

Neue Methoden zur Steuerung der Wassergabe mit neuronalen Netzen in der Bewässerungslandwirtschaft

Niels Schütze

Institut für Hydrologie und Meteorologie
Technische Universität Dresden
ns1@rcs.urz.tu-dresden.de

Abstract: Im Wettbewerb um die knappe Ressource Wasser steht die Bewässerungslandwirtschaft als größter Verbraucher mit dem geringsten Wirkungsgrad neben allen anderen Konsumenten. Zur Verbesserung des Wirkungsgrads ist man in der Bewässerungslandwirtschaft auf simulationsbasierte Optimierungsmethoden angewiesen. Bei der Anwendung der simulationsbasierten Optimierung stößt man bei allen Bewässerungsverfahren auf Probleme, da die numerischen Simulationsmodelle komplex, aufwändig und mitunter auch instabil sind. Zudem erschwert die nichtlineare Zielfunktion eine optimale Steuerung in Echtzeit. An Stelle der problematischen direkten Minimierung einer Zielfunktion mit numerischen Bewässerungsmodellen wird in dieser Arbeit ein zweiphasiges Vorgehen vorgeschlagen, welches die Strömungsmodellierung und die optimale Bestimmung der Bewässerungsparameter entkoppelt. Dafür wurde eine neue neuronale Netzarchitektur, die selbstorganisierende Merkmalskarte mit variabler Ein-/Ausgabefunktion (SOM-MIO), entwickelt. In der Arbeit wird die SOM-MIO Architektur zusammen mit ein-, zwei- und dreidimensionalen Strömungsmodellen innerhalb der zweiphasigen Optimierungsstrategie verwendet. Umfangreiche Anwendungen der SOM-MIO Architektur erfolgten zur deterministischen und stochastischen Steuerung der Tropfbewässerung sowie zur Steuerung der Furchenbewässerung bei bedarfsgerechter Bewässerungsplanung und bei Defizitbewässerung.

1 Der Hintergrund der Arbeit

Durch die Steuerung eines Bewässerungsvorganges auf einem Feld kann mit den unterschiedlichen Bewässerungsverfahren (siehe Abb.1) die räumliche Verteilung der für ein Bewässerungsereignis zur Verfügung stehenden Wassermenge auf und schließlich im Boden gezielt beeinflusst werden. Die Ausbreitung des Wassers wird durch die Parameter Intensität (oder Zuflußrate) und die Bewässerungszeit gesteuert. Eine schlechte Steuerung der Wasserverteilung hat eine Reihe von negativen Folgen. Beispielsweise führt eine zu große Intensität bei allen Verfahren zu hohem oberirdischen Abfluß, d.h. zu großen Verlusten, weil das Wasser vom Feld wegfließt. Andererseits kann eine zu geringe Intensität der Wasseraufleitung z.B. bei der Furchenbewässerung eine sehr ungleiche Verteilung des Wassers längs des Feldes bewirken. Eine gute Steuerung der Wasserverteilung erreicht hingegen eine gleichmäßige Befeuchtung des Pflanzenwurzelraums ohne Verluste von Wasser durch Abfluß oder Perkolation.

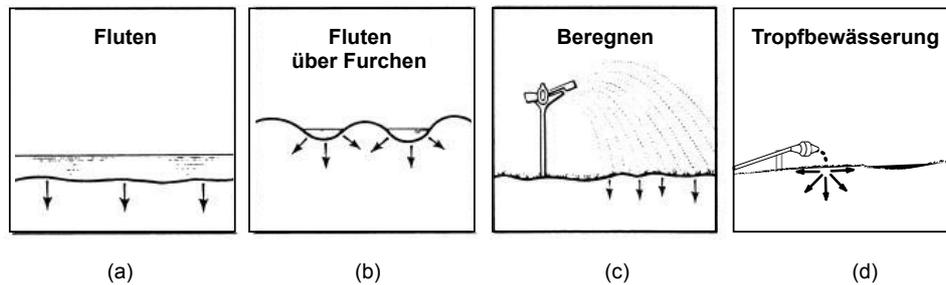


Abbildung 1: Die wichtigsten Bewässerungsverfahren (nach [Ach80]).

In Gebieten, in denen die Bewässerungslandwirtschaft eine lange Tradition besitzt und eine bestimmte soziale Kontinuität gewahrt werden konnte, führt heute noch die Anwendung überlieferter Bewässerungsregeln zu erstaunlich guten Ergebnissen. Bei der Neuerschließung, Umstrukturierung oder Ausweitung bewässerter Anbauflächen wird seit mehreren Jahrzehnten versucht, den Mangel dieses umfassenden empirischen Wissens durch den Einsatz von Optimierungsalgorithmen in Verbindung mit Simulationsmodellen auszugleichen.

Gegenwärtig ist man nur mit numerischen Prozeßmodellen¹ in der Lage, die Wasserbewegung auf dem Feld und im Boden für die unterschiedliche Bewässerungsverfahren (siehe Abb.1) in geeigneter Weise zu simulieren. Dadurch kann man auf teure Feldexperimente verzichten, die sehr langwierig (Tage und Wochen) sind oder zu irreversiblen Schäden (z.B. Versalzung durch Grundwasseranstieg bei Überbewässerung) führen können. Leider ist die Berechnung von Bewässerungsszenarien mit numerischen Prozeßmodellen relativ aufwändig. Beim Einsatz iterativer Optimierungsverfahren oder bei Untersuchungen von stochastischen Einflüssen mittels Monte-Carlo Simulation ergeben sich Berechnungszeiten von Stunden bis Tagen, die in der Bewässerungspraxis nicht akzeptiert werden. Außerdem erfordert die Arbeit mit den Simulationswerkzeugen spezifische mathematische Kenntnisse, um numerische Instabilitäten bei den Simulationen von Bewässerungsvorgängen zu vermeiden.

In der Praxis wird daher häufig auf vereinfachte Modelle zurückgegriffen, die lediglich die Wasserbewegung auf dem Feld simulieren und die infiltrierte Wassermenge abschätzen [McC00, RW97, ZSFP01]. Auf die schwierigere Simulation der Dynamik der Wasserbewegung im Boden wird verzichtet. Dadurch kann man jedoch keine Informationen darüber erhalten, ob das infiltrierte Wasser sich hauptsächlich in den Pflanzenwurzelraum bewegt oder in für die Pflanze unereichbare Bodenschichten perkoliert.

Deshalb wird in dieser Arbeit eine neue Strategie zur optimalen Steuerung der Wassergabe entwickelt, die die komplexen Prozeßmodelle in der Bewässerungslandwirtschaft praktisch anwendbar macht, ohne deren Nachteile in Kauf nehmen zu müssen.

¹Diese Modelle wurden auf der Grundlage der physikalischen Gesetzmäßigkeiten der Wasserbewegung, die bei der Bewässerung durch Schwerkraft, Druckkraft und Kapillarkraft verursacht wird, abgeleitet.

2 Das gekoppelte Furchenbewässerungsmodell FIM

Das am Institut für Hydrologie und Meteorologie der TU Dresden entwickelte Bewässerungsmodellsystem FIM² erlaubt die physikalisch begründete Simulation der ober- und unterirdischen Strömungsvorgänge bei der Furchenbewässerung oder Überflutung auf eines Feldes in hoher zeitlicher und räumlicher Auflösung. Zudem bildet es auch alle anderen Prozesse des Wassertransportes und der -speicherung (Niederschlag, Verdunstung, Transpiration, Austausch mit dem Grundwasser) und des Pflanzenwachstums (Biomasseentwicklung, Wurzelwachstum) über eine gesamte Wachstumsperiode detailgetreu ab.

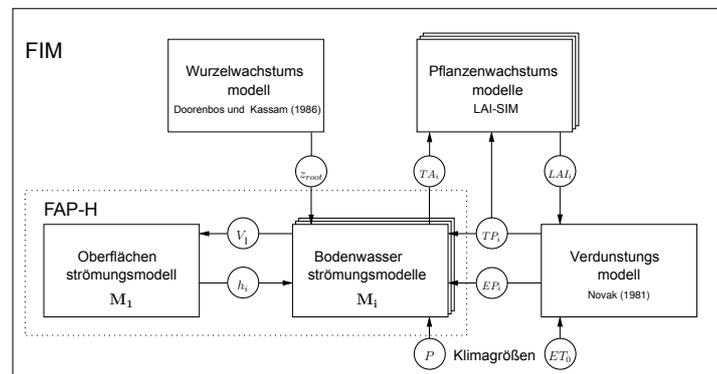


Abbildung 2: Komponenten des Furchenbewässerungsmodells FIM.

Wie in Abbildung 2 zu sehen, sind in der Komponente FAP-H des Gesamtmodells FIM ein beschleunigungsfreies Oberflächentrömungsmodell und mehrere Instanzen des Bodenwasserströmungsmodells Hydrus-2D [vsvG96] zur dreidimensionalen Simulation eines Bewässerungsvorgangs miteinander gekoppelt. Das Oberflächenströmungsmodell M_1 berechnet mit $\vec{h} = (h_2, \dots, h_i, \dots, h_n)$ den Wasserstand als obere Randbedingung für die einzelnen Hydrus-2D-Modelle $M_2 \dots M_n$ und umgekehrt ist das entlang der Furche infiltrierte Volumen V_I , welches durch die Bodenwasserströmungsmodelle bestimmt wird, eine Eingangsgröße für das Oberflächenströmungsmodell. Die Modelle $M_1 \dots M_n$ bilden ein nichtlineares Gleichungssystem

$$\begin{aligned} y_1 &= \vec{h} = (h_2, \dots, h_i, \dots, h_n) = M_1(y_2, \dots) \\ y_2 &= \int_{i=2}^n M_i(h_i, \dots), \end{aligned} \quad (1)$$

welches für jeden Zeitschritt gelöst werden muß. Um den Berechnungsaufwand des gekoppelten Strömungsmodells FAP-H in Grenzen zu halten, wurde neben Hydrus-2D ein analytisches Oberflächenströmungsmodell von [SS92] eingesetzt. Dennoch dauert die Simulation der Wasserbewegung auf dem Feld und im Boden während einer Wachstumsperiode auf einem Standard-PC etwa 1 Stunde.

²Eine ausführliche Darstellung des Modellsystems ist in [W05] zu finden.

3 Das Zusammenspiel von Bewässerungsmodell und KNN

Die angesprochenen Probleme, die sich bei der simulationsbasierten Optimierungsverfahren ergeben, werden durch ein neues zweiphasiges Vorgehen überwunden. Dabei spielen zwei Hauptakteure zusammen. Den einen verkörpern die physikalisch begründeten Strömungsmodelle, die eine wirklichkeitsgetreue Beschreibung aller möglichen (also auch bisher nicht stattgefundenen) Bewässerungsvorgänge gewährleisten. Für das Erlernen der günstigsten Bewässerungsparameter ist dagegen der zweite Akteur, ein künstliches neuronales Netz (KNN), verantwortlich. Im ersten Schritt generieren die mit lokalen Daten betriebenen Prozeßmodelle eine umfassende Palette von Bewässerungsszenarien. Mit Hilfe der so gewonnenen Zustandsgrößen und Input-/Outputdatensätzen, trainiert das neuronale Netz das Verhalten des zu untersuchenden Bewässerungssystems. Im zweiten Schritt liefert der Einsatz des neuronalen Netzes allein vor Ort die optimalen Bewässerungsparameter für das Feld, für das es den Trainingsprozeß durchlaufen hat, d.h. es ermittelt unter Berücksichtigung der lokalen Zustandsgrößen die Steuerparameter für einen Bewässerungsvorgang, der ein für die Pflanze in der jeweiligen Wachstumsperiode günstigstes Feuchteprofil erzeugt.

Das zweiphasige Vorgehen kann für weitere Aufgaben (z.B. Monte-Carlo Simulationen) sinnvoll sein. Im folgenden werden deshalb neben den Lernstrategien zur Bestimmung der optimalen Bewässerungsparameter (inverse Modellierung und nichtlineare Optimierungsverfahren) auch Strategien für die Simulation von Bewässerungsvorgängen (Vorwärts-Modellierung) vorgestellt.

3.1 Vorwärts-Modellierung

Wenn neuronale Netze die numerischen Bewässerungsmodelle ausschließlich zur Simulation des Wassertransportes auf und im Boden ersetzen sollen, dann wird diese Aufgabe als Vorwärts-Modellierung bezeichnet. Für die Vorwärts-Modellierung muß das neuronale Netz die folgende Abbildung f :

$$f : (\vec{x}_{ini}, \vec{x}_{bound}, t) \longrightarrow \vec{y} \quad (2)$$

mit den Anfangsbedingungen \vec{x}_{ini} ³, den definierten Randbedingungen \vec{x}_{bound} ⁴ und t , der Bewässerungszeit erlernen. Für diese Aufgabe wird das neuronale Netz in der Phase des Lernvorganges parallel zum Bewässerungsmodell arbeiten und berechnet in jedem Lernschritt ein Ergebnis \vec{y}' , das mit dem Ergebnis des Bewässerungsmodells \vec{y} ⁵ verglichen wird. Überwachte Lernverfahren versuchen im Laufe des Lernvorganges durch Veränderung der Gewichte des neuronalen Netzes den Fehler $\|\vec{y} - \vec{y}'\|$ zu minimieren.

Die Menge der Trainingsdaten entsteht durch zufällige oder systematische Variation der Elemente des Inputdatensatzes $(\vec{x}_{ini}, \vec{x}_{bound}, t)$. Die Wertebereiche der einzelnen Elementen

³ \vec{x}_{ini} beschreibt u.a. die Feuchteverteilung vor der Bewässerung im Feldboden.

⁴ \vec{x}_{bound} beschreibt u.a. die hydraulischen Randbedingungen wie den Zufluß des Wassers auf das Feld.

⁵ \vec{y} beschreibt u.a. die Feuchteverteilung nach der Bewässerung im Feldboden.

te bestimmen später den Geltungsbereich des angelernten neuronalen Netzes und die Kardinalität der Trainingsmenge n_D u.a. die Güte der Approximation von $f(\vec{x}) \rightarrow \vec{y}$, also die Güte der Simulation der Bewässerungsvorgänge durch das neuronale Netz.

3.2 Optimale Steuerung mit inverser Modellierung

Für die Steuerung von Bewässerungsvorgängen, insbesondere bei der Tropfbewässerung, ist die inverse Modellierung durch neuronaler Netze attraktiv. Dafür muß das neuronale Netz eine zu der Abbildung

$$\vec{y} = f(\vec{x}_1, \vec{x}_2, \vec{x}_3, \dots) \quad (3)$$

inverse Funktion

$$\vec{x}_i = f_i^{-1}(\vec{x}_1, \dots, \vec{x}_{i-1}, \vec{x}_{i+1}, \dots, \vec{y}) \quad (4)$$

erlernen. Aus Gleichung 2 ergeben sich die folgenden drei möglichen Abbildungen: $f_1^{-1} : (\vec{y}, \vec{x}_{ini}, t) \rightarrow \vec{x}_{bound}$, $f_2^{-1} : (\vec{y}, \vec{x}_{bound}, t) \rightarrow \vec{x}_{ini}$ und $f_3^{-1} : (\vec{y}, \vec{x}_{ini}, \vec{x}_{bound}) \rightarrow t$. f_1^{-1} und f_3^{-1} führen zu den optimalen Steuerparametern für die einzelnen Bewässerungsverfahren, falls die inverse Funktion eindeutig ist. Gegenüber der Lernstrategie bei der Vorwärts-Modellierung werden dafür die Belegungen der Ein- und Ausgänge am neuronalen Netz vertauscht, so daß das ANN die Funktion f^{-1} abbilden kann.

3.3 Optimale Steuerung bei Mehrdeutigkeit

Die inverse Modellierung ist für die Bewässerungssteuerung nicht mehr anwendbar, wenn die inverse Abbildung nicht eindeutig ist, d.h., wenn mehrere Parameterkombinationen existieren, die den gleichen Endzustand \vec{y} im Boden einstellen können. Dies trifft im besonderen auf die Furchenbewässerung zu. Zur Lösung dieser Aufgabe mittels neuronaler Netze werden in der Arbeit zwei weitere unterschiedliche Lernstrategien vorgestellt:

- das Lernen der optimalen Parameter auf der Basis der Ergebnisse eines Optimierungsalgorithmus, d.h. das neuronale Netz lernt die optimale Steuerung vom Optimierungsalgorithmus und
- das Lernen der Bewässerungssteuerung mit Bewertung durch eine Zielfunktion (Abb.3), d.h. das neuronale Netz ersetzt dann den Optimierungsalgorithmus.

3.3.1 Lernverfahren mit nichtlinearem Optimierungsalgorithmus

Es macht durchaus Sinn, die Steuerungsaufgabe zuerst auf dem klassischen Wege zu lösen, um danach ein neuronales Netz mit der optimalen Lösung für einen Bewässerungsvorgang anzulernen. Dadurch erhält man erstens ein Werkzeug mit einem zuverlässigen Reaktionsverhalten (Zeit und Parameter), und zweitens ist das ANN im Gegensatz zu der Kombination von numerischem Bewässerungsmodell und numerischem Optimierungsalgorithmus sehr einfach anwendbar.

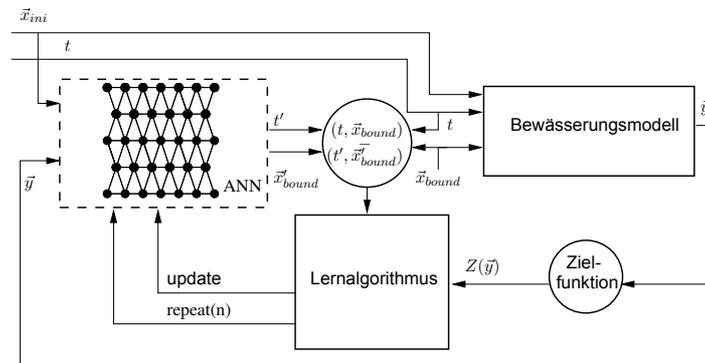


Abbildung 3: Lernverfahren mit Bewertung der Trainingselemente.

3.3.2 Lernverfahren mit Bewertung der Trainingselemente

Für das alternative Verfahren wird ausgenutzt, daß der Lernvorgang von neuronalen Netzen ein stochastischer Vorgang ist, der u.a. den statistischen Eigenschaften der Eingangssignale folgt. Deswegen wird die definierte Zielfunktion Z zu jedem simulierten Szenario berechnet und deren Wert in eine Häufigkeit n_i des entsprechenden Datensatzes in der Trainingsmenge übersetzt (Abb.3). Dadurch werden die Szenarios mit großen Zielfunktionswerten bevorzugt gelernt. Für die Berechnung der Häufigkeit wird eine geeignete Funktion η verwendet, für die gilt:

$$n_i = \eta(\vec{y}_i) : n_i = \eta(Z(\vec{y}_i)) \text{ mit } Z(\vec{y}_i) \in [0, 1] \subseteq \mathbb{R}, n_i \in \mathbb{N} \quad (5)$$

mit n_i als Häufigkeit des i -ten Datensatzes in der neuen Lerndatenmenge. Abbildung 3 zeigt das Vorgehen im Detail wiederum für die online-Variante. Zunächst berechnet das Bewässerungsmodell für ein Bewässerungsszenario $(t, \vec{x}_{ini}, \vec{x}_{bound})$ den Endzustand \vec{y} , und zusätzlich wird für dieses Ergebnis die Zielfunktion $Z(\vec{y})$ ausgewertet. Der Lernalgorithmus entscheidet dann aufgrund der Transformationsfunktion (Gl.5), mit welcher Häufigkeit das Trainingsmuster $(t, \vec{x}_{ini}, \vec{x}_{bound}, \vec{y})$ von dem neuronalen Netz gelernt wird.

4 SOM-MIO – eine Netzarchitektur mit variabler Ein-/Ausgabefunktion

Nichtüberwachte Lernstrategien und die dazu passenden Architekturen neuronaler Netze nutzt man üblicherweise zum Lösen von Klassifikationsaufgaben (wie z.B. der Schrift- oder Spracherkennung). Verwendete Architekturen, wie z.B. die selbstorganisierenden Merkmalskarten (SOM), besitzen daher keine Ausgabeschicht und lernen auf der Basis der vollständigen Vektoren $\vec{x}_{SOM} = (\vec{x}, \vec{y})$ aus der Trainingsdatenmenge.

Dennoch gibt es verschieden Wege, eine Ausgabeschicht in SOM zu integrieren. Die in

der Literatur beschriebenen Möglichkeiten erweitern die Merkmalsvektoren \vec{m}_i entweder um einen skalaren Wert [Koh01] oder lokale lineare Abbildungen [RMS92], die durch zusätzliche Ausgabevektoren und -felder realisiert werden. Bei den lokalen lineare Abbildungen ändert sich der Charakter der Lernstrategie mit den Erweiterungen der Netzstruktur und dem dazu erforderlichen Lernalgorithmus schlagartig zu einem überwachten Verfahren. Um jedoch das Potential der nichtüberwachten Lernstrategie für die Bewässerungssteuerung auszuschöpfen, soll im folgenden ein anderer und neuer Weg gezeigt werden.

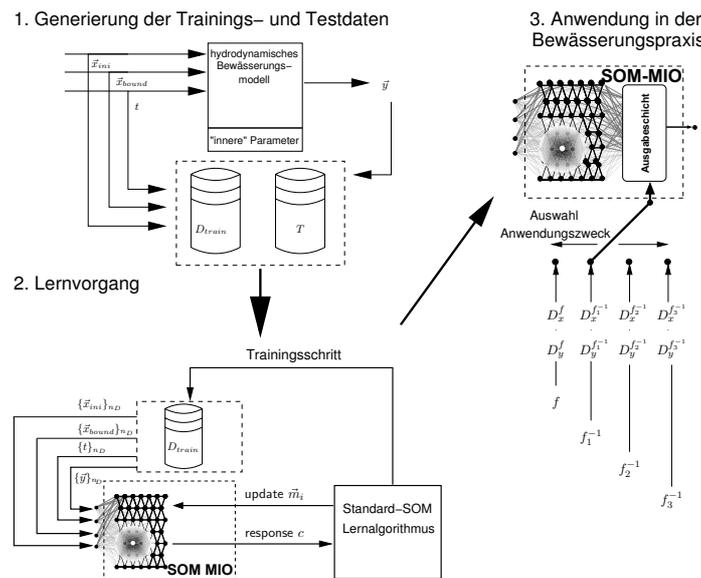


Abbildung 4: Nichtüberwachte Lernstrategien mit SOM

Die Verwendung einer nichtüberwachten Lernstrategie bringt den Vorteil, daß eine einmalig trainierte Merkmalskarte in der Anwendung für mehrere Zwecke benutzbar ist. Für den Lernvorgang kann dabei auf den Standard-SOM Lernalgorithmus zurückgegriffen werden. Zur Erweiterung ihrer Einsatzmöglichkeiten wird der trainierten Merkmalskarte eine Ausgabeschicht mit variabler Ausgabefunktion (SOM-MIO) hinzugefügt (Abb. 4). In der Ausgabefunktion agieren die Neuronen der Merkmalskarte als Stützstellen für ein neues mehrdimensionales Interpolationsverfahren. Dazu müssen für eine spezielle Aufgabe die passenden Dimensionen der Merkmalsvektoren selektiert werden.

Wie Abbildung 4 zeigt, werden zu diesem Zweck zwei korrespondierende Matrizen D_x ⁶ und D_y eingeführt, die die Eingabe- und Ausgabeelemente einer betrachteten Anwendungsaufgabe spezifizieren:

$${}^6 D_x \text{ ist z.B. im 3D-Fall: } D_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, D_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ oder } : D_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

$$\mathcal{D}_x = \text{diag} \{d_i\}, d_i = \begin{cases} 1 & i \in \text{Indizes der Eingabelemente} \\ 0 & i \in \text{Indizes der Ausgabeelemente} \end{cases} \quad (6)$$

Zur Berechnung der Ausgabewerte werden zunächst in Anlehnung an den Lernalgorithmus die n_{bmu} nächsten Merkmalsvektoren \vec{m}_c zu dem maskierten Eingangsvektor $\vec{x}_{SOM} \mathcal{D}_x$ als Stützstellen ermittelt:

$$\|\vec{x}_{SOM} - \vec{m}_c\|_{n_{bmu}} = \min_{n_{bmu}} \{\|\vec{x}_{SOM} \mathcal{D}_x - \vec{m}_i \mathcal{D}_x\|\}_i. \quad (7)$$

Danach wird in einem neuen Interpolationsverfahren eine Triangulation χ verwendet, die auf der Basis n nächsten Neuronen der SOM ein neues Berechnungsnetz erzeugt, welches im zweidimensionalen Fall aus Dreiecken besteht. Die Erzeugung des Berechnungsnetzes $\{\vec{t}_j\}_{n_\chi}$ mit n_χ Elementen erfolgt mit der *Delaunay-Triangulation* χ , die den Quickhull-Algorithmus [BDH96] benutzt:

$$\{\vec{t}_j\}_{n_\chi} = \chi(\{\vec{m}_i \mathcal{D}_x\}_{n_{bmu}}). \quad (8)$$

Die Triangulation χ findet auf der Basis der Komponenten der Merkmalsvektoren \vec{m}_i der SOM-MIO statt, die durch die gewählte Ein-/Ausgabefunktion $f(D_x, D_y)$ mit der Eingabemaske D_x und der Ausgabemaske D_y spezifiziert wurden. Nach der Generierung des triangulären Berechnungsnetzes kann für jede Eingabe \vec{x} ein Element \vec{t}_j^* aus dem generierten Berechnungsnetz $\{\vec{t}_j\}_{n_\chi}$ gefunden werden, welches \vec{x} einschließt oder für den Fall, daß der Eingabevektor nicht in der konvexen Hülle liegt, die alle Merkmalsvektoren der SOM-MIO einschließt, wird das Dreieck des Berechnungsnetzes mit dem nächstliegenden Masseschwerpunkt ermittelt. Auf der Basis der Merkmalsvektoren der Neuronen an den Ecken des einschließenden Elements und der zu D_x korrespondierenden Ausgabemaske D_y erfolgt abschließend die Berechnung der Ausgabe y' mittels baryzentrischer Interpolation. Für Eingangswerte außerhalb der konvexen Hülle der Merkmalsvektoren wird die baryzentrische Interpolation unter der Annahme eines konstanten Gradienten durchgeführt.

5 Die Anwendung der SOM-MIO in der Bewässerungslandwirtschaft

In der zu diesem Artikel gehörenden Dissertation [Sch05] werden SOM-MIO zusammen mit ein-, zwei- und dreidimensionalen Strömungsmodellen innerhalb der zweiphasigen Optimierungsstrategie verwendet. Bei den dort präsentierten Beispielen wird sichtbar, daß das neuentwickelte Prinzip zur optimalen Steuerung der existierenden Bewässerungsverfahren (Beregnung, Oberflächenbewässerung und Mikrobewässerung) allgemein nutzbar ist.

Die Dissertation zeigt umfangreiche Anwendungen der SOM-MIO Architektur zur deterministischen und stochastischen Steuerung der Tropfbewässerung sowie zur Steuerung der Furchenbewässerung bei bedarfsgerechter Bewässerungsplanung und bei Defizitbewässerung. Diese Beispiele demonstrieren neben der geforderten Genauigkeit, Schnelligkeit und Stabilität das Potential für eine signifikante Steigerung des Bewässerungswirkungsgrads gegenüber den herkömmlichen Verfahren zur Steuerung der Wassergabe.

Außerdem wird am Beispiel der Bewässerungssteuerung gezeigt, wie die neue SOM-MIO Architektur zu einer wesentlich schnelleren Lösung von komplexen Aufgaben wie Monte-Carlo-Simulationen oder die Lösung von geschachtelten Optimierungsproblemen beitragen kann. Die Lösungen, die für Bewässerungsverfahren gefunden wurden, lassen sich durchaus auf andere Probleme in der Hydrologie übertragen. Beispielsweise wird der Einsatz von neuronalen Netzen zur Lösung von aktuellen Problemen wie die Echtzeit-Hochwasservorhersage durch die Verfügbarkeit von Trainingsdaten eingeschränkt. Hier werden die entwickelten Lernverfahren dazu benutzt, die Trainingsdatenmenge um fehlende Szenarios zu ergänzen, die von physikalisch-begründeten Niederschlags-Abfluß- und hydraulischen Gerinnemodellen generiert werden [SCG⁺05].

Literatur

- [Ach80] W. Achtnich. *Bewässerungslandbau*. Ulmer, 1980.
- [BDH96] C. B. Barber, D.P. Dobkin und H.T. Huhdanpaa. The Quickhull Algorithm for Convex Hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996.
- [DK86] J. Doorenbos und A. H. Kassam. Yield response to water. FAO Irrigation and Drainage Paper No. 33, Rome (Italy), 1986.
- [Koh01] Teuvo Kohonen. *Self-Organizing Maps*. Springer, Berlin, 2001.
- [McC00] D. McClymont. FIDO v2: Furrow Irrigation Design Optimiser. Manual, 2000.
- [Nov81] V. Novak. The structure of Evapotranspiration (in Slovak) I. and II. *Vodohosp.*, 29:476–492 and 581–582, 1981.
- [RMS92] H. Ritter, T. Martinetz und K. Schulten. *Neuronale Netze*. Addison Wesley, Bonn, 1992.
- [RW97] N.S. Raghuvanshi und W.W. Wallender. Economic Optimization of furrow irrigation. *Journal of Irrigation and Drainage Engineering*, 123(5):377–384, 1997.
- [SCG⁺05] G.H. Schmitz, J. Cullmann, W. Goerner, F. Lennartz und W. Droege. PAI-OFF: Eine neue Strategie zur Hochwasservorhersage in schnell reagierenden Einzugsgebieten. *Hydrologie und Wasserbewirtschaftung – Hydrology and Water Resources Management*, 49(5):226–234, 2005.
- [Sch05] Niels Schütze. *Neue Methoden zur Steuerung der Wassergabe mit neuronalen Netzen in der Bewässerungslandwirtschaft*. Dissertation, Technische Universität Dresden, 2005.
- [SS92] G. H. Schmitz und G. J. Seus. Mathematical Zero-Inertia Modeling of Surface Irrigation: Advance in Furrows. *Journal of Irrigation and Drainage Engineering*, 118(1), Januar 1992.
- [vvvG96] J. Šimůnek, M. Šejna und M.Th. van Genuchten. HYDRUS-2D. Manual, International Ground Water Modeling Center, Colorado School of Mines, 1996.
- [Wö5] Th. Wöhling. *Physically based modeling of furrow irrigation systems during a growing season*. Dissertation, Dresden University of Technology, 2005.
- [ZSFP01] D. Zerihun, C.A. Sanchez und K.L. Farewell-Poe. Analysis and Design of Furrow Irrigation Systems. *Journal of Irrigation and Drainage Engineering*, 127(3):161–169, 2001.



Niels Schütze wurde am 26. April 1968 in Dresden geboren. Nach dem Abitur an der Spezialschule physikalisch-technischer Richtung "Martin Andersen Nexö" studierte er ab 1989 an der Technischen Universität Dresden Informatik mit dem Schwerpunkt Künstliche Intelligenz. Sein Studium schloß er 1995 mit der Diplomarbeit zum Thema "Entwurf und Entwicklung eines Systems zum fallbasierten Schließen für die Prognose von Algenmassenentwicklungen in Standgewässern", die aus einer Zusammenarbeit mit dem Institut für Hydrobiologie der TU Dresden entstand.

Seine Promotion beendete er nach einer interdisziplinären Forschungsarbeit am Institut für Hydrologie und Meteorologie mit summa cum laude. Gegenwärtig weilt Niels Schütze als Postdoc am Technion in Haifa (Israel) und beschäftigt sich mit der stochastischen Optimierung der Bewirtschaftung komplexer Gewässersysteme. Niels ist verheiratet und hat drei Kinder. In seiner Freizeit geht er gern klettern und tanzen.