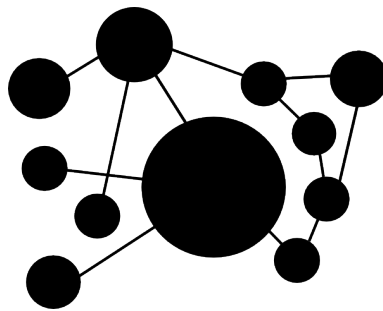


# Seminararbeit

## **Extrapolation von Zeitreihen mit Hilfe von künstlichen neuronalen Netzen am Beispiel von Börsenprognosen**



Sebastian Schötteler – Matrikelnummer 24 29 289

Benedikt Hofrichter – Matrikelnummer 22 72 198

21. November 2015



**TECHNISCHE HOCHSCHULE NÜRNBERG**  
GEORG SIMON OHM

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>III</b>
<b>Tabellenverzeichnis</b>	<b>IV</b>
<b>Formelverzeichnis</b>	<b>V</b>
<b>Abkürzungsverzeichnis</b>	<b>VI</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Ziel und Aufbau dieser Arbeit . . . . .	2
<b>2 Konzeption</b>	<b>5</b>
2.1 Fachliche Konzeption der Anwendung . . . . .	5
2.1.1 Beschreibung der Anwendung . . . . .	5
2.1.2 Mockup der Anwendung . . . . .	5
2.2 Konzeption des künstlichen neuronalen Netzes . . . . .	5
2.2.1 Typ des künstlichen neuronalen Netzes . . . . .	6
2.2.2 Grundlegende Topologie des künstlichen neuronalen Netzes . . . . .	7
2.2.3 Lernverfahren des künstlichen neuronalen Netzes . . . . .	8
2.3 Beschreibung von Frameworks . . . . .	8
2.3.1 SNNS . . . . .	8
2.3.2 JavaNNS . . . . .	8
2.3.3 JavaNNS . . . . .	8
2.3.4 Neuroph . . . . .	8
2.4 Wahl des geeignetsten Frameworks . . . . .	8
<b>3 Umsetzung</b>	<b>9</b>
3.1 Erstellung des künstlichen neuronalen Netzes . . . . .	9
3.1.1 Ermittlung der genauen Topologie . . . . .	9
3.1.2 Ermittlung der Transferfunktion . . . . .	11

3.1.3	Ermittlung der Lernregel und Lernrate . . . . .	12
3.2	Das erstellte künstliche neuronale Netz . . . . .	12
3.3	Implementierung der Anwendung . . . . .	12
3.4	Zusammenführung des Netzes mit der Anwendung . . . . .	12
<b>4</b>	<b>Beschreibung der Anwendung</b>	<b>13</b>
4.1	Elemente der GUI . . . . .	13
4.2	Architektur der Anwendung . . . . .	13
<b>5</b>	<b>Fazit</b>	<b>14</b>
	<b>Literaturverzeichnis</b>	<b>15</b>

## Abbildungsverzeichnis

1.1	Vorgehensdiagramm . . . . .	3
2.1	Mockup der Anwendung . . . . .	5
2.2	Grundlegendes Konzept des KNN . . . . .	7
3.2	KNN nach 4-09-1 Muster . . . . .	9
3.1	KNN nach 4-05-1 Muster . . . . .	10
3.3	KNN nach 4-13-1 Muster . . . . .	10
3.4	Sigmoide Funktion . . . . .	11
3.5	Tangens Hyperbolicus Funktion . . . . .	12

## Tabellenverzeichnis

3.1	Jeweilige Topologien & korrespondierende MSE . . . . .	10
3.2	Jeweilige Transferfunktionen & korrespondierende MSE . . . . .	11
3.3	Jeweilige Lernregeln und beste Lernraten & korrespondierende MSE . . . . .	12

## Formelverzeichnis

Formel 3.1	Optimale Anzahl Neuronen in der versteckten Schicht . . . . .	9
Formel 3.2	Sigmoide Funktion . . . . .	11
Formel 3.3	Tanh Funktion . . . . .	11

## Abkürzungsverzeichnis

<b>KNN</b>	Künstliches neuronales Netz.....	1
<b>DAX</b>	Deutscher Aktienindex.....	2
<b>GUI</b>	Graphical User Interface .....	4

# 1 Einleitung

## 1.1 Motivation

Die Untersuchung und Extrapolation von Zeitreihen ist ein bedeutendes Thema in zahlreichen Gebieten. Typische Anwendungsbereiche sind zum Beispiel die Prognose von Wetterdaten, von Therapieverläufen in der Medizin, von Arbeitslosenzahlen auf dem Arbeitsmarkt sowie von Börsenkursen. Um eine Zeitreihe möglichst genau zu extrapolieren, wird auf mehreren Hilfsmitteln zurückgegriffen. Einer dieser Hilfsmittel können künstliche neuronale Netze sein.

Bei künstlichen neuronalen Netzen handelt es sich um Netzwerke mit künstlichen Neuronen als Knoten, die mittels gerichtete Verbindungen Eingaben einlesen, weiterverarbeiten und die daraus resultierenden Ergebnisse an weitere Neuronen weiterleiten oder als Ergebnis ausgeben. Bei der Terminologie von künstlichen neuronalen Netzen wird bewusst auf Begriffen der Biologie zurückgegriffen, da künstliche neuronale Netze das biologische Gehirn als Vorbild nutzen und dessen Herangehensweise auf analoger Weise umzusetzen zu versuchen. Man nennt das Verfahren dieser Netze aus diesem Grunde auch *naturanaloge Verfahren*.

Warum sind diese Netze nun so interessant für Prognosen? Das Erstellen von zum Beispiel Börsenprognosen basiert in der Regel auf Auswertungen von Informationen verschiedenster Quellen. Die Art von Auswertungen, wie Börsenexperten sie vornehmen, ist weder vollständig formalisierbar noch besonders exakt, da uneinheitlich und in weiten Zügen intuitiv. Besonders schwer ist hier das Ermitteln von *nichtlinearen Zusammenhängen*. Ein Künstliches neuronales Netz (KNN) ist jedoch in der Lage, diese Zusammenhänge zu finden und diese objektiv und vorurteilsfrei zu bewerten. Somit sind künstliche neuronale Netze prinzipiell in der Lage, jedes beliebige Muster in jedem beliebigen Markt zu erkennen - auch solche, die noch nie zuvor von irgend jemand entdeckt wurden.

Ob und wie gut KNN zur Prognose geeignet sind, ist pauschal nicht zu beantworten. In manchen Gebieten mag die Prognosefähigkeit durchaus ausreichen. Je höher die geforderte Genauigkeit jedoch wird, desto diskutabler wird ein Einsatz von KNN. Eine typische



Grauzone ist hier die Prognose von Börsenkursen. Während Befürworter auf die Eigenschaft von KNN hinweisen, nichtlineare Muster zu erkennen und entsprechend zu behandeln, argumentieren Kritiker, dass ein System, das dem menschlichen Lernen nachempfunden wurde, die gleichen Fehler machen wird wie der Mensch. Generell ist jedoch zu sagen, dass die Prognosequalität von künstlichen KNN über die Jahre stets angestiegen ist.

## 1.2 Ziel und Aufbau dieser Arbeit

Als Ziel dieser Seminararbeit wird versucht ein KNN zu erschaffen, das prinzipiell in der Lage ist, den Börsenkurs des Deutschen Aktienindex (DAX) vorherzusagen. Der Fokus dieser Arbeit liegt hierbei nicht auf möglichst genaue Prognosen, sondern auf das Erlangen eines Grundverständnisses über die Funktionsweise von KNN. Trotzdem ist ein bestimmtes Maß an Genauigkeit ein wichtiges Kriterium, das es zu berücksichtigen gilt. Dieses KNN soll anschließend in einer Anwendung überführt werden, die die Prognosen und die dazugehörige Prognosequalität visualisiert.

Zur Erlangung des Ziels der Seminararbeit müssen mehrere Teilschritte durchgeführt werden. Das Vorgehen während dieser Seminararbeit wird im folgenden Diagramm visualisiert:

Zunächst wird mit einer Konzeptionsphase begonnen. In dieser wird zunächst eine fachliche Konzeption der Anwendung erstellt, in der die Funktionalitäten der Anwendungen genau spezifiziert werden. Abgerundet wird diese Fachkonzeption durch ein Mockup. Nachdem die Konzeption der Anwendung abgeschlossen wurde, wird das benötigte KNN für diese Anwendung konzeptioniert. Dabei ist zu ermitteln, welches KNN hinsichtlich Typ, Topologie und Lernverfahren am Besten für die Prognose des Deutschen Aktienindex (DAX) geeignet ist. Zur Erstellung von KNN stehen mehrere Frameworks zur Verfügung, drei dieser Frameworks werden genauer analysiert und im Anschluss das für diese Seminararbeit geeignetste Framework ausgewählt.

Nachdem die Konzeptionsphase abgeschlossen ist, wird mit der Umsetzung der Anwendung begonnen. Die Umsetzungsphase besteht im Wesentlichen aus drei Teilen. Zu einem aus der Implementierung des KNN und der Anwendung und zum anderen aus der Zusammenführung der beiden Elemente.

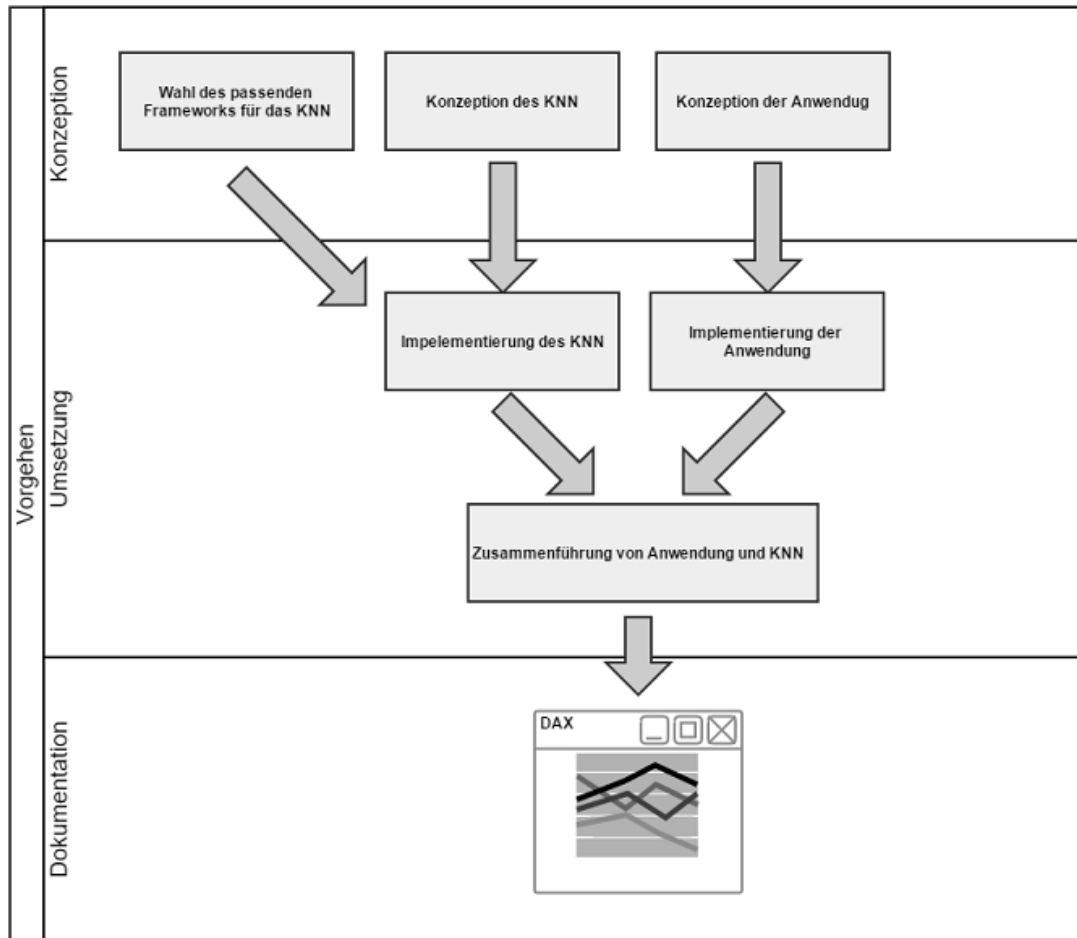


Abbildung 1.1: Vorgehensdiagramm

Nachdem die Anwendung entsprechen umgesetzt wurde, wird diese beschrieben. Konkret werden hierbei die einzelnen Elemente des Graphical User Interface (GUI) sowie die technische Architektur der Anwendung beschrieben.

## 2 Konzeption

### 2.1 Fachliche Konzeption der Anwendung

#### 2.1.1 Beschreibung der Anwendung

#### 2.1.2 Mockup der Anwendung

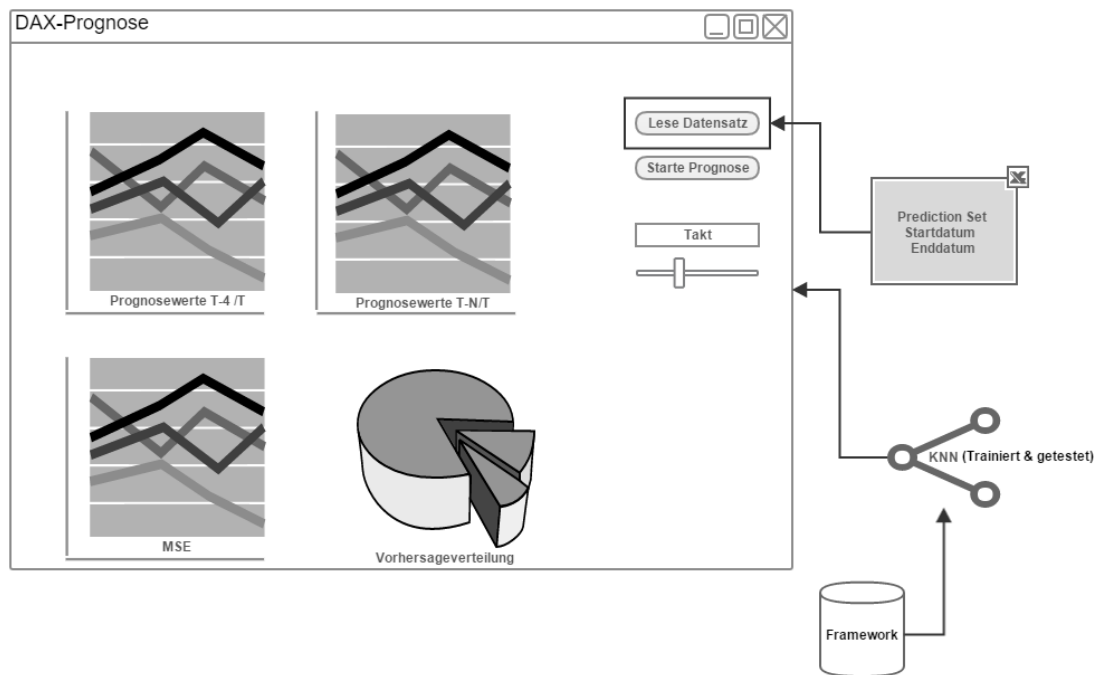


Abbildung 2.1: Mockup der Anwendung

### 2.2 Konzeption des künstlichen neuronalen Netzes

in den folgenden drei Abschnitten wird auf die Konzeption des KNN eingegangen.

### 2.2.1 Typ des künstlichen neuronalen Netzes

Grundsätzliche lassen sich KNN in zwei Oberklassen unterteilen. Die Hetero-Assoziativen Netze sowie die Auto-Assoziativen Netze. Hetero-Assoziative Netze bilden einen Vektor  $A$  der Länge  $n$  auf einem Vektor  $B$  einer meist kürzeren Länge  $m$   $\{m \in \mathbb{N} | m \leq n\}$  ab. Auto-Assoziative Netze wiederum bilden einen Eingabevektor der Länge  $n$  auf einem Ausgabevektor der gleichen Länge ab.

Innerhalb dieser zwei Klassen lassen sich KNN wiederum in mehr Modelle aufteilen. folgende Tabelle liefert eine Übersicht der bekanntesten Modelle von KNN unterteilt in Klassen:

Hetero- assoziative Netzmodelle	Auto-assoziative Netzmodelle
(M)Adaline	Hopfield-Netze
Perzeptron	Boltzmann Maschinen
Multilayerperzeptron	-

Da es sich bei dem DAX-Kurs um einen skalaren Wert handelt, der auf Grund mehrerer vorhergehender DAX-Kurse prognostiziert wird, wir ein Netzmodell aus der Klasse der Hetero-Assoziativen Netze benötigt. Für das Vorhaben ist demnach nur die linke Spalte relevant. Ein weiteres wichtiges Kriterium des benötigten Netzwerkmodells ist die Fähigkeit, mit nicht separierbaren Funktionen zu arbeiten.

Seien  $X_0$  and  $X_1$  zwei Datenmengen im  $n$ -dimensionalen euklidischen Raum. Dann sind die Mengen  $X_0$  and  $X_1$  genau dann als „linear separierbar“, wenn es  $n + 1$  Werte  $w_1, w_2, \dots, w_n, k$ , gibt, sodass jeder Punkt  $x \in X_0$  die Bedingung  $\sum_{i=1}^n w_i x_i > k$  erfüllt und jeder Punkt  $x \in X_1$  die Bedingung  $\sum_{i=1}^n w_i x_i < k$  erfüllt.

### 2.2.2 Grundlegende Topologie des künstlichen neuronalen Netzes

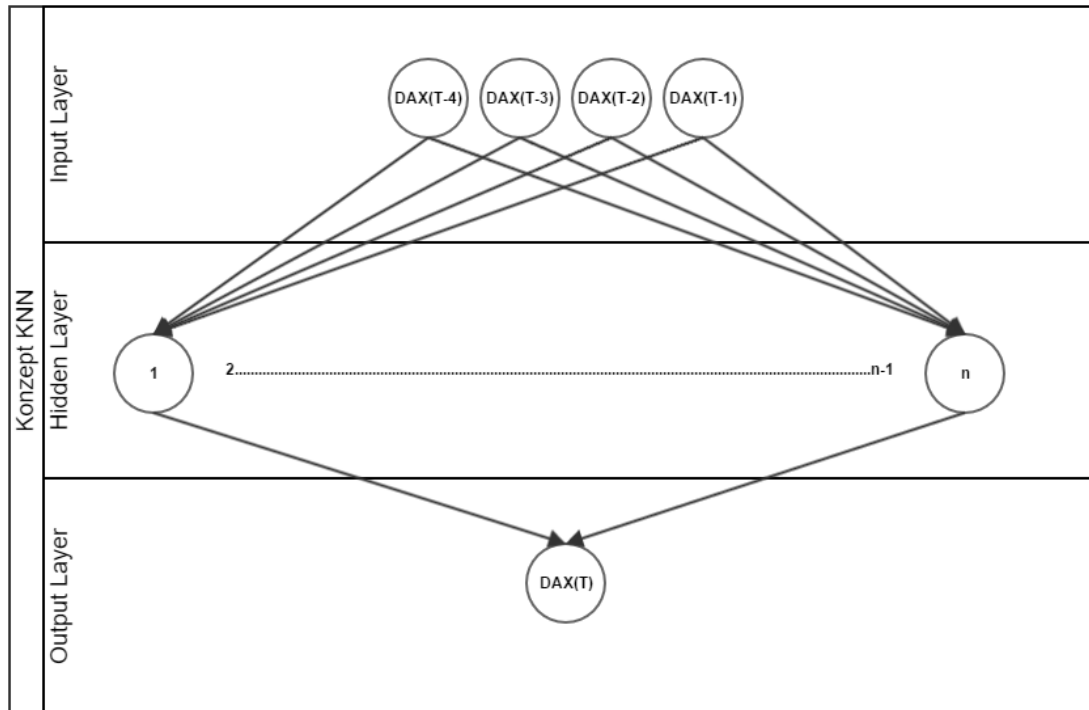


Abbildung 2.2: Grundlegendes Konzept des KNN

### **2.2.3 Lernverfahren des künstlichen neuronalen Netzes**

## **2.3 Beschreibung von Frameworks**

### **2.3.1 SNNS**

### **2.3.2 JavaNNS**

### **2.3.3 JavaNNS**

### **2.3.4 Neuroph**

## **2.4 Wahl des geeignetsten Frameworks**

## 3 Umsetzung

### 3.1 Erstellung des künstlichen neuronalen Netzes

beschreibung folgende 3 abschnitte sukzessive normalisierungsfunktion beispieldatensatz  
(training / test) verteilung(training/test) menge(training/test) Anzahl Durchläufe: 100k  
Lernrate 0.5

#### 3.1.1 Ermittlung der genauen Topologie

In der Literatur wird dabei oft auf die folgende Gleichung zur Ermittlung der optimalen Menge an Neuronen der versteckten Schicht angegeben:

$$N_h = \frac{N_d}{10 * (N_i + N_o)} \quad (3.1)$$

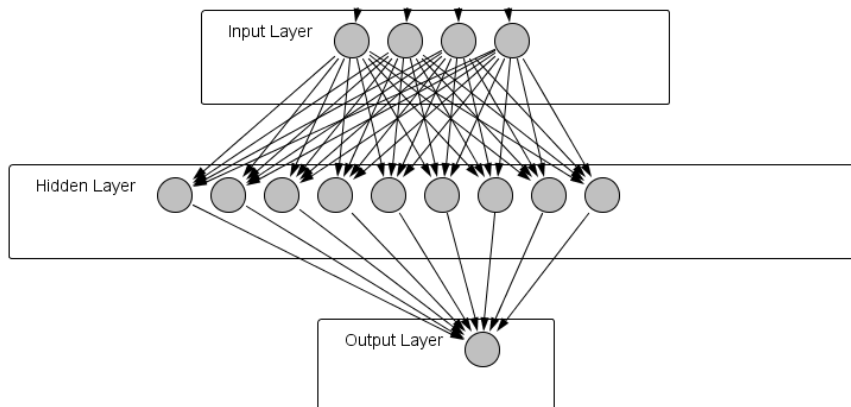


Abbildung 3.2: KNN nach 4-09-1 Muster



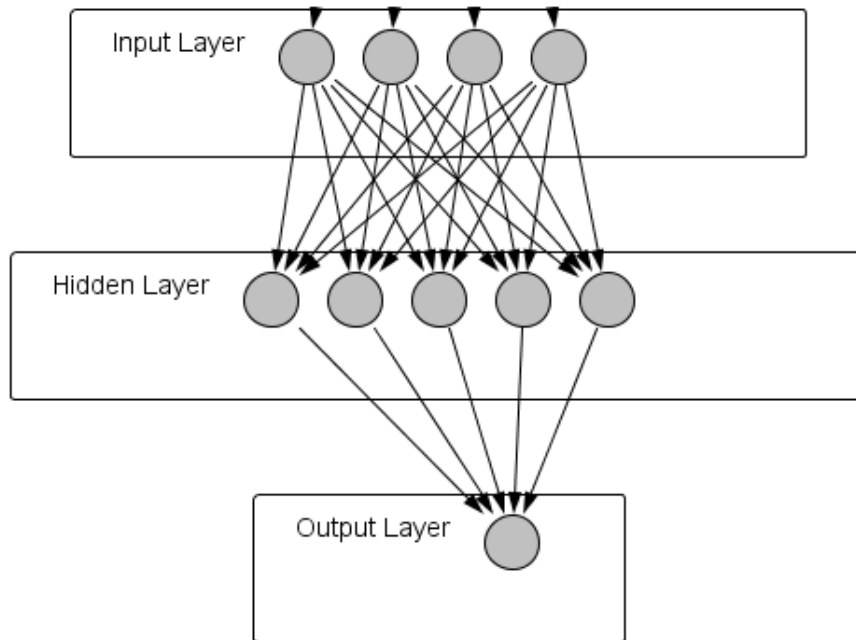


Abbildung 3.1: KNN nach 4-05-1 Muster

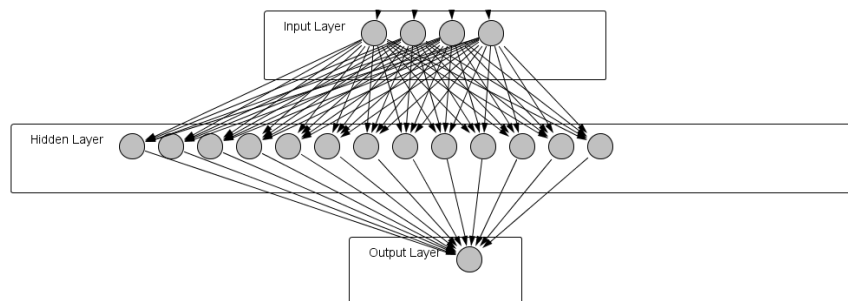


Abbildung 3.3: KNN nach 4-13-1 Muster

Topologie	Mean Squared Error
4-05-1	0.000
4-09-1	0.000
4-13-1	0.000

Tabelle 3.1: Jeweilige Topologien &amp; korrespondierende MSE

Transferfunktion	Mean Squared Error
4-05-1	0.000
4-09-1	0.000
4-13-1	0.000

Tabelle 3.2: Jeweilige Transferfunktionen &amp; korrespondierende MSE

### 3.1.2 Ermittlung der Transferfunktion

Sigmoide Funktion:

$$f(x) = \frac{1}{1 + e^{-cx}} \quad (3.2)$$

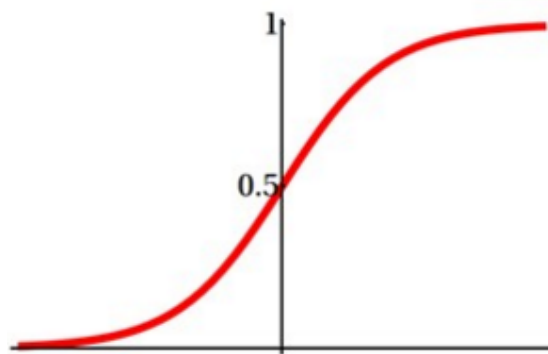


Abbildung 3.4: Sigmoide Funktion

Tangens Hyperbolicus:

$$f(x) = \tanh(x) \quad (3.3)$$

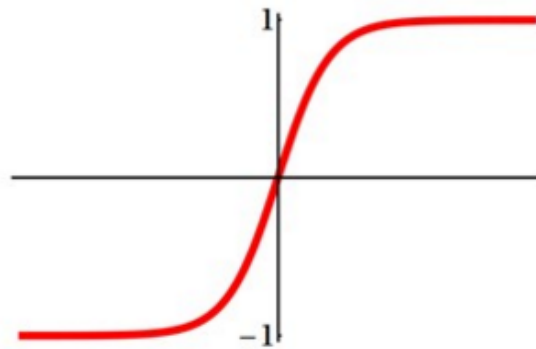


Abbildung 3.5: Tangens Hyperbolicus Funktion

### 3.1.3 Ermittlung der Lernregel und Lernrate

Lernregel	beste Lernrate	Mean Squared Error
Backpropagation	0.000	0.000
Momentum Backpropagation	0.000	0.000
Resilient Propagation	0.000	0.000

Tabelle 3.3: Jeweilige Lernregeln und beste Lernraten &amp; korrespondierende MSE

## 3.2 Das erstellte künstliche neuronale Netz

## 3.3 Implementierung der Anwendung

## 3.4 Zusammenführung des Netzes mit der Anwendung

## **4 Beschreibung der Anwendung**

### **4.1 Elemente der GUI**

### **4.2 Architektur der Anwendung**

## **5 Fazit**

Das prognostizieren von Börsenkursen mittels künstlichen neuronalen Netzen ist möglich.

## **Literaturverzeichnis**