BAUTEILERKENNUNG UNTER ZUHILFENAHME KÜNSTLICHER NEURONALER NETZE

Almut Bastian, Matthias Schleinkofer Lehrstuhl Bauinformatik, Technische Universität München

Kurzfassung: Die Erstellung und Aktualisierung von modellbasierten Plänen bestehender Gebäude ist eine häufige Anwendung bei der Sanierung von Altbauten und – bei deren Abriss - beim Recycling und Überführung ins Stoffflussmanagment. Eine bequeme und vielversprechende Art, die notwendigen Daten zu sammeln, ist das Laseraufmaßverfahren. Um eine nachfolgende Produktplanung möglich zu machen, müssen die mit dieser Methode erhaltenen Modelle in ein Produktmodell überführt werden. Für diesen Schritt wurde ein Verfahren aus dem Bereich der künstlichen Intelligenz verwendet, indem die Bauteile mit Hilfe eines künstlichen neuronalen Netzes in Bauteilklassen eingeordnet werden.

1 Motivation

1.1 Vom Laseraufmaß zum Stoffflussmanagement für Altbauten

Umbau und Altbausanierung machen einen Großteil des Baugeschehens in den Industrieländern aus. Deshalb ist es für eine langfristige Schonung von Umwelt und Ressourcen besonders wichtig, hierfür leistungsfähige, computergestützte Planungs- und Simulationswerkzeuge bereitzustellen. In einem vom Bayerischen Staatsministerium für Umwelt, Gesundheit und Verbraucherschutz geförderten Forschungsprojekt werden hierzu Grundlagen einer modellbasierten Planung für den Baubestand entwickelt. Kern ist ein so genanntes Produktmodell, in dem neben der dreidimensionalen Geometrie eines Bauwerks alle relevanten Produktdaten (Materialien, physikalische Eigenschaften, ökologische Daten, etc.) gespeichert sind und für Sanierungsplanungen und Simulationen über den gesamten Lebenszyklus des Bauwerks genutzt werden können (s. Abb. 1). Der Schwerpunkt des Projekts liegt auf einer weitgehend automatisierten, zweistufigen Erstellung dieses Produktmodells. Für eine erfolgreiche Planung ist eine exakte

Bestandsaufnahme des aktuellen Bauzustandes unerlässlich. In einem ersten Schritt wird deshalb das geometrische Volumenmodell mit Methoden der lasergestützten Ingenieurvermessung aufgebaut. Von einem Projektpartner wird die erhobene Punktwolke analysiert und zunächst in ein Flächenmodell und anschließend in ein Volumenmodell überführt. Aus diesem wird in einem zweiten Schritt das Produktmodell abgeleitet [1].

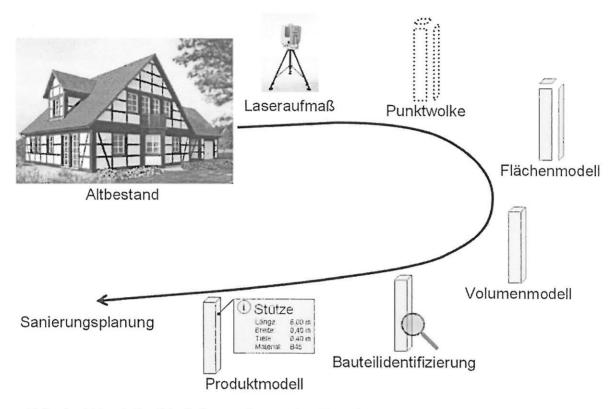


Abb. 1: Ablauf der Modellgenerierung im Forschungsprojekt BayFORREST F248

1.2 Bauteilklassifizierung als Problem der Mustererkennung

Die Überführung des Volumenmodells in ein Produktmodell mit AEC-Objekten (Architecture, Engineering and Construction) soll automatisch erfolgen. Dafür ist es erforderlich, dass der Computer die Objekte des Volumenmodells als Bauteile identifiziert. Während in anderen Bereichen die Mustererkennung schon sehr weit fortgeschritten ist [6], existieren im Bereich Bauwesen bislang wenige Ansätze zur Identifizierung von Bauteilen. Statt der klassischen Lösungen zur Klassifizierung wie Expertensystem [5], Entscheidungsbaum [2] und fuzzy logic [3] wird hier als Herangehensweise die Verwendung künstlicher neuronaler Netze untersucht. Um Bauteile, die in diesem Modell als topologisch eindeutige Volumenobjekte vorliegen, zu erkennen, ist es nahe liegend, deren charakteristische Merkmale zur Entscheidungsfindung heranzuziehen, und nicht, wie in der Bildanalyse in Verbindung mit künstlichen neuronalen Netzen üblich, diese zuerst zu rastern.

2 Neuronale Netze zur Bauteilklassifizierung

2.1 Grundlagen

Künstliche neuronale Netze erhalten an definierten Eingabestellen – den Eingabeneuronen – Signale, die diese an verborgene Schichten weiterleiten und dort bearbeiten. Die Ausgabe wird an speziellen Ausgabeneuronen an den Benutzer übergeben (s. Abb. 4). Wie das Netz die eingegebenen Signale bearbeitet, wird in der Lernphase festgelegt. Vorgegebene Trainingsmuster errechnen hierbei eine Ausgabe, die mit der Sollausgabe verglichen wird. Daraus errechnet sich für das Netz eine Neugewichtung der Signalweitergabe und -auswertung.

In dieser Arbeit wird ein hierarchisches, vorwärtsgerichtetes Netz eingesetzt, der Trainingsalgorithmus bedient sich des Backpropagation-Verfahrens, einem Gradientenabstiegsverfahren [4].

2.2 Bauteilklassifizierung

Es wurde versucht, unter den im Folgenden erläuterten Voraussetzungen eine geeignete Netztopologie, eine geeignete Trainingsbasis und geeignete Eingabeparameter zu finden, mit denen nach entsprechendem Training eine fehlerlose Klassifizierung der Trainingsdaten sowie die weitgehend korrekte Klassifizierung untrainierter Muster möglich ist.

2.2.1 Eingesetztes Netz

Jede der vorgegebenen zu klassifizierenden Bauteilklassen "Wand", "Decke", "Fenster", "Türe", "Stütze", "Träger", "Deckendurchbruch" ist einem Ausgabeneuron zugeordnet, die Entscheidung des Netzes entspricht der Klasse, dessen Neuron die maximale Aktivierung besitzt. Somit ist die Anzahl der Ausgabeneuronen auf sieben festgelegt. Auf diesen Konstanten aufbauend wurden Art und Anzahl der Eingabeparameter sowie die Größe der verborgenen Schicht variiert und dabei verschiedene Trainingsbasen verwendet, bis die optimale Kombination an Eingaben und Anzahl der Eingabe- und verborgenen Neuronen sowie eine brauchbare Trainingsbasis gefunden wurden.

2.2.2 Aufbereitung der Trainingsbasis

Zum Training standen im CAD System Autodesk Architectural Desktop erstellte Produktmodelle von drei verschiedenen Gebäuden zur Verfügung. Um aus diesen Daten eine geeignete Trainingsbasis zu erzeugen, mussten Datensätze, die redundante Eingabeparameter liefern, eliminiert werden. Damit kommt es, je nach Symmetrie des Gebäudes, zu einer mehr oder weniger starken Reduktion und Umgewichtung der Trainingsbasis. Abb. 3 zeigt die Verteilung der sieben in 2.2.1 genannten Bauteilklassen in einem Einfamilienhaus (Abb. 2).

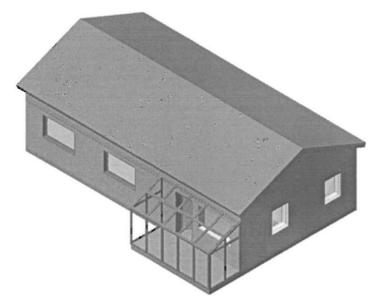


Abb. 2: Musterhaus nach EN-EV 832

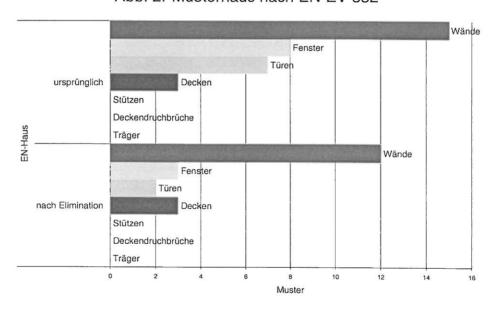


Abb. 3: Anzahl der Muster der einzelnen Bauteilklassen im EN-Haus

2.2.3 Auswahl der als Eingabeparameter verwendeten Geometrieattribute

Es standen folgende Attribute zur Verfügung:

- Größe des in horizontaler Richtung kleinsten umhüllenden Quaders sowie die Ausdehnung in Vertikalrichtung
- Körperoberflächen- und Volumenzahl
- Koordinaten des Schwerpunktes
- Informationen "ist in x anderen Objekten enthalten" und "enthält x andere Objekte"

Variantenstudien mit verschiedenen Kombinationen von nach Art und Anzahl unterschiedlichen Eingabeparametern führten im Wesentlichen zu zwei Ergebnissen:

- Zu viele Eingaben können das Netz "verwirren".
- Die absoluten Werte der Abmessungen scheinen weniger aussagekräftig zu sein, als deren Verhältnisse zueinander.

Die Kombination beider Ergebnisse führte zu dem Schluss, die Abmessungen als Eingabeparameter nicht durch zusätzliche Übergabe ihrer Verhältnisse zueinander zu ergänzen, sondern sie vollständig durch diese zu ersetzen. Die Eingabe wurde in Anbetracht dieser Beobachtungen auf die drei informativsten Parameter beschränkt:

- "x/y", d. h. das Verhältnis der größeren horizontalen Abmessung des kleinsten umhüllenden Quaders zur kleineren
- "x/z", d. h. das Verhältnis der größeren horizontalen Abmessung des kleinsten umhüllenden Quaders zur vertikalen
- "ist enthalten in", d. h. die Anzahl der Objekte, in denen der betrachtete Volumenkörper enthalten ist

Mit der optimalen Größe der verborgenen Schicht von sieben Neuronen und den sieben vorgegebenen Bauteilklassen entsteht das unten dargestellte Netz.

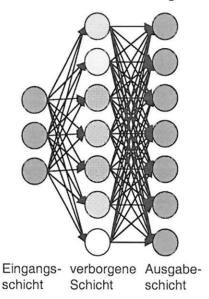


Abb. 4: Verwendetes Netz

2.2.4 Vorgehensweise bei Training und Test

Eine vollständig korrekte Klassifizierung der Trainingsdaten durch das trainierte Netz ist Voraussetzung für dessen Anwendung auf neue, nicht in der Trainingsbasis enthaltene Daten. Darum wurde der jeweilige Trainingserfolg zunächst durch Vorlage und Entscheidung über genau diejenigen Muster, mit denen das Netz zuvor trainiert wurde, überprüft. Sowohl der Anteil der Fehlentscheidungen als auch der mittlere Netzfehler beziehen sich also in Kapitel 2.2.5 ausschließlich auf die Trainingsbasis. Die Überprüfung der Generalisierungsfähigkeit durch neue, nicht in der Trainingsbasis enthaltene Muster erfolgt im darauf folgenden Kapitel 2.2.6.

2.2.5 Trainingsergebnisse

Das Netz wurde zunächst jeweils einzeln mit den zur Verfügung stehenden Produktmodellen trainiert. Nur das mit dem EN-Haus (Abb. 2) trainierte Netz konnte die vollständig korrekte Klassifizierung seiner Trainingsdaten erlernen (s. Abb. 7). Auch der mittlere Netzfehler pro Ausgabeneuron und Trainingsmuster konvergiert hier stetig zu einem sehr kleinen Wert, was auch auf ein erfolgreiches Training schließen lässt (s. Abb. 5). Betrachtet man die Entwicklung der Kantengewichte, exemplarisch in Abb. 6 dargestellt, so sieht man, dass ab etwa 900 Trainingsdurchläufen keine merklichen Veränderungen mehr stattfinden. Das Netz scheint also zur Ruhe gekommen zu sein, was den Abschluss des Trainings weiter bestätigt.

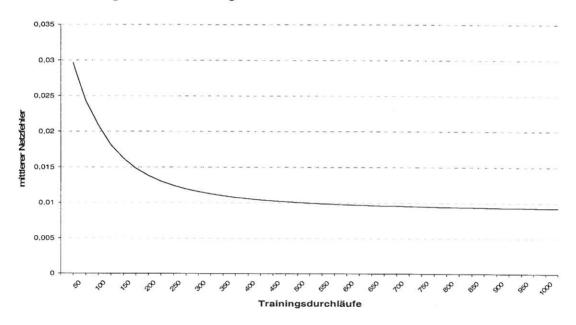


Abb. 5: Entwicklung des mittleren Netzfehlers beim Training mit dem EN-Haus

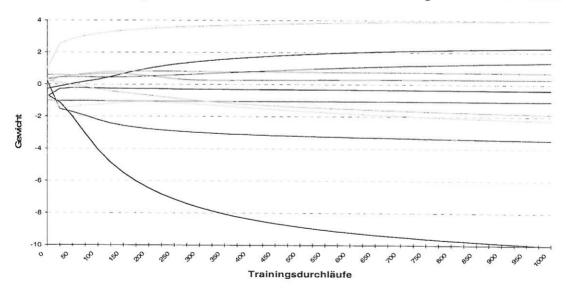


Abb. 6: Gewichte zwischen Eingangs- und verborgener Schicht (exemplarisch)

Die mit den beiden anderen Produktmodellen trainierten Netze klassifizierten auch nach über 1000 Trainingsdurchläufen noch bis zu dreißig Prozent ihrer eigenen Trainingsdaten fehl. Das Hauptproblem dieser beiden Netze war die Unterscheidung Fenster/Türe. Eine Analyse der entsprechenden Muster ergab, dass – bedingt durch die Größe der Fenster – eine Abgrenzung zu Türen mit den hier verwendeten Unterscheidungsparametern nicht eindeutig möglich ist.

2.2.6 Test auf Generalisierungsfähigkeit

Die Generalisierungsfähigkeit des mit dem EN-Haus trainierten Netzes wurde anhand der anderen beiden Produktmodelle sowie anhand eines fiktiven Testmodells für die im EN-Haus vorkommenden Bauteile getestet. Das fiktive Testmodell ist ein Teilmodell eines gewöhnlichen Wohnhauses, zusammengesetzt aus einfachen und eindeutig zu unterscheidenden Bauteilen.

Das mit dem EN-Haus trainierte Netz ist zwar nicht in der Lage, die Muster der beiden realen Modelle vollständig korrekt zu klassifizieren; bei Vorlage des fiktiven Testmodells jedoch ist das mit dem EN-Haus trainierte Netz schon nach 75 Trainingsdurchläufen in der Lage, alle Bauteile des – nicht trainierten - Testmodells korrekt zu klassifizieren (s. Abb. 7).

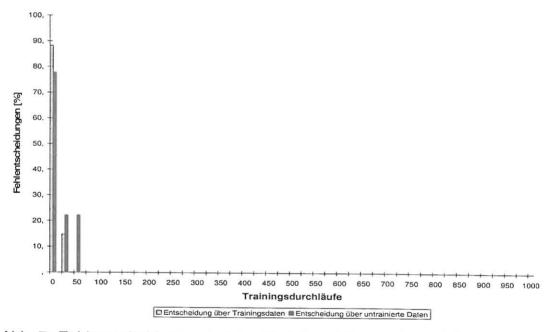


Abb. 7: Fehlentscheidungen bei den Trainingsdaten und untrainierten Daten

3 Zusammenfassung und Ausblick

Bauteilklassifizierung mit Hilfe neuronaler Netze ist prinzipiell möglich. Der Trainingserfolg und damit auch die Generalisierungsfähigkeit hängen jedoch von Art, Struktur und Anzahl der zum Training verwendeten Muster ab. Die verwendete Trainingsbasis ist folglich unbedingt geeignet zu wählen, eine Zusammenstellung einer Trainingsbasis

durch unselektierte und unaufbereitete Summation der Daten von bestehenden Modellen alleine führt nicht zum Erfolg.

Das hier auftretende Hauptproblem – die Unterscheidung von in ihren Abmessungen ähnlichen Fenstern und Türen – könnte durch die Übergabe weiterer, bisher nicht zur Verfügung stehender Information, z. B. den Abstand der Unterkante des Objekts zum enthaltenden Objekt, beseitigt werden. Dies entspricht der Einführung einer vierten Dimension im Eingaberaum und könnte die zuvor im dreidimensionalen Eingaberaum offenbar nicht trennbaren Klassen nun unterscheidbar machen. Das wiederum würde die gesuchte Funktion, die das künstliche neuronale Netz zu approximieren versucht, durch das Netz abbilden lassen und damit das hier auftretende Hauptproblem der Trennung Fenster/Türe beseitigen.

Literatur

- [1] BayFORREST F248: Vom Laseraufmaß zum Stoffflussmanagement für Altbauten. http://www.inf.bauwesen.tu-muenchen.de/forschung/bayforrest/f248, Stand 01.07.2005
- [2] Laurent, M.; Schinabeck, J.; Sorger, C.: From laser scan to product model. Abschlussbe-richt Software Lab 2002/03 am Lehrstuhl für Bauinformatik, München, 2003
- [3] Reif, G.: Moderne Aspekte der Wissensverarbeitung. Ein Interaktiver Lernbehelf für das Web Based Training, Diplomarbeit, Graz, 2000
- [4] Rojas R.: Theorie der neuronalen Netze. Eine systematische Einführung, Springer-Verlag, Berlin, 1991
- [5] Wikipedia. Die freie Enzyklopädie (eds.): Expertensystem, http://de.wikipedia.org, Stand 11.07.2005
- [6] Zell, A.: Simulation Neuronaler Netze, München, Addison-Wesley, 2000