störunterdrückung mit klassischen statistischen Methoden erläutert. Diese wurden im Weiteren aufgegriffen um aufzuzeigen, wie die Forschungsergebnisse aus den letzten Dekaden mit Deep Learning Architekturen verknüpft werden können.

Historisch wurden hier Algorithmen, welche auf spektraler Subtraktion basieren, genutzt. Die Algorithmen von Eprahim und Malah, bei denen Gain-Masken für TF-Slots aus einem autoregressiven Prozess geschätzt werden, erfuhren ebenfalls große Beliebtheit. Die Konzepte des a-priori und a-posteriori SNR, die hierfür eingesetzt werden, sind auch noch als Grundlage für maskierungsbasierte Netze aktuell, indem die Schätzung der SNR durch neuronale Netze geleistet- und die entsprechende Dämpfung aus erprobten Methoden, wie dem Wiener Filter und ähnlichen, gebildet wird.

Im Weiteren wurden die Grundlagen neuronaler Netze erörtert, indem deren Aufbau und Funktionsweise beschrieben wurde. Hierzu zählen die Fehlerrückführung und Optimierungsalgorithmen. Entsprechend wurden Deep Learning Techniken vorgestellt, die im Bereich der Sprachverbesserung von besonderer Bedeutung sind. Hierzu zählen die rekurrenten Netze, wie z.B. des LSTM, welches eine Referenz aktueller Daten zu Daten aus vorherig gesehenen Zeitpunkten bilden kann. Convolutional Neural Networks rücken ebenfalls mehr in den Fokus für die Verarbeitung von Audio Daten. Hier kann mit der Technik der Dilatation das rezeptive Feld der Faltung vergrößert werden, um längerfristige Abhängigkeiten zu erkennen. Außerdem werden ResNets angesprochen, die in der Literatur häufig Verwendung aufgrund ihrer Fähigkeit zur Verhinderung von verschwindenden Gradienten bei der Fehlerrückführung finden.

Für die Rauschunterdrückung mit neuronalen Netzen wurden die Grundlagen häufig eingesetzter Methoden beschrieben. Die Ideal Binary Mask(IBM) bietet hier als Trainingsziel die Möglichkeit zur binären Maskierung für TF Slots, welche dem Kriterium eines bspw. positiven SNR entsprechen. Diese werden aktuell in Separierungsaufgaben, wie dem Abtrennen eines bestimmten Sprechers eingesetzt, oder als Front-End in automatischen Spracherkennungssystemen. Bei einer binären Maskierung leidet

die Qualität der Sprache, so dass die Anwendung dieser allein, für Ergebnisse, die von einem Menschen wahrgenommen werden sollen, nicht zufriedenstellend ist.

Die Ideal Ratio Mask (IRM) versucht dieses Problem zu lösen, in dem die Relation der Energie des ungestörten Signals, zum Signal mit aufaddierter Störung gebildet und als Trainingsziel formuliert wird.

Im Anschluss wurden objektive Evaluierungsmaßnahmen vorgestellt, welche in der Literatur häufig Anwendung finden. Hierbei soll die PESQ Metrik mit der von Menschen empfundenen Qualität der Sprache korrelieren. Die STOI Metrik bildet einen Wert, der die Verständlichkeit von Sprache widerspiegeln soll. Außerdem wurde die Source-to-Distortion Ratio vorgestellt, welche das ungestörte Signal in Relation zu absoluten verbleibenden Störungen setzt. Dies kann als Indikator für die Entstörleistung, sowie induzierte Störungen durch Systeme herangezogen werden.

In Sektion „Stand der Technik“ wurden für die Forschung aktuell wichtige Architekturen und Frameworks vorgestellt.

Das EHNet bildet hierbei eine Ratio Mask im Frequenzbereich, vom gestörten zum verbesserten Signal, mit einem Kontextfenster.

Das DeepXi Framework versucht eine a-priori SNR Schätzung mit Hilfe des KNN zu leisten. Auf dieser Grundlage können bereits ausgiebig erprobte Verfahren aus dem Bereich der Sprachverbesserung, wie etwa der MMSE-LSA Gain oder ein Wiener Filter, als Maskierungsmethode angewendet werden. Außerdem werden generative Methoden, wie das SEGAN und WAVENET vorgestellt. Zudem wird ein kurzer Einblick in die aus der Industrie bekannten Informationen und Projekte gegeben, welche versuchen, ein vollständiges Sprachverbesserungssystem für eine Vermarktung mit Gewinnerzielungsabsicht zu liefern.

Folgend wurden Methoden zur Datenvorverarbeitung, sowie die Erzeugung des Datensatzes, für eigene Versuche, erläutert. Für alle Versuche wurde ein Datensatz aus der Überlagerung von 1500 Äußerungen, mit Länge einiger Sekunden, mit sechs verschiedenen Störungen gebildet. Diese befinden sich bei verschiedenen Störabständen von 20 dB bis -5 dB. Für die STFT wurde dabei eine Anzahl von 512 Stützstellen mit Hann Fenster verwendet, welche zur Bildung des einseitigen Frequenzspektrums dienten. Die logarithmierten Amplituden wurden dabei als Eingangsdaten für die Versuche genutzt. Hierfür wurde ein Kontextfenster aus dem zeitlichen Verlauf der Daten gebildet, welche der Strukturerkennung des KNN dienlich sein soll. Für weitere Versuche sollte an dieser Stelle die Frequenzauflösung variiert werden. In der Literatur finden sich hier Ansätze mit multiplen Auflösungen, welche parallel als Eingang dienen. Eine grobe Auflösung könnte hier beim Erkennen von Strukturen nützlich sein, eine feinere Auflösung vermag möglicherweise besser die Charakteristiken von Störungen mit granularer Schwankung der Amplituden erkennbar zu machen. Ein weiterer Ansatz, der als Erweiterung bzw. Abstraktion der hier vorgestellten Netze dienen könnte, ist das Einbringen einer Gammaton-Filterbank, welche zunächst im Zeitbereich angewendet wird, um dessen einzelne Frequenzbänder se-

perat in den Frequenzbereich zu wandeln. Hier könnte von einer größeren Auflösung profitiert werden, sowie der latenten Überlagerung von möglichen Features aus der Struktur der Sprache, welche durch die Überlagerung der Trennfrequenzen des Gammaton Filters entstehen. Möglicherweise könnten dadurch die Eingangsdaten weiter abstrahiert werden, so dass diese als dreidimensionale Struktur in Form von: [STFT Fenstergröße X N-Gammaton Filterkanäle X T zeitliche Frames] eines Kontextfenseters erscheinen, für welche ebenfalls Convolution, Pooling und weitere Operatoren bestehen, die aus den Zusammenhängen eines höherdimensionalen Raumes bei der Erkennung von Features profitieren könnten.

Als erste Implementierung mit neuronalem Netz wurde versucht eine Ideal Binary Mask aus den STFT Features zu erzeugen. Dieses Netz wurde dabei als Struktur aus einer Publikation entnommen. Es zeigte sich, dass die Schätzung eines vollständigen STFT-Frames der Länge 257 besonders schwierig ist. Hier waren die gewählten Strukturen nicht in der Lage das Netz zur Konvergenz zu führen. Wie auch häufig in der Literatur, wurde versucht die Abbildung über eine Darstellung mit Subbändern durchzuführen. Hierfür wurde eine Einteilung des STFT Frames vorgenommen, welche mit den Trennfrequenzen der Bark-Skala korrespondieren. Dies geschah aufgrund der Tatsache, dass die Verarbeitung von Audiosignalen im menschlichen Ohr ebenfalls im Bereich einzelner Bark-Bänder stattfindet. Hier zeigte sich jedoch ebenfalls keine Konvergenz, welche auf eine zu grobe Unterteilung zurückgeführt wurde. Eine Unterteilung der Bark-Bänder in jeweils drei weitere symmetrische Unterbänder konnte das Netz zur Konvergenz führen. Eine weitere Schwierigkeit bestand in dem ungleich verteilten Datensatz der Binärmaske, welcher zu ca. 75 % aus Nullen, sowie 25 % aus Einsen bestand, wodurch die Notwendigkeit einer angepassten Fehlerfunktion des Optimierers bestand, der mit unterschiedlicher Gewichtung die korrekte bzw. falsche Klassifikation bewertet.

Die Auswertung der Ergebnisse zeigte hier, dass ca. 20 % bis 80 % der Klassen mit Wert 1 erkannt werden können, jedoch mit einer Beeinträchtigung durch steigenden falsch-positiv Anteil. An dieser Stelle soll erwähnt sein, dass sich, wie aus der Literatur bekannt, weitere Features aus einem Sprachsignal bilden lassen, welche sich zuträglich für eine Bildung der Binärmaske zeigen. Aus der Forschung existieren hier eine Vielzahl von Transformationen, wie etwa die Mel-Frequenz Cepstrum-Koeffizienten oder die Relative Spectral Transform - Perceptual Linear Prediction (RASTA-PLP), welche sprecherunabhängige Informationen generieren soll. Für weitere Versuche und eine Verbesserung der Ergebnisse sollten daher weitere spektrale und temporale Features als zusätzliche Eingangsdaten verwendet werden. Die IBM kann, sofern eine ausreichende Klassifizierungsgenauigkeit erreicht werden kann, als Post-Processing Methode angewendet werden, in dem z.B. TF-Slots anhand ihrer Wahrscheinlichkeit, keine wichtigen Informationen zu tragen, weiter gedämpft wer-

den. Die Ideal Ratio Mask ist, im Gegensatz zur Ideal Binary Mask, als das Mittel der Wahl, wenn die Ergebnisse von Menschen wahrgenommen werden, zu betrachten. Anfängliche Versuche wurden hier mit Fully-Connected Netzen durchgeführt, welche jedoch keine Konvergenz zeigten. Dieses Problem wird teilweise in der Literatur aufgegriffen und durch den Einsatz von Pre-Training mit Boltzmann Maschinen gemindert, indem hierdurch die Komplexität des Trainingsvorgangs reduziert wird. Im Rahmen dieser Arbeit wurde jedoch ein Ansatz über andersartige Strukturen gewählt. Aufgrund des häufigen Einsatzes von Residual-Verbindungen, in Sprachverbesserungssystemen der Literatur, wurde ein Ansatz eines typischen Residual-Netzes abstrahiert und für eine Verarbeitung der beschriebenen STFT-Daten aus einem Kontextfenster optimiert.

In diesem Zuge stellte sich die Frage der Trainingsstrategie. Der gewählte Datensatz ist aufgrund der Größe der Daten nicht in der Lage vollständig in den VRAM der Grafikkarte geladen zu werden. Für zu große Datensätze wird daher häufig eine Generatorfunktion herangezogen, welche einen Teil des Datensatzes in einer gewissen Anzahl von Batches zum Optimierer liefert. Es ergeben sich jedoch Unterschiede im Trainingsprozess und dem Update der Gradienten in Abhängigkeit der gewählten Hyperparameter. So können mit erhöhter Batchgröße mehr Daten gleichzeitig verarbeitet werden, dies kann sich jedoch auch negativ auf die Konvergenz der Gewichte ausüben.

Die STFT Matrizen von Sprachaufnahmen müssen als dünn besetzt angesehen werden, welches sich auch, wie bereits angesprochen, in der Klassenverteilung der IBM, widerspiegelt. Zusätzlich können sich große Änderungen der Daten, bspw. durch Änderung des Sprechers oder durch unterschiedliche Äußerung ergeben, auch wenn diese mit der selben Störung überlagert sind. Aufgrund dieser Tatsache wurde im Rahmen dieser Arbeit eine neue Trainingsstrategie entwickelt. Diese basiert auf der Unterteilung des Datensatzes in Subdatensätze, welche aus jeweils 200 Kontextfenstern bestanden. Es wurde für jeweils einen Subdatensatz ein Training über 10 Epochen durchgeführt. Hierfür wurde eine sich stufenweise reduzierende Lernrate eingesetzt. Diese wurde mit beginnendem Training des nächsten Subdatensatzes zurückgesetzt. Es findet daher eine starke Anpassung der Gewichte auf einen spezifischen Teil der Daten statt, bevor neue Daten gesehen werden können. Dies soll die Fehlerfunktion zu einem stabilen Minimum treiben und ein erneutes Abdriften verhindern. Für die Trainingsstrategie wurde die Überlegenheit gegenüber einer Standard-Generatorfunktion gezeigt, da es hiermit zur deutlich schnelleren Konvergenz des Netzes, sowie besseren Ergebnissen anhand der objektiven Evaluierungsmaßen kommt. Für eine weitere Verbesserung der Trainingsstrategie sollten die Optimierer AdaGrad und AdaDelta untersucht werden, da für diese von Vorteilen bei der Optimierung von dünn besetzten Daten berichtet wird. Diese können mit weiteren Strategien der Lernratenabschwächung kombiniert werden. Die Ergebnisse aus der ResNet Architektur zeigten sich als besonders gut im Erkennen und der

Dämpfung der Sirenenstörung, welches auf die klare Struktur der Störung und der besonderen Fähigkeit der Strukturerkennung von CNNs zurückgeführt wird. Für andere Störungen bleiben die Ergebnisse jedoch hinter anderen vorgestellten Methoden zurück.

Im Anschluss daran wurde eine neuartige Architektur vorgestellt, welche im Rahmen dieser Arbeit entwickelt wurde. Hierfür wurde erneut die Unterteilung des Frequenzspektrums in Bark-Bänder vorgenommen. Diesmal wurde diese innerhalb des Netzgraphen vorgenommen. Um für die Subbänder relevante Features zu erkennen, wurden eindimensionale Convolutions auf diesen durchgeführt. Diese nutzen dabei eine Dilatation der Kernel, sowie wachsende Fenstergrößen, welche mit der Tiefe des Netzes zunehmen. Max- und Global Average Pooling ermöglichen hierbei eine Konvergenz. Auf die parallelen CNN folgen auf einzelne Subband beschränkte LSTMs, die im Anschluss konkateniert und durch einen Dense Layer abgeschlossen werden. Diese Architektur ist dabei das Ergebnis ausgiebiger Testung aus verschiedenen Konfigurationen der genannten Techniken. Somit konnten nicht nur Ergebnisse im Vergleich zum Residual CNN verbessert werden sondern auch eine durchschnittliche Verbesserung von 0.52 Punkten über alle gewählte Störungen bei höheren SNR erzielt werden. Für die STOI Metrik konnte eine Verbesserung für die Störung durch Sirene und eine minimale Verbesserung durch die Störung mit Straßenlärm gezeigt werden. Auch in dieser Architektur konnte die Überlegenheit der vorher vorgestellten Trainingsstrategie mit Subdatensätzen bestätigt werden. Im Weiteren wurden Abstraktionsversuche der Architektur beschrieben, die teils zu schlechteren Ergebnissen führten, wie etwa der Umkehr der Convolution zur Frequenzachse. Auch wurde versucht Residual Strukturen mit in die Architektur einzubringen.

Aufgrund einer aktuellen Publikation, in der DenseNet Blöcke mit Residual Verbindungen kombiniert werden, wurden anfängliche Untersuchungen zur Performanz von DenseNet Blöcken vorgenommen. Hierfür wurde ein DenseNet implementiert und so abstrahiert, dass es, anstatt einer Klassifikation, eine Regression vornimmt. Auch hier konnte die Fähigkeit des DenseNets zur Sprachverbesserung bestätigt werden, jedoch bleiben die Ergebnisse hinter den anderen vorgestellten Methoden zurück. Für weiterführende Untersuchungen wurde daher eine bereits entwickelte Architektur empfohlen.

Im Kapitel zur Anpassung der Fehlerfunktion wurden die Nachteile der Berechnung der Fehlerfunktion für die MSE Methode erörtert. Diese korreliert nicht ausreichend mit den objektiven Evaluierungsmetriken, die das menschliche Empfinden bewerten sollen. Begründet wird dies damit, dass hierdurch zwei Fehlertypen entstehen. Eine zu starke oder zu schwache Dämpfung des Signals, welche für das menschliche Empfinden als unterschiedlich abträglich bewertet werden. Durch unterschiedliche Bewertung, unter Zuhilfenahme weiterer Methoden des PESQ Standards, versucht

die PMSQE Fehlerfunktion diese Effekte auszugleichen.

Der zur Berechnung beigelegte Programmcode aus der Publikation zielt darauf ab, die Berechnung der Fehlerfunktion auf einem kurzen Zeitintervall im Zeitbereich des Signals durchzuführen. Für den hier gewählten Maskierungsansatz musste daher eine Lösung zur Abhilfe gefunden werden. Aufgrund der Fähigkeit der PMSQE auch frameweise Metriken auszugeben, konnte durch eine Anpassung des Netzes ein Training ermöglicht werden. Dafür wurden die STFT Eingangsdaten unverändert in das Netz gegeben, um mit der Methode des Tensorslicing in Kombination mit einer Wandlung der Daten zum Leistungsspektrum ein direktes Ergebnis der entstörten STFT Amplituden, innerhalb des Netzgraphen, zu erzeugen. Dieses wurde für die Erweiterung des Netzes um einen zweiten Ausgang genutzt, auf welchen die PMSQE Fehlerfunktion angewendet werden konnte. Die Ergebnisse hieraus zeigen eine deutliche Verbesserung der Entstörleistung für die Störung im Autoinnenraum, die mit anderen Methoden scheiterte.

Nachdem die vorgestellten Netze die Fähigkeit zur Berechnung der IRM bewiesen haben, wurde zudem untersucht, ob diese auch in der Lage sind eine Schätzung von a-priori SNR zu leisten. Hierfür wurden die theoretischen Grundlagen, sowie der parametrisierte Wiener Filter erläutert. Die gewählte Architektur zeigte jedoch in der aktuellen Konfiguration keine ausreichende Leistungsfähigkeit zur Schätzung der a-priori SNR.

Abschließend wurden Nachbearbeitungsmethoden untersucht. Dafür wurden zunächst die Histogramme der geschätzten Gain-Parameter für eine IRM mit denen der ground truth verglichen. Es zeigte sich ein Problem durch Überglättung der inferierten Werte, wie es auch in der Literatur bereits berichtet wurde. An dieser Stelle sei erwähnt, dass es möglicherweise einen Zusammenhang bei der Konstruktion der IRM und den durch das Netz induzierten Fehlern gibt bzw. eine andere Verteilungsfunktion möglich ist, wenn die IRM nicht über das bereits logarithmierte Spektrum konstruiert wird. Für weitere Untersuchungen, auch im Hinblick auf die Entstörleistung, sollte hier die Fähigkeit zur Schätzung der Gain-Parameter und deren Fehler im Hinblick auf die IRM genauer untersucht werden, mit direktem Vergleich der Konstruktion der IRM über die Amplituden des Leistungsspektrum allein.

Zudem wurde das Verfahren der Global Variance Equalization implementiert, indem zunächst dimensionsunabhängige und dimensionsabhängige Varianzen der STFT Features über die Frequenz-Bins untersucht wurden. Hier konnte ebenfalls eine Minderung der Varianz durch Inferenz des Netzes gesehen werden. Die Varianz wurde im Folgenden angehoben, um diese an die Werte des ungestörten Signals anzugeleichen. Die Auswertung zeigte, dass es durch diese Nachbearbeitungsmethode bei vier Störungen zu Verbesserungen der Metrik kam, vor allem die Störung durch eine Sirene konnte stark von der Nachbearbeitung profitieren.

Eine weitere Möglichkeit der Nachbearbeitung besteht in einer Erhöhung der Dämp-

fung durch einen zusätzlichen Parameter in der Gain-Funktion. Hier wurden die objektiven Metriken für verschiedene Parameter untersucht und es wurde gezeigt, dass diese Technik als Trade-off zwischen der Qualität und Verständlichkeit der Sprache eingesetzt werden kann. Außerdem wurden beide Nachbearbeitungsmethoden miteinander kombiniert, wodurch gezeigt werden konnte, dass die GVE in Teilen vermag induzierte Störungen durch Erhöhung des Dämpfungsfaktors wieder auszugleichen. Abschließend bleibt zu erwähnen, dass in dieser Arbeit hauptsächlich die Auswirkungen der Sprachverbesserung auf die menschliche Wahrnehmung untersucht wurden. Die Bestrebung der Sprachentstörung als Front-End für automatische Spracherkennungssysteme zeigt dabei weitere Herausforderungen, die einer Analyse bedürfen.

Die Untersuchungen und Recherchen zur Entstörung von Sprachsignal mit Deep Learning Methoden zeigte ein aktives und in schneller Wandlung befindliches Forschungsfeld, welches kurz vor einer Anwendung in breiten Bereichen von Kommunikationssystemen zu stehen scheint. Dies bestätigt sich an den ersten Beta-Versionen industrieller Sprachverbesserungssysteme, die auf eine ganzheitliche Nutzung in einer Vielzahl von Umgebungen abzielen. Auch wenn es hier teils weiter zu Berichten schlechter Performanz in einigen Umgebungen kommt, sind die Ergebnisse doch bereits bemerkenswert. Ebenso zeigen sich bereits gute offene Forschungsergebnisse mit respektabler Entstörleistung, wie beim DeepXi Framework. Aktuelle Bemühungen in der Forschungsgemeinschaft bestehen in der Vereinheitlichung und der ermöglicht kollaborativer Forschung. Daher gibt es im Zuge der INTERSPEECH Konferenz in diesem Jahr, Bestrebungen die Vergleichbarkeit von Ergebnissen zu verbessern, in dem Datensätze mit Sprache und Störgeräuschen sowie Baseline Architekturen bereitgestellt werden. Hier soll die Diskrepanz zwischen synthetischen Aufnahmen und den Gegebenheiten von Aufnahmen aus der echten Welt, die eine Breite von Schwierigkeiten mit sich bringen, gemindert werden. Wie auch aus den Ergebnissen der Arbeit zu sehen, bleibt festzuhalten, dass das Forschungsfeld der Sprachverbesserung in den nächsten Jahren weiter im Fokus bleiben wird. Gerade die Generalisierung auf möglichst viele Umgebungen, weitere Verbesserung der Entstörleistung und Anhebung der Verständlichkeit, sowie die Implementierung auf einer breiten Anzahl von Konsumgeräten, die entsprechenden Beschränkungen der Hardware unterliegen, werden die Betätigungen in diesem Forschungsfeld in den nächsten Jahren prägen.

## A Literaturverzeichnis

- [1] Bert-Uwe Köhler: *Konzepte der statistischen Signalverarbeitung* Heidelberg: 2005
- [2] Xu, Yong & Du, Jun & Dai, Li-Rong & Lee, Chin-Hui. (2015). *A Regression Approach to Speech Enhancement Based on Deep Neural Networks*. IEEE/ACM Transactions on Audio, Speech, and Language Processing. 23. 7-19. 10.1109/TASLP.2014.2364452.
- [3] Mark Marzinzik, Birger Kollmeier: *Ein Überblick über die Störgeräuschunterdrückungsalgorithmen nach Ephraim-Malah* Übersichtsarbeit Carl von Ossietzky Universität Oldenburg, 2001
- [4] Y. Ephraim and D. Malah: *Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator* in IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 32, no. 6, pp. 1109-1121, December 1984.
- [5] Cappe (1994): *Elimination of the musical noise phenomenon with the Ephraim and Malah noise suppressor*. IEEE Trans-actions on Speech and Audio Processing 2 (2), 345- 349
- [6] Y. Ephraim and D. Malah: *Speech enhancement using a minimum mean-square error log-spectral amplitude estimator*, in IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 33, no. 2, pp. 443-445, April 1985.
- [7] A. Nicolson, K. K. Paliwal: *Deep learning for minimum mean-square error approaches to speech enhancement*, Speech Communication 111 (2019) 44 - 55, <https://doi.org/10.1016/j.specom.2019.06.002>.
- [8] Christian W. Dawson and Robert Wilby (1998): *An artificial neural network approach to rainfall-runoff modelling*, Hydrological Sciences Journal, 43:1, 47-66, DOI: 10.1080/02626669809492102
- [9] R. Rojas: Neural Networks, Chapter 7, Springer-Verlag, Berlin, 1996

- 
- [10] Diederik P. Kingma, Jimmy Ba *Adam: A Method for Stochastic Optimization* 2015 arXiv:1412.6980
  - [11] Gang Chen: *A Gentle Tutorial of Recurrent Neural Network with Error Backpropagation*, 2018 arXiv:1610.02583
  - [12] Sepp Hochreiter and Jürgen Schmidhuber: *Long Short-Term Memory*. Neural Comput. 9, 8 (November 1997), 1735–1780. DOI:<https://doi.org/10.1162/neco.1997.9.8.1735>
  - [13] Nal Kalchbrenner, Ivo Danihelka, Alex Graves: *Grid Long Short-Term Memory*, 2015 arXiv:1507.01526
  - [14] Kaiming He et al: *Deep Residual Learning for Image Recognition* 2015 arXiv:1512.03385
  - [15] Aaron Nicolson Kuldip K. Paliwal: *Bidirectional Long-Short Term Memory Network-based Estimation of Reliable Spectral Component Locations* Conference: Interspeech 2018
  - [16] Li N, Loizou PC: *Factors influencing intelligibility of ideal binary-masked speech: implications for noise reduction*. J Acoust Soc Am. 2008;123(3):1673–1682. doi:10.1121/1.2832617
  - [17] Naik, G. R., & Wang, W. (Eds.). (2014): *Blind Source Separation*. Signals and Communication Technology. doi:10.1007/978-3-642-55016-4
  - [18] Shasha Xia, Hao Liand Xueliang Zhang: *Using Optimal Ratio Mask as Training Target for Supervised Speech Separation* <https://arxiv.org/pdf/1709.00917.pdf>
  - [19] Narayanan, A., & Wang, D. (2013): *Ideal ratio mask estimation using deep neural networks for robust speech recognition*. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. doi:10.1109/icassp.2013.6639038
  - [20] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio: *Generative Adversarial Networks* arXiv:1406.2661
  - [21] Deepak Baby: *iSEGAN: Improved Speech Enhancement Generative Adversarial Networks* arXiv:2002.08796v1
  - [22] Han Zhao, Shuayb Zarar, Ivan Tashev, Chin-Hui Lee: *Convolutional-Recurrent Neural Networks for Speech Enhancement* arXiv:1805.00579

- [23] Shaojie Bai, J. Zico Kolter, Vladlen Koltun: *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling* arXiv:1803.01271
- [24] Dario Rethage, Jordi Pons, Xavier Serra: *A Wavenet for Speech Denoising* arXiv:1706.07162
- [25] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu: *Pixel recurrent neural networks* arXiv:1601.06759, 2016.
- [26] Yi Hu and Philipos C Loizou: *Evaluation of objective measures for speech enhancement*. Interspeech, 2006.
- [27] Santiago Pascual1, Antonio Bonafonte1, Joan Serra SEGAN: Speech Enhancement Generative Adversarial Network arXiv:1703.09452
- [28] Maas, A.L. et al.: *Rectifier Nonlinearities Improve Neural Network Acoustic Models*. Proc. 30 th Int. Conf. Mach. Learn. 28, 6 (2013)
- [29] Quan Wang et al.: *VoiceFilter: Targeted Voice Separation by Speaker-Conditioned Spectrogram Masking* arXiv:1810.04826v6
- [30] Kin Wah et al.: *Singing Voice Separation Using a Deep Convolutional Neural Network Trained by Ideal Binary Mask and Cross Entropy* arXiv:1812.01278
- [31] Yuxuan Wang, Kun Han, and DeLiang Wang: *Exploring Monaural Features for Classification-Based Speech Segregation*, IEEE transactions on audio, speech, and language processing, Vol.21, No.2, February 2013
- [32] Aaron Nicolson, Jack HansonJames Lyons, James Lyons, Kuldip Palwal: *Spectral Subband Centroids for Robust Speaker Identification Using Marginalization-based Missing Feature Theory* International Journal of Signal Processing Systems, 2018
- [33] Christopher Hummersone, Toby Stokes and Tim Brookes: *On the Ideal Ratio Mask as the Goal of Computational Auditory Scene Analysis*
- [34] A. W. Rix, J. G. Beerends, M. P. Hollier and A. P. Hekstra: *Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs* 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221), Salt Lake City, UT, USA, 2001, pp. 749-752 vol.2, doi: 10.1109/ICASSP.2001.941023.

- [35] C. H. Taal, R. C. Hendriks, R. Heusdens and J. Jensen: *An Algorithm for Intelligibility Prediction of Time-Frequency Weighted Noisy Speech* in IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 7, pp. 2125-2136, Sept. 2011, doi: 10.1109/TASL.2011.2114881.
- [36] Li, R., Sun, X., Liu, Y. et al.: *Multi-resolution auditory cepstral coefficient and adaptive mask for speech enhancement with deep neural network* EURASIP J. Adv. Signal Process. 2019, 22 (2019). <https://doi.org/10.1186/s13634-019-0618-4>
- [37] Y. Wang and D. Wang, *Towards Scaling Up Classification-Based Speech Separation* in IEEE Transactions on Audio, Speech, and Language Processing, vol. 21, no. 7, pp. 1381-1390, July 2013, doi: 10.1109/TASL.2013.2250961.
- [38] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár: *Focal Loss for Dense Object Detection* arXiv:1708.02002
- [39] Hiroaki Nakajima, Yu Takahashi, Kazunobu Kondo, Yuji Hisamimoto *Monaural source enhancement maximizing source-to-distortion ratio via automatic differentiation* arXiv:1806.05791
- [40] Yong Xu, Jun Du, Zhen Huang, Li-Rong Dai, Chin-Hui Lee: *Multi-Objective Learning and Mask-Based Post-Processing for Deep Neural Network Based Speech Enhancement* arXiv:1703.07172
- [41] K. Tan, J. Chen and D. Wang: *Gated Residual Networks With Dilated Convolutions for Monaural Speech Enhancement* in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 27, no. 1, pp. 189-198, Jan. 2019, doi: 10.1109/TASLP.2018.2876171.
- [42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun: *Identity Mappings in Deep Residual Networks* arXiv:1603.05027
- [43] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger, *Densely Connected Convolutional Networks* arXiv:1608.06993
- [44] Mohammad Nikzad, Aaron Nicolson, Yongsheng Gao, Jun Zhou, Kuldeep K. Paliwal, Fanhua Shang, *Deep Residual-Dense Lattice Network for Speech Enhancement* arXiv:2002.12794
- [45] Yong Xu, Jun Du, Li-Rong Dai, Chin-Hui Lee, Chin-Hui Lee: *Global variance equalization for improving deep neural network based speech enhancement* July 2014, DOI: 10.1109/ChinaSIP.2014.6889204

- 
- [46] Samba Raju Chiluveru, Manoj Tripathy: *Low SNR speech enhancement with DNN based phase estimation* Int J Speech Technol 22, 283–292 (2019). <https://doi.org/10.1007/s10772-019-09603-y>
  - [47] Se Rim Park, Jinwon Lee: *A Fully Convolutional Neural Network for Speech Enhancement* arXiv:1609.07132
  - [48] Jian Tang, Yan Song, LiRong Dai, Ian McLoughlin *Acoustic Modeling with Densely Connected Residual Network for Multichannel Speech Recognition*
  - [49] Loizou PC, Kim G.: *Reasons why current speech-enhancement algorithms do not improve speech intelligibility and suggested solutions.* IEEE Trans Audio Speech Lang Process. 2011;19(1):47-56. doi:10.1109/TASL.2010.2045180
  - [50] J. M. Martin-Doñas, A. M. Gomez, J. A. Gonzalez and A. M. Peinado: *A Deep Learning Loss Function Based on the Perceptual Evaluation of the Speech Quality* in IEEE Signal Processing Letters, vol. 25, no. 11, pp. 1680-1684, Nov. 2018, doi: 10.1109/LSP.2018.2871419.
  - [51] Morten Kolbæk, Zheng-Hua Tan, Søren Holdt Jensen, and Jesper Jensen: *On Loss Functions for Supervised Monaural Time-Domain Speech Enhancement* arXiv:1909.01019
  - [52] Wang Y, Narayanan A, Wang D. *On Training Targets for Supervised Speech Separation.* IEEE/ACM Trans Audio Speech Lang Process. 2014;22(12):1849-1858. doi:10.1109/TASLP.2014.2352935
  - [53] Zhang Huimin, Jia Xupeng, and Li Dongmei. 2019. *An Iterative Post-processing Approach for Speech Enhancement.* In Proceedings of the 2019 4th International Conference on Multimedia Systems and Signal Processing (ICMSSP 2019). Association for Computing Machinery, New York, NY, USA, 130–134. DOI:<https://doi.org/10.1145/3330393.3330427>

## B Quellenangaben

- [A1] Wikipedia - *Backpropagation* 14.03.2020 <https://de.wikipedia.org/wiki/Backpropagation>
- [A2] Wikipedia - *Stochastic Gradient Descent* 14.03.2020 [https://en.wikipedia.org/wiki/Stochastic\\_gradient\\_descent2](https://en.wikipedia.org/wiki/Stochastic_gradient_descent2)
- [A3] Ben Khuong, *The Basics of Recurrent Neural Networks (RNNs)* <https://medium.com/towards-artificial-intelligence whirlwind-tour-of-rnns-a11effb7808f>
- [A4] Christopher Olah, *Understanding LSTM Networks* <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- [A5] Wikipedia - *MNIST* 18.03.2020 [https://en.wikipedia.org/wiki/MNIST\\_database](https://en.wikipedia.org/wiki/MNIST_database)
- [A6] Arden Dertat *Convolutional Neural Networks* <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2>
- [A7] Shiva Verma *Understanding 1D and 3D Convolution Neural Network* <https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610>
- [A8] [https://en.wikipedia.org/wiki/Generative\\_adversarial\\_network](https://en.wikipedia.org/wiki/Generative_adversarial_network) 26.03.2020
- [A9] <https://github.com/anicolson/DeepXi>
- [A10] [https://de.wikipedia.org/wiki/Unüberwachtes\\_Lernen](https://de.wikipedia.org/wiki/Unüberwachtes_Lernen) 02.04.2020
- [A11] <https://developer.nvidia.com/gtc/2019/video/S9247> 07.05.2020
- [A12] [https://en.wikipedia.org/wiki/Mel-frequency\\_cepstrum](https://en.wikipedia.org/wiki/Mel-frequency_cepstrum) 09.05.2020
- [A13] <https://de.wikipedia.org/wiki/Bark-Skala> 12.05.2020

- [A14] <https://towardsdatascience.com/visualizing-intermediate-activations-of-a-cnn-trained-on-the-mnist-dataset-2c34426416c8>
- [B1] <https://mc.ai/convolution-operation-comprehensive-guide/>
- [R1] <https://librosa.github.io/librosa/>
- [R2] <https://github.com/Sato-Kunihiko/audio-SNR/>
- [R3] <https://voice.mozilla.org/de/datasets>
- [R4] <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>