

# Physik-geführte NARXnets (PGNARXnets) zur Zeitreihenvorhersage

Silas Aaron Selzer<sup>1</sup>, Fabian Bauer<sup>2</sup>, Sebastian Bohm<sup>1</sup>, Peter Bretschneider<sup>2</sup>, Erich Runge<sup>1</sup>

<sup>1</sup> Fachgebiet Theoretische Physik I  
Technische Universität Ilmenau  
Weimarer Straße 25  
98693 Ilmenau

E-Mail: {silas-aaron.selzer, sebastian.bohm, erich.runge}@tu-ilmenau.de

<sup>2</sup> Fachgebiet Energieeinsatzoptimierung  
Technische Universität Ilmenau  
Gustav-Kirchhoff-Straße 5  
98693 Ilmenau

E-Mail: {fabian.bauer, peter.bretschneider}@tu-ilmenau.de

## Kurzfassung

Durch stark steigende Datenmengen und der Optimierung der Rechentechnik konnten in den letzten Jahren massive Fortschritte in maschinellen Lernverfahren erzielt werden. In vielen Fällen ist die Datenerfassung jedoch mit einem großen Zeit- und Kostenaufwand verbunden. Daher wäre es von Vorteil, bereits aus Teilinformationen valide Schlussfolgerungen ziehen zu können. Für Anwendungen auf naturwissenschaftlich-technische Problemstellungen existiert typischerweise sehr klar definiertes Vorwissen in Form von physikalischen Gleichungen oder empirischen Beziehungen. Eine Synthese von theoretischem Vorwissen verbunden mit datenwissenschaftlichen Methoden erscheint für solche Situationen sehr vielversprechend. In diesem Beitrag wird am Beispiel der Zeitreihenvorhersage an einem chaotischen System gezeigt, dass durch die Berücksichtigung des zugrundeliegenden Differentialgleichungssystems in

DOI: 10.58895/ksp/1000138532-15 erschienen in:

**Proceedings - 31. Workshop Computational Intelligence : Berlin, 25. - 26. November 2021**

DOI: 10.58895/ksp/1000138532 | <https://www.ksp.kit.edu/site/books/m/10.58895/ksp/1000138532/>

der Verlustfunktion eines künstlichen neuronalen so genannten NARX-Netzes die Extrapolationsfähigkeit des Modells verbessert werden kann. Dazu wirken die Physik-geführten Verlustterme in Kombination mit klassischen Verlusten als eine Art Regularisierung. Verwendet man nur Physik-geführte Verluste ist es darüber hinaus möglich, neuronale Netze *label-free* zu trainieren. Dadurch entfällt die kosten- und zeitintensive Beschaffung großer Trainingsdatensammlungen.

## 1 Einführung

Die Zeitreihenprognose ist ein wichtiger Bereich des maschinellen Lernens. Zeitreihenvorhersage bedeutet, dass zukünftige Werte einer Funktion auf der Basis mehrerer historischer Werte ermittelt werden. Neben den zurückliegenden Funktionswerten können weitere exogene Einflussgrößen gegeben sein, welche auf die Zeitreihenprognose einwirken. Anwendung findet die Zeitreihenprognose in vielen Bereichen, vom Energiesektor (z.B. Vorhersage des Energieverbrauchs von Mehrfamilienhäusern [1]) über die Hydrologie (z.B. Vorhersage des Zu- und Abflusses von Stauseen [2]) bis hin zur Finanzwirtschaft (z.B. Vorhersage von Devisenkursen, Aktienindizes oder dem Wirtschaftswachstum [3, 4]). Durch die Analyse des Verhaltens in der Vergangenheit und der Annahme, dass das zukünftige Verhalten deterministisch ist, sollen zukünftige Werte vorausgesagt werden.

Für die Vorhersage von Zeitreihen mittels neuronaler Netze werden beispielsweise sogenannte NARXnets (Nonlinear autoregressive with external input networks) verwendet [5, 6]. Konkret versteht man unter NARXnets rekurrente dynamische neuronale Netze, welche die Vorhersage der gesuchten Größe zum nächsten Zeitschritt  $\hat{J}(t_n)$  aus den vorangegangenen  $k$  Werten der exogenen Eingangsvariablen  $(I(t_{n-1}), \dots, I(t_{n-k}))$  und zurückliegenden  $l$  Werten des vorherzusagenden Ausgangssignals  $(J(t_{n-1}), \dots, J(t_{n-l}))$  berechnen. Die Vorhersage des NARXnet wird nach der Berechnungsvorschrift

$$\hat{J}(t_n) = \text{NARXnet}(J(t_{n-1}), \dots, J(t_{n-k}), I(t_{n-1}), \dots, I(t_{n-l})) \quad (1)$$

bestimmt. Bei der Anwendung der NARXnets auf naturwissenschaftlich-technische Probleme, wie das Lösen von Differentialgleichungen, erzielen diese vergleichsweise gute Ergebnisse. In Hinblick auf die aufkommende Thematik des Theorie-gesteuerten Lernens [7] stellt sich die Frage, ob die Ergebnisse der NARXnets durch Erweiterung der Verlustfunktion um einen Physik-geführten Verlustterm (Physik-geführte NARXnets) verbessert werden können.

Ziel des Theorie-gesteuerten maschinellen Lernens (auch als Physik-informiertes, Physik-geführtes oder Physik-inspiriertes Lernen bezeichnet) ist die Schaffung einer Synthese aus den datenwissenschaftlichen und Physik-basierten Modellen. Dadurch wird es möglich, sowohl theoretisches Vorwissen über das System in Form von physikalischen Gleichungen oder empirischen Abhängigkeiten als auch Informationen aus Daten zu nutzen (siehe Abbildung 1) [8, 9]. Durch den massiven Anstieg verfügbarer Daten und Rechenressourcen konnten in den letzten Jahren große Fortschritte bei maschinellen Lernverfahren erzielt werden. Dies spiegelt sich beispielsweise in der Bild- [10] oder auch Spracherkennung [11] wider. Als problematisch bei naturwissenschaftlich-technischen Problemen stellt sich jedoch zumeist eine zeit- und kostenaufwändige Datenerfassung dar. Zudem besteht oft ein Ungleichgewicht zwischen beispielsweise sehr vielen Betriebsdaten, die Systemzustände nahe dem Normalzustand beschreiben, während Vorhersagen besonders wichtig sind für seltene außergewöhnliche Situationen mit starken Abweichungen, in denen letztlich extrapoliert werden muss. Ziel wäre es, bereits aus einer geringen Menge an Teilm Informationen valide Schlussfolgerungen zu ziehen. Die maschinellen Lernverfahren zeigen bei geringen Datenmengen oftmals schlechte oder keine Konvergenz. Um diesem Problem entgegenzuwirken, kann theoretisches Vorwissen, welches bisher in ML-Verfahren nicht genutzt wird, in Form von physikalischen Gesetzen oder empirischen Regeln in den Lernprozess eingepflegt werden. Dadurch wird der Lösungsraum beschränkt und man kann bereits mit geringen Datenmengen gute Ergebnisse erzielen. [12]

Für das Theorie-gesteuerte maschinelle Lernen gibt es vielfältige Ansätze: Der von Raissi et al. [12] vorgeschlagene Ansatz zur datengesteuerten Lösung und datengestützten Bestimmung der Koeffizienten in partiellen

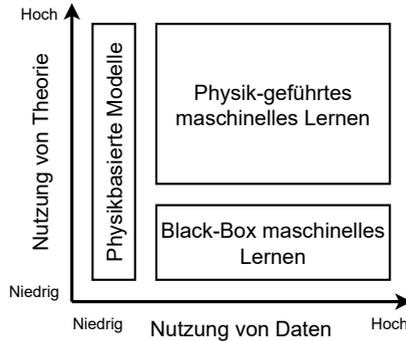


Bild 1: Einordnung des Physik-geführten maschinellen Lernens im Vergleich zu rein datenwissenschaftlichen Black-Box und theoretischen Physik-basierten Ansätzen. [8, 9]

Differentialgleichungen basiert auf der Verwendung der umgestellten Differentialgleichung als Verlustfunktion. Indem die automatische Differenzierung nicht nur für das Backpropagationverfahren genutzt wird, sondern auf die Bestimmung der Ableitungen in der Differentialgleichung in Hinblick auf die Eingangsneuronen ausgedehnt wird, lassen sich die Ableitungen berechnen. Dieser Ansatz wurde auf weitere Probleme aus dem Bereich der Geologie [13], der Nanooptik [14] oder dem Stahlbau [8] übertragen. Weiterhin wurden erste Bibliotheken [15] für diesen Ansatz erstellt.

Ein alternativer Ansatz ist es, die klassischen Verlustfunktionen durch Physik-geführte Verlustterme zu ergänzen. Dadurch verspricht man sich, die Vorteile der Datenwissenschaften mit Vorwissen des Systemverhaltens zu verbinden und somit die Vorteile beider Modellierungsformen zu kombinieren. Beispielsweise kann physikalisches Vorwissen in Form von Differentialgleichungen, Bilanzgleichungen, Monotoniebedingungen oder auch Nichtnegativ-Werten in einen zusätzlichen Verlustterm einfließen. Ein solcher Ansatz wurde von Karpatne et al. [9] zur Vorhersage der Temperatur in Seen genutzt. Auch die Lösung von Eigenwertgleichungen [16] oder die Vorhersage des Schwing-Verhaltens von Gebäuden während bzw. nach Erdbeben [17] wurden mit ähnlichen Ansätzen untersucht. Eine Zeitreihenvorhersage mittels Physik-geführter NARXnets wurde anhand der Dynamik von thermochemischen Energiespeichern betrachtet [18].

In den bisherigen Untersuchungen zur Kombination von theoretischem Vorwissen und den NARXnets zur Zeitreihenvorhersage wurden physikalisches Vorwissen in Form von Bilanzgleichungen, Monotoniebedingungen sowie Nicht-negativ-Werten in die Verlustfunktion mit eingebaut [18] oder aber die Physik in Form einer Strukturänderung aber Beibehaltung der klassischen Verlustfunktion [19] dem neuronalen Netz aufgezwungen. Der hier vorgeschlagene Ansatz zielt darauf ab, die physikalische Konsistenz der Modelle durch Verlustfunktionen zu sichern, welche auf den zugrundeliegenden Differentialoperator des Problems zugeschnitten sind. Es zeigt sich jedoch, dass der Ansatz, die automatische Differenzierung zur Berechnung der Ableitungen in den Verlustfunktionen (ähnlich der Physik-informierten neuronalen Netze [12]) zu nutzen, deutlich schlechtere Ergebnisse im Vergleich zu klassischen NARXnets aufweist. Grund dafür ist vermutlich, dass die Zeit als zusätzlicher Parameter in den Eingangsneuronen eingespeist werden muss. Das Netzwerk wird dadurch so trainiert, dass die Zeit als Hauptinformationsquelle dient und die zurückliegenden Funktionswerte ignoriert werden. Das NARXnet ist jedoch gerade darauf ausgelegt, Informationen aus den zurückliegenden Funktionswerten zu nutzen, um damit den nächsten Funktionswert vorherzusagen. Durch die Nutzung der zurückliegenden Funktionswerte, welche in die Eingangsneuronen eingespeist werden, lassen sich die Ableitungen in der Verlustfunktion mit linksseitigen Differenzenquotienten abschätzen und somit die Differentialgleichung als Verlustfunktion wählen. Untersucht wurden einmal die Kombination aus klassischen und Physik-geführten Verlusttermen wie auch die alleinige Nutzung der Physik-geführten Verlustterme.

Die Ausarbeitung ist folgendermaßen gegliedert. In Abschnitt 2 wird der Lösungsansatz der Physik-geführten NARXnets allgemein eingeführt. Nachfolgend wird dieser auf das Beispiel einer chaotische Zeitreihe (die modifizierte Van der Pol Gleichung) übertragen. Die Problemstellung wird in Abschnitt 3 erläutert. Die Ergebnisse sind in Abschnitt 4 dargestellt. Abschnitt 5 fasst die Untersuchungen zusammen und gibt einen Ausblick auf weitergehende Fragestellungen.

## 2 Physik-geführte NARXnets

In den bisherigen Standard-Ansätzen der NARXnets wird als Verlustfunktion

$$\min_{\Theta} Loss_{Std.} = \min_{\Theta} Loss_{emp.}(J, \hat{J}) + \lambda_{reg.} Loss_{reg.}(\Theta) \quad (2)$$

verwendet, die darauf abzielt, die empirischen Verluste  $Loss_{emp.}(J, \hat{J})$  der Modellvorhersage  $\hat{J}$  durch die Veränderung der Modellparameter  $\Theta$  zu minimieren. Teilweise wird der besagte Ansatz um einen weiteren Term  $Loss_{reg.}(\Theta)$  ergänzt, welcher die Komplexität des Modells bestraft. Ziel ist es, dadurch die Überanpassung an die vorliegenden Daten zu verhindern. Durch den Parameter  $\lambda_{reg.}$  wird das relative Gewicht dieses Terms gegenüber dem empirischen Fehler eingestellt. Wie bereits in der Einführung angeschnitten, hängt die Effektivität dieses Ansatzes maßgeblich von der Größe der Trainingsdatenmenge ab. Weiterhin gibt es keine Garantie, dass solch ein Ansatz die zugrundeliegende Physik korrekt wiedergibt bzw. respektiert. [9]

Der Ansatz der Physik-geführten NARXnets zielt auf Zeitreihenvorhersagen ab, bei denen die zurückliegenden Funktionswerte wie auch exogene Eingangsvariablen in das Netz eingespeist werden sowie das theoretische Vorwissen über den Prozess in Form von Differentialgleichungen oder Differentialgleichungssystemen vorliegt. Das betrachtete physikalische System wird durch die vorherzusagende Variable  $J$ , die exogenen Eingangsvariablen  $I$  wie auch weitere physikalische Variablen  $\mathbf{Z}$  mittels einer hier generisch geschriebenen Differentialgleichung

$$\mathcal{F}(I, J, \mathbf{Z}) = 0 \quad (3)$$

beschrieben. [9]

Durch die Physik-geführte Verlustfunktion soll nun gemessen werden, in wie weit die Vorhersage von  $\hat{J}$  die Differentialgleichung erfüllt. Dazu wird gerade die Nichterfüllung besagter Differentialgleichung

$$Loss_{PG} = \lambda_{PG} Loss(\mathcal{F}(I, \hat{J}, \mathbf{Z})) \quad (4)$$

bestraft, wobei  $\lambda_{PG}$  ein Hyperparameter des Physik-geführten Verlusts darstellt [9]. Als Verlustmaße können etablierte Maße, wie z.B. der mittlere absolute (MAE) oder auch der mittlere quadratische Fehler (MSE) verwendet werden. Die weiteren Betrachtungen konzentrieren sich auf Differentialgleichungen mit zeitlichen Ableitungen. Da die Zeit nicht als Eingabe in den NARXnets dient, wird in diesem Beitrag vorgeschlagen, die Ableitungen in der Verlustfunktion durch linksseitige Differenzenquotienten abzuschätzen. Für alle exogenen Variablen wie auch die Zielvariable sind die zurückliegenden Funktionswerte bekannt, da diese als Eingangsgrößen in das neuronale Netz dienen. Darüber hinaus müssen die Zeitschrittweite  $\Delta t$  und die Konstanten der Differentialgleichungen gegeben sein. Eine besonders robuste Formel zur Abschätzung einer Ableitung nach [20] ist gegeben durch

$$\left. \frac{d^m J}{dt^m} \right|_{t_i=t_0} \approx \sum_{n=0}^{N-1} a_n^{(m,N)} J(t_i + s_n) \quad \text{mit } N > m, \quad (5)$$

wobei  $m$  der Ordnung der Ableitung,  $N$  der Anzahl der Stützstellen und  $s_n$  den Stützstellen relativ zu  $t_i$  entspricht. Durch die Lösung des linearen Gleichungssystems

$$\begin{pmatrix} 1 & \cdots & 1 \\ s_1 & \cdots & s_N \\ \vdots & \ddots & \vdots \\ s_1^{N-1} & \cdots & s_N^{N-1} \end{pmatrix} \begin{pmatrix} a_0^{(m,N)} \\ a_1^{(m,N)} \\ \vdots \\ a_{N-1}^{(m,N)} \end{pmatrix} = m! \begin{pmatrix} \delta_{0,m} \\ \delta_{1,m} \\ \vdots \\ \delta_{N-1,m} \end{pmatrix} \quad (6)$$

können für jedes  $m$  die Koeffizienten  $a_n^{(m,N)}$  bestimmt werden, wobei  $\delta_{i,j}$  das Kronecker-Delta ist [21].

Mit diesem Physik-geführten Ansatz lassen sich zwei Typen von Physik-geführten NARXnets ableiten. Zum einen können die Physik-geführten NARXnets allein durch den Physik-geführten Verlust (Gleichung 4) trainiert werden. Ein solches Netz lässt sich ohne die Vorgabe der korrekten Lösung trainieren. Im Gegensatz zu traditionellen Ansätzen wird das Netz somit ausschließlich an nicht gelabelten Datensätzen ausgewertet (engl. *label-free*

*learning*). Weiterhin kann der Standard-Ansatz (Gleichung 2) um den Physik-geführten Verlustterm (Gleichung 4) ergänzt werden, um den klassischen datenwissenschaftlichen Ansatz mit Vorwissen zu kombinieren.

### 3 Problemstellung

In diesem Abschnitt wird die Problemstellung eingeführt, an welcher die vorgestellten Ansätze getestet und mit dem Standardansatz verglichen werden. Als Anwendungsbeispiel wird die modifizierte Van der Pol Gleichung verwendet und das Rahmenwerk der Physik-geführten NARXnets auf diese Problemstellung übertragen. Gerade in der modifizierten Form weist die Van der Pol Gleichung ein chaotisches Verhalten auf, sodass die zeitliche Entwicklung zunächst nicht prognostizierbar wirkt, obwohl die mathematische Beschreibung des Systems deterministisch ist.

#### 3.1 Van der Pol Gleichung

Zur Beschreibung des allgemeinen Phänomens von Relaxationsschwingungen wurde 1926 die Van der Pol Gleichung von Balthasar van der Pol eingeführt [22]. Es handelt sich dabei um eine nichtlineare gewöhnliche Differentialgleichung zweiter Ordnung, welche die Form

$$\frac{d^2x}{dt^2} - \varepsilon (1 - x^2) \frac{dx}{dt} + x = 0 \tag{7}$$

mit einem Parameter  $\varepsilon > 0$  annimmt. Fügt man einen anregenden Term hinzu, ergibt sich

$$\frac{d^2x}{dt^2} - \varepsilon (1 - x^2) \frac{dx}{dt} + x = A \cos(2\pi ft). \tag{8}$$

Für beliebige Werte der Konstanten  $A$  und  $\varepsilon$  erhält man einen Grenzyklus oder quasiperiodische Lösungen, aber kein chaotisches Verhalten. Dieses tritt erst durch die Modifizierung der Van der Pol Gleichung auf. [23]

## 3.2 Modifizierte Van der Pol Gleichung

Ueda et al. [24] haben für die Beschreibung des angeregten Oszillators mit negativem Widerstand eine modifizierte Van der Pol Gleichung

$$\frac{d^2x}{dt^2} - \varepsilon (1 - x^2) \frac{dx}{dt} + x^3 = A \cos(2\pi ft) \quad (9)$$

formuliert, welche ein chaotisches Verhalten für bestimmte Werte der Koeffizienten  $A$  und  $\varepsilon$  aufweist. Für die weitere Diskussion derselben wird die nicht-lineare gewöhnliche Differentialgleichung zweiter Ordnung in ein System aus zwei nichtlinearen gewöhnlichen Differentialgleichungen erster Ordnung

$$\frac{dx}{dt} = y \quad (10)$$

$$\frac{dy}{dt} = \varepsilon (1 - x^2)y - x^3 + A \cos(2\pi ft) \quad (11)$$

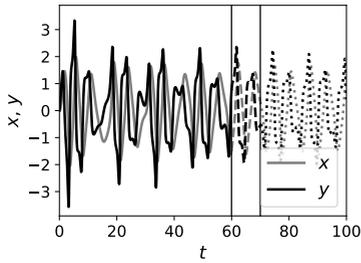
umgewandelt. Je nach Wahl der Koeffizienten  $A$  und  $\varepsilon$  tritt chaotisches oder aber ein reguläres Verhalten auf. [23]

In Abbildung 2 ist die numerisch berechnete Lösung (mittels explizitem Runge-Kutta-Verfahren der fünften Ordnung [25]) gerade für die Wahl der Koeffizienten dargestellt, sodass sich ein chaotisches Verhalten ergibt.

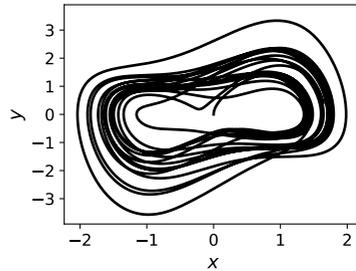
Anhand dieser Zeitreihen soll die Performance der Physik-geführten im Vergleich zu den klassischen NARXnets untersucht werden. Als Datengrundlage dient die Zeitreihe mit den Koeffizienten  $A = 1$  und  $\varepsilon = 1$ . Diese wurde in einen Zeitbereich von 0 bis 100 aufgenommen. In Abbildung 2a ist die Aufteilung der Zeitreihe in Trainingsdaten (6000 Werte), Validierungsdaten (1000 Werte) und Testdaten (3000 Werte) gezeigt.

Um neben der numerischen Lösung der Zeitreihe auch das Training mit *realen* Daten zu testen, wird die numerische Lösung verrauscht. Dazu wird auf die numerische Lösung weißes Rauschen nach der Formel

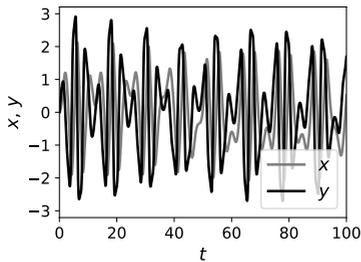
$$x_{\text{real}} = x + v \cdot N(\mu = 0, \sigma_x) \quad (12)$$



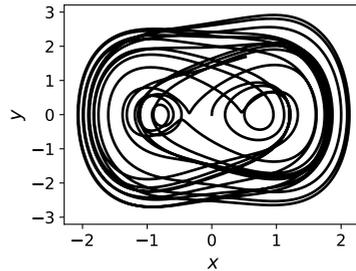
(a) Zeitliche Entwicklung:  $A = 1$  und  $\varepsilon = 1$



(b) Phasendiagramm:  $A = 1$  und  $\varepsilon = 1$



(c) Zeitliche Entwicklung:  $A = 1$  und  $\varepsilon = 0.1$



(d) Phasendiagramm:  $A = 1$  und  $\varepsilon = 0.1$

Bild 2: Modifizierte Van der Pol Gleichung. Für die Initialbedingung gilt:  $x(0) = 0$  und  $y(0) = 0$ . Die Frequenz wurde mit  $f = \frac{1}{2\pi}$  gewählt.

gelegt, wobei  $v$  die Rauschstärke und  $N(\mu = 0, \sigma_x)$  die Normalverteilung mit dem Erwartungswert  $\mu = 0$  und der Standardabweichung der numerisch berechneten Werte  $\sigma_x$  darstellt.

Die Zeitreihe mit den Koeffizienten  $A = 1$  und  $\varepsilon = 0.1$  dient als weitere Testdatenmenge um die Generalisierungsfähigkeit der Physik-geführten und klassischen NARXnets zu überprüfen. Für beide Zeitreihen wurden äquidistante Zeitschritte mit identischer Schrittweite gewählt.

### 3.3 Zeitreihenvorhersage mit neuronalen Netzen

In dem besagten Beispiel ist es das Ziel, mit den unterschiedlichen NARXnet-Ansätzen, die Lösung der modifizierten Van der Pol Gleichung  $x_n$  vorherzusagen. Die klassische NARXnet-Struktur

$$\hat{x}_n = \text{NARXnet}(x_{n-1}, \dots, x_{n-k}, y_{n-1}, \dots, y_{n-l}), \quad (13)$$

wird für das Beispiel dahingehend verändert, dass neben  $x_n$  auch  $y_n$  prognostiziert wird. Dadurch ist es möglich, das Differentialgleichungssystem (Gleichung 10 und 11) in die Verlustfunktion einzupflegen und diese somit auf den zugrundeliegenden Differentialoperator zuzuschneiden (nachfolgend in Unterunterabschnitt 3.3.2 erläutert). Um die Vorhersagen  $\hat{x}_n$  und  $\hat{y}_n$  zum Zeitpunkt  $t_n$  zu bestimmen, werden den neuronalen Netzen die  $k$ -Zeitschritte zurückliegenden Funktionswerte von  $x$  und  $y$  sprich  $x_{n-1}, \dots, x_{n-k}$  und  $y_{n-1}, \dots, y_{n-k}$  als Eingabe vorgegeben.

Für die drei verschiedenen Lösungsansätze (PGNARXnet, PGNARXnetlf und NARXnet; dargestellt in Abbildung 3) werden jeweils identische Architekturen (auch mit identischer Initialisierung der Gewichte und Schwellenwerte), Optimierungseinstellungen und Trainingsdaten verwendet. Das heißt, die drei Netze unterscheiden sich einzig und allein durch ihre Verlustfunktion. Auf die Verlustfunktionen wird nachfolgend eingegangen.

#### 3.3.1 NARXnet

Die zu trainierenden Parameter des NARXnet (Gewichte und Schwellenwerte) werden durch die Minimierung der Kombination des mittleren quadratischen Fehlers der klassischen Verlustfunktionen der beiden Größen  $x$  und  $y$

$$\text{Loss}_{\text{NARXnet}} = \text{MSE}_{\text{klass},1} + \text{MSE}_{\text{klass},2} \quad (14)$$

berechnet. Unter der klassischen Verlustfunktion werden die empirischen Verluste, also die Abweichung der Vorhersage des neuronalen Netzes von der vorgegebenen Lösung, verstanden. Da gerade zwei Variablen als Ausgangsgröße

dienen, werden für beide Vorhersagen ( $\hat{x}_n$  und  $\hat{y}_n$ ) die klassischen Verlustterme

$$MSE_{\text{klass},1} = \frac{1}{M} \sum_{n=k}^{M+k} (x_n - \hat{x}_n)^2 \quad (15)$$

und

$$MSE_{\text{klass},2} = \frac{1}{M} \sum_{n=k}^{M+k} (y_n - \hat{y}_n)^2 \quad (16)$$

ausgewertet, wobei die Größen, gekennzeichnet mit Dach, die Vorhersage des neuronalen Netzes markieren.  $M$  gibt die Anzahl der Vorhersagen an, die im Fall des Trainings der Batchgröße  $M = M_{\text{Batch}}$  entspricht. Da die Vorhersage ab dem  $k$ -ten Zeitschritt des Datensatzes erfolgt, wird die Summation ab diesem Zeitwert gestartet. Verwendet wird der mittlere quadratische Fehler, um größere Abweichungen stärker zu bestrafen. Durch die automatische Differenzierung wird die Verlustfunktion nach allen zu trainierenden Parametern abgeleitet. Mit dem *Adam*-Optimierer in seiner *AMSGrad*-Ausführung [26] werden die zu trainierenden Parameter mittels eines Gradientenverfahrens mit einer adaptiven Schrittweite verändert.

### 3.3.2 PGNARXnetf

Ziel der Untersuchungen ist es zu überprüfen, ob die Ergebnisse der NARXnets durch die Wahl Physik-geführter Verlustfunktionen verbessert werden können. Dazu werden die beiden in Abschnitt 2 vorgeschlagenen Ansätze mit den zuvor gezeigten NARXnet verglichen. Zunächst soll an dieser Stelle das PGNARXnetf eingeführt werden. Darunter ist ein NARXnet zu verstehen, welches allein durch Physik-geführte Verlustterme trainiert wird. Da keine klassischen Verlustterme bei dieser Netzart verwendet werden, wird das Netzwerk *label-free* trainiert.

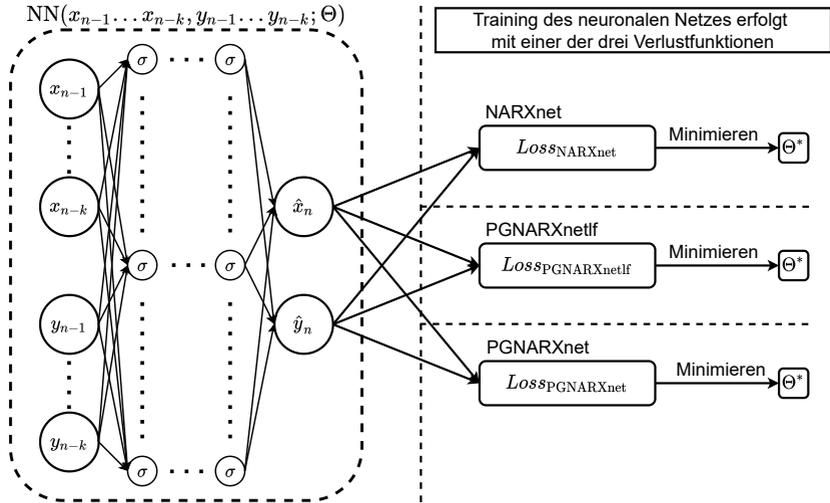


Bild 3: Veranschaulichung der neuronalen Netzstruktur mit drei verschiedenen Lösungsansätzen: NARXnet, PGNARXnetlf und PGNARXnet.  $\Theta$  stellt die Parameter der neuronalen Netze (Gewichte und Schwellenwerte) dar, wobei  $\Theta^*$  die optimierten Parameter angibt.  $\sigma$  markiert die Aktivierungsfunktion.

Die Verlustfunktion des PGNARXnetlf setzt sich aus zwei verschiedenen Verlusttermen zu einer kombinierten Verlustfunktion

$$Loss_{PGNARXnetlf} = MSE_{ODE,1} + MSE_{ODE,2} \quad (17)$$

zusammen. Die Verlustfunktion wird auf das zugrundeliegende Differentialgleichungssystem bestehend aus zwei nichtlinearen gewöhnlichen Differentialgleichungen erster Ordnung (Gleichung 10 und 11) zugeschnitten.

Der erste Verlustterm

$$MSE_{ODE,1} = \frac{\lambda_{ODE,1}}{M} \sum_{n=k}^{M+k} \left( \frac{d\hat{x}_n}{dt} - \hat{y}_n \right)^2 \quad (18)$$

spiegelt die erste Gleichung des Differentialgleichungssystems wider, wobei die Ableitung mittels linksseitigem Differenzenquotienten

$$\frac{d\hat{x}_n}{dt} \approx \frac{a_0 \cdot x_{n-k} + \dots + a_{N-2} \cdot x_{n-1} + a_{N-1} \cdot \hat{x}_n}{\Delta t} \quad (19)$$

abgeschätzt wird.  $N$  gibt die Anzahl der Stützstellen an. In den Differenzenquotienten fließen die als Eingangsgrößen dienenden zurückliegenden Funktionswerte  $x_{n-1}, \dots, x_{n-k}$  wie auch die bekannte Zeitschrittweite  $\Delta t$  ein. Die Vorfaktoren  $a_0, \dots, a_{N-1}$  werden durch die Lösung des linearen Gleichungssystems (Gleichung 6) bestimmt.

Der zweite Verlustterm

$$MSE_{ODE,2} = \frac{\lambda_{ODE,2}}{M} \sum_{n=k}^{M+k} \left( \frac{d\hat{y}_n}{dt} - \varepsilon (1 - \hat{x}_n^2) \hat{y}_n + \hat{x}_n^3 - A \cos(2\pi f t_n) \right)^2 \quad (20)$$

zwingt dem neuronalen Netz die zweite Differentialgleichung des Systems und damit die modifizierte Van der Pol Gleichung auf. Die Ableitung wird, wie oben, mittels linksseitigem Differenzenquotienten

$$\frac{d\hat{y}_n}{dt} \approx \frac{a_0 \cdot y_{n-k} + \dots + a_{N-2} \cdot y_{n-1} + a_{N-1} \cdot \hat{y}_n}{\Delta t} \quad (21)$$

abgeschätzt, wobei die zurückliegenden Funktionswerte  $y_{n-1}, \dots, y_{n-k}$  wie auch die Zeitschrittweite  $\Delta t$  verwendet werden. Darüber hinaus müssen die Koeffizienten  $A$ ,  $\varepsilon$  und  $f$  wie auch die Zeit  $t_n$  bekannt sein, da diese in den Verlustterm integriert werden.

Die Vorfaktoren  $\lambda_{ODE,1}$  und  $\lambda_{ODE,2}$  in den Verlusttermen dienen der Gewichtung, welche für das PGNARXnet entscheidend ist.

### 3.3.3 PGNARXnet

Neben dem PGNARXnetlf wird eine weitere Netzart eingeführt. In dem PGNARXnet werden die Ansätze des NARXnet und des PGNARXnetlf kombiniert. Die Verlustfunktion setzt sich aus vier Verlusttermen

$$Loss_{PGNARXnet} = MSE_{\text{klass},1} + MSE_{\text{klass},2} + MSE_{ODE,1} + MSE_{ODE,2} \quad (22)$$

sowohl den klassischen als auch den Physik-geführten Verlusttermen zusammen. Die Gewichtung der Physik-geführten Verlustterme gegenüber den klassischen Verlusttermen kann über die Vorfaktoren  $\lambda_{\text{ODE},1}$  und  $\lambda_{\text{ODE},2}$  eingestellt werden. Von dieser Art der Verlustfunktion erhofft man sich, dass die Vorteile beider Verlusttermarten überwiegen und somit die Performance der Netze in Hinblick auf die Abweichung der Vorhersage wie auch der Generalisierung verbessert werden. Es ist jedoch zu beachten, dass durch die Hinzunahme der klassischen Verlustterme das Training nicht mehr *label-free* erfolgen kann.

## 4 Ergebnisse

Die Architektur der neuronalen Netze wird in den Versuchen variiert. Verwendet werden jeweils Netze, bestehend aus zwei verdeckten Schichten. Die Anzahl der Neuronen in den verdeckten Schichten  $N_{\text{hidden},1}$  und  $N_{\text{hidden},2}$  werden nach den etwas abgewandelten heuristischen Regeln [27, 28, 29] mit

$$N_{\text{hidden},1} = 2 \cdot (2 \cdot d + 1) \quad \text{und} \quad N_{\text{hidden},2} = 2 \cdot \sqrt{N_{\text{hidden},1}} \quad (23)$$

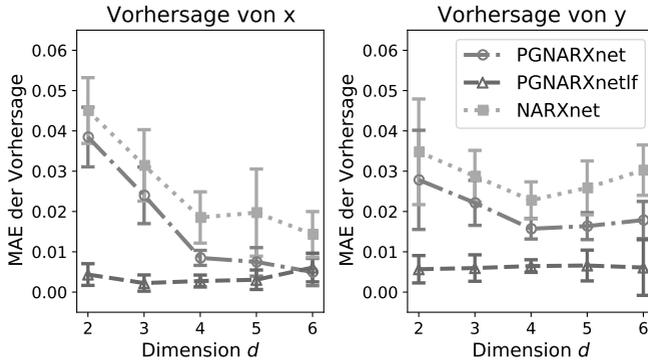
gewählt, wobei  $d$  die *Embedding-Dimension* [30] angibt und  $N_{\text{hidden},2}$  zur nächsten natürlichen Zahl aufgerundet wird. Aufgrund der zwei Funktionswerte als Ausgabegrößen wurde die Anzahl der Neuronen in den Schichten verdoppelt. Die erste heuristische Regel ist auf das Kolmogorov-Theorem in Bezug auf die Approximation von Funktionen zurückzuführen [31]. Das Kolmogorov-Theorem wurde durch Hecht-Nielsen [32] abgewandelt und auf neuronale Netze übertragen (*Kolmogorov Mapping Neural Network Existence Theorem*). Die zweite heuristische Regel besagt, dass die Anzahl an Neuronen in der zweiten Schicht als Quadratwurzel des Produktes der Dimension der ersten versteckten Schicht und der Dimension der Ausgabeschicht gewählt wird [29]. Die *Embedding-Dimension*  $d$  wurde durch Bestimmung des Minimums der *Mutual Information* [33, 34], welches sich in unserem Fall bei einer Verschiebung von  $\tau = 85$  befindet, und nachfolgender *false nearest neighbour procedure* [30] mit Hilfe des Programms von Wallot et al. [35] berechnet und beträgt gerade 3. Die Zeitschrittweite  $\tau = 85$  wäre in Anbetracht dessen, dass der Fehler des Differenzenquotienten mit der

Tabelle 1: Vergleich der Güte der Vorhersage eines klassischen NARXnet für verschiedene Zeitschrittweiten  $\tau$ . Die Netze werden mit den Trainingsdaten mit einer Rauschstärke von  $\nu = 0.2$  trainiert und die Vorhersage an den Testdaten mit einer Rauschstärke von  $\nu = 0$  getestet.

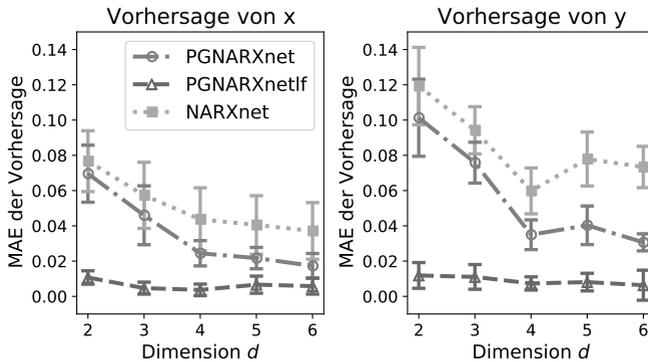
Zeitschrittweite $\tau$	MAE der Vorhersage von $x$	MAE der Vorhersage von $y$
1	$0.0314 \pm 0.0086$	$0.0288 \pm 0.0064$
85	$0.0221 \pm 0.0121$	$0.0401 \pm 0.0148$

Zeitschrittweite ansteigt, zu groß und daraus resultiert ein großer Fehler in Hinblick auf die Abschätzung der Ableitungen mit dem Differenzenquotienten. Deswegen wurde die Entscheidung getroffen, keine zusätzliche Verschiebung zu verwenden. Die Vorhersage des NARXnet in Bezug auf die Variable  $x$  wird dadurch etwas schlechter, wohingegen der mittlere absolute Fehler der Vorhersage des NARXnet in Bezug auf die Variable  $y$  sogar abnimmt (siehe Tabelle 1). Aufgrund der Ergebnisse in der gleichen Größenordnung ist die Entscheidung,  $\tau = 1$  zu wählen, legitim.

Zum Training der Netze erfolgt eine Normalisierung der Daten auf den Bereich  $[-1, 1]$ . Als Aktivierungsfunktion wurde die hyperbolische Tangensfunktion verwendet. Die Netze wurden jeweils mit fünf verschiedenen zufälligen Initialisierungen trainiert. Die Ergebnisse zeigen jeweils die Mittelwerte mit den zugehörigen Standardabweichungen. Die Rechnungen wurden mit der *TensorFlow*-Bibliothek [36] und der darauf aufbauenden *Keras*-Bibliothek [37] ausgeführt. Weiterhin sollte erwähnt werden, dass keine vollständige Optimierung des Netzwerks durchgeführt wurde. Im Vordergrund dieser Arbeit waren besagte Lösungsansätze zu testen und miteinander bzw. mit dem Standardansatz zu vergleichen. Um jedoch die beste *Embedding-Dimension* für den Versuch zu finden, wurde diese in den Tests variiert (siehe Abbildung 4). Trainiert wurden die Netzwerke mit einer Rauschstärke  $\nu$  von 0.2 und der mittlere absolute Fehler anhand der Testdaten wie auch der Zielzeitreihe mit einer Rauschstärke von  $\nu = 0$  bestimmt, um das Erlernen der Funktion während des Trainings zu überprüfen.



(a) Testdaten der Trainingszeitreihe:  $A = 1$  und  $\varepsilon = 1$



(b) Test der Extrapolationsleistung auf neue Daten:  $A = 1$  und  $\varepsilon = 0.1$

Bild 4: Mittlerer absoluter Fehler der Vorhersage in Abhängigkeit der Dimension  $d$ . Die Rauschstärke  $\nu$  auf den Trainings- und Validierungsdaten betrug 0.2 und die Rauschstärke auf den Testdaten 0.

Mit steigender *Embedding-Dimension*  $d = 2 \dots 4$  lässt sich für das NARXnet und das PGNARXnet eine Abnahme des mittleren absoluten Fehlers der Vorhersage feststellen. Der mittlere absolute Fehler der Vorhersage von  $y$  steigt für größere *Embedding-Dimension* sowohl für die Testdaten wie auch die Zielzeitreihe mit neuen Koeffizienten an. Hingegen lässt sich eine weitere Abnahme des mittleren absoluten Fehlers für die Vorhersage von  $x$  feststellen. Das PGNARXnet erzielt für alle *Embedding-Dimensions* im Vergleich zum NARXnet einen geringeren Fehler. Durch die Ergänzung des Physik-geführten Ver-

lustterms erfolgt eine Regularisierung sowie eine Einschränkung des Lösungsraums, sodass diese im Vergleich zu den klassischen NARXnets das Differentialgleichungssystem besser erlernen. Von den diskutierten Verläufen weicht der mittlere absolute Fehler des PGNARXnetf ab. Vor allem bei geringen *Embedding-Dimensions* werden hier bereits sehr gute Werte erzielt. Bei der Bestimmung der *Embedding-Dimension* wird deutlich, die Extrapolationsleistung des PGNARXnetf ist im Vergleich zu dem klassischen NARXnet-Ansatz wie auch dem PGNARXnet-Ansatz verbessert (siehe Abbildung 4b).

Die Wahl der *Embedding-Dimension*  $d = 3$  fiel aufgrund des im Mittel besten Ergebnis für das PGNARXnetf in Bezug auf die Testdaten. Anhand dieses beispielhaften Netzes werden die nachfolgenden Untersuchungen durchgeführt.

## 4.1 Performance der Verlustfunktionen

Bei der Diskussion der Verlustfunktionen sind zwei verschiedene Verluste zu unterscheiden: Die Trainings- und Validierungsverluste. Die jeweiligen Verlustfunktionen werden bei den Trainings- und Validierungsverlusten nach jeden Batch ausgewertet. Nach jeder Epoche wird der Mittelwert dieser Verlustwerte berechnet. Die Trainings- und Validierungsverluste unterscheiden sich allein durch die Daten, an welchen diese berechnet werden. Anhand der Trainingsverluste wird das Netz optimiert. Überprüft wird der Optimierungsfortschritt wie auch die Verallgemeinerbarkeit der Ergebnisse durch den Validierungsfehler. Da die prinzipielle Entwicklung der Verlustfunktionen sehr ähnlich sind, wobei die Trainingsverlustwerte typischerweise niedriger ausfallen, sollen an dieser Stelle die Entwicklung der unterschiedlichen Validierungsfehler der verschiedenen Netzarten am Beispiel der *Embedding-Dimension* von  $d = 3$  und dem Training mit einer Rauschstärke von  $v = 0.2$  diskutiert werden (siehe Abbildung 5).

Vergleicht man die Verläufe der unterschiedlichen Netzarten PGNARXnet, PG-NARXnetf und NARXnet, unterscheiden sich diese stark voneinander. Der Validierungsfehler der NARXnet  $Loss_{NARXnet}$  nimmt nur minimal ab und erreicht bereits nach 114 Epochen sein Minimum. Im Gegensatz dazu verringert sich der Validierungsfehler des PGNARXnetf  $Loss_{PGNARXnetf}$  über die 15000 Epochen und erreicht innerhalb dieses Trainingsabschnitts

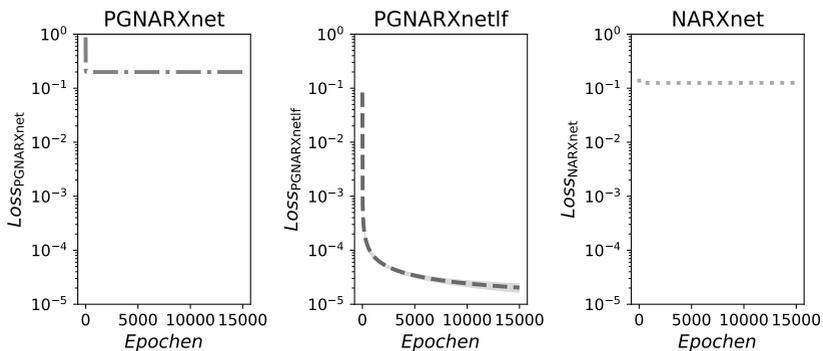


Bild 5: Vergleich der Entwicklung der unterschiedlichen Validierungsfehler der verschiedenen Netzarten. Dargestellt sind die Mittelwerte mit Standardabweichung der Netze mit einer *Embedding-Dimension* von  $d = 3$  und dem Training mit einer Rauschstärke von  $v = 0.2$ .

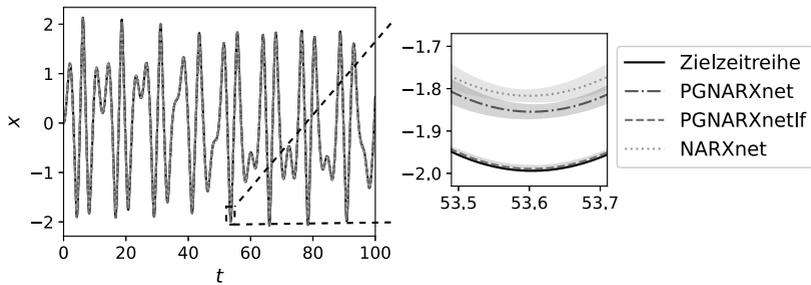
kein Minimum. Bedingt durch den Ausgleich der verschiedenen Verlustterme und die dominierenden klassischen Verluste lässt sich beim PGNARXnet  $Loss_{PGNARXnet}$  ein ähnliches Verhalten wie beim NARXnet feststellen. Ein Minimum lässt sich nach 256 Epochen beobachten.

Anhand dieser Ergebnisse liegt die Vermutung nahe, dass die Physik-geführten NARXnets und dabei primär das PGNARXnetlf die zugrundeliegende Funktion im Vergleich zu den anderen Netzen besser erlernt.

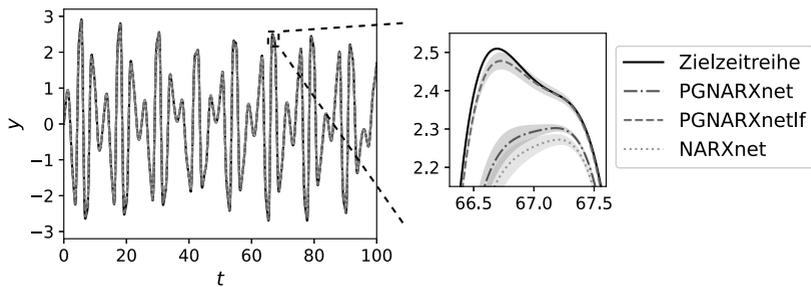
## 4.2 Vorhersage

Die Vorhersage spiegelt das in vorigen Abschnitt angedeutete Verhalten wider. In Abbildung 6 ist beispielhaft die Vorhersage anhand der Zielzeitreihe mit im Vergleich zur Trainingszeitreihe abgeänderten Koeffizienten für die besagte Netzarchitektur mit einer *Embedding-Dimension*  $d = 3$  gezeigt. Im Vergleich zum Training wurde die Rauschstärke der Zielzeitreihe auf  $v = 0$  gesetzt. Somit kann anhand der Vorhersage überprüft werden, inwiefern das zugrundeliegende Differentialgleichungssystem durch das neuronale Netz erlernt wird.

Anhand der kompletten Zeitreihe lassen sich die Unterschiede der Vorhersage nur schwierig abschätzen. Schaut man sich doch gerade die lokalen Minima und Maxima der modifizierten Van der Pol Gleichung an, stellt man fest, dass



(a) Vorhersage von  $x$



(b) Vorhersage von  $y$

Bild 6: Vergleich der Vorhersagen anhand der Zielzeitreihe mit den Koeffizienten:  $A = 1$ ,  $\varepsilon = 0.1$  und einer Rauschstärke von  $v = 0$ . Ziel ist der Test der Extrapolationsfähigkeit.

die Vorhersage des NARXnet am stärksten von der Zielzeitreihe abweicht. Auch die Vorhersage des PGNARXnet weist eine vergleichsweise große Abweichung auf, wobei diese im Vergleich zur Vorhersage des NARXnet geringer ist. Dies resultiert aus der Ergänzung der Physik-geführten Verlustterme, welche hier als eine Art Regularisierung wirken. Mit Abstand die beste Vorhersage für das gewählte Beispiel zeigt das PGNARXnetlf. Tabelle 2 fasst diese Beobachtungen zusammen. Das besagte Beispiel zeigt: Durch die Vorgabe der Differentialgleichungen des zugrundeliegenden Systems kann die Extrapolationsfähigkeit des PGNARXnetlf gegenüber dem klassischen NARXnet erhöht werden, da die Dynamik des Systems explizit durch die Vorgabe der zeitlichen Ableitungen in den Differentialgleichungen und dem Zusammenhang der Ableitung mit den zurückliegenden Funktionswerten aufgezwungen wird.

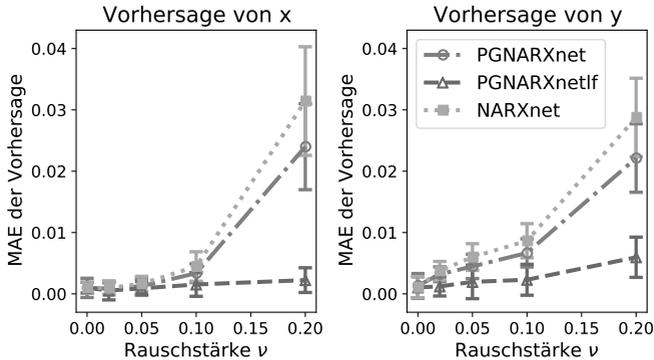
Tabelle 2: Vergleich der mittleren absoluten Fehler der Vorhersage von  $x$  und  $y$ . Die Netze werden mit den Trainingsdaten mit einer Rauschstärke von  $\nu = 0.2$  trainiert und die Vorhersage an der Zielzeitreihe ( $A = 1$  und  $\varepsilon = 0.1$ ) mit einer Rauschstärke von  $\nu = 0$  getestet.

	Vorhersage von $x$	Vorhersage von $y$
$MAE_{\text{PGNARXnet}}$	$0.0460 \pm 0.0167$	$0.0758 \pm 0.0116$
$MAE_{\text{PGNARXnetlf}}$	$0.0047 \pm 0.0034$	$0.0110 \pm 0.0070$
$MAE_{\text{NARXnet}}$	$0.0573 \pm 0.0187$	$0.0942 \pm 0.0134$

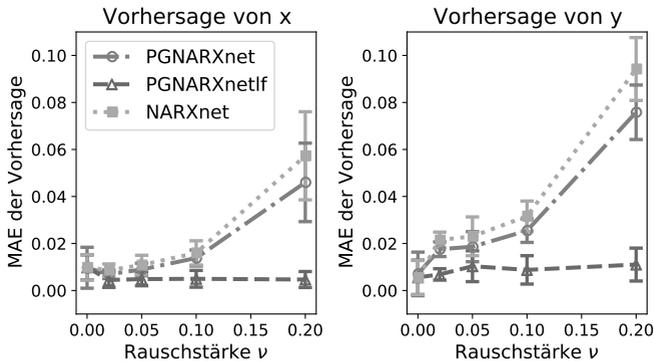
Für das identische Netzwerk wurde weiterhin die Rauschstärke  $\nu$  auf der Trainingszeitreihe verändert und der mittlere absolute Fehler der Vorhersage anhand der Testdaten sowie der Zielzeitreihe mit veränderten Koeffizienten bei einer Rauschstärke von  $\nu = 0$  bestimmt (siehe Abbildung 7). Bei geringen Rauschstärken auf den Trainings- und Validierungsdaten zeigen alle drei verschiedenen Netzarten ähnliche Ergebnisse. Mit steigender Rauschstärke nimmt der mittlere absolute Fehler der Vorhersage des PGNARXnet wie auch des NARXnet deutlich zu, wobei der mittlere absolute Fehler des NARXnet größer ist als der des PGNARXnet. Der mittlere absolute Fehler der Vorhersage für das PGNARXnetlf steigt dagegen nur leicht an bzw. bleibt annähernd konstant.

## 5 Fazit

Die Physik-geführten NARXnets weisen für analytisch bzw. numerisch berechnete Daten ähnliche Fehlerwerte wie die klassischen NARXnets auf. Der Vorteil besteht jedoch darin, dass die Konsistenz der zugrundeliegenden Physik durch die Physik-geführten Verlustterme gesichert wird. Erhöht man das Rauschen auf den Trainingsdaten, lernen die Physik-geführten NARXnets das zugrundeliegende Differentialgleichungssystem besser. Vor allem spiegelt sich dies bei dem Übertrag der Netze auf neue Zeitreihen mit gleichem Differentialgleichungssystem aber veränderten Koeffizienten wider. Die Extrapolationsleistung kann somit durch die Physik-geführten Verlustterme verbessert werden. Weiterhin ermöglichen die Physik-geführten NARXnets, welche durch die alleinige Nutzung der Physik-geführten Verlustterme trainiert



(a) Testdaten der Trainingszeitreihe:  $A = 1$  und  $\varepsilon = 1$



(b) Test der Extrapolationsleistung auf neue Daten:  $A = 1$  und  $\varepsilon = 0.1$

Bild 7: Mittlerer absoluter Fehler der Vorhersage in Abhängigkeit der Rauschstärke  $\nu$  auf den Trainings- und Validierungsdaten. Die Vorhersage wurde an nicht-verrauschten Daten getestet, um das Erlernen des Differentialgleichungssystems zu prüfen.

werden, sogenanntes *label-free learning*. Außerdem wirkt der Physik-geführte Verlustterm bei den PGNARXnets als eine Art Regularisierung und reduziert zugleich das Problem der Überanpassung.

Um die Vorhersage unter Vorgabe verrauschter Daten zu verbessern, sind derzeit weiterführende Arbeiten geplant. Die Idee ist, zur Optimierung Differenzenquotienten, welche stabil gegenüber Rauschen sind, zu verwenden und einhergehend dazu mehr Stützstellen für diese Differenzenquotienten zu berücksichtigen. Beispielsweise wäre eine Vorschlag Savitzky-Golay-

Differentiatoren zu nutzen, welche auf dem Savitzky-Golay-Filter [38] basieren.

## Literatur

- [1] R. K. Jain, K. M. Smith, P. J. Culligan und J. E. Taylor. „Forecasting energy consumption of multi-family residential buildings using support vector regression: Investigating the impact of temporal and spatial monitoring granularity on performance accuracy“. In: *Applied Energy*, Band 123, S. 168-178. 2014. <https://doi.org/10.1016/j.apenergy.2014.02.057>
- [2] T. S. Hu, K. C. Lam und S. T. Ng. „River flow time series prediction with a rangedependent neural network“. In: *Hydrological Sciences Journal*, Band 45, Nr. 5, S. 729-745. 2001. <https://doi.org/10.1080/02626660109492867>
- [3] R. Trippi und E. Turban. „Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real-World Performance“. Burr Ridge Illinois: Irwin Professional Publishing Co. 1996. [https://doi.org/10.1016/S0169-2070\(96\)00728-5](https://doi.org/10.1016/S0169-2070(96)00728-5)
- [4] W. Hunag, K. K. Lai, Y. Nakamori, S. Wang und L. Yu. „Neural networks in finance and economics forecasting“. In: *International Journal of Information Technology & Decision Making*, Band 6, Nr. 1, S. 113-140. 2007. <https://doi.org/10.1142/S021962200700237X>
- [5] S. Chen, S. A. Billings und P. M. Grant. „Non-linear system identification using neural networks“. In: *International Journal of Control*, Band 51, Nr. 6, S. 1191-1214. 1990. <https://doi.org/10.1080/00207179008934126>
- [6] X. Zhang. „Time series analysis and prediction by neural networks“. In: *Optimization Methods and Software*, Band 4, Nr. 2, S. 151-170. 1994. <https://doi.org/10.1080/10556789408805584>

- [7] A. Karpatne, G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova und V. Kumar. „Theory-Guided Data Science: A New Paradigm for Scientific Discovery from Data“. In: *IEEE Transactions on Knowledge and Data Engineering*, Band 29, Nr. 11, S. 2318-2331. 2017. <https://doi.org/10.1109/TKDE.2017.2720168>
- [8] M. A Kraus und A. Taras. „Physik-informierte Künstliche Intelligenz zur Berechnung und Bemessung im Stahlbau“. In: *Stahlbau*, Band 89, Nr. 10, S. 824-832. 2020. <https://doi.org/10.1002/stab.202000074>
- [9] A. Karpatne, W. Watkins, J. Read und V. Kumar. „Physics-guided Neural Networks (PGNN): An Application in Lake Temperature Modeling“. *arXiv-preprint*. 2018. arXiv:1710.11431v2
- [10] A. Krizhevsky, I. Sutskever und G. E. Hinton. „Imagenet classification with deep convolutional neural networks“. *Proceedings Advances in Neural Information Processing Systems* 25, S. 1090-1098. 2012. <https://doi.org/10.1145/3065386>
- [11] G. E. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath und B. Kingsbury. „Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups“. *IEEE Signal Processing Magazine*, Band 29, Nr. 6, S. 82-97. 2012. <https://doi.org/10.1109/MSP.2012.2205597>
- [12] M. Raissi, P. Perdikaris und G. E. Kariadakis. „Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations“. In: *Journal of Computational Physics*, Band 378, S. 686-707. 2019. <https://doi.org/10.1016/j.jcp.2018.10.045>
- [13] S. Karimpouli und P. Tahmasebi. „Physics informed machine learning: Seismic wave equation“. In: *Geoscience Frontiers*, Band 11, Nr. 2, S. 1196-2001. 2020. <https://doi.org/10.1016/j.gsf.2020.07.007>

- [14] Y. Chen, L. Lu, G. E. Karniadakis und L. D. Negro. „Physics-informed neural networks for inverse problems in nano-optics and metamaterials“. In: *Optics Express*, Band 28, Nr. 8, S. 11618-11633. 2020. <https://doi.org/10.1364/OE.384875>
- [15] L. Lu, X. Meng, Z. Mao und G. E. Karniadakis. „DEEPXDE: A deep learning library for solving differential equations“. *arXiv-preprint*. 2020. arXiv:1907.04502v2
- [16] M. Elhamod, J. Bu, C. Singh, M. Redell, A. Ghosh, V. Podolskiy, W.-C. Lee und A. Karpatne. „CoPhy-PGNN: Learning Physics-guided Neural Networks with Competing Loss Functions for Solving Eigenvalue Problems“. *arXiv-preprint*. 2020. arXiv:2007.01420v4
- [17] R. Zhang, Y. Liu und H. Sun. „Physics-guided Convolutional Neural Network (PhyCNN) for Data-driven Seismic Response Modeling“. *arXiv-preprint*. 2019. arXiv:1909.08118v1
- [18] T. Parditia, T. Walser, S. Oladyshkin und W. Nowak. „Improving Thermochemical Energy Storage Dynamics Forecast with Physics-Inspired Neural Network Architecture“. In: *Energies*, Band 13, Nr. 15, S. 3873-3899 2020. <https://doi.org/10.3390/en13153873>
- [19] M. Bolderman, M. Lazar und H. Butler. „Physics-Guided Neural Networks for Inversion-based Feedforward Control applied to Linear Motors“. *arXiv-preprint*. 2021. arXiv:2103.06092v1
- [20] B. Fornberg. „Generation of Finite Difference Formulas on Arbitrarily Spaced Grids“. In: *Mathematics of Computation*, Band 51, Nr. 184, S. 699-706. 1988. <https://doi.org/10.1090/S0025-5718-1988-0935077-0>
- [21] C. Taylor. „Finite Difference Coefficients Calculator“. Massachusetts Institute of Technology, <https://web.media.mit.edu/~crtaylor/calculator.html>. 2019.
- [22] B. van der Pol. „On relaxation oscillations“. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, Band 2, S. 978-992. 1926. <https://doi.org/10.1080/14786442608564127>

- [23] Y. H. Ku und X. Sun. „CHAOS in Van der Pol’s Equation“. In: *Journal of the Franklin Institute, Pergamon Press plc*, Band 327, Nr. 2, S. 197-207. 1990. [https://doi.org/10.1016/0016-0032\(90\)90016-C](https://doi.org/10.1016/0016-0032(90)90016-C)
- [24] Y. Ueda und N. Akamatsu. „Chaotically Transitional Phenomena in the Forced Negative-Resistance Oscillator“. In: *IEEE Transactions on Circuits and Systems*, Band 28, Nr. 3, S. 217-224. 1981. <https://doi.org/10.1109/TCS.1981.1084975>
- [25] J. R. Dormand and P. J. Prince. „A family of embedded Runge-Kutta formulae“. In: *Journal of Computational and Applied Mathematics*, Band 6, Nr. 1, S. 19-26. 1980. [https://doi.org/10.1016/0771-050X\(80\)90013-3](https://doi.org/10.1016/0771-050X(80)90013-3)
- [26] S. J. Reddi, S. Kale und S. Kumar. „On the Convergence of Adam and Beyond“. *arXiv-preprint*. 2018. arXiv:1904.09237
- [27] T. Masters. „Practical Neural Network Recipes in C++“. San Diego: Academic Press, Inc. 1993.
- [28] R. J. Frank, N. Davey und S. P. Hunt. „Time Series Prediction and Neural Networks“. In: *Journal of Intelligent and Robotic Systems*, Band 31, S. 91-103. 2001. <https://doi.org/10.1023/A:1012074215150>
- [29] J. M. P. Menezes und G. A. Barreto. „Long-term time series prediction with the NARX network: An empirical evaluation“. In: *Neurocomputing*, Band 71, Nr. 16-18, S. 3335-3343. 2008. <https://doi.org/10.1016/j.neucom.2008.01.030>
- [30] M. B. Kennel, R. Brown und H. D. I. Abarbanel. „Determining embedding dimension for phase-space reconstruction using a geometrical construction“. In: *Physical Review A*, Band 45, Nr. 6, S. 3403-3411. 1992. <https://doi.org/10.1103/PhysRevA.45.3403>
- [31] A. N. Kolmogorov. „On the Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition“. In: *Doklady Akademii Nauk SSSR*, Band 144, S. 679-681. 1957. *American Mathematical Society Translation*, Nr. 28, S. 55-59. 1963.

- [32] R. Hecht-Nielsen. „Kolmogorov’s Mapping Neural Networks Existence Theorem“. In: *First IEEE International Conference on Neural Networks*, San Diego, Band 3, S. 11-14. 1987.
- [33] A. M. Fraser und H. L. Swinney. „Independent coordinates for strange attractors from mutual information“. In: *Physical Review A*, Band 33, Nr. 2, S. 1134-1140. 1986. <https://doi.org/10.1103/PhysRevA.33.1134>
- [34] W. Liebert und H. G. Schuster. „Proper choice of the time delay for the analysis of chaotic time series“. In: *Physics Letters A*, Band 142, Nr. 2-3, S. 107-111. 1989. [https://doi.org/10.1016/0375-9601\(89\)90169-2](https://doi.org/10.1016/0375-9601(89)90169-2)
- [35] S. Wallot und D. Møster. „Calculation of Average Mutual Information (AMI) and False-Nearest Neighbors (FNN) for the Estimation of Embedding Parameters of Multidimensional Time Series in Matlab“. In: *Frontiers in Psychology*, Band 9, S. 1679-1689. 2018. <https://doi.org/10.3389/fpsyg.2018.01679>
- [36] M. Abadi et al. „TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems“. <https://www.tensorflow.org/>. 2015.
- [37] F. Chollet et al. „Keras“. <https://keras.io>. 2015.
- [38] A. Savitzky und J. E. Golay. „Smoothing and Differentiation of Data by Simplified Least Squares Procedures“. In: *Analytical Chemistry*, Band 36, Nr. 8, S. 1627-1639. 1964. <https://doi.org/10.1021/ac60214a047>