



## KI-Tool

### ENTWICKLUNG EINES SOFTWARE-TOOLS ZUR ANWENDUNG VON KÜNSTLICHER INTELLIGENZ

Autoren: Ebert, P.; Schmidt, S.; Vogel, C.; Kratzsch, A.  
Projektleiter: Prof. Dr.-Ing. A. Kratzsch  
Ansprechpartner: M. Eng. Christian Vogel, c.vogel@hszg.de  
Projektlaufzeit: 01.01.2021 – 31.12.2021

### Motivation

Die künstliche Intelligenz (KI) mit deren wesentlichen Teilgebieten Soft-Computing/Maschinelles Lernen und kognitive Systeme hat in den letzten Jahren durch die exponentielle Zunahme der Leistungsfähigkeit von Rechentechnik eine neue, entscheidende Entwicklungsstufe erreicht und wird mittelfristig in alle Lebensbereiche (z.B. Mobilität, Energie, Produktion, Finanzen, Gesundheit, Landwirtschaft, Medien, Militärwesen) auf vielfältige Weise vordringen.

Zukünftig werden die Regionen und Länder sowie Unternehmen wirtschaftlich besonders erfolgreich sein, welche die Kompetenzen im Bereich der KI maßgeblich vorantreiben, Standards setzen und die einhergehenden Risiken beherrschen.

Die umfassende Programmierung sowie Implementierung von KI lässt sich aktuell zum großen Teil einzig durch die Anwendung anspruchsvoller Programmiersprachen und deren Frameworks realisieren, wodurch nur ein kleiner spezialisierter Personenkreis den Umgang mit KI beherrscht.

### Zielstellung

Das globale Ziel des Vorhabens ist die Entwicklung eines Software-Tools (KI-Tool), welche eine vereinfachte Erstellung, Bewertung und Implementierung von KI-Methoden bereit stellt. Somit können auch Anwender ohne Programmierkenntnisse KI-Applikationen entwickeln.

### Das Perzeptron

Eines der einfachsten aber immer noch aktuellen neuronalen Netze ist das Perzeptron. Dieses ist in der Lage mehrere Eingangswerte zu verarbeiten und ein oder mehrere Ausgangswerte in Form von Wahrscheinlichkeiten (Wert zwischen 0 und 1) auszugeben. Die Einfachheit und ein breites Anwendungsfeld qualifiziert das Perzeptron für das KI-Tool.

Das Multilayer-Perzeptron (MLP) besteht aus einer Eingangsschicht (input layer), einer oder mehreren Zwischenschichten (hidden layer) und einer Ausgangsschicht (output layer).

Die untere Darstellung (Abbildung 1) zeigt ein Perzeptron mit einem aus 3 Eingangsneuronen bestehenden input layer (x0, x1, x2), einem aus 2 Neuronen bestehenden hidden layer (h0, h1) und einem aus 2 Ausgangsneuronen bestehenden output layer (y0, y1).

Die in den Layern liegenden Neuronen sind mit den Neuronen des nachfolgenden Layers über Gewichtungen (w0 - w9) verbunden. Die einzelnen Neuronen sind als Platzhalter eines Zahlenwertes zu verstehen. Die Berechnung dieses Zahlenwertes ist beispielhaft im grün hinterlegten unteren Bildausschnitt zu sehen (Abbildung 1, Berechnungsschritte Neuron h0). In der Praxis erfolgt diese Berechnung Layer-weise vom input layer ausgehend bis zum output layer (forward pass). Dies geschieht unter anderem mithilfe der Matrizen-Multiplikation, wodurch die gesamte Berechnung aufgrund der entsprechenden Software-Bibliotheken sehr performant ausgeführt werden kann.

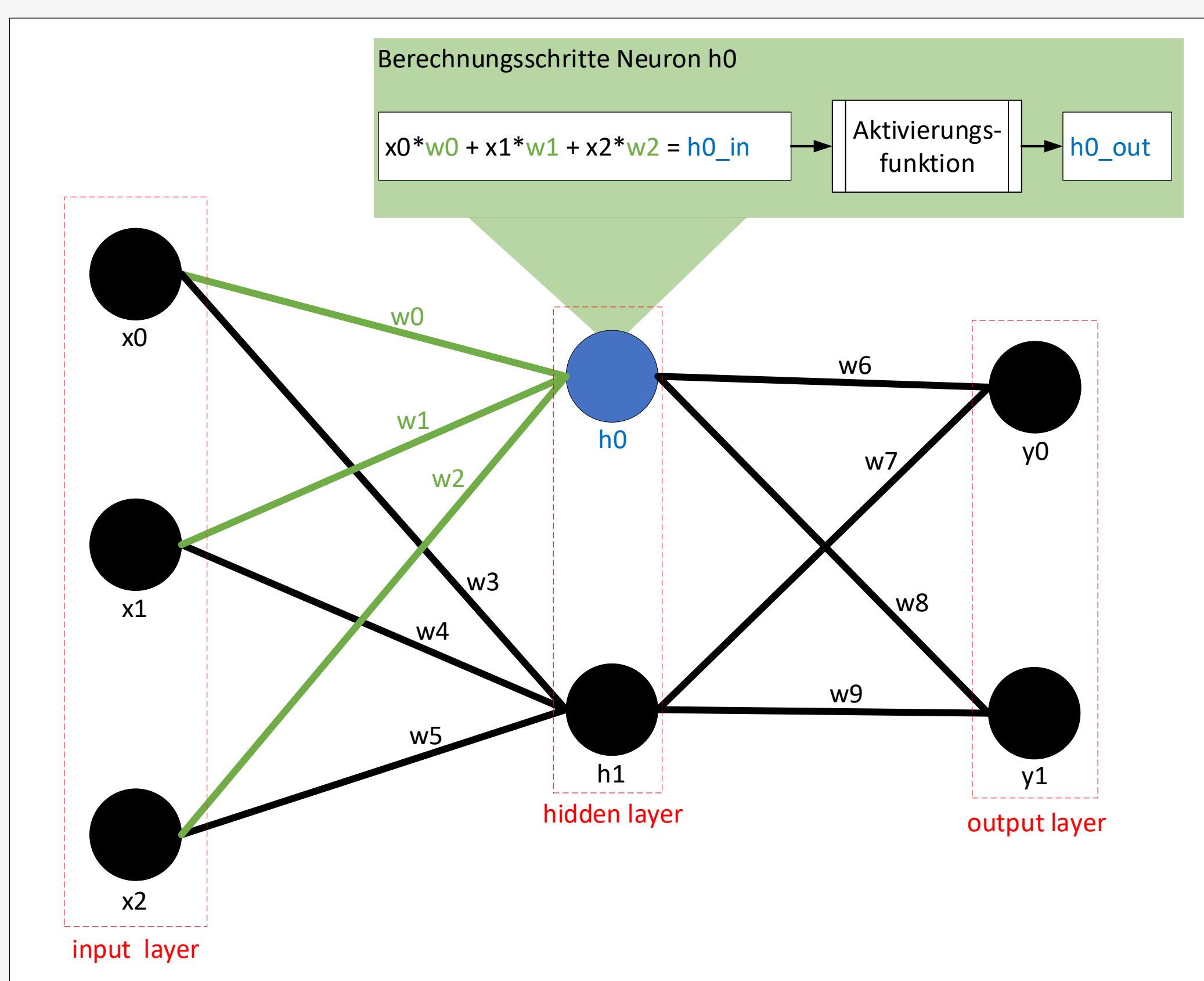


Abbildung 1: Veranschaulichung eines Multilayer-Perzeptrons

Zur Berechnung des Neuronen-Ausgangs (Abbildung 1, h0\_out) wird eine Aktivierungsfunktion verwendet. Diese dient dazu, die Neuronen-Eingangswerte (Abbildung 1, h0\_in) zu verallgemeinern bzw. zu approximieren und ist für ein funktionierendes neuronales Netz unerlässlich. Es gibt unterschiedliche Aktivierungsfunktionen zur Auswahl, welche in Abhängigkeit zur Netztopologie entsprechend ausgewählt werden.

Einer der häufigsten Anwendungen für das Perzeptron ist die Objekterkennung und die Klassifizierung. Für die Klassifizierung werden aussagekräftige Eigenschaften als Eingangsneuron benutzt (z. B. x0 = Größe, x1 = Gewicht, x2 = Felllänge). Jedes Ausgangsneuron steht für eine zu differenzierende Klasse (z. B. Unterscheidung von 2 Hunderassen, y0... Schäferhund, y1... Jack Russel Terrier).

Für die Objekterkennung wird ein digitales Bild dem input layer zugeführt. Dabei wird das Bild zuvor komprimiert und formatiert, um die Anzahl der Eingangsneuronen gering zu halten.

Quelle: Ertel, W. (2016). Grundkurs Künstliche Intelligenz: Eine praxisorientierte Einführung (4. Aufl. 2016.). Wiesbaden: Springer Vieweg.

### KI-Tool Anforderungen

Zur Bereitstellung einer benutzerfreundlichen und technisch umfangreichen Benutzeroberfläche mussten System-Anforderungen an das zu entwickelnde KI-Tool definiert werden:

- Einstellungsmöglichkeiten für die Dimensionierung der Netzes (Anzahl und Größe der Layer)
- Einstellungsmöglichkeiten für die Konfigurierbarkeit des Trainings
- Festlegung eines einheitlichen und geeigneten Datei-Standard für gelabelte Daten
- Informationsrückfluss muss für den Nutzer in geeigneter Form stattfinden, bspw. Anzeige der Trainingskurve
- Festlegung eines geeigneten Datei-Format für das finale trainierte Netz, um eine breite Anwendung sicher zu stellen
- Für die Analyse und Beurteilung der Anwendungsfähigkeit des trainierten Netzes muss ein geeignetes Verfahren entwickelt und dem Anwender im Rahmen des KI-Tools bereitgestellt werden.

### Software-Architektur

Von den System-Anforderungen und der grundsätzlichen Funktionsweise eines Perzeptron abgeleitet, wurde die folgende SW-Architektur entwickelt (Abbildung 2), welche die wichtigsten Funktionen des KI-Tools abbildet.

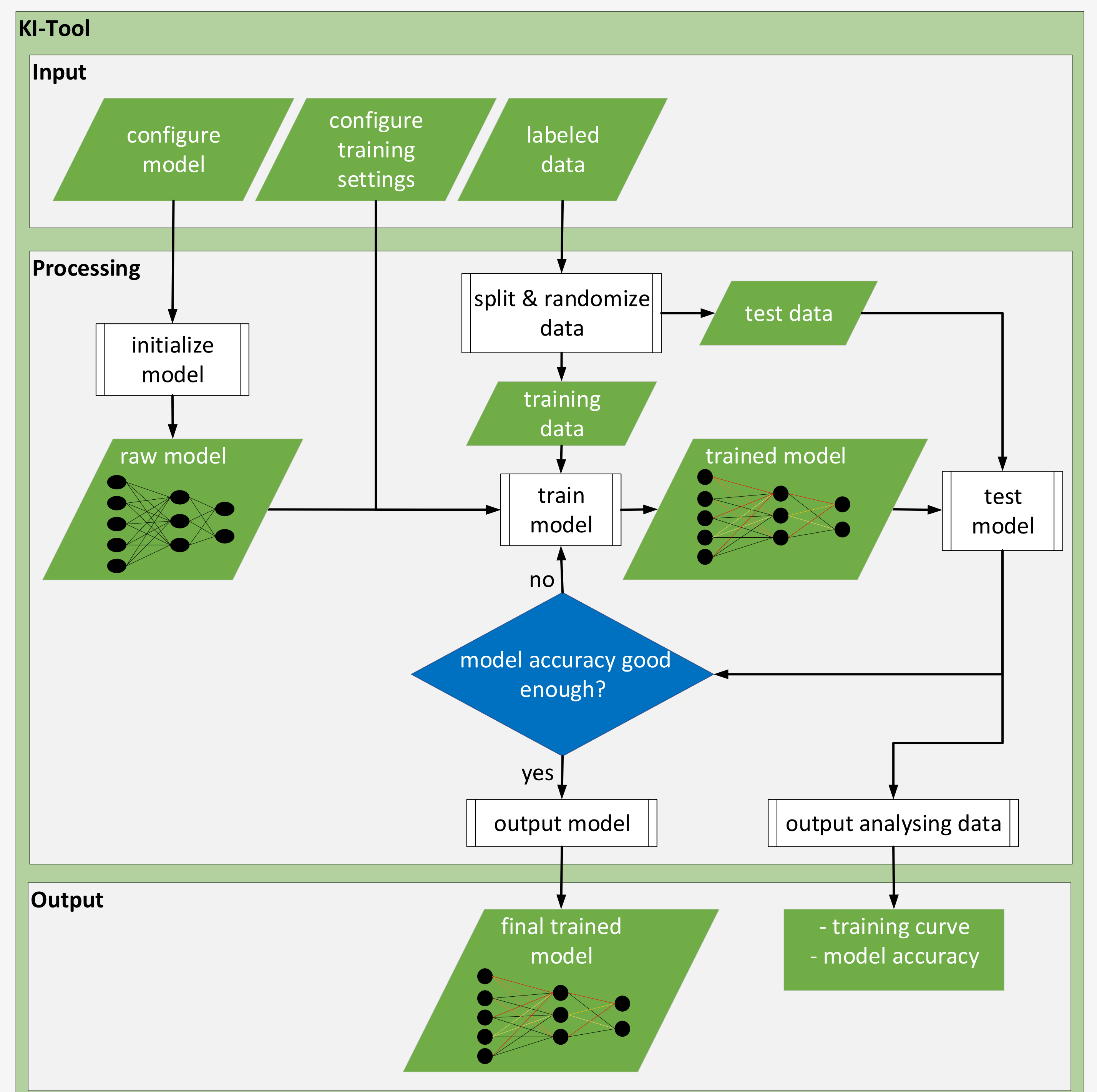


Abbildung 2: Software-Architektur des KI-Tools

Über den Input muss der Anwender dem KI-Tool eine ausreichende Menge von gelabelten Daten zur Verfügung stellen (Abbildung 2, labeled data) und z. B. über einen Dateipfad angeben. Des Weiteren konfiguriert der Anwender die Größe des Netzes (Abbildung 2, configure model) und die Trainingsrandbedingungen (Abbildung 2, configure training settings).

Das KI-Tool wird dann selbstständig das Netz trainieren und testen (Abbildung 2, Processing). Abschließend wird ein trainiertes Netz abgelegt und zur Weiternutzung zur Verfügung gestellt (Abbildung 2, final trained model). Weiterhin wird dem Anwender die Trainings- und Test-Analysedaten angezeigt, um die Robustheit des Netzes beurteilen zu können.

### Ausblick

Mithilfe einer modularen Programmierweise wird die Erweiterungsmöglichkeit für das KI-Tool sichergestellt. Das implementierte Multilayer-Perzeptron bildet eine Arbeitsgrundlage für vielfältige Anwendungen. Für einige anspruchsvollere Applikationen ist dies jedoch nicht ausreichend. Im weiteren Verlauf des Projektes wird deshalb angestrebt, fortgeschrittenere Netztypen zu implementieren wie z. B. Faltungsnetzwerke.

