HÖGSKOLAN SKÖVDE

Institutionen för kommunikation och information
Examensarbete i datalogi 20p (kursnamn+poäng)
C-nivå
Vårterminen 2007

Artificiella neuronnät & biometri

-verifiering utav användare via tangentbordsskrivning

Eddie Ehlin

Artificiella neuronnät & biometri

Examensrapport inlämnad av Eddie Ehlin till Högskolan i Skövde, för Kandidatexamen (B.Sc.) vid Institutionen för kommunikation och information. Arbetet har handletts av Fredrik Johansson.

Datum

Härmed intygas att allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och att inget material är inkluderat som tidigare använts för erhållande av annan examen.

Signerat: _____

Artificiella neuronnät & biometri

Eddie Ehlin

Sammanfattning

Detta arbete handlar om beteendeinriktad biometri och artificiella neuronnät av typen feedforward och hur de tillsammans kan användas för att verifiera användare. Det har av tidigare arbete bekräftats att det är möjligt att verifiera användare, men tidigare resultat har däremot inte utfört tester med avseende på avvikelser i data (beteende) och dess inverkan på verifieringen. Det är detta som utgör det huvudsakliga målet för detta arbete, nämligen att undersöka hur avvikelser i data påverkar verifiering och utifrån det också undersöka neuronnätens noggrannhet vid verifiering.

Nyckelord: Artificiella neuronnät, biometri, beteendeinriktad biometri, tangentbordsskrivning, feedforward, identitetsverifiering och backpropagation.

Innehållsförteckning

1 Introduktion

Allt fler datorsystem förs in i världen och det i sin tur för med sig allt fler resurser som kan behöva skydd. En resurs, exempelvis en personlig handlig (dokument), skyddades tidigare genom att resursen hölls inlåst (fysiskt). Tillträde till resursen kunde då endast erhållas genom att gå via de (personer) som ansvarade för den. Sättet att gå tillväga för att erhålla tillträde till resursen skiljer sig inte särskilt mycket från idag, förutom på en punkt, nämligen det att personalen som förr skyddade resurser om möjligt ersatts av datorer.
Säkerhetspersonal har sina rutiner att följa, och skulle det uppstå situationer där säkerhetspersonalens rutiner ej räcker till kan säkerhetspersonalen förmodligen agera på eget initiativ. Just det att agera på eget initiativ är något som ej, idag, är möjligt för en dator och det är därför mycket viktigt att så gott det går försöka undvika situationer där datorns rutiner ej räcker till.

Eftersom att en dator ej har möjligheten att agera på eget initiativ så har det lett till att det inom området säkerhet har tagits fram en rad olika tekniker och tillvägagångssätt för att skydda resurser. Den mest frekvent använda tekniken för att med hjälp av en dator skydda en resurs är att använda lösenord, eventuellt i kombination med ett användarnamn. Tekniken med lösenord används flitigt än idag och tack vare den långa tid som detta tillvägagångssätt har använts så medför det ett väl testat och beprövat skydd för en resurs. Tekniken för att skydda en resurs med lösenord har med tiden utökats med en rad olika förbättringar, varav kryptering och tillfälliga lösenord är två utökningar som idag är mycket vanliga.

Utöver de utökningar som gjorts på tekniken för användandet av lösenord så har det också tillkommit nya tekniker inom säkerhetsområdet. Ett av dessa områden är biometri, vilket är ett relativt stort område innehållandes en rad olika tekniker som alla, i någon mån, är baserade på fysiska- eller beteendeegenskaper hos människor. Tillvägagångssättet är likt den teknik som utnyttjar lösenord, och den största skillnaden är att lösenord har bytts ut mot biometrisk data.

Detta arbete behandlar en beteendeinriktad teknik inom området för biometri. Fokus är personers beteende vid textinmatning på tangentbord (tangentbordsskrivning). Biometrisk data av denna typ har en tendens till att innehålla avvikelser, eftersom en person sällan skriver helt identiskt. För att råda bot på det så tillämpas artificiella neuronnät på dessa biometriska data. Enligt Callan (1998), Rogers (1996) och Russell & Norvig (2002) så lämpar sig just artificiella neuronnät för bearbetning/analys av denna typ av data. Just därför att artificiella neuronnät har en förmåga att känna igen mönster, så kan avvikelsernas inverkan eventuellt förminskas. På så vis kan vi erhålla en robustare miljö för igenkänning av personer utifrån deras beteende. Olika typer av artificiella neuronnät som kan användas för detta ändamål torde vara modulära, rekurrenta, neuronnät med radialbasfunktioner eller neuronnät av typen feedforward med Backpropagation. Feedforward är den typ utav artificiella neuronnät som används i detta arbete, eftersom denna typ användes av Rogers (1996) vars arbete står som grund samt utgångspunkt för detta arbete. Syftet med arbetet är att klarlägga hur den utvalda typen av artificiellt neuronnät hanterar avvikelser i data, vilket inte har framgått enligt Rogers (1996). Detta utförs genom implementation av en prototyp för

verifiering av användare. Experiment utförs genom att utsätta prototypen (neuronnäten) för olika sorters data, för att framhäva dess egenskaper vad det gäller hanterandet av bl.a. avvikelser i data.

2 Bakgrund

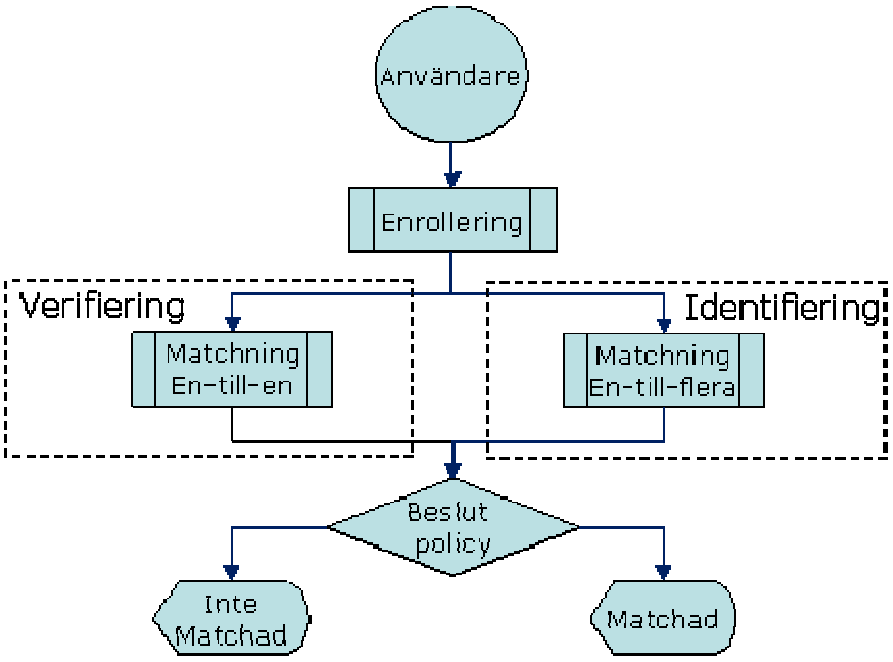
Det finns idag en rad olika sätt för att identifiera användare vid en fysisk inlogging. Med fysisk inlogging menas att användaren har fysisk tillgång till inloggningsterminalen, exempelvis tangentbord eller fingeravtrycksläsare. Olika tillvägagångssätt för identifiering av fysisk inloggning kommer att presenteras i detta kapitel, detta för att ge en bild av hur fysiska inloggningssystem fungerar. Utöver detta kommer viktiga definitioner, begrepp och tekniker (främst den som arbetet tillämpar) att tas upp.

Sektionerna 2.1 och 2.2 samt dess undersektioner är baserat på tidigare material, nämligen: (Internationella biometrigruppen, 2007), (Monrose & Rubin, 1997), (Demir, 2002), (Svenska biometriföreningen) och (Joyce & Gupta, 1990)

2.1 Identitetsverifiering och identifiering

> ”I det vardagliga livet så *verifieras* din identitet av de flesta människor som du gör affärer med eller träffar på. Du påstår dig vara någon och sedan försöker du bevisa din påstådda identitet. Däremot när man träffar på sin familj och bekanta så behöver man inte tala om vem man är först, istället är det dina bekanta som *identifierar* dig genom att se ditt ansikte eller höra din röst.” (Demir, 2002)

Enligt (Svenska biometriföreningen) framgår det att biometritekniker generellt kan användas för två syften: *verifiering* och *identifiering*, vilka kommer att beskrivas i denna sektion (2.1) tillsammans med viktiga begrepp inom området biometri.



Figur 1 Biometriteknikers syften (Svenska biometriföreningen)

Enrollering är, enligt (Svenska biometriföreningen), när en digital kopia av individens biometriska information skapas. Den digitala kopian kallas för en template (alternativt användarprofil) och kan lagras antingen lokalt på en dator, en server eller på ett smart kort.

Identitetsverifiering innebär att när en användare försöker logga in så utger sig denne för att vara någon användare i systemet. Efter det att användaren angett sin påstådda identitet tar inloggningssystemet direkt upp den användarprofil som matchar den angivna (en-till-en relation mellan användarprofiler, se Figur 2). Den öppnade användarprofilen används sedan för jämförelse med den påstådda identiteten. Visar det sig att användarprofilen stämmer överrens med den påstådda erhåller användaren tillgång till systemet.

Figur 2 En-till-en relation (Svenska biometriföreningen)

Värt att tillägga här är att när en användare loggar in så utger denne sig för att vara en användare till systemet och tack vare detta kan den tänkta användarprofilen snabbt hämtas (en-till-en relation). Detta leder till att inloggningssystemet blir snabbare, eftersom inloggningssystemet inte behöver göra sökningar efter användarprofiler.

Identifiering är inte ett hypotestest[1] vilket identitetsverifiering är, utan nu erhåller användaren sin identitet utav inloggningssystemet. Användaren frågar alltså inloggningssystemet "Vem är jag?", och inloggningssystemet påbörjar då sin sökning bland dess användarprofiler och returnerar den användarprofil som hör till användaren, om den nu existerar (se Figur 3).

Figur 3 En-till-många relation (Svenska biometriföreningen)

Identifieringssystem har alltså inte samma möjlighet att avgränsa sitt sökområde och därmed så kan dessa system ta betydligt längre tid på sig för att identifiera en användare. Utöver det att identifieringssystem kan ta längre tid så kan dessa lida av att felaktiga matchningar kan uppstå, just därför att en större mängd användarprofiler måste gås igenom och jämföras. Detta på grund av att en en-till-en relation ej direkt kan upprättas, och därmed så blir avgränsningen av sökområdet svårare i ett identifieringssystem.

[1] Hypotestest, ett påstående som antingen är sant eller falskt.

2.2 Biometri

> "The automated use of physiological or behavioral characteristics to determine or verify identity." (Internationella biometrigruppen, 2007)

Biometri i sig innebär att något levande mäts, och utifrån definitionen ovan så kan biometri delas in i två kategorier, nämligen:

- Fysiologisk
- Beteendeinriktad

Inom dessa två kategorier beskrivs i följande sektioner olika metoder för mätning. Detta för att ge en bild av vad som skiljer kategorierna åt. Men först beskrivs biometri i sin helhet på ett översiktligt sätt.

Identitetsverifikation baserad på biometri är en strukturerad och maskinell kontroll utav användarens identitet. Genom antingen människokroppens (användarens) fysiologiska egenskaper så som iris, fingeravtryck, näthinna, handavtryck eller användares beteende som till exempel användarens röst. Skillnaden mellan beteendeinriktad biometri och den fysiologiska biometrin är vag, eftersom de beteendeinriktade egenskaperna delvis är baserade på fysiologiska egenskaper. Se nedan för ett exempel.
Hur en användares röst låter beror huvudsakligen på två saker, nämligen:

- Hur användarens stämband är, fysiologiskt sett.
- Vilket tonläge användaren väljer att använda (beteendeinriktat).

Biometriska identitetsverifikationssystem som är baserade på beteende har en tendens till att utsättas för mer variationer i mätdata jämfört med fysiologiska identitetsverifikationssystem. Anledningen till det är att fysiologiska egenskaper ofta är mer statiska, en människas näthinna är något som sällan ändras, medan en människas röst är något som kan ändras och påverkas av flera orsaker, så som: humör eller sjukdom/skada.

Utöver de fysiologiska faktorer en människas röst har så kan ljudet påverkas av omgivningen och utrustningen. Det medför att utrustningen för analys av användaren också måste bete sig korrekt och i den mån det är möjligt ta hänsyn till dessa faktorer som kan påverka röstens karaktär och själva ljudet.

Nedan följer det nu två sektioner som var för sig kommer att ta upp lite kortfattade exempel på olika metoder inom de båda kategorierna som nämndes ovan.

2.2.1 Fingeravtryckskläsning, en fysiologisk metod

Undersökningar som gjorts har visat att fingeravtryck är mycket unika. Enligt internationella biometrigruppen (2007) finns det undersökningar som visar att sannolikheten för att två människor har identiska fingeravtryck är mindre än en på miljarden.

Utöver det att sannolikheten för felmatchning är låg så är fingeravtryck stabila i den bemärkelsen att de inte ändras (förutsatt att ingen skada sker). Ett exempel på hur fingeravtrycksläsning fungerar kommer nu att gås igenom, följt av eventuella fördelar och nackdelar med denna metod.

Låt oss säga att vi skall logga in på företagets huvuddator, och för att kunna logga in där måste ens identitet bekräftas genom att ens fingeravtryck läses av och matchas mot en giltig systemanvändares fingeravtryck. Företaget har i detta fall utnyttjat att det finns två tekniker för att läsa av ett fingeravtryck, nämligen:

- Via en optisk sensor.
- Via en kapacitiv sensor.

Detta för att dra nytta av den stabila fysiologiska egenskapen hos människor. Fingret sätts nu mot avläsningsyta nummer ett, där optisk avläsning av fingeravtrycket sker. Efter det att första avläsningen är gjord sätts fingret på avläsningsyta nummer två, där avläsning av elektriska strömmar mellan dalar och höjder på fingeravtrycket sker.
Inloggningssystemet söker nu efter en matchande användarprofil, med hjälp av vårt nyligen avlästa fingeravtryck. I tabell 1 följer fördelar och nackdelar med de två olika teknikerna som nämnts ovan:

Egenskap	Optisk	Kapacitiv
Billig	Ja	Nej
Feltolerant	Ja	Okänt
Säker	Medel	Ja

Tabell 1 Jämförelse mellan optisk och kapacitv avläsning av fingeravtryck.

2.2.2 Tangentbordsstatistik, en beteendeinriktad metod

> "Tangentbordsstatistik (keystroke dynamics) är processen där man analyserar sättet en användare skriver i en terminal genom att mäta tangentbordets inmatningar i millisekunder eller mikrosekunder i ett försök att verifiera användare baserad på den vanemässiga skrivrytmens mönster." (Demir, 2002)

Att föra statistik över hur användare beter sig då de skriver vid ett tangentbord är något som det skrivits en hel del om. Arbeten har gjorts och även en hel del forskning har lagts ned på hur verifiering av en användare kan gå till men också för att se ifall metoden är tillräckligt säker för att användas för verifiering av användare.

Enligt Rogers (1996), Joyce & Gupta (1990) och Monrose & Rubin (1997) har det visat sig att denna form av statistik och analys utav användaren är ett bra sätt att gå tillväga för att verifiera en användare. Verifiering av användare via tangentbordsskrivning är dessutom ett alternativ som i stort sett inte innebär några kostnader, eftersom det enda som behövs är ett tangentbord. Andra biometriska identitetsverifikationssystem kräver ofta att ny hårdvara införskaffas, vilket gör de mer kostsamma vid införandet.

Tack vare den låga kostnaden för att verifiera användare via tangentbordsskrivning, så kan denna teknik vara lämplig att kombinera med andra biometriska verifieringstekniker. En kombination av olika sådana tekniker kallas multi-modala system, vilket tas upp i sektion 2.2.3.

Hur verifieringen går till beskrivs i efterföljande stycken och precis som tidigare görs detta genom ett exempel. Låt oss säga att vi återigen skall logga in på företagets huvuddator, nu med ett helt annorlunda och nytt inloggningssystem där verifiering av användare sker via tangentbordsskrivning. Vi sätter oss vid inloggningsterminalen, där datorn sedan ber oss att skriva in en textsträng. Textsträngen som skall skrivas in kan antingen vara angiven i förväg eller så kan en godtycklig textsträng få anges, detta beror på implementationen av inloggningssystemet. Under tiden vi skriver in textsträngen så analyserar datorn vårt beteende och sammanställer sedan en profil efter det att vi skrivit in hela textsträngen. Användarprofilen som genererats av inloggningssystemets analys används sedan för att matcha systemets användarprofiler för att se ifall vi har behörighet att logga in vid denna terminal. Ett verifieringssystem via tangentbordsskrivning kan antingen göras som ett identitetsverifieringssystem eller som ett identifieringssystem (se sektion 2.1), det vill säga att en användare antingen anger en identitet i förväg som sedan skall verifieras eller så får inloggningssystemet säga vem användaren är.

Inloggningssystemet mäter tiden det tar då en tangent trycks ned och tills det att samma tangent släpps upp, dessutom mäts tiden det tar innan nästa tangent trycks ned och sedan görs samma mätning för den.

Låt oss säga att vi skall mata in textsträngen "foo". Inloggningssystemet startar tidtagningen då tangenten ”f” trycks ned och när tangenten ”f” släpps upp så stoppas den tidtagningen. Samtidigt som den tidtagningen stoppas så startar en annan tidtagning, nämligen den som används för att ta reda på hur lång fördröjning det är mellan det att tangenten ”f” släppts upp och att tangenten ”o” tryckts ned. Detta görs tills slutet av textsträngen är nådd och alla tider som erhållits används sedan för att matcha inloggningssystemets lagrade användarprofiler.

Det tillvägagångssätt som beskrivits i exemplet ovan är av typen statiskt, det vill säga att analysen av användaren är förutbestämd i rum och tid, exempelvis vid inloggningen. Statisk verifiering ger en stark och robust verifiering av användare men den lider också av att inte ha kontinuerlig säkerhet eftersom en användare exempelvis kan glömma att logga ut ur systemet, det vill säga att statisk verifiering av användare inte kan hantera/upptäcka byte av användare.

Dynamisk verifiering analyserar och övervakar däremot användaren under hela tiden som användaren är inloggad; antingen fram till det att användaren explicit loggar ut ur systemet eller det att användarens beteendemönster ändras, vilket leder till att användaren loggas ut ur systemet.

Sektionen avrundas nedan med en jämförelse mellan de två identitetsverifikationssystem som tagits upp, nämligen fingeravtrycksläsning och tangentbordsskrivning.

Egenskap	Tangentbordsskrivning	Fingeravtrycksläsning
Billig	Ja	Nej
Kräver ny hårdvara	Nej	Ja
Säker	Se kap. 6	Ja

Tabell 2 Jämförelse mellan verifiering av användare via tangentbordsskrivning och fingeravtrycksläsning.

2.2.3 Multi-modala system

Det har visat sig att biometriska verifieringssystem var för sig inte är pålitliga till 100%. Även med en sannolikhet för exempelvis två identiska fingeravtryck som är mindre än en på miljarden (se sektion 2.2.1), så är det möjligt att två identiska fingeravtryck kan uppstå från två unika användare. Genom att kombinera biometriska verifieringssystem med andra biometriska verifieringssystem så kan säkerheten förbättras.

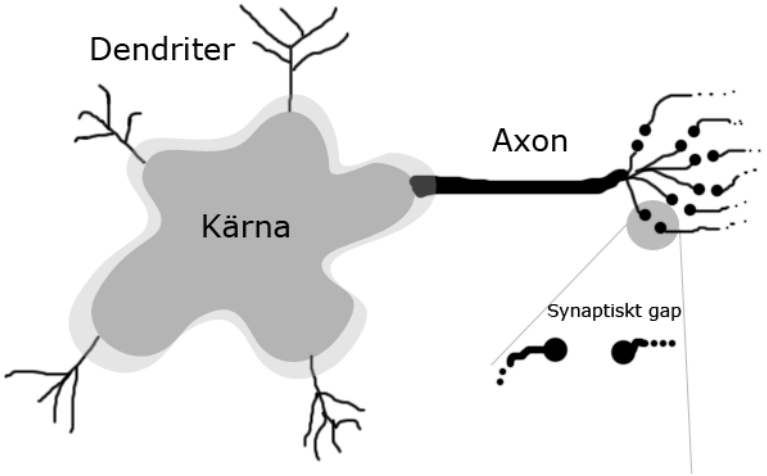
Genom att kombinera flera verifieringssystem så ökar säkerheten men det leder också till att användarvänligheten blir sämre och att allt fler faktorer/parametrar måste stämma in för att erhålla tillgång till systemet. Detta kan vara både till fördel och även nackdel.

2.3 Artificiella neuronnät

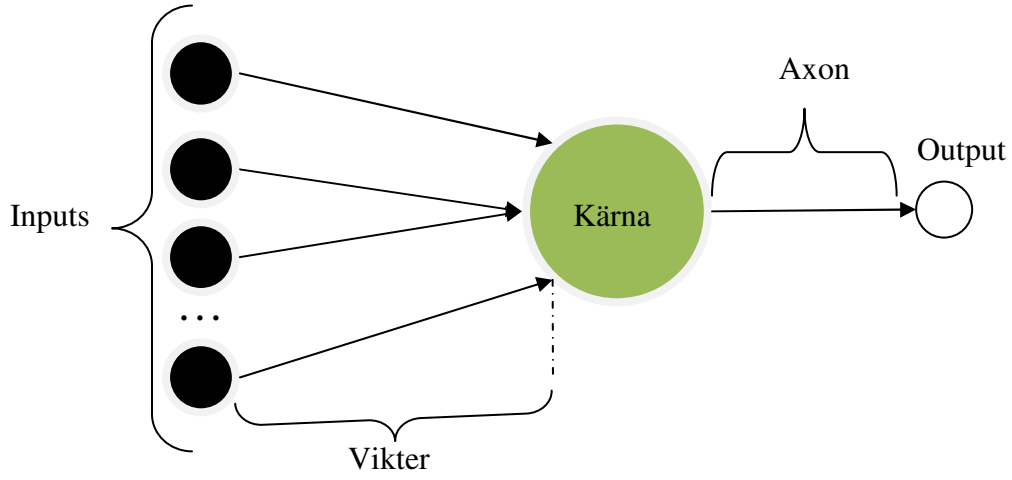
Syftet med denna sektion är att successivt bygga upp en förståelse för artificiella neuronnät, dock bara de detaljer som anses vara nödvändiga för att förstå den teknik som används i detta arbete. För ytterligare material angående artificiella neuronnät se exempelvis (Callan, 1998).

2.3.1 Artificiella neuronens beståndsdelar

Artificiella neuronens uppbyggnad och dess beståndsdelar påminner mycket om den biologiska neuronen. Den artificiella neuronen däremot är en förenkling av den biologiska, vilket framgår av figurerna som följer (Figur 4 & Figur 5) nedan:



Figur 4 Biologisk neuron (förenklad).



Figur 5 Artificiell neuron

För att framhäva den artificiella neuronens likheter vad det gäller uppbyggnad och beståndsdelar så följer det nedan en beskrivning av de beståndsdelar en artificiell neuron är uppbyggd av tillsammans med en mappning mot den biologiska neuronen.

Inputs är den artificiella neuronens inkommande anslutningar från andra neuroner medan de biologiska neuronerna ansluter till varann via så kallade dendriter[2].

Vikter beskriver hur stark en artificiell neurons anslutning är till en annan neuron. En stark anslutning från neuronen α till neuronen β innebär att α har möjligheten att påverka β, kraftigt. En stark anslutning mellan två biologiska neuroner innebär att anslutningens synaptiska gap är kort, vilket leder till bra ledningsförmåga av signal.

Kärnan hos en artificiell neuron är, enkelt uttryckt, en beräknings-/insamlingsenhet som i sig består av en aktiveringsfunktion. Biologiska neuronens kärna har även den i uppgift att samla ingående signaler och beräkna.

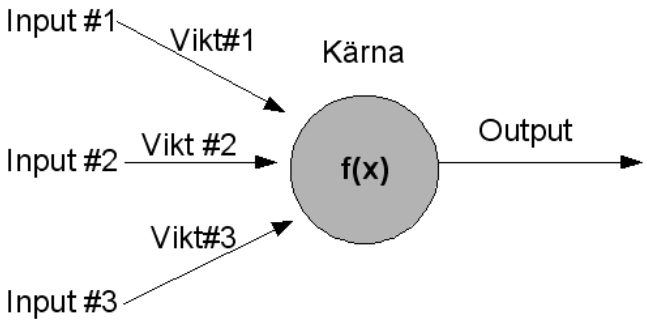
Output är artificiella neuronens utgående anslutning, via vilket den har möjlighet att påverka andra neuroner som den har anslutning till. Neuronens utgående signal överförs via dess s.k. axon och detsamma gäller den biologiska neuronen.
Artificiella neuronens beståndsdelar är:

Inputs, vikter och output kommer att tas upp i sektion 2.3.2 och beskrivas mer ingående där. Detta för att på ett enkelt sätt fokusera på just den enskilda artificiella neuronen i denna sektion.

[2] En dendrit är den del av neuroner eller nervceller som leder nervsignaler in till cellen.

Den artificiella neuronens kärna är, som det tidigare sagts, en beräknings-/insamlingsenhet och dess konstruktion är enkel. Det kärnan gör är att samla de ingående signalerna och sedan addera dem. Resultatet (summan) av de adderade signalerna används sedan för att se ifall tillräcklig så kallad input finns. Visar det sig att resultatet av de adderade signalerna befinner sig över ett visst tröskelvärde så avfyrar neuronen en signal via sin output (axon).

Ett exempel på hur en neuron tar in signaler och avgör ifall någon signal skall skickas kommer nu att gås igenom, exemplet kommer att utgå från Figur 6.

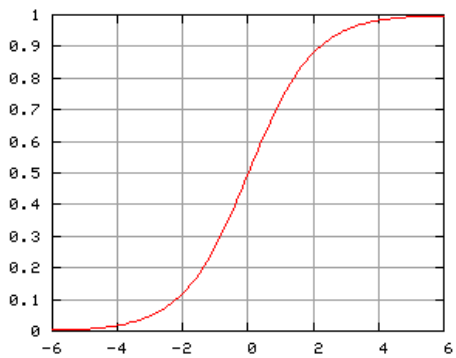


Figur 6 En neuron med tre inkommande anslutningar.

Enligt Figur 6 framgår det att tre stycken neuroner har koppling till denna neuron och därmed har de också möjligheten att påverka neuronen i figuren. Påverkan sker genom att en signal skickas. Låt oss nu säga att dessa tre neuroner avfyrar varsin signal och det är dags för neuronen i figuren att samla in dessa signaler, vilket görs på följande vis:

$$\sum_{i=1}^{3} Input_i Vikt_i$$

Resultatet av denna summering används sedan för att erhålla neuronens utgående signalstyrka, vilket görs genom att neuronens aktiveringsfunktion appliceras på den erhållna summan. Varje neuron har en så kallad aktiveringsfunktion (*f(x)* i figuren ovan). Det finns ett antal olika aktiveringsfunktioner och den som tas upp här är aktiveringsfunktionen vid namnet sigmoid (se Figur 7).



Figur 7 Aktiveringsfunktionen sigmoid: $\boldsymbol{sig(x) = \frac{1}{1+e^{-x}}}$

Enligt Figur 7 framgår det att sigmoid är en deriverbar funktion. Detta har visat sig vara en viktig egenskap för nätets möjligheter till inlärning, enligt bl.a. Russell & Norvig (2002) och Rogers (1996).

Låt oss nu anta att vår summa erhöll värdet 2.3 efter det att vi adderat de ingående signalerna, enligt summeringen ovan. Neuronens utgående signal bestäms nu genom:

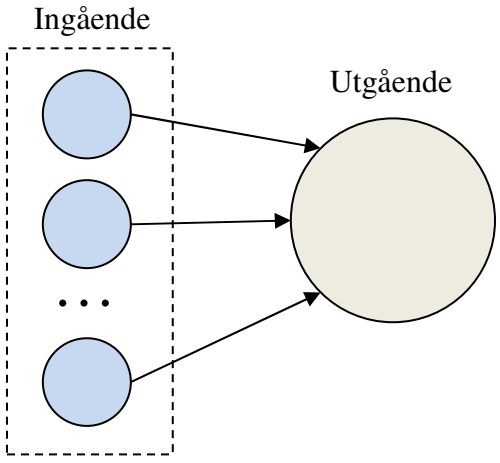
$$sig(2.3) = \frac{1}{1 + e^{-2.3}}$$

Neuronens utgående signal blir ~0.91 enligt aktiveringsfunktionen sigmoid. Det som nu gåtts igenom är precis vad en neurons enda syssla är. Som det sades tidigare så är en neuron en enkel insamlings-/beräkningsenhet och en ensam neuron kan inte åstadkomma särskilt mycket. Däremot kan mycket intressanta saker åstadkommas ifall flera neuroner kopplas samman, vilket tas upp i nästa sektion (2.3.2).

2.3.2 Feedforward-nätverket, ett neuronnät

> "A neural network is a collection of simple processing units which sends signals to one another along the weights." (Callan, 1998)

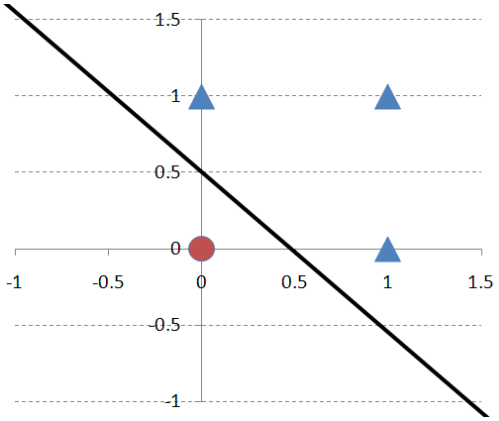
Denna typ av neuronnät, även kallat Backpropagation neural network (BPNN), lades först fram av Werbos (1974) och senare utav Parker (1985) och Rumelhart & McClelland (1986). Ett neuronnät av typen feedforward (BPNN) är egentligen en utökning av ett neuronnät vid namnet ”Single layer perceptron” (SLP), vilket är ett neuronnät innehållandes endast *två* lager varav det sista lagret innehåller endast en neuron, och det kan se ut som följande:



Figur 8 Ett neuronnät av typen SLP.

Ett neuronnät av denna typ (SLP) har, enligt bl.a. (Minsky & Papert, 1969), endast möjligheten att lösa problem som är linjärt separerbara, det vill säga att problemrymden innehåller två klasser, exempelvis logiska operatorer så som OR och

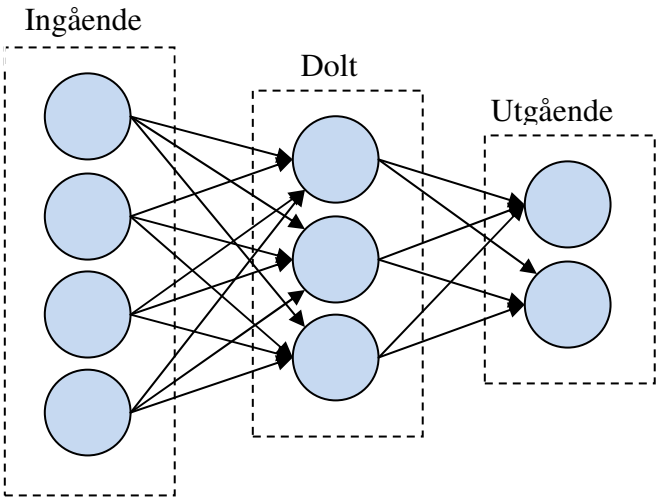
AND. Se figur nedan för en problemrymd för den logiska operatorn OR, där det framgår tydligt att endast två klasser finns.



Figur 9 Ett linjärt separerbart problem (Logiska operatorn OR)

För att neuronnät skall ha möjligheten att lösa problem som ej är linjärt separerbara, det vill säga att det är inte möjligt att med en rät linje i problemrymden separera samtliga klasser, så behövs fler lager. Detta var självklart redan då Minsky & Papert (1969) presenterade sitt matematiska bevis för att ett SLP inte kunde lösa annat än linjärt separerbara problem. Det som däremot inte var självklart var tillvägagångssättet för att träna (se sektion 2.3.3) ett neuronnät innehållandes ett eller fler lager mellan det ingående och utgående lagret.

En algoritm vid namnet Backpropgation löste det ovan nämnda problemet, att träna neuronnät innehållandes flera lager. Algoritmen lades först fram av Werbos (1974), vilket sedan vidareutvecklades av en rad olika personer. Tack vare denna algoritm kan nu artificiella neuronnät appliceras på nya typer av problem, och det är Backpropagation och flerlagriga neuronnät som utgör en central bit i detta arbete. Eftersom detta arbete, till viss del, innebär att återupprepa ett tidigare arbete utfört av Rogers (1996). Nedan följer nu mer ingående information angående den typ av neuronnät (se Figur 10) som i detta arbete kommer att användas.



Figur 10 Ett feedforward-nätverk innehållandes tre lager.

I Figur 10 framgår det att olika lager av neuroner finns. Dessa lager skiljer sig något från varandra och nedan följer det en kortfattad beskrivning för varje lagertyp. Det skall dock tilläggas att neuronerna i lagren är identiska i bemärkelsen att alla neuroner har input, output och en aktiveringsfunktion. Däremot kan neuronerna i de olika lagren ha olika aktiveringsfunktioner beroende på lagrets syfte etc.

De olika lagrens (ingående, dolt och utgående) syften är som följer. Det första, ingående, lagret har inga inkommande anslutningar från andra neuroner utan detta lagrets inkommande signaler är från omgivningen där nätet befinner sig. En identitetsfunktion används ofta som aktiveringsfunktion för neuronerna i detta lager, vilket innebär att signalen från omgivningen enbart skickas vidare till nästa lager av neuroner. Det framgår också tydligt i figuren ovan att varje neuron i det första lagret har en koppling till varje neuron i det efterföljande lagret. Har varje lager sådana kopplingar så är det ett så kallat "fully connected feedforward" nätverk.

Lagret efter det ingående (första) lagret är ett dolt lager. Ett neuronnät kan innehålla flera dolda lager; antalet dolda lager beror på den applikation eller det problem som neuronnätet utsätts för. Vanligast förekommande aktiveringsfunktion i dessa lager för denna sortens applikationer är enligt Rogers (1996) sigmoid (se även sektion 2.3.1). Antalet neuroner i de dolda lagren är något som varierar från fall till fall och tas ofta fram via experiment. Detta för att ett neuronnät med för många neuroner i de dolda lagren har en tendens till att bli specialiserade, vilket är en egenskap man sällan vill ha hos ett neuronnät. Ett specialiserat neuronnät har svårt att hantera ny, ej tidigare sedd, data.

Det sista lagret, det utgående lagret, är ett lager som för information från neuronnätet och ut till omgivningen där neuronnätet befinner sig. Med hjälp av den information som når omgivningen kan man träna neuronnätet. Genom att omgivningen har en förväntan för varje neuron i det sista lagret, ett så kallat målvärde, så kan varje neuron i det sista lagret belönas/straffas individuellt. Det finns olika tillvägagångssätt för att träna ett neuronnät av typen feedforward, den som används i detta arbete är, som det tidigare nämnts, Backpropagation. Mer om träning utav neuronnät följer i sektion 2.3.3.

2.3.3 Träning

> “Neural networks can be explicitly programmed to perform a task by manually creating the topology and then setting the weights of each link and threshold. However, this by-passes one of the unique strengths of neural nets: the ability to program themselves.” (Fraser, 1998)

Denna sektion kommer att behandla olika generella metoder för träning/inlärning, vilket sedan följs av en kort genomgång av algoritmen Backpropagation. Men först lite generellt om vad träning innebär, särskilt i samband med neuronnät.

En vanlig algoritm, i valfritt t.ex. programmeringsspråk, är statisk och utstadgad från dess början och dess funktion är tydligt specifierad. Neuronnät däremot skiljer sig på denna punkt, just för att man med neuronnät skapar själva neuronnätet som sedan tränas till att lösa problemet. Alltså behövs exempelvis ingen programkod, vad det

gäller algoritmen, skrivas. Neuronnätet har den egenskapen att det är adaptivt och problemet ligger istället i att konstruera ett lämpligt neuronnät och sedan träna det på lämpligt sätt. Resten är upp till neuronnätet. Träningen som neuronnätet utsätts för gör att neuronnätet anpassar sig till det problem som skall lösas (om möjligt). Det som sker när ett neuronnät tränas är att vikterna (styrkan på kopplingarna) mellan neuronerna modifieras. Ett neuronnät som tränas för länge kommer att bli specialiserat och därmed kan underliga beteenden uppstå då neuronnätet utsätts för ny data (input), vilket sällan är en egenskap som eftersträvas hos ett neuronnät.

Övervakad inlärning innebär att det i samband med träningsprocessen finns tillgång till facit. Detta innebär att man för varje input till neuronnätet också har ett, i förväg angivet, output att vänta sig. Genom översedd inlärning försöker man få neruonnätet att, via modifiering av dess vikter, efterlikna det förväntade resultatet (facit).

Oövervakad inlärning kallas det när det inte finns något facit att tillgå, det vill säga att en mängd input finns men inget, i förväg angivet, output (facit). Neuronnätet utforskar själv problemrymden och neuronnätets uppgift är att minimera en kostnadsfunktion. Denna typ av inlärning kan användas för bl.a. datakomprimering samt klustring. Neuronnätet har möjligheten att, eventuellt, finna relationer mellan olika områden inom hela problemområdet, oavsett vare sig det tidigare är utforskat eller ej.

Förstärkningsinlärning innebär att neuronnätet interagerar med en omgivning och för varje beslut neuronnätet tar så erhålls en respons från dess omgivning. Responsen kan antingen vara i form av en belöning eller en bestraffning och neuronnätets mål är att finna ett sätt som leder till att det blir bestraffat så lite som möjligt. Precis som tidigare så ändras neuronnätets beteende genom att modifiering av nätets vikter sker.

Backpropagation faller under katergorin övervakad inlärning och den kommer att kortfattat beskrivas nedan. För en mer utförlig beskrivning över hur Backpropagation fungerar se exempelvis (Werbos, 1994). Eftersom det är en övervakad inlärning, så finns det i förväg en koppling mellan input och output angiven. Neuronnätet utsätts för en mängd input och den erhållna outputen jämförs med den, enligt facit, förväntade outputen och ett så kallat felvärde räknas ut genom:

$$error = expected - actual$$

Detta görs, som sagt, för varje neuron i det utgående lagret. Felvärden, även kallat ”blame”, propageras sedan bakåt i nätet, via dess vikter. Detta för att eventuell modifiering av vikter skall kunna vara möjlig. Varje neuron mellan det ingående lagret och det utgående lagret erhåller en s.k. ”blame”, vilket är baserad på hur stark inverkan (stark koppling) den hade på neuronen den har utgående anslutningar till. Algoritmen består kort sagt utav följande steg, nämligen:

1. Beräkna neuronnätets output, utifrån dess input.
2. Felvärde beräknas för varje neuron i det ugående lagret.
3. Felvärde(n) propageras bakåt.
4. Modifiering av vikter.

Dessa steg är vad som kallas en epok. Stegen ovan utförs tills det att neuronnätet befinner sig vid en satt gräns. Och vid det läget anses nätet ha lärt sig och skall därmed inte tränas mer.

Två faktorer som kan spela en stor roll i hur neuronnätets anpassning sker är *learning rate* och *momentum*. Den förstnämnda är en faktor som bestämmer hur stor förändring skall ske på vikterna, utöver felvärdet. Momentum däremot är inte alltid nödvändig, men tack vare momentum så kan neuronnätets anpassning förhoppningsvis förbättras. Detta genom att lägga på en procentdel av en vikts föregående ändring till nuvarande ändring, vilket leder till att lokala minimum inom rymden för errors kan undvikas (dock inte alltid).

”Root Mean Squared Error” (RMSE) har i detta arbete använts som mått för att följa de artificiella neuronnätens eventuella framsteg i deras träning. RMSE som mått innebär att för en epok beräkna felvärdet (error) för varje neuron (n) i det utgående lagret. Felvärden kvadreras sedan och adderas. Efter detta räknas medelvärdet ut, baserat på samtliga kvadrerade felvärden och slutligen tas kvadratroten på detta värde, se formeln enligt (Wikipedia , 2001) nedan.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(error_i)^2}{n}}$$

3 Problemdefinition

I detta arbete ligger fokus på att undersöka hur tangentbordsstatistik och artificiella neuronnät, av typen feedforward, kan användas tillsammans för att verifiera användare. Utgångspunkten för arbetet blir den lösning som togs upp av Rogers (1996), i vilken det inte framgår hur avvikelser i data påverkar de artificiella neuronnäten och dess förmåga att verifiera.

Att utifrån användares beteende vid tangentbordsskrivning kunna verifiera dem är något som är av intresse eftersom det inte kräver någon ny hårdvara, vilket har framgått tidigare (se sektion 2.2.2). Eftersom ingen ny hårdvara krävs, så är också priset för införandet lågt och kan säkerheten höjas till ett lågt pris så är det, som sagt, av intresse. Det är därför av intresse att undersöka hur pass robust en lösning baserad på tangentbordsstatistik och artificiella neuronnät är, vad det gäller avvikelser i data samt pricksäkerhet vid verifiering.

3.1 Problemprecisering

Syftet med detta arbete är att undersöka huruvida den konfiguration som Rogers (1996) använt för att verifiera användare via tangentbordsskrivning är ett säkert och pålitligt tillvägagånssätt, med avseende på robusthet (neuronnätens förmåga att generalisera utifrån variationer i data) och pricksäkerhet.

3.2 Motivering

Motiveringen till detta arbete är att det är viktigt att titta på biometribaserade alternativ till traditionella verifieringsmetoder. Anledningen till varför det är viktigt är att de biometriska egenskaper (beteenden) som används i detta arbete är av sådan karaktär att de är svåra att stjäla samt föra vidare.

Rogers (1996) lade fram en lovande lösning baserad på artificiella neuronnät. Denna lösning omfattar endast möjligheten att verifiera användare men tar inte upp någonting vad det gäller avvikelser i data och dess inverkan på verifiering. Avvikelser i data och dess inverkan på verifiering är, i detta sammanhang, en parameter som behöver undersökas, eftersom användares beteenden sällan är identiska.

3.3 Mål

Arbetets mål är att undersöka huruvida den av Rogers (1996) föreslagna konfigurationen av ett artificiellt neuronnät lämpar sig för verifiering utav användare via tangentbordsskrivning och därmed kunna klarlägga det som ej framgår i Rogers (1996) föreslagna lösning, närmare bestämt hur avvikelser i data påverkar verifieringen samt pricksäkerheten.

4 Metod

Det framgår av kapitel 3 att en befintlig lösning utgör en central del i detta arbete. Den befintliga lösningen, (Rogers, 1996), innehåller implicit en hypotes och hypotesen utgör det huvudsakliga målet med arbetet, nämligen att undersöka ifall hypotesen håller. För att undersöka hypotesen tillsammans med den konfiguration av neuronnät som framgår av lösningen behandlas ett litet antal variabler samt data och inom ramarna för metoden experiment är detta, enligt Berndtsson, Hansson, Olsson och Lundell (2004), vad experiment handlar om och därmed har metoden experiment valts. Experiment utgör därmed den huvudsakliga metoden i detta arbete.

För att det sedan ska vara möjligt att undersöka samt utvärdera den befintliga lösningen behövs en applikation som gör det möjligt att utföra nödvändiga tester, och därmed behövs en implementation som gör detta möjligt. En sådan implementation kräver i sin tur data, i form av personers beteende, vilket också är nödvändigt för att undersöka och utvärdera den befintliga lösningen. Arbetet handlar alltså inte bara om att undersöka en befintlig lösning, utan arbetet innebär också framtagning (implementering) utav en applikation som tillåter att den av Rogers (1996) framlagda hypotes undersöks.

Metoder som identifierats, och som förövrigt ingår i detta arbete, är experiment med tillhörande implementering. Experimentets upplägg och de moment som ingår följer i sektion 4.1 och detaljer kring implementation följer i sektion 4.2. Detaljer angående analys av artificiella neuronnät följer till sist i sektion 4.3.

4.1 Experiment

Arbetets huvudsakliga syfte är, som sagt, att undersöka ifall den hypotes utifrån den befintliga lösningen håller, vilket innebär att ta reda på ifall lösningen är ett säkert och robust tillvägagångssätt för identitetsverifiering. För att ta reda på detta behövs lösningen testas, och två faktorer som ingår i testning är robusthet och pricksäkerhet.

Robustheten framhävs genom att utsätta neuronnät för ej tidigare sedd data, innehållandes avvikelser. Neuronnätets output noteras sedan för att se hur stor vikt avvikelser har.

Pricksäkerheten erhålls utifrån scenariot intrångsförsök. Samtliga neuronnät utsätts för en och samma, ej tidigare sedd, data. Detta för att se hur samtliga neuronnät beter sig samt erhålla en övergripande bild vad det gäller säkerheten vid verifiering.

Utifrån det att den befintliga lösningen testas med avseende på de två ovan nämnda faktorerna är det möjligt att se ifall hypotesen håller.

4.2 Implementering

I de undersektioner som följer kommer detaljer angående implementation samt format på exempeldata (dataformat) att beskrivas.

4.2.1 Exempeldata och dess format

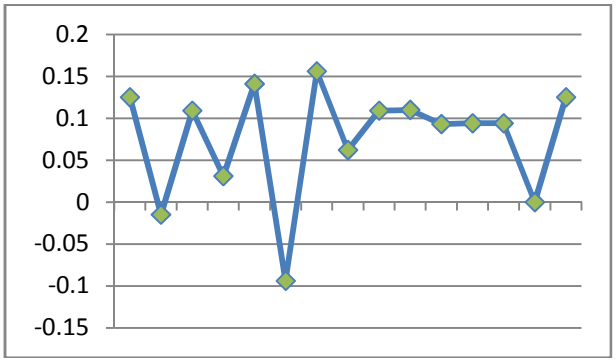
Exempeldata utgör i detta arbete en mycket viktig del, då desto mer data gör det möjligt att testa lösningen ännu mer. Genom att ha en stor mängd exempeldata gör det dessutom möjligt att testa lösningens pricksäkerhet vid verifiering mer utförligt. Pricksäkerhet i detta sammanhang innebär hur pass väl de artificiella neuronnäten utifrån en mängd exempeldata klarar att godkänna de dataexempel (mönster) som neuronnätet är uttänkt för.

Varje exemplar av exempeldata innehåller beteendemönster för 20 textsträngar (10 inmatningar per textsträng). Valet av textsträng nummer ett, *blackbox*, var att försöka finna en textsträng innehållandes tecken som låg relativt långt ifrån en någorlunda normal fingersättning. Textsträng nummer två, *biometrics are fun*, valdes enbart för att erhålla en större datamängd att presentera för de artificiella neuronnäten.

Exempeldata innehåller, som sagt, personers beteendemönster och i detta fall är ett beteendemönster uppbyggt av tider, vilka är baserade på tangentbordsnedtryckningar. För varje tecken i textsträngen tas två tider, nämligen nedhållningstid och fördröjningstid. Nedhållningstid innebär hur länge tangenten har hållits nere. Fördröjningstid innebär istället hur lång tid det tar från det att tangenten släpps upp tills det att nästa tangent trycks ned. För en illustration över hur ett faktiskt dataexempel kan se ut se Tabell 3 och för en visuell presentation se Figur 11.

Tecken	b	l	a	c	k	b	o	x
Hålltid	0.125	0.109	0.141	0.156	0.109	0.093	0.094	0.125
Fördröjning	-0.015	0.031	-0.094	0.062	0.11	0.094	0.0	NULL

Tabell 3 Dataexempel för textsträngen blackbox.



Figur 11 Dataexempel, visuellt, för textsträngen blackbox

```
Användarnamn
<textsträng#1>
Tidsstämpel#1|Tidsstämpel#2|Nedhållningstid|Fördröjningstid|Tidsst
ämpel#3|…|Nedhållningstid|
…
<textsträng#2>
…
```

Data 1 Dataformat för exempeldata med dess interna layout.

Enligt formatet (Data 1) framgår det att det för varje textsträng blir en del innehåll, samt att varje element separeras av tecknet ”|”. För varje tecken i textsträngen erhålls

två tidsstämplar (se ovan). Tidsstämplarna har i syfte att förmedla när tangent α trycks ned (Tidsstämpel#n) och när tangent α släpps upp (Tidsstämpel#(n+1)). Utifrån dessa tidsstämplar kan sedan två intressanta tider beräknas, nämligen:

1. Nedhållningstiden = $Tidsstämpel\#(n+1) - Tidsstämpel\#n$
2. Fördröjningstiden = $Tidsstämpel\#(n+2) - Tidsstämpel\#(n+1)$

Ur det faktiska och redan beräknade dataexemplet ovan (Tabell 3), för strängen *blackbox*, så framgår det att negativa tider infinner sig. Negativa tider innebär då att en tangent som förväntas att tryckas ned trycks ned innan den föregående tangenten har släppts upp. Det är sedan utifrån dessa tider, nedhållningstid och fördröjnignstid, som användare verifieras. Därav rödmarkeringen ovan.

Nedan följer ett räkneexempel för följande data: 143|200|α_1|β_1|232|270|α_2|. Detta för att på ett enkelt sätt klargöra hur de olika tiderna erhålls samt utesluta eventuell misstolkning.

$$\alpha_1 = 200 - 143$$

$$\beta_1 = 232 - 200$$

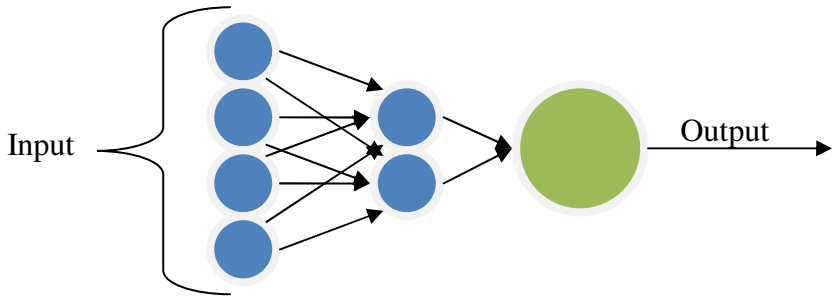
$$\alpha_2 = 270 - 232$$

Utseendet på exempeldata ser alltså ut som följande: 143|200|***57***|***32***|232|270|***38***|.

4.2.2 Val av konfiguration utav neuronnät

Detta arbete är baserat på Rogers (1996) arbete och som också är utgångspunkten för arbetet, vilket medför att vi valt samma konfiguration samt typ av artificiella neuronnät. Valet föll därmed på att använda artificiella neuronnät av typen feedforward.

Konfigurationen som valts innebär att varje användare som skall kunna verifieras erhåller ett alldeles eget neuronnät. Konfigurationen för de artificiella neuronnäten ser ut som följande:



Figur 12 Enkel konfiguration, vilken har möjligheten att säga Ja eller Nej.

En konfiguration, enligt Figur 12, har möjligheten att säga utifrån dess input ifall det är användare Foo eller inte. Konfigurationen har alltså i syfte att känna igen en och endast en användares beteendemönster (input). Det behövs således ett sådant artificiellt neuronnät för varje användare.

4.3 Val av analysteknik för analys utav neuronnäten

Det finns ett antal olika tillvägagångssätt för att undersöka samt analysera artificiella neuronnät, så som analys av dess vikters utveckling, analys utav artificiella neuronnätets interna aktivering, matematisk analys (i den mån det är möjligt) eller att empiriskt föra anteckningar över hur det artificiella neuronnätet beter sig. Det sistnämnda alternativet innebär att man kan se det artificiella neuronnätet som en svart låda och på så vis fokusera sig på hur det faktiskt beter sig.

Inom detta arbete behövde fokus läggas på hur de utav artificiella neuronnäten betedde sig, och därmed så tillämpas det sistnämnda alternativet ovan, nämligen att se artificiella neuronnäten som svarta lådor.

5 Genomförande

I detta kapitel beskrivs de moment som ingår i experimentet utförligt och utöver detta tas implementeringsdetaljer upp vad det gäller dataformat samt applikationen och dess artificiella neuronnät. Val som gjorts motiveras och ytterligare teori som används i samband med implementering tas upp i mån av behov. Kapitlet inleds med att presentera de experiment som utförts samt en redogörelse över hur de utförts, vilket följs av implementeringsdetaljer.

5.1 Experiment

För att framhäva denna konfiguration utav artificiella neuronnäts förmåga att generalisera (robusthet) utsätts ett tränat neuronnät för ej tidigare sedd data, vilket framgår i sektion 4.1. Denna data som neuronnätet utsätts för är till en början baserad på ett dataexempel som neuronnätet tränats utifrån. I dataexemplet införs sedan avvikelser, systematiskt, för att sedan återigen presenteras för neuronnätet och då för att notera avvikelsernas betydelse.

Det systematiska tillvägagångssättet för att införa avvikelser i dataexemplet innebär att värden inom dataexemplet modifieras och positionen inom dataexemplet som modifieras är slumpartat utvalt. Modifieringen som utförs på den utvalda positionen i dataexemplet är vald utifrån följande variationsgrad: ± 1%, ± 10%, ± 50%, ±80% eller ± 500% . Antalet element (positioner) i dataexemplet som erhåller modifiering är i ett testfall 1 och andra testfallet hela 10 stycken. Exempelvis kan ett element i ett dataexempel på position 4 modifieras genom att dess värde ökas med 10%. Värdet på positionen kan exempelvis representera en nedhållningstid. Det är alltså dessa tider (nedhållningstid & fördröjningstid) som modifieras systematiskt.

Vad det gäller neuronnätens pricksäkerhet vid verifiering så framhävs denna egenskap genom scenariot *intrångsförsök*. Genom att utsätta samtliga tränade neuronnät för ett och samma dataexempel (fritt från avvikelser) är det möjligt att se hur pass säkra samt pricksäkra de är vad det gäller verifiering. Det dataexempel som används är baserat på mitt beteende och därmed skall endast ett neuronnät (mitt neuronnät) godkänna denna input. Ett neuronnäts sista neuron, den i utgående lagret, erhåller hög aktivering ifall dess indata (input) innehåller mönster som neuronnätet känner igen, annars erhålls låg aktivering. Hög aktivering innebär ett utgående värde kring 1 och låg aktivering är kring värdet 0.

5.2 Implementering

5.2.1 Datainsamling

En ytterst enkel applikation vars enda syfte var att lagra personers beteendemönster, enligt formatet som framgick i sektion (4.2.1) implementerades. Applikationen presenterades sedan, publikt, på två utvalda ställen i syfte för att erhålla *verklig* exempeldata som sedan kunde användas för att träna de artificiella neuronnäten.

För att erhålla en så stor mängd exempeldata som möjligt så ansågs det vara, som sagt, lämpligt att publikt fråga efter personers beteendemönster. Följande har använts som pool för insamling av exempeldata:

- www.sweclockers.com/forum/
- IRC[3]-kanalen #netbsd.se på nätverket irc.gimp.net
- Nära och bekanta

Erhållen mängd exempeldata utifrån ovanstående blev totalt 52 olika beteenden.

Datainnehållet valdes till att innehålla en mängd överflödig data, i form av tidsstämplar. Detta för att underlätta ifall det vid senare skede skulle visa sig att andra beräkningar skulle behöva utföras på exempeldata. Nedan följer ett exempel innehållandes *ett* dataexempel under textsträngen *blackbox*.

```
eddie
<blackbox>
1177152742562|1177152742687|125|0|1177152742687|1177152742812|125|0|1
177152742812|1177152742984|172|-109|1177152742875|1177152743031|156|-
78|1177152742953|1177152743062|109|109|1177152743171|1177152743250|79
|93|1177152743343|1177152743437|94|0|1177152743437|1177152743562|125|
…
<biometrics are fun>
…
```

Data 2 Ett faktiskt dataexempel för textsträngen blackbox

5.2.2 Artificiella neuronnät, dess implementationsdetaljer

Enligt sektion 4.2.2 framgår det att denna konfiguration kräver ett artificiellt neuronnät för varje användare som skall kunna verifieras. De artificiella neuronnäten som används skapas utifrån antalet tecken i den textsträng som valt att användas för verifiering, vilket framgår nedan.

Varje enskilt neuronnät inom denna lösning är identiska vad det gäller antalet neuroner i de olika lagren. Det som initialt skiljer dem åt däremot är värden på de interna vikterna, vilka sedan modifieras via träning. Antalet neuroner i det första lagret ges av följande:

$$\#ingående = \big((length\ of\ string) * 2\big) - 1$$

Enligt formeln (#ingående) ovan framgår det att för exempelvis textsträngen *blackbox* så erhålls värdet 15, och därmed består det ingående lagret av 15 neuroner. Vad det gäller lager nummer två så erhålls antalet neuroner enligt följande:

$$\#dolt = \left\lfloor \frac{\Big(\big((length\ of\ string) * 2\big) - 1\Big)}{2} \right\rfloor$$

Enligt formeln (#dolt) ovan framgår det att för textsträngen *blackbox* innehåller det dolda lagret 7 neuroner. Sedan vad det gäller det sista lagret så innehåller det endast en neuron. Formeln är baserad på en tumregel som kretsar inom området för

[3] IRC – Internet Relay Chat

neuronnät av denna typ. Tumregeln lyder: För ett dolt lager använd hälften av antalet neuroner i det ingående lagret som utgångspunkt för antalet neuroner i det dolda lagret.

5.2.3 Artificiella neuronnät, dess träning

Utifrån den exempeldata som erhållits genom datainsamling tränas de artificiella neuronnäten. I denna lösning dedikeras varje exemplar av exempeldata ett eget artificiellt neuronnät. Varje neuronnät tränas var för sig enligt algoritmen Backpropagation (se sektion 2.3.3). I samband med träningen utsätts varje neuronnät för samtliga exemplar av exempeldata, tills det att neuronnätets beteende anses vara korrekt. Neuronnätets beteende anses vara korrekt då dess RMSE-värde antingen når värdet 0.01 eller att antalet träningsepoker når sitt slut. Antalet träningsepoker som neuronnäten har tillgodo räknas ut enligt följande:

$$\#epok = 200 * \#users$$

För en användarmängd på 50 blir antalet träningsepoker som neuronnäten har tillgodo 10000, för att erhålla ett RMSE-värde på 0.01. Det behövs sällan så många träningsepoker för att erhålla ett RMSE-värde kring 0.01, men det beror på de olika användarnas beteenden. Parametrarna *learning rate* och *momentum* har i enlighet med konfigurationen beskriven utav Rogers (1996) erhållit värdena 0.45 och 0.9 respektive.

6 Resultat

För att kunna dra någon slutsats om huruvuida den hypotes som Rogers (1996) lade fram håller eller ej så behövdes ett antal tester utföras. Tester vars huvudsakliga syfte var att erhålla svar på följande frågor:

- Hur robust är lösningen?
- Hur säker är lösningen (pricksäkerhet) ?

I kommande sektioner tas dessa tester upp tillsammans med de resultat som erhållits.

6.1 Robusthet

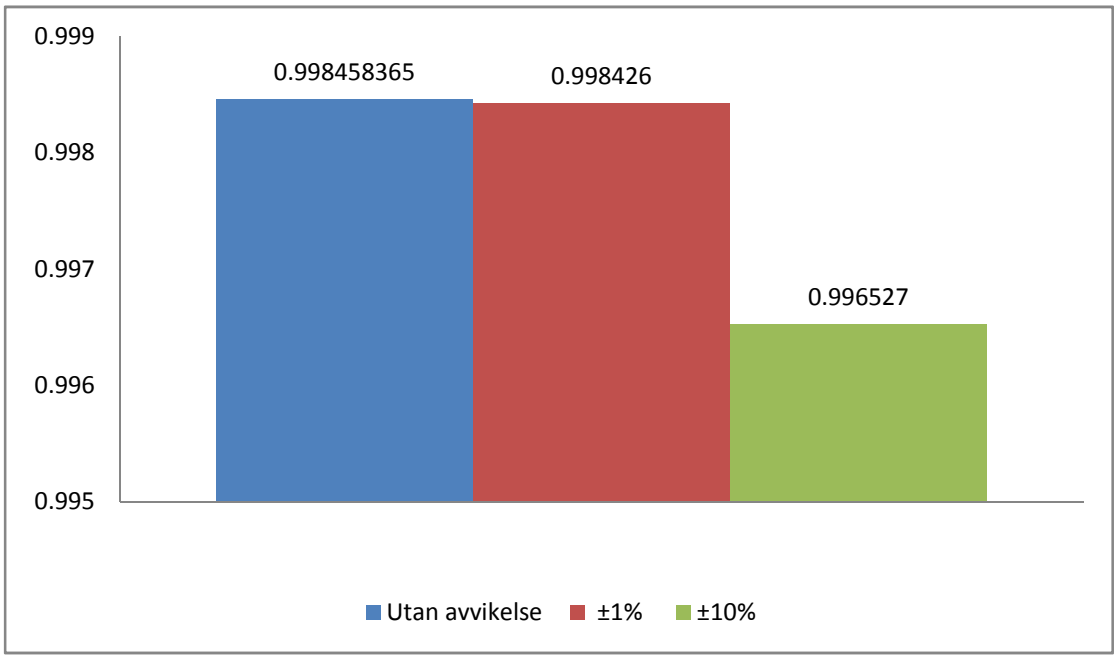
För att framhäva ett artificiellt neuronnäts förmåga att generalisera utsätts neuronnätet för data innehållandes avvikelser. Den data (utan avvikelser) som neuronnätet utsätts för är data som neuronnätet skall godkänna, eftersom innehållet består av mönster som neuronnätet har tränats för att känna igen. Genom att sedan införa små, stora och en blandning av små och stora ändringar på slumpvalda ställen i den data som neuronnätet utsätts för kan dess förmåga att generalisera framhävas.

Samtliga värden som förekommer i Figur 13 till och med Figur 17 är medelvärden, vilka är uträknade utifrån minst 30 körningar utav vardera test. Anledningen till varför alla test ej erhöll samma antal körningar är att vissa tester behövde köras mer för att undersöka eventuell spridning i resultaten.

Värt att tillägga är att antalet tecken i textsträngen som används för verifiering har , i detta fall, visat sig ha en mindre betydelse. Använda textsträngar är följande två: *blackbox* och *biometrics are fun*.

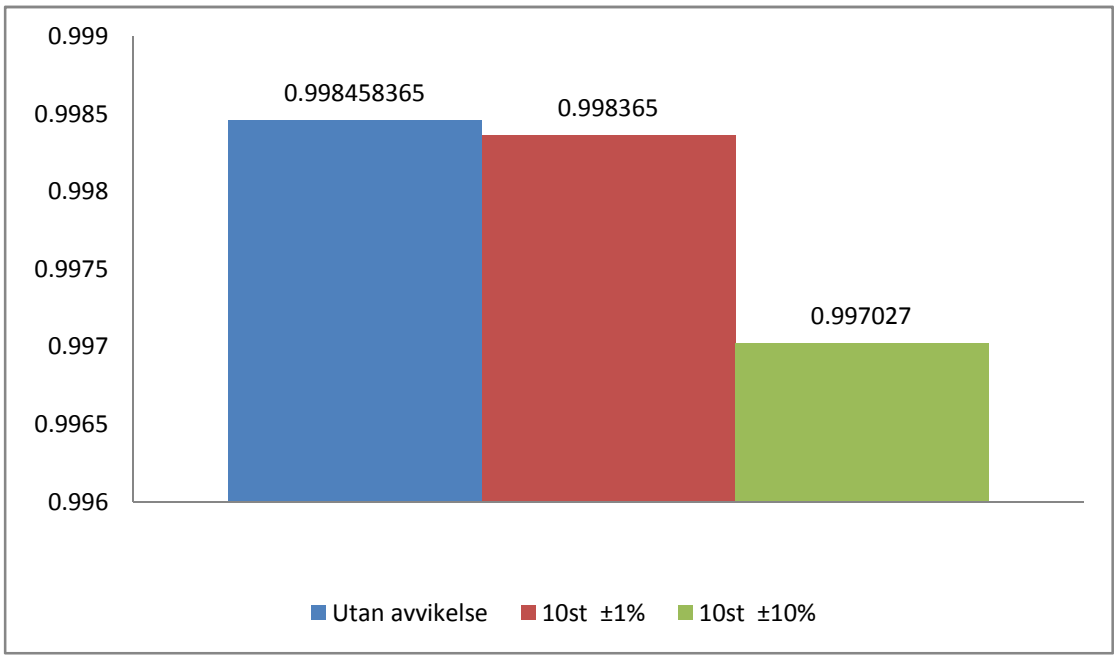
6.1.1 Data innehållandes små avvikelser

Efter att ha utsatt neuronnät för data innehållandes en mindre avvikelse (±1% eller ±10%) erhölls följande resultat:



Figur 13 Neuronnätets output (Y-axeln) baserad på data innehållandes en mindre avvikelse.

Efter att ha utsatt neuronnät för data innehållandes 10 mindre avvikelser (±1% eller ±10%) erhölls följande resultat:

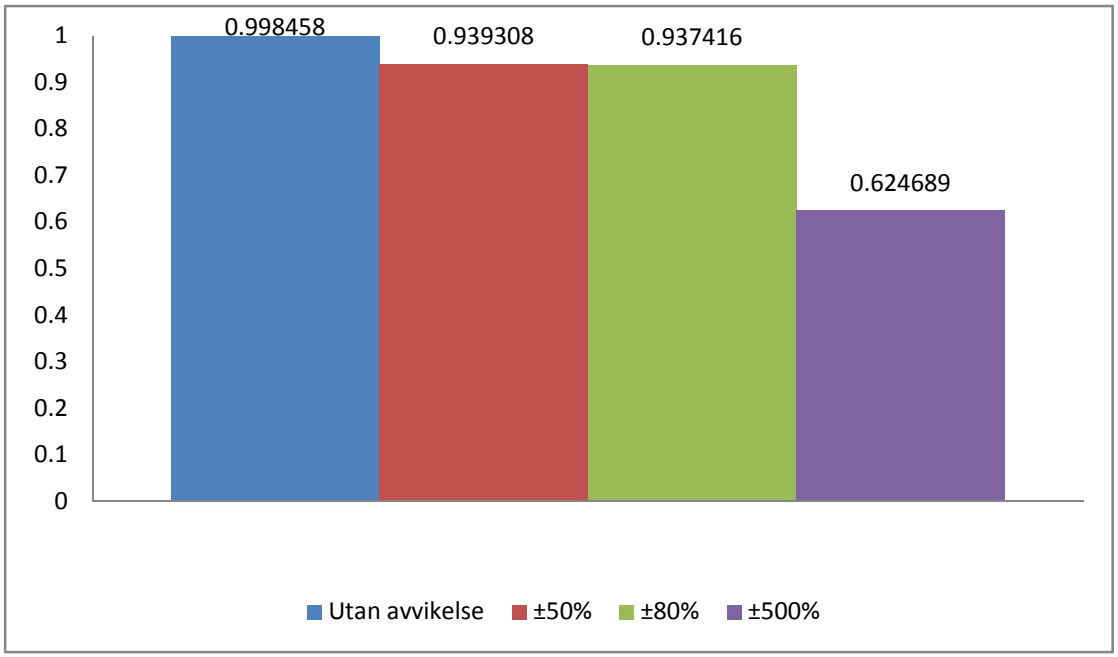


Figur 14 Neuronnätets ouoput (Y-axeln) baserad på data innehållandes 10 mindre avvikelser.

En lämplig gräns enligt Rogers (1996) för *verifierad* är 0.9 och enligt Figur 13 och Figur 14 framgår det att en eller flera mindre avvikelser inte har en särskilt stor inverkan på verifieringen.

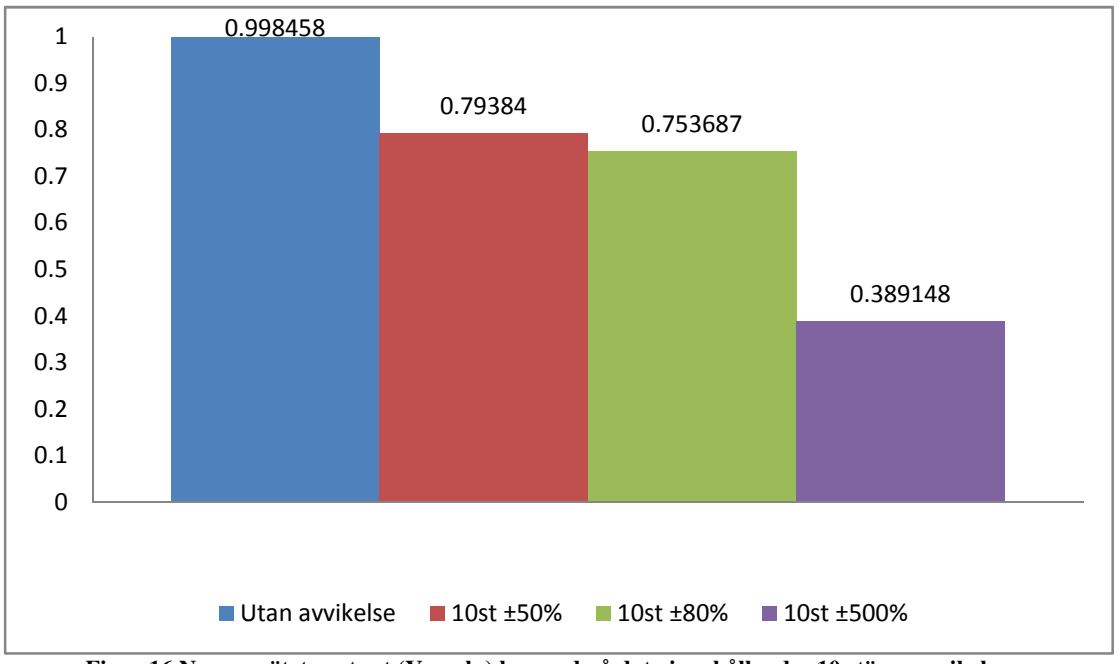
6.1.2 Data innehållandes större avvikelser

Efter att ha utsatt neuronnät för data innehållandes en större avvikelse (±50%, ±80% eller ±500%) erhölls följande resultat:



Figur 15 Neuronnätets output (Y-axeln) baserad på data innehållandes en större avvikelse.

Efter att ha utsatt neuronnät för data innehållandes 10 större avvikelser (±50%, ±80% eller ±500%) erhölls följande resultat:

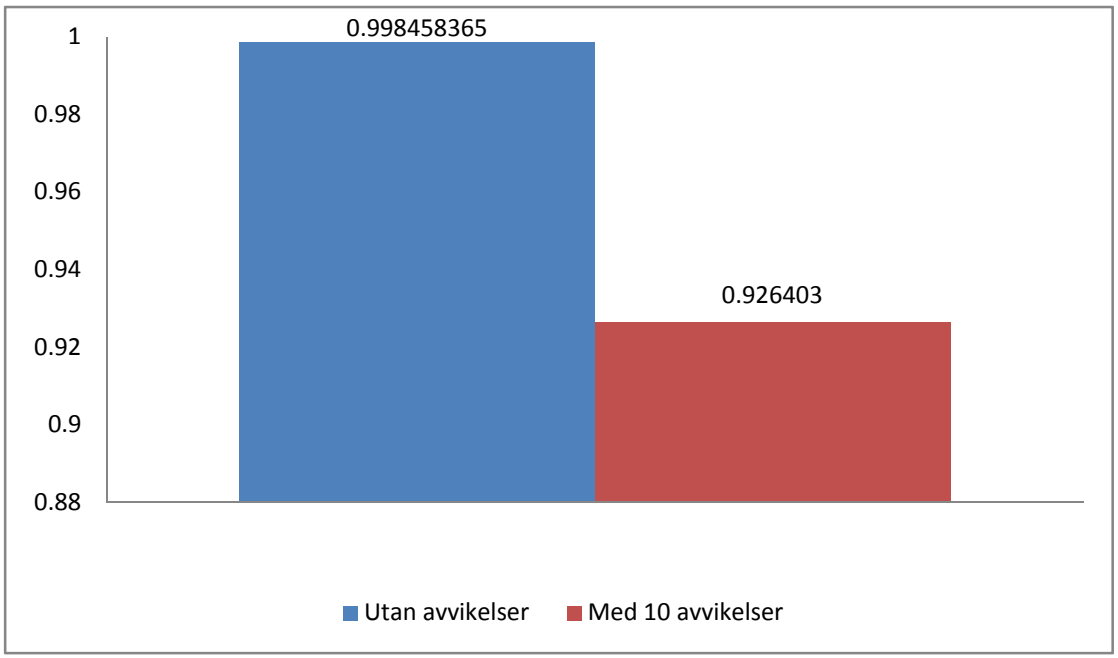


Figur 16 Neuronnätets output (Y-axeln) baserad på data innehållandes 10 större avvikelser.

Utifrån den gräns (0.9) som Rogers (1996) ansåg vara lämplig för verifierad så framgår det av Figur 15 att enskilda avvikelser av storleksordningen ±50% eller ±80% accepteras av neuronnätet. Ett större antal avvikelser av storleksordningen ±50%, ±80% eller ±500% accepteras däremot inte, vilket är bra (se Figur 16).

6.1.3 Data innehållandes blandade avvikelser

Efter att ha utsatt neuronnät för data innehållandes en blandning utav avvikelser av storleksordningen ±1%, ±10%,±50% eller ±80% erhölls följande resultat:

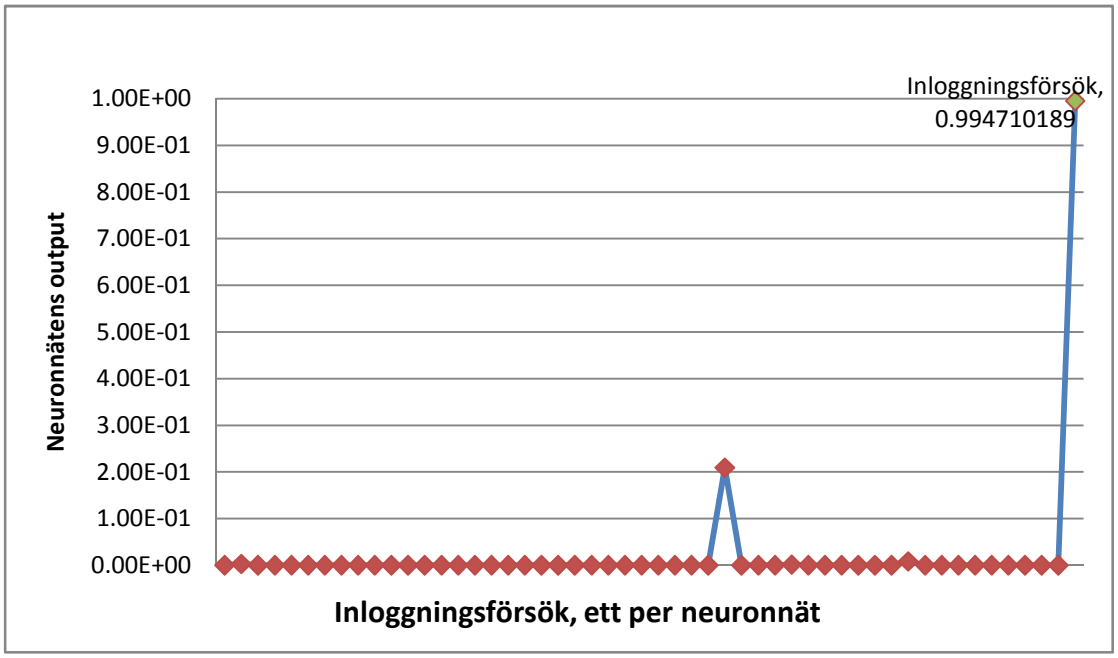


Figur 17 Neuronnätets output (Y-axeln) baserad på data innehållandes 10 slumpvalda avvikelser.

Det framgår av Figur 17 att avvikelser, av olika storlekar, i data accepteras relativt väl utav neuronnätet vilket både kan vara bra och dåligt beroende på applikation. En människas beteende innehåller avvikelser, vilket har framgått av den insamlade mängden data. Eftersom den data som används inom detta arbete innehåller avvikelser så är det att föredra att neuronnäten faktiskt tar hänsyn till avvikelserna och gränsen för verifierad kan ju alltid sättas till lämpligt värde, vilket kan variera från fall till fall.

6.2 Pricksäkerhet

Samtliga tränade neuronnät, i detta fall 52st, utsätts för ny och ej tidigare sedd data. Detta i hopp om att neuronnätet utifrån ny data skall klassificiera mig som dess uttänkte användare, d.v.s. bete sig felaktigt. Resultatet som erhölls är lovande, då samtliga neuronnät inte godkände mitt beteendemönster. Det neuronnät som godkände mitt beteendemönster var det neuronnät som var uttänkt för mitt beteendemönster, vilket framgår av följande figur:



Figur 18 Neuronnätens output efter att ha utsatts för ett och samma dataexempel (intrångsförsök).

Under träningen utav de artificiella neuronnäten noterades det att inte alla neuronnät erhöll ett RMSE-värde kring 0.01 utan många neuronnät hamnde kring 0.03, vilket har visat sig vara tillräckligt (se Figur 18).

7 Slutsats

Kapitlet sammanfattar det arbete som utförts genom att de resultat som erhållits diskuteras. Förslag till framtida arbete ges också.

Utifrån den lösning som lades fram av Rogers (1996) framgår det att artificiella neuronnät av typen feedforward och beteendeinriktad biometri kan användas för verifiering utav användare. Den typ av beteendeinriktad biometri som lösningen utnyttjar är användarnas beteende då dessa skriver på tangentbordet, men lösningen har ej fokus på avvikelser i beteenden utan den har snarare fokus på möjligheten att kunna verifiera användare.

Utifrån den mängd data (beteenden) som samlats in för detta arbete framgår det att viss avvikelse finns i människors beteende vid tangentbordsskrivning, vilket givetvis kan variera från fall till fall samt människa till människa.

Enligt de tester och experiment som utförts inom detta arbete så har det visat sig att lösningen är pricksäker vad det gäller verifiering. Detta framgår tydligt av Figur 18 att neuronnätens beteende var mycket bra och skulle en gräns, exempelvis 0.9, sättas för *verifierad* så var neuronnäten inte ens i närheten att misslyckas. Antingen beror det på att jag, Eddie, har ett väldigt ovanligt sätt att skriva på tangentbordet eller så är det helt enkelt ett bra (pricksäkert) tillvägagångssätt för verifiering utav användare. Utöver lösningens pricksäkerhet så har den också visat sig vara robust; neuronnäten tar hänsyn till avvikelser i data. Mindre avvikelser i data har visat sig ha en mindre inverkan på verifiering medan större avvikelser har en större inverkan på verifiering, vilket framgår av sektion 6.1.

Utifrån de resultat som erhållits verkar det som den hypotes som Rogers (1996) lade fram håller.

7.1 Diskussion

Tillvägagångssättet som använts i detta arbete, som förövrigt är detsamma som använts i Rogers (1996) lösning, för att verifiera användare har visat sig fungera bra; långt över förväntan. Utifrån de testresultat som erhållits framgår de artificiella neuronnäten tar hänsyn till avvikelser, och kanske lite väl stort hänsynstagande från neuronnätens sida. En viss mån av hänsynstagande är nödvändigt men frågan är hur stort det skall vara med tanke på säkerhet; högre hänsynstagande gör det lättare att lura neuronnäten. Detta är förmodligen en balansgång och hur den ter sig skulle vara intressant att undersöka vidare.

Lösningen som sådan skulle lämpa sig ypperligt för statisk verifiering, vilket har bekräftats genom detta arbete. Och då främst statisk verifiering vid fysisk inloggning och inte fjärr-inloggning eftersom den data som används vid verifiering varken bör generas eller skickas över nätverk utav tredje part. Vad det gäller dynamisk verifiering däremot så skulle det förmodligen krävas en utökad mängd beteendemönster så som vid ordbehandling eller vardagligt internetsurfande där beteendena förmodligen skiljer sig åt mellan de olika sysselsättningarna. För att eventuellt göra det lättare för neuronnäten kunde man kanske kombinera olika beteendemönster för att tillsammans utgöra en användarprofil, exempelvis musrörelse. Detta vore värt att undersöka eftersom dynamisk verifiering har möjligheten att verifiera användare dynamiskt, d.v.s. att verifieringen inte är bunden till tid och rum.

Genom att verifiera användare dynamiskt kan exempelvis oönskat användarbyte vid en terminal undvikas.

Möjliga utökningar för lösningen är många men en intressant och kanske självklar utökning vore att utöver beteende vid inskrivning av textsträng också kolla på antalet felstavningar, d.v.s. föra statistik över hur tangenten backspace används. Detta är något som inte gjorts i detta arbete eftersom arbetets mål är att undersöka avvikelser och dess inverkan på verifiering, och därmed har arbetet utgått från den av Rogers (1996) föreslagna lösningen.

Vad det gäller mängden data så skulle det också kunna vara intressant att undersöka hur storleken på datamängden (antalet beteendemönster) skulle påverka lösningens robusthet. I detta arbete har 10 beteendemönster per användare använts vid träning och det kan mycket väl vara det som har lett till att neuronnäten accepterar så pass stor mängd avvikelser. Ett större antal beteendemönster per användare skulle förmodligen leda till mer specialiserade neuronnät, och därmed inte acceptera lika mycket avvikelser. Detta vore, som sagt, intressant eftersom att man då borde kunna anpassa lösningen och dess acceptansnivå (för avvikelser) utifrån antalet beteendemönster.

7.2 Framtida arbete

Det vore intressant att utföra ett skalbarhetstest genom att utifrån en större mängd användare (fler än 52) se hur lösningen skulle te sig. Genom en större mängd data skulle det också vara möjligt att se hur pass specialiserade neuronnäten kan bli och hur de då skulle bete sig vid verifiering; undersöka om neuronnäten är noggrannare och därmed har mindre hänsynstagande till avvikelser.

Hur avvikelser i data och dess placering inom dataexempel påverkar, om det påverkar, neuronnätet vid verifiering vore intressant att undersöka. Genom att undersöka hur avvikelser och dess placering påverkar kan det kanske visa sig att vissa positioner i dataexemplet är mer mottagliga för avvikelser än andra positioner.

Det skulle också vara intressant att undersöka ifall lösningen skulle kunna utökas från identitetsverifiering till identifiering.

Tack

Jag vill först och främst tacka min handledare Fredrik Johansson för den tid och allt det engagemang han lagt ned på detta arbete. Ett stort tack vill jag också rikta till samtliga personer som bidragit med data till detta arbete, vilket har utgjort en bra grund för testerna inom detta arbete. Vidare vill jag tacka min examinator Jonas Mellin för hans förslag och åsikter kring detta arbete.

8 Litteraturförteckning

Berndtsson, M., Hansson, J., Olsson, B., & Lundell, B. (2004). *Planning and Implementing your Final Year Project with Success!* Springer.

Callan, R. (1998). *Essence Of Neural Networks.* Harlow, UK: Pearson Higher Education.

Demir, G. (2002). *Identitetsverifiering via tangentbordsstatistik.* Linköping.

Fraser, N. (1998). *Training neural networks.* Hämtat från Neil´s neural nets: http://www.virtualventures.ca/~neil/neural/neuron-d.html den 17 04 2007

Internationella biometrigruppen. (2007). Hämtat från Internationella biometrigruppen: http://www.biometricgroup.com den 05 03 2007

Joyce, R., & Gupta, G. (1990). Identity authentication based on keystroke latencies. *Communication of the ACM 33(2)*, (ss. 168-176).

Minsky, M., & Papert, S. (1969). *Perceptrons.* Cambridge Massachusetts: The MIT Press.

Monrose, F., & Rubin, A. (1997). Authentication via Keystroke Dynamics. *Fourth ACM Conference on Computer and Communication Security.* Zurich, Switzerland.

Parker, D. (1985). *Learning logic.* . Cambridge, MA.: Center for Computational Research in Economics and Management Science, MIT.

Rogers, J. (1996). *Object-Oriented Neural Networks in C++*. London: Elsevier Science & Techonology.

Rumelhart, D., & McClelland, J. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition (I & II).* Cambridge: MIT Press.

Russell, S., & Norvig, P. (2002). *Artificial Intelligence, a modern approach.* Prentice Hall.

Svenska biometriföreningen. Hämtat från http://www.biometricassociation.org/ den 10 04 2007

Werbos, P. (1974). *Beyond regression: new tools for prediction and analysis in behavior sciences.* Phd. Thesis, Harvard, Cambridge, MA.

Werbos, P. (1994). *The Roots of Backpropagation.* New York: John Wiley & Sons, Inc.

Wikipedia . (den 15 Januari 2001). Hämtat från The Free Encyclopedia: http://en.wikipedia.org den 05 03 2007