

Evolution av artificiella neuronnät

Elias Riedel Gårding
NMA11
Norra Real, Stockholm

9 maj 2014

Abstract

We examine the characteristics of evolution by natural selection and how it can be applied to the field of artificial neural networks as a training method and as an improvement method for existing trained networks. We construct a simulation providing the prerequisites for evolution by natural selection and find that, given certain circumstances, evolution of a desired trait does occur. We do not succeed in using a training algorithm to produce neural networks mimicking a predefined behaviour, and so we cannot examine whether evolution by natural selection can be used to improve an existing algorithm.

The investigation raises many related questions, concerning both the fine details of cases like ours and the possible role of evolution in different applications.

Förord

Detta arbete gjordes som gymnasiearbete under läsåret 2013-2014 på Norra Real, Stockholm, under handledning av Anny Primdahl Markussen från Norra Real samt Felix Rios och Jonas Hallgren från KTH (Kungliga Tekniska Högskolan, Stockholm).

Arbetet var från början tänkt som ett samarbete mellan mig och Oscar Berggren (NMA11), men han kunde tyvärr inte fullfölja arbetet på grund av sjukdom. Jag riktar ett stort tack till Oscar för hans många insikter och idéer, utan vilka arbetet inte hade varit genomförbart, och jag skriver arbetet i vi-form trots att de senare delarna tekniskt sett borde vara i jag-form.

Jag tackar även Anny Markussen för hennes värdefulla handledning, feedback och kommentarer, och Felix Rios och Jonas Hallgren för våra givande dialoger och diskussioner om ämnet.

Innehåll

1	Bakgrund	3
1.1	Evolution genom naturlig och artificiell selektion	3
1.2	Artificiella neuronnät	3
1.2.1	Funktion	3
1.2.2	Inläring	5
1.3	Syfte	6
2	Metod	7
2.1	Simulationen	7
2.1.1	Individer	7
2.1.2	Genetik	8
2.2	Översättning av designad algoritm	9
2.3	Icke-styrd evolution av slumpmässigt genererade nät	9
2.3.1	Allmänna observationer	9
2.3.2	Empirisk undersökning av förbättring av ätförmåga	10
3	Diskussion	11
3.1	Generaliteten hos undersökningen	11
3.2	Mätningen	11
3.3	Konvergens hos backpropagation och genetisk algoritm	11
3.4	Slutsats	12
3.5	Förslag till vidare undersökningar	12

1 Bakgrund

1.1 Evolution genom naturligt och artificiell selektion

I naturen sker evolution genom naturligt urval av populationer av individer. Grundprincipen är att de organismer som reproducerar och får fertil avkomma i störst utsträckning bidrar till överlevnaden hos sina gener.

För att ett system skall kunna evolvera krävs att tre grundläggande kriterier är uppfyllda ([1], 2.2):

- Det måste kunna *reproducera*.
- Det måste finnas *variation*, det vill säga systemet måste undergå förändring.
- Det måste finnas *selektionstryck*, det vill säga generna måste ha inverkan på individens reproduktionsförmåga.

Det är viktigt att notera att evolution genom naturligt urval inte strävar mot något specifikt mål. De gener som sprids mest är de gener vars egenskaper ger dem bäst chans att spridas. Detta säger inget om hur dessa gener sprids; exempelvis behöver en lyckad gen inte nödvändigtvis bidra till dess bärarens överlevnad, utan kan till exempel göra den benägen att offra sig själv till förmån för andra organismer som bär på samma gen.

Det finns emellertid också exempel på evolution genom artificiellt urval, eller avel. Avelsprocessen liknar den naturliga evolutionsprocessen, men urvalet sker genom att organismer med egenskaper som anses önskvärda av uppfödaren ges mer reproduktionsmöjligheter.

Distinktionen mellan naturlig (icke-styrd) och artificiell (styrd) selektion är viktig att hålla i tankarna, då detta arbete behandlar båda.

1.2 Artificiella neuronät

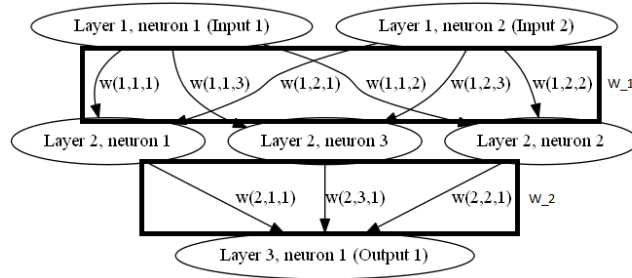
I detta arbete modellerar vi hjärnor som artificiella neuronät. Vi använder den enklaste typen av neuronät, *feedforward*-nät, beskrivna av många källor, exempelvis [8]. Här nedan definierar vi dessa så som de används i detta arbete.

1.2.1 Funktion

1.2.1.1 Definition av koncept Ett artificiellt feedforward-neuronät innehåller N lager. Varje lager innehåller ett visst antal neuroner. Vi definierar vektorn

$$S = [S_1 \ S_2 \ \dots \ S_{N-1} \ S_N]$$

så att S_n är lika med antalet neuroner i lager n .



Figur 1: Ett exempelnät med 3 lager.

Vi kan nu definiera L som listan av alla lager:

$$L = \left[\vec{l}_1 \quad \vec{l}_2 \quad \dots \quad \vec{l}_{N-1} \quad \vec{l}_N \right]$$

där $\vec{l}_n \in \mathbb{R}^{S_n}$ är lika med vektorn $[v_{n,1} \quad v_{n,2} \quad \dots \quad v_{n,S_n-1} \quad v_{n,S_n}]$ och $v_{n,k} \in \mathbb{R}$ är *värdet* på neuron k i lager n . Notera att L inte kan ses som en matris, då alla lager inte måste innehålla samma antal neuroner.

Vi definierar *vikten* mellan neuron a i lager n och neuron b i lager $n+1$ som $w_{n,a,b}$. Vi kan då definiera $(S_n \times S_{n+1})$ -matrisen W_n sådan att $(W_n)_{a,b} = w_{n,a,b}$:

$$W_n = \begin{bmatrix} w_{n,1,1} & w_{n,1,2} & \cdots & w_{n,1,S_{n+1}-1} & w_{n,1,S_{n+1}} \\ w_{n,2,1} & w_{n,2,2} & \cdots & w_{n,2,S_{n+1}-1} & w_{n,2,S_{n+1}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{n,S_n-1,1} & w_{n,S_n-1,2} & \cdots & w_{n,S_n-1,S_{n+1}-1} & w_{n,S_n-1,S_{n+1}} \\ w_{n,S_n,1} & w_{n,S_n,2} & \cdots & w_{n,S_n,S_{n+1}-1} & w_{n,S_n,S_{n+1}} \end{bmatrix}$$

I klarspråk: Mellan varje par av neuroner i intilliggande lager finns en vikt. Dessa listas i matrisen W_n , där n är numret på det första av de två lagren.

Till varje neuron hör dessutom en *bias-vikt*. Till neuron k i lager n hör bias-vikten $b_{n,k} \in \mathbb{R}$. Vi kan alltså samla samtliga bias-vikter i en lista av vektorer:

$$B = \left[\vec{b}_1 \quad \vec{b}_2 \quad \dots \quad \vec{b}_{N-1} \quad \vec{b}_N \right]$$

där $\vec{b}_n \in \mathbb{R}^{S_n}$ är lika med vektorn $[b_{n,1} \quad b_{n,2} \quad \dots \quad b_{n,S_n-1} \quad b_{n,S_n}]$, på precis samma sätt som L . Det är värt att notera att värdet på \vec{b}_1 aldrig används och därför inte har någon effekt på resultatet.

Vi definierar också *transferfunktionen* $\text{trans} : \mathbb{R} \rightarrow \mathbb{R}$ sådan att $-1 \leq \text{trans } x \leq 1$. Transferfunktionen normaliserar alltså ett reellt tal till att ligga mellan -1 och 1. I vårt fall är $\text{trans } x = \tanh x$.

1.2.1.2 Evaluering Neuronnätet kan beskrivas som en funktion $E : \mathbb{R}^{S_1} \rightarrow \mathbb{R}^{S_N}$. För att evaluera funktionen för en vektor av invärden \vec{i} låter vi

$\vec{l}_1 = \vec{i}$. Sedan använder vi oss av följande regel, ett lager i taget:

$$v_{n+1,k} = \text{trans} \left(b_{n+1,k} + \sum_{i=1}^{S_n} v_{n,i} w_{n,i,k} \right)$$

eller uttryckt i ord: Ta alla neuroner med anslutningar till den aktuella neuronen (det vill säga alla neuroner i föregående lager). Multiplicera varje sådan neurons värde med vikten mellan denna och den neuron vars värde vi försöker beräkna, och lägg ihop resultaten. Lägg till den aktuella neuronens bias-vikt och beräkna transferfunktionen av resultatet. Detta är den aktuella neuronens värde.

Vi noterar att summan $\sum_{i=1}^{S_n} v_{n,i} w_{n,i,k}$, om vi beräknar den för hela lagret, ges av matrismultiplikation med W_n . Vi ser vidare att additionen av biasvikten $b_{n+1,k}$, om vi utför även den på hela lagret, kan ses som att vi adderar vektorn \vec{b}_{n+1} till hela lagret. Vi får att

$$l_{n+1}^{\vec{}} = \text{trans} \left(\vec{l}_n W_n + \vec{b}_{n+1} \right) \quad (1)$$

om vi tillåter transferfunktionen att agera på en vektor genom att agera på varje element för sig.

För att få fram vårt utvärde upprepar vi denna process tills vi har räknat ut värdet på det sista lagret, $l_N^{\vec{}}$. Detta är funktionens utvärde, $E(\vec{i})$.

1.2.2 Inlärning

Ett artificiellt neuronnät kan tränas att associera invärden med vissa utvärden. Det finns olika sätt att åstadkomma detta, men gemensamt för alla sätt är att det finns en målfunktion som nätet tränas att imitera.

1.2.2.1 Backpropagation Ett vanligt sätt att träna neuronnät är använda *backpropagation*-algoritmen. Algoritmen är relativt komplicerad och finns beskriven utförligt i mycket litteratur (se exempelvis [7], kap. 7), så vi kommer inte beskriva den här. Det bör emellertid nämnas att backpropagation-algoritmen inte använder sig av evolutionära eller genetiska principer, utan i stället utnyttjar neuronnätets matematiska egenskaper.

1.2.2.2 Genetisk algoritm Ett annat sätt att träna neuronnät är genom en *genetisk algoritm* (se exempelvis [5], eller [2] för ett praktiskt exempel). En genetisk algoritm går till så att ett näts vikter och bias-vikter görs ekvivalenta med ett genom, det vill säga en sekvens som motsvarar arvsmassan i den biologiska evolutionen (se avsnitt 2.1.2 för vårt tillvägagångssätt). Från en startpopulation av slumpmässigt genererade nät skapas en ny population genom att genomen i startpopulationen kombineras och muteras för

att bilda nya på ett sätt som liknar biologisk reproduktion. Näten rangordnas enligt en *fitnessfunktion* som avgör hur bra nätet uppfyller målet, och individer med hög fitness har större chans att reproducera. Denna process upprepas ett visst antal gånger eller tills ett önskat maximalt fitness-värde är uppnått. På så sätt får de bättre lämpade näten större inverkan på den framtida populationen än de sämre lämpade, och medelvärdet av populationens fitness-värden ökar.

En genetisk algoritim kan liknas vid avel av djur, då individerna utsätts för ett artificiellt selektionstryck; evolutionen är styrd (se avsnitt 1.1). Till skillnad från vid evolution genom naturligt urval krävs en uttryckligen definierad fitnessfunktion som avgör hur "bra" en individ är.

1.2.2.3 Icke-styrd evolution Ett sätt som liknar en genetisk algoritim, men har fundamentala skillnader mot en sådan, är att träna neuronnät genom icke-styrd evolution, motsvarande naturligt urval. Problemet med detta är att naturligt urval som bekant inte strävar mot något mål, så att använda ordet "träna" i detta sammanhang är missvisande. Vi undersöker i detta arbete bland annat hur icke-styrd evolution kan tillämpas på neuronnät.

Ett drag som skiljer den icke-styrda evolutionen, som träningsmetod sedd, från de tidigare nämnda träningsmetoderna är att den är autonom, det vill säga att näten kan utvecklas utan övervakning. Det finns inte heller någon distinktion mellan träningssteg och tillämpningssteg, det vill säga näten kan utvecklas samtidigt som de utför sin tänkta uppgift. Därför torde ett motiverat användningsområde av den vara att kontinuerligt förbättra en population som är framtagen med någon annan träningsmetod (om detta visar sig möjligt).

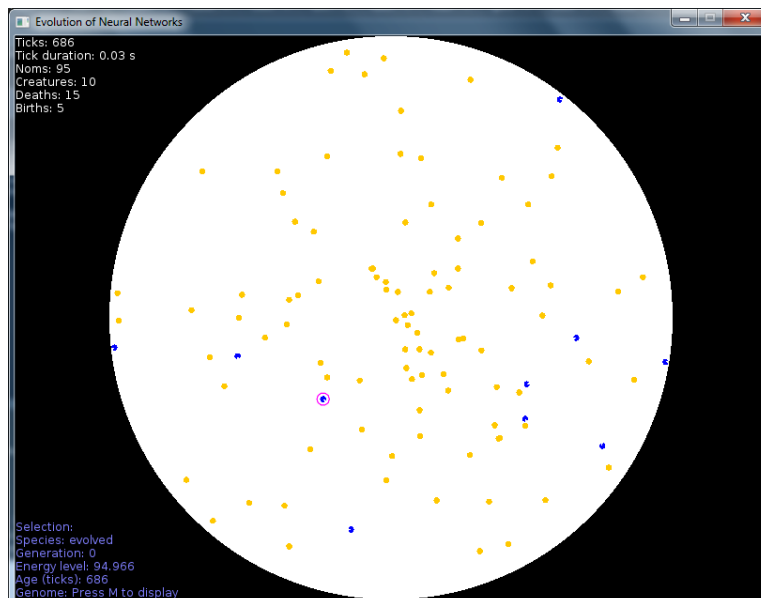
1.3 Syfte

Syftet med detta arbete var att undersöka hur en existerande algoritim kan översättas till ett artificiellt neuronnät och sedan eventuellt förbättras genom icke-styrd evolution.

2 Metod

2.1 Simulationen

Vi använde LÖVE¹, ett ramverk för spelutveckling i skriptspråket Lua², för att konstruera en simulation av evolution av artificiella neuronnät. Figur 2 visar en bild av simulationen. Vi ska nu beskriva simulationen i detalj.



Figur 2: En ögonblicksbild av en körning av simulationen.

2.1.1 Individer

I simulationen finns varelser i form av cirkulära fysikaliska objekt som har en hastighet, påverkas av en friktionskraft proportionell mot hastigheten, samt kolliderar med varandra i fullständigt elastiska kollisioner.

Varje individ har en hjärna. Hjärnan matas kontinuerligt med indata och producerar utdata. Indatan består av

- Den egna hastigheten (storlek och riktning)
- Den egna energinivån
- Position och vinkel till ett antal av de närmaste varelserna
- Position och vinkel till ett antal av de närmaste matbitarna

¹<http://love2d.org>

²<http://www.lua.org>

och utdatan består av

- Den önskade egna accelerationen
- Den önskade egna vinkelhastigheten
- Det önskade tillståndet på varelsens reproduktionssignal (på eller av)

Utdatan produceras genom att indatan matas in i ett neuronnät, och utvärdena från nätet blir utdatan.

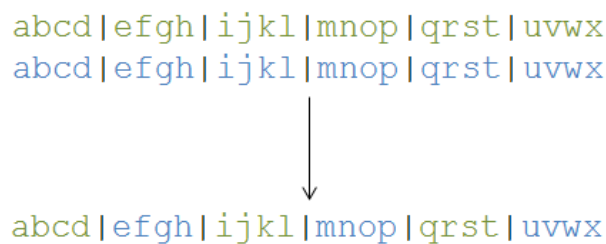
Varje varelse har en energinivå, vilken sjunker när varelsen rör sig eller reproducerar och ökar när varelsen äter matbitar som finns utspridda i världen.

När två varelser, som båda har sin reproduktionssignal aktiv och har tillräckligt hög energinivå för reproduktion, kommer inom ett bestämt avstånd från varandra dras en mängd energi från båda föräldrar, och de reproducerar och bildar en ny varelse. Avkommans hjärna är en kombination av de båda föräldrarnas hjärnor (detta beskrivs i nästa avsnitt).

2.1.2 Genetik

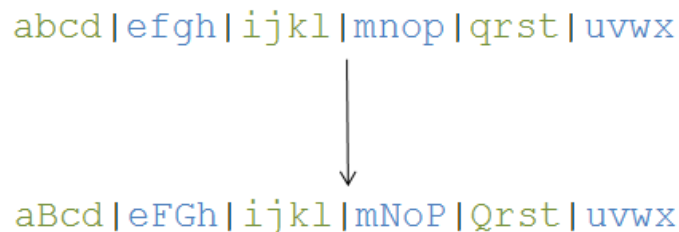
Alla neuronnät i simulationen har samma antal lager och samma antal neuroner i varje lager (alla har samma S). Därför kan varje neuronnät entydigt beskrivas av en lista G av alla vikter följt av alla bias-vikter (denna lista innehåller alltså alla värden i W och B och har längd $\sum_{n=1}^{N-1} (S_n S_{n+1}) + \sum S$). En varelses *genom* definieras som listan G för varelsens neuronnät. När två varelser reproducerar kombineras deras genom och bildar avkommans genom i två steg: kombination och mutation.

När genomen kombineras bildas det nya genomet genom att grupper av en given storlek tas från varannan förälder. Figur 3 illustrerar processen.



Figur 3: Kombination av två genom.

När genomen har kombinerats muteras det resulterande genomet. Mutationen sker genom att varje gen (varje "bokstav" i genomet, det vill säga varje vikt eller bias-vikt) med en viss sannolikhet ändras med en slumpmässig positiv eller negativ ändring. Figur 4 visar hur detta kan se ut.



Figur 4: Mutation av det resulterande genomet. En versal motsvarar att genen has muterat.

Det resulterande genomet översätts sedan till avkommans neuronnet, det vill säga listorna W och B extraheras från genomet.

2.2 Översättning av designad algoritm

För att kunna svara på frågeställningen hur vi kan översätta en designad algoritm (med algoritm menas beteendemönster hos en varelse) till ett neuronnet krävdes först en designad algoritm att försöka översätta. Vi skrev en enkel (helt klart suboptimal; hur bra algoritmen är är inte konceptuellt viktigt). Algoritmen bestod i stora drag i att, med reproduktionssignalen aktiverad så länge individen har tillräckligt mycket energi för reproduktion, gå mot den närmaste matbiten med en hastighet och en vinkelhastighet proportionell mot avståndet respektive vinkelavståndet till denna.

Vi implementerade en backpropagation-algoritm (som beskriven av bland andra [3]) för att producera neuronnet som imiterade denna designade algoritm, i avsikt att sedan låta dessa utgöra startpopulationen i icke-styrd evolution. Vi lyckades emellertid inte få denna algoritm att konvergera.

Vi erhöLL samma resultat när vi implementerade en genetisk algoritm i samma syfte. Dessa resultat diskuteras vidare i avsnitt 3.3.

2.3 Icke-styrd evolution av slumpmässigt genererade nät

Eftersom översättningen av den designade algoritmen misslyckades lät vi startpopulationen i simulationen bestå av varelser med slumpmässigt genererade genom.

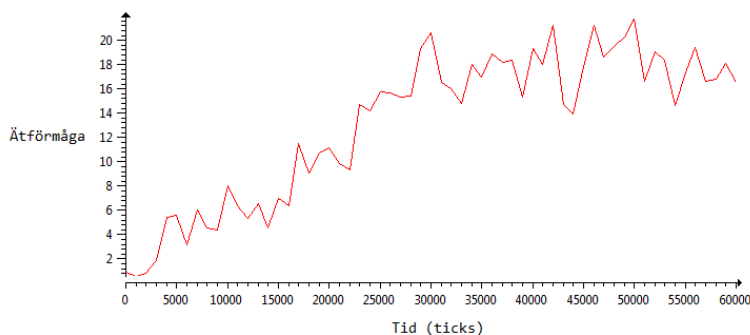
2.3.1 Allmänna observationer

Det var, vid upprepade körningar av simulationen, tydligt att effekterna av den naturliga evolutionsprocessen var synliga med blotta ögat. Från början var varelsernas beteende högst slumpmässigt, men efter att simulationen hade körts en tid var det tydligt att majoriteten av varelserna sökte sig till de utspridda matbitarna.

Man kunde också notera att populationen i majoriteten av körningarna konvergerade mot att antingen röra sig framåt eller bakåt (varelserna gavs nämligen möjlighet att accelerera framåt eller bakåt). Detta tolkas som ett exempel på genetisk drift i form av flaskhalseffekt (se [6]).

2.3.2 Empirisk undersökning av förbättring av ätförmåga

Vi utförde mätningar för att undersöka hur varelsernas ätförmåga utvecklades över tiden. För att mäta den övergripande ätförmågan hos populationen mellan tidpunkterna t och $t + \Delta t$ (tidpunkterna mäts i *ticks*, simulationens minsta tidsintervall) betraktade vi två värden: antalet matbitar som åts i intervallet (m) samt medelantalet matbitar som existerade i intervallet (M , vilket motsvaras av $\frac{\int_t^{t+\Delta t} n(\tau) d\tau}{\Delta t}$ om $n(\tau)$ är antalet matbitar som fanns vid tiden τ). Den totala ätförmågan definierade vi då som $\frac{m}{M}$.



Figur 5: Ätförmågan hos en population med 5, 4 och 3 neuroner i respektive lager.

Figur 5 visar hur ätförmågan varierade för en population, från början slumpmässigt genererad, bestående av nät med $S = [5 \ 4 \ 3]$ (en input för närmaste matbit, inga för närmaste varelser). För så enkla neuronät gick evolutionen att direkt observera i termer av den ovan definierade ätförmågan. För större nät kunde beteendeändringen observeras subjektivt, men att kvantifiera observationerna visade sig problematiskt. Värdena fluktuerar men en uppåtgående trend är tydlig. Det är värt att notera att ökningen avstannar efter cirka 30000 ticks.

3 Diskussion

3.1 Generaliteten hos undersökningen

Denna undersökning behandlar ett mycket specifikt fall av en mycket generell frågeställning. Även i kontexten av vår typ av simulation (varelser som kan äta och reproducera) finns det otaliga parametrar som kan varieras och som alla potentiellt har inverkan på evolutionsprocessens effektivitet. Vidare bör nämnas att det önskade beteendet (att äta maximal mängd mat) i vårt fall är nära knutet till individernas överlevnad och därmed även till genernas spridning. Styrkan hos denna "indirekta feedback" från målfunktionen är naturligtvis avgörande för resultatet.

3.2 Mätningen

Mätningen av ätförmåga ger en översiktlig bild av hur en population av neuronnät kan undergå evolution. Den visar tydligt att evolution *kan* förekomma, men man bör vara försiktig med att dra generella slutsatser ur den (se ovan). De parametrar som användes var noggrant utvalda genom försök och misstag för att möjliggöra evolution.

Det bör också noteras att en del av ökningen i ätförmåga kan härledas till en ökning i populationens storlek, vilken började på 20 och uppnådde ett stabilt medelvärde på 30 efter cirka 10000 ticks. Detta är en marginell felkälla, men bör nämnas.

En annan felkälla i mätningen är valet av storhet som mättes. Storheten tog exempelvis ingen hänsyn till populationens storlek (detta bör inte ha inverkat nämnvärt, då storleken var relativt konstant på 30 individer) eller varelsernas energiförbrukning. Olika val av storhet ger svar på olika frågor om populationens beteende.

3.3 Konvergens hos backpropagation och genetisk algoritm

Man kan spekulera runt varför de båda träningsalgoritmerna misslyckades. En uppenbar möjlighet är att något programmeringstekniskt fel introducerades när programkoden skrevs. Möjligheten existerar också att de använda neuronnäten var för små och därför saknade möjlighet att imitera den designade algoritmen. En tredje möjlighet är att algoritmens parametrar (exempelvis populationsstorleken i den genetiska algoritmen eller inlärningshastigheten i backpropagation) var illa anpassade för ändamålet. Att träna artificiella neuronnät är vid sidan av att vara en vetenskap också ett hantverk som vi saknar erfarenhet i. Det är troligt att mer erfarna personer inom fältet hade kunnat producera neuronnät som imiterade den designade algoritmen.

Att träningen misslyckades har betydelse för vilka slutsatser som kan dras av undersökningen. Det står klart att artificiella neuronnät *kan* tränas med hjälp av backpropagation eller en genetisk algoritm; detta är föremål för

mycket forskning och exemplifieras av bland andra [5]. Av undersökningen framgår också att icke-styrd evolution, åtminstone under vissa förhållanden, kan öka en populations lämplighet för en viss uppgift. Vi har däremot inte uttryckligen visat att en redan existerande algoritm kan förbättras genom översättning till neuronät följt av icke-styrd evolution.

3.4 Slutsats

Vi har sett att icke-styrd evolution av en population av artificiella neuronät är principiellt möjlig, och potentiellt användbar som träningsmetod eller förbättringsmetod av redan tränade neuronät. Det finns emellertid många av oss utforskade faktorer som kan påverka användbarheten hos detta tillvägagångssätt.

3.5 Förslag till vidare undersökningar

Utfallet av denna undersökning lämnar många relaterade frågeställningar obesvarade, och många nya undersökningar skulle kunna göras för att undersöka ämnet närmare. Exempelvis kan en fungerande genetisk algoritm eller backpropagation-algoritm, eller någon annan lämplig träningsalgoritm, implementeras för att undersöka effekterna av icke-styrd evolution på redan tränade neuronät.

En faktor som tycks avgörande för framgången hos nät är deras storlek. Denna faktor kan undersökas på många sätt. Man kan jämföra framgången hos evolutionen vid olika nätstorlekar; man kan också låta nätets arkitektur (däribland dess storlek) ingå i genomet (se [4], 2.3 för exempel).

I detta arbete användes enbart den enklaste formen av artificiella neuronät, feedforward-nät. Vidare undersökningar skulle kunna undersöka möjligheterna med andra typer av nät.

Referenser

- [1] Berg, J. M. (2002) *Biochemistry*. Tillgänglig på Internet: <http://www.ncbi.nlm.nih.gov/books/NBK21154/> New York: W H Freeman.
- [2] Buckland, M. (utgivningsår okänt) *Genetic Algorithms in Plain English* <http://www.ai-junkie.com/ga/intro/gat1.html>
- [3] Heaton, J. (2010) *Neural Network Training*. <https://www.youtube.com/watch?v=CVJ0seIJnww> samt de följande tre delarna.
- [4] Mitchell, M. (1996) *An Introduction to Genetic Algorithms*. MIT Press. <http://books.google.se/books?hl=sv&lr=&id=0eznlz0TF-IC&oi=fnd&pg=PP9&dq=genetic+algorithm+introduction&ots=>

sfrG23-cRj&sig=q3vimN8D7iYabNcBlExFpT8xQ8I&redir_esc=y#
v=onepage&q=genetic%20algorithm%20introduction&f=false

- [5] Montana, D. J., Davis, L. (1989) *Training Feedforward Neural Networks Using Genetic Algorithms* Tillgänglig på Internet: <http://ijcai.org/Past%20Proceedings/IJCAI-89-VOL1/PDF/122.pdf>
- [6] O'Neil, D. (1997–2014) *Small Population Size Effects* http://anthro.palomar.edu/synthetic/synth_5.htm
- [7] Rojas, R. (1996) *Neural Networks - A Systematic Introduction*. Berlin: Springer-Verlag. Tillgänglig på Internet: <http://page.mi.fu-berlin.de/rojas/neural/>
- [8] Stefanov, E. (2014) *Neural Networks*. <http://www.emilstefanov.net/Projects/NeuralNetworks.aspx>