

Qualitätsüberwachung beim Gesenkschmieden durch den Einsatz Künstlicher Neuronaler Netze

Bernd-Arno Behrens, Klaus-Georg Kosch*, Hagen Schmidt

Institut für Umformtechnik und Umformmaschinen, Leibniz Universität Hannover

In der vorliegenden Arbeit wird der Einsatz Künstlicher Neuronaler Netze (KNN) zur Qualitätsüberwachung von Schmiedeprozessen untersucht. Ziel ist die Schaffung eines Qualitätsüberwachungssystems, mit dem direkt nach der Umformung Aussagen über die erreichte Produktqualität getroffen werden können. Hierzu sollen Künstliche Neuronale Netze eingesetzt werden, die selbständig aus bereits vorliegenden Messdaten die notwendigen mathematischen Zusammenhänge bilden. Nach erfolgter Trainingsphase ist es möglich, anhand aktueller Messdaten unmittelbar nach der Fertigung die Schmiedeteilqualität im Prozess zu bewerten.

Schlüsselwörter: Qualitätsüberwachung, Künstliche Neuronale Netze, Gesenkschmieden, Produktionstechnik, Modellbildung

Einleitung

Im Bereich der Warmmassivumformung erfolgt die Qualitätsbestimmung im Allgemeinen durch mechanische Vermessung der Bauteile im Anschluss an den Umformvorgang. Diese Vermessung ist erst nach einer Abkühlphase möglich. Die Abkühlzeit ist abhängig von der Wärmekapazität des Werkstoffs, der Bauteilgröße und seiner Geometrie. Im Zeitraum von der Umformung bis zur Qualitätsbestimmung werden viele Teile ähnlicher Qualität erzeugt. Werden die Prozessparameter nicht eingehalten, so werden bis zur Messung und dem Detektieren des Fehlers ausschließlich Ausschussteile produziert. Ziel des vorgestellten Projektes ist eine Klassifizierung der Bauteilqualitäten direkt nach dem Umformvorgang mit Hilfe eines Künstlichen Neuronalen Netzwerks. So können Fehler direkt nach der Umformung erkannt werden. Der Prozess kann somit schnellstmöglich angepasst werden, so dass die folgenden Bauteile eine gute Qualität aufweisen. Im Allgemeinen werden KNN genutzt, um Prozessdaten zu klassifizieren. Die zugrundeliegenden Regeln lernt das KNN selbständig anhand dieser Daten. Nach dem Lernprozess ist die Software in der Lage, bereits erlernte Muster sowie ähnliche Muster richtig zu

klassifizieren. Eine Zuführung von Fachwissen ist dazu nicht erforderlich [1]. Dies verkürzt die Entwicklungszeit und schafft eine hohe Flexibilität in der Neustrukturierung von Herstellungsprozessen.

Aufbau eines FeedForward-Netzes

Ein neuronales Netz ist ein informationsverarbeitendes System, das in Anlehnung an das biologische Vorbild aus miteinander verbundenen Elementen besteht. Diese Elemente werden als Neuronen, Einheiten, Zellen, Units oder Knoten bezeichnet und tauschen Information in Form von Aktivierung der Neuronen über gerichtete Verbindungen aus. Dabei wird zwischen verschiedenen Arten von Neuronen unterschieden. Als Eingabeneuronen (input units) werden Einheiten bezeichnet, die Signale aufnehmen und die Eingabe an das Netz weiterleiten. Über die Ausgabeneuronen (output units) wird die Ausgabe nach außen geleitet. Verdeckte Neuronen (hidden units), befinden sich zwischen Eingabe- und Ausgabeneuronen und sind für die eigentliche Informationsverarbeitung innerhalb des neuronalen Netzes verantwortlich [2].

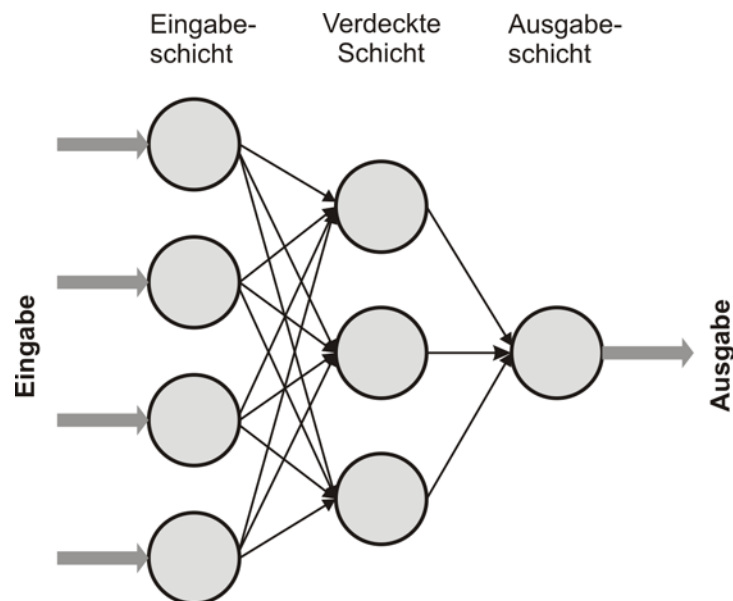


Abbildung 1: Schematische Darstellung eines FeedForward-Netzes [3]

In Abbildung 1 ist der prinzipielle Aufbau eines KNN dargestellt. Die übereinander angeordneten Neuronen bilden eine Schicht bzw. einen Layer. Ein neuronales Netz besitzt in der Regel nur eine Eingabe- und eine Ausgabeschicht, wohingegen es mehrere verdeckte Schichten enthalten kann. Die einzelnen Neuronen sind durch

Kanten miteinander verbunden. Die Stärke dieser Verbindungen wird durch einen Gewichtungsfaktor festgelegt [4].

Das Gewicht der Verbindung von Neuron i zu Neuron j wird als w_{ij} bezeichnet. Alle Gewichte eines Netzes können zur Gewichtsmatrix W zusammengefasst werden. FeedForward-Netze enthalten ausschließlich Verbindungen in eine Richtung, von den Eingabeneuronen in Richtung der Ausgabeneuronen. Dabei kann es sich um ebenenweise verbundene FeedForward-Netze handeln, bei denen einzig Verbindungen von einer Schicht zur jeweils nächsten existieren. Gibt es in einem Netz auch solche Verbindungen, die ganze Ebenen überspringen, sogenannte shortcut-connections, so liegt ein allgemeines FeedForward-Netz vor [2].

Ein künstliches Neuron ist eine Einheit, die Zahlenwerte entgegen nimmt, verarbeitet und so eine Ausgabe erzeugt, die an nachgelagerte Neuronen oder als Netzausgabe weitergegeben wird. In Abbildung 2 ist die prinzipielle Arbeitsweise eines Neurons dargestellt.

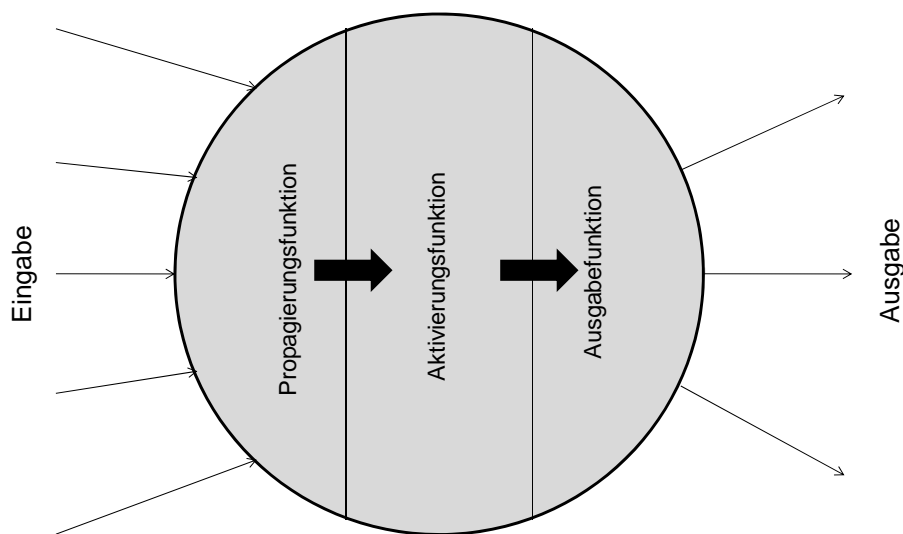


Abbildung 2: Künstliches Neuron [5]

Die Transformation der Eingabe eines Neurons in dessen Ausgabe erfolgt mit Hilfe von Propagierungs-, Aktivierungs- und Ausgabefunktionen. Dabei erfolgt die Ermittlung der Netzeingabe eines Neurons mittels der Propagierungsfunktion. Die Aktivierungsfunktion dient der Verarbeitung dieser Eingabe, während die eigentliche Ausgabe durch die Ausgabefunktion bestimmt wird. Ein Neuron empfängt meist nicht ausschließlich Werte eines sondern mehrerer vorgelagerter Neuronen. Daher müssen zunächst die einzelnen Eingabewerte der Neuronen bestimmt werden. Diese

werden durch Ausgabe der sendenden Einheit und dem Gewicht der Verbindung zwischen den beiden Neuronen beeinflusst. Meist sind diese beiden Werte multiplikativ miteinander verknüpft [4], so dass gilt:

$$input_{ij} = a_i \cdot w_{ij} \quad (Gl. 1)$$

$input_{ij}$	Eingabewert des sendenden Neurons i für Neuron j
a_i	Aktivierungszustand von Neuron i
w_{ij}	Gewichtung der Verbindung zwischen Neuron i und Neuron j
i	sendendes Neuron
j	empfangendes Neuron

Mit steigendem Ausgabebetrag der sendenden Einheit i und mit höherem Betrag des Verbindungsgewichts zwischen den Einheiten i und j erhöht sich der Einfluss auf die empfangende Einheit j. Folglich ist, falls einer dieser Terme gleich null ist, kein Einfluss vorhanden. Wie im Folgenden die Netzeingabe aus diesen einzelnen Inputwerten ermittelt wird, bestimmt die Propagierungsfunktion. Meist wird als Propagierungsfunktion die Linearkombination [4], d.h. die gewichtete Summe verwendet:

$$netinput_j = \sum_i input_{ij} = \sum_i a_i \cdot w_{ij} \quad (Gl. 2)$$

$netinput_j$ Gesamtfremderregung von Neuron j

Der Aktivierungszustand eines Neurons wird mit Hilfe der Aktivierungsfunktion aus dem vorherigen Aktivierungszustand sowie der Netzeingabe berechnet, wobei der vorherige Aktivierungszustand häufig nicht in die Berechnung mit eingeht [6]. Normalerweise besitzen alle Neuronen derselben Schicht die gleiche Aktivierungsfunktion [7].

Programmierung und Aufbau des Künstlichen Neuronales Netzes

Für den Aufbau des Künstlichen Neuronales Netzes wurde eine eigene Software programmiert. Die Software wurde mit dem Programm MATLAB erstellt und übernimmt die Aufgabe, die bei den Schmiedeversuchen aufgezeichneten Daten zu verarbeiten. Die Daten werden eingelesen und vorverarbeitet. Anschließend werden Kennwerte ermittelt und das KNN trainiert. Alle Ergebnisse werden nach dem Training zur Weiterverarbeitung gespeichert.

Die grafische Benutzeroberfläche (GUI) ermöglicht den Zugang zu allen Programmfunktionen des Hauptprogramms. Dabei sind die vier Funktionsbereiche in vier Abschnitte umgesetzt (Abbildung 5).

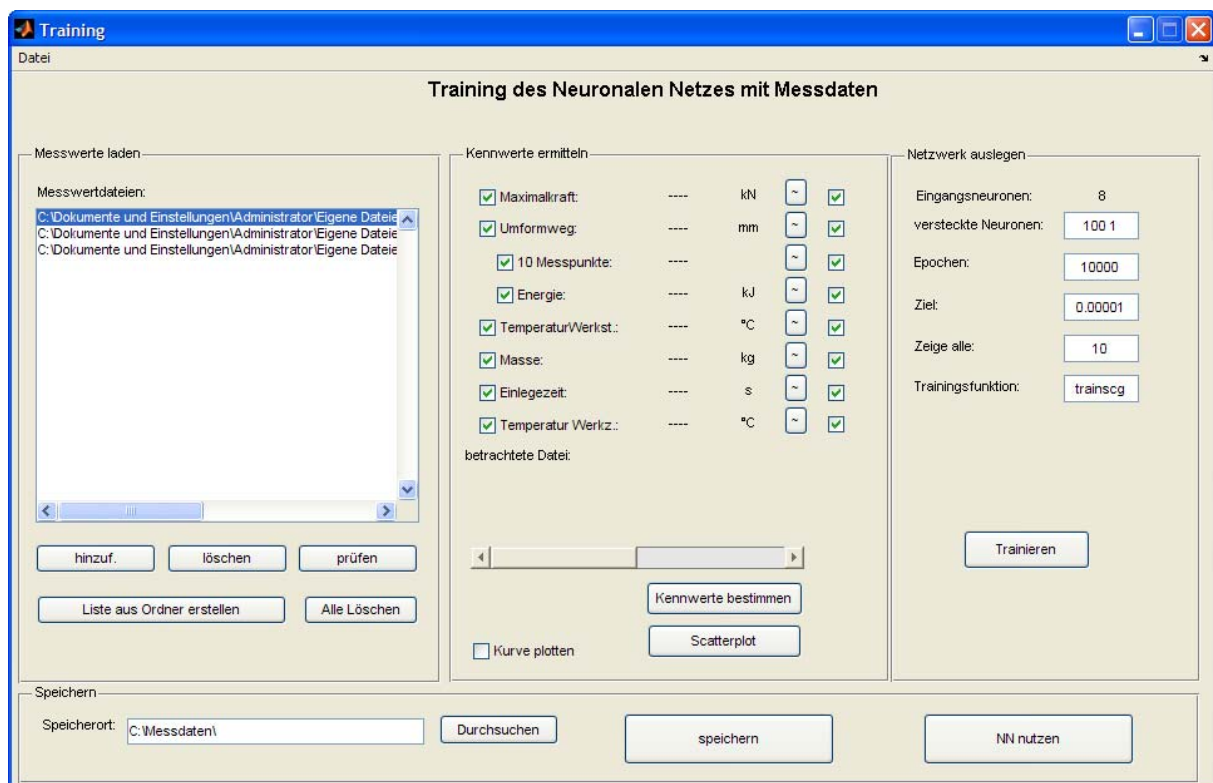


Abbildung 5: Grafische Benutzeroberfläche des Hauptprogramms

Der erste Abschnitt „Messwerte laden“ ermöglicht die Zusammenstellung von Dateilisten. Im zweiten Abschnitt werden vom Benutzer die Einstellungen zur Kennwertberechnung ausgewählt. Zu den einzelnen Kennwerten können Hilfsdiagramme angezeigt werden. Anhand der Hilfsdiagramme ist eine schnelle Fehlerdiagnose möglich. Im dritten Abschnitt wird das Netzwerk konfiguriert. Nach der Kennwertberechnung wird in diesem Abschnitt das Training des Netzwerks gestartet. Im vierten Abschnitt kann der Benutzer die berechneten Daten und die Einstellungen der GUI

speichern. Dies ermöglicht ein späteres Fortsetzen der Berechnungen und eine Weiterverwendung der Daten.

Im Bereich „Kennwerte ermitteln“ kann der Nutzer Einfluss auf die Berechnung der Schlüsselpositionen nehmen. Je nach Versuchsanordnung können Kennwerte komplett von der Weiterverarbeitung ausgeschlossen werden. Während der Berechnung öffnet das Programm alle Dateien der Dateiliste nacheinander und liest die Daten in den Speicher. Die Kennwerte werden ermittelt und jeweils neben der Kennwertbezeichnung angezeigt. Anschließend können die Ergebnisse der einzelnen Dateien aufgerufen werden. Zu dem Dateinamen werden die passenden Kennwerte in den Textfeldern darüber angezeigt. Treten Fehler während der Kennwertberechnung auf, können diese über die Anzeige der Diagramme lokalisiert werden. Die Diagramme visualisieren je nach Typ den jeweiligen Kennwert für die gewählte Datei oder alle Dateien im Überblick.

Im Bereich „Netzwerk auslegen“ der GUI werden die Parameter für die Erstellung und das Training des Netzwerks festgelegt ([Abbildung 6](#)).

Eingangsneuronen:	8
versteckte Neuronen:	100 1
Epochen:	10000
Ziel:	0.00001
Zeige alle:	10
Trainingsfunktion:	trainscg

Trainieren

Abbildung 6: Auslegung des Neuronalen Netzwerks

Das Programm erstellt ein FeedForward-Netzwerk mit den angegebenen Parametern. Die Anzahl der Eingangsneuronen wird anhand der eingehenden Datenkanäle bestimmt. Die Auswahl kann durch Checkboxen im Feld „Kennwerte

ermitteln“ getroffen werden. Im Feld „versteckte Neuronen“ wird angegeben, dass ein Layer mit (hier: 100) versteckten Neuronen und (hier: 1) Ausgangsneuronen angelegt wird. Die Anzahl der Epochen (Trainingszyklen) wird auf maximal 10000 festgelegt. Alle 10 Epochen wird beim Training ein Diagramm aktualisiert, welches den Trainingsverlauf visualisiert. In Abbildung 7 ist das Netzwerk mit 8 Eingangsneuronen schematisch dargestellt.

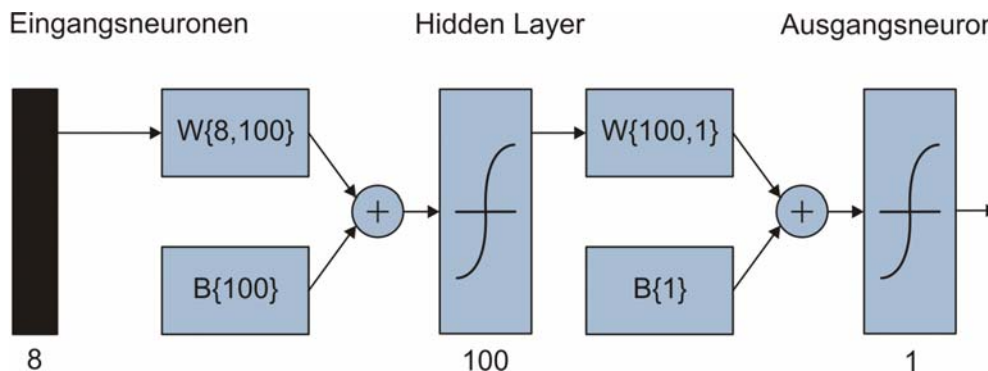


Abbildung 7: Schematische Netzwerkdarstellung

Diese wirken über 8×100 Verbindungsgewichte $W\{8,100\}$ auf die 100 Neuronen in der versteckten Ebene (Hidden Layer). In der Box unter den Verbindungsgewichten befinden sich 100 Schwellwerte $B\{100\}$, mit denen das Signal zusätzlich verrechnet wird. Im Hidden Layer bestimmt eine Aktivierungsfunktion, welche Neuronen aktiviert werden. Die Neuronen geben ein Signal über 100 Verbindungsgewichte $W\{100,1\}$ an ein einzelnes Ausgangsneuron weiter. Nach Addition eines einzelnen Verbindungsgewichts $B\{1\}$ mit den Ausgabewerten der Neuronen im Hidden Layer, wird im Ausgangsneuron die Netzantwort berechnet. Die binäre Ausgabe des Ausgangsneurons gibt an, ob die Werte an den Eingangsneuronen als gut oder schlecht klassifiziert werden. Durch die Angabe der Epochen in der GUI legt der Benutzer fest, wie oft die Daten vom Netzwerk trainiert werden. Mittels verschiedener Lernalgorithmen werden die Korrekturwerte für die Gewichte und die Schwellwerte bestimmt. Je häufiger die Daten trainiert werden, desto genauer kann das Netzwerk den Prozess abbilden. Von Epoche zu Epoche steigt die Qualität des Netzwerks.

Durch übermäßiges Training der gleichen Daten kann ein „Auswendiglernen“ der Daten auftreten. Um dies zu verhindern, wird das Netzwerk nach jeder Epoche mit noch nicht trainierten Daten geprüft. Wird bei diesem Test ein Übertrainieren festgestellt, so wird das Training gestoppt. Nach jeder Epoche wird zusätzlich die

Qualität des Netzwerks überprüft. Dazu werden ebenfalls noch nicht trainierte Daten eingesetzt. Vor dem Training werden die Daten in drei Gruppen nach dem Zufallsprinzip eingeteilt. Von den Daten werden 60 % als Trainingsdaten, 20% zur Prüfung der Generalisierungsfähigkeit und weitere 20% zur Validierung verwendet. Dabei wird der Performance-Wert über die Anzahl der Trainingsepochen aufgetragen. Die waagerechte Linie markiert das Trainingsziel. Die abfallende Kurve stellt den Fehlerwert des Netzwerks, der von Durchlauf (Epoche) zu Durchlauf sinkt, dar. Unterschreitet dieser das angestrebte Trainingsziel, so wird die Lernprozedur abgebrochen. Das trainierte Netzwerk ist nun in der Lage, die Daten mit der geforderten Genauigkeit zu berechnen. Nach Beenden des Lernprozesses liegt das trainierte Netzwerk im Speicher vor und kann zusammen mit den anderen Daten gespeichert werden.

Schmiedeversuche für das Training des Künstlichen Neuronalen Netzes

Für die Gewinnung von variablen Eingangsdaten für das Training des Künstlichen Neuronalen Netzes ist ein flexibler Prozess mit unterschiedlichen Einflussmöglichkeiten notwendig. Die experimentellen Versuche wurden auf der am IFUM vorhandenen Spindelpresse der Firma Siempelkamp mit einer Nennkraft von 8000 kN durchgeführt.

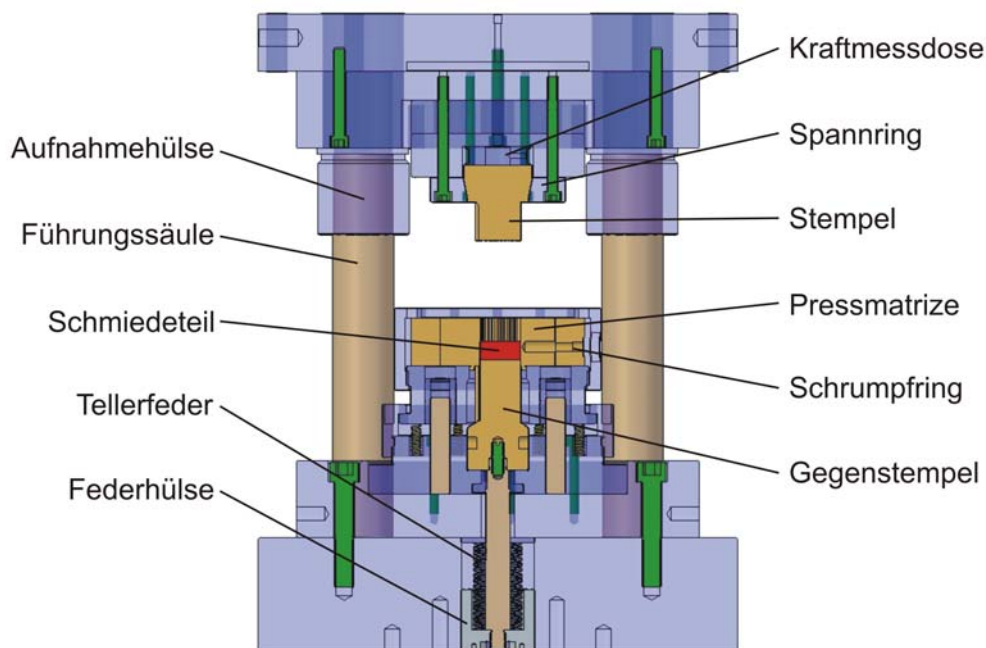


Abbildung 8: Darstellung des Umformwerkzeugs

Zur Abbildung eines industrienahen Prozesses und zum Nachweis der universellen Anwendbarkeit der KNN auf beliebige Schmiedeprozesse wird ein Zahnrad im geschlossenen Gesenk hergestellt. Hierfür wird ein Modellwerkzeug für das Gesenkschmieden konstruiert (Abbildung 8).

Als Werkstücke werden gedrehte Rohteile aus 16MnCr5 (1.7131) verwendet. Die Höhe des Rohteils (normal: 34,625 mm) und der Rohteildurchmesser (normal: 32 mm) sind variable Prozessgrößen. Die Ober- und Unterwerkzeuge werden über Heizmanschetten erwärmt. Die Rohteile werden vor dem Erwärmungsprozess auf einer elektronischen Waage gewogen. Die Rohteilmasse wird dabei gezielt variiert. Anschließend erfolgt die Erwärmung des Rohteils in einem Induktionsdurchstoßofen auf Schmiedetemperatur (optimal: $T = 1200\text{ °C}$), wobei die Temperatur der erwärmten Werkstücke von einem Pyrometer erfasst wird. Die Werkstücktemperatur wird durch Anpassung der Heizleistung des Induktionsofens variiert. Die Handhabung der erwärmten Rohteile erfolgt manuell und unter Aufnahme der Einlegezeit als variable Prozessgröße. Im Umformwerkzeug und an der Umformmaschine sind unterschiedliche Sensoren zur Kennwertaufnahme integriert bzw. installiert (Abbildung 9).

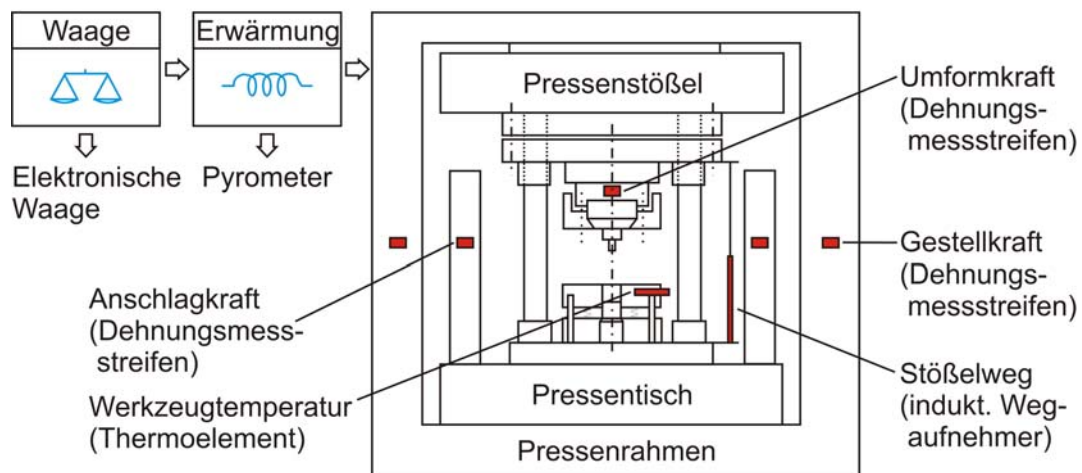


Abbildung 9: Schematische Darstellung des gesamten Pressenaufbaus

Neben der Rohteilmasse, der Rohteiltemperatur nach der Erwärmung und der Einlegezeit werden maschinenseitig Kennwerte für das Training des KNN aufgenommen. Die Werkzeugtemperatur wird mit einem Thermoelement aufgezeichnet. Die Messung der Umform-, Anschlag- und Gestellkraft erfolgt mittels Dehnungsmessstreifen. Der Stößelweg wird mit einem induktiven Wegaufnehmer aufgezeichnet. Aus dem Stößelweg wird der Umformweg und anschließend zusammen mit der Umformkraft die Umformenergie berechnet. Im Anschluss an die durchgeführten

Versuche werden die abgekühlten Bauteile auf einer 3D-Koordinatenmessmaschine vermessen.

Trainings- und Testphase

Die Anzahl der Ein- und Ausgabeneuronen ist durch die Anzahl der Eingabedaten sowie der vorliegenden Zieldaten festgelegt. Die Messergebnisse aus den Versuchen entsprechen dem Netzeingang, die Gutteil-/Schlechtteilbewertung dem Netzausgang. Die Eingabedaten sind Maximalkraft, Umformweg, 10 Messpunkte (10 äquidistante Messpunkte über Kraftverlauf), Umformenergie, Temperatur des Rohteils, Masse, Einlegezeit und Temperatur des Werkzeugs. Dadurch ist die Anzahl der Eingabeneuronen auf höchstens 17 festgelegt. Die Ausgabe liefert ein Ausgabeneuron, das die Bewertung in gut bzw. schlecht vornimmt. Eine Zwischenschicht erscheint für das vorliegende Problem als ausreichend. Die Anzahl der verdeckten Neuronen wird in der Zwischenschicht variiert.

Das neuronale Netz soll eine Klassifizierung der Bauteile in Gut- und Schlechtteile vornehmen. Dazu muss zunächst die Bauteilqualität bewertet werden (Tabelle 1). Hierzu werden mehrere Kriterien wie Kopfkreis-, Fußkreis-, Teilkreisdurchmesser und Breite des Zahnrades herangezogen. Des Weiteren werden die Werte zur Beurteilung der Ausformung des Zahnrades, Ba(99%) und Ba(95%), ermittelt. Diese geben die Breite des Zahnrades bei 99% bzw. 95% des Kopfkreisdurchmessers an. Nur wenn ein Zahnrad bezüglich jedes dieser Merkmale in dem vorab festgelegten Bereich liegt, wird die Qualität mit „gut“ bewertet. Die Festlegung der Gutteilbereiche für die Qualitätsmerkmale Kopfkreis-, Teilkreis-, Fußkreisdurchmesser und Breite wird hier beispielhaft vorgenommen. Zunächst wird jeweils der Mittelwert des entsprechenden Merkmals über alle 281 vermessenen Teile gebildet. Die obere und die untere Grenze werden bestimmt als der Mittelwert $\pm 0,5\%$.

Für die Trainingsdurchläufe liegen insgesamt 281 Datensätze vor, davon werden entsprechend der getroffenen Festlegung 56 als Gutteile und 225 als Schlechtteile klassifiziert. Das so viele "Schlechtteile" enthalten sind, lässt sich damit begründen, dass ein sinnvolles Training des KNN den kompletten zu erwartenden Wertebereich der Eingangsparameter umfassen muss. Im Ergebnis sind hier die Gutteile nicht so überrepräsentiert, wie dies in einem Fertigungsprozess anzustreben wäre.

Tabelle 1: Festlegung der Qualitätskriterien zur Bestimmung der Bauteilqualität

	Kopfkreis- durchmesser	Teilkreis- durchmesser	Fußkreis- durchmesser	Breite	Ba (99%)	Ba (95%)
min	47,389	43,529	37,315	18,089	0,000	12,400
max	48,538	43,720	38,476	18,622	15,045	18,312
mid	48,247	43,648	37,458	18,381	6,704	16,205
gut von	48,006	43,430	37,271	18,289	10,000	15,000
gut bis	48,488	43,867	37,646	18,473	20,000	20,000

Im Anschluss eines Trainings ist die Qualität des trainierten Netzes zu bewerten. Die Güte des Netzes kann über den Generalisierungsfehler gemessen werden, für den verschiedene Funktionen Einsatz finden. Hierfür wird der mittlere quadratische Fehler zur Bewertung herangezogen [8]:

$$MSE = \frac{1}{n} \sum_{i=1}^n (t_{calc} - t_{real})^2 \quad (\text{Gl. 3})$$

MSE mean square error (mittlerer quadratischer Fehler)

t_{calc} vom Netz errechnete Ausgabe

t_{real} vorgegebene Ausgabe, teaching input

n Anzahl der Datensätze

Reduzierung der notwendigen Eingangsparameter

Es soll das optimale Lernverfahren ermittelt werden, mit dem die nachfolgenden Untersuchungen durchgeführt werden können. Dazu werden acht Trainingsalgorithmen ausgewählt:

- Backpropagationverfahren (traingd),
- Erweiterung des Backpropagationverfahrens mit adaptiver Lernrate (traingda),
- Backpropagationverfahren mit Momentum (traingdm),
- Backpropagationverfahren mit adaptiver Lernrate,
- Momentum (traingdx),
- Levenberg-Marquardt-Verfahren (trainlm),
- Resilient Propagation-Verfahren (trainrp) und
- Konjugierte Gradientenverfahren (trainscg).

Mit jedem Lernverfahren werden Trainingsdurchläufe mit 5, 10, 15 und 20 verdeckten Neuronen durchgeführt. Da eventuell Abhängigkeiten von der Initialisierung der Verbindungsgewichte bestehen, erfolgen für jede Netzeinstellung 10 Trainingsdurchläufe. Für jeden einzelnen Trainingsdurchlauf werden der mittlere quadratische Fehler (MSE) und die benötigte Anzahl an Epochen ermittelt (Abbildung 10 und 11).

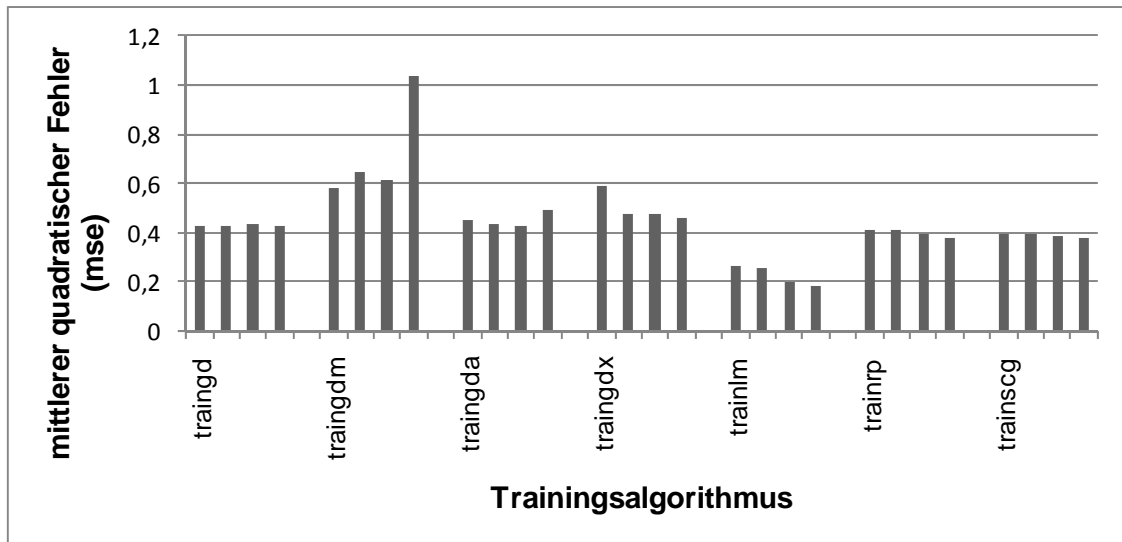


Abbildung 10: MSE - Training mit allen Eingangsparametern

Die vier Balken, die für jedes Lernverfahren dargestellt sind, stellen den Mittelwert über jeweils zehn Trainingsdurchläufe mit 5, 10, 15 und 20 Neuronen (von rechts nach links) in der verdeckten Schicht dar.

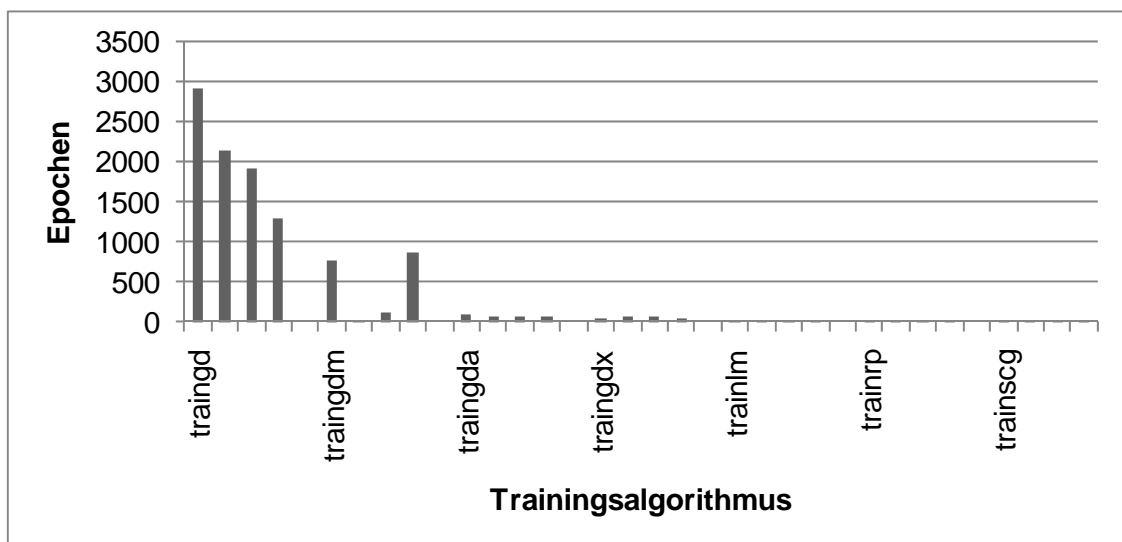


Abbildung 11: Epochen - Training mit allen Eingangsparametern

In den Trainingsdurchläufen mit allen Eingangsparametern stellt sich die Levenberg-Marquardt-Methode als vorteilhaft für die vorliegende Problemstellung dar. Die Mittelwerte des mittleren quadratischen Fehlers sind hierbei gering. Der Levenberg-

Marquardt-Algorithmus gilt als rechen- und zeitaufwändig, was sich jedoch im vorliegenden Fall nicht bestätigt, da die mittlere Anzahl von Epochen die geringste der sieben getesteten Lernverfahren ist. Aufgrund dessen wird in den weiteren Trainingsdurchläufen der Levenberg-Marquardt-Algorithmus angewandt.

Durch eine Reduzierung der Eingangsparameter kann die Zahl der notwendigen Messensoren reduziert werden. Die durchgeführten Trainingsdurchläufe zur Reduzierung der Eingangsparameter sollen zeigen, ob und wie stark sich die Ergebnisse bei Trainingsdurchläufen mit weniger als 8 Eingangsparametern verschlechtern. Die Anzahl der verdeckten Neuronen wird wie zuvor variiert, d.h. auf 5, 10, 15 und 20 gesetzt. Dabei wird die Anzahl der Eingangsparameter in jedem Abschnitt um einen reduziert. Die Anzahl aller möglichen Kombinationen beläuft sich somit auf 254. Für jeden einzelnen Trainingsdurchlauf werden Informationen über die Netztopologie, die benötigte Anzahl an Epochen und der mittlere quadratische Fehler ermittelt. Mit jeder Netzeinstellung werden 10 Trainingsdurchläufe durchgeführt, weshalb sich für die Kriterien jeweils 10 Werte ergeben. Diese werden für jede Netzeinstellung gemittelt, so dass als Ergebnis der Mittelwert des mittleren quadratischen Fehlers bzw. der Epochenanzahl über 10 Trainingsdurchläufe vorliegt.

In Abbildung 12 sind beispielhaft die Ergebnisse aus den durchgeführten Tests zur Parameterreduzierung dargestellt.

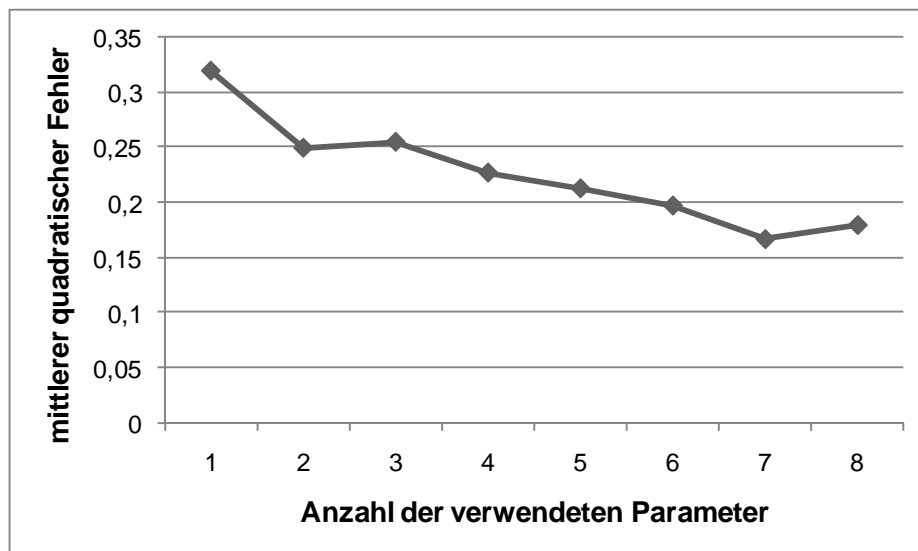


Abbildung 12: MSE in Abhängigkeit der verwendeten Eingangsparameter

Mit einer Reduktion der Eingangsparameter geht tendenziell eine Verschlechterung der Ergebnisse einher. Insgesamt kann festgestellt werden, dass bei Nichtberücksichtigung der Parameter Umformweg, Rohteilmasse und Werkzeugtemperatur

(Parameteranzahl = 5) im Vergleich zur Verwendung von acht Parametern ein geringer Einfluss auf den MSE vorliegt. Hierbei muss im jeweiligen Anwendungsfall abgeschätzt werden, inwieweit eine Zunahme des Fehlers toleriert werden kann.

Zusammenfassung und Ausblick

Die durchgeführten Untersuchungen verdeutlichen die Möglichkeiten bei der Qualitätsüberwachung von Schmiedeprozessen durch den Einsatz Künstlicher Neuronaler Netze. Mit Hilfe eines trainierten Netzes können unter Berücksichtigung weniger Messdaten Ausschussteile direkt nach dem Umformprozess identifiziert werden. Durch die Reduzierung der Eingangsparameter wird der Messaufwand weiter verringert. In industriellen Prozessen werden die meisten der notwendigen Prozessgrößen bereits zur Qualitätsüberwachung eingesetzt, wodurch eine Implementierung des dargestellten Qualitätsüberwachungskonzeptes schnell realisierbar ist.

In der zukünftigen Forschung sollen die bisher erzielten Ergebnisse weitergeführt und ausgebaut werden. Dafür soll das bestehende Qualitätsüberwachungssystem um ein in den Prozess integriertes Regelungssystem erweitert und anhand eines industrienahe, automatisierten Schmiedeprozesses validiert werden.

Danksagung

Die vorgestellten Ergebnisse entstanden im Rahmen des Forschungsprojekts „Bildung eines Prozessmodells für die Qualitätsüberwachung beim Gesenkschmieden durch den Einsatz Künstlicher Neuronaler Netze“ am Institut für Umformtechnik und Umformmaschinen (IFUM) der Leibniz Universität Hannover (LUH). Die Autoren danken der Deutschen Forschungsgesellschaft (DFG) für die finanzielle Unterstützung dieses Projektes.

Literatur:

- [1] Schnerr, O.: *Automatisierung der Online-Qualitätsüberwachung beim Kunststoffspritzgießen*, Dissertation RWTH Aachen, 2000
- [2] Zell, A.: *Simulation neuronaler Netze*, Oldenbourg Verlag, München, 2000
- [3] Ernst, F.B.G.: *Qualitätskontrolle auf Basis optischer Prozessdiagnostik und neuronaler Netze beim Thermischen Spritzen*, Shaker Verlag, Aachen 2007

- [4] Rey, G. D.; Wender, K. F.: *Neuronale Netze – Eine Einführung in die Grundlagen, Anwendungen und Datenauswertung*, Verlag Hans Huber, Hogrefe AG, Berlin 2008
- [5] Kohut, R.: *Neuronale Netze als Modell boolescher Funktionen*, 2007
- [6] Scherer, A.: *Neuronale Netze: Grundlagen und Anwendungen*, Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig/Wiesbaden 1997
- [7] Adamy, J.: *Fuzzy-Logik, Neuronale Netze und Evolutionäre Algorithmen*, Shaker Verlag GmbH, Aachen 2007
- [8] Woernle, M.I.: *Anwendbarkeit künstlicher neuronaler Netze zur Untergrundbewertung in der oberflächennahen Geothermie*, Schriftenreihe angewandte Geologie Karlsruhe, Karlsruhe 2008

Autoren:

Prof. Dr.-Ing. Bernd-Arno Behrens studierte Maschinenbau an der Universität Hannover und promovierte am Institut für Umformtechnik und Umformmaschinen (IFUM) in Hannover. Seit Oktober 2003 ist er Leiter des IFUM der Leibniz Universität Hannover.

Dipl.-Ing. Klaus-Georg Kosch studierte Maschinenbau an der Leibniz Universität Hannover. Seit Ende 2009 ist er wissenschaftlicher Mitarbeiter in der Abteilung Massivumformung am IFUM.

Dipl.-Ing. Hagen Schmidt studierte Bauingenieurwesen an der Bauhaus-Universität Weimar. Von 2003 bis 2010 war er wissenschaftlicher Mitarbeiter in der Abteilung Massivumformung am IFUM.

*Korrespondenzautor:

Dipl.-Ing. Klaus-Georg Kosch
Abteilung Massivumformung
Tel: 0511-762 2168, Fax: 0511-762 3007
kosch@ifum.uni-hannover.de