

Contents

- 2 Machine Learning 7
 - 1 “Earning from Data” — Finanzanalyse 7
 - 1.1 Formen der Finanzanalyse 7
 - 1.2 Einsatz von Machine Learning 7

Machine Learning

SVMs und Neuronale Netzwerke

1 “Earning from Data” — Finanzanalyse

Die Finanzanalyse versucht künftige Kursverläufe vorherzusagen, um Gewinne im Finanzhandel zu erzielen. In unserem Projekt versuchen wir dies durch den Einsatz von *Machine Learning* zu erreichen.

1.1 Formen der Finanzanalyse

Die drei grundlegenden Formen der Finanzanalyse bilden die *Fundamentalanalyse*, *Technische Analyse* und die *Sentimentanalyse*.

Die Fundamental- und die Technische Analyse gehen davon aus, dass nicht alle verfügbaren Informationen in den Kursen verarbeitet sind. Es sei somit möglich, schneller als andere Marktteilnehmer zu agieren und daraus Profite zu erzielen.

1.1.1 Fundamentalanalyse

Bei der Fundamentalanalyse werden sogenannte *Fundamentaldaten* analysiert, um den wirklichen Wert eines Finanzproduktes zu bestimmen. Zu diesen Daten zählen unter anderem:

- Kurs-Gewinn-Verhältnis
- Gesamtkapitalrendite
- Eigenkapitalquote
- Bruttoinlandsprodukt
- Einzelhandelsverkäufe

1.1.2 Technische Analyse

Im Gegensatz zur Fundamentalanalyse werden bei der Technischen Analyse *Charts*, d.h. Abbildungen von Kursverläufen analysiert. Dazu werden sowohl reine Preisverläufe, als auch von diesen abgeleitete *Indikatoren* berücksichtigt. Zu diesen Indikatoren gehören unter anderem:

- Moving Average
- Bollinger Bands
- Stochastic Oscillator
- Relative Strength Index
- Fractals

1.1.3 Sentimentanalyse

Die Sentimentanalyse befasst sich mit der Stimmung von Investoren, um daraus zu schließen, ob eine *bullische Phase* (steigender Trend) oder eine



Abb. 2.1: Ein Chart (eine Kerze $\hat{=}$ eine Stunde) des EUR/USD Währungspaares mit den genannten Beispielindikatoren (siehe 1.1.2)

bärische Phase (fallender Trend) bevorsteht. Dazu können Mittel wie etwa *Meinungsumfragen* oder die Analyse von *Börsenbriefen* eingesetzt werden.

1.2 Einsatz von Machine Learning

Unser Projekt befasst sich ausschließlich mit der Technischen Analyse (siehe 1.1.2). Der klassische Ansatz würde die Aufgabe der Analyse einem Menschen oder einem selbstgeschriebenen Algorithmus überlassen.

Wir hingegen versuchen diese Aufgabe auf ein *neuronales Netzwerk* zu übertragen. Hierbei spielt die Auswahl der Features und der verwendeten Parameter wie z.B. Schichtenanzahl des Netzwerks eine sehr wichtige Rolle.

1.2.1 Verwendete Features

Wir haben uns für die Verwendung des *Stochastic Oscillator* und des *Relative Strength Index* (jeweils mit Standardparametern) als Features entschieden. Es handelt sich dabei um zwei der bekanntesten Indikatoren für die technische Analyse.

Stochastic Oscillator Der Stochastic Oscillator ist ein von Dr. George Lane in den 1950er Jahren entwickelter Indikator. Dieser versucht, bevorstehende Trendwenden anhand von Unterstützungs- und Widerstandslinien zu erkennen.

Er setzt sich aus zwei Elementen zusammen: Der %K- und der %D-Kurve, wobei %D ein gleitender Durchschnitt von %K ist.

Für unsere Features betrachten wir jedoch nur die %K-Kurve. Diese lässt sich wie folgt berechnen:

$$\%K = 100 * \frac{Preis - T_5}{H_5 - T_5}$$

mit:

$Preis$ = Aktueller Schlusskurs

T_5 = Tief der letzten fünf Perioden

H_5 = Hoch der letzten fünf Perioden



Abb. 2.2: Beispielchart des EUR/USD mit dem Stochastic Oscillator (Standardparameter) im unteren Fenster (blau: %K-Kurve, rot: %D-Kurve)

Relative Strength Index Der Relative Strength Index (dt.: “Relative Stärke Index”) ist ein von J. Welles Wilder Ende der 1970er Jahre entwickelter Indikator. Wie auch der Stochastic Oscillator versucht dieser, bevorstehende Trendwenden zu erkennen. Hierzu analysiert dieser Indikator Kursbewegungen entsprechend ihrer Richtung und Größe.

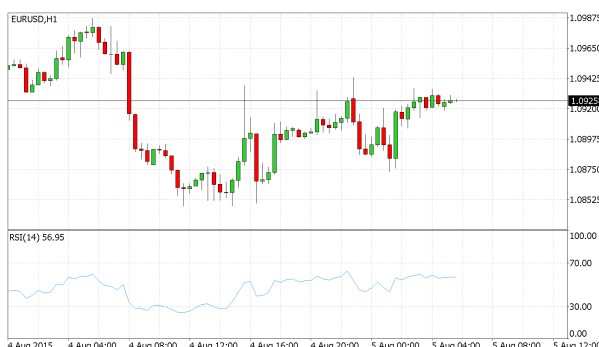


Abb. 2.3: Beispielchart des EUR/USD mit dem Relative Strength Index (Standardparameter) im unteren Fenster

1.2.2 Verwendete Datensätze

Für unsere Datensätze haben wir jeweils 65536 gelabelte Featurevektoren für verschiedene Währungspaare gesammelt. Jeder Featurevektor setzt sich aus jeweils vier von -1 bis $+1$ normalisierten Werten des Stochastic Oscillators und des Relative Strength Indexes zusammen. Somit sind diese insgesamt 8 Elemente lang. Bei unseren Labels unterscheiden wir lediglich, ob der Kurs in der auf die Features folgenden Stunde gestiegen oder gefallen ist.

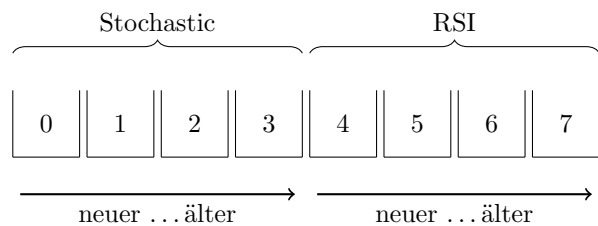


Abb. 2.4: Schematischer Aufbau unserer Featurevektoren

Insgesamt haben wir jeweils mehr als zehn Jahre an Features und Labels für die Währungspaare CHF/JPY, EUR/AUD, EUR/CAD, EUR/CHF, EUR/USD und GBP/JPY zusammengestellt.

Diese Daten haben wir mittels eines selbstgeschriebenen Skripts für MetaTrader in einem CSV-ähnlichen Format exportiert.

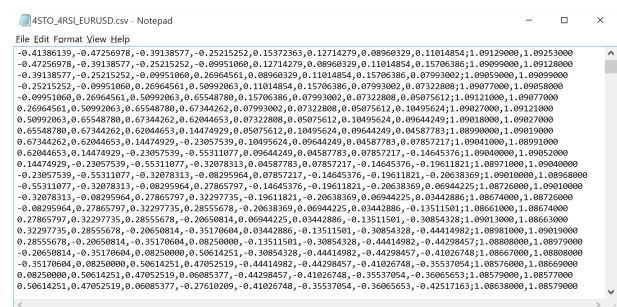


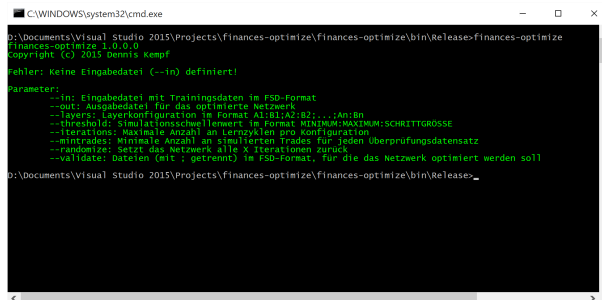
Abb. 2.5: Ausschnitt aus unserem Datensatz für den EUR/USD

1.2.3 Verwendete Parameter

Die Bestimmung der Parameter für das neuronale Netzwerk stellt für unser Projekt die interessanteste, aber auch schwierigste Aufgabe dar. Denn das Verändern eines einzelnen Parameters hat meist ein vollkommen anderes Ergebnis zur Folge. Zu den zu optimierenden Parametern gehören unter anderem:

- Schichtenanzahl des neuronalen Netzwerks
- Neuronenanzahl in den Schichten
- Anfangsgewichte des Netzwerks
- Lerndurchläufe des Netzwerks

Um diese Aufgabe zu lösen haben wir uns dazu entschieden ein Optimierungsprogramm (finances-optimize.exe) zu schreiben. Dieses iteriert durch alle möglichen Netzwerkkonfigurationen, überprüft sie anhand von Testdatensätzen und speichert das beste Ergebnis in einer Datei.



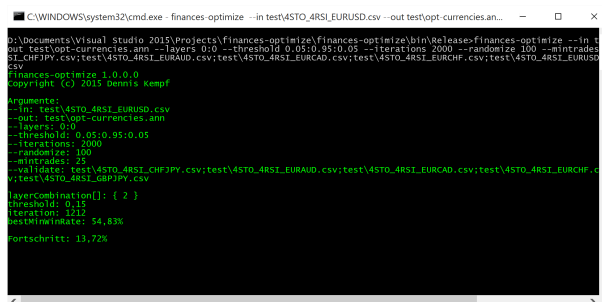
```

C:\WINDOWS\system32\cmd.exe
D:\Documents\Visual Studio 2015\Projects\Finances-optimize\Finances-optimize\bin\Release\finances-optimize
finances-optimize 1.0.0.0
Copyright (C) 2015 Dennis Kempf

fehler: keine Eingabedatei (--in) definiert!

Parameter:
--in: Eingabedatei mit Trainingsdaten im FSD-Format
--out: Ausgabedatei für das optimierte Netzwerk
--layers: Layerkonfiguration im Format A1:B1:A2:B2;...;An:Bn
--threshold: Simulationsschwellenwert im Format MINIMUM:MAXIMUM:SCHRITTGROSSE
--iterations: Maximale Anzahl an Lernzyklen pro Konfiguration
--mintrades: Minimale Anzahl an simulierten Trades für jeden Überprüfungsdatensatz
--randomize: Setzt das Netzwerk alle X Iterationen zurück
--validate: Boolean (false = getrennt) im FSD-Format, für die das Netzwerk optimiert werden soll
  
```

Abb. 2.6: Alle möglichen Parameter für das Optimierungsprogramm



```

C:\WINDOWS\system32\cmd.exe - finances-optimize -in test\4STO_4RSI_EURUSD.csv -out test\opt-currencies.ann
D:\Documents\Visual Studio 2015\Projects\Finances-optimize\Finances-optimize\bin\Release\finances-optimize --in test\4STO_4RSI_EURUSD.csv --out test\opt-currencies.ann --layers 0:0 --threshold 0.05:0.95:0.05 --iterations 2000 --randomize 100 --mintrades 10 --validate test\4STO_4RSI_CHFJPY.csv;test\4STO_4RSI_EURAUD.csv;test\4STO_4RSI_EURCAD.csv;test\4STO_4RSI_EURCHF.csv;test\4STO_4RSI_EURUSD.csv
finances-optimize 1.0.0.0
Copyright (C) 2015 Dennis Kempf

Parameter:
--in: test\4STO_4RSI_EURUSD.csv
--out: test\opt-currencies.ann
--layers: 0:0
--threshold: 0.05:0.95:0.05
--iterations: 2000
--randomize: 100
--mintrades: 10
--validate: test\4STO_4RSI_CHFJPY.csv;test\4STO_4RSI_EURAUD.csv;test\4STO_4RSI_EURCAD.csv;test\4STO_4RSI_EURCHF.csv;test\4STO_4RSI_EURUSD.csv
LayerCombination[]: ( 2 )
Threshold: 0.15
Iteration: 1215
BestResult: 54.83%
Schrittschritt: 13.72%
  
```

Abb. 2.7: Optimierungsprozess auf Basis linearer Regression (neuronales Netzwerk ohne “hidden layer”)

1.2.4 Ergebnis

Überraschenderweise lassen sich mithilfe der linearen Regression äußerst schnell sehr gute Ergebnisse erreichen. Die aus der Optimierung hervorgehenden Netzwerke bieten oftmals eine Vorhersagegenauigkeit von $\approx 60\%$, ob ein Kurs in der nächsten Stunde steigt oder fällt, für alle getesteten Datensätze (CHF/JPY, EUR/AUD, EUR/CAD, EUR/CHF, EUR/USD, GBP/JPY).

Ein ähnliches Ergebnis lässt sich auch mit tiefen neuronalen Netzen erreichen. Hierbei besteht allerdings das Problem, dass es sehr schnell zu “overfitting” kommt, und dass die Optimierung wesentlich länger dauert.

1.2.5 Verbesserungsmöglichkeiten

Features Für unsere Features greifen wir lediglich auf zwei Indikatoren zurück. Eine breitere Vielfalt könnte zu noch besseren Ergebnissen führen. Zu untersuchen wäre dabei auch, welche Auswirkungen eine Veränderung der Anzahl an Werten jedes Indikators im Featurevektor hat.

Datensätze Die meisten unserer Datensätze stehen in Verbindung zum Euro und stammen aus dem gleichen Zeitraum. Auch hier wäre zu untersuchen, wie sich eine größere Diversität auf das Ergebnis auswirkt.

Optimierungsprogramm Das Optimierungsprogramm ist aktuell noch single-threaded, d.h. es arbeitet nur auf einem einzigen Rechenkern. Jedoch ist die Aufgabe, alle möglichen Netzwerkkonfigurationen zu evaluieren, sehr gut auf mehrere Rechenkerne verteilbar.

Zukünftige Revisionen sollten deshalb auch von diesen Gebrauch machen, um die Rechenzeit drastisch zu verkürzen. Hierdurch wird auch das Optimieren von tiefen neuronalen Netzen deutlich vereinfacht und praktikabler.