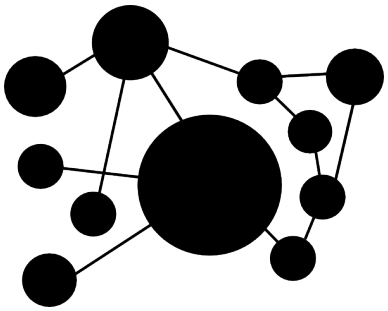
Seminararbeit

Prognose 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. Dezember 2015

TECHNISCHE HOCHSCHULE NÜRNBERG
GEORG SIMON OHM

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Formelverzeichnis

Abkürzungsverzeichnis

KNN Künstliches neuronales Netz 15
DAX Deutscher Aktienindex

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 mehrere Hilfsmittel zurückgegriffen. Eins dieser Hilfsmittel können künstliche neuronale Netze (KNN) sein.

Bei KNN handelt es sich um Netzwerke mit künstlichen Neuronen als Knoten, die mittels gerichteter Verbindungen Eingaben einlesen, weiterverarbeiten und die daraus resultierenden Ergebnisse an weitere Neuronen weiterleiten oder als Ergebnis ausgeben. Bei der Terminologie von KNN wird bewusst auf Begriffe der Biologie zurückgegriffen, da KNN 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 verschiedener 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 KNN ist jedoch in der Lage, diese Zusammenhänge zu finden und diese objektiv und vorurteilsfrei zu bewerten. Somit sind diese 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 gefor-

derte Genauigkeit jedoch wird, desto diskutabler wird ein Einsatz von KNN. Eine typische Grauzone ist hier wieder 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, das die Prognosequalität von KNN über die Jahre stets angestiegen ist.

1.2 Ziel der Arbeit

In dieser Seminararbeit sollen KNN erschaffen werden, die in der Lage sind, Börsenkurse zu prognostizieren. Konkret sollen jeweils ein KNN zur Prognose des Kurses vom DAX, vom Nikkei sowie vom Dow Jones konzeptioniert und umgesetzt werden. Diese KNN sollen anschließend in einer Webanwendung überführt werden. Diese soll die Prognosefähigkeit der KNN visualisieren und Vergleiche zwischen einzelnen Prognosen ermöglichen. In dieser Seminararbeit liegt der Fokus auf das Erlangen eines Grundverständnisses über KNN, und nicht auf das komplette Ausreizen der Prognosefähigkeit von KNN. Trotzdem spielt die Prognosequalität der erstellten KNN eine wichtige Rolle in dieser Seminararbeit.

2 Konzeption

2.1 Konzeption der Anwendung

2.1.1 Grundidee

Bekannte Zusammenhänge zwischen prognostiziertem und tatsächlichem Wert sind meist durch einfache Rechenoperationen umsetzbar. So kann beispielsweise eine Erhöhung der Mehrwertsteuer für ein Produktpreis P mit einer Rechenoperation P = P * M errechnet werden. Für diese Zusammenhänge ist es meist nicht notwendig und sinnvoll, komplexe und vernetzte Softwaresysteme zu entwickeln.

Wie anfangs erwähnt wird bei Börsenprognosen versucht, *nicht-lineare Zusammenhänge* innerhalb von Daten zu finden, da lineare und bekannte Zusammenhänge, die zudem noch exakt sind quasi nicht vorhanden sind. Deshalb ist es für die Anwendung sinnvoll und notwendig einen komplexeren und vernetzeren Ansatz zu verfolgen, auch wenn dies mit einer Steigerung der Kosten einher geht.

Neben dem Hauptziel, die Zeitreihenextrapolation von Börsendaten, soll gezeigt werden, wie es konzeptionell und technisch möglich ist, eine derartige Softcomputing-Lösung zu realisieren und in einen Unternehmenskontext zu integrieren.

Weitere Faktoren, die bei der Entscheidung über die eingesetzten Frameworks und Werkzeuge mit eingeflossen sind, sind Skalierbarkeit, Wartbarkeit, Erreichbarkeit sowie Anschaulichkeit.

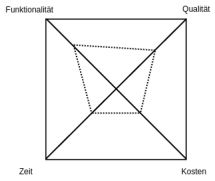


Abbildung 2.1: Abb - nicht lineare

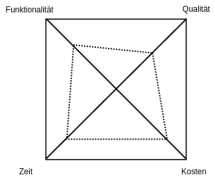


Abbildung 2.2: Abb - lineare

2.1.2 Mockup

Quandl Data-Provider Für Börenprognosen werden quantitative Maßeinheiten benötigt, die im Zusammenhang mit Kurs stehen. Die Wohl aussagekräftigste sind Börsenschluss-Index-Werte von vergangenen Handelsperioden. Je mehr dieser Daten vorhanden, je realistischer diese Daten sind, desto warscheinlicher ist es, einen Schlusswert in der Zukunft vorherzusagen. Aus diesem Grund spricht die Anwendung, genauer der Rest-Controller der Stockmarket-Webapp den Rest-Service von Quandl an. Quandl versteht sich als Datenhändler / Datenanbieter mit integrierter Suchfunktion. Das Angebot umfasst die globale Finanzbranche, so z.B. sämtliche nationale und internationale Börsenkurse der vergangenen Dekaden. Die Rest-API bietet die Export-Formate *XML*, *JSON* und *CSV* an. ; Diese ermöglicht der Anwendung mit echten Börsendaten zu arbeiten.

[https://www.quandl.com/blog/getting-started-with-the-quandl-api].
;

Stockmarket-Webapp Bei der Stockmarket-Webapp handelt es sich um einen Restful-Service der als Schnittstelle zwischen der Quandl-Rest-API, der Visualisierung sowie dem Neuroph Werkzeug fungiert. Wie in Abbildung ABBILDUNG zu sehen ist gibt es keine Kommunikation zwischen Fremd-Services, wenn diese nicht über den Controller der Stockmarket-Webapp statt findet. Hierüber können gleich mehrere Vorteile realisiert werden. 1) Single-Point-Of-Failure: Wenn der Datenlieferant Quandl insolvent geht oder nicht ganz so drastisch, wenn die Anfrage-Mechanismen der Quandl-Rest-API geändert werden. Ein weiteres Szenario, könnte aus Sicht der Visualisierung einen aus Performance-Bedingten Wechsel der JavaScript-Bibliothek darstellen. Sofern für diese Szenarien keine SPOF-Lösung existiert, wird eine entsprechende Anpassung womöglich ein umfassendes Refactoring aller

Anwendungskoponenten erfordern. 2) Skalierbarkeit: Den Fall angenommen, weitere Datensätze sollen an andere Visualisierungen bzw. die Visualisierung um Diagramme erweitert werden, so ist bereits mit einer einseitigen Anpassung der Visualisierung die Anforderung erfüllt. Die durch die Rest-Spezifikation definierte Tatsache der einheitlichen Repräsentation von Anfragen spielt einem hier in die Hände.
Die Aufgabe der Stockmarket-Webapp ist es eine schnittstellen - und benutzerdefinierte Datenaufbereitung zu realisieren.

Visualisierung Die Grundidee die Visualisierung von der Anwendungslogik weitgehend zu trennen hat diverse Vorteile. Die Abhängigkeitsreduktion von anderen Komponenten, sowie Sicherheitsaspekte spielen dabei eine besondere Rolle. Die Visualisierung ist eine rein Client-seitig laufende Anwendung. Diese enthält alle notwendigen Logikroutinen, um Daten anzufragen und entsprechend zu verarbeiten. Die Implementierung dieser Logik wird mit der JavaScript-Bibliothek jQuery umgesetzt. Diese speißt und manipuliert das Document-Object-Model(DOM) je nach *Datenlage*. Das DOM kann auch als HTML-Gerüst bezeichnet werden. Für die visuellen Effekte, die eine bessere Veranschaulichung der Graphen ermöglichen soll, wird CSS eingesetzt. Um das Layout für Geräte mit unterschiedlicher Displaygröße gleichermaßen nutzbar zu machen, wird die CSS-Bibliothek Bootstrap verwendet.

2.2 Technologie

Warum Java, jQuery, Spring Boot:

2.2.1 Frameworks

Zu den nachfolgenden Frameworks und Bibliotheken werden hauptsächlich anwendungsrelevante Funktionalitäten erläutert, da es sonst den Rahmen dieser Seminararbeit sprengen würde.

Spring Boot

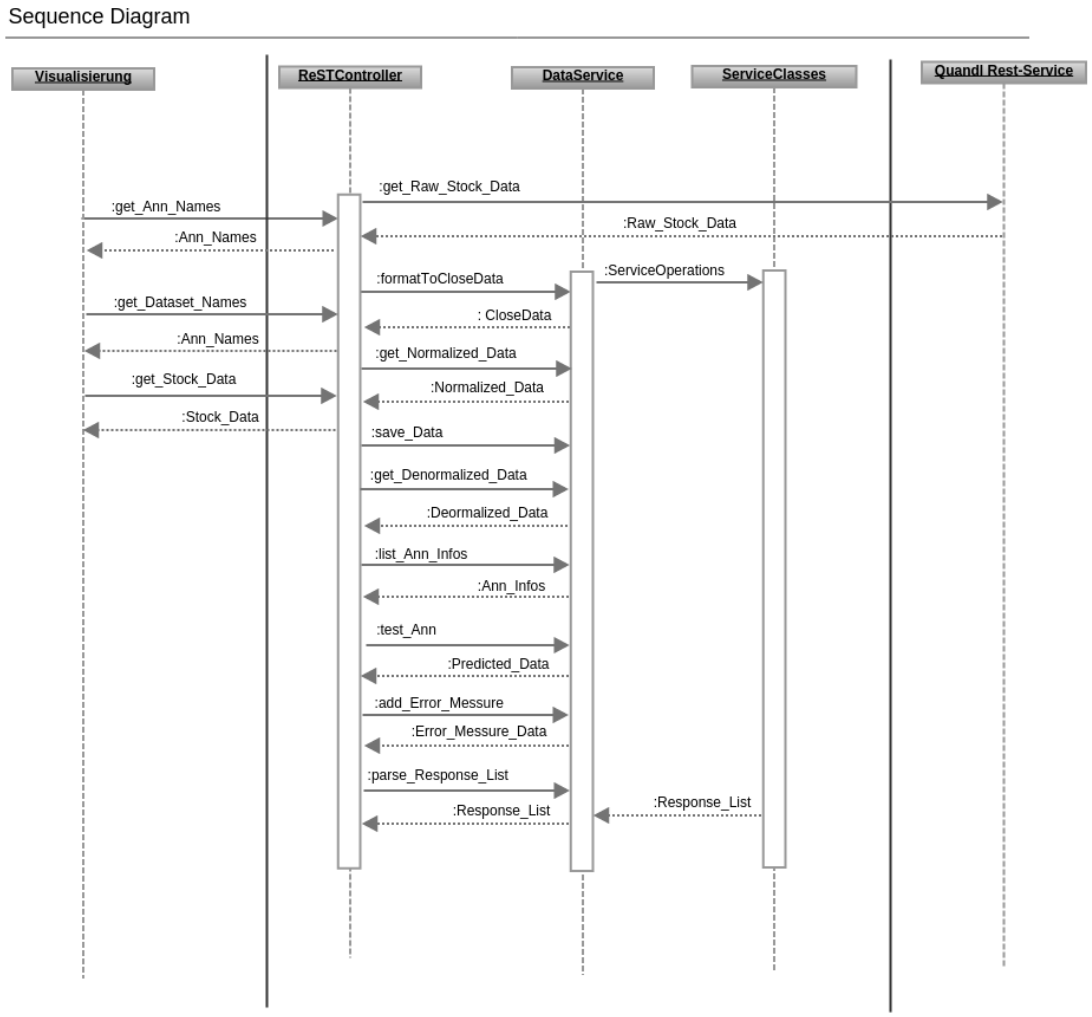


Abbildung 2.3: Sequenzdiagramm

Dependency Injection

Annoationen

Annotationen am Bespiel von Rest-Kommunikation

Dynamik mit c3js Framework

Layouts und CSS Bootstrap

2.2.2 Das Neuroph Framework

2.2.3 Neuroph API

2.3 Apache Maven

Die benötigten Frameworks können auf verschiedene Wege eingebunden werden. Die Softwarepakete können per Hand heruntergeladen und in den entsprechenden Verzeichnispfad kopiert werden, was allerdings bei größeren Projekten hinsichtlicher der Übersicht und Verwaltbarkeit problematisch ist. Ein besserer Ansatz ist es ein Build-Tool wie Apache Maven zu verwenden, was auch für diese Anwendung eingesetzt wurde. Alternative Build-Tools sind z.B. Apache Ant und Gradle.
Zentraler Beschreibungspunkt, der das Projekt benötigter Abhängigkeiten und Build-Prozesse ist die POM.xml. POM steht dabei für *Projekt Object Model*. Die Softwarepakte heißten im Maven-Jargon *Dependencies*, also Abhängigkeiten. Dependencies werden in *Repositories* verwaltet. Es gibt zwei Arten, *local* und *remote* Repositories. Die lokale Datenhaltungskomponente stellt eine exakte Kopie aller heruntergeladenen Dependencies sowie deren Verzeichnisstruktur, von rechnerfernen Repositories dar.

2.4 Konzeption des künstlichen neuronalen Netzes

In den folgenden Abschnitten wird ein KNN konzeptioniert, dass als Vorlage für alle zu erstellenden KNN zur Prognose von Börsenkursen dienen soll.

2.4.1 Wahl des Netztyps

Zunächst ist zu ermitteln, welche Netztypen sich zur Prognose von Börsenkursen grundsätzlich eignen. Nicht jeder Netztyp ist gleichermaßen zur Prognose geeignet. Bestimmte KNN sind beispielsweise überhaupt nicht in der Lage, Prognosen zu erstellen.

Grundsätzlich lassen sich KNN in zwei Oberklassen unterteilen. Es gibt hetero-assoziative Netze sowie die auto-assoziative 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 mehrere Netztypen aufteilen. Die folgende Tabelle liefert hierzu eine Übersicht:

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

Das KNN soll mithilfe von mehreren vorhergehenden Börsenkursen den zukünftigen Börsenkurs prognostizieren. Da es sich bei den zu prognostizierenden Börsenkurs um einen skalaren Wert handelt, ist die Anzahl der Eingabeneuronen (und damit die Anzahl der Elemente des Eingabevektors) höher als die Anzahl der Ausgabeneuronen (und damit höher als die Anzahl der Elemente des Ausgabevektors). Somit sind für diese Seminararbeit nur hetero-assoziative Netze von Relevanz.

Aus der Menge der hetero-assoziativen Netze ist nun der Netztyp zu ermitteln, der für die Anwendung am geeignetsten ist.

Zur Wahl eines geeigneten Netztyps kann zunächst die Lineare Separierbarkeit betrachtet werden:

Definition 1. *Definition 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 „linear separierbar“, wenn es $n + 1$ Werte $w_1, w_2, .., w_n, k$, 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.

Um das Verständnis der oben genannten Definition zu erleichtern, kann die folgende Abbildung betrachtet werden:

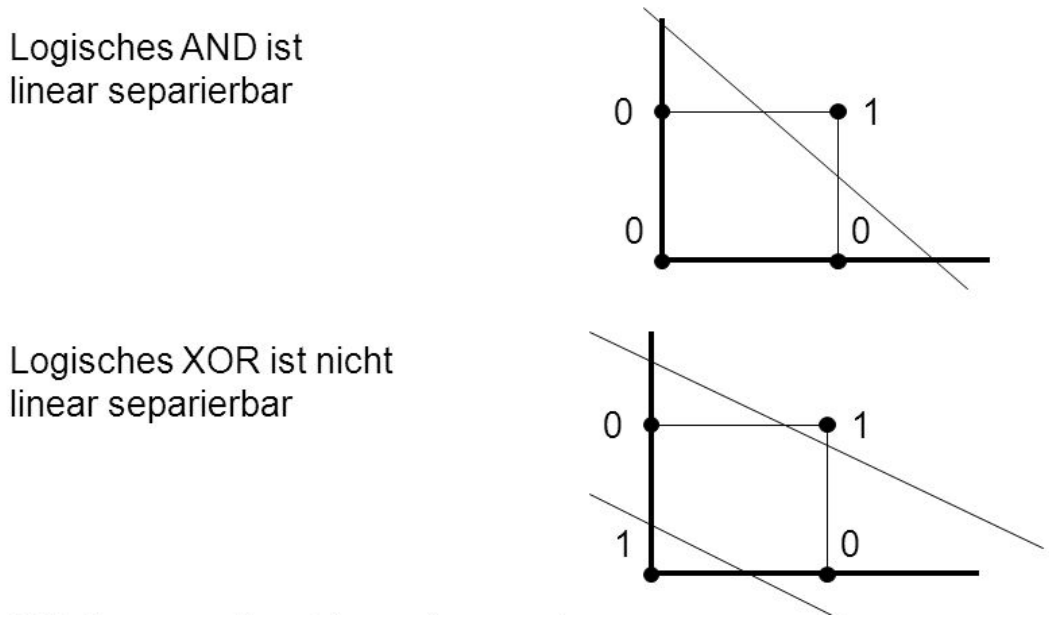


Abbildung 2.4: Bildliche Erläuterung der linearen Separierbarkeit

Man erkennt also, das eine zweidimensionale Funktion dann als linear separierbar gilt, wenn zwischen zwei Ergebnismengen der Funktion eine Gerade gelegt werden kann. Analog setzt sich dies in Funktionen höherer Dimensionen fort. Ist die Funktion zum Beispiel dreidimensional, erfolgt die Separierung durch eine Ebene.

Es ist bewiesen, dass einschichtige KNN nur in der Lage sind, linear separierbare Funktionen zu berechnen. Den Konkreten Beweis dazu liefern Minski & Papert am Beispiel des XOR-Problems:

Beweis 1. *Beweis der Eingeschränkten Fähigkeit von KNN anhand des XOR-Problems*

Gegeben sind:

Ein Perzetron der folgenden Bauart:

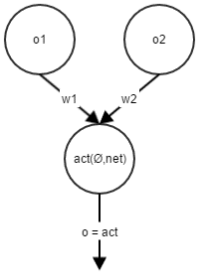


Abbildung 2.5: Beispielperzeptron zur Darstellung des XOR-Problems

und folgende Rahmenbedingungen:

$w_1 \cdot 1 + w_2 \cdot 2 = net$
$f_(act)(o) = id$
$\emptyset = Schwellenwert$

Dann gilt folgendes:
$a) w_1 \cdot 0 + w_2 \cdot 0 \leq \emptyset$ *Bei einem Inputvektor (0,0) liefert der Output 0.*
$b) w_1 \cdot 0 + w_2 \cdot 0 \geq \emptyset$ *Bei einem Inputvektor (0,1) liefert der Output 1.*
$c) w_1 \cdot 0 + w_2 \cdot 0 \geq \emptyset$ *Bei einem Inputvektor (1,0) liefert der Output 1.*
$d) w_1 \cdot 0 + w_2 \cdot 0 \leq \emptyset$ *Bei einem Inputvektor (1,1) liefert der Output 0.*

Der Widerspruch ergibt sich wie folgt:
$(b + c) : w_1 + w_2 \geq \emptyset \wedge (d) w_1 + w_2 \leq \emptyset$

Dieser Beweis kann ebenfalls auf andere nicht linear separierbare Funktionen angewandt werden.Somit steht fest, dass ein einschichtiges Perzeptron nicht in der Lage sein kann, nicht linear separierbare Funktionen zu approximieren.

Auf Basis der oben genannten Tatsache kann ermittelt werden, ob ein einschichtiges Perzeptron zur Approximation von Börsenkursen geeignet ist. Dafür wurde ein ein Perzeptron folgender Bauart entwickelt und untersucht, ob dieses Konvergiert.

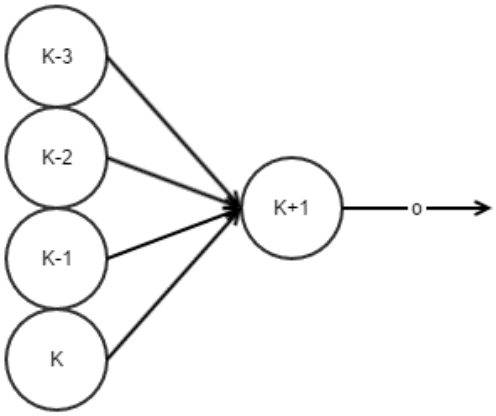


Abbildung 2.6: Grundlegendes Konzept des KNN

Bei Betrachtung des Netzwerkfehlers des Perzeptrons erkennt man, dass das Perzeptron nicht konvergiert:

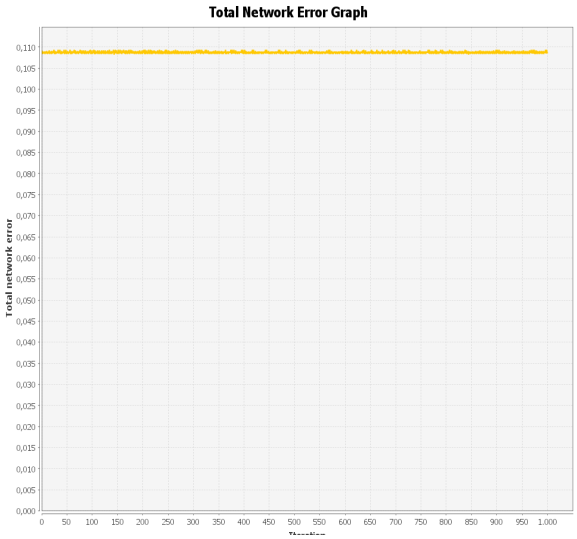


Abbildung 2.7: Grundlegendes Konzept des KNN

Der Netzwerkfehler des Perzeptrons bleibt über alle Iterationen konstant auf einen Niveau von circa $0,10$.

Nun ist es sinnvoll, folgendes Theorem zu berücksichtigen:

Theorem 1. *Konvergentheorem von Rosenblatt*
Der Lernalgorithmus des Perzeptrons konvergiert in endlicher Zeit, d.h. das Perzeptron kann in endlicher Zeit alles lernen, was es repräsentieren kann.

Betrachtet man alle oben genannten mathematischen Gegebenheiten, ergibt sich die folgende Relation:

Perzeptron konvergiert $\rightarrow$ Funktion Linear separierbar $\rightarrow$ Perzeptron geeignet. und natürlich analog:Perzeptron konvergiert nicht $\rightarrow$ Funktion nicht Linear separierbar $\rightarrow$ Perzeptron nicht geeignet.

Folglich bleibt nur noch das Multilayerperzeptron als Mögliche Auswahl übrig. Das dieses Künstliches neuronales Netz (KNN) tatsächlich zur Prognose geeignet ist, belegt das folgende Theorem:

Theorem 2. *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, da ein Multilayerperzeptron als universeller Aprroximator fungiert.

2.4.2 Wahl der Topologie

Zur Prognose des Börsenkurses sollen die letzten vier Börsenkurse als Input dienen. Durch diesen Input soll der Börsenkurs am nächsten Tag prognostiziert werden. Zur richtigen Dimensionierung der inneren Schicht können einige Richtlinien berücksichtigt werden:

- Die Anzahl der versteckten Neuronen in der inneren Schicht sollte nicht zu groß gewählt werden, damit das Netz das antrainierte Verhalten nicht “auswending“'lernt und dieses dann nur bereits trainierte Muster anwenden kann und es somit die Generalisierungsfähigkeit verliert. Man spricht in diesem Fall von Overfitting
- Die Anzahl der versteckten Neuronen in der inneren Schicht sollte auch nicht zu klein gewählt werden, da eine gewisse Menge an Neuronen wichtig sind, um sich Regeln merken zu können.
- Eine grobe Annäherung zur Bestimmung der Obergrenze der Anzahl von Neuronen in der versteckten Schicht liefert die folgende Formel:

$$N_h = \frac{N_d}{10 * (N_i + N_o)} \tag{2.1}$$

Nh ist hierbei die Obergranze, Ni ist die Anzahl der Inputneuronen und No die Anzahl der Outputneutonen. Da 450 Trainingsdaten verwendet werden. Bedeutet das für diese SEminararbeit konkret:

$$N_h = \frac{450}{10 * (4 + 1) = 9} \tag{2.2}$$

Somit ergibt sich insgesamt die folgende Topoologie:

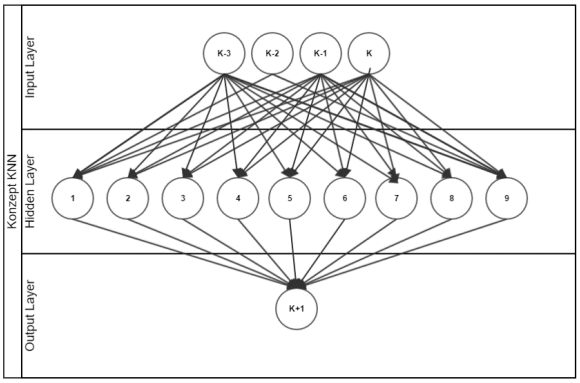


Abbildung 2.8: Grundlegendes Konzept des KNN

Die oben abgebildete Topologie stellt ein solides Grundkonstrukt dar, das in der Umsetzungsphase noch weiter optimiert werden kann.

2.4.3 Wahl des Lernverfahrens

Grundsätzlich gibt es drei grobe Klassifikation von Lernverfahren. in diesem Abschnitt werden alle drei Lernverfahren näher vorgestellt und anschließend eine Begründete Auswahl der Lernverfahrens getroffen.

- Überwachtes Lernen: Beim überwachten Lernen sind sowohl die Eingabedaten sowie die dazugehörigen Ausgabedaten bekannt. Mit Hilfe dieser Daten kann das KNN dann trainiert werden. Die berechneten Ausgabedaten können anschließend mit den tatsächlichen Ausgabedaten verglichen werden. Dieser Fehler wird dann genutzt, um die Verbindungsgewichte des KNN anzupassen. Typische Vertreter dieses Lernverfahrens sind die sogenannten Backpropagation-Lernverfahren.
- Bestärkendes Lernen Ähnlich wie das überwachte Lernen, jedoch biologisch motivierter ist das sogenannte bestärkende Lernen. Hier sind dem KNN die Eingabewerte zwar bekannt, aber die dazugehörigen Ausgabewerte nur zum Teil oder gar nicht. Das KNN wird lediglich darüber informiert, das Ergebnis richtig bzw. falsch war. Es ist ein sehr Zeitaufwändiges Lernverfahren, da es die Gewichte auf Grund der spärlichen Information nur sehr langsam anpassen kann. Dieser Verfahren kann als Mischung aus überwachtes Lernen und unüberwachtes Lernen gesehen werden.
- Unüberwachtes Lernen Das unüberwachte Lernen ist biologisch gesehen am plausibelsten. Bei diesem Lernverfahren besteht existieren nur Eingabemuster, es existieren keine erwünschten Ausgaben oder Angaben, ob das Netz die Eingaben richtig oder falsch klassifiziert hat. Stattdessen versucht der Lernalgorithmus selbständig, Gruppen ähnlicher Eingabevektoren zu identifizieren und diese auf Gruppen ähnlicher oder benachbarter Neuronen abzubilden.

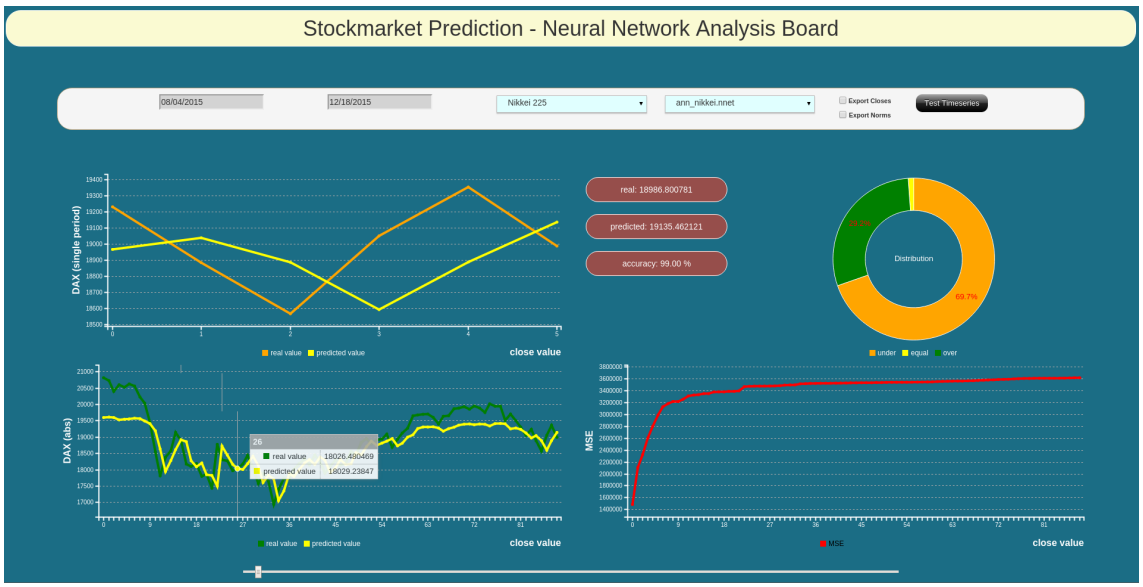
Da sowohl die Eingabewerte als auch die Ausgabewerte der zu verwendenden Datensätze bekannt sind, bietet sich das überwachte Lernen an. Verglichen mit den anderen Lernverfahren ist dies die effizienteste Lernmethode. Sie verfügt zwar über kein biologisches Vorbild, dieser Umstand hat aber für diese Seminararbeit keine Relevanz.

3 Umsetzung

3.1 Umsetzung der Anwendung

3.1.1 Oberfläche

Die Oberfläche ist grob eingeteilt in zwei funktional getrennte Bereiche. Der weiße Kontainer beinhaltet ein Formular das der Anfragestellung dient. Hierin werden die Rahmenbedingung der Vorhersage eingegeben. Zum einen Start - und Enddatum, sowie welche Datenmengen, die von Quandl bezogen werden mit welchem Neuronalen Netz getestet werden sollen. Sofern die davon rechts gelegenen Checkboxen angeklickt abgeschickt werden, so werden CSV-Datein mit den entsprechenden Rahmenbedingung in einem konfigurierbarem Verzeichnis gespeichert. *Export Closes* steht hierbei für denormalisierte Börsenschlusswerte, *Export Norms* für die gleichen normalisierten Werte. Mit *Test Timeseries* werden alle Informationen serialisiert und die Anfrage durchgeführt.
Das Ergebnis der Anfrage wird, sofern sie die Fehlerprüfung übersteht, unterhalb des Formulars angezeigt. Die Ausgewählten Werte bleiben hierbei dem Formular hinterlegt. Die Antwort wird in Form von Diagrammen und Displays angezeigt. Alle Informationen werden bzgl. eines zeitlichen Intervalls aktualisiert, somit wird eine Art Echtzeit-Darstellung realisiert.
Die Javascript Bibliothek *C3js* wird bei der Realisierung von vier Diagrammen eingesetzt. Das erste links oben angeordnete visualisiert diskrete *tatsächliche* Werte (real value) und *vorhergesagte Werte* (predicted value) für einen definierten Zeitraum *single period*. Dies dient vorallem der Übersicht, so wird eine Art *Zoom*-Modus realisiert. Das zweite unten links angeordnete Diagramm visualisiert die gleichen Werte, jedoch über den gesamten Zeitraum *absolute-graph* dar, was einem Resümee bzgl. der Vorhersage zuträglich ist.
Das dritte und letzte Diagramm stellt den Mean-Square-Error-Graph (MSE) der gesamten Zeitreihe dar. Die erste drei Diagramme sind vom gleichen Typ. Eine weitere statistische Auswertung, die oben rechts also c3-donut-Diagramm umgesetzt ist, ist die Klassen-



Zuordnung von vorhergesagten Werten in die Klassen *under* (unterhalb vom tatsächlichen Wert), *equal* und *over*. Da in der Realität die Werte fast nie exakt übereinstimmen, wurde ein Konfidenz-Interval definiert, dass eine gewisse Fehlertoleranz gegenüber der Differenz beider Werte zulässt und equal-Klasse somit faktisch keine leere Menge bildet.
Des weiteren wurden Displays mit eigens geschriebenem CSS-Klassen definiert, die den tatsächliche, den vorhergesagten sowie die *accurancy* (Exaktheit) Anzeigt. Die Accuracy lässt sich als Betrag der Differenz zweier Werte verstehen.
Am unteren Rand ist ein Schieberegler integriert um die Echtzeit-Intervalllänge zu verändern.

Im folgenden wird ein Anfrage-Antwort-Zyklus (Request-Response-Lifecycle) der Umsetzung detailliert beschrieben und erläutert, welche Rolle die bereits vorgestellten Technologien und Überlegungen in die Implementierung eingeflossen sind, sowie wie diese miteinander verknüpft sind.

3.1.2 Dataset und AnnNames ajax

Die Auswahl der Quandl-Datensätze, sowie der .nnet-Datei werden ausschließlich mit Hilfe der Stockmarket-Webapp befüllt. Hierbei werden die Informationen mit AJAX-Aufrufen abgefragt.

3.1.3 Anfrage

3.1.4 Anfragestellung

Das Auslösen einer Anfrage wurde bereits oben geschildert. Der Button-Click wird mit einem jQuery-Event-Listener behandelt. Dieser führt eine Fehlerprüfung der Formulareingabe durch. Wenn ein Enddatum eingegeben wird, dass kleiner dem Startdatum ist, so wird eine entsprechende Fehlermeldung in einem *div*-Tag geladen. Gleiches passiert das, wenn unvollständige Angaben gemacht werden.
Beim Bestehen der Prüfung wird ein AJAX-Aufruf (Asynchronous JavaScript and XML) für den ReSTController der Stockmarket-Webapp ausgeführt. Hierbei wird eine URI konstruiert, die zum einen die Basis-URL des Webservice anspricht, zum anderen die Formulareingabedaten als Parameter enthält. Wenn der Aufruf erfolgreich durchgeführt wird, werden die Hilfsvariablen im Javascript der Visualisierung erneut initialisiert, damit es bei mehreren hintereinander folgenden Ausführungen ohne Seitenaktualisierung keine veränderte Verarbeitungsroutine der Antwortdaten gibt.

3.1.5 Antwort-Erstellung

Die Antwort-Erstellung obliegt der Stockmarket-Webapp, die Visualisierung liefert die Eingabedaten hierzu. Der ReSTController ist das Verbindungsglied im Anfrage-Antwort-Lebenszyklus, aber auch die Komponente, die die Programmablauflogik der Antworterstellung implementiert. Dieser enthält drei Rest-Endpunkte, entsprechend der drei Anfragen, die die Visualisierung per AJAX realisiert.

Listing 3.1: ReSTController Snippet

```
@RestController
```

```
@RequestMapping(value = "/stock")
public class ReSTController implements ServletContextAware, ServletConfigAware {

    @Autowired
    private EnvService envService;

    @Autowired
    private DataService dataService;

    \vdots

    @RequestMapping(value= "/data", method = RequestMethod.GET)
    public String data(@RequestParam String startDate,
                       @RequestParam String endDate,
                       @RequestParam String collapse,
                       @RequestParam String stock,
                       @RequestParam String ann,
                       @RequestParam Boolean saveNorm,
                       @RequestParam Boolean saveClose )
                       {
        // Programmablauflogik
        \vdots
        }

   @RequestMapping(value= "/ann_names", method = RequestMethod.GET)
   public List<String> annNames()
   {
   \vdots
   }

   @RequestMapping(value= "/datasets", method = RequestMethod.GET)
   public String datasets()
   {
   \vdots
   }
}
```

Die ReSTController-Klasse ist mit *@RestController* annotiert, was eine Implementierung der *ServletContextAware* und *ServletConfigAware* erfordert. Diese stellen sicher, dass der ReSTController auf den Servlet-Kontext zurückgreifen kann sowie die Servlet-Konfiguration beachtet wird. Diese wird unter anderem dazu verwendet, die Basis-URL zu spezifizieren. Die Rest-Endpunkte werden relativ zu dieser Basis-URL abgebildet. Die Annotation *@RequestMapping* erlaubt die Kennzeichnung von Klassen und Methoden als Zielpunkte gleichermaßen. Entsprechend der HTTP-Verbs können Anfrage-Methoden (RequestMethod) spezifiziert werden. Für den ReSTController wird ein Basis-Endpunkt */stock* definiert.

Quandl-API URL-Parameter	Bedeutung
• order	Reihenfolge der Ergebnismenge
• exclude_column_names	Schließt Header-Information aus
• start_date	Startdatum
• end_date	Enddatum
• collapse	Datenfequenz

Tabelle 3.1: verwendete URL-Parameter - Quandl API

Das *data*-Methode implementiert die Programmlogik für die Antworterstellung, die annNames-Methode liest die Bezeichnungen der .nnet-Dateien, die *datasets*-Methode die Rest-Endpunkte der Quandl-Datasets zu den entsprechenden Börsenkursen aus der *app.properties* Datei.
Die URI des AJAX-Aufruf des Eingabeformulars wird in der Methodensignatur auf Vollständigkeit und Datentyp-Korrektheit mit der *@RequestParam*-Annotation überprüft. *@RequestParam* fordert einen URL-Parameter gleichnamig zum Variablennamen zu übergeben, alternativ könnte man eine *@Param*-Annoation verwenden, die eine unterschiedliche Namensgebung erlaubt, und sich an der Reihenfolge der Parameter orientiert.
Die Antworterstellung ist auf sechs Serviceklassen aufgeteilt, die wiederum gekapselt in einer Serviceklasse sind. Der ReSTController bindet ausschließlich den diese Klasse, den *Data-Service* ein.
Erster Schritt ist das Einlesen einiger Konfigurationseigenschaften aus der *app.properties*-Datei. Unter anderem das gewünschte Rückgabeformat des Quandl-Datasets sowie die Basis-URL der Quandl-API. Anschließend wird wird aus den zur Vefügung stehenden Informationen eine URI konstruiert und eine Restabfrage der Quandl-Api durchgeführt. Das Ergebnis wird als Liste von Zeichenketten gespeichert. Dabei wird die Daten mit folgenden Spezifikationen angefragt:

Der zweite von zehn Methodenaufrufen, die der *data-Service* organisiert ist das Formatieren der Ergebnismenge in eine für die Neuroph-API passende Struktur. Diese Aufgabe übernimmt der FormService.

Die Konversion beachtet zwei Kriterien, zum einen die Anzahl der Input-und Output-Neuronen, die in der Konfigurationsdatei unter der Eigenschaft *format.period.length* also Summe angegeben ist, sowie die Spaltenposition desjenigen Wertes, der vorhergesagt werden soll, also der Börsenschlusswert. Es werden ausschließlich Börsenschlusswerte und deren Datum berücksichtigt. Eine Zeile in der konvertierten Liste besteht aus dem Datum des letzten (spätesten) betrachteten Wert innerhalb einer Periode, sowie die Werte der gesamten Pe-

riode. Das Ergebnis wird als Listen von Listen von Zeichenketten gespeichert, was einen bessere Verarbeitung für die folgenden Schritte ermöglicht.

Als zweiter Schritt steht die Normalisierung an. Diese Aufgabe wird im *NormDenorm-Service* durchgeführt. Wie in Kapitel 3.3 beschrieben gibt es unterschiedliche Ansätze für die Normalisierungsfunktion, die von der Wahl der Aktivierungsfunktion abhängt. Somit ist es sinnvoll die Formeln für die Normalisierung und Denomalisierung auszulagern um keinen Flaschenhals für künftige Anpassungen oder Experimente zu bilden. Die *normalize*-Methode ließt zuerst die entsprechende Formel, die maximale Anzahl gewünschter Nachkommstellen sowie die Periodenlänge aus der *app.properties* Datei. Die Erfüllbarkeit der Funktion in Kapitel 3.3 hängt von der Möglichkeit ab, das globale Maximum und Minimum zu bestimmen. Damit wird gewährleistet, dass aussschießlich Werte zwischen Null und Eins angenommen werden können. Da unsere Datenmenge eine feste Größe besitzt ist dies kein Problem. Eine überaus nützliches Softwarepaket ist der *ScriptEngineManager*, diese die Art der Modularität erlaubt. Dieser kann einen als Javascript initialisierten *ScriptEngine* verwenden um die eingelesene Funktion (Zeichenkette) entsprechen zu interpretieren. Die private Methode *processFormula* wird für Normalisierung und Denormalisierung verwendet. Diese konvertiert schließlich jeden Wert der Liste über die übergebene Funktion. Anschließend wird die Liste eine Variable im ReSTController zurückgegeben.

Die nächsten zwei Schritte setzt die Anfrage der Checkboxen in die Tat um und speichern, sofern im Formular angeklickt, transformierte Börsensschlussdaten sowie normalisierte Daten als CSV-Dateien in ein Verzeichnis, dessen Pfad in der *app.properties*-Datei festgelegt ist. Der Dateinamen ist autogeneriert und besteht aus dem angefragten Zeitraum sowie dem Namen des Datasets. Notwendig war diese Funktionalität vor allem für das Training der Neuronalen Netze, das im Neuroph Studio durchgeführt wurde, also außerhalb der Anwendung.

Im fünften Schritt wird das Neuronale Netz mit den normalisierten Daten getestet. Der *Ann-Service* implementiert hierzu eine Methode *testAnn*. Um die Erweiterbarkeit auch für

künftige Umsetzungen von Neuronalen Netzen möglichst einfach zu gewährleisten, so z.B. eine andere Topologie zu verwenden, die auf mehr Eingangsneuronen setzt, wird die Periodenlänge aus der *app.properties*-Datei gelesen. Der Testalgorithmus wird allgemeingültig gehalten, und funktioniert für alle Neuronalen Netztypen, die ein Ausgangsneuron besitzen. An dieser Stelle wird das erste mal die Neuroph-Core-Bibliothek innerhalb der Stockmarket-Webapp verwendet. Zuerst wird eine *Dataset*-Instanz mit der Anzahl der Input- und Outputneuronen initialisiert. Anschließend wird die normalisierte Liste zeilenweise iteriert. Jede Zeile wird in zwei Arrays von *double*-Werten geteilt, ein Input- und Output-Array und anschließend dem DataSet-Objekt hinzugefügt. Als nächstes wird das Multi-Layer-Perzeptron über den hinterlegten Dateipfad geladen. Hierzu bietet Neuroph die *createFromFile*-Methode. Anschließend wird das DataSet-Objekt in eine Liste von *DataSetRow*-Instanzen gesplittet. Eine DataSetRow ist das wichtigste Attribut der DataSet-Klasse und repräsentiert einen Testschritt. Für jedes Input-Array in einer solche DataSetRow wird ein Ausgabewert errechnet, welcher zu einer Liste aus Zeichenketten hinzugefügt wird. Nachdem die Iteration abgeschlossen ist und die Menge der vorhergesagten Werte vollständig kalkuliert wurde, wird die Liste mit der Testdatenliste erweitert. Diese Aufgabe übernimmt der entsprechend *Form-Service*.

Der sechste Schritt ist die Denormalisierung der gewonnen Ergebnisliste. Wie oben bereits erwähnt wird die *processFormula*-Methode des *NormDenorm-Serive* lediglich mit anderen Parameterwerten ausgeführt. Es wird die Denormalisierungsfunktion, die maximale Anzahl der Nachkommstellen sowie die Periodenlänge aus der *app.properties*-Datei geladen und übergeben. Eine Funktionalität die derzeit nicht benötigt wird, aber dennoch implementiert ist, stellt die Möglichkeit dar Spalten der List von den Operationen auszuschließen. Wären zu diesem Zeitpunkt bereits denormalisierte Werte vorhanden, wäre dies deshalb kein Problem.

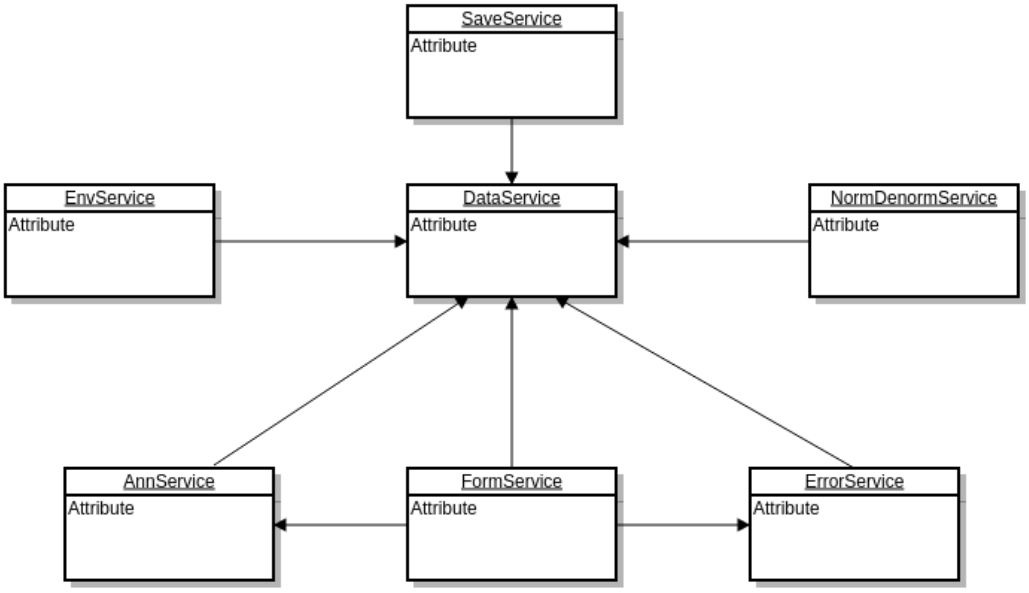
Der siebte und letzte Schritt, der eine Modifikation der Liste vollführt ist das hinzufügen von Fehlermaßen. Hierfür ist der *Error-Service* zuständig. Grundidee des Error-Service ist, eine Methode pro Fehlermaß umzusetzen, wobei jede Methode eine Liste von Zeichenketten zurückgibt. Der *Data-Service* greift ausschließlich auf eine im zugedacht Methode zu, die für alle diese Fehlermethoden zuständig ist. Hier ist das die *getAllErrorMessures*-Methode. Als erstes wird geprüft ob ein Fehlermaß überhaupt hinzugefügt werden soll. Diese Informationen sind in der *app.properties*-Datei gelistet. Für jedes Fehlermaß gibt es einen

Eintrag mit dem Präfix *error.avail*, der als *Boolean*-Wert interpretiert wird. Je nach dem ob diese wahr sind, werden die entsprechenden Fehlermaße berechnet. Die Ergebnislisten dieser Methoden werden zu einer großen Ergebnisliste hinzugefügt und anschließend mit der *addCols*-Methode des *FormService* hinzugefügt. Für jedes weitere Fehlermaß muss also lediglich ein Eintrag in der app.properties sowie eine Methoden Implementierung umgesetzt werden, welche in der Hauptmethode entsprechend ausgeführt wird. Somit können die Ziele von Funktionsumfang, Flexibilität und Performance harmonieren.
Der Mean-Square-Error (MSE) wird bereits in ..... erklärt. Des Weiteren wird hier die *Accuracy* (Exaktheit) sowie die *Distribution* (Klassenzuteilung) berechnet. Das hierfür benötigt Konfidenzintervall bleibt ebenfalls als Eigenschaft in der Konfigurationsdatei hinterlegt.

Abschließender Schritt zur Antworterstellung ist die Konversion der generierten Liste in einen geeignete Struktur, damit die Visualisierung eine möglichst einfache Handhabe bei dem Verarbeiten der Daten hat. Die Aufgabe übernimmt sinngemäß der *FormService*. Die Methode *parseResponseList* gibt ein *JSONArray*-Objekt zurück. Dieses wird iterativ befüllt, indem jede Liste innerhalb der Ergebnisliste in einer fest definierte Struktur innerhalb einer jeweils neuen JSONObject-Instanz abgebildet wird. Zu jedem Wert gibt es einen textuellen Schlüssel, somit kann auf einen Schleifendurchlauf eines JSONObjekt innerhalb des Arrays in der Visualisierung verzichtet werden und der entsprechende Wert unmittelbar angesprochen werden. Der *Data-Service* wandelt das JSONArray-Objekt in einen String. Dieser wird der Visualisierung nun als Antwortnachricht übergeben.

Listing 3.2: Beispiel - JSONArray

```
[
   {
      "date":"2015-12-07",
      "acc":"100.00",
      "input":[
         "11261.240234",
         "11190.019531",
         "10789.240234",
         "10752.099609"
      ],
      "dist":{
         "equal":"0",
         "over":"0",
         "under":"1"
      },
      "ro":"10886.089844",
      "mse":"454.036617",
      "po":"10864.781709"
```




```
    }
]
```

3.1.6 Antwort-Verarbeitung

Die Zeichenkette des JSON-Arrarys wird in einer globalen Variable zwischengespeichert. Da alle Diagramme und Displays einem einheitlichen Takt folgen sollen, wird die Initialisierung der Graphikkomponenten, sowie deren gesamter Aktualisierungszyklen von einer Funktion *setDeceleratingTimeout* aufgerufen. Nach einer Verzögerung, die durch eine globale Intervalllängendefinition beeinflusst wird und über die JavaScript-spezifische Methode *window.setTimeout()* realisiert wird, wird jeweils die *processValues*-Funktion abgearbeitet. In dieser wird jeweils eine JSONObject-Repräsentation verarbeitet und für die Anzeige für das kommende Intervall vorbereitet sowie anschließend an die Darstellungsfunktionen (Rendering-Funktionen) übergeben. Grob gesagt werden entsprechende temporäre Arrays mit den Werten der Repräsentation befüllt und übergeben. Neben den Darstellungen wird der Schieberegler, der den Wert der globalen Variable *Interval* in Echtzeit verändert, in die Visualisierungslandschaft integriert.

3.1.7 Konfigurationsdatei der Stockmarket-Webapp

Listing 3.3: app.properties - Konfigurationsdatei der Stockmarket-Webapp

```
## filepath where ann files are stored
media.source.base=/home/bthofrichter/Schreibtisch
media.source.ann=/source/ann
media.source.analyzed=/source/analyzed/

quandl.api.baseurl=https://www.quandl.com/api/v3/datasets

quandl.api.key=KsHDYzZK6uyyynwQNS7p

## response format by quandl API
quandl.data.dataset.format=csv
quandl.data.dataset.order=asc
quandl.data.dataset.header=true
quandl.data.dataset.close.pos=4

##
format.period.length=5
format.data.included=true
format.data.norm.precision=12
format.data.denorm.precision=6
format.data.error.acc.precision=2

dist.conf=10

## quandl datasets
quandl.dataset.dax=/YAHOO/INDEX_GDAXI
quandl.dataset.nikkei_225=/YAHOO/INDEX_N225
quandl.dataset.djia=/YAHOO/INDEX_DJI


## quandl dataset names for visualisation
quandl.dataset.name.dax=DAX,dax
quandl.dataset.name.nikkei_225=Nikkei 225,nikkei_225
quandl.dataset.name.djia=Dow Jones (DJIA),djia

## X := current value
## MIN := global minimum
## MAX := global maximum
normalize.formula=((X-MIN)/(MAX-MIN))*0.8+0.1
denormalize.formula=((X-0.1)/0.8)*(MAX-MIN)+MIN

## ERROR-AVAILABILITY
error.avail.dist=true
error.avail.mse=true
```

```
error.avail.acc=true
```

3.1.8 Maven POM der Anwendung

Die in ListingPOM zeigt die Grundstruktur der POM.xml der Anwendung. Diese enthält alle wichtigen Informationen um die Anwendung in der Beschriebenen Systemlandschaft zu bauen.
Kurze Erläuterung zum Aufbau der POM.xml:
Project-Knoten: Enthält alle anderen Knoten. Die Deklaration in Zeile 1–4 verweist auf eine entsprechende Namensraum und Schema-Defintion. modelVersion-Knoten: Die Versionsnummer muss mit den Versionsinformationen im Proect-Knoten übereinstimmen. In Zeile 7-9 sind die typischen 3 Knoten *groupId*, *artifactId* und *version* zur Beschreibung einer Dependency. Da diese Knoten nicht mit dem dependency-Knoten eingeschlossen ist, bezieht sich die Dependency-Information auf sich selbst. Sofern die Stockmarket-Webapp also in einem anderen Projekt als Softwarepakte hinzugefügt werden soll, müsste man diese Knoten einfach in den dependency-Knoten aufführen.

In Zeile 11 beschreibt der packaging-Knoten das Zielformat der kompilierten Anwendung. Für eine Spring-Webapp eignet sich ein Webarchiv (war).

In Zeile 13-17 werden genaue Angaben zur Systemlandschaft, genauer zum Tomcat-Server und zum JDK (Java Development Kit) gemacht. Die Stockmarket-Webapp setzt auf Java 8 und einen Tomcat 7.0.64. Diese Properties sind zur Absicherung gedacht. Prinzipiell ist es auch möglich ein Java 7 oder eine andere Tomcat Version einzusetzen. Da aber keine vollständige Prüfung jeglicher Versionen durchgeführt wurde und auch wirtschaftlich unsinnig wäre einigt man sich auf eine spezielle Version, mit der alle Funktionalität der Anwendungen getestet werden.

Der Parent-Knoten in den Zeilen 19-23 kann als *higher-lever*-Dependency gesehen werden. Diese bindet das Spring-Boot-Start-Parent Paket ein.

In den Zeilen 25–31 werden Dependencies nach bereits beschriebenem Format aufgeführt.

Zeile 35-40 listet Remote-Repositories. Sinnvoll ist eine solche Definition vorallem dann, wenn mehrere Abhängigkeiten hieraus benötigt werden.

Der build-Knoten Knoten der in Zeile 44-68 beschrieben ist, ist optional und nimmt auf den Maven-Build-Zyklus einfluss. Für die Stockmarket-Webapp ist es notwenig eine Dependency, die sich nicht per Standard-Deklaration in Kontext einfügen lässt, auf diesem Weg zu integrieren. Hierbei handelt sich das Neuroph-Core-2.9.jar Softwarepaket. Da die Neuronalen Netze mit dem Neuroph-Studio, das mit dieser Version arbeitet, erstellt und trainiert wurde, war es notwendig die gleiche Version in der Anwendung zu verwenden. Da das offizielle Remote-Repository allerdings diese Version noch nicht bereitstellt musste die JAR-Datei während der *install*-Phase des Maven-Builds berücksichtigt wird. In dieser Phase werden die Abhängigkeiten in das lakale Repository kopiert, was für das automatische Deployment im Tomcat-Server führt (bei entsprechender Konfiguration).
Dank Maven kann der Build-Prozess der Stockmarket-Webapp dynamisch, übersichtlich und effizient gestaltet werden.

Listing 3.4: POM.xml Snippet - Stockmarket-Webapp

```
1  <project xmlns="http://maven.apache.org/POM/4.0.0"
2           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3           xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
4                               http://maven.apache.org/maven-v4_0_0.xsd">
5      <modelVersion>4.0.0</modelVersion>
6      <groupId>de.soco.stockmarket</groupId>
7      <artifactId>stockmarket-app</artifactId>
8      <version>1.0</version>
9      <packaging>war</packaging>
10     <properties>
11         <maven.compiler.source>1.8</maven.compiler.source>
12         <maven.compiler.target>1.8</maven.compiler.target>
13         <tomcat.version>7.0.64</tomcat.version>
14     </properties>
15     <parent>
16         <groupId>org.springframework.boot</groupId>
17         <artifactId>spring-boot-starter-parent</artifactId>
18         <version>1.3.0.M5</version>
19     </parent>
20     <dependencies>
21         <dependency>
22             <groupId>org.springframework.boot</groupId>
23             <artifactId>spring-boot-starter-web</artifactId>
24         </dependency>
25           ...
26     </dependencies>
27     <repositories>
```

```
28          <repository>
29              <id>spring-releases</id>
30              <url>https://repo.spring.io/libs-release</url>
31          </repository>
32          ...
33      </repositories>
34  \vdots
35      <build>
36          <finalName>Stockmarket-Webapp</finalName>
37          <plugins>
38              <plugin>
39                  <groupId>org.apache.maven.plugins</groupId>
40                  <artifactId>maven-dependency-plugin</artifactId>
41                  <version>2.10</version>
42                  <executions>
43                      <execution>
44                          <id>copy-installed</id>
45                          <phase>install</phase>
46                          <goals>
47                              <goal>copy-dependencies</goal>
48                          </goals>
49                          <configuration>
50                              <overWriteReleases>false</overWriteReleases>
51                              <overWriteSnapshots>false</overWriteSnapshots>
52                              <overWriteIfNewer>true</overWriteIfNewer>
53                              <outputDirectory>${project.build.directory}/Stockmarket-Webapp/WEB-INF/lib</outp
54                          </configuration>
55                      </execution>
56                  </executions>
57              </plugin>
58          </plugins>
59      </build>
60  </project>
```

3.2 Umsetzung des künstlichen neuronalen Netzes mit Neuroph

In diesem Abschnitt wird beschrieben, wie das KNN aus der Konzeptionsphase umgesetzt wurde. Zum Umsetzung wurde Neurophstudio verwendet. Neurophstudio ist ein Teil von Neuroph und erlaubt das Erstellen, Trainieren sowie Testen von KNN. Nachdem das KNN angelegt wurde, musste es noch trainiert und anschließend getestet werden. Für diesen Vorgang sind Trainings- sowie Testdaten zur nötig. Die benötigten Daten konnten als

Excel-Datei von der nachfolgenden Webseite bezogen werden: *http://www.quandl.com/*. Es wurden die letzten 600 Datensätze extrahiert und dann in 2 Datensätze aufgeteilt: Einen Datensatz bestehend aus 450 Trainingsdaten sowie einen Datensatz bestehend aus 150 Testdaten.Da diese Datensätze noch nicht normalisiert waren, die Daten jeodch in normalisierter Form für das KNN zur Verfügung stehen müssen, wurden diese mit folgender Formel normalisiert:

$$N_h = \frac{A - min(A}{max(A) - min(A)} \cdot 0,8 + 01 \tag{3.1}$$

Mit Hilfer dieser Normalisierungsformle ist sichergestellt, dass sich alle WErte der Datensätze im Intervall $[0, 1]$ befinden,wobei die Multiplikation mit $0,8$ sowie die Addition mit $0,1$ Extremwerte abmildern soll.

Nachdem alle Komponenten für die Erstellung eines fertigen KNN vorhanden waren, konnte mit dem Training begonnen werden. Dafür wurden 200.000 Trainingszyklen gestartet und mit einer Lernrate von $0,7$ verwendet. Dieser Wert hat sich als ein guter Startwert herausgestellt.

3.3 Optimierung des künstlichen neuronalen Netzes

Nachdem das Grundmodell des KNN Funktionsfähig war, wurde dieses noch weiter optimiert, darauf wird nun in den folgenden Unterabschnitten genauer eingegangen.

3.3.1 Optimierung der Topologie

Das im Abschnitt 3.2 wird in diesem Abschnitt hinsichtliche der verwendeten Topolgie optimiert. Dabei wurden mit der Topologie begonnen und sukzessive Neuronen in der Zwischenschicht hinzugefügt bzw. weggenommen und für jeden Trainings- und Testverlauf der MSE notiert. die Topologie mit den geringsten MSE im Testverlauf wurde dann übernommen. DIe ERgebniss können aus der folgenden Tabelle ebtnommen werden:

Topologie	Training-MSE	Test-MSE
4-05-1	0.000	0.000
4-07-1	0.000	0.000
4-11-1	0.000	0.000
4-01-1	0.000	0.000
4-14-1	0.000	0.000
4-17-1	0.000	0.000

Tabelle 3.2: Jeweilige Topologien & korrespondierende MSE

Demnach wird die neue Topologie des KNN mit sieben Neuuronen in der versteckten Schicht arbeiten.

3.3.2 Wahl der optimalen Transferfunktion

(a) Sigmoid (b) Tanh

Abbildung 3.1: Die Sigmoide Funktion und die Tanh Funktion im Vergleich

$$(a)\ f(x) = \frac{1}{1+e^{-cx}} \qquad (b)\ f(x) = tanh(x) \tag{3.2}$$

Das KNN wurde einmal mit einer Sigmoiden Funkiton trainiert und getesten und anschließend nochmas mit einer Tanh Funktion trainiert und geteste. Die Ergebnisse können aus der untenstehnen Tabelle entnommen werden:

Transferfunktion	Mean Squared Error
Sigmoid	0.000
Tanh	0.000

Tabelle 3.3: Jeweilige Transferfunktionen & korrespondierende MSE

3.3.3 Wahl der optimalen Lernregel

Innerhalb des Verfahrens der überwachten Lernens existieren mehrere Lernregeln, um das Netz zu trainieren. Die bekannteste Lernregel dürfte hier das Backpropagation sein. Diese Regel gibt es ebenfalls in mehreren Variationen wie das momentum Backpropagation sowie das Resilient Backpropagation. Diese Fehlerrückführungsverfahren werden nun jeweils einzeln beschrieben und anschließedend genauer analysiert und das für die Anweungs am besten geeignstete Verfahren ausgewählt.

- Backpropagation: Dies ist das klassische Fehlerrükcführungsverfahren zum Anpassen der Verbindungsgewichte. Die Gewichtsveränderung erfolgt durch einen Fehlersignal, dass aus der Abweichung von tatsächlicher und prognostizierter Ausgabe berechnet wird. Die Gewichtsveränderung erfolgt hierbei schichtweise von den Ausgangsneuronen bis zu den Eingangneuronen.
- Momentum Backpropagation: Dieses Backpropagationverfahren fügt dem klassichen Verfahren einen Trägheitsterm hinzu, indem die Gewichtsveränderung zum Zeitpunkt $t-1$ berücksichtigt wird. Der Momentumwert gibt dabei die Sträkre an, wie startk dieser Wert berücksichtigt wird (zwischen 0 und 1).Durch diesen Trägheitsterm wird die Wahrscheinlichkeit, dass das KNN beim Training in ein lokales Minimum oszilliert und sich somit nicht weiter den Idealwert approximieren kann. Auch die Wahl der Lernrate gestaltet sich hier weniger kritisch.
- Zur Bestimmung des Fehlers zwischen der prognostizierten and tatsächlichen Ausgabe kann der MSE benutzt werden. Die MSE-Formel würde in den konkreten Fall der Anwendung wie folgt lauten:

$$\frac{1}{2}\sum_{i=1}^{n}(KT_i - KV_i)^2 \tag{3.3}$$

Wobei n für die Anzahl der Datensätze steht, KT_i für den tatsächlichen Ausgabewert für den Datensatz i steht und KV_i für den prognostizierten Datensatz i steht.

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

Tabelle 3.4: Jeweilige Lernregeln & korrespondierende MSE

3.3.4 Wahl der optimalen Lernrate

Die Wahl der Lernrate erweist sich oft als ein schwieriges Unterfangen und wird oft über ein trial and error Verfahren ermittelt.

Ein zu hohe Lernrate eine zu niedrige Lernrate

Das ... Empfiehlt.....

3.4 Die endgültigen künstlichen neuronalen Netze

3.5 Zusammenführung der Komponenten

4 Beschreibung der Anwendung

4.1 Architektur der Anwendung

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Das hier ist der zweite Absatz. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Und nun folgt – ob man es glaubt oder nicht – der dritte Absatz. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte

möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Nach diesem vierten Absatz beginnen wir eine neue Zählung. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

4.2 Elemente der GUI

Das hier ist der zweite Absatz. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Und nun folgt – ob man es glaubt oder nicht – der dritte Absatz. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Nach diesem vierten Absatz beginnen wir eine neue Zählung. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Das hier ist der zweite Absatz. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst

viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

5 Analyse

Und nun folgt – ob man es glaubt oder nicht – der dritte Absatz. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Nach diesem vierten Absatz beginnen wir eine neue Zählung. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Das hier ist der zweite Absatz. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Und nun folgt – ob man es glaubt oder nicht – der dritte Absatz. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

6 Fazit

Nach diesem vierten Absatz beginnen wir eine neue Zählung. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Das hier ist der zweite Absatz. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Und nun folgt – ob man es glaubt oder nicht – der dritte Absatz. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Nach diesem vierten Absatz beginnen wir eine neue Zählung. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Literaturverzeichnis