

Nutzung Künstlicher Neuronaler Netze zur Optimierung von Tragwerken

Alexandra Fürbach

1. Einleitung

Am Beispiel der Dickenoptimierung eines Scheibentragwerks wurde die Möglichkeit der Nutzung Künstlicher Neuronaler Netze (KNN) für die Bestimmung der statischen Zulässigkeit des Tragwerks untersucht. Dies wurde in MATLAB unter Zuhilfenahme von Excel zur Dateneingabe umgesetzt.

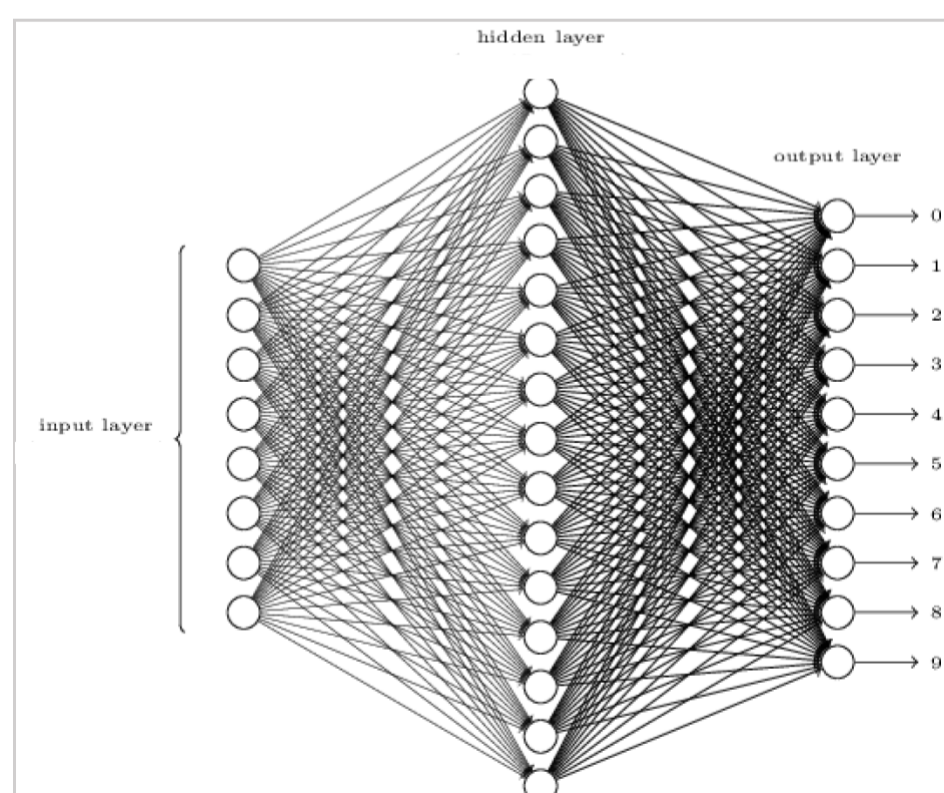
2. Programmablaufübersicht

Die Randbedingungen des Bauteils (wie Geometrie, Materialdaten und die Diskretisierung) werden in einem Excel-Input-File festgelegt. Dieses wird später in MATLAB eingelesen und zunächst zur Verschiebungs- und Schnittgrößenberechnung verwendet. Für die Verwendung von KNN müssen nun Trainingsdaten generiert werden. Ein separates Skript liest diese ein und trainiert ein KNN, welches anschließend weiterverwendet werden kann.

3. Finite-Elemente-Methode

Die Finite-Elemente-Methode (FEM) ist ein numerischer Berechnungsansatz zur Bestimmung von Verschiebungen und Schnittgrößen. Die zu untersuchende Struktur wird in sog. Finite Elemente diskretisiert, die je 4 Knoten besitzen. Die Berechnung erfolgt anschließend je Knoten unter Verwendung von bilinearen Ansatzfunktionen.

4. Künstliche Neuronale Netze



Struktur Künstlicher Neuronaler Netze

Künstliche Neuronale Netze stellen eine Teildisziplin der Künstlichen Intelligenz dar. Ihr Aufbau ist an die neurologische Struktur des

menschlichen Gehirns angelehnt: Neuronen sind über Gewichtungen (bzw. Synapsen) miteinander verbunden und leiten bestimmte Reize weiter. Im Rahmen der KI spricht man von sogenannten Layers, welche Neuronen in variabler Anzahl beherbergen. Zusätzlich kann das KNN hinsichtlich der Wahl der Aktivierungs-, und der Trainingsfunktion modifiziert werden. Um das KNN ausreichend gut trainieren zu können, ist es unabdingbar, dem Netz eine ausreichend große Anzahl an Trainingsdaten mit einer ausgeglichenen Verteilung für die möglichen Target-Werte zur Verfügung zu stellen.

Im Rahmen der Arbeit wurde mit einem sog. Patternnet gearbeitet, das in der Lage ist, Muster zu erkennen und hier die beiden Targets „zulässig“ und „nicht zulässig“ bereitstellt. Im Training wird dem KNN ein Inputvektor (hier: Elementdicken) und ein zugehöriger Target-Wert (hier: „zulässig“ oder „nicht zulässig“) zur Verfügung gestellt. Bei späterer Verwendung des KNN wird ebenfalls ein Inputvektor mit Elementdicken eingelesen, und das KNN trifft schlussendlich eine einzige Aussage über die Gesamtzulässigkeit des Inputvektors – also der Elementdickenverteilung.

5. Optimierung

Die Bauteildickenoptimierung wurde mit zwei Optimierungsverfahren durchgeführt – mit der „Halbierung der Dicken“ und der „Sensitivitätsanalyse“. Diese beiden Verfahren unterscheiden sich in ihrem Aufbau folgendermaßen: während bei Verwendung der „Halbierung der Dicken“ die Optimierung je Element ausgeführt wird und somit das Problem auftritt, dass die Elemente, die zu Beginn optimiert werden, stärker in ihrer Dicke reduziert werden als die, welche in den letzten Einträgen des Elementdickenvektors stehen, werden in der „Sensitivitätsanalyse“ je Iterationsschritt alle Elemente optimiert. Letzteres führt in Kombination mit KNN zu dem Problem, dass nicht ersichtlich ist, *welche* Elemente die in ihrer Dicke für die Statik Problematischen sind.

6. Fazit

Die Verwendung von KNN ist unter Berücksichtigung des für die Berechnung benötigten zeitlichen Rechenaufwands sinnvoll, wenn sich die zu Beginn im Input-File festgelegten Randbedingungen (Geometrie etc.) nicht mehr verändern. Sobald z. B. die Laststellung oder der E-Modul verändert wird, sind sowohl die Generierung neuer Trainingsdaten als auch ein erneutes Training erforderlich. Kann hingegen ein trainiertes KNN immer wieder eingesetzt werden, ist dies eine gute Möglichkeit, um Rechenzeit zu sparen, da die Antwort des KNN im Vergleich zu Ergebnissen aus einer FE-Rechnung deutlich schneller erhalten werden kann. Die beiden hier verwendeten Optimierungsverfahren eignen sich weiterhin in Kombination mit KNN aus den o. g. Gründen nur bedingt.