

Neuronale Netze zur Klassifikation elektrischer Entladevorgänge

Steffen Seitz¹, Jens Müller¹, Ronald Tetzlaff¹, Peter Holstein²

¹ Technische Universität Dresden, 01062 Dresden

² SONOTEC Ultraschallsensorik Halle GmbH, 06112 Halle (Saale)

Kurzfassung

Moderne Ultraschallverfahren spielen im Umfeld der Instandhaltung eine wichtige Rolle. Die Erkennung von Rissen in Rohren oder von Teilentladungen in Umspannwerken sind dabei nur zwei typische Einsatzgebiete von Ultraschallsensoren in der Praxis. In diesem Beitrag sollen neue Mustererkennungsverfahren für Ultraschalldaten am Beispiel von Münzwürfen vorgestellt sowie deren Leistungsfähigkeit auch im Hinblick auf eine Anwendung für elektrische Entladevorgänge diskutiert werden. Dabei wird mittels eines Pegelkreuzungsverfahrens ein Weg zur Datenreduktion aufgezeigt.

Einleitung

Die Detektion von Teilentladungen an Schaltschranken oder Isolatoren in Umspannwerken ist für die Sicherheit des Personals sowie die Betriebskosten der Betreiber ein elementarer Aspekt. Könnten Teilentladungen sowie deren Position frühzeitig bestimmt werden, wäre es möglich, beschädigte Bauteile präventiv auszutauschen, um etwaige Schäden an Personal oder Material zu vermeiden. Die Detektion von Teilentladungen kann in der Praxis entweder mittels eines kapazitiven Feldsensors, Ultraschall-Geräuscherkennung oder optisch über eine Koronakamera erfolgen [1]. Während erstere zur Detektion der Position der Entladung geeignet ist, aber keinerlei Aussagen über deren Stärke treffen kann [1], ist es möglich durch letztere die Stärke des Feldes, allerdings nicht dessen Koordinaten exakt zu bestimmen. Ultraschallsensoren sind in der Lage, sowohl den Ort, als auch die Stärke des Feldes über Triangulation sowie eine Geräuscherkennung zu bestimmen. Durch einen höheren Automatisierungsgrad der Signalverarbeitung wäre eine schnellere sowie einfachere Auswertung der Sensordaten möglich, was beispielsweise eine Realisierung von automatischen Frühwarnsystemen ermöglichen würde. Dazu müssen die Rohdaten eines Ultraschallsensors vorverarbeitet, entsprechende Merkmale extrahiert und auftretende Muster klassifiziert werden. Für die Aufgaben der Merkmals-Extraktion und Klassifikation werden in diesem Beitrag Methoden des maschinellen Lernens genutzt. Wie in [2] gezeigt wurde, können diese Methoden auf Zellularen Nichtlinearen Netzwerken (CNN) abgebildet werden, was sehr effiziente Implementierungen auf massivparallelen Rechnerstrukturen mit geringer Leistungsaufnahme ermöglicht.

Klassifikation mit Neuronalen Netzwerken

Im Bereich des maschinellen Lernens konnten durch den Einsatz neuraler Netze in den letzten Jahren große Fort-

schritte erzielt werden [4]. Es gibt verschiedene Arten neuronaler Netze, welche sich je nach Anwendungsgebiet in ihrer Leistungsfähigkeit unterscheiden. Zwei der populärsten Netzwerke sind die sogenannten Multilayer-Perceptrons (MLP) sowie Convolutionel Neuronal Networks (ConvNets), deren Grundprinzipien im Folgenden kurz erläutert werden sollen.

Multilayer Perceptrons

Ein MLP besteht aus $K \in \mathbb{N}^+$ Schichten (Input-, Hidden- und Outputlayer) von Neuronen (Abb. 1).

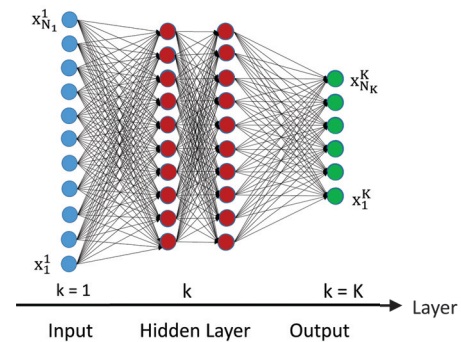


Abbildung 1: Prinzipieller Aufbau eines MLP

Jede Schicht $k \in \{1, \dots, K\}$, umfasst $N_k \in \mathbb{N}^+$ Neuronen, wobei die Anzahl der Input-Neuronen N_1 , von der Menge der Eingangsdaten sowie die Größe des Output-Layers N_K von der Anzahl der Klassen bestimmt ist. Der Ausgabewert $y_i^k \in \mathbb{R}$ des i -ten Neurons der Schicht k entspricht dabei der durch einen zufällig initialisierten Wichtungsvektor $\mathbf{w}_i^k \in \mathbb{R}^{N_k}$ gewichteten Summe sämtlicher Eingänge x_i^k der Schicht k worauf ein Bias $b_i^k \in \mathbb{R}$ addiert wird. Diese werden mit einer Aktivierungsfunktion A zum Eingangswert $x_i^{k+1} \in \mathbb{R}$ eines Neurons der $k+1$ Schicht gemäß:

$$y_i^k = A \left(\mathbf{w}_i^k \top \mathbf{x}^k + b_i^k \right), \quad (1)$$

$$y_i^k = x_i^{k+1}, \quad (2)$$

mit $\mathbf{x}^k = [x_1^k, \dots, x_{N_k}^k]^\top \in \mathbb{R}^{N_k}$ verrechnet. Zwischen den Schichten kann die Aktivierungsfunktion dabei variieren, innerhalb einer Schicht unterscheiden sie sich jedoch nicht. Typische Funktionen sind dabei die Sigmoid, Tanh, ReLu oder Softmax im Falle der Outputneuronen [5]. Dieser Prozess wird für jede Schicht durchgeführt und wird als Feedforward Propagation bezeichnet, wobei das klassifizierte Ereignis dem Maximalwert der Output-Neuronen entspricht. Im Falle von Softmax

ist dies mit der jeweiligen Wahrscheinlichkeit gleichzusetzen. Nach dem Feedforward Prozess kann dann ein Fehler $l(\mathbf{w}, b)$ zum tatsächlichen Ergebnis pro Beispiel bestimmt werden. Als Fehlerfunktion wird dabei typischerweise die Kreuzentropie verwendet:

$$l_{\text{cross}}(\mathbf{w}, b) = - \sum_{i=1}^{N_K} [\hat{y}_i \ln(y_i^K) + (1 - \hat{y}_i) \ln(1 - y_i^K)]. \quad (3)$$

Dabei sind \hat{y}_i die tatsächlichen Referenzwerte der Ausgabeschicht K . Dies kann dann über alle $M \in \mathbb{N}^+$ dem Netzwerk zugeführten Beispiele zum Gesamtfehler $L(\mathbf{w})$ summiert werden:

$$L(\mathbf{w}, b) = \frac{1}{M} \sum_{n=1}^M l_{\text{cross},n}(\mathbf{w}, b) + \lambda R_n(\mathbf{w}). \quad (4)$$

Dabei wird λ als Regularisierungsparameter und $R_n(\mathbf{w})$ als Regularisierungsterm bezeichnet [6]. Der Fehler aus (4) kann mittels Anwendung der Kettenregel auf die einzelnen Neuronen rückgekoppelt werden, was auch als Backpropagation bezeichnet wird und mit Hilfe von Optimierungsverfahren wie Stochastic Gradient Descent (SGD) minimiert werden kann [3]. Dieser Lernprozess wird für eine bestimmte Anzahl an Epochen wiederholt und führt ohne die bereits erwähnte Regularisierung zu einer Überanpassung (Overfitting), sodass keine Generalisierung des Gelernten auf dem Netzwerk während des Trainings vorenthaltene Daten möglich ist. Neben dem Regularisierungsparameter existieren weitere Möglichkeiten, der Überanpassung entgegen zu wirken. Eine populäre Methode ist dabei die Einführung eines sogenannten Dropout-Layers [7], wobei zufällig einige Neuronen aus der Anordnung während des Trainings entfernt werden werden.

Convolutional Neuronal Networks

ConvNets beruhen auf einer mehrdimensionalen Faltungsoperation mehrerer Elemente eines Layers mit verschiedenen Filtermatrizen, deren Filterkoeffizienten analog zu den Gewichten des MLP gelernt werden können. Auf das Ergebnis der Faltungsoperation wird analog zur Berechnung aus (1) eine Aktivierungsfunktion angewendet. Dieser Vorgang kann für verschiedene Filter wiederholt werden, sodass deren Ergebnisse die sogenannten Feature Maps bilden (Abb. 2).

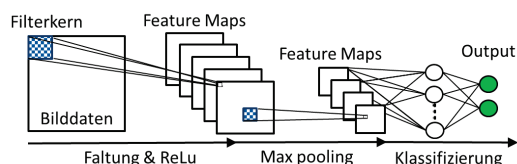


Abbildung 2: Prinzipieller Aufbau eines ConvNet

Nach jedem Faltungs-Layer folgt meist ein sogenannter Maxpooling-Layer [8]. Dieser verringert die Auflösung

der Bilddaten, wodurch abstraktere Merkmale extrahiert werden können. Dieser Vorgang aus Faltung und Maxpooling wiederholt sich bis zur eigentlichen Klassifikation, welche bei ConvNets sowohl durch ein Neuronales Netz als auch durch sogenannte Support Vector Machines erfolgen kann. Im Gegensatz zum MLP werden bei ConvNets nicht nur die Gewichte der Neuronen, sondern auch die Werte der angewendeten Filter durch Backpropagation gelernt. Diese Technik stellt derzeit den „State of the Art“ in verschiedenen Aufgabenbereichen (z.B. Computer-Vision) des maschinellen Lernens dar [8].

Anwendung auf Ultraschalldaten

Speziell bei Netzwerken mit mehreren Hidden-Layern, sogenannten Deep Networks, müssen während des Lernprozesses eine hohe Anzahl an Berechnungen der Gewichte durchgeführt werden. Daher ist der größte Flaschenhals dieser Methoden die zur Verfügung stehende Rechenleistung. Denn für große Werte von N , müssen bereits viele Eingangsneuronen festgelegt und damit ein großer Teil der Berechnungen während des Einlesen der Daten vom Input-Layer in den ersten Hidden-Layer angewendet werden. Der in diesem Beitrag verwendete Ultraschallsensor mit einer Abtastrate von 256 kHz, liefert Signale mit einer Frequenz zwischen 10–100 kHz. Um die daraus resultierende Datenmenge zu reduzieren, kann beispielsweise eine Vorverarbeitung durch eine Kurzzeit-Fouriertransformation (STFT) erfolgen. Ein Netzwerk könnte nun lernen, die auf diese Weise entstandene Feature-Matrix oder auflösungsreduzierte STFT-Bilddaten (Abb. 3) zu klassifizieren.

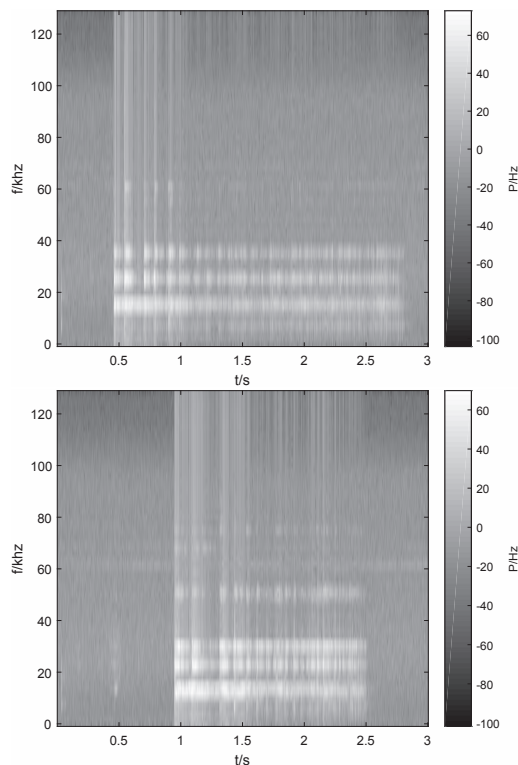


Abbildung 3: SFTF-Bilder einer 5 Cent Münze (oben), sowie einer 50 Cent Münze (unten)

Für die maschinelle Klassifikation von Teilentladungen gegenüber anderen akustischen Signalen, sind größere Datenmengen dieser Signale erforderlich. Da Teilentladungen nur selten im Praxisumfeld zu finden sind, wurden für diesen Beitrag ersatzweise sechs verschiedene Münzen auf variablen Untergründen fallen gelassen und deren Ultraschallsignale aufgezeichnet. Dadurch lässt sich ein leichter reproduzierbarer Datensatz erzeugen, um Möglichkeiten der Datenreduktion zu untersuchen. Die Unterschiede der verwendeten Münzen in den STFT-Bildern sind in Abb. 3 beispielhaft für eine 5- bzw. 50-Cent Münze dargestellt.

Zum Vergleich dazu sind in Abb. 4 STFT-Daten realer Teilentladung (oben) sowie der Durchschlagsfall (unten) abgebildet. Aus diesen Abbildungen wird ersichtlich, dass Teilentladungsprozesse eher von stochastischer Natur sind und somit im Gegensatz zum Durchschlagsfall und den Münzwürfen zunächst keine klar abgrenzbaren Muster erkennbar sind. Dennoch sind die Münzwürfe als Testfall verwendbar, um Methoden zur Datenreduktion untersuchen zu können.

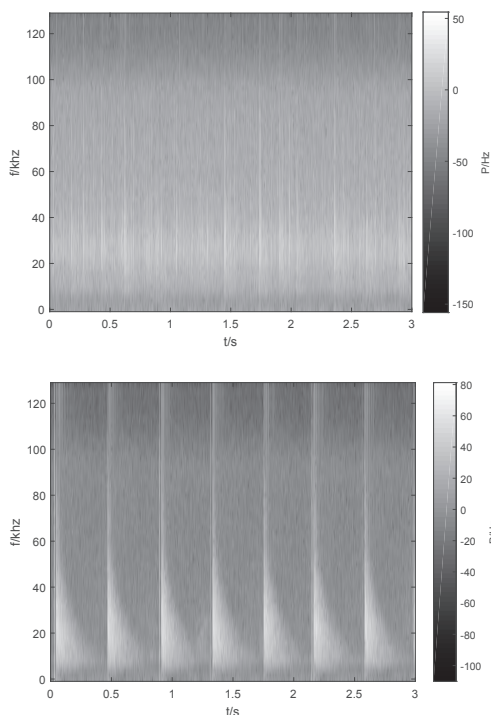


Abbildung 4: STFT-Bilder einer Teilentladung (oben), sowie von mehreren Durchschlägen in kurzen Zeitabständen (unten)

Mit Hilfe der in Abb. 3 und Abb. 4 gezeigten STFT-Bilder können die Gewichte eines Netzwerks gelernt und klassifiziert werden. Allerdings umfassen die entsprechenden Bilddaten trotz Auflösungsreduktion etwa 52.000 Pixel, was zum bereits beschriebenen Problem der für ein komplexes Neuronales Netzwerk zu geringen Rechenleistung führen kann.

Datenreduktion

In diesem Beitrag wird, dieses Problem durch einen auf CNN beruhendes Pegelkreuzungsverfahren zu lösen (Abb. 5).

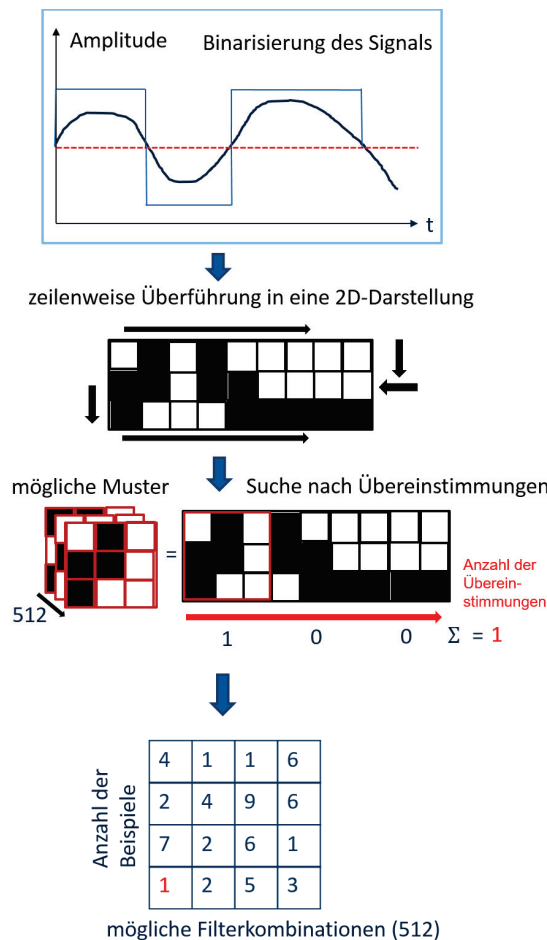


Abbildung 5: Schema des Pegelkreuzungsverfahrens

Dabei werden die Sensordaten mit deren Mittelwert verglichen und das Signal binarisiert. Das dadurch erhaltene binäre Signal kann nun mithilfe eines Filters mit beliebig vielen Elementen nach allen möglichen Kombinationen des Filters durchsucht werden. Dabei wird gezählt wie oft ein Muster innerhalb der Mustermatrix zu finden ist. Für einen Filter mit neun Elementen ergeben sich daraus 512 maximal mögliche Filterkombinationen m. Die jeweilige Häufigkeit des Auftretens wird für jeden Filter und jedes zur Verfügung stehende Trainingsbeispiel M notiert. Die so entstandene Trainingsmatrix $T \in \mathcal{R}^{m \times M}$ besitzt nun nur noch einen Bruchteil der Größe des STFT-Datensatzes und kann durch das neuronale Netzwerk klassifiziert werden.

Ergebnisse

Im Folgenden werden nun die Ergebnisse der Klassifikation dargelegt und diskutiert.

Insgesamt wurden die Ultraschallsignale sechs verschiedener Euro-Münzen aufgezeichnet, welche aus verschiedenen Höhen jeweils 100 mal auf unterschiedliche Un-

tergründe fallen gelassen wurden. Aus diesem Datenset wurde dem MLP bzw. ConvNet jeweils 80% der Daten während des Trainings- sowie 20% beim Testdurchlauf zugeführt. Daraus ergibt sich eine Chance von 16% die richtige Münze durch Raten zu bestimmen.

Die zur Klassifikation verwendeten MLP-Netzwerke bestehen aus zwei Hidden Layern mit jeweils 2500 Neuronen, einem Input- sowie einem Outputlayer mit sechs Neuronen analog zu Abb. 1. Als Aktivierungsfunktion wurde im Hidden Layer Tanh sowie Softmax im Outputlayer eingesetzt und zur Optimierungs diente das SGD mit Momentum. Die so jeweils 50 Epochen lang trainierten Netzwerke konnten mithilfe von Dropout sowie einer L2-Regularisierung analog zu [6] mit $\lambda = 10^{-3}$ reguliert werden.

Das verwendete ConvNet bestand aus einem Input Layer, zwei Faltungslayern, gefolgt vom jeweils einem Maxpooling layer sowie zwei vollständig verbundenen Neuronenlayern (Dense Layer) analog zum MLP. Weitere Details können Abb. 6 entnommen werden.

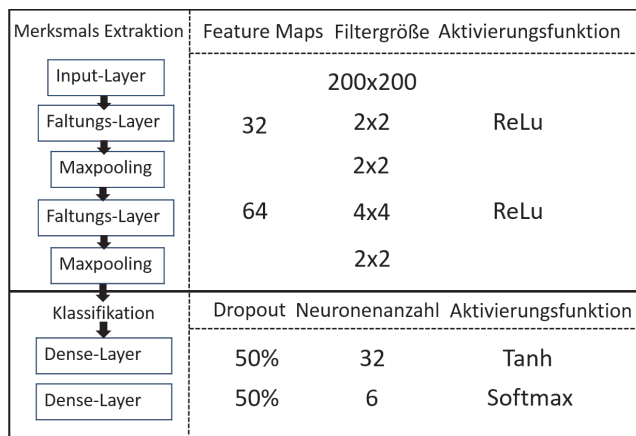


Abbildung 6: Struktur des ConvNets

Für die Berechnungen der Gewichte bzw. für die Angabe der Rechenzeit t_c standen ein Intel i3-Prozessor mit 3,7 GHz für das MLP sowie eine zusätzliche Nvidia GForce GTX 640 für die ConvNets zur Verfügung.

Die Klassifikationsergebnisse der einzelnen Netzwerke sind jeweils in Tabelle 1 zusammengefasst.

	Netzwerkart		
	MLP1	MLP2	ConvNet
Datensatz	STFT	Pegelkreuz.	STFT
t_c	6600 s	1100 s	300 s
θ	85%	85%	52%

Tabelle 1: Berechnung der verschiedenen Netzwerke

Daraus wird ersichtlich, dass sich die Rechenzeit t_c des MLP durch Einsatz des Pegelkreuzungsverfahrens um den Faktor sechs verringern ließ, während die Klassifizierungsrate θ auf beiden Datensätzen 85% beträgt. Für die Gewichte des ConvNets belief sich t_c durch die

Möglichkeit der Parallelisierung mit einer Grafikkarte auf lediglich 300 s. Jedoch konnten dabei nur 54% der Münzen korrekt klassifiziert werden. Da ähnliche Systeme auf simpleren Bilddatensätzen wie MNIST [9] bereits Klassifikationsraten von 99% und mehr erreichen [10], besteht weiterer Verbesserungsbedarf.

Zusammenfassung

In diesem Beitrag konnte gezeigt werden, dass eine Anwendung neuronaler Klassifikationsverfahren auf Ultraschalldaten unter der Voraussetzung ausreichend vorhandener Datenmengen grundsätzlich möglich ist. Des Weiteren konnte eine Möglichkeit zur Datenreduktion vorgestellt werden, wodurch die Rechenzeit für komplexe Modelle mit vielen Neuronen reduziert werden konnte.

Literatur

- [1] D. Borneburg et.al.: „Detektion, Lokalisierung und Echtzeitdarstellung von Korona-Entladungen an elektrischen Betriebsmitteln mittels Visualisierung der UV-Emission“, Fachtagung der Energietechnischen Gesellschaft im VDE, 2009
- [2] J. Müller et.al.: „Learning Deep CNN Structures“, CNNA, 2016
- [3] G. Hinton et.al.: „Reducing the Dimensionality of Data with Neural Networks“, Science vol. 313 , 2006
- [4] G. Hinton et.al.: „Deep neural networks for acoustic modeling in speech recognition“, IEEE Signal Processing Magazine vol. 29 , 2012
- [5] X. Glorot et.al.: „Understanding the difficulty of training deep feedforward neural networks“, AISTATS , 2010
- [6] A. NG: „Proceedings of the twenty-first international conference on Machine learning“, ICML , 2004
- [7] Nitish Srivastava et.al.: „Dropout: A Simple Way to Prevent Neural Networks from Overfitting“, Journal of Machine Learning Research , 2014
- [8] A. Krizhevsky et.al.: „ImageNet Classification with Deep Convolutional Neural Networks“, NIPS , 2012
- [9] Y. LeCun et.al.: „The MNIST database of handwritten digits“, 2010, Homepage URL: <http://yann.lecun.com/exdb/mnist/>
- [10] L. Wan et.al.: „Regularization of Neural Networks using DropConnect“, ICML, 2013