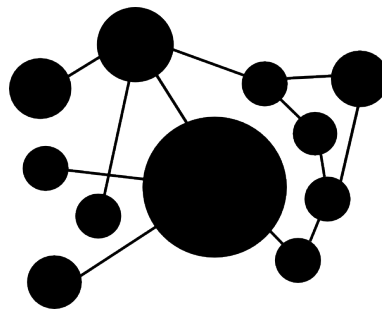


Seminararbeit

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

25. November 2015



TECHNISCHE HOCHSCHULE NÜRNBERG
GEORG SIMON OHM

Inhaltsverzeichnis

Abbildungsverzeichnis	III
Tabellenverzeichnis	IV
Formelverzeichnis	V
Abkürzungsverzeichnis	VI
1 Einleitung	1
1.1 Motivation	1
1.2 Ziel und Aufbau dieser Arbeit	2
2 Konzeption	4
2.1 Fachliche Konzeption der Anwendung	4
2.1.1 Grundidee	4
2.1.2 Mockup	4
2.2 Konzeption des künstlichen neuronalen Netzes	5
2.2.1 Wahl des Netztyps	5
2.2.2 Wahl der Grundlegenden Topologie	6
2.2.3 Wahl des Lernverfahrens	7
2.3 Beschreibung von KNN-Frameworks	7
2.3.1 SNNS	7
2.3.2 JavaNNS	7
2.3.3 Neuroph	7
2.4 Wahl des geeignetsten Frameworks	7
3 Umsetzung	8
3.1 Erstellung des künstlichen neuronalen Netzes	8
3.1.1 Ermittlung der konkreten Topologie	8
3.1.2 Ermittlung der Transferfunktion	10
3.1.3 Ermittlung der Lernregel und Lernrate	11
3.2 Das erstellte künstliche neuronale Netz	11

3.3	Implementierung der Anwendung	11
3.4	Zusammenführung des Netzes mit der Anwendung	11
4	Beschreibung der Anwendung	12
4.1	Elemente der GUI	12
4.2	Architektur der Anwendung	12
5	Fazit	13

Abbildungsverzeichnis

1.1	Vorgehensdiagramm	3
2.1	Mockup der Anwendung	4
2.2	Grundlegendes Konzept des KNN	7
3.2	KNN nach 4-09-1 Muster	8
3.1	KNN nach 4-05-1 Muster	9
3.3	KNN nach 4-13-1 Muster	9
3.4	Sigmoide Funktion	10
3.5	Tangens Hyperbolicus Funktion	11

Tabellenverzeichnis

3.1	Jeweilige Topologien & korrespondierende MSE	9
3.2	Jeweilige Transferfunktionen & korrespondierende MSE	10
3.3	Jeweilige Lernregeln und beste Lernraten & korrespondierende MSE	11

Formelverzeichnis

Formel 3.1	Optimale Anzahl Neuronen in der versteckten Schicht	8
Formel 3.2	Sigmoide Funktion	10
Formel 3.3	Tanh Funktion	10

Abkürzungsverzeichnis

KNN	Künstliches neuronales Netz	1
DAX	Deutscher Aktienindex.....	2
GUI	Graphical User Interface	2

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.

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

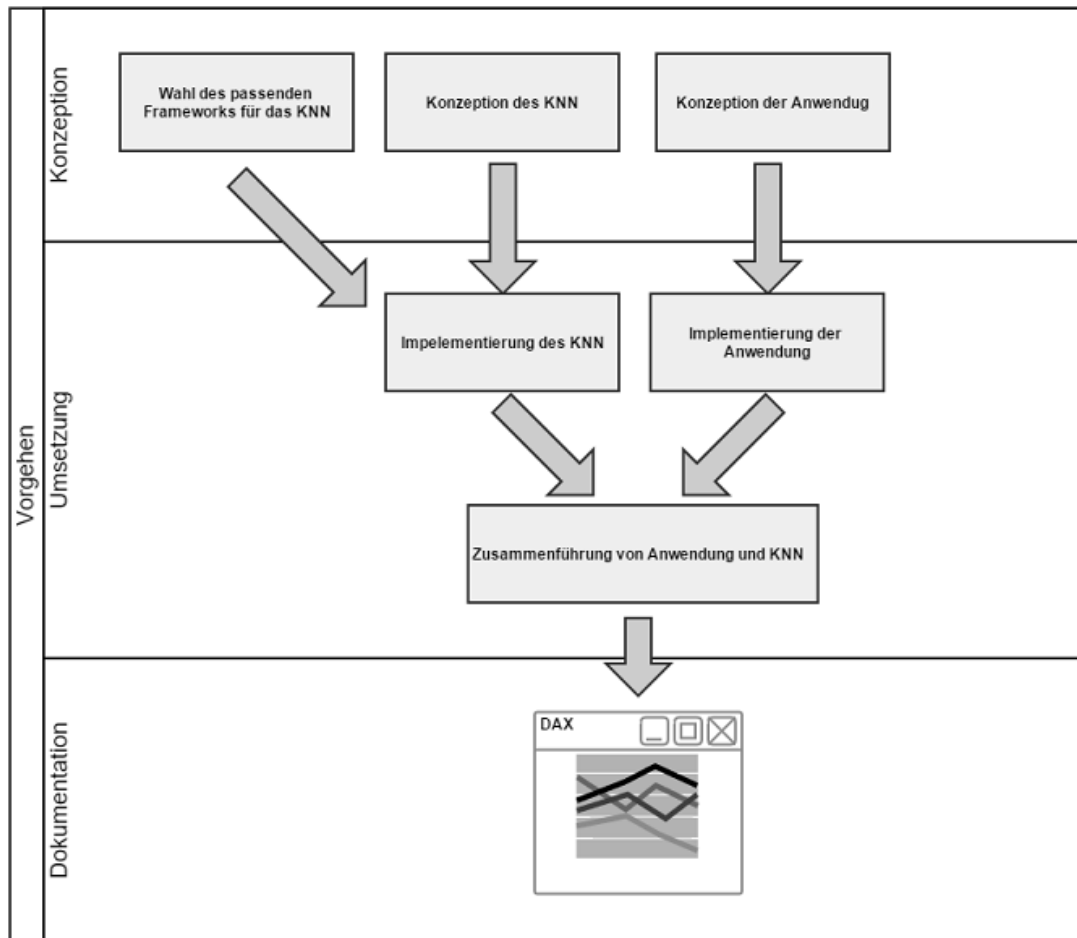


Abbildung 1.1: Vorgehensdiagramm

2 Konzeption

In diesem Kapitel wird auf die fachliche Konzeption der Anwendung, die Konzeption des KNN sowie auf die Beschreibung und Wahl des geeignetsten Frameworks zur Programmierung des KNN eingegangen.

2.1 Fachliche Konzeption der Anwendung

2.1.1 Grundidee

2.1.2 Mockup

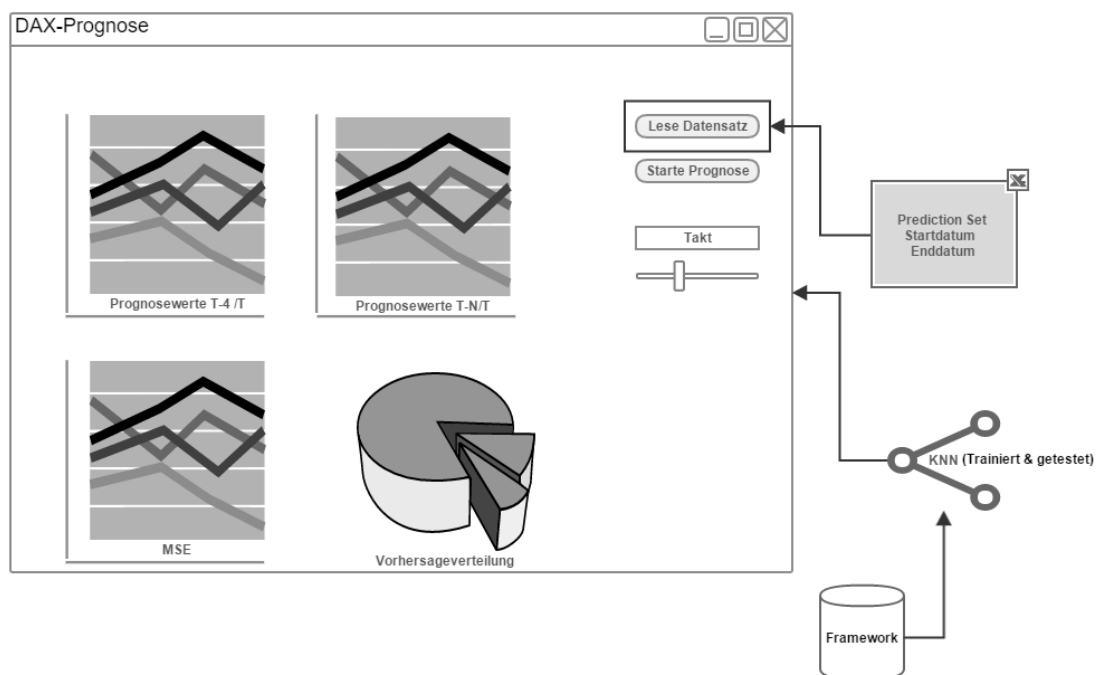


Abbildung 2.1: Mockup der Anwendung

2.2 Konzeption des künstlichen neuronalen Netzes

In den folgenden drei Abschnitten wird auf die Konzeption des KNN eingegangen. Zunächst wird der Typ des KNN festgelegt. Anschließend wird die zugrundeliegende Topologie des KNN und das anzuwendende Lernverfahren bestimmt.

2.2.1 Wahl des Netztyps

Grundsätzlich 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 $m \in \mathbb{N} | m \leq n$ 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 mehrere Modelle aufteilen. Die 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 aufgrund mehrerer vorhergehender DAX-Kurse prognostiziert wird, wird ein Netzmodell aus der Klasse der Hetero-Assoziativen Netze benötigt. Für die Anwendung ist demnach nur die linke Spalte der Tabelle relevant.

Für den nächsten Auswahlsschritt können die folgenden zwei Theoreme betrachtet werden:

Theorem der linearen Separierbarkeit. *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.*

Da es sich beim Börsenkurs auf Grund des obigen Theorems um eine nicht linear separierbare Funktion handelt, das Perzeptron und die Adaline aber nur linear separierbare Funktionen approximieren können, fallen diese Möglichkeiten weg. Nicht jedoch das mehrschichtige vorwärtsgerichtete Netz. Dass dieses KNN auch tatsächlich dafür geeignet ist, belegt das folgende Theorem:

Theorem von Kolmogorov. *Für $n \in \mathbb{N} | n > 2$ lässt sich jede reellwertige Funktion $f : [0; 1]^n \rightarrow [0; 1]$ durch ein dreischichtiges vorwärtsverknüpftes Netz mit maximal n Einheiten in der Eingabeschicht, $(2n + 1)$ Einheiten in der Zwischenschicht und $2n + 1$ Einheiten in der Ausgabeschicht berechnen.*

Ein Börsenkurs kann prinzipiell jede beliebige Funktion annehmen. Durch das obige Theorem ist jedoch sichergestellt, dass das mehrschichtige vorwärtsgerichtete Netz in der Lage ist, diese Funktionen zu approximieren.

2.2.2 Wahl der Grundlegenden Topologie

Zur Prognose werden die vier vergangenen Daten zur Prognose des zukünftigen Wertes als Input Layer genutzt. Da das Ergebnis ein skalarer Wert ist, wird nur ein Neuron als Output Layer benötigt. Die optimale Größe des Hidden Layers wird in Punkt (Kapitelverweis) ermittelt. Somit sieht die Topologie des KNN bis jetzt wie folgt aus:

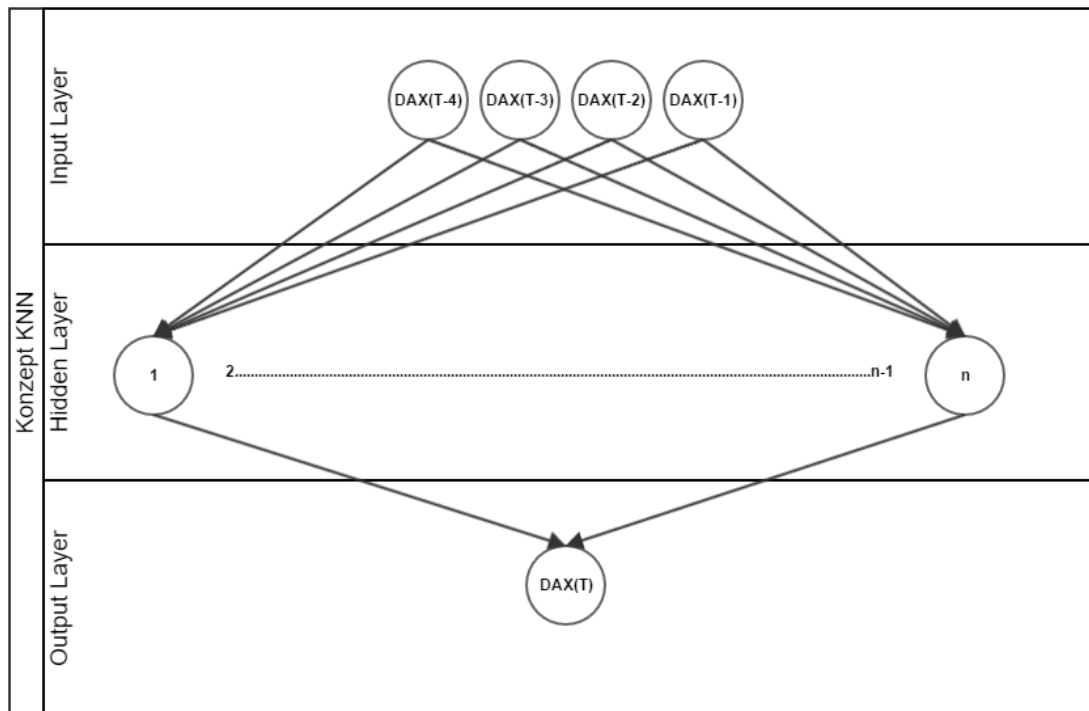


Abbildung 2.2: Grundlegendes Konzept des KNN

2.2.3 Wahl des Lernverfahrens

2.3 Beschreibung von KNN-Frameworks

2.3.1 SNNS

2.3.2 JavaNNS

2.3.3 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 konkreten Topologie

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

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

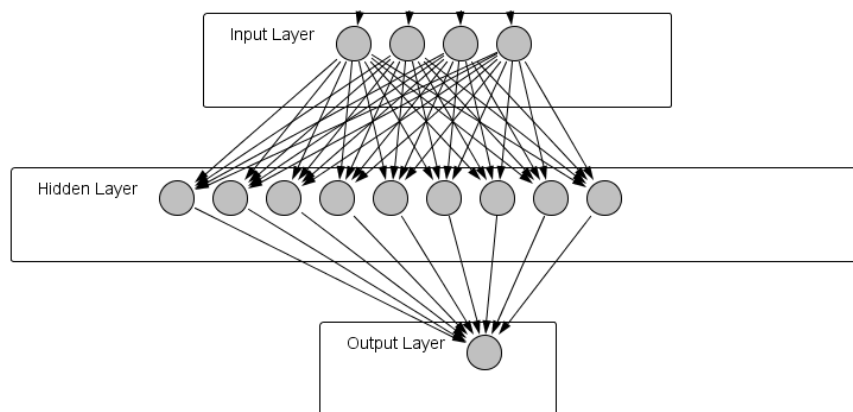


Abbildung 3.2: KNN nach 4-09-1 Muster

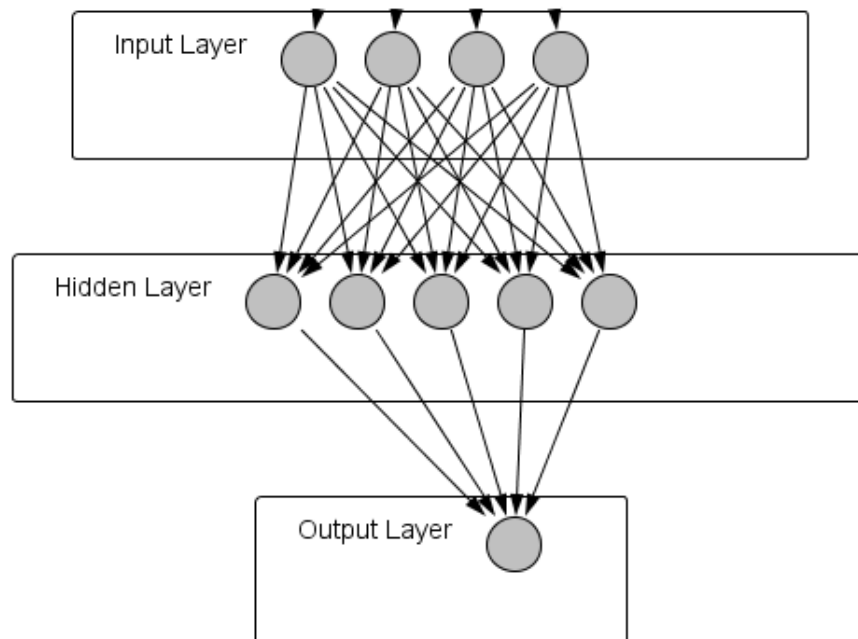


Abbildung 3.1: KNN nach 4-05-1 Muster

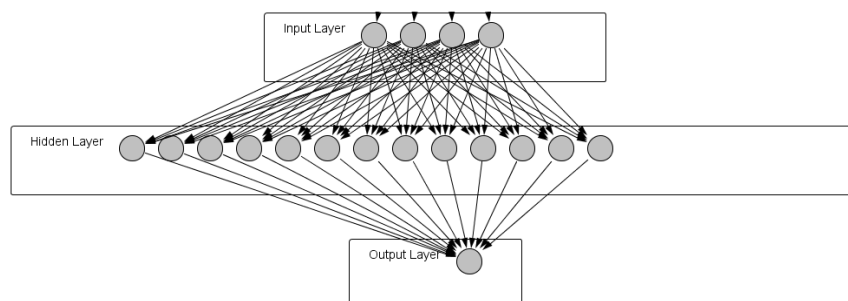


Abbildung 3.3: KNN nach 4-13-1 Muster

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

Tabelle 3.1: Jeweilige Topologien & 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 & korrespondierende MSE

3.1.2 Ermittlung der Transferfunktion

Sigmoide Funktion:

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

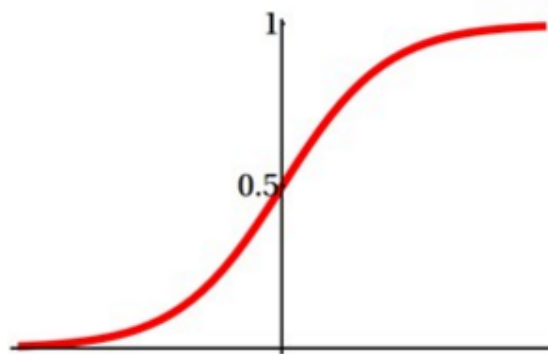


Abbildung 3.4: Sigmoide Funktion

Tangens Hyperbolicus:

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

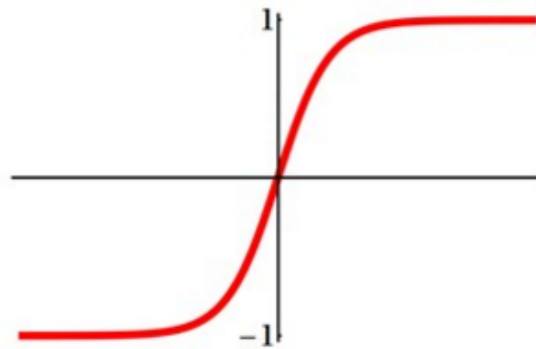


Abbildung 3.5: Tangens Hyperbolicus Funktion

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 & 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.