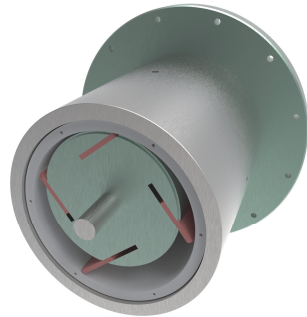


Systemgrößenmodellierung mit neuronalen Netzen – Ergänzung zur Baugruppenoptimierung

Kaczmarek, D.; Lohrengel, A.

Dieser Artikel beschreibt eine Möglichkeit, wie die Aussagekraft eines Simulationsmodells gesteigert werden kann, indem Prüfstands Ergebnisse zur Validierung genutzt werden. Dabei wird mit Hilfe von neuronalen Netzen ein Ersatzmodell eines Teilsystems geschaffen. Dieses neuronale Netz dient dann als zusätzliches Teilsystem, welches zur Optimierung einer Baugruppe dienen soll. Dabei wird eine kleine Ausführung zu den neuronalen Netzen als Einleitung zum Themenfeld dienen.



This article describes a way to increase the validity of a simulation model by using test bench results for validation. A replacement model of a subsystem is created with the help of neural networks. This neural network then serves as an additional subsystem, which is to serve for the optimization of an assembly. A small version of the neural networks will serve as an introduction to the topic.

Hintergrund

Im Rahmen eines Forschungsprojektes soll eine verfahrenstechnische Anlage zur Wassergewinnung aus der Luft entwickelt werden. Für dieses Verfahren wird ein Verdichter benötigt, welcher mit den konventionellen Typen nicht bedient werden kann. Daher ist ein Berechnungsmodell entstanden, um die Eigenschaften des neu entwickelten Verdichters abschätzen zu können. Dieses Berechnungsmodell besteht aus verschiedenen Teilmodulen, welche unterschiedliche physikalische

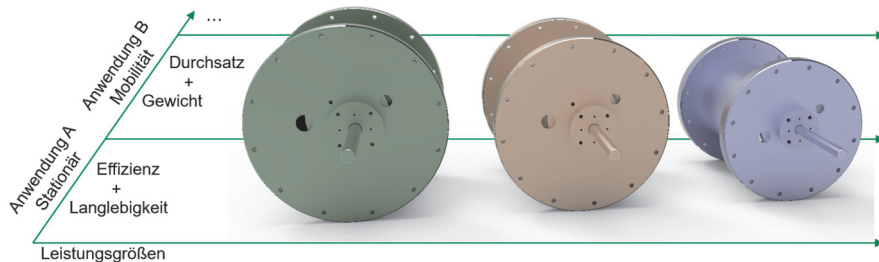


Abbildung 1: Anforderungsspezifische Baureihenentwicklung

Systemgrößen kalkulieren. Gesamtheitlich betrachtet werden so alle Teilsysteme des Verdichters kombiniert, um so die jeweiligen Baugrößen des Verdichters hinsichtlich Baugruppenänderungen simulieren zu können. Das so entwickelte Berechnungsmodell erlaubt mit Hilfe eines Optimierungsvorgangs eine Veränderung der Geometrie des Verdichters hinsichtlich der Maximierung von Anforderungserfüllungen. /1/

Aus der Abbildung 1 kann das derzeitige Grundprinzip abgeleitet werden. Mit Hilfe des Berechnungsansatzes ist es möglich, hinsichtlich spezieller Kundenspezifikationen, eine Baureihe mit bestimmten Leistungsabstufungen zu generieren.

Anhand praktischer Erfahrungen wurde festgestellt, dass das Berechnungsmodell die Systemgröße des Druckes überschätzt. Demnach gilt das modellierte thermodynamische System nicht für den Systemdruck und muss korrigiert werden.

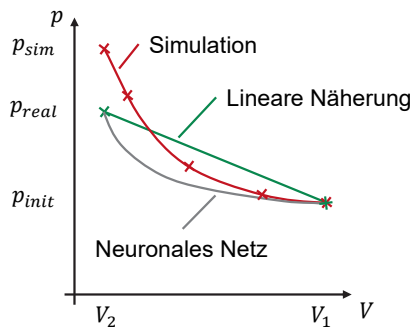


Abbildung 2: Modellierungsfehler

Bekannt aus dem Versuch sind der Anfangsdruck sowie Enddruck des Verdichters. Interessant an dem System ist allerdings das Verhalten des Druckes während des Verdichtungs Vorganges. Mit Hilfe der ausschließlichen Messung des Anfangs und Endzustandes des Gases können demnach keine validen Aussagen über das Verdichtungsverhalten innerhalb einer Kammer getroffen werden (Abbildung 2 – Lineare Näherung). Dafür wird das bisherige Berechnungsmodell herangezogen. Dieses Berechnungsmodell soll für eine Verhaltensnachbildung des Druckes genutzt werden (Abbildung 2 – Simulation). Es soll durch die Multiplikation einer Funktion wie in der Formel 1 gezeigt, eine Nachahmung des implementierten Druckverhaltens erreicht werden.

$$p(V_i) = \frac{p_{sim}}{p_{sim}(V_i)} \cdot p_{real} \quad 1$$

Mit Hilfe dieser Funktion ist es möglich, das Systemverhalten wie in Abbildung 2 als *Neuronales Netz* zu generieren. Der Indizier beschreibt das jeweilige Volumen in dem nummerischen Teilschritt *i*. Diese Funktion kann nun in diversen Dimensionen aufgestellt werden. Beispielsweise kann eine Druckveränderung hinsichtlich der

Verdichtungsfrequenz, Umgebungstemperatur oder aber auch von anderen Systemgrößen abhängig sein. Als Voraussetzung zur Einbindung der Systemgröße in das Gesamtverhalten, muss die Systemgröße als Messgröße erfasst werden.

Neuronale Netze

Neuronale Netze sind eine Entwicklung aus dem Jahr 1948, dort wurden von *Warren McCulloch* und *Walter Pitts* die ersten Entwürfe von neuronalen Netzen vorgestellt. Allerdings waren aus damaliger technischer Sicht die Anwendungsmöglichkeit stark begrenzt, da nicht ausreichend Rechenleistung sowie Daten zum Anlernen von diesen Netzen vorhanden waren. Erst wesentlich viel später, im Jahr 1982, wurde ein Verfahren von *Paul Werbos* an neuronalen Netzen vorgestellt, die mittels *Backpropagation lernen*. Mit dessen Hilfe ist es möglich, mehrschichtige neuronale Netze anlernen zu können.

Ein künstliches Neuron besteht allgemein aus den Eingängen, einer Gewichtung, einer Eingangsfunktion und der Aktivierungsfunktion. In der nachfolgenden Abbildung ist der Grundaufbau eines Neurons *Perceptron* abgebildet.

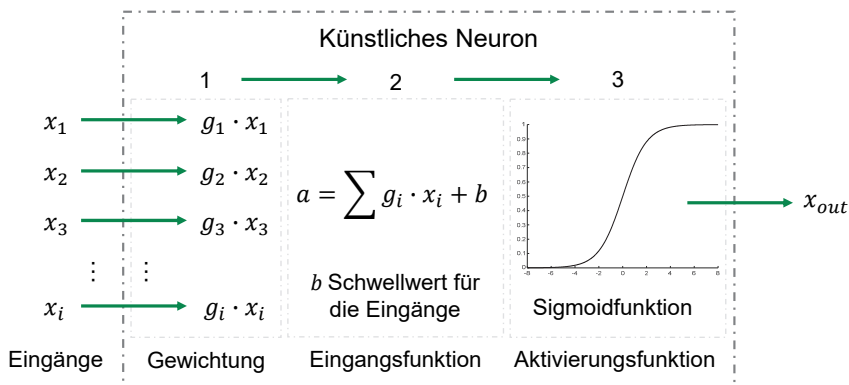


Abbildung 3: Grundaufbau eines einzelnen *Perceptron*

Die Abbildung 3 zeigt die künstliche Nachbildung des natürlichen Vorbildes in der Abstraktion von mathematischen Zusammenhängen. Wie gesehen werden kann, sind die dahinter zu verstehenden Mechanismen reine Matrizenrechnungen. Die *Eingänge* des Neurons entsprechen den Einflussgrößen eines nachzubildenden Systems, diese werden bei der Übergabe mit einer *Gewichtung* versehen. Diesen mathematischen Zusammenhang zwischen den *Eingängen* und der *Gewichtung* wird auch *Schur-Produkt* genannt. Werden diese jeweiligen *Eingänge* mit der *Gewichtung* und einem *Schwellwert* summiert, wird die *Aktivität* des Neurons ermittelt. Nun wird die *Aktivierungsfunktion* mit dem Wert aus der *Eingangsfunktion* ausgewertet, die hier hinterlegte Funktion entscheidet über das „*Feuern*“ des künstlichen Neurons./2/

Eine Verhaltensanpassung des *Perceptron Netzes* kann mit Hilfe der *Schwellwerte* und den jeweiligen *Gewichtungen* durchgeführt werden. Dabei wird zu Beginn ein Gütefunktional erstellt, dieses beschreibt die Genauigkeit der Modellvorhersage mit dem tatsächlichen System. Ein typisches und mögliches Gütefunktional kann die Form 2 haben./2/

$$\varepsilon(x) = Y_{\text{Sensor}}(x) - y_{\text{sim}}(x) \quad 1$$

$$E(\varepsilon) = \frac{1}{N} \sum_{i=1}^N \varepsilon(x)^2 \quad 2$$

Die Funktion 2 beschreibt die mittlere Gesamtabweichung des entwickelten Modells zum realen System und sollte idealerweise den Wert von Null annehmen. Diese Problematik ist ein mehrdimensionales Optimierungsproblem und kann beispielsweise über das *Backpropagation Verfahren* und einem Optimierungsalgorithmus gelöst werden. Das *Backpropagation Verfahren* wird im nächsten Abschnitt näher erläutert.

Neuronale Netze können in diversen Topologien zusammengestellt werden, um so unterschiedlichste Aufgaben zu erfüllen. Für den praktischen Anwendungsfall in der Systemmodellierung eines Verdichters, bietet sich das von *Frank Rosenblatt* erstmals entwickelte *Perceptron* an. Ein *feedforward Perceptron Netz* besitzt nach dem Anlernen die Fähigkeit, ein Verhaltensmuster wiederzugeben. Dies bedeutet im übertragenden Sinn, dass der Druckverlauf des Verdichters mit Hilfe eines *Perceptron Netzes* nachgeahmt werden kann. Dadurch, dass es sich hierbei um eine Verhaltenswiedergabe handelt, müssen Techniken des *supervised learning*

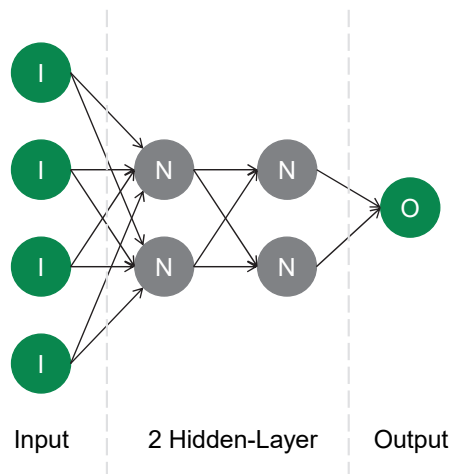


Abbildung 4: Feedforward Perceptron Netz

(überwachtes Lernen) angewandt werden. Die grundlegende Topologie des betrachteten Systems ist in Abbildung 4 dargestellt.

Backpropagation Verfahren

Das *Backpropagation Verfahren* gehört mit zu den gradientenbasierenden Optimierungsverfahren und soll in diesem Abschnitt nicht in allen rechnerischen Details wiedergegeben werden. Vielmehr dient dieser Abschnitt dazu, ein grundlegendes Verständnis des *Backpropagation Verfahrens* zu vermitteln.

Die grundlegende Idee des Verfahrens ist, das in Formel 2 gezeigte Gütefunktional nach den beschriebenen einflussnehmenden Parametern, den *Gewichtungen* und den *Schwellwerten*, gradientenbasierend zu optimieren. Dabei wird das neuronale Netz gesamtheitlich betrachtet und eine *globale* oder *lokale* Lösung gefunden. Wird in diesem Beispiel ein einzelnes *Perceptron* betrachtet, gilt folgendes Verhalten des Neurons:

$$x_{aus,1,n} = f_{aktiv,1}(\sum_i g_{1,j,n} \cdot x_{ein,1,j} + b_{1,n}) \quad 3$$

$$a_{1,n} = \sum_i g_{1,j,n} \cdot x_{ein,j} + b_{1,j,n} \quad 4$$

Wobei x_{in} dem *Eingangsvektor*, b dem *Schwellwert* eines Neurons, g dem *Gewichtungsvektor*, x_{out} dem *Ausgabewert* und f_{aktiv} der Auswertung der *Aktivierungsfunktion* entsprechen. Dabei bedeutet der Index i die Anzahl an *Eingängen* je Neuron und n die Nummer des Neurons in einem Layer. So kann ein Ausgabevektor für einen Layer (mehrere parallel angeordnete Neuronen) aufgebaut werden

$$\begin{bmatrix} a_{1,1} \\ \vdots \\ a_{1,n} \end{bmatrix} = \begin{bmatrix} g_{1,1,1} & \cdots & g_{1,j,1} \\ \vdots & \ddots & \vdots \\ g_{1,1,n} & \cdots & g_{1,j,n} \end{bmatrix} \cdot \begin{bmatrix} x_{ein,1,1} \\ \vdots \\ x_{ein,1,j} \end{bmatrix} + \begin{bmatrix} b_{1,1} \\ \vdots \\ b_{1,n} \end{bmatrix} \quad 5$$

$$\begin{bmatrix} x_{aus1,1} \\ \vdots \\ x_{aus1,n} \end{bmatrix} = \begin{bmatrix} f_{aktiv}(a_{1,1}) \\ \vdots \\ f_{aktiv}(a_{1,n}) \end{bmatrix} \quad 6$$

Für den zweiten Layer kann nach dem gleichen Prinzip der Ausgabevektor bestimmt werden, nur ist der Ausgabevektor des ersten Layers, der Eingabevektor des zweiten Layers. So kann zwischen den zwei Layer gesagt werden, dass diese im mathematischen Sinn verkettet sind und entsprechend bei einer Differenzierung die Kettenregel angewendet werden muss.

$$(u \circ v)'(x) = u'(v(x)) \cdot v'(x)$$

Das *Backpropagation Verfahren* nutzt die oben erwähnte Rechenregel aus, um das gesamtheitliche Netz zu betrachten. Die Ausgabe eines neuronalen Netzes kann über die *Gewichtungsmatrix* sowie dem *Schwellwertvektor* beeinflusst werden. Beide dieser Größen fließen in die *Aktivität* des Neurons ein und beeinflussen demnach das Gütefunktional. Wird ein dreischichtiges Netz betrachtet, kann dies wie folgt geschrieben werden

$$\frac{dE(a_3(a_2(a_1)))}{da_1} = \frac{dE(a_3(a_2(a_1)))}{da_3} \cdot \frac{da_3}{da_2} \cdot \frac{da_2}{da_1} \quad 7$$

Im Grunde bedeutet dies lediglich, die *Aktivität* des dritten Neurons besitzt eine direkte Abhängigkeit zu dem ersten Neuron, die über die zwischenliegenden Neuronen mit beeinflusst werden. Im Normalfall müsste noch eine Verkettung der *Aktivitäten* zu der *Aktivierungsfunktion* vorhanden sein, diese wurde allerdings weggelassen. Hier wird eine weitere Eigenschaft des *Backpropagation Verfahren* sichtbar, das Verfahren benötigt eine differenzierbare *Aktivierungsfunktion*. Daher werden zumeist eine Linearfunktion, Sigmoidfunktion oder andere differenzierbare Funktionen als *Aktivierungsfunktion* ausgewählt.

Damit die Einflussnahme der in Gleichung 7 gezeigten Verkettung deutlich wird, hier eine kurze Ausführung der Differenzierung.

$$\frac{da_2}{da_1} = \frac{d(G_2 \cdot x_{ein,2} + b_2)}{da_1} \quad 8$$

Dadurch das der Eingabevektor des zweiten Netzes der Ausgabevektor des ersten Netzes ist, kann diese Formulierung ebenso anders geschrieben werden.

$$\frac{da_2}{da_1} = \frac{d(G_2 \cdot x_{aus,1} + b_2)}{da_1} \quad 9$$

Nun kann beobachtet werden, dass diese Differenzierung zu

$$\frac{da_2}{da_1} = \frac{d(G_2 \cdot x_{aus,1})}{da_1} \quad 10$$

vereinfacht werden kann, da keine Abhängigkeit des *Gewichtungsfaktors* zwischen den *Layer* vorhanden ist. Entsprechend kann diese Formulierung auch für alle weiteren Differenzierungen in Gleichung 7 eingefügt werden und so die Rückführung auf den *Eingabevektor* des ersten *Layers* gezeigt werden. Aus diesem Grund ist der Name des Verfahrens entstanden.

Nummerisch werden letztendlich die Gradienten (Änderung des Gütefunktional in Abhängigkeit zu den Einflussparametern) der einzelnen *Layer* bestimmt. Dabei wird mit dem letzten *Layer* (Ausgabelayer) angefangen und so rückführend alle weiteren Gradienten der nachfolgenden *Layer* gebildet. So können für die *Schwellwerte* sowie *Gewichtungen* eine Abstiegsrichtung festgelegt werden, sodass das Gütefunktional

minimal wird. Das Initialnetz (zufällige *Gewichtungen* sowie *Schwellwerte*) und der Zusammenhang von Messwerten sowie Systemgrößen sind Voraussetzungen für das Verfahren.

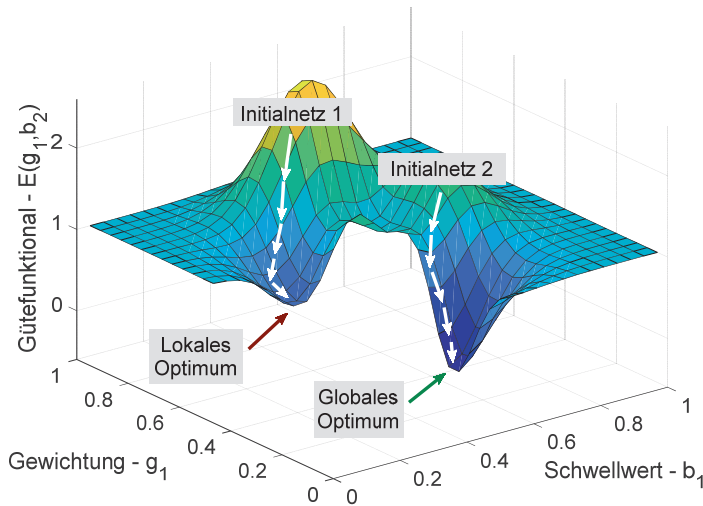


Abbildung 5: Generalisierungsproblematik /1/

Dadurch, dass es sich um einen gradientenbasierten Ansatz handelt, bedeutet dies, dass durch das Initialnetz die Endlösung bereits vorgegeben wird. Es werden demnach startwertabhängige Endnetze generiert, die als Beschreibung des Systemverhaltens dienen. Damit diese Startwertabhängigkeit besser Verstanden wird, eine kleine Visualisierung des Problems in Abbildung 5. Aus der Abbildung wird deutlich, dass je nach festgelegten Initialnetz, induziert durch den Gradienten, eine andere *Gewichtung* und anderer *Schwellwert* als eine mögliche Lösung zu finden sind. Letztendlich kann beispielweise bei dem Verhalten der angelegten Messwerte eine sehr genaue Übereinstimmung bei dem globalen Optimum zwischen Realsystem und Modell erreicht werden. Werden nun Systemparameter betrachtet, welche außerhalb des angelegten Bereichs hinausgehen, entstehen große Abweichungen des Modells. Diese Extrapolationsproblematik kann auch mit der Aufteilung der Messwerte in Trainingsdatensätzen und Validationsdatensätzen untersucht werden. Angelegnt wird das Netz mit den Trainingsdaten, die Güte kann dann mit den Validationsdatensätzen überprüft werden.

Ausgangsposition

Als kleines Beispiel soll in diesem Abschnitt eine eigens modifizierte Funktion mit Hilfe eines neuronalen Netzes nachmodelliert werden. Der genannte Verdichter arbeitet nach dem Verdrängerprinzip, was bedeutet, dass die Druckerzeugung des Verdichters maßgeblich von der Geometrie des Verdichters abhängig ist und im Idealfall keinen frequenzabhängigen Anteil der Druckdifferenz besitzt. In der Praxis hat sich allerdings ein frequenzabhängiger Anteil ausgebildet, welcher im bisher implementierten Berechnungsmodell /1/ nicht berücksichtigt wird. Diese Berücksichtigung soll mit Hilfe eines neuronalen Netzes erfolgen, um so die Genauigkeit des Modells zu verbessern. Zu Beginn eine kurze Vorstellung des betrachteten fiktiven Messsignals in Abbildung 6.

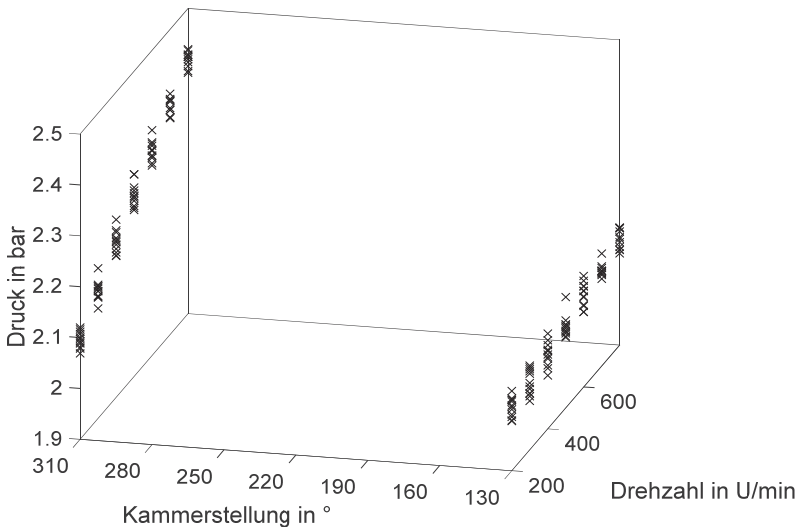


Abbildung 6: Fiktive Messergebnisse des Initial- und Enddruckes in Abhängigkeit der Frequenz

Die Abbildung 6 zeigt die jeweilig gemessenen Initial- sowie Enddrücke in Abhängigkeit der Drehzahl. Dabei wurden die jeweiligen Messpunkte mit einem Kreuz in dem Diagramm abgebildet. Bekannt sind im realen Fall ebenso ausschließlich die Initial- sowie Enddrücke, der Druck innerhalb einer Kammer während des Umlaufs ist letztendlich nicht bekannt. Um diese Druckentwicklung über den Umfang ebenso modellieren zu können, wird, nachdem die Mittelwerte der Messergebnisse gebildet worden sind, nun die Formel 1 angewendet. Das so erzielte Ergebnis ist in der nachfolgenden Abbildung 7 festgehalten.

Es wird nun versucht, mit Hilfe dieses Signales und eines neuronalen Netzes eine Systemnachbildung zu erstellen. Dazu wird die *Toolbox Neural Fitting* von *Matlab*®

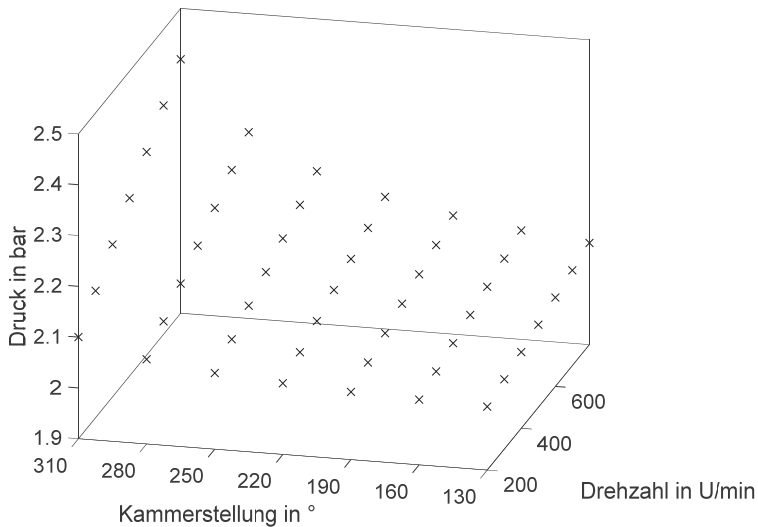


Abbildung 7: Trägerfunktion der Druckerzeugung in Abhängigkeit der Kammerstellung und der Drehzahl

genutzt. Diese Toolbox bietet eine breite Möglichkeit an verschiedenen *Performance* Untersuchungen sowie bereits inkludierte Vorbereitungsmaßnahmen an. Im Anschluss sollen dann verschiedene Varianten des neuronalen Netzes diskutiert werden. /4/

Ausführung

Die gezeigten Datensätze in dem vorangegangenen Kapitel sind mit Hilfe der *Neural Fitting Toolbox* in einer höheren Messgrößenanzahl als Inputparameter gewählt worden. Der nächste Schritt ist die Aufteilung des Datensatzes in *Trainingsdatensatz*, *Validationsdatensatz* und *Testdatensatz*. Das neuronale Netz wird mit dem *Trainingsdatensatz* angeleitet.

Mit Hilfe des *Validationsdatensatzes* wird dann die Generalisierung des neuronalen Netzes gemessen. Dazu wird mit Hilfe des erstellten Modells die Vorhersage zu dem *Validierungsdatensatz* überprüft und damit das Gütefunktional berechnet sowie als Vergleichsgröße genutzt. Generell kann gesagt werden, dass ein neuronales Netz durch zu intensives Anlernen des Trainingsdatensatzes zu einer Art Verinnerlichung des Datensatzes führen kann. Das bedeutet, dass das neuronale Netz eine exakte Nachbildung der Punktwolke ist, dieses Phänomen wird auch *Overfitting* genannt. So wird das Gütefunktional zwar bei dem Trainingsatz ein Minimum erreichen, jedoch würde eine Überprüfung mit dem Validierungssatz zu einer wesentlich

größeren Abweichung vom eigentlichen Modell führen. Daher werden der *Trainingsdatensatz* und *Validationsdatensatz* gleichzeitig zum Anlernen des Netzes überprüft. Ein gutes Indiz zur Generalisierung des Modells wird dann erreicht, wenn nach Möglichkeit das Gütefunktional von dem *Trainingsdatensatz* sowie dem *Validationsdatensatz* nach den jeweiligen Optimierungsepochen gleichzeitig ein Minimum erreichen.

Der *Testdatensatz* besitzt die Aufgabe, das Modell auf die Performance zu überprüfen, dabei wird dieser Datensatz ausgewertet und so auch geschaut, ob alle nötigen Informationen in dem erstellten Modell vorhanden sind. Es kann beispielsweise sein, dass gewisse Informationen des *Testdatensatzes* nicht in dem *Trainingsdatensatz* enthalten sind und entsprechend ein Modell erstellt worden ist, welches nicht dem gewollten entspricht. Im Grunde kann so ebenfalls das Gütefunktional vom *Testdatensatz* sowie vom *Validierungsdatensatz* überprüft werden. Liegen die jeweiligen Gütefunktionalwerte wesentlich auseinander, ist der Informationsgehalt der Trainingsdaten nicht ausreichend. Entweder kann dann eine neue Zuordnung zu den jeweiligen Datensätzen eine Besserung erreicht, oder es müssen gänzlich bessere Informationsdaten verwendet werden. Die Toolbox *Neural Fitting* bietet eine automatisch zufällige Zuordnung dieser Datensätze an. Hier kann ein Prozentsatz des Datensatzes ausgewählt und entsprechend zum Anlernen sowie zur Performanceüberprüfung genutzt werden.

Die *Neural Fitting Toolbox* erlaubt das Erzeugen von einem zweischichtigen neuronalen Netz, welches ein sehr flaches Netz repräsentiert. Eine Möglichkeit in der genannten Toolbox das Verhalten des neuronalen Netzes anzupassen, falls unbefriedigende Ergebnisse generiert werden, ist unter anderem die Anzahl an Neuronen pro *Layer*. Die Standardwerte liegen hier bei zehn Neuronen, welches nach Belieben abgeändert werden können. Je mehr Neuronen ein Layer besitzt, desto komplexere Funktionen können abgebildet werden. Allerdings existieren keine Regeln für die Anzahl nötiger Neuronen oder Hidden Layers eines neuronalen Netzes. Dieser Prozess ist ein iteratives Herangehen, indem versucht wird, mit einer kleinen Anzahl an Neuronen eine gute Problemmodellierung zu erzeugen. Es wird demnach mit einer kleinen Anzahl an Neuronen angefangen und iterativ die Performance des Netzes bei einer höheren Anzahl von Neuronen bewertet.¹⁴

Es wird ein neuronales Netz mit einem Neuron im Hidden Layer erzeugt und mit Hilfe des konjugierten Gradientenverfahrens die jeweiligen *Gewichte* und *Schwellwerte* angepasst. Dabei wird die Performance des Netzes in der nachfolgenden Abbildung 8 als mittlerer Abweichungsfehler von den drei bereits erwähnten Datensätzen vorgestellt. Dieser Abweichungsfehler beschreibt den in Formel 2 beschriebenen Fehler.

In der dargestellten Abbildung wird gezeigt wie sich der Abweichungsfehler über die einzelnen Optimierungsepochen der *Gewichtungen* und *Schwellwerte* verändert. Der Abbruch folgt in der 43-igsten Epoche, dort wird ein Minimum des Validierungsdatensatzes gefunden. Alle weiteren Epochen führten nicht zu einer Verbesserung des Modells. Wenn der mittlere Abweichungsfehler des

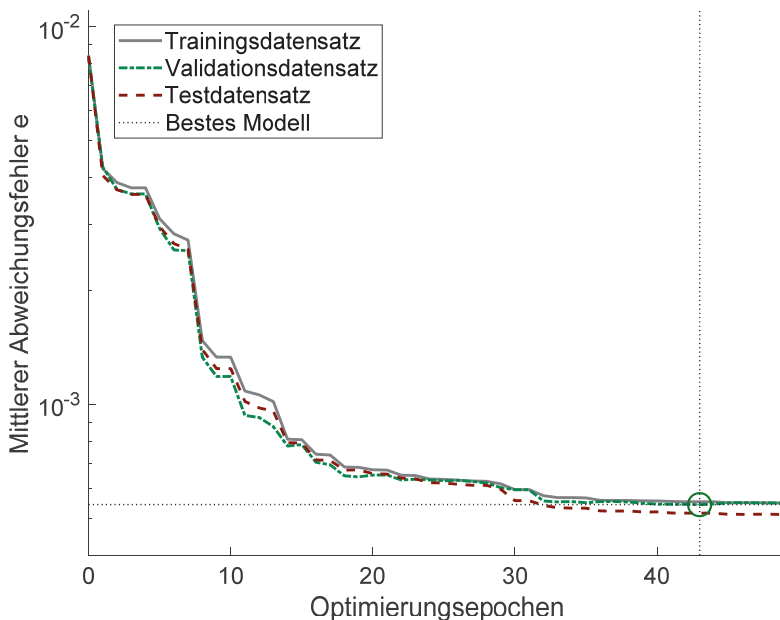


Abbildung 8: Mittlerer Abweichungsfehler ($\bar{\epsilon}_{43} = 5,44 \cdot 10^{-4}$) über die Optimierungsepochen

Validierungsdatensatzes anfängt zu steigen, ist dies ein Indiz dafür, dass ein *Overfitting* des Trainingsdatensatzes stattfindet. Daher wird der Berechnungsablauf, wie auch bereits erwähnt, bei einem Minimum des mittleren Abweichungsfehlers vom Validierungsdatensatz und Trainingsdatensatz abgebrochen.

Die lineare Regression des neuronalen Netzes zu den gemessenen Ergebnissen verdeutlicht die Verhaltensanalyse des modellierten Systems. Mit Hilfe dieser Bewertung, kann geschaut werden, ob das Modell das gewünschte Verhalten erzeugt. In Abbildung 9 wird die lineare Regression zwischen den fiktiven Messergebnissen und der Ausgabe des neuronalen Netzes dargestellt. Die jeweiligen Punkte (Datenpunkte) bilden immer ein Vektorpaar zwischen der Frequenz und der Kammerstellung ab. Zusätzlich bilden die Werte auf der Ordinate und Abszisse entsprechend die Druckerwartung bei dem modellierten Netz sowie dem Messergebnis in Abhängigkeit dieses Vektorpaars ab. Die eingezeichnete Gerade der Netzausgabe zeigt die Modellausgabe im Verhältnis zu den tatsächlichen Werten. Das bedeutet letztendlich, dass das erzeugte Modell in allen

vier Darstellungen der Regressionen, das tatsächliche Modell im unteren Druckbereich überschätzt und im höheren Druckbereich unterschätzt.

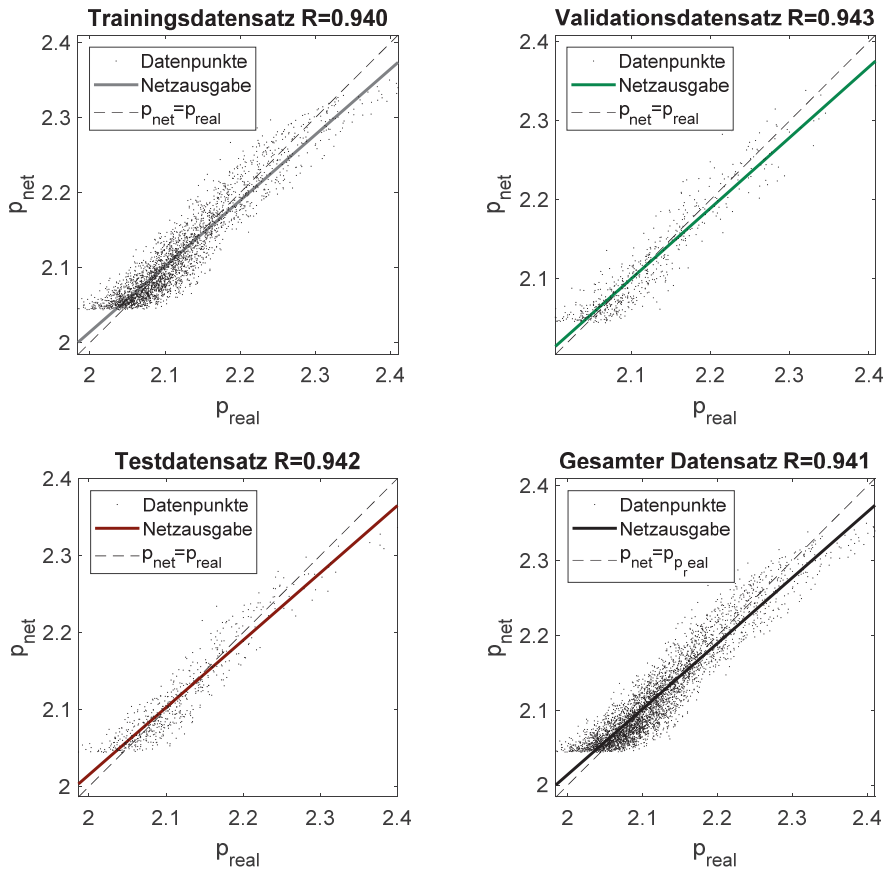


Abbildung 9: Lineare Regression zwischen der Modellausgabe und den fiktiven Messwerten

In diesem Beispiel wurde bereits mit einem Neuron im Layer eine sehr gute Übereinstimmung zwischen dem neuronalen Netz und den fiktiven Ergebnissen erzeugt. Das Endergebnis dieser Modellerzeugung ist nachfolgend in Abbildung 10 gezeigt.

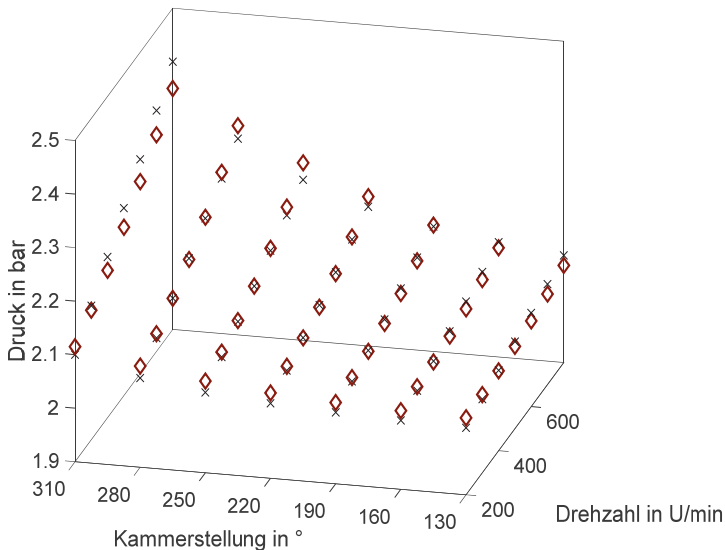


Abbildung 10: Ausgabe des neuronalen Netzes **rot**, fiktive Messdaten **schwarz**

Dieses Ergebnis soll nun mit verbessert werden, indem die Neuronenanzahl des *Hidden Layers* variiert wird. Bei dieser Methode wurde die Anzahl der Neuronen schrittweise erhöht und der mittlere Abweichungsfehler sowie die Regressionsanalyse betrachtet. Zudem wurde das Antrainieren einer Netzkonfiguration nicht nur einmal durchgeführt, sondern mehrere Male hintereinander. Der Grund für dieses Vorgehen ist in der Abbildung 5 gezeigt. Zusätzlich wird bei jedem anlernen des Netzes die drei zufälligen Datensätze neu zusammengestellt, so wird zum einen das multidimensionale Optimierungsproblem mit anderen Startwerten behandelt und der Informationsgehalt der jeweiligen Datensätze verschieden belegt. Das beste Ergebnis dieser Rechenvorgänge wurde dann tabellarisch pro Netzkonfiguration aufgeschrieben und so gesamtheitlich betrachtet. Das Ergebnis dieser Untersuchung zeigt folgende Tabelle 1.

Anhand der Tabelle kann gesehen werden, dass noch eine erhebliche Verbesserung der Performance des neuronalen Netzes mit der Erhöhung von einem Neuron auf zwei Neuronen im *Hidden Layer* erreicht werden kann. Bei einer weiteren Erhöhung wird keine wesentliche Verbesserung mehr erreicht. Auch der Regressionskoeffizient wird bei einer Zunahme von mehr als zwei Neuronen nicht weiter verbessert. Bemerkenswert ist die Abnahme der Optimierungsschritte bei Zunahme der Neuronen. Dies ist wohl mit einer Zunahme der jeweiligen Freiheitsgrade zu begründen, welche lokale Extremwertstellen begünstigen. Das Ergebnis des Netzes mit zwei Neuronen ist in der nächsten Abbildung zu finden.

Tabelle 1: Verbesserungsrechnungen mit verschiedenen Anzahlen an Neuronen in dem *Hidden Layer*

Anzahl Neuronen	$\bar{\varepsilon}$ Abweichungsfehler 10^{-4}	Regressionskoeffizient Gesamtdatensatz R	Epoche
1	5,44	0,94	43
2	3,87 (-29%)	0,96 (+2,1%)	12
3	3,55 (-35%)	0,96 (+2,1%)	30
4	3,59 (-34%)	0,96 (+2,1%)	17
10	3,79 (-30%)	0,96 (+2,1%)	8
20	3,78 (-31%)	0,96 (+2,1%)	9

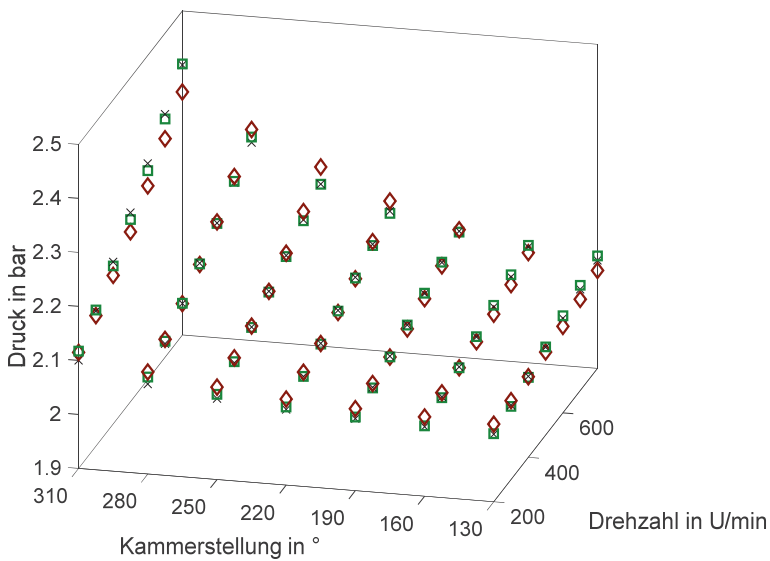


Abbildung 11: Gegenüberstellung der alten **rot** und verbesserten Modellierung **grün** und den fiktiven Messdaten **schwarz**

Geplant ist eine Übernahme einer solchen Funktion, um den Druckverlauf über den Umfang des Verdichters zu modellieren. So soll zum einen der Druck des modellierten Systems korrigiert und zum anderen ebenso das dynamische Verhalten des Verdichters genauer ermittelt werden. Die Anzahl der zu berücksichtigenden

Parameter belaufen sich im direkten Vergleich zu dem Beispiel auf fünf verschiedene Parameter.

Fazit

Zur Nutzung dieser Vorgehensweise für das genannte Ziel, den Druckverlauf über den Umfang des Verdichters realitätsnäher zu beschreiben, kann ein solches Verfahren grundsätzlich angewendet werden. Es existieren bereits ähnliche und erfolgreiche Anwendungen im Bereich der Kraftfahrzeuge /5/. Der nächste Schritt ist die Weiterentwicklung des genannten Vorgehens. Dabei soll es ermöglicht werden über eine Baureihe, eine Extrapolation des Druckes in realitätsnaher Form zu schaffen. Hierbei sollen mit Hilfe von beispielsweise der Reynoldszahl, im Ansaugtrakt sowie Ausgabebereich und weiteren Messungen an modifizierten Produkten, eine Extrapolation über untersuchte Grenzen hinaus ermöglicht werden.

Literatur

- /1/ Kaczmarek D., Lohrengel A. : *Automatisierte Baugruppenoptimierung hinsichtlich Produktanforderungen am Beispiel eines Drehschieberverdichters* – 17. Gemeinsames Kolloquium Konstruktionstechnik, 2019 Aachen
- /2/ Cowan, Jack D.: Discussion: McCulloch-Pitts and related neural nets from 1943 to 1989. In: *Bulletin of Mathematical Biology* Bd. 52 (1990), Nr. 1–2, S. 73–97
- /3/ Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. In: *Psychological Review* Bd. 65 (1958), Nr. 6, S. 386–408
- /4/ *Beale, Mark Hudson, Martin T Hagan, und Howard B Demuth: „Neural Network Toolbox™ 7“, 2010*
- /5/ *Hafner, Michael ; Schüler, Matthias ; Isermann, Rolf: Einsatz schneller neuronaler Netze zur modellbasierten Optimierung von Verbrennungsmotoren. In: MTZ - Motortechnische Zeitschrift Bd. 61 (2000), Nr. 11, S. 798–805*