Universität Stuttgart
Fakultät für Bau- und Umweltingenieurwissenschaften

Michael Arnold

Künstliche neuronale Netze für statische Berechnungen

Motivation und Zielsetzung

Künstliche neuronale Netze (KNN) sind in der Lage, Sprache zu verstehen oder autonome Fahrzeuge zu steuern. Es ist ihnen möglich, bei einer Vielzahl von Eingangswerten eine Lösung zu finden. Basierend auf diese Fähigkeit wurde in dieser Arbeit ein KNN dazu verwendet, um statische Berechnungen einer Scheibe auszuführen. Für die Ausführung in MATLAB mussten die hierfür benötigten Trainingsdaten erst mit ANSYS erzeugt werden.

Künstliche neuronale Netze

KNN sind Strukturen, welche aus Knoten und Schichten bestehen. In diesem Aufbau werden durch die Verwendung von Gewichts- und Schwellenwerten Ausgabewerte berechnet, auf die jedes Element einen Einfluss besitzt.

$$\mathbf{a}^{(m)} = \mathbf{f}^{(m)}(\mathbf{W}^{(m)}\mathbf{a}^{(m-1)} + \mathbf{b}^{(m)})$$

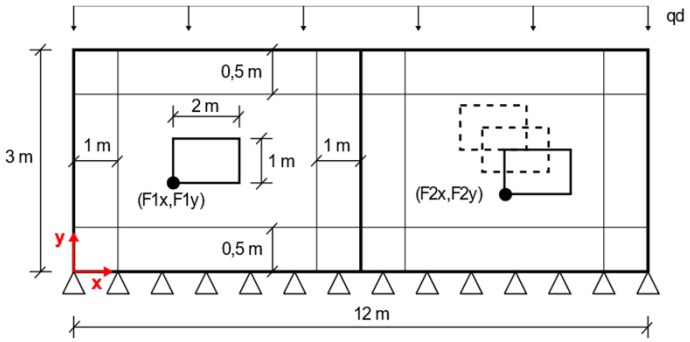
Da nach dem Erstellen des Netzes die Parameter noch nicht eingestellt sind, müssen diese durch ein Optimierungsverfahren angepasst werden. Hierfür wird der Levenberg-Marquardt-Algorithmus verwendet.

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left[\mathbf{J}^{\mathrm{T}}(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k) + \mu_k \mathrm{I}\right]^{-1} \mathbf{J}^{\mathrm{T}}(\mathbf{x}_k)\mathbf{e}(\mathbf{x}_k)$$

Dieser verbindet die Vorteile des Gradientenabstieg und des Gauß-Newton-Verfahrens, wodurch sich die Geschwindigkeit und Erfolgschance des Trainingsvorgangs erhöht.

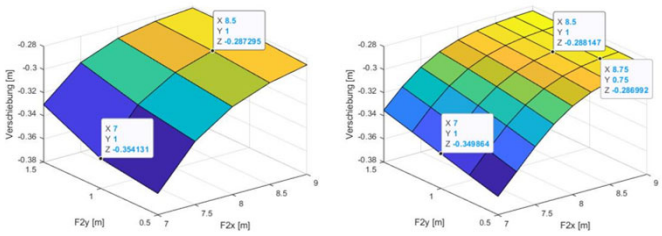
Berechnung der Trainingsdaten

Für die Erstellung der Trainingsdaten wurden mit ANSYS die Verschiebungen und Spannungen eines Beobachtungspunkts für verschiedene Scheibenausführungen ermittelt. Diese unterschieden sich dadurch, dass deren beide Öffnungen innerhalb eines vorgeschriebenen Bereichs frei positioniert werden konnten.



Training und Bewertung des Modells

Anschließend wurden mit den Daten in MATLAB ein KNN trainiert, welches in der Lage sein sollte, für noch offene Varianten der Scheibe eine Lösung zu berechnen. Hierfür wurden die bestehenden Strukturen der MATLAB Toolbox verwendet.



Verschiebungen, rechts: Trainingsdaten, links: Ausgabewerte des KNN

Dem trainierten Modell war es daraufhin möglich die Lücken innerhalb der Trainingsdaten sinnvoll zu vervollständigen.

Literatur

- Hagan, M. T., Demuth, H. B., Beale, M. H. & De Jesús, O., 2016. Neural network design. 2 Hrsg. s.l.:s.n.
- Goodfellow, I., Bengio, Y. & Courville, A., 2018. Deep Learning: Das umfassende Handbuch. 1. Hrsg. Frechen: mitp.

Betreuer: Tobias Willmann, M. Sc.