

Aktiemarknadsprediktion med artificiella neurala nätverk

Joel Andersson

Datateknik

Åbo Akademi

April 10, 2017

Contents

1	Inledning	3
2	Teori	4
2.1	Maskininlärning	4
2.2	Artificiella neurala nätverk	4
2.2.1	Användningsområden	5
2.2.2	Klassificering	5
2.2.3	Framkopplingsnätverk	6
2.2.4	Perceptronen	6
2.2.5	Sigmoidfunktionen	8
2.2.6	Inlärningsmetoder	8
2.2.7	Bakåtpropagering	9
2.2.8	Radial Basis Function	10
2.3	Genetiska algoritmer	10
2.4	Aktiemarknaden	11
2.4.1	Teknisk analys	12
3	Jämförelse av experiment	13
3.1	Forecasting closing price indices using neural networks.	13
3.1.1	Slutsats	15
3.2	Stock Market Prediction Using Machine Learning Techniques	15
3.2.1	Slutsats	16
3.3	A Neural Network based approach to support the Market Making strategies in High-Frequency Trading	16
3.3.1	Slutsats	19
3.4	Stock Market Forecasting Algorithm Based on Improved Neural Network	20
3.4.1	Slutsats	22
4	Sammanfattning	22

Referat

Artificiella neurala nätverk kan tillämpas för att lösa en myriad av klassificeringsproblem och i denna avhandling undersöks deras potential inom prognos av aktiekurser. Först genomgås den underliggande teorin och sedan presenteras, analyseras och jämförs några försök på tillämpning av sådana artificiella neurala nätverk. En av de populäraste metoderna är att använda flerskikt perceptroner med bakåtpropagering, men problemet med bakåtpropagering är att den inlärningsmetoden ibland minimerar felet till lokala minimum istället för globala minimum. Däremot är genetiska algoritmer bra på optimeringsproblem och kan användas för att hitta globala minimum i högdimensionella rymder. Därför undersöks även möjligheten att använda en kombination av genetiska algoritmer och artificiella neurala nätverk för att förbättra inlärningskvaliteten.

1 Inledning

Prediktion av aktiekurser är ett intressant problem eftersom aktiekurser inte följer något direkt mönster utan beror av väldigt många yttre orsaker. Om man kan förutsäga huruvida värdet på aktier kommer stiga eller sjunka har man stor potential att tjäna pengar, därför har många forskare inom datavetenskap och ekonomi försökt utveckla system som kan göra just detta med så hög precision som möjligt.

En aktiemarknad är en allmän marknad (fysisk eller virtuell) där man kan köpa och sälja aktier. Det finns flera olika sådana, t.ex. Helsingforsbörsen, New York stock exchange och NASDAQ. Man kan antingen spara sina aktier för att få dividender eller sälja för vinst. När man pratar om att aktiemarknaden sjunker eller stiger syftar man egentligen på att ett aktieindex sjunker eller stiger. Ett aktieindex är det sammanlagda värdet av de aktier som ingår i indexet.

Baserat på den information, om ett aktieindex, som finns tillgänglig kan man pussla samman mer beskrivande variabler, till exempel glidande medelvärde och relativt styrkeindex, som gör att man kan packa stora mängder information i små paket. Om man då samlar ihop en bunt liknande variabler kan man bygga ett

relativt simpelt artificiellt neuralt nätverk, ANN, eftersom det blir färre inputs än om man försökte använda alla enskilda värden som input.

ANN är en vanlig metod för att förutsäga marknadens riktning [7], eftersom de är kraftfulla verktyg för att finna otydliga relationer och samband mellan input och output. Men bakåtpropageringsalgoritmen, som ofta används för inläring av flerskiktade ANN har en tendens att bara minimera tills den hittar lokala minimum istället för globala minimum [18]. Genetiska algoritmer å andra sidan kan utnyttjar det naturliga urvalet för att hitta globala minimum [18].

I denna avhandling undersöks hur lämpliga ANN är för att förutsäga aktiekursens riktning och hur de kan tillämpas för så kallad hög-frekvent handel, samt om man kan uppnå bättre resultat genom att kombinera dem med genetiska algoritmer.

2 Teori

2.1 Maskininläring

Maskininläring definierades år 1959 av Arthur Samuel: "Maskininläring är ett delfält av datorvetenskap som ger datorer förmågan att lära sig utan att bli explicit programmerade" [2].

Den grenen av maskininläring vi är intresserade av är artificiella neurala nätverk, vilket tas upp i följande kapitel.

2.2 Artificiella neurala nätverk

Artificiella neurala nätverk, även kallade neurala nätverk, är inspirerade av biologiska neurala nätverk [2]. Inom biologin är neuronerna en cell som kan sända och bearbeta kemiska eller elektriska signaler [2]. Varje neuron har input (dendrit), en aktiveringsfunktion (cellkropp) och output (axon) [2]. Axonen blir sedan uppdelad i många delar och kopplas till dendriterna av andra neuroner [2]. Om den totala signalstyrkan som neuronens alla dendriter tar emot är tillräckligt stor "avfyrrar" den neuronerna i sin tur en signal, via dess axon, till alla neuroner den är kopplad till [6].

ANN är en mängd artificiella neuroner (kallas ofta noder) sammansatta i ett nätverk, där de är uppdelade i skikt/lager [6]. Det finns alltid ett inputlager och ett output lager. Om man har ett eller flera dolda lager är det fråga om flerskiktsperceptroner [5]. Man kan även ha så kallade partiska noder (eng. bias nodes) som själva inte tar emot någon input men de ger alltid ut en output till de noder de är kopplade till[5].

2.2.1 Användningsområden

Artificiella neurala nätverk är främst användbara vid sådana problem där skapandet av en konventionell algoritm skulle bli för komplicerad, eller där man vill analysera stora mängder data för att hitta mönster som inte är tydliga.

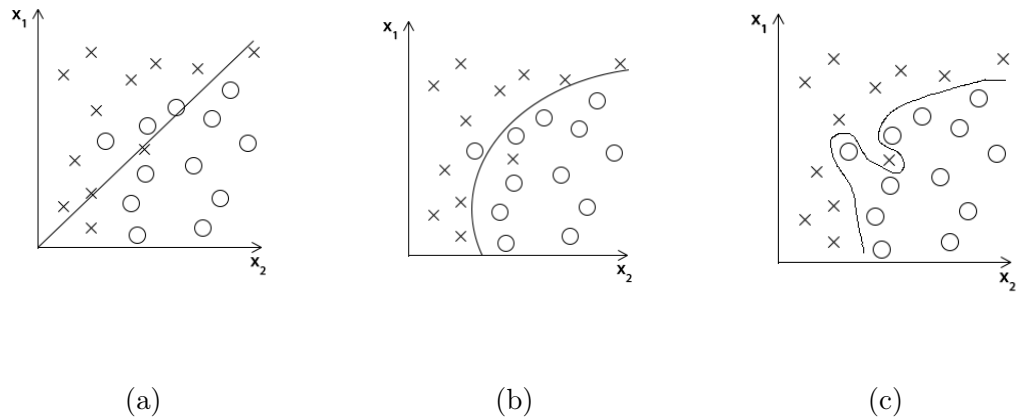
Självkörande bilar [15]. Tesla, bland andra, har implementerat program i sina bilar som gör att de kan kontrollera sig själva till en viss grad. Kräver bildigenkännings-mjukvara vilket man ofta använder neurala nätverk till.

Klassificering [5]. Neurala nätverk är mycket bra på att känna igen mönster och kan således användas till att klassificera nästan vad som helst som uttrycker särskilda mönster. T.ex. har man börjat använda neurala nätverk för att diagnosticera cancer [16]. Ett annat användbart klassificeringsproblem är igenkänning av tecken, textigenkänning med andra ord [5].

Robotteknik. Man har exempelvis använt sig av neurala nätverk för att tolka EMG signaler och på så sätt styra en handprotes [9].

2.2.2 Klassificering

Klassificering innebär att man jämför egenskaper som något objekt uttrycker och sedan avgör till vilken klass objektet hör. Klassificering kan ibland göras till acceptabla noggrannheter med hjälp av en linjär klassificerare [5]. Om man fortsätter mata ett artificiellt neuralt nätverk med samma data kan det uppnå perfekt separation i klassificeringen men det lönar sig nästan aldrig för då har nätverket inte fått en generaliserad modell utan en överanpassad (eng. overfitted) modell [5] som i figur 1c. För de flesta tillämpningarna lönar det sig att hålla komplexiteten på en medelhög nivå [5].



Figur 1: Ett hypotetiskt klassificeringsproblem med två egenskapsvariabler x_1 och x_2 . Cirkelar representerar objekt från klass C1 och kryss representerar objekt från klass C2. (a) Beslutsgränsen separerar klasserna till viss mån men det finns några fall som blev fel. Modellen är av för låg komplexitet [5]. Med en mer komplex modell än i (a) kan man uppnå bättre klassificering [5] som i (b). En modell med hög komplexitet ger perfekt separation som visas i (c).

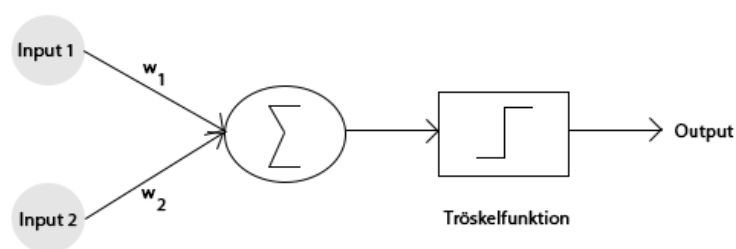
2.2.3 Framkopplingsnätverk

Feed-forward networks, framkopplingsnätverk hädanefter, är ANN i vilken information bara förs framåt, utdata från ett lager blir indata i nästa lager [5], till skillnad från t.ex. rekurrenta neurala nätverk. Det första lagret är alltså det enda som får input utifrån [5]. Framkopplingsnätverk en ofta använd typ av neurala nätverk.

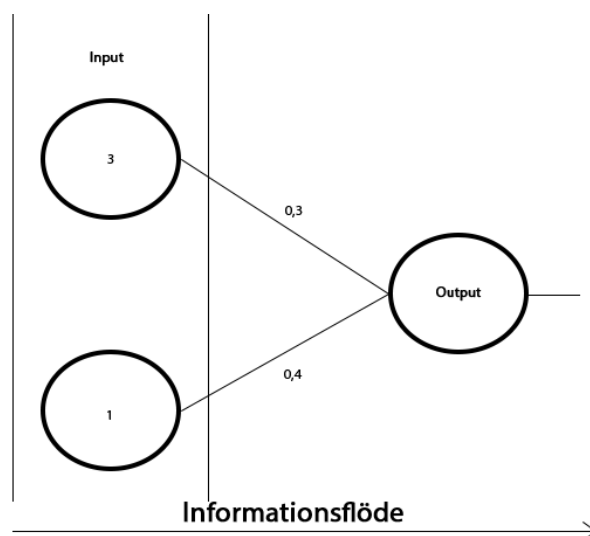
2.2.4 Perceptronen

Perceptronen är hörnstenen i ANN. Den tar emot input-data i vektorform och matar in värdena i aktiveringsfunktionen som beräknar outputen [2]. Varje värde har en samhörande vikt som bestämmer hur stor inverkan just detta värde skall ha på outputen [2]. Den viktade summan fås genom att summera alla värden i inputen multiplicerat med motsvarande vikt, som i figur 2.

$$a_j = \sum_i (w_{ji} Z_i)$$



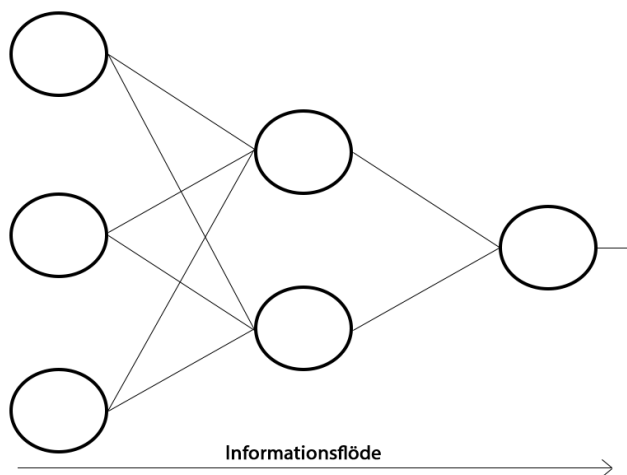
Figur 2: En neuron med tröskelvärdesfunktion.



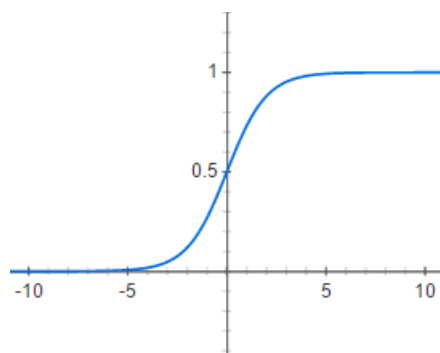
Figur 3: En perceptron med två inputs med värden 3 och 1 respektive vikterna 0,3 och 0,4; och en output.

Outputen för varje nod a är summan av varje input-värde Z gånger motsvarande vikt w [5]. Om summan av ingångsvärdena blir större än tröskelvärdet aktiveras perceptronen och ger värdet 1, annars ger den värdet 0 [2].

Flerskiktsperceptroner, MLP, är neurala nätverk med åtminstone ett dolt lager, se figur 4. Om man parar ihop lagren kan man betrakta dem som enskilda perceptroner men inlärningen fungerar inte riktigt på samma sätt [5]. Istället kan man t.ex. använda bakåtpropagering [5]. MLP illustrerad i figur 4.



Figur 4: Flerskiktsperceptron med ett dolt lager.



Figur 5: Sigmoidfunktionen

2.2.5 Sigmoidfunktionen

Ibland använder man sigmoidfunktionen som aktiveringsfunktion, i de artificiella neuronerna, vilken ger som svar hur "starkt" neuronerna avfyras, som ett värde mellan noll och ett [5]. Eftersom man nu får ett varierande värde, i motsats till tröskelvärdesfunktionen som bara ger antingen noll eller ett, kan det tolkas som hur säker det neurala nätverket är på sitt svar. Sigmoidfunktionen i figur 5.

$$S(t) = \frac{1}{1 + e^{-t}}$$

2.2.6 Inlärningsmetoder

Målet med träning av ett nätverk är inte att lära nätverket en exakt representation av dess träningsdata utan att bygga en statistisk modell av processen som

genererar denna data [5]. Detta är viktigt om nätverket ska kunna göra bra prediktioner baserat på nya inputs [5].

Man delar in inlärningsmetoderna i övervakad och oövervakad inlärning [6]. Övervakad inlärning innebär att man ger nätverket både input och önskad output [6]. Sedan jämför man outputen med den önskade outputen och beräknar skillnaden, felet, och justerar vikterna enligt det [6]. Före inlärningen börjar sätter man ofta vikterna enligt hur man uppskattar att de borde vara eller låter dem anta slumpvärden [6]. Denna process av inputmatning och korrigerande av vikterna fortsätter tills outputen är tillräckligt nära den förväntade outputen eller inlärningsprocessen gjort ett givet antal iterationer [6].

Oövervakad inlärning innebär att man inte egentligen vet vad svaret är så man "säger" inte åt nätverket vad som är rätt eller fel. Istället låter man nätverket organisera sig så att varje nod svarar olika kraftigt på olika inputs, vilket betyder att man kan upptäcka olika samband som annars kanske inte syns direkt i insamlad data [6]. Oövervakad inlärning kräver ofta mer tid och resurser än övervakad inlärning [2].

2.2.7 Bakåtpropagering

Bakåtpropagering, BP, är en inlärningsalgoritm som kan användas för flerskiktade neurala nätverk [2]. Idén med bakåtpropagering är att korrigera vikterna för alla inputs enligt felet i outputen [5]. De noder som "gissar" rätt får större vikt och de som gissar fel får mindre vikt [5]. Takten med vilken nätverket lär sig omnämns inlärningstakten (eng. learning rate) [2]. En högre inlärningstakt gör att inlärningsprocessen tar kortare tid men kvaliteten på inlärningen är lägre och en lägre inlärningstakt gör att inlärningsprocessen tar längre tid men kvaliteten är högre [2].

Om man föreställer sig ett nätverk med deriverbara aktiveringsfunktioner blir outputerna av noderna deriverbara funktioner av både input variablerna och vikterna (och statiska partiska inputs) [5]. Låt definiera en fel-funktion så som "summa-av-kvadrat-felet", vilken är en deriverbar funktion av nätverkets outputar, då är felet självt en deriverbar funktion av vikterna [5]. Således kan derivatorna av felet i förhållande till vikterna evalueras och dessa derivator kan

användas för att hitta de värden för vikterna som minimerar fel-funktionen [6]. Detta betyder att bakåtpropageringsalgoritmen kan minimera felet men risken finns att den fastnar i ett lokalt minimum istället för ett globalt minimum [18].

Bakåtpropagering är en bra inlärningsalgoritm men eftersom BP ibland hittar lokala minimum istället för globala minimum. Med andra ord kunde nätverket tränas bättre. En lösning är att använda genetiska algoritmer för att optimera vikterna och antalet dolda noder.

2.2.8 Radial Basis Function

”Radial basis function”, härnäst bara RBF pga. brist på svenskt begrepp, är en modell av neurala nätverk i vilken outputen av en dold nod bestäms på basen av avståndet mellan input vektorn och en prototypvektor [6] som motsvarar ett exempel från dess träningsdata [13]. Kan uttryckas som

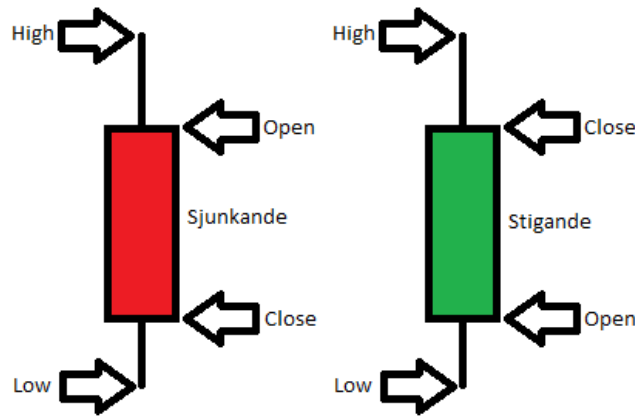
$$h(x) = \sum_{i=1}^n w_i \Phi(\|x - y_i\|)$$

där x är inputen, w är vikten för inputen x och y är prototypvektorn.

2.3 Genetiska algoritmer

Genetiska algoritmer, GA, är en heuristisk metod som imiterar evolutionen för att hitta optimala lösningar [11]. Idén är att man har en population av en godtycklig mängd objekt med slumpmässigt genererade egenskaper och jämför dem med det mål man har [11]. Genom det naturliga urvalet väljs de bäst lämpade objekten (närmast målet) och deras genetiska data kombineras och bildar en ny version som möjligtvis är lite närmare målet [11]. För att inte algoritmen ska stanna i ett visst tillstånd införs slumpmässiga mutationer [11].

Föreställ dig att vi vill använda genetiska algoritmer för att komma fram till strängen "blå bär". Varje objekt i vår population är av samma längd som "blå bär", alltså sju tecken. Varje tecken är en egenskap som kan kombineras och muteras. Vår population består av "blå kor", "popcorn" och "skidbar". GA:n jämför objekten och kommer fram till att "blå kor" har fem tecken gemensamt med målet, "popcorn" har noll tecken gemensamt och "skidbar" har två. Den



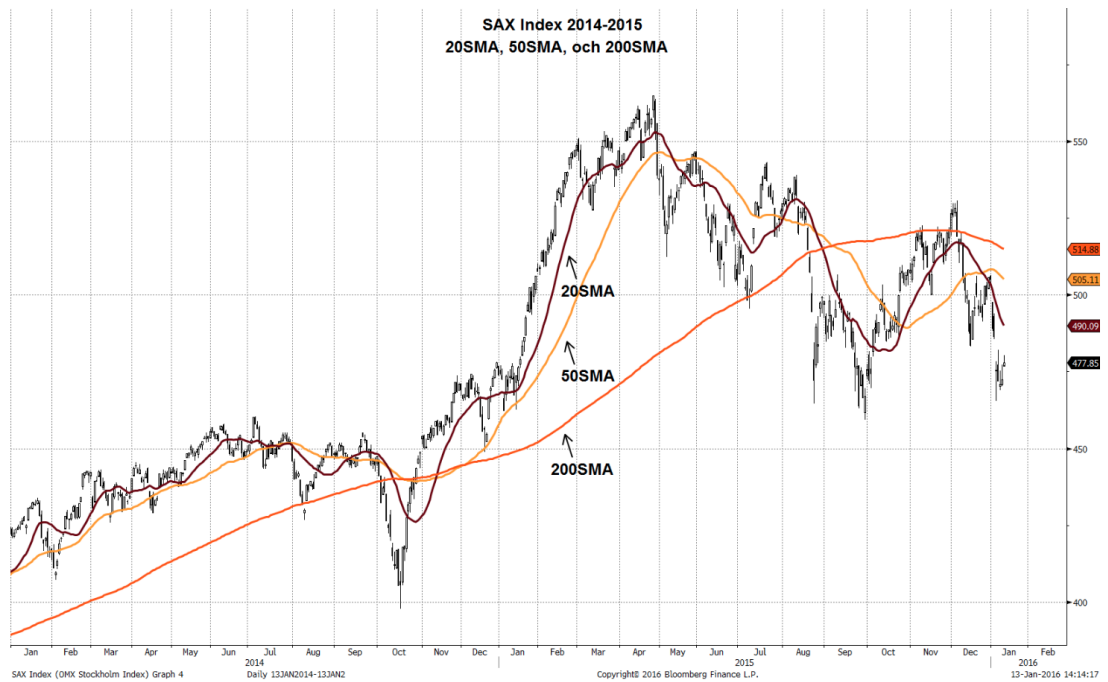
Figur 6: Aktiekursvärden illustrerade med ett candlestick-diagram

genetiska algoritmen väljer då två objekt (där de som har fler tecken har större chans att bli valda) att kombinera t.ex. genom korsning vilket innebär att man tar första delen av ena objekten och andra delen av det andra objektet. Låt säga att "blå kor" och "skidbar" väljs. Vi fick nu ett nytt objekt "blå bar". På detta vis håller den genetiska algoritmen på tills den nått målet, men i detta fall kommer målet aldrig nås, eftersom det inte finns ett "ä" någonstans i de objekt vi har, så vi måste introducera mutationer. Mutationer är alltså slumpmässiga förändringar och i vårt fall kunde det betyda att något tecken vid parningen blir ett som ingendera föräldrarna har.

2.4 Aktiemarknaden

För alla aktier på aktiemarknaden kan information om dess kurs hämtas. Tanken är då att det med hjälp av dessa värden, eller kombinationer av dem, för de senaste X antal tidsenheter ska vara möjligt att förutsäga aktiekursen, eller åtminstone dess riktning, med en viss sannolikhet.

Volymen är den mängd aktier som byter händer under en given tidsperiod [3]. Högsta kurs (eng. high) är det högsta värdet aktiekursen haft under en viss tid. Låt säga en timme, eller en dag [3]. Lägsta kurs (eng. low) är det lägsta värde aktiekursen haft under perioden [3]. Öppningskurs (eng. open) är det värde kursen har i början av perioden [3]. Stängningskurs (eng. close) är det värde kursen har i slutet av perioden [3]. Illustrerat i figur 6.



Figur 7: Enkla glidande medelvärden som beaktar 20, 50 och 200 dagar [3]

2.4.1 Teknisk analys

Inom teknisk analys finns tekniska indikatorer, TI, vilka kombinerar rå information om aktiekurser för att ge en mer indikativ bild om vad som händer med aktien [4]. TI:n kategoriseras enligt följande: eftersläpande och ledande TI:n [4]. Eftersläpande TI:n följer prisets rörelse, trenden, efter att den börjat. Ett exempel på eftersläpande TI är glidande medelvärde [4]. Ledande TI:ns signal föregår prisets rörelse, dvs. de genererar sin signal före en ny trend. Ett exempel på ledande TI är relativt styrkeindex [4].

Det glidande medelvärdet är genomsnittspriset över en given tidsperiod [3]. Det följer aktiekursen men jämnar ut kursen så att stora fluktuationer försvinner [3]. Det finns olika sorters glidande medelvärden och ett av de vanligaste är det enkla glidande medelvärdet, SMA, [3]. Ju fler dagar ett glidande medelvärde beaktar desto mindre känsligt för förändring kommer det vara. Man kan få väldigt olika grafer beroende på hur många dagar man vill att det glidande medelvärdet ska beakta åt gången, vilket framgår tydligt i figur 7.

Relativt styrkeindex mäter prisförändringens hastighet [4].

$$RSI = 100 - \frac{100}{1 + RS}$$

Där RS är antalet uppgångar av n dagar delat på antalet nedgångar av n dagar [4]

3 Jämförelse av experiment

3.1 Forecasting closing price indices using neural networks.

Patel och Marwala [1] har utvecklat två olika modeller för neurala nätverk för att förutsäga följande dags stängningskurs. En flerskiktsperceptron och ett RBF-nätverk. Nätverkens uppgift är att baserat på de tidigare dagarnas stängningskurser förutsäga följande dags stängningsvärde. Båda ANN-typerna har fyra outputar som indikerar åt vilket håll och hur starkt följande dags stängningsvärde kommer röra sig. De fyra output-klasserna är kraftig ökning, svag ökning, svag minskning, kraftig minskning.

MLP:s dolda lager använder en icke-linjär aktiveringsfunktion: den hyperboliska tangenten (ser ut som sigmoidfunktionen men brantare och värdet kan röra sig mellan -1 och 1 istället för 0 och 1) och för output-lagret sigmoidfunktionen. RBF-nätverket använder en Gaussisk aktiveringsfunktion (normalfördelning) i dess dolda lager och en linjär aktiveringsfunktion (0 eller 1, likt perceptronen) i output-lagret.

Den data som användes delades upp i tre delar: träning, validering, och test; för att undvika över- och underanpassning. Patel och Marwala valde att använda aktieindexen Dow Jones Industrial Average, JSE All Share, NASDAQ 100 och Nikkei 225 Stock Average som input-data till deras nätverk. Patel och Marwala normaliserade sin data enligt följande formel.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Där X-norm är normaliserad data, X är ursprunglig data, X-min är det minsta värdet och X-max är det största värdet.

Antalet dolda noder optimerades genom att minimera en fel-funktion som avbildade (eng. mapping) antalet dolda noder mot de färdiga nätverkens noggrannhet. Varje dags stängningskurs används som en skild input för nätverken.

Antalet inputs optimerades på samma sätt som antalet dolda noder. Efter den här optimeringen testades de kombinationer med bäst resultat och man kom fram till att MLP:n gav bäst resultat för Dow Jones med 16 input dagar och 58 dolda noder (61,5 % noggrannhet), för JSE med 18 input dagar och 26 eller 35 dolda noder (67,5 % noggrannhet), för NASDAQ med 19 input dagar samt 148 dolda noder (69 % noggrannhet) och för Nikkei med 18 input dagar samt 103 dolda noder (66 % noggrannhet). RBF-nätverken hade bäst noggrannhet med 18 input dagar och 73 dolda noder för Dow Jones (58 % noggrannhet), med 9 input dagar och 18 dolda noder för JSE (61,5 % noggrannhet), med 19 eller 15 input-dagar och 64 eller 99 dolda noder för NASDAQ (63 % noggrannhet) samt med 17 eller 18 input-dagar och 95 eller 138 dolda noder för Nikkei (61,5 % noggrannhet). Flerskiktsperceptronerna presterade alltså betydligt bättre i detta test.

Glidande medelvärden med olika antal dagar som input iaktogs och de konstaterades öka nätverkens noggrannhet vid prognos av Dow Jones Industrial Average, Nasdaq 100 och Nikkei 225 Stock Average indexen. Noggrannheten vid prognos av JSE All Share ökade inte när glidande medelvärde introducerades som input.

De fem bästa MLP-nätverken från input-tidsfönster-optimeringen och glidande-medelvärdes-undersökningen valdes ut och användes i en kommitté där nätverkens outputar användes som röster i ett röstningssystem som bestämmer vilken av de fyra klasserna som uppfylls följande dag. Om majoriteten av de artificiella neurala nätverken klassificerade en output som samma klass klassificerade röstningssystemet kommitténs output som den klassen. Om två nätverk ger samma output och två andra ger en annan output klassificerar röstningssystemet kommitténs output som obestämmd.

Kommittén uppnådde en träffsäkerhet på 74,9 % för Dow Jones Industrial Average, 67,8 % för JSE All Share, 64 % för Nikkei 225 Stock Average. För prognos av NASDAQ användes en ensam MLP med 19 input-dagar, 5 dagars glidande medelvärde och 148 dolda noder. Den ensamma MLP:n uppnådde en precision på 70,4 %.

3.1.1 Slutsats

Den kommitté av artificiella neurala nätverk Patel och Marwala utvecklade uppnådde en precision på 72 % som bäst och 64 % som värst. Det vill säga tillräckligt goda resultat för att vara lönsamt. Patel och Marwala påstår att en högre precision kunde uppnås med hjälp av en komplex kommitté av klassificerare. Man kunde även använda genetiska algoritmer vid optimeringen av ANN arkitekturen.

3.2 Stock Market Prediction Using Machine Learning Techniques

Usmani et al. [12] har likt Marwala och Patel jämfört flerskiktsperceptroner och RBF-nätverk för att göra aktiekursprognos. Utöver det har de också testat en vanlig perceptron. Usmanis team använder likt Patel och Marwala ett dolt lager i deras MLP och RBF-nätverk. Deras nätverk appliceras på KSE-100 indexet.

Usmanis team använder 6 olika sorters input i sina nätverk som alla normaliseras så att de får ett värde mellan 1 och -1.

1. Marknadshistorik. Marknadshistorik i form av SMA och en annan teknisk indikator, ARIMA, användes. ARIMA är, likt SMA, en form av glidande medelvärde men mycket mer komplicerad och jag kommer inte gå in på hur den fungerar här. För en förklaring på hur ARIMA fungerar, se [17].
2. Nyheter. Med hjälp av "text mining" tolkas nyheternas affärs-, finans- och politiksegment, samt internationella händelser. Här användes en tjänst som kallas OpinionFinder. Tolkas som antingen positiv eller negativ.
3. Allmänhetens sinnesstämning. Twitter användes som källa för allmänhetens sinnesstämning. OpinionFinder användes också för detta. Tolkas som antingen positiv eller negativ.
4. Handelsvarors priser. Priset på handelsvaror såsom bensin, guld och silver påstås kunna påverka aktiekurser. Därför används de som input.
5. Ränta. Räntorna från två Pakistanska banker används som input.

6. Utrikeshandel. Växlingskursen för den Pakistanska rubeln och den Amerikanska dollarn används som input.

3.2.1 Slutsats

Usmani et al. kom fram till att bensin priset var det som påverkade KSE-100-indexets pris till största grad och utrikeshandeln påverkade inte indexet alls.

Den enkla perceptronen uppnådde en precision på 60 %, MLP:n 77 % och RBF-nätverket 63 %. Resultaten liknar de som Patel och Marwala fick och man kan igen konstatera att flerskiktsperceptronen var det bästa alternativet av de som testades.

3.3 A Neural Network based approach to support the Market Making strategies in High-Frequency Trading

Silva et al. [10] utvecklade ett nätverk som optimerades för aktieprognos över korta tidsintervall, specifikt 5, 10 och 15 minuter. Detta är intressant eftersom man potentiellt kan göra större vinster ju oftare man handlar med aktier. Huvudmålet var att skapa en ANN-modell som kan förutsäga positiva aktiekurs-fluktuationer inom korta tidsintervall.

Silva et al. valde ut 5 aktier från det brasilianska indexet Bovespa Index. BBDC4, BBSA3, USIM5, PETR4 och VALE5. Bovespa Index är enligt Silvia et al. ett index av hög "kvalitet" och innehåller således bara sådana aktier som kan anses vara likvida och mycket viktiga för den brasilianska marknaden.

Aktiekursernas data hämtas i realtid i rå form och omvandlas till, ett mera kornigt, candlestick-format av 5, 10 och 15 minuters intervall. Men om man beaktar 15 minuters intervall följande varandra får man totalt bara fyra stycken varje timme. Silva et al. valde att använda ett glidande tidsfönster, för att gå runt detta, så att nya candlesticks genereras varje minut. Det vill säga givet ett tidsintervall på 15 minuter delas det upp i 15 tidsenheter t_0, t_1, t_2 osv. så att vid t_0 börjar en candlestick som slutar vid t_{14} och vid t_1 börjar en som slutar vid t_{15} . Således fås totalt 45 candlesticks per timme.

Tidsintervallet omnämns tidsfönster, och är tidsskillnaden mellan början och

slutet av en köp/sälj-operation. T.ex. om systemet förutsäger att en aktiekurs kommer få en positiv fluktuation på 7 cent inom 15 minuter är tidsfönstret 15 minuter. Då gör systemet en ny köporder för att köpa denna aktie för det nuvarande värdet och en säljorder för det nuvarande plus 7 cent. Då kan systemet göra en vinst på 7 cent i detta tidsfönster per aktie. Prognosen görs baserat på det nuvarande tidsfönstrets tidsenheter (candlesticks) och sedan kontrollerar nätverket automatiskt, över alla tidsenheter i nästa tidsfönster, om den förutsagda ökningen uppstår. Denna verifikation görs enligt följande formel:

$$var = Max(t_i, t_j) - OpenPrice(t_i)$$

Där t_i är den tidpunkt där operationen börjar, t_j är där den slutar (alltså $t_i +$ tidsfönster) och $Max(t_i, t_j)$ är det högsta kursen aktien uppnått mellan t_i och t_j . $OpenPrice(t_i)$ är öppningskursen för den candlestick som finns vid t_i .

I experimenten varierade systemet tidsfönstret för att avgöra vilken kornighet som passar bäst vid aktiekursprognos.

Silva et al. utvecklade en flerskiktsperceptron med tre lager (input, dolt lager och output) och bakåtpropagering som inlärningsalgoritm. Nätverkets inlärningstakt är 0,3. Som aktiveringsfunktion valdes sigmoidfunktionen. Antalet noder i det dolda lagret definierades enligt följande:

$$dolda\ noder = (attribut + klasser)/2$$

Där attribut är antalet inputs + antalet statiska partiska noder och klasser är antalet outputs.

De valde att använda 14 inputvariabler i deras neurala nätverk, varav 10 är tekniska indikatorer. De övriga fyra är öppningskursen, högsta kursen, lägsta kursen samt stängningskursen för tidpunkten $t_i - TF$, där TF är tidsfönstrets kornighet. Nedan listas alla tekniska indikatorer som användes. Mer information om dem hittas i deras artikel.

1. Relativt styrkeindex (RSI)
2. Glidande medelvärde (SMA)

3. Exponentiellt glidande medelvärde (EMA). Likt glidande medelvärde men med vikter. Större vikt åt senare element.
4. Glidande medelvärde konvergens divergens (MACD).

$$MACD = EMA[12] - EMA[26]$$

$$TRIGGER = MME[9, MACD]$$

MACD är skillnaden mellan två EMAn, ofta en kort och en lång, där de generellt har 12 respektive 26 perioder. TRIGGER är signalen som formas av 9 perioder av MACD.

5. Average Directional Movement Index (ADX).
6. Aroon Indicator.

$$AroonUp = \frac{N - MAX}{N} * 100$$

$$AroonDown = \frac{N - MIN}{N} * 100$$

$$AroonIndicator = AroonUp - AroonDown$$

Aroon Indikatorn beräknar tiden det tar för kursen att nå ett maximum och minimum värde, givet som procent.

7. Bollinger Band. Det finns ett övre band, mellanband, och undre band. Det mellersta bandet är helt enkelt glidande medelvärdet. Det övre är mellanbandet plus standardavvikelsen och det undre är mellanbandet minus standardavvikelsen.
8. Commodity Channel Index (CCI).

$$CCI = \frac{TP - SMA[PT]}{0.015 * SMA[TP]}$$

[Jag antar att TP/PT är samma sak men felstavat på ena stället]. Där TP är medelvärdet av högsta-, lägsta- och stängningskurserna. CCI är en indikator som kan identifiera nya trender eller varna för extrema förhållanden. Generellt sett beräknar CCI det nuvarande kursnivån i förhållande till ett medelvärde över en given tidsperiod.

9. Chande Momentum Oscillator (CMO). Liknar RSI.

$$CMO = 100 * \frac{Upp - Ner}{Upp + Ner}$$

Där upp är antalet dagar där kursens värde ökade och ner är de dagar då kursens värde minskade.

10. Rate of Change (ROC).

$$ROC = \frac{CP - CPA}{CPA}$$

Där CP är den senaste stängningskursen och CPA är ett medelvärde av ett givet antal stängningskurser.

Eftersom det handlar om intradayshandel tränas det neurala nätverket först med halva dagens data och sedan körs test på andra halvan, med tanken att nätverket alltid ska vara så uppdaterat som möjligt vad gäller aktiekursens beteende.

3.3.1 Slutsats

I slutändan kom Silva et al. fram till att ett tidsfönster på 5 minuter är det mest effektiva för deras system. Med 5 minuters tidsfönster uppnåddes en medelprecision på 77,6 % för BBAS3; 73,53 % för BBDC4; 76,12 % för PETR4; 83,86 % för USIM5 och 71,75 % för VALE5. Med 15 minuters tidsfönster uppnåddes ungefär 10 % lägre precision för alla aktier och 10 minuters tidsfönster var lite bättre.

Silva m.fl. föreslår likt Marwala och Patel att man kunde optimera det artificiella neurala nätverkets parametrar för bättre resultat [Genetiska algoritmer?].

Det framgår tydligt att detta system uppnår högre precision än Marwala och Patels system. En orsak till detta kunde vara att Silva et al. använde bättre input-data. En annan orsak kunde vara att Marwala och Patel ger sitt ANN för svår uppgift, Silvas systems uppgift är bara att förutsäga om det kommer en positiv fluktuation inom en kort tidsram medan M. och P. vill veta hur stor fluktuationen är och i vilken riktning den går en hel dag framåt. Eftersom den kortare tidsramen verkar ge bättre resultat kan man spekulera att det betyder att

deras nätverk lärt sig förutsäga hur aktier köps och säljs eftersom volym direkt påverkar kursen, speciellt på kort sikt [3].

3.4 Stock Market Forecasting Algorithm Based on Improved Neural Network

Luo, Chen och Jiang [18] utvecklar och jämför en MLP som använder en kombination av genetiska algoritmer och bakåtpropagering med en vanlig MLP med bakåtpropagering.

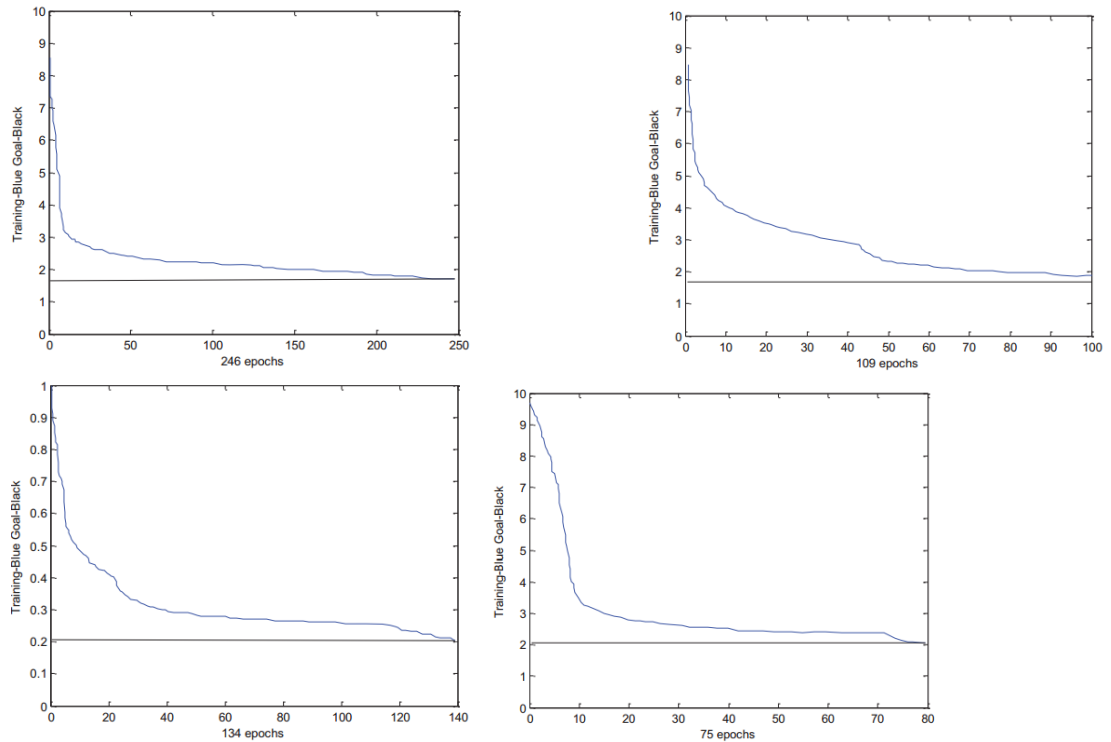
Luo et al. valde Shanghai Index som tränings- och testdata. Som input valdes öppningskursen, högsta kursen, lägsta kursen, stängningskursen, volymen för de tre senaste dagarna, ett 5 dagars glidande medelvärde samt två tidsserier, en baserad på kursen och en baserad på volymen. Det framgår inte tydligt hur många dolda lager det finns i deras MLP:n men de pratar om "det dolda lagret" så vi kan ganska lugnt anta att det bara finns ett. Antalet noder i det dolda lagret är 20, och det värdet kom de fram till genom att pröva sig fram. Nätverket har ett output lager med 6 noder som ger en prognos för öppningskursen, högsta kursen, lägsta kursen, stängningskursen, volymen och ett 5 dagars glidande medelvärde som output för följande dag.

Vi kan anta att de använder en tröskelvärdesfunktion som aktiveringsfunktion eftersom de nämner tröskelvärden: "Each individual contains all the weights and thresholds of the neural network...".

En population av nätverk skapas där alla individers vikter och tröskelvärden anses vara deras gener. Först tränas nätverken med hjälp av bakåtpropagering på träningsdatan och sedan får varje nätverk ett "lämplighetsvärde" som baseras på hur bra nätverket i dess nuvarande form presterar. Individerna som "paras" väljs enligt deras lämplighetsvärde så att de som har störst lämplighetsvärde har störst chans att bli valda, enligt

$$P_i = \frac{f_i}{\sum f}$$

Där P_i är sannolikheten att element i kommer väljas, f_i är element i s lämplighetsvärde och $\sum f$ är summan av alla lämplighetsvärden i hela populationen.



Figur 8: I den vänstra spalten visas inlärningstiderna (mätt i "epoker") för nätverken då applicerade på China Telecom och till höger då applicerade på Broad Group. Den övre raden är MLP:ns inlärning och den undre raden är GA-MLP:ns inlärning. Den vanliga MLP:n konvergerade efter 246 epoker då applicerad på China Telecom och efter 109 epoker då applicerad på Broad Group. Deras GA-MLP konvergerade efter 134, respektive 75 epoker. Figurerna är tagna från deras artikel.

Sedan görs en korsning av elementens gener (plus en 9 % chans för mutation). Selektion och parningsstegen upprepas tills träningsmålet nås eller tills antalet iterationer når en given gräns.

För att verifiera deras resultat jämför de sitt nätverk med en MLP som tar öppningskursen, högsta kursen, lägsta kursen, stängningskursen och volymen för Shanghai Index som input [Notera att detta inte är samma input som i deras GA-MLP, förutsatt de inte bara varit slarviga i sitt skrivande]. Som testdata användes China Telecom och Broad Group.

Som det framgår i figurerna konvergerade GA-MLP:ns inlärning mycket tidigare än den vanliga MLP:n trots att den vanliga MLP:n hade färre inputs.

Precisionen för nätverken mättes inte på samma sätt som tidigare artiklar där de helt enkelt mätte träffsäkerheten utan här mäter de istället hur stort felet i förutsägelseerna är, men vi kan fortfarande jämföra resultaten inom artikeln.

MLP:ns fel för China Telecom som följer: öppningskursen 1,51 %; högsta kursen 3,08 %; lägsta kursen 3,17 %; stängningskursen 4,12 %. GA-MLP:ns fel för China Telecom som följer: öppningskursen 1,16 %; högsta kursen 2,52 %; lägsta kursen 2,55 %; stängningskursen 3,03 %. MLP:ns fel för Broad Group som följer: öppningskursen 0,45 %; högsta kursen 1,86 %; lägsta kursen 1,02 %; stängningskursen 1,71 %. GA-MLP:ns fel för Broad Group som följer: öppningskursen 0,36 %; högsta kursen 1,45 %; lägsta kursen 0,91 %; stängningskursen 1,59 %.

3.4.1 Slutsats

Den MLP som använder genetiska algoritmer lär sig alltså snabbare och ger bättre förutsägingar. Skillnaden i prediktion kanske vore mindre om man använde samma inputs för båda nätverken men GA-MLP:n lärde sig i vilket fall som helst snabbare än MLP:n. GA-MLP:n borde teoretiskt sett heller inte fastna i lokala minimum som MLP:n vilket tyder på att den borde kunna nå högre precision.

4 Sammanfattning

Efter att ha jämfört RBF-nätverk, MLP och GA-MLP kan vi med ganska stort förtroende säga att MLP:n uppnår högre precision än RBF-nätverket för att göra prognos av aktiekurser. Det finns dock ännu för få tillgängliga artiklar för att definitivt säga om det är GA-MLP:n presterar bättre än vanliga MLP:n som bara använder bakåtpropagering. Vi kan dock konstatera att den artikel som undersöktes där de jämför en GA-MLP med en MLP tyder på att GA-MLP:n skulle göra bättre prognos, vilket teorin också understryker. Tekniska indikatorer verkar också vara en bra idé att använda som input. Det vi med säkerhet kan konstatera är att artificiella neurala nätverk, rätt implementerade, kan tillämpas för att göra prognos av aktiekurser till en sån precision att det vore lönsamt.

Vidare vore det intressant att forska mera i att kombinera artificiella neurala nätverk med genetiska algoritmer och olika kombinationer av tekniska indikatorer.

References

- [1] P. B. Patel and T. Marwala, "Forecasting closing price indices using neural networks," 2006 IEEE International Conference on Systems, Man and Cybernetics, Taipei, 2006, pp. 2351-2356.
- [2] Bell, Jason. Machine Learning: Hands-On for Developers and Technical Professionals (1). Somerset, US: Wiley, 2014. ProQuest ebrary. Web. 8 March 2017.
- [3] M. Hallström. . Teknisk Analys Glidande Medelvärden [blogginlägg]. 03-04-2017 från <https://www.swedbank-aktiellt.se/blogg/hallstrom/jeejxi.csp>
- [4] M. A. A. Ibrahim and K. Raahemifar, "Beating the zero-sum game using GA to optimize technical financial market indicators," 2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), Vancouver, BC, 2016, pp. 1-6.
- [5] C. M. Bishop. Neural Networks for Pattern Recognition. Oxford: Clarendon, 1995.
- [6] R. E. Uhrig, "Introduction to artificial neural networks," Industrial Electronics, Control, and Instrumentation, 1995., Proceedings of the 1995 IEEE IECON 21st International Conference on, Orlando, FL, 1995, pp. 33-37 vol.1.
- [7] | A. P. N. Refenes, A. N. Burgess and Y. Bentz, "Neural networks in financial engineering: a study in methodology," in IEEE Transactions on Neural Networks, vol. 8, no. 6, pp. 1222-1267, Nov 1997.
- [8] N. Baba, N. Inoue and H. Asakawa, "Utilization of neural networks and GAs for constructing reliable decision support systems to deal stocks," Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, Como, 2000, pp. 111-116 vol.5.
- [9] F. E. R. Mattioli, E. A. Lamounier, A. Cardoso, A. B. Soares and A. O. Andrade, "Classification of EMG signals using artificial neural networks for

virtual hand prosthesis control," 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Boston, MA, 2011, pp. 7254-7257.

[10] E. Silva, D. Castilho, A. Pereira and H. Brandao, "A neural network based approach to support the Market Making strategies in High-Frequency Trading," 2014 International Joint Conference on Neural Networks (IJCNN), Beijing, 2014, pp. 845-852.

[11] Reeves, Colin R. Rowe, Jonathan E.. 2002. Genetic Algorithms - Principles and Perspectives. Secaucus: Kluwer Academic Publishers.

[12] M. Usmani, S. H. Adil, K. Raza and S. S. A. Ali, "Stock market prediction using machine learning techniques," 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, 2016, pp. 322-327.

[13] Chris McCormick. . Radial Basis Function Network Tutorial [blogginlägg]. Hämtat 03-04-2017 från <http://mccormickml.com/2013/08/15/radial-basis-function-network-rbfn-tutorial/>

[14] Broomhead, D. S.; Lowe, David (1988). Hämtat från www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA196234

[15] Teslas hemsida. https://www.tesla.com/sv_SE/autopilot

[16] Ganesan, N. "Application of Neural Networks in Diagnosing Cancer Disease Using Demographic Data". International Journal of Computer Applications. <http://www.ijcaonline.org/journal/number26/pxc387783.pdf>

[17] Forecasting Arima Models, <https://support.sas.com/documentation/cdl/en/etsug/6393>

[18] B. Luo, Y. Chen and W. Jiang, "Stock Market Forecasting Algorithm Based on Improved Neural Network," 2016 Eighth International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Macau, 2016, pp. 628-631.