

# Trendbewertung in Ultraschall-Sensorsystemen mit LSTM basierten RNN-Autoencodern

Steffen Seitz<sup>1</sup>, Christian Probst<sup>2</sup>, Jens Müller<sup>1</sup>, Ronald Tetzlaff<sup>1</sup>, Peter Holstein<sup>2</sup>

<sup>1</sup> Technische Universität Dresden, 01062 Dresden

<sup>2</sup> SONOTEC Ultraschallsensorik Halle GmbH, 06112 Halle (Saale)

## 1 Kurzfassung

Moderne Verfahren zur Trendbewertung bei Instandhaltung sind ein wichtiger Aspekt der Arbeits- und Ausfallsicherheit komplexer industrieller Anlagentechnik.

In diesem Beitrag soll gezeigt werden, wie mit Hilfe Long Short-Term Memory (LSTM) basierter Recurrent Neural Network (RNN)-Autoencoder eine Trendbewertung durch die Auswertung sowohl synthetischer als auch realer Messdaten von Kugellagern verschiedener Schädigungsgrade durchgeführt werden kann. Im Beitrag soll letztlich eine Möglichkeit der Nutzung des Rekonstruktionsfehlers als Maß für die Degeneration eines überwachten Prozesses bei Langzeitüberwachung diskutiert werden.

## 2 Einleitung

Die Anwendung Ultraschall-basierter berührungsloser Messverfahren spielt insbesondere im Umfeld von risikoreicher Prozesse eine wichtige Rolle. Beispielhaft können dadurch auf Verschleiß basierende Risse in Rohren oder Gastanks erkannt, sowie die Degeneration verschiedener sich verändernder Maschinenzustände über einen Wartungszeitraum überwacht werden. Eine datengetriebene Auswertung durch entsprechende Algorithmen wäre in der Lage, eine Dauerüberwachung des Prozesses zu ermöglichen, da entsprechende Sensordaten kontinuierlich zur Verfügung stehen. Geeignete Lernverfahren sind in der Lage aus diesen Daten relevante Information zu extrahieren und so analog zum menschlichen Experten für eine Diagnose des überwachten Prozesses zu nutzen.

Dabei spielt die Auswahl der dem Entscheider zugeführten Merkmale eine wichtige Rolle. Die für eine Diagnose des Systems nötige Information kann je nach Prozess in verschiedenen Merkmalen enthalten sein. Menschliche Entscheider nutzen entsprechendes Fachwissen um aus der zugrunde liegenden Zeitreihe eine korrekte Merkmalsauswahl treffen zu können. Ein maschinelles Lernverfahren muss in der Lage sein, entsprechendes Fachwissen, ohne menschliches Eingreifen (z.B. Labeln), als unüberwachtes Lernverfahren aus der Zeitreihe des Sensors zu extrahieren, um so eine korrekte Beurteilung des aktuellen Systemzustands zu ermöglichen.

## 3 Autoencoder und LSTM

In aktuellen Arbeiten zur automatischen unüberwachten Extraktion von Merkmalen aus Daten werden häufig so-

genannte Autoencoder herangezogen. Diese werden unter anderem zu Übersetzung menschlicher Sprache (als sogenannter Sequenz zu Sequenz Autoencoder) [1], zur Sequenzierung von Zellen in der Biologie [2], zur Sprachsynthese [3] und der Generierung von Bildbeschreibungen benutzt [4]. Verschiedene Formen des Autoencoders sind Gegenstand aktueller Forschung. Beispielsweise können Autoencoder auch dazu benutzt werden Verteilungen über Daten zu berechnen [5], wodurch generative Sampling ermöglicht wird.

### Autoencoder

Ein auf klassischen Sigmoid Neuronen bestehender Autoencoder (Abbildung 1) ist in seiner Struktur einem Multilayer Perceptron (MLP) ähnlich. Eingangsdaten der Länge  $L$  eines Sensors  $X = \{x_1, x_2, \dots, x_L\}$  werden durch ein Netzwerk vollverbundener Feedforward Sigmoidneuronen in eine Schicht mit Ausgabewerten  $X' = \{x'_1, x'_2, \dots, x'_L\}$  überführt. Das Netzwerk wird dabei darauf trainiert, die Eingabewerte an der Ausgabeschicht wieder zu rekonstruieren. Um das Lernen der Einheitsfunktion zu verhindern, besitzt ein hidden layer in der Mitte des Netzwerks eine niedrigere Dimension als der Eingabevektor, auch Kontextvektor oder Latent genannt. Die niedrige Dimension des Latents zwingt den Autoencoder Information über die Eingangsdaten in komprimierter Form im Kontextvektor zu speichern. Dieser Schritt wird auch als Encoder Schritt bezeichnet. Aus diesem Kontextvektor wird dann im sogenannten Decodierungsschritt versucht, anhand der gespeicherten Information das Originalsignal zu bestimmen.

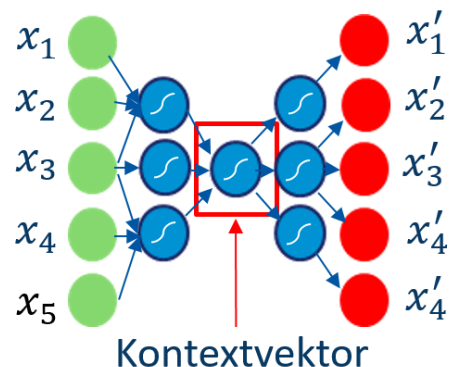


Abbildung 1: Klassischer MLP Autoencoder

## Recurrente Neuronale Netze und Long Short-Term Memory (LSTM)

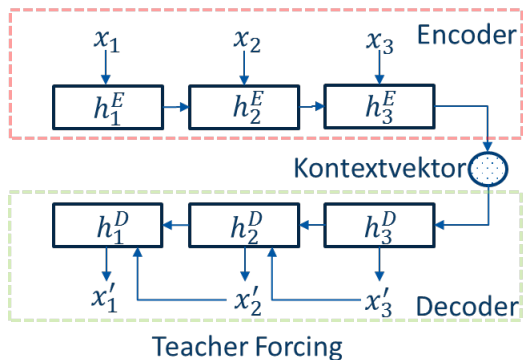
Die im klassischen Autoencoder verwendeten Sigmoid Neuronen sind nicht in der Lage zeitliche Abhängigkeiten zwischen den einzelnen Zeitschritten der Sensordaten zu verarbeiten, da das genutzte Feedforward Model keine Rekurrenzen zulässt. Da die Information, welche in den Sensordaten zugrunde liegt, allerdings auch solche zeitliche Abhängigkeit besitzt, ist es sinnvoll, diese in das Modell zu übernehmen.

Klassische Rekurrente Neuronale Netzwerke sind allerdings nicht in der Lage, Abhängigkeiten für viele Schritte  $X_l = \{x_1, x_2, \dots, x_l\}$  im Betrachtungszeitraum  $l$  oder auch Lookback genannt, mit  $l < L$  zu verarbeiten [6].

Aus diesem Grund wurden die sogenannten Long Short-Term Memory Networks (LSTM) eingeführt [7]. Diese regulieren durch eine Torstruktur den Informationsfluss und ermöglichen so auch das Lernen von langen Sequenzen. LSTM Zellen besitzen dabei einen hidden state  $h$ , welcher über eine nichtlineare Funktion an den Zellzustand gekoppelt ist. Dieser hidden state wird bei der Berechnung einzelner Zeitschritte jeweils weitergegeben, um so bereits Erlerntes in die aktuelle Berechnung mit einfließen lassen zu können.

### LSTM Autoencoder

Entsprechend der Vorgehensweise zur Konstruktion eines auf Sigmoid Neuronen basierenden Autoencoders, lässt sich diese Struktur auch aus LSTM Zellen konstruieren (Abbildung 2). Im Gegensatz zum MLP Autoencoder ist



**Abbildung 2:** Verwendetes Encoder-Decoder Model für einen Input von  $\{x_1, x_2, x_3\}$  zur Schätzung von  $\{x'_1, x'_2, x'_3\}$

dieser in der Lage, zeitliche Abhängigkeiten in ihren Kontextvektor zu überführen um so auch auf Zeitreihen basierende Signale auswerten zu können.

Analog zu [8] und [9] wird die Struktur darauf trainiert, die invertierte Eingangszeitreihe  $X'_i = \{x'_l, \dots, x'_2, x'_1\}$  zu errechnen. Dazu wird die Eingangszeitreihe  $X_i = \{x_1, x_2, \dots, x_l\}$  im Encoderschritt zusammen mit den hidden Encoder states  $\mathbf{h}_{i-1}^E$ , wobei  $i \in \{1, 2, \dots, l\}$  den aktuellen Encoderzeitschritt darstellt, durch die LSTM Zellfunktionen verrechnet und der Zellzustand an den

nächsten Zeitschritt übergeben. Dabei ist  $\mathbf{h}_i^E \in R^c$  mit  $c$  gleich der Anzahl der LSTM Zellen des ditten layers im Encoder.

Der finale hidden state des Encoders  $\mathbf{h}_l^E$  wird im Folgenden zur Initialisierung des  $\mathbf{h}_l^D$  im Decoders genutzt, sodass  $\mathbf{h}_l^E = \mathbf{h}_l^D$  gilt. Für jeden weiteren Decoderzeitschritt  $l - i$  errechnet dieser zusammen aus dem letzten hidden state  $\mathbf{h}_{l-i+1}^D$  zusammen mit der Ausgabe des vorhergehenden Decoderzeitschritts  $x'_{l-i+1}$  den aktuellen Ausgabewert  $x'_{l-i}$ . Um Fehlerfortpflanzung zu verhindern steht dem Decoder für jedes  $l - i$  der korrekten Ausgabewert des Zeitschritts  $l - i + 1$  zur Verfügung („Teacher Forcing“), aus denen dieser seinen Folgeausgabewert berechnet.

Als Kostenfunktion zwischen tatsächlichem und prädiertem Wert wird der mittlere quadratische Fehler zwischen Ein- und Ausgabewert minimiert (Gleichung 1).

$$F = \sum_{i=1}^l \|x_i - x'_i\|^2 \quad (1)$$

### Codierung durch lineare Prädiktion

Die durch Anwendung des LSTM erhaltenen Ergebnisse wurden mit einem klassischen Prädiktionsverfahren verglichen, welches ebenfalls für nichtstationäre Signale verwendet werden kann.

*Codierung durch lineare Prädiktion* oder kurz LPC [10] setzt einen quasistationären Ansatz bei der Berechnung von Prädiktorkoeffizienten  $\alpha_k$  durch Minimierung der Summe Fehlerquadrate voraus. Mit deren Hilfe lässt sich analog zu Gleichung 2 der Folgewert  $x'_{LPC}$  aus vorangegangenen Zeitabschnitt berechnen, wobei  $k$  gleich der zeitlichen Verschiebung bzw. der Anzahl der verwendeten Prädiktorkoeffizienten ist und  $l$  in diesem Zusammenhang auch als Prädiktionsordnung bezeichnet wird.

$$x'_{LPC,i} = \sum_{k=1}^l a_k \cdot x_{i-k} \quad (2)$$

$$F_{LPC} = \sum_{i=1}^l \|x_i - x'_{LPC,i}\|^2 \quad (3)$$

Zur Definition des Prädiktionsfehlers bzw. der Kostenfunktion wurde Gleichung 3 zugrunde gelegt.

## 4 Datenbasis

Zur Evaluierung des LSTM Autoencoders für die Trendbewertung wurden zwei Datensätze ausgewertet.

## Synthetischer Schwingungsdatensatz

Zur Prüfung der prinzipiellen Funktionsweise wurde ein künstlich erzeugter Datensatz einer stationären sinusförmigen Schwingung erstellt. Während des Trainings wurde der Autoencoder darauf trainiert diesen Sinus zu rekonstruieren. Während des Tests wurde die dem Autoencoder zugeführte Schwingung additiv mit einem Rauschsignal überlagert und der Rekonstruktionsfehler notiert.

## Realer Kugellagerdatensatz

Um den Autoencoder an einem industrienahen Beispiel testen zu können wurden drei intakte Kugellager durch Lasergravuren fünf verschiedener Stärken (Fehlergrad 1 bis 5) beschädigt (Abbildung 3).



Abbildung 3: Präparierte Kugellager

Bei der Messung selbst wurde zunächst das intakte Lager (Fehlergrad 0) in den Versuchsaufbau integriert und ein Ultraschallsensor auf dem Versuchsobjekt angebracht, der Aufbau wird in Abbildung 4 gezeigt. Nach der Messung des aktuellen Fehlergrades wurde das Lager durch den Laser in den nächsten Fehlergrad überführt und erneut gemessen. Diese Prozedur wurde für die drei Lager schrittweise bis zum größten Fehlerzustand 5 fortgeführt, um so einen Verschleißprozess zu emulieren.

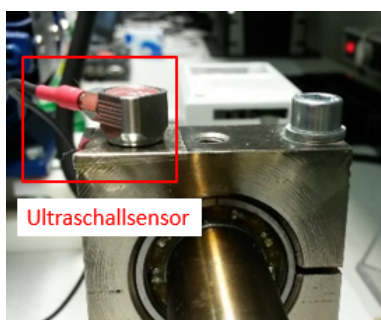


Abbildung 4: Kugellager mit Ultraschallsensor

Die Lagerdrehfrequenz einer Messung betrug dabei 500 rpm, 1200 rpm sowie 1800 rpm. Den Ausführungen in Abschnitt 3 folgend wurden zum Training des Autoencoders die Daten des intakten Lagers (Fehlergrad 0) genutzt.

Um verschiedene Veränderungen vergleichen zu können, wurden sowohl die durch Lasergravuren beschädigten Kugellager als auch das durch Rauschen überlagerte Testsignal des Fehlergrades 0 bei der Auswertung des Rekonstruktionsfehlers untersucht.

## Ergebnisse

### Synthetischer Schwingungs Datensatz

Der eine Epoche lang auf dem rauschfreien Signal trainierte Autoencoder ist in der Lage das sinusförmige Testsignal weitgehend zu rekonstruieren und so entrauschen (Abbildung 5).

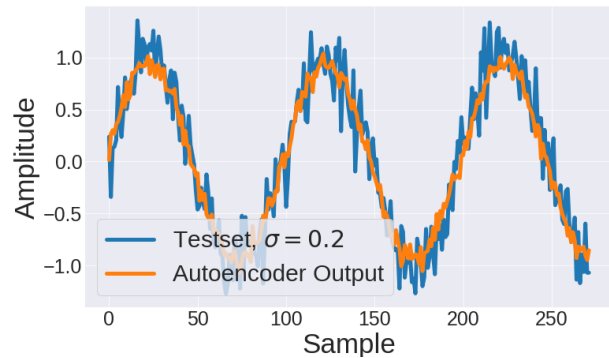


Abbildung 5: Autoencoder Eingangssignal (blau) und Ausgangssignal (orange) für eine verrauschte Sinusschwingung

Je größer die Standardabweichung dabei gewählt wird, desto höher ist der Rekonstruktionsfehler sowohl des Autoencoders als auch des LPC Verfahrens (Abbildung 6). Beide Verfahren sind demzufolge in der Lage den Verschleiß des Systems abzubilden und so eine Trendbewertung zu ermöglichen.

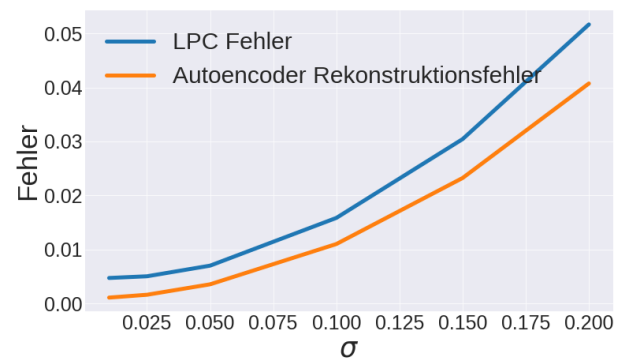
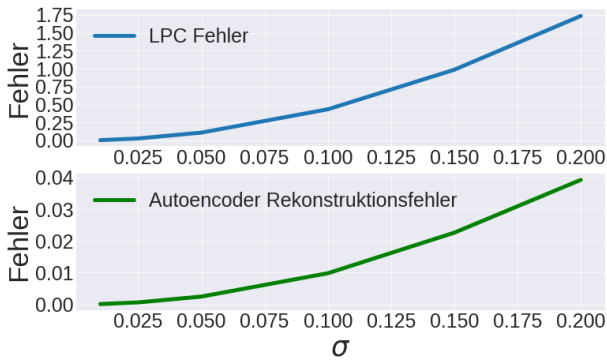


Abbildung 6: Vergleich des Rekonstruktionsfehlers auf der mit synthetischen Verschleiß überlagerten Sinusschwingung für Autoencoder und LPC Verfahren (stationäres Signal)

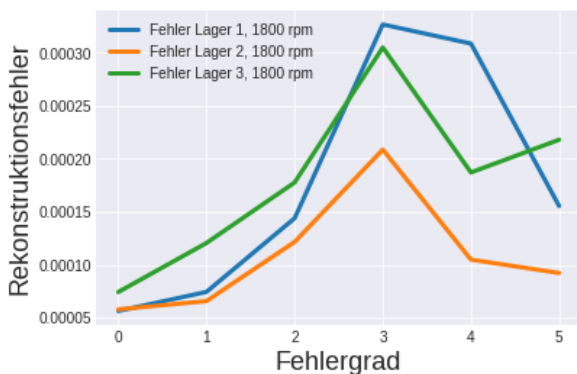
### Realer Kugellager Datensatz

Dieser Effekt ist auch für das mit Rauschen überlagerte reale Kugellagersignal (Fehlergrad 0) in Abbildung 7 sichtbar. Wobei ein Fehler von eins oder größer (vgl. LPC Fehler) eine sehr große Abweichung vom tatsächlichen Signal darstellt, da die Werte der Zeitreihen, aufgrund einer zur besseren Verarbeitung nötigen Skalierung, zwischen null und eins liegen.



**Abbildung 7:** Vergleich des Rekonstruktionsfehlers auf den synthetischen Verschleiß überlagerten Kugellagerdaten (Fehlergrad 0) für Autoencoder und LPC Verfahren (nicht-stationäres Signal)

Der Rekonstruktionsfehler des LPC Verfahrens ist für das nichtstationäre Kugellagersignal deutlich größer als der Fehler des Autoencoders. Im Folgenden sollen nun der Rekonstruktionsfehler des Autoencoders bei denen durch Lasergravur induzierten Fehlerzuständen für 1800 rpm ausgewertet werden .

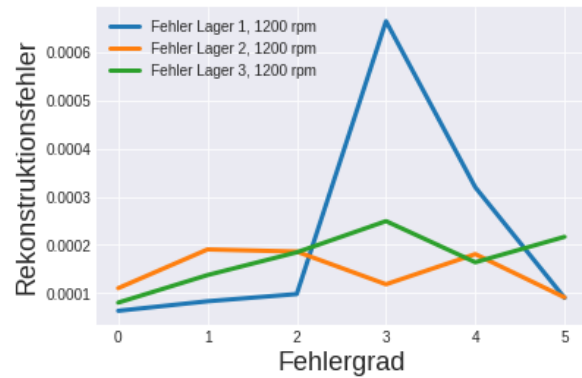


**Abbildung 8:** Vergleich des Rekonstruktionsfehlers des Autoencoders Kugellagerdaten mit Lasergravur bei 1800 rpm Drehzahl

Aus Abbildung 8 wird ersichtlich, dass der Fehler für alle drei Lager bis zum Fehlergrad 3 ansteigt, anschließend fällt er wieder leicht ab. Dieses Verhalten wäre für eine Ausfallvorhersage von Relevanz, da ein sich in Fehlerzustand 3 oder 4 befindliches Lager leicht anhand dieses Peaks bestimmbar wäre. Für eine Drehzahl von 1200 rpm und 500 rpm lässt sich diese Aussage nicht allgemein gültig treffen (Abbildung 9).

## Zusammenfassung

In diesem Beitrag konnte gezeigt werden, dass eine Anwendung von LSTM-Autoencodern auf Ultraschalldaten zur Trendbewertung am Beispiel verschieden präparierter Kugellager grundsätzlich möglich ist. Je nach Drehzahl lassen sich die verschiedenen Fehlergrade anhand des Rekonstruktionsfehlers unterschiedlich gut unterscheiden. Es konnte gezeigt werden das für eine Drehzahl von 1800 rpm Fehlergrad 3 anhand einer Charakteristik des Fehler-



**Abbildung 9:** Vergleich des Rekonstruktionsfehlers des Autoencoders Kugellagerdaten mit Lasergravur bei 1200 rpm Drehzahl

verlaufes bestimmbar ist, wodurch es möglich wäre, eine Ausfallwarnung bei Detektion dieses Zustandes anzugeben.

## Literatur

- [1] Cho, Kyunghyun, et.al.: „Learning phrase representations using rnn encoder-decoder for statistical machine translation“, arXiv preprint arXiv:1406.1078, 2014.
- [2] Ronneberger, Olaf, et.al.: „U-Net: Convolutional Networks for Biomedical Image Segmentation“, MICCAI 2015.
- [3] Li, Jiwei, et.al.: „A hierarchical neural autoencoder for paragraphs and documents“, arXiv preprint arXiv:1506.01057, 2015.
- [4] Bengio, Samy et.al.: „Scheduled sampling for sequence prediction with recurrent neural networks“, Advances in Neural Information Processing Systems, pp. 1171-1179, 2015
- [5] Kingma, Diederik et.al.: „Autoencoding Variational Bayes“, arXiv preprint, arXiv:1312.6114, 2014
- [6] Gers, Felix et.al.: „Learning to Forget: Continual Prediction with LSTM“, Journal Neural Computation archive, Volume 12 Issue 10, 2000
- [7] Hochreiter, Sepp et.al.: „Long short-term memory“, Journal Neural Computation archive, Vol. 9, Issue 8, S. 1735–1780, 1997
- [8] Sutskever, Ilya et.al.: „Sequence to sequence learning with neural networks“, Advances in Neural Information Processing Systems 27, pp. 3104–3112, 2014
- [9] Malhotra, Pankaj et.al.: „LSTM-based Encoder-Decoder for Multi-sensor anomaly Detection“, ICML Anomaly Detection Workshop, 2016
- [10] Hoffmann, Rüdiger et.al.: „Intelligente Signalverarbeitung 1“, Springer Verlag, 2. Auflage, Seite 363 u.F.