Neuronale Netze für die Approximation ökologischer Simulationen - Ein Vergleich anhand von SimapD

Ontje Lünsdorf[1], Jens Finke[1]

Zusammenfassung

In diesem Paper werden verschiedene Arten neuronaler Netze und Support Vector Machines auf die Anwendbarkeit für die Prognose von Simulationsergebnissen untersucht, die durch das Simulationsmodell SimapD erzeugt werden. Ziel ist es, bei einer Vielzahl von Szenarien die Bewertung zu beschleunigen, wie es z.B. häufig für die Optimierung benötigt wird. Anstelle der eigentlichen Simulation wird eine analytische Annäherung des Ergebnisses durch ein neuronales Netz angestrebt, wodurch sich die Laufzeit einer Optimierung erheblich reduzieren lässt. Voraussetzung ist, dass ein geeignetes Approximationsverfahren gewählt wird, das die Simulationsergebnisse möglichst genau annähert. Um dabei kein neuronales Lernverfahren durch ungünstige Parametrisierung oder Trainingsdaten zu diskriminieren, werden hier algorithmische Vorgehensweisen präsentiert, so dass jedes Verfahren mit möglichst optimalen Parametern für die Problemstellung evaluiert wird.

Die Analyse in diesem Paper zeigt, dass sich der Aufwand für die Evaluation verschiedener neuronaler Netze lohnt, da deren Approximationsgüte für SimapD-Simulationen erheblich variiert. Für dieses Simulationsmodell zeigten die Support Vector Machines die beste Leistung hinsichtlich Trainingsaufwand und Approximationsgüte.

1 Einleitung

Ein wichtiges Anwendungsgebiet ökologischer Modelle ist die Analyse und Bewertung von Auswirkungen verschiedener Managementmaßnahmen auf die komplexen Zusammenhänge in einem Ökosystem, um so die Entscheidung über effektive Maßnahmen zu unterstützen. In diesem Kontext dient das Simulationsframework SimapD dazu, Maßnahmen zur Entschneidung der Landschaft und für die Etablierung eines überregionalen Korridornetzes zu evaluieren, damit die

[1] Carl von Ossietzky Universität Oldenburg, Uhlhornsweg 84, Umweltinformatik, D-26111 Oldenburg; E-Mail: {ontje.luensdorf | jens.finke}@informatik.uni-oldenburg.de

z.B. durch das Straßennetz resultierende Landschaftszerschneidung vermindert wird. Mögliche Maßnahmen sind dabei z.B. der Bau von Über- oder Unterführungen.

Bei der Auswahl geeigneter Maßnahmen handelt es sich häufig um ein Optimierungsproblem, da in der Regel die verfügbaren Ressourcen (Zeit, Finanzmittel, etc.) begrenzt sind. Aus diesem Grund sind Entscheidungsträger an den Maßnahmen interessiert, die sich unter begrenzten Ressourcen realisieren lassen und den größten positiven Effekt erzielen. Um ein solches Optimierungsproblem effizient lösen zu können, sind bereits für kleine Probleme Heuristiken nötig, die den Lösungsraum geschickt absuchen und so schnell eine möglichst gute Lösung finden. Jede Lösung stellt dabei einen bestimmten Einsatz von Managementmaßnahmen dar, die durch eine Simulation zu bewerten ist. Je nach Szenario sind dabei eine Vielzahl von Iterationen durchzuführen, deren Simulation insgesamt zu einer langen Laufzeit der Optimierung führt.

In diesem Paper wird die Anwendbarkeit von neuronalen Netzen bei der approximativen Bewertung einer Lösung, d.h. eines Maßnahmenkatalogs, untersucht, um bei der Optimierung die zeitaufwändigen SimapD-Simulationen zu umgehen. Statt der Simulation werden die neuronalen Netze während der Optimierung für die Bewertung einer Lösung eingesetzt. Dabei muss allerdings sichergestellt sein, dass die Approximationsverfahren das Simulationsergebnis auch hinreichend genau annähern können.

Der Aufbau des Papers ist wie folgt: Zunächst wird in Abschnitt 2 das Simulationsframework SimapD kurz dargestellt und das eigentliche Optimierungsproblem erläutert. Anschließend werden in Abschnitt 3 die für die Approximation untersuchten Typen von neuronalen Netzen vorgestellt. Um ein geeignetes Verfahren für die Approximation von SimapD-Simulationen zu ermitteln, wurden die Ergebnisse der verschiedene Netze mit unterschiedlichen Vergleichsmetriken analysiert, die in Abschnitt 4 beschrieben werden. Die erzielten Ergebnisse und deren Diskussion sind in Abschnitt 5 bzw. Abschnitt 6 dargestellt. Abschließend zieht Abschnitt 7 ein Fazit der hier durchgeführten Untersuchung.

2 Simulationsmodell SimapD

Die Approximationsgüte neuronaler Netze wurde anhand des Simulationsframeworks

SimapD (Simulation of antropogenic disturbances) untersucht. Das Framework erlaubt die Abschätzung der Wirkung künstlicher Zerschneidungselemente (Straßen, Bahnstrecken, etc.) auf die Konnektivität bzw. Durchlässigkeit von Korridornetzen für überregionale Wanderungen wild lebender Tiere (Finke 2007). Die anthropogenen Störungen hindern die Tiere daran, entfernte Habitate zu erreichen, was zu einer Vielzahl negativer ökologischer Effekte führt (Baier et. al. 2006). In SimapD wird ein Korridornetz durch einen Graphen repräsentiert, dessen Knoten die einzelnen Habitate beschreiben, die mittels der Korridore (die Kanten des Graphen) miteinander verbunden sind. Dieses Korridornetz beschreibt auf sehr abstrakte Weise die Wanderrouten der Tiere (vgl. Abbildung 2).

Jeder Korridor kann durch eine anthropogene Störung beeinträchtigt sein, wodurch dessen Mortalitäts- und Widerstandswirkung beeinflusst wird. Je höher die Widerstandswirkung ist, desto seltener wird der Korridor von einem Individuum für die Durchquerung genutzt. Je höher die Mortalitätswirkung der Störung ist, desto häufiger wird ein Individuum bei der Durchquerung des Korridors getötet. Diese beiden Attribute bestimmen, wie gut ein einzelner Korridor zu durchqueren ist. Mit Hilfe einer individuen-orientierten Simulation wird in SimapD die Güte des gesamten Korridornetzes unter Berücksichtigung der einzelnen Widerstands- und Mortalitätswerte der Korridore ermittelt. Das Resultat der Berechnung ist der Durchlässigkeitsindex λ (Finke 2007), der für die Bewertung der Lösungen während der Optimierung herangezogen wird. Je höher dieser Wert ist, desto durchlässiger und weniger zerschnitten ist das betrachtete Korridornetz.

2.1 Optimierungsproblem

Mit Hilfe von SimapD können verschiedene Maßnahmen, die auf die Verbesserung der Widerstands- und Mortalitätswirkung einzelner Korridore abzielen, simuliert und ihre Effektivität mit anderen Maßnahmen anhand des Durchlässigkeitsindex verglichen werden. Die dabei zu berücksichtigenden begrenzten Ressourcen können für die Kombination verschiedener Maßnahmen eingesetzt werden, wodurch sich die Frage ergibt, wie die Ressourcen am Besten auf das Korridornetz zu verteilen sind.

Für die Optimierung wird dieses allgemeine Problem auf ein Kombinationsproblem reduziert, das sich gut mit existierenden Heuristiken, wie z.B. der Tabusuche,

bewältigen lässt. Dazu werden virtuelle Kosten eingeführt, die die verfügbaren Ressourcen repräsentieren und in diskreten Einheiten den einzelnen Korridorattributen zugewiesen werden können. Je nach Zuweisung verbessert eine Kosteneinheit entweder die Mortalitäts- oder die Widerstandswirkung des Korridors. Die Effektivität der Verbesserung wird über eine Kostenfunktion abgebildet, die zu einer Menge von Kosten die erwartete Veränderung des jeweiligen Korridorattributs ermittelt (Finke und Sonnenschein 2007b). Um die Güte der Kostenverteilung zu bewerten, werden zunächst für alle Korridore die veränderten Mortalitäts- und Widerstandswerte berechnet und anschließend mit der individuen-orientierten Simulation der resultierende Durchlässigkeitsindex λ für das gesamte Korridornetz bestimmt.

2.2 Optimierung von Korridornetzen

Um die optimale Verteilung der Kosten auf die Korridore zu finden, d.h. eine optimale Lösung des Optimierungsproblems, wird als Heuristik die Tabusuche verwendet, die eine möglichst gute Lösung effizient anzunähren vermag (Gendreau 2003). Ausgehend von einer initialen Lösung werden schrittweise leicht veränderte Nachbarlösungen generiert und evaluiert. Die beste Nachbarlösung wird als Lösung für den nächsten Iterationsschritt verwendet. Im Gegensatz zu einfachen Hill-Climbing Algorithmen können die Zwischenlösungen bei der Tabusuche durchaus schlechter sein als die bisher beste gefundene Lösung. Auf diese Weise kann die Suche lokale Maxima im Lösungsraum überwinden, in der Hoffnung das globale Maximum zu finden.

Jede Nachbarlösung stellt eine andere Kostenverteilung auf das Korridornetz dar, die durch die Simulation bewertet werden muss, um die Lösung mit dem besten λ-Wert zu finden. Je nach Nachbarschaft, müssen dazu eine Vielzahl von Simulationen durchgeführt werden. Um die Laufzeit der Optimierung zu reduzieren, soll keine SimapD-Simulation zur Ermittlung des λ-Werts angestoßen werden, sondern das Ergebnis mit Hilfe eines Approximationsverfahrens möglichst genau angenähert werden.

Das Approximationsverfahren ist dazu zunächst mit einer (im Vergleich zur Optimierung) kleinen Menge an Kostenverteilungen und den aus der Simulation

resultierenden λ-Werten zu trainieren. Danach kann das Approximationsverfahren das Ergebnis der Simulation für beliebige Kostenverteilungen annähern. Im restlichen Paper wird für verschiedene Typen neuronaler Netze untersucht, wie gut und wie effizient sie bei einer gegebenen Kostenverteilung die λ-Werte der SimapD-Simulation approximieren können.

3 Approximationsverfahren

Neuronale Netze werden in der Praxis oft als universelle Approximatoren eingesetzt, weshalb für diesen Anwendungsfall eine aktuelle Auswahl an Lernverfahren untersucht wird. Alle hier betrachteten neuronalen Netze sind aus Schichten von Neuronen aufgebaut, durch die eine Eingabe schrittweise propagiert wird. Solche Netze werden auch als Mehrschicht-Perzeptren bezeichnet. Typischerweise verfügen sie über eine Eingabe-, eine versteckte und eine Ausgabeschicht. Die Neuronen in diesen Schichten besitzen verschiedene Funktionen und Parameter mit denen aus der Eingabe der vorhergehenden eine Ausgabe für die nachfolgende Schicht errechnet wird. Diese Funktionen unterscheiden sich je nach Typ des neuronalen Netzes.

Die Parameterwerte der Neuronen müssen auf jede Problemstellung angepasst werden. Dieser Prozess wird als Lernen oder Training bezeichnet. In dieser Arbeit wurden die folgenden vier Lernverfahren untersucht.

3.1 Errorbackpropagation

Mit dem Errorbackpropagation-Verfahren können klassische Mehrschicht-Perzeptren trainiert werden (Haykin 1999). Die Eingabeverarbeitung eines Neurons basiert hier auf einer sigmoiden Funktion. Aufgrund der Differenzierbarkeit dieser Funktion kann aus dem Ausgabefehler für jedes Neuron und jeden Parameter ein Anpassungsfaktor ermittelt werden. Wiederholt man diesen Schritt, nähert sich der Ausgabefehler einem Minimum an.

3.2 Smooth-Function-Approximation

Dieses Lernverfahren findet ebenfalls bei klassischen Mehrschicht-Perzeptren

Anwendung (Ferrai und Stengel 2005). Im Gegensatz zum Errorbackpropagation-Verfahren liegt hier kein iterativer Prozess zu Grunde, sondern eine Formel. Dieses Verfahren erlaubt also die analytische Bestimmung der Parameter des Mehrschicht-Perzeptrons. Zwei Varianten dieses Lernverfahrens wurden untersucht. In der ersten Variante enthält die versteckte Schicht des Mehrschicht-Perzeptrons genau so viele Neuronen wie es Trainingsdaten gibt. In der zweiten Variante ist die Neuronenanzahl variabel.

3.3 Radiale-Basis-Funktionsnetze

Radiale-Basis-Funktionsnetze sind ebenfalls als Mehrschicht-Perzeptron darstellbar, allerdings werden hier radiale Basisfunktionen und Distanzfunktionen statt der sonst üblichen sigmoiden Funktionen verwendet (Haykin 1999). Zum Training wird eine angepasste Form des Errorbackpropagationverfahrens verwendet. Zusätzlich wurde ein einfacheres, analytisches Lernverfahren für Radiale-Basis-Funktionsnetze untersucht (Borgelt et al. 2003).

3.4 Support-Vector-Machines

Auch Support-Vector-Machines können als Mehrschicht-Perzeptron dargestellt werden (Haykin 1999). Dabei werden die Eingabedaten so in einen höherdimensionalen Raum transformiert, dass sie sich durch eine Hyperebene linear separieren lassen. Diese Hyperebene wird durch die Support Vectors beschrieben. Diese Vektoren können auch als Stützstellen eines Polynoms angesehen werden, dass die zu beschreibende Funktion approximiert. Im Gegensatz zu den anderen Verfahren wurden die Support-Vector-Machines nicht selbst implementiert, sondern die Bibliothek libsvm verwendet (Chang und Lin 2007).

4 Vergleich der Approximationsgüte

Wie bereits erwähnt, gilt es, die Güte verschiedener Approximationsverfahren bei Anwendung auf die SimapD-Simulationen zu ermitteln und zu vergleichen. Das Problem, also die Simulation, wird dabei als Blackbox angesehen, über deren Funktionsweise keine Informationen verfügbar sind. Stünden Informationen über die

Funktionsweise zur Verfügung, müssten die Approximationsverfahren spezifisch angepasst werden, was neben dem erheblichen Aufwand auch nicht bei jedem Verfahren im selben Umfang möglich wäre. So erfordert z.B. das Error-Backpropagation-Verfahren differenzierbare Aktivierungs- und Ausgabefunktionen der Neuronen und die Kernel-Funktionen der Support-Vector-Machines müssen die Bedingungen des so genannten Mercer-Theorems erfüllen. Das Berücksichtigen von Wissen über die Simulationsfunktionsweise wäre also jeweils nur im Rahmen dieser Bedingungen möglich.

Um die Vergleichbarkeit der verschiedenen Typen von neuronalen Netzen zu gewährleisten, müssen diese geeignet parametrisiert und mit entsprechenden Daten trainiert werden. Das Problem ist, dass die Verfahren für gute Ergebnisse ganz unterschiedliche Parameter und je nach Lernmethode unterschiedliche Trainingsdaten benötigen. Dazu wurden die im Folgenden beschriebenen nicht-diskriminierenden Methoden angewandt, um die Approximationsverfahren geeignet zu trainieren und deren Vergleichbarkeit zu gewährleisten.

4.1 Parametrisierung

Die Approximationsverfahren verfügen im Allgemeinen über Parameter, mit denen problemspezifische Anpassungen vorgenommen werden können. Analog zu der Anpassung der Approximationsverfahren an das gegebene Problem können ohne a-priori-Informationen keine Schlüsse auf sinnvolle Parametrisierungen der Verfahren gezogen werden. Um alle Verfahren gleich zu behandeln, wurde eine algorithmische Parametrisierung durchgeführt, die sich an einer Trainingsfehlerschwelle t orientiert. Alle Verfahren wurden so parametrisiert, dass sie mindestens diese Schwelle erreichen. Auf diese Weise wird eine Festlegung der Parameterwerte nach den gleichen Gesetzmäßigkeiten garantiert wodurch die Vergleichbarkeit gegeben ist.

Die Parameterbestimmung wird mit gegebenen Trainings- und Validierungsdatenmengen (T bzw. V) durchgeführt. Zunächst wird jedes Approximationsverfahren mit der Trainingsdatenmenge T und initialen Parametern trainiert. Anschließend kann der mittlere quadratische Fehler des Verfahrens bei der Approximation dieser beiden Mengen berechnet werden.

Aus dem Trainingsfehler e_t und dem Validierungsfehler e_v wird der so genannte

Parametrisierungsfehler e_p berechnet. Dessen Berechnung ist abhängig von der Trainingsfehlerschwelle *t*. Es gilt: $e_p = 1 + e_t - t$, wenn $e_t \geq t$ und $e_p = 1 - e^{-e_v}$ sonst.

Wird die Trainingsfehlerschwelle nicht erreicht, ist der Parametrisierungsfehler immer größer als 1, d.h. das Lernverfahren kann die Trainingsdaten nicht genügend genau annähern. Wird die Schwelle erreicht, ist der Parametrisierungsfehler abhängig von dem Validierungsfehler, der über eine sigmoide Funktion auf das Wertebereichsintervall [0,1[abgebildet wird. Je kleiner der Parametrisierungsfehler ist, desto besser nähert daher das neuronale Netz die zu lernenden Daten mit den gewählten Parametern an. Die Berücksichtigung des Validierungsfehlers e_v sorgt für Parametersätze, mit denen auch die Validierungsdaten und nicht ausschließlich die Trainingsdaten gut angenähert werden.

Aus der Menge der möglichen Parameter ist für jedes Approximationsverfahren der Parametersatz zu bestimmen, mit dem der Parametrisierungsfehler minimal wird. Die Parametrisierungsfehlerfunktion beruht sowohl auf dem Approximationsverfahren als auch auf der Simulation und ist somit nicht-linear. Diese Bedingung schränkt die möglichen Ansätze zur Lösung dieses Parameter-Optimierungsproblems ein. Die Ansätze können weiterhin nicht auf die Ableitungen der Parametrisierungsfehlerfunktion beruhen, da diese für SimapD-Simulationen nicht gebildet werden kann. Darüber hinaus verwenden viele Approximationsverfahren ganzzahlige Parameter, wie z.B. die Anzahl der Neuronen in der versteckten Schicht, und eingeschränkte Wertebereiche.

Ein Verfahren mit dem sich alle diese Randbedingungen bei der Optimierung der Parameter integrieren lassen, ist das Downhill-Simplex-Verfahren (Nelder und Mead 1965). Dieses Verfahren beruht auf dem Simplex, der kleinsten Punktmenge, aus der in einem N-dimensionalem Raum ein Volumen gebildet werden kann. Dieses Verfahren ist deshalb auch in großen Parameterräumen anwendbar. Außerdem ist es leicht zu implementieren, robust und unterstützt mit leichten Anpassungen auch ganzzahlige Parameter.

4.2 Trainingsdatenauswahl

Das Problem der Trainingsdatenauswahl ist analog zu dem Problem der Parametrisierung. Bei jedem Approximationsverfahren kann eine andere

Trainingsdatenverteilung zum optimalen Ergebnis führen. Auch hier wird wieder ein algorithmisches Verfahren gewählt, um für jedes Approximationsverfahren spezifische Trainingsdaten nach den gleichen Gesetzmäßigkeiten zu erzeugen.

Der Algorithmus geht von der Annahme aus, dass der erhöhte Fehler eines Approximationsverfahrens in einem lokalen Bereich des Eingaberaums durch zusätzliche Trainingsdaten in diesem Bereich minimiert werden kann.

Initial ist eine Menge fester Trainings- und Validierungsdaten vorgegeben. Mit den Trainingsdaten wird das Verfahren initialisiert und trainiert. Anschließend wird der Validierungsfehler für alle Validierungsdaten berechnet. Das Datum, das den größten Fehler hat - also am schlechtesten approximiert wurde - wird mit in die Trainingsdatenmenge aufgenommen. Die Validierungsmenge wird mit zusätzlichen Validierungsdaten um dieses neue Trainingsdatum herum ergänzt. Auf diese Weise werden aus problematischen Bereichen des Eingaberaums mehr Trainingsdaten ausgewählt. Dieser Prozess wird in einer für den gesamten Vergleich festgelegten Anzahl an Iterationen wiederholt.

Für jedes Approximationsverfahren werden die Trainingsdaten auf diese Weise separat ausgewählt, so dass jedes Netz mit den für den Lernalgorithmus optimalen Parametern trainiert wird.

4.3 Bewertung

Grundlage der Bewertung der Approximationsgüte der untersuchten Verfahren ist die Evaluierungsdatenmenge *A.* Diese Datenmenge ist, im Gegensatz zu den Trainings- und Validierungsdaten, für alle Verfahren gleich, so dass sich eine vergleichbare quantitative Aussage über die Approximationsgüte treffen lässt. Die Evaluierungsdaten sind so gewählt, dass sie den Eingaberaum möglichst gleichmäßig abdecken.

Die folgenden Metriken bilden die Bewertungsgrundlage für die Güte der Approximation der SimapD-Simulationsergebnisse bei einer gegebenen Kostenverteilung:

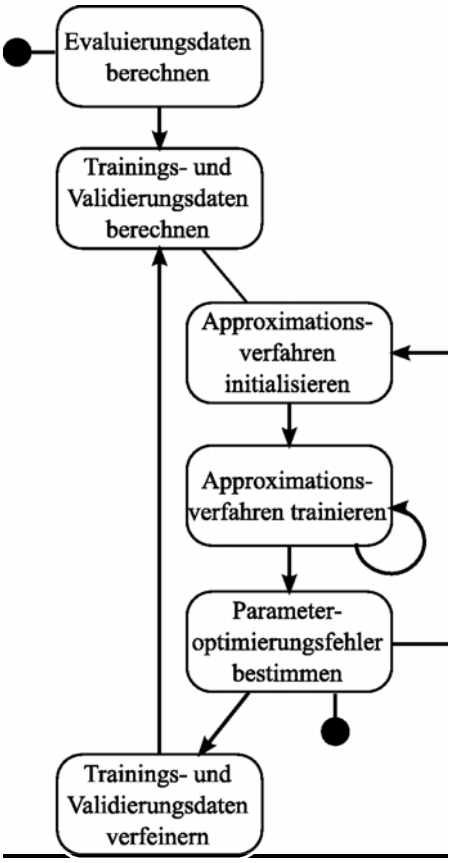
- Die *Genauigkeit* eines Approximationsverfahrens wird durch den mittleren quadratischen Fehler (MSE) der vorhergesagten Werte gegenüber den

Evaluierungsdaten ermittelt.

- Unter der *Generalisierung* wird die Fähigkeit verstanden, wie gut die Verfahren in Bereichen approximieren, die weit von Trainingsdaten entfernt sind. Für jedes Evaluierungsdatum v wird dazu die minimale euklidische Distanz $d_{\min}$ zu den verwendeten Trainingsdaten berechnet. Die normierten Fehler der Evaluierungsdaten werden anschließend mit den normierten Minimaldistanzen gewichtet und aufsummiert: $G = \frac{1}{|A|}\sum_{v \in A} |d_{\min}(v)| \cdot |e(v)|$.

  Dieses Generalisierungsmaß G wird also durch die Fehler der am weitesten von den Trainingsdaten entfernten liegenden Evaluierungsdaten bestimmt.

Neben diesen beiden wichtigsten Bewertungskriterien wurden die Verfahren darüber hinaus noch anhand der Trainingsdatenverteilung, des Trainingsaufwands und des Approximationsaufwands evaluiert (Lünsdorf 2007).



4.4 Ablauf des Verfahrenvergleichs

Die bisher vorgestellten Verfahren und Ansätze zum Vergleich verschiedener Approximationsverfahren sind in der Abbildung rechts als Ablaufdiagramm zusammengefasst. Initial wird die Menge der gleich verteilten Evaluierungsdaten erzeugt, anhand derer die Approximationsgüte der Verfahren berechnet wird. Anschließend wird eine vorher festgelegte Anzahl von Vergleichsschritten durchgeführt. In jedem Schritt wird die Parameteroptimierung mit dem Downhill-Simplex-Verfahren in einer ebenfalls vorher festgelegten Anzahl an Vergleichsschritten durchgeführt und danach die Trainings- und Validierungsdatenmenge entsprechend des Algorithmus aus 4.2 verfeinert.

5 Resultate

Das Vergleichsverfahren wurde auf mehrere SimapD-Szenarien angewendet, die

sich in der Größe und dem Vernetzungsgrad des Graphen sowie der Ausprägung der Mortalitäts- und Widerstandswirkung der Korridore unterscheiden.

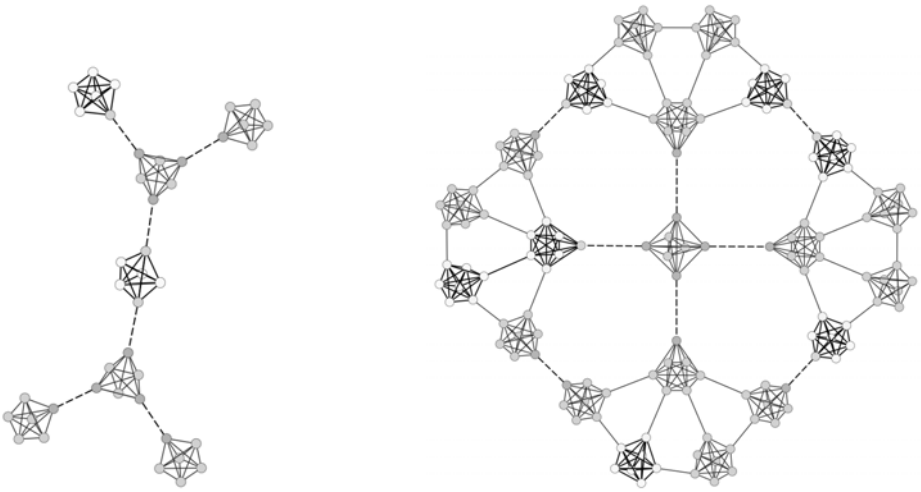


Abbildung 2: Zu simulierende Korridornetze: Baum (links) und vernetzter Baum (rechts)

Exemplarisch werden im Folgenden zwei typische Ergebnisse bezüglich der Approximationsgüte der betrachteten Verfahren dargestellt. Weitere Beispiele finden sich in (Lünsdorf 2007). Abbildung 2 zeigt die Graphen der zu optimierenden Korridornetze. Die Strichstärke mit der ein Korridor im Netzwerk dargestellt ist, zeigt wie durchlässig dieser ist. Je dicker dieser ist, desto durchlässiger ist er. Die gestrichelten Korridore sind die Korridore, an denen potentiell Kosten zur Optimierung des Netzwerks zu investieren sind.

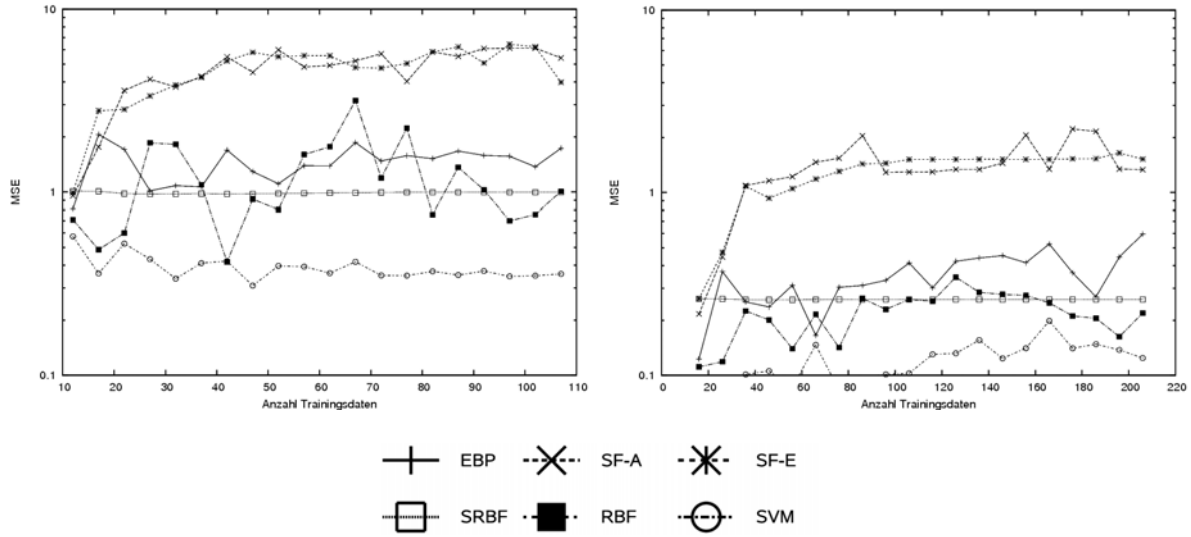


Abbildung 3: Evaluierungsfehler für Baum (links) und vernetzter Baum (rechts)

In Abbildung 3 ist der Evaluierungsfehler der verschiedenen Lernverfahren für die beiden Korridornetze nach jedem Trainingsschritt dargestellt. Auf der horizontalen

Achse ist die Anzahl der verwendeten Trainingsdaten abgetragen. Analog dazu zeigt Abbildung 4 das Generalisierungsmaß für die erlernten neuronalen Netze und Support-Vector-Machines.

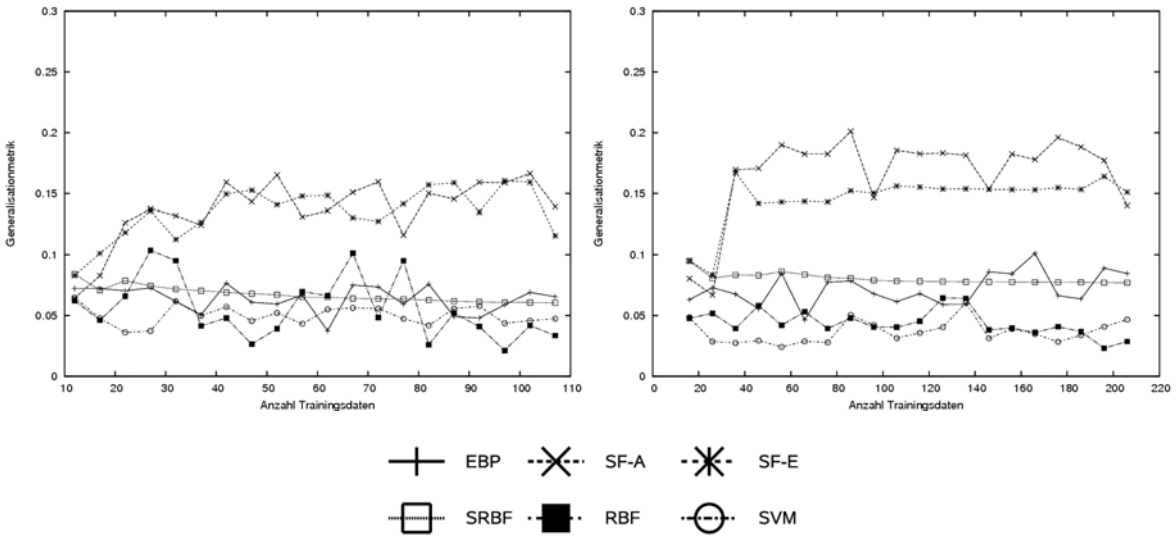


Abbildung 4: Generalisierungsfehler für Baum (links) und vernetzter Baum (rechts)

6 Diskussion

Die Resultate der Szenarienvergleiche zeigen ein eindeutiges Bild: Die Support-Vector-Machines eignen sich am besten zur Approximation von SimapD-Simulationen. Nur in wenigen Fällen erreichen Radiale-Basis-Funktionsnetze dieselbe Approximationsgenauigkeit. Ein Grund hierfür lässt sich in der Regulierung zwischen Generalisation und Approximationsgenauigkeit vermuten. Wegen dieser können die Support-Vector-Machines ungünstige oder problematische Trainingsdaten ignorieren, so dass der Verlauf der entstehenden Approximationsfunktion gut an charakteristische Trainingsdaten angepasst werden kann. Dank dieser Regulation ist das Support-Vector-Lernverfahren vermutlich robuster gegenüber ungünstig verteilten Trainingsdaten als alle anderen Verfahren.

Da bei der Support-Vector-Machine ebenfalls Radiale-Basis-Funktionen als Kernel-Funktionen benutzt werden, ist nicht verwunderlich, dass die Radialen-Basis-Funktionsnetze teilweise dieselbe Genauigkeit und Generalisierung erreichen. Es ist zu vermuten, dass bei mehr Trainingsiterationen die Leistung der Radialen-Basis-Funktionsnetze die der Support-Vector-Machines erreicht. Allerdings ist der Trainingsaufwand bei Radialen-Basis-Funktionsnetzen erheblich höher, als bei den

Support-Vector-Machines, wodurch die Zeitersparnis bei der Optimierung eines Korridornetzes relativiert wird.

Das Error-Backpropagation-Verfahren liefert die zweitbesten Resultate. Zwar erreichen die Simplen-Radialen-Basis-Funktionsnetze in einigen Szenarien eine höhere Genauigkeit, doch ist die Generalisierungsfähigkeit der Error-Backpropagation-Netze durchweg besser.

Probleme treten besonders bei dem Smooth-Function-Approximation Lernverfahren auf. Dieses Verfahren nähert zwar die Trainingsdaten optimal an, doch verfügt das Verfahren über keine Parameter, über die eine bessere Approximation der Validierungsdaten zu erreichen wäre. Bei weitergehenden Versuchen in kontinuierlichen Eingaberäumen zeigte sich, dass dieses Lernverfahren gute Resultate liefern kann, sofern die Trainingsdaten gleich verteilt sind. Allerdings erreichte das Error-Backpropagation-Verfahren auch in diesen Untersuchungen bereits bei weniger Trainingsdaten dieselbe Approximationsgüte. Die Smooth-Function-Approximations-Verfahren können nach diesen Ergebnissen nur in Situationen eingesetzt werden, in denen die Trainingszeit von hoher Relevanz ist und Trainingsdaten mit geringem Aufwand ermittelt werden können.

7 Fazit

Insgesamt ist die Leistung der neuronalen Netze in Bezug auf die Approximationsfähigkeit von SimapD-Simulationen unbefriedigend. Die Radialen-Basis-Funktionsnetze können zwar gute Approximationen liefern, jedoch ist der Rechenaufwand für das Training zu hoch. Wesentlich besser sind hinsichtlich der Approximationsgüte und des Trainingsaufwandes die Support-Vector-Machines für SimapD geeignet. Diese Aussagen lassen sich nicht direkt auf andere Simulationsverfahren übertragen. Allerdings zeigen die Ergebnisse, dass es abhängig von den charakteristischen Eigenschaften der Simulation signifikante Unterschiede zwischen den unterschiedlichen Approximationsverfahren geben kann und eine eingehende Untersuchung verschiedener Verfahren sinnvoll ist. Dazu kann die in diesem Paper skizzierte Vorgehensweise auch für andere Simulationsmodelle angewendet werden.

8 Literatur

Baier, H.; Erdmann, F.; Holz, R.; Waterstraat, A. (Hrsg) (2006): Freiraum und Naturschutz – Die Wirkung von Störungen und Zerschneidungen in der Landschaft. Springer Verlag

Borgelt, C.; Klawonn, F.; Kruse, R.; Nauck, D. (2003): Neuro-Fuzzy-Systeme. Dritte Auflage. Vieweg & Sohn Verlagsgesellschaft.

Chang, C. und Lin, C. (2007): LIBSVM: a library for support vector machines, 30. Novmber 2007. http://www.csie.ntu.edu.tw/~cjlin/libsvm

Finke, J. (2007): Ein Simulationsframework zur Bewertung von großräumigen Habitatnetzen. Modellierung und Simulation von Ökosystemen – Workshop Kölpinsee 2006. Shaker Verlag.

Finke, J. und Sonnenschein, M. (2007b): Simulation and Optimization of Habitat Network Permeability. In: J. Marx Gomez, M. Sonnenschein, M. Müller, H. Welsch, C. Rautenstrauch (Eds): Information Technologies in Environmental Engineering - ITEE 2007, Third International ICSC Symposium, Oldenburg. Springer Verlag, ISBN 978-3-540-71334-0, pp. 433-444

Ferrari, S. und Stengel, R. F. (2005): Smooth Function Approximation Using Neural Networks. IEEE Transactions on Neural Networks, Vol 16.

Gendreau, M. (2003): An Introduction to Tabu Search. In: F. Glover und G. A. Kochenberger (Eds.): Handbook auf Metaheuristics. Kluwer, 37-54.

Haykin, S. (1999): Neuronal Networks, a comprehensive foundation. Zweite Auflage. Prentice-Hall.

Lünsdorf, O. (2007): Evaluation der Approximationsgüte von SimapD-Simulationen durch neuronale Netze. Masterarbeit, Carl von Ossietzky Universität Oldenburg, Abteilung Umweltinformatik.

Nelder, J. A. und Mead R. (1965): A Simplex Method for Function Minimization. Computer Journal, Vol. 7, Seite 308-313.