

Artificiella neurala nätverk som lösning på segmenteringsproblemet vid gestigenkänning

Joakim Ernhagen

**Artificiella neurala nätverk som lösning på segmenteringsproblemet vid
gestigenkänning**

Examensrapport inlämnad av Joakim Ernhagen till Högskolan i Skövde, för
Kandidatexamen (B.Sc.) vid Institutionen för kommunikation och information.
Arbetet har handletts av Henrik Engström.

2008-05-29

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 neurala nätverk som lösning på segmenteringsproblemet vid gestigenkänning

Joakim Ernhagen

Sammanfattning

Ett av de stora problemen med gestigenkänning är det som kallas segmenteringsproblemet. Segmenteringsproblemet är problemet att hitta var en gest börjar och var den slutar, d v s att segmentera upp gester. I detta arbete testas en lösning till detta problem då gester består av figurer ritade i luften. Lösningen går ut på att använda två stycken artificiella neurala nätverk av strukturen *FeedForward*. Det ena nätverket tränas på att känna igen när handen accelererar från stillastående, vilket representerar starten av en gest, och det andra tränas på att känna igen när handen saktar ner till stillastående, vilket representerar slutet av en gest. Datan till de artificiella neurala nätverken kommer från ett *sliding window* på ett antal tidssteg där hastighets- och accelerationsförändringar läses av som skalärer.

Nyckelord: Gestigenkänning, Rörelseigenkänning, Segmenteringsproblemet, Gametrack

Innehållsförteckning

1	Introduktion	1
2	Bakgrund	2
2.1	Allmänt om artificiella neurala nätverk (ANN).....	3
2.2	Allmänt om gestigenkänning	5
2.3	Lokalisera gester	6
2.4	Följa gester.....	7
2.5	Behandla datan.....	7
2.6	Klassificering av gester.....	8
3	Problembeskrivning	10
3.1	Tidsaspekter	10
3.2	Träning av nätverken	11
3.3	Utvärdering	11
4	Metod	13
4.1	Tidsaspekter	13
4.2	Träning av nätverken	13
4.3	Utvärdering	14
5	Resultat	15
5.1	Tidsaspekter	15
5.2	Träning av nätverken	15
5.3	Utvärdering	15
5.4	Analys	16
5.4.1	Kantiga kontra icke kantiga gester	17
5.4.2	Sammanfattning av analysen.....	17
6	Slutsats	19
6.1	Diskussion.....	19
6.1.1	Tidsfönstret.....	19
6.1.2	Hur fler gester skulle kunna hittas.....	19
6.1.3	Handkontrollen till Nintendo Wii som input apparatur	20
6.2	Framtida arbete	21

1 Introduktion

Det finns gott om apparatur för att ge input till en dator, t ex mus, tryckkänsliga skärmar och tangentbord. Gemensamt för dessa är att användaren indirekt ger datorn instruktioner på så sätt att datorn övervakar användarens påverkan på input apparaturen, som t ex att användaren trycker ner "Enter" på tangentbordet. Ett alternativ till den indirekta instruktionsgivningen är att datorn istället använder någon apparatur till att övervaka användaren och sedan tolkar användarens gester. Det problem som uppstår då är att datorn lätt kan feltolka vad användaren vill då det inte är lika självklart vad som menas med en viftande hand som med ett tryck på "Enter" på tangentbordet (Harling, 1993).

Två vanliga tekniker för att övervaka användaren är med sensorhandskar eller med kameror. Ett stort problem för dessa tekniker är att lyckas lokalisera när en gest utförs. Anledningen till detta är att inga signaler talar om när en gest tar sin början eller när den når sitt slut då detta måste utvinnas från själva gesten. Detta att försöka hitta början och slut av en gest är ett känt problem som går under benämningen segmenteringsproblemet (Harling & Edwards, 1997).

Det finns 4 olika grupper av gester, en av dessa är *SPDL, Static Posture Dynamic Location*. Denna grupp innehåller gester där handformationen är densamma genom hela gesten och den tillsammans med handens rörelse ger gestens betydelse (Harling & Edwards, 1997). Det är en grupp gester inom denna grupp som detta arbete tar upp, en grupp där handformationen är densamma för alla gester och därför kan bortses ifrån, vilket innebär att gesterna består av figurer ritade i luften.

Artificiella neurala nätverk är kända för sin förmåga att producera specifik utdata på viss indata vilket betyder att de kan känna igen mönster. För att ett ANN ska kunna producera rätt utdata så tränas nätverket med en mängd exempel med indata och önskad utdata. Genom att träna ett ANN med indata med brus, d v s med viss variation men med samma önskade utdata, så blir nätverket mer generellt och känner igen mönster även om de har små avvikelser.

Detta arbete kommer att använda sig av *Gametrack™ USB Direct Motion Capture System* för att läsa av utförda rörelser. Detta är en kontroll vars värden kan räknas om så att en riktningsvektor fås mellan kontrollen och handen vilket betyder att figurer ritade i luften kan utvinnas. Målet är att med datan som fås från denna kontroll få två artificiella neurala nätverk att peka ut start- och slutpunkter för gester. Rörelsedatan kommer att sparas till fil och algoritmen behöver inte vara optimerad på så vis att den ska hinna med att läsa av och avgöra om något är start- eller slutpunkt med en viss frekvens. Datan i filerna kommer att vara sparad med viss frekvens och när algoritmen utvärderas kommer det antas att den lyckas arbeta med rätt frekvens oavsett om så är fallet eller inte.

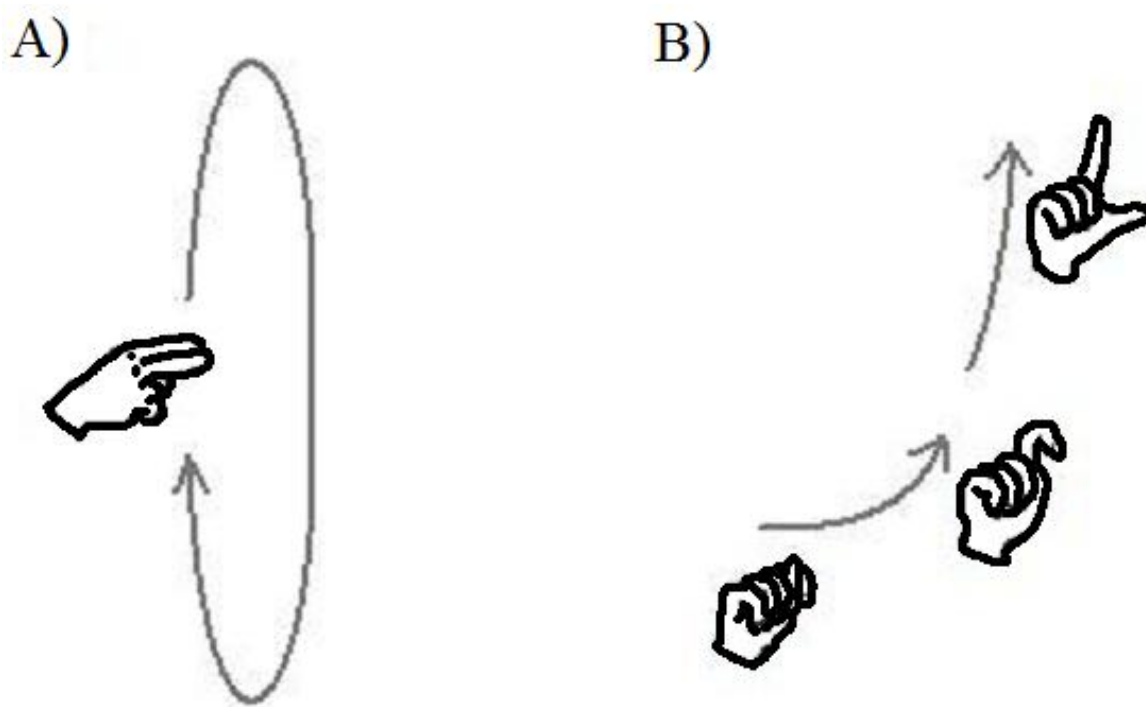
Utvärderingen av algoritmen går ut på att samla statistik över hur många rätt tidpunkter som pekas ut, fel tidpunkter som pekas ut samt hur många gester som hittas. Den data vilken utvärderingen kommer göras på är sådan att den inte använts till träningen av nätverken.

2 Bakgrund

Med rörelseigenkänning menas förmågan att utifrån ett antal rörelserelaterade observationer dra slutsatser om vilken typ av rörelse som avses (Wickman, 2007). Gestigenkänning är ett område inom rörelseigenkänning. En gest består av ett antal poser, vars position och rörelse har betydelse, där det med pose menas en statisk handformation där fingersättning har betydelse (Harling & Edwards, 1997).

Enligt Harling & Edwards (1997) finns det fyra typer av gester vilka har olika svårighetsgrad när det gäller att bestämma vilken typ av gest som avses. Dessa fyra typer av gester i svårighetsgradsordning med lättast att klassificera först är:

- SPSL – static posture, static hand location.
Fingersättningen är statisk och handens position är statisk.
- DPSL – dynamic posture, static hand location.
Fingrarnas rörelse har betydelse, handens position är statisk.
- SPDL – static posture, dynamic hand location.
Fingersättningen är statisk, handens rörelse påverkar gestens betydelse.
- DPDL – dynamic posture, dynamic hand location.
Fingrarnas rörelse har betydelse, handens rörelse påverkar gestens betydelse.



Figur 1. Exempel på gester, gesterna i figuren har ingen specifik betydelse utan är endast ämnade som illustrationer av hur gester kan se ut. A) Ett exempel på en gest ur gruppen *SPDL*, fingersättningen är den samma under hela gesten och handens position ändras. B) Ett exempel på en gest ur gruppen *DPDL*, fingersättningen ändras under gestens gång tillsammans med handens position.

Högskolan i Skövde driver ett projekt för att skapa spel vars syfte är att användas vid rehabilitering av strokepatienter för att ge dessa mindre monotona

rehabiliteringsmetoder. Spelen ska finnas att spela på en spelkonsol vilken använder sig av *Gametrack™ USB Direct Motion Capture System* för input. Detta är en kontroll vilken har två utdragbara snören. Fästet till ett sådant snöre sitter på en joystick liknande del av kontrollen vilket gör att inte bara avståndet som snöret dras ut kan avläsas utan även vilken riktning detta sker i (se Figur 2).



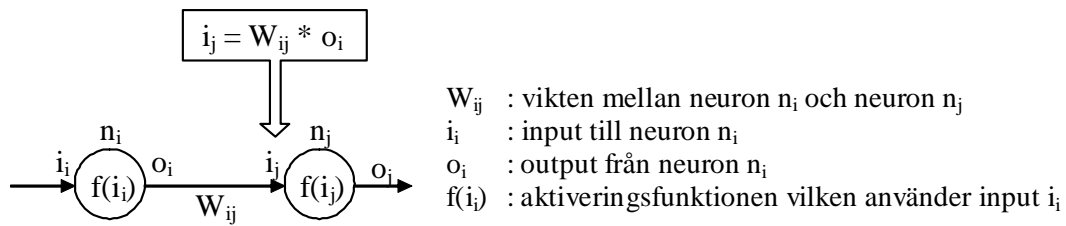
Figur 2. Ett foto av *Gametrack* kontrollen vilken används i detta arbete.

Syftet med detta arbete är att skapa en algoritm vilken kan användas i projektet som bedrivs av Högskolan i Skövde, därför kommer samma kontroll att användas för detta arbete som används av den spelkonsol Högskolan i Skövde tagit fram och detta är även anledningen till att gesterna i detta arbete är anpassade efter denna kontrollers möjligheter, nämligen figurer ritade i luften.

2.1 Allmänt om artificiella neurala nätverk (ANN)

Artificiella neurala nätverk är inspirerade av hur nervsystemet fungerar och består av ett antal neuron $n_1, n_2, \dots, n_{i-1}, n_i$ vilka är sammankopplade med ett antal viktade och riktade kopplingar. En koppling mellan neuronerna n_i och n_j betecknas W_{ij} . Varje neuron i nätverket utgör ett enkelt beräkningselement som beräknar en viss output o_i givet en input i_i . Input till ett neuron är output från föregående neuron multiplicerat med vikten på kopplingen mellan dessa (Buckland, 2002) (se Figur 3).

Bakgrund

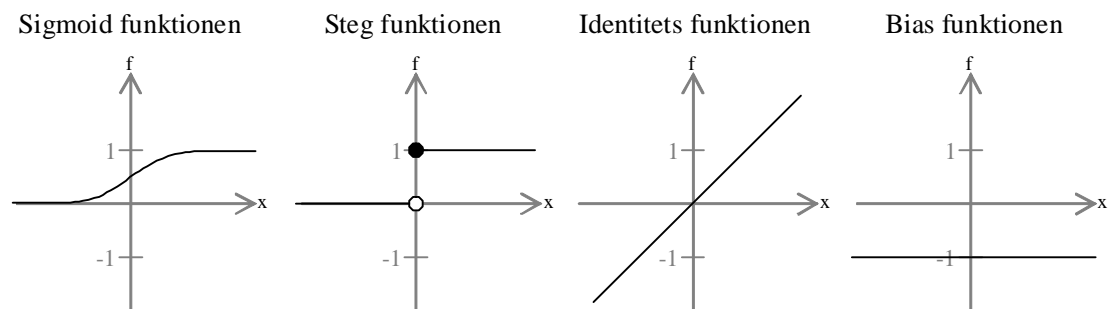


Figur 3. Hur input till ett neuron beräknas utifrån output från ett annat tillsammans med kopplingens vikt.

Det finns tre typer av neuron, input neuron, output neuron och gömt neuron.

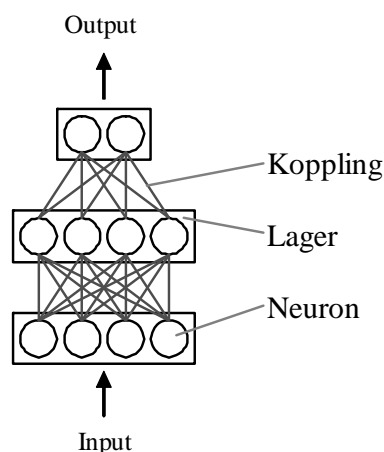
- Input neuron – får sin data från miljön, kan ses som en sensor som känner av miljön.
- Output neuron – påverkar miljön genom att värdet läses av och används.
- Gömt neuron – varken input- eller output neuron, sitter inne i nätverket och möjliggör mer komplexa beteenden.

Beräkningsmomentet för ett neuron utgörs av en aktiveringsfunktion $f(x)$. Det finns flera olika aktiveringsfunktioner (se Figur 4).



Figur 4. De olika aktiveringsfunktionerna för en neurons beräkningsmoment.

Artificiella neurala nätverk består oftast av ett antal lager vilka innehåller en eller ett antal neuron. I ett sådant nätverk är ett lager input lager och innehåller endast input neuron och ett lager output lager och innehåller endast output neuron. Inne i nätverket kan det finnas ett eller ett antal lager innehållande endast gömda neuron. När lager av neuron används så är varje neuron i ett lager kopplad till samtliga neuron i nästa lager (se Figur 5).



Figur 5. Ett exempel på ett ANN uppbyggt av lager av neuron.

Det finns olika typer av strukturer för ANN. En vanlig nätverksstruktur är *FeedForward* nätverk. Dessa nätverk består av ett antal lager av neuron. Var lager utom output lagret innehåller ett extra neuron av typen bias, utan bias neuron fungerar inte ett *FeedForward* nätverket så som det är tänkt. Styrkan i ett *FeedForward* nätverk ligger i att ett sådant kan lära sig att ge specifik output på given input, något som går till på så vis att exempel körs med input och vilken output den ska ge och vikterna ändras för att minska felet så att den output som ges blir mer lik den i exemplet.

För mer information om ANN, se Buckland (2002).

2.2 Allmänt om gestigenkänning

Det finns fyra steg vilka genomgås i given ordning vid gestigenkänning (Harling & Edwards, 1997).

- Lokalisera när gesten utförs.
- Följa gesten från dess start till dess slut.
- Behandla datan från gesten så att den går att jämföra med kända gester.
- Jämföra gesten med kända gester för att klassificera den.

Dessa punkter har dock olika svårighetsgrad beroende på vilken typ av gestigenkänning det gäller. När det t ex gäller att rita figurer på en handdator så är segmenteringen, att lokalisera när gesten utförs, inget problem då lösningen ges via att pennan trycks mot skrivytan vid början av en gest och lyfts vid slutet. Samma sak gäller för steget med att följa en gest från dess början till dess slut om endast statiska stillastående handformationer utgör gesterna, detta då dessa gester saknar förändringar att följa.

Vart och ett av dessa steg är ett eget forskningsområde och kan studeras separat. T ex så har Harling & Edwards (1997) fokuserat på att lokalisera när gester utförs, Fels & Hinton (1993) inriktat sig på att lokalisera gester och att klassificera dessa och Harling (1993) undersökt variationer av ANN för att utvärdera vilken som ger den bästa klassificeringen (minst antal missade kända gester).

2.3 Lokalisera gester

När det gäller att lokalisera gester så är ett problem att hitta när en gest börjar och när den avslutas. Detta problem går under benämningen segmenteringsproblemet och är ett viktigt problem att lösa då det utan denna segmentering inte skulle gå att klassificera gester.

Den enklaste metoden för att lösa segmenteringsproblemet är att en signal skickas för att indikera starten av en gest och att en signal skickas för att indikera slutet av denna gest. Denna teknik används av t ex handdatorer där startsignalen ges när pennan trycks mot skrivytan och stoppsignalen ges när pennan lämnar skrivytan. Harling (1993) använde sig av en signalbaserad teknik för segmenteringen då syftet med hans arbete var att utvärdera olika ANN strukturer för att få en så bra klassificering som möjligt av statiska handformationer. Vad Harling gjorde var att han lät användaren trycka ner en specifik knapp på tangentbordet när den statiska gesten var formad med handen och denna knapptryckning talade om för systemet att handens värden skulle läsas av. Signalbaserade lösningar är dock inga bra lösningar då en idé med att få datorer att känna igen gester enligt Harling (1993) är för att kunna skapa nya användargränssnitt vilka övervakar användaren istället för att övervaka t ex ett tangentbord.

Harling & Edwards (1997) använde sig av hur handen fungerar för att presentera en lösning på segmenteringsproblemet. Deras undersökningar visade att människor spänner handen mer under medvetna gester då gesten är en statisk handformation och att handen genomgår ett mer avslappnat stadie vid ett byte mellan två medvetna gester. Harling och Edwards fick fram hur pass spänd handen var genom att se på fingrarnas böjning, fingrarna på en hand i helt avslappnat tillstånd har nämligen en viss böjning. En uträkning gjordes för alla fingrarna för att få fram hur pass spänd handen var och detta används på så sätt att när handens spänningsvärde var inom ett visst avstånd från avslappnat läge så markerade detta en gräns mellan två gester.

En mycket intressant lösning på segmenteringsproblemet är den som Fels & Hinton (1993) testade. Arbetet gick ut på att känna igen gester vilka representerade ord. Dessa ord skulle sedan uttalas av en dator med syntetisk röst. Gesterna bestod av ett basord, vilket markerades med en statisk handformation, en ändelse på basordet, vilken markerades genom att föra handen i en av sex riktningar och tillbaka, samt hur snabbt och hur högt ordet skulle uttalas, vilket avgjordes av hur snabb respektive stor fram och tillbaka rörelsen var. Fels och Hinton använde sig av 5 olika ANN för att lösa uppgiften, ett för att avgöra basordet, ett för att avgöra ändelsen, ett för att avgöra hur snabbt uttalandet skulle göras, ett för att avgöra ljudstyrka och ett för att registrera när en gest utfördes. Fels och Hinton ville hitta en tidpunkt då de fyra ordavgörande nätverken skulle matas med data och denna tidpunkt fick bli när handen var i slutet av utåt rörelsen. Det som var specifikt med denna tidpunkt var att handen alltid saktar ner när användaren ska byta riktning till tillbaka rörelsen.

Det ANN vilket användes av Fels och Hinton för att avgöra tidpunkten för mätning fick som input ett fönster på 10 tidssteg med 5 värden för var tidssteg (50 värden totalt). De fem värdena från varje tidssteg var förändringen i x, y och z, handens hastighet som skalär och handens acceleration som skalär. Detta nätverk hade endast ett output neuron, vilket tolkades som en etta eller en nolla där en etta betydde att en mätpunkt hittats och en nolla betydde att ingen mätpunkt hittats. Nätverket tränades med exempel vilka letades upp för hand i sparad rörelsedata och när det testades så

hittades 94,6% av alla mätpunkter och av resterande tidssteg pekades 0,1% felaktigt ut som mätpunkter.

2.4 Följa gester

Beroende på gester och teknik så är problemet med att följa en gest från början till slut av olika svårighetsgrad. Att t ex följa en gest på en handdator där det endast är en rörelse i 2 dimensioner som ska följas är mycket lättare än att läsa av dynamiska handformationer. Teknikmässigt skiljer sig svårighetsgraden i att det är lättare att få ut korrekta värden för hur en handformation förändras om en handske med sensorer används än om handen ska observeras med kamera.

I många fall används handskar med sensorer för att avläsa användarens hand, arbeten vilka använder handske är Fels & Hinton (1993), Harling (1993) och Harling & Edwards (1997). Det finns små variationer i hur dessa tre arbeten valt att göra om handskens mätningar till värden vilka kan matas in i ett ANN. T ex så i vissa fall tas ingen hänsyn till handledens vridning och i vissa fall tas både handledens vridning och handens vinkel in i beräkningen. Gemensamt för alla dessa tre arbeten är dock att endast statiska stillastående handformationer kontrolleras, d v s att en gest aldrig består av mer än en pose och handen inte förflyttas, så det finns ingen gest att följa då start och slut inträffar i samma tidpunkt.

Viblis & Kyriakopoulos (1999) försöker att avläsa handens konturer med en kamera för att få information om handformationen. I arbetet testas en algoritm vid namn *Snakes* vilken enligt Viblis och Kyriakopoulos är väldigt bra på att hitta konturer och följa dessa om de förflyttas från en bild till nästa. Deras undersökningar visade dock att *Snakes* inte är problemfritt. Algoritmen har svårt att hitta konturerna runt handen om handen inte befinner sig väldigt nära kameran och att området vilket ringas in av *Snakes* ofta krymper till ett område som inte markerar handens konturer utan endast ett område innanför dessa, vilket gör det problematiskt att se hur fingrarnas positioner förändras.

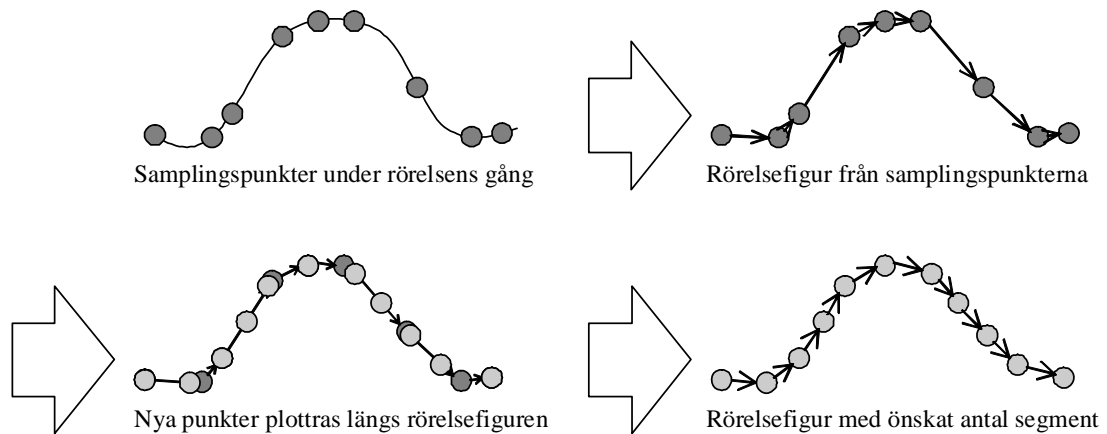
2.5 Behandla datan

Att behandla datan från den utförda gester är inte alltid något stort steg. När gester endast består av en mätning som är fallet i Fels & Hinton (1993), Harling (1993) och Harling & Edwards (1997) så behövs endast en tolkning av den avlästa datan göras. De handskar som används mäter böjningen av fingrarna och denna information räknas så att det blir ett värde i ett visst intervall, t ex $[0,1]$. Samma sak gäller för rotation av handled och lutningen på handen då dessa används. Dessa värden är sedan färdiga att matas in i ett ANN.

Steg med att behandla datan blir relevant först när gester är dynamiska, så som är fallet för en handdator där en gest är en rörelse i 2 dimensioner. Rabin (2006, kap 6.4) beskriver ett sätt att behandla datan så att storleken på gester, hastigheten den gjorts i samt var i koordinatsystemet den utförs inte spelar någon roll. Detta går till på så sätt att gestens beståndsdelar ses som koordinater i ett 2 dimensionellt plan. För att få bort relevansen av var dessa koordinater befinner sig så används riktningsvektorerna mellan dessa för att beskriva gester.

Efter att Rabin har fått fram riktningsvektorerna mellan koordinaterna så kontrolleras den totala längden av alla dessa. Denna längd delas sedan upp i ett önskat antal segment och längden för ett sådant segment sparas undan. Riktningsvektorerna följs sedan och nya koordinater sätts ut med den sparade längden som intervall. När detta

är gjort tas riktningsektorerna mellan dessa nya koordinater fram och normaliseras och blir den behandlade datan vilken tack vare normaliseringen och stegandet är helt oberoende av storleken på den utförda gester och hastigheten den gjordes med (se Figur 6).



Figur 6. En förklaring av hur en gest följs och sparas för att sedan behandlas i Rabin (2006, kap 6.4).

2.6 Klassificering av gester

När gester är lokaliserade, följda och sparade och den sparade gestens data har behandlats för att få önskat utseende så är klassificeringen det ända som är kvar. Enligt Harling (1993) finns det två typer av klassificeringsmetoder, en metod vilken bygger på att systemet lär sig känna igen gester, med ANN, och en metod vilken bygger på att de gester vilka ska kännas igen finns lagrade i någon typ av databas.

Metoder vilka bygger på lärande system används i Harling (1993) och Fels & Hinton (1993) medan metoder vilka bygger på att gester sparas i databaser används i Rabin (2006, kap 6.4). ANN är populärare än databaser, anledningen till detta är antagligen att de, enligt undersökningar gjorda av Harling (1993) av tidigare arbeten, känner igen gester bättre än egenskriverna jämförelsealgoritmer.

Harling (1993) använder sig av lika många output neuron som det finns kända gester. Detta är ett beslut Harling tagit utifrån analys av andras arbeten där det visade sig att i samtliga tidigare arbeten vilka var med i analysen så användes denna teknik, ett val vilket gjorts för att minimera misstolkning av output. När nätverket tränas så ska datan från en gest ge en etta på det output neuron vilket representerar gester och en nolla på alla de andra. Denna teknik används även av de nätverk Fels & Edwards (1993) använder sig av för att klassificera basordet och för att bestämma ändelsen av basordet i sitt arbete. Basordsnätverket har lika många output neuron som det finns basord och ändelsenätverket har lika många output neuron som det maximalt finns ändelser till ett basord.

Rabin (2006, kap 6.4) använder sig inte av några ANN för att klassificera en utförd gest. Det Rabin använder sig av är mallar bestående av en serie normaliserade riktningsektorer. Efter att datan i Rabins algoritm har behandlats så består även denna av en serie normaliserade riktningsektorer. Dessa vektorer går att jämföra med mallarnas riktningsektorer genom att ta dot-produkten mellan gestens och mallens vektorer en och en från början till slut, addera dessa med varandra och sedan dividera med antalet vektorer i mallen. Detta ger ett värde i intervallet $[-1,1]$ där -1 betyder att

Bakgrund

gesterna är helt olika varandra och 1 betyder att de är helt lika. För att få ett bättre och mer lättläst värde vill dock Rabin ha likheten i intervallet [0,1], vilket han enkelt gör genom att addera 1 och sedan dividera med 2 vid var dot-produkts uträkning. (se Figur 7).

$$\text{Likhet} = \frac{\sum_{i=0}^{n-1} \frac{r_i * f_i + 1}{2}}{n}$$

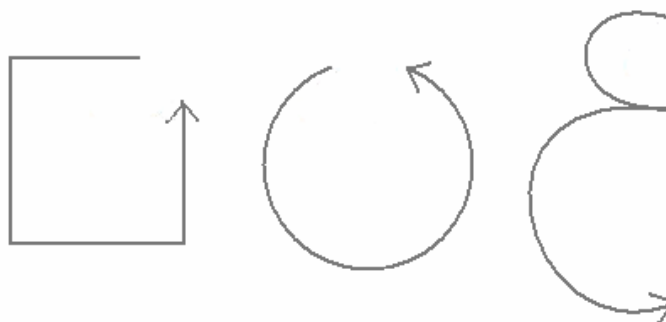
n : antal riktningsvektorer
 r_r : riktningsvektor från rörelsen
 f_r : riktningsvektor från mall (facit)

Figur 7. Formeln Rabin använder för att beräkna likheten mellan en gest och en mall.

Efter att en gest jämförts med alla mallar så tas den mall fram vilken gesten var mest lik. Likhetsvärdet mellan gesten och denna mall kontrolleras sedan mot ett givet tröskelvärde som talar om hur lik en mall gesten måste vara för att gesten ska klassas som den gest mallen beskriver och om så är fallet så har den utförda gesten hittats.

3 Problembeskrivning

Syftet med detta arbete är att lösa segmenteringsproblemet vid igenkänning av en viss typ av gester i gruppen *SPDL (Static Posture Dynamic Location)*. Den typ av gester inom denna grupp vilken ska tas upp är för en enda godtycklig pose, vilket betyder att posen saknar betydelse då alla poser är ekvivalenta, och endast handens rörelser har betydelse, vilket ger att gesterna är figurer ritade i luften (se Figur 8). Anledningen till detta är att den input apparatur, *Gametrack* kontrollen, som ska användas läser hur utdragna snörena på kontrollen är samt vilken riktning de är utdragna i, vilket kan räknas om till koordinater i 3 dimensioner.



Figur 8. Gester vilka består av figurer ritade i luften. Fingersättningen är inte med i gestens beskrivning då denna är densamma för alla gester och därför kan bortses ifrån.

Den lösning vilken ska användas för att lösa segmenteringsproblemet i det här fallet då endast handens rörelser övervakas måste på något sätt vara beroende av hur handen rör sig. Ett arbete vilket tidigare använt sig av rörelser för att lokalisera en specifik händelse är Fels & Hinton (1993) där tidpunkten för en mätning hittas genom att låta ett ANN övervaka handens hastighet och acceleration som skalärer i ett tidsfönster på 10 tidssteg för att avgöra när handen saktar ner för att ändra riktning. Det ANN vilket Fels och Hinton använder sig av är ett *FeedForward* nätverk med ett gömt lager. Detta ANN har 50 input neuron av typen identitet, 10 gömda neuron av typen sigmoid och 1 output neuron av typen sigmoid. Den ensamma output neuronen läses av och jämförs med ett tröskelvärde vilket säger om det är en etta eller en nolla, d v s om det är en mätpunkt eller om det inte är det.

Fels och Hinton's lösning för att lokalisera en specifik händelse är det som kommer att användas i detta arbete för att lösa segmenteringsproblemet. Det kommer dock att krävas två olika ANN istället för endast ett då två olika händelser måste sökas. Den första är när handen accelererar för att påbörja en gest och den andra är när handen saktar ner och stannar när gesten avslutas. Båda dessa ANN ska få som input ett fönster bestående av 10 tidssteg där handens hastighet och acceleration ges som skalärer för var tidssteg (20 input värden).

3.1 Tidsaspekter

När ett fönster bestående av tidssteg ska ges som input till ett ANN och detta ANN ska lyckas hitta ett mönster i denna datan så krävs det att tidsstegen alltid är lika långa. Detta betyder att tidsstegen måste låsas till ett givet intervall vilket måste hållas. Skulle detta intervall inte gå att hålla så kommer inte mönstret som nätverket

Problembeskrivning

ska känna igen att se ut som det som nätverket tränats för och då är det omöjligt för nätverken att ge rätt output vid de tidpunkter där det önskas.

Då det är stor risk för att felaktiga tidpunkter pekats ut är det viktigt att start- och slutpunktsnätverken körs parallellt. Om detta inte skulle göras så skulle en felaktig slutpunkt mitt i en gest förstöra segmenteringen. Att nätverken arbetar parallellt betyder att alla slutpunkter som hittas kan markera slutet på en riktig gest och datan mellan denna och föregående startpunkt kontrolleras.

Det är viktigt att tänka på att en användare skulle kunna sitta och röra handen hur länge som helst vilket till slut skulle resultera i att en så väldig mängd data sparats undan att minnet blir överbelastat, detta då all data måste behållas fram till en ny startpunkt hittats. En annan aspekt är också att en gest inte tar oändligt lång tid att utföra. Efter en för lång period utan ny startpunkt så är det alltså något som är fel. Detta betyder att en tidsgräns måste sättas för hur lång tid det maximalt får ta att utföra en gest och om inte gesten avslutats innan dess så ska systemet bortse från jämförelser fram till en ny startpunkt hittats.

3.2 Träning av nätverken

Det är viktigt att exemplen vilka nätverken tränas med är från ett antal personer och inte bara från en eller ett fåtal personer. Anledningen till detta är att ett ANN som tränas med exempel vilka är snarlika endast lär sig att ge rätt utdata för den indatan som använts vid träningen. Genom att ha exempel från flera olika personer, förutsatt att dessa inte rör sig identiskt, så blir det ett visst brus i exemplen vilket gör det artificiella neurala nätverken mer toleranta för små avvikelser i indatan utan att ge fel utdata.

Exempel för träning där indatan ska ge en nolla är även viktigt att ge, annars skulle alla exempel nätverket tränas med att ha en etta som output vilket skulle betyda att nätverket kan komma att alltid ge en etta som output oavsett input, vilket inte är ett önskat beteende. Även de tidpunkter vilka ska ge en nolla måste plockas ut för hand. Om allt som inte skulle ge etta togs ut skulle exemplen ha stor redundans och exempel för vad som ska ge nolla skulle vara klart överrepresenterade vilket kan få ett ANN som tränas att ändra sina vikter så att det som skulle ge ett istället ger värden närmare noll.

3.3 Utvärdering

Det som är intressant att utvärdera är antalet rätt utpekade start- och slutpunkter samt hur många av dessa som tillsammans ger ett segment motsvarande en av gesterna i datan. Det önskade utfallet är att antalet utpekade gester är högt då detta arbete går ut på att segmentera utförda gester. Antalet start- och slutpunkter som hittas är intressanta då det kan visa sig att t ex slutpunkter är svårare att hitta än startpunkter och att detta resulterar i att gester missas.

De ANN vilka kommer att söka efter start- och slutpunkter kommer båda att ha en ensam output neuron. Detta neuron läses av och jämförs mot ett tröskelvärde. Förändringar av detta värde kommer att ställa in hur tolerant avgörandet är av om det är en start-/slutpunkt eller inte. Om nätverken tränas med exempel på vilka den ska ge en etta och gränsvärdet sätts till 1,0 så måste användarens gestbegynnelse och avslut vara nästan identiskt med de personer vilka skapat exemplen. Om tröskelvärdet istället skulle vara mycket lägre, t ex 0,1, då skulle istället nästan allting räknas som

Problembeskrivning

start och slut vilket skulle ge mängder med segment men inga som motsvarar de utförda gesterna.

För att få ett så bra beteende som möjligt för start- och slutpunktsnätverken så är det viktigt att experimentera fram det tröskelvärde vilket ger bäst resultat. Det är även mycket möjligt att det ena nätverket behöver ha ett annat tröskelvärde än det andra. Det kan mycket väl hända att systemet lättare hittar gester om startpunktsnätverket har ett högre tröskelvärde för att förhindra att felaktiga tidpunkter ses som start och på så vis förstör en gest medan slutpunktsnätverket måste ha ett lägre tröskelvärde för att inte missa den riktiga slutpunkten för en gest.

När nätverken ska reagera på när handen accelererar och saktar ner så är det även viktigt att se vad detta betyder för de olika gester som kan skapas. En cirkel kan ritas i en enda rörelse, men vad händer om gesten är en fyrkant? Kommer nätverken att registrera hörnen som en slutpunkt följt av en startpunkt eller kommer hörnen att utföras så pass snabbt att nätverken inte hinner reagera på detta?

Nätverken tränas på utplockade tidsfönster från sparad data, det är viktigt att det inte är denna data som används vid utvärderingen. Det ända det skulle bevisa är att de artificiella neurala nätverken har lärt sig de exempel de tränats med. Därför måste en del av den sparade datan användas till träningen och en annan till utvärderingen.

4 Metod

Liksom Fels & Hinton (1993) används ANN för att lokalisera specifika tidpunkter under körning. Ett ANN kommer att användas för att lokalisera när en gest påbörjas och ett ANN kommer att användas för att lokalisera när en gest avslutas. Båda dessa ANN kommer att vara *FeedForward* nätverk. Som i Fels och Hinton's arbete kommer nätverken att ha 10 gömda neuron och 1 output neuron av typen sigmoid. Det som skiljer nätverken är att det endast finns 20 input neuron, dessa av typen identitet, då rotation av handen till skillnad från i Fels och Hinton's arbete inte mäts och därför inte behöver filtreras bort, vilket är vad Fels och Hinton gör med sina extra 30 input neuron.

4.1 Tidsaspekter

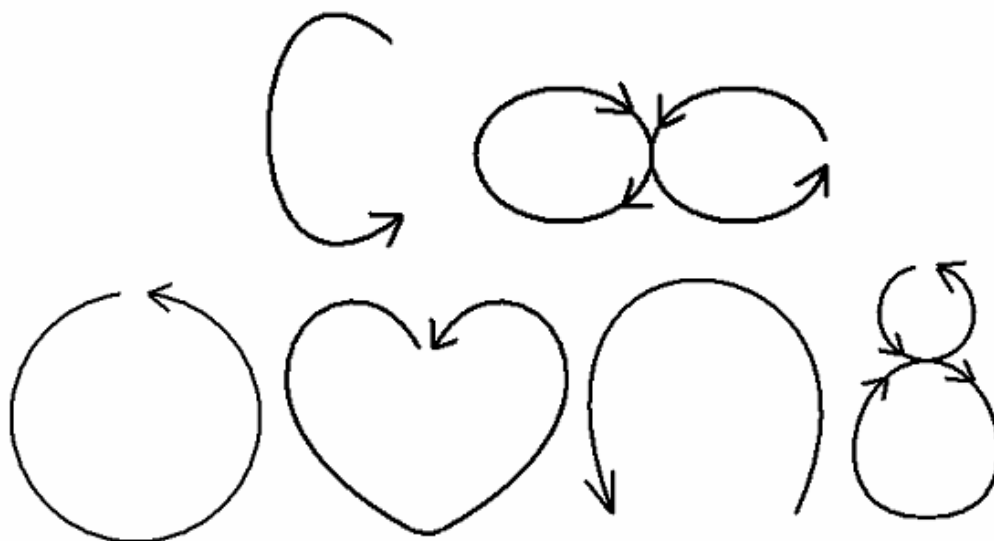
För att sätta en frekvens med vilken koordinaten för handen ska läsas av testas olika frekvenser för att vara säkra på att det går att köra med den givna frekvensen då datan sparas. Som utgångspunkt kommer 60 Hz att användas, den frekvens vilken Fels & Hinton (1993) använder sig av med lyckat resultat, och om denna frekvens inte skulle gå så testas en lägre fram till en fungerande frekvens hittas.

Olika tider för hur lång tid det maximalt får ta att utföra en gest kan ge olika resultat. En för kort tid skulle resultera i att längre gester missas, därför kommer tiden att sättas så att den överskrider den tid det längst tar för någon person att utföra en gest.

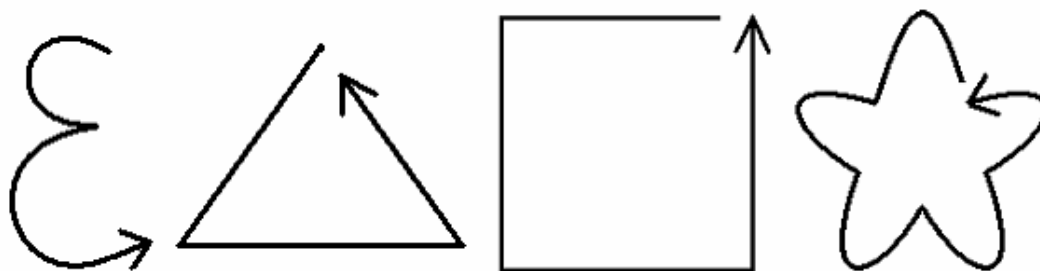
4.2 Träning av nätverken

För att få mer generella nätverk samlas data in från ett antal personer. Denna data består av 20 utförda gester per person varav 10 används till träning av nätverken och 10 för utvärdering. Anledningen till detta är att algoritmen ska utvärderas på data vilken inte använts för att träna de artificiella neurala nätverken.

Insamlingen av gestdatan görs med en applikation vilken visar en figur som ska ritas i luften (se Figur 9 och Figur 10). När figuren är utförd markeras detta genom att trycka på mellanslagstangenten vilket ändrar till en ny slumpvald figur. Under körningen sparas hastigheten och accelerationen handen rör sig i samt riktningen den rör sig i och allt stämplas med hur lång tid det har gått sen applikationen startades. Var gång en ny gest visas markeras detta i datafilerna med en textsträng längst till höger för att det lättare ska gå att hitta raden där en gest börjar i datan.



Figur 9. Figurer utan kanter vilka är med i samlingen av gester.



Figur 10. Figurer med kanter vilka är med i samlingen av gester.

4.3 Utvärdering

Den output vilken ges av nätverken ska tolkas som en etta eller en nolla. För att göra detta jämförs outputen mot ett tröskelvärde vilket bestämmer om det är en etta eller en nolla. Genom att ändra tröskelvärdena för om en start- eller slutpunkt hittats så kan beteendet för algoritmen förändras, därför måste olika kombinationer av tröskelvärden för start- och slutpunktsnätverken utvärderas. Detta görs genom att kombinationer av tre olika tröskelvärden körs där värdena är 0,5, 0,75 och 1,0. Dessa värden är utvalda för att få ett begränsat antal kombinationer och för att lägre tröskelvärden än 0,5 antas peka ut alldeles för många felaktiga tidpunkter.

När start- och slutpunkter letas via hastighets- och accelerationsförändringar så finns det en stor risk att kantiga gester är svårare att hitta än icke kantiga gester. Detta därför att vid var hörn i en kantig gest kan det bli så att en ny startpunkt hittas. Pga detta görs även en utvärdering där dessa två typer av gester är åtskilda för att undersöka om detta är ett problem.

5 Resultat

5.1 Tidsaspekter

Tester av vilken frekvens som fungerar stabilast vid avläsning och behandling av koordinatdatan från *Gametrack* kontrollen ledde till att frekvensen blev satt till 50 Hz, dvs att kontrollen läses av 50 gånger per sekund och värdena sparas till fil.

Innan körning för utvärdering sätts ett värde för hur många sekunder en gest maximalt får ta att göra, denna tid sätts till en tid under vilken alla personer vilka bidragit med gestdata klarat av att göra godtycklig gest, 30 sekunder.

5.2 Träning av nätverken

8 olika personer bidrog med data. Träningen av nätverken skedde sedan på så sätt att start- och slutpunkterna för var gest letades upp för hand i datan och för var start- och slutpunkt som matades in som exempel för vad som skulle ge en etta så matades även en tidpunkt vilken skulle ge en nolla in. Detta gav 160 exempel för var nätverk där 80 av exemplen visar vad som skulle ge en etta och 80 vad som skulle ge en nolla.

Start- och slutletarnätverken tränades båda 300 000 cykler med en *learning rate* på 0,1. Detta gav en genomsnittlig *error rate* på 0,000726 för startnätverket och 0,001012 för slutnätverket vid test på den data de tränats med. Anledningen till att pressa ner det genomsnittliga felvärdet var att få nätverken att verkligen kunna skilja på det som angivits som en start- respektive slutpunkt och vad som angivits som en händelse vilken inte är någon sådan.

5.3 Utvärdering

Utvärderingen sker på så vis att ett utvärderingsprogram vilket använder algoritmen för start- och slutpunktsletning går igenom en datafil innehållande acceleration och hastighet och ser om några start- och slutpunkter hittas. Om några sådana hittas kontrolleras tidpunkterna mot tidpunkter vilka letats upp för hand och som finns i en annan fil för att se om detta är en tänkt start- eller slutpunkt och det kontrolleras även om ett segment beskrivande en utförd gest hittats. I filen med rätt start- och slutpunkter finns även angivet om en viss gest innehåller kanter eller ej för att programmet efter körning ska kunna skapa en statistikfil innehållande information även för hur väl algoritmen fungerar för kantiga kontra icke kantiga gester.

5.4 Analys

	Totalt	
	procent	tröskelvärden (start/slut)
Bästa startpunktsletning	94.87%	0.5/alla
Sämsta startpunktsletning	79.49%	1.0/alla
Bästa slutpunktsletning	88.46%	alla/0.5
Sämsta slutpunktsletning	64.10%	alla/1.0
Mest felaktiga startpunkter	14.21%	0.5/alla
Minst felaktiga startpunkter	9.50%	1.0/alla
Mest felaktiga slutpunkter	12.96%	1.0/0.5
Minst felaktiga slutpunkter	6.60%	0.75/1.0 & 0.5/1.0
Bästa gestutpekningen	2.56%	1.0/alla
Sämsta gestutpekningen	0.00%	0.75/alla & 0.5/alla

Tabell 1. En sammanställning av de bästa och sämsta resultaten vid körning samt vilka tröskelvärde som användes då dessa resultat gavs. Informationen är hämtad från tabellerna i Bilaga 1.

Vid högre tröskelvärde för start minskar antalet felaktigt utpekade startpunkter. Detta betyder att risken för att en felaktig startpunkt dyker upp under utförandet av en gest minskar vilket är anledningen till att när tröskelvärdet 1,0 användes för start så hittades flest gester (se Tabell 1). I denna mätning blev det ingen skillnad oavsett tröskelvärdet för slut vad antalet hittade gester anbelangar, men flest slutpunkter hittades då tröskelvärdet var satt till 0,5. Den tidigare tanken att ett högt tröskelvärde för start och ett lågt för slut stämmer alltså, då ett större antal hittade slutpunkter ökar chansen att gester hittas.

Det högsta antalet hittade startpunkter ges då tröskelvärdet för start är 0,5. Detta är dock även det tröskelvärde vilket ger högst antal felaktigt utpekade startpunkter vilket ökar risken för att en felaktig startpunkt hittas under utförandet av en gest. Antalet hittade startpunkter och antalet felaktigt utpekade startpunkter är alltså beroende av varandra, och samma sak gäller för slutpunkter. Skillnaden är den att en ökad mängd felaktiga slutpunkter inte förstör segmenteringen, det enda som sker är att fler rörelsesegment måste kontrolleras.

Då resultatet visar att det som bäst endast går att hitta ett väldigt litet antal gester så går denna lösning inte att använda utan att den byggs ut på så sätt att den blir mer tolerant för felaktiga startpunkter. Den måste även bli mer tolerant mot felaktiga slutpunkter. Dessa förstör inget på så sätt att inga gester försvinner, istället kan det bli så att samma gest registreras flera gånger eller att en del av den tänkta gestalten registreras som någon annan gest (t ex en cirkel som blir till ett c).

5.4.1 Kantiga kontra icke kantiga gester

	gester med kanter	
	procent	tröskelvärden (start/slut)
Bästa startpunktsletning	96.67%	0.75/alla & 0.5/alla
Sämsta startpunktsletning	83.33%	1.0/alla
Bästa slutpunktsletning	93.33%	alla/0.5
Sämsta slutpunktsletning	76.67%	alla/1.0
Mest felaktiga startpunkter	15.14%	0.5/alla
Minst felaktiga startpunkter	10.36%	1.0/alla
Mest felaktiga slutpunkter	14.00%	1.0/0.5
Minst felaktiga slutpunkter	7.15%	0.75/1.0 & 0.5/1.0
Bästa gestutpekningen	6.67%	1.0/alla
Sämsta gestutpekningen	0.00%	0.75/alla & 0.5/alla

Tabell 2. En sammanställning av de bästa och sämsta resultaten för de gester vilka innehåller kanter. Informationen är hämtad från tabellerna i Bilaga1.

	gester utan kanter	
	procent	tröskelvärden (start/slut)
Bästa startpunktsletning	93.75%	0.5/alla
Sämsta startpunktsletning	77.08%	1.0/alla
Bästa slutpunktsletning	85.42%	alla/0.5
Sämsta slutpunktsletning	56.25%	alla/1.0
Mest felaktiga startpunkter	13.54%	0.5/alla
Minst felaktiga startpunkter	8.87%	1.0/alla
Mest felaktiga slutpunkter	12.20%	1.0/0.5
Minst felaktiga slutpunkter	6.21%	0.75/1.0 & 0.5/1.0
Bästa gestutpekningen	0.00%	alla/alla
Sämsta gestutpekningen	0.00%	alla/alla

Tabell 3. En sammanställning av de bästa och sämsta resultaten för de gester vilka inte innehåller kanter. Informationen är hämtad från tabellerna i Bilaga1.

Teorin om att det skulle hittas fler felaktiga start- och slutpunkter i kantiga gester än i icke kantiga stämde, som bäst pekades 10,36 % felaktiga startpunkter och 7,15 % felaktiga slutpunkter ut i kantiga gester medan resultatet för icke kantiga var 8,87 % och 6,21 %. Förvånande nog var siffrorna för hittade start- och slutpunkter högre för kantiga än för icke kantiga, 96,67 % av startpunkterna och 83,33 % av slutpunkterna för kantiga medan icke kantiga hittade 93,75 % och 77,08 %. Alla funna gester var kantiga gester, vilket även det var en överraskning. Antalet funna gester var dock så pass lågt att ingen riktig slutsats kan dras angående problem vid just kantiga gester, antalet felaktigt utpekade start- och slutpunkter talar dock för att det kan vara ett problem.

5.4.2 Sammanfattning av analysen

Den bästa kombinationen av tröskelvärden var 1,0 för start och 0,5 för slut. Anledningen till detta är att ett högt tröskelvärde för start minskar risken för att felaktiga startpunkter förstör segmenteringen medan ett lågt tröskelvärde för slut minskar risken att slutpunkter missas.

Resultat

I frågan om att det skulle vara svårare att hitta kantiga gester än icke kantiga går det inte att dra några slutsatser. Anledningen till detta är för det första att ett väldigt lågt antal gester hittades och för det andra att ingen analys gjorts angående hur personerna vilka bidragit med data utfört gesterna och att det finns en stor risk att många gester utförts slarvigt på så sätt att hörn hoppats över för att göra rörelserna snabbare, vilket skulle göra alla antaganden i frågan ogiltiga. Anledningen till att gester kan ha utförts slarvigt är att applikationen vilken använts till att samla in rörelsedata inte visar upp för användaren hur denna utför gester, den visar endast upp en bild beskrivande gester som ska utföras.

6 Slutsats

Målet med detta arbete var att med två artificiella neurala nätverk lyckas peka ut när gester startar och när de slutar. Utvärderingen av algoritmen gick till på så sätt att statistik togs fram över hur många start- och slutpunkter som hittades samt hur många av dessa som pekade ut ett segment motsvarande en utförd gest.

Statistiken visade att trots att de flesta start- och slutpunkterna hittades så lokaliserades väldigt få gester. Anledningen till det låga antalet hittade gester var det höga antalet felaktigt utpekade startpunkter, vilka om de dyker upp under en gests utförande förstör segmenteringen av den gesten. Detta betyder att målet delvis uppfyllts, de flesta start- och slutpunkterna pekades ut vilket var målet men antalet felaktigt utpekade startpunkter förstörde segmenteringen vilket som följd förstörde syftet med att lyckas peka ut start- och slutpunkter vilket är att segmentera upp gesterna.

6.1 Diskussion

6.1.1 Tidsfönstret

Valet att gå efter Fels & Hinton's (1993) metod med ett tidsfönster på 10 tidssteg fungerade inte lika bra som förväntat. Antagligen beror detta på att förändringen i hastighet och acceleration det söktes efter i deras arbete var mer specifik. När handen rör sig fram och sedan tillbaka så blir det ett tvärt stopp i rörelsen, någonting som inte sker någon annanstans i rörelsen. Då detta är så pass specifikt så fungerar det med ett litet tidsfönster, vilket Fels och Hinton bevisade genom att få ett lyckat resultat. När det gäller gester vilka är att rita figurer i luften är inte förändringarna lika specifika då det under ritandet finns tillfällen då rörelsen går långsammare för att sedan snabbas på och vice versa. För att de artificiella nätverken ska kunna se skillnad på vad som är en start- eller slutpunkt och vad som bara är hastighets- och accelerationsförändringar under gestens gång krävs det ett större tidsfönster där skillnaden mellan dessa två fall blir mer markant.

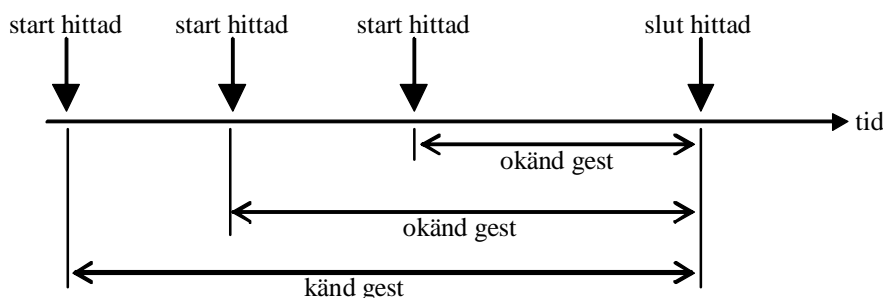
6.1.2 Hur fler gester skulle kunna hittas

Ett problem som uppstår när antalet felaktigt utpekade slutpunkter ökar är att om flera slutpunkter hittas runt den riktiga slutpunkten så kommer samma gest att registreras flera gånger (se Figur 11). Detta går dock att komma runt om den senast hittade gesten sparas undan och inte registreras förrän en ny startpunkt hittats. Denna lösning duger dock endast om algoritmen används i ett sammanhang där gester kommer i en kontinuerlig ström utan längre uppehåll mellan varandra. Ett bättre alternativ skulle då vara att plocka ut den första gest som systemet känner igen och sedan rensa startpunkten så att inga nya jämförelser görs förrän en ny startpunkt har hittats.

.		
.		
.		
STOP FOUND	4.7752	
STOP FOUND	4.7969	
STOP FOUND	4.8188	X
STOP FOUND	4.8403	
STOP FOUND	4.8618	
.		
.		
.		

Figur 11. Ett klipp från en av statistikfilerna som visar flera hittade slutpunkter runt den riktiga, den riktiga slutpunkten är markerad med ett kryss och värdet som är angivet är antalet sekunder sen applikationen startade.

För att lösa problemet med att felaktiga startpunkter under en gest förstör segmenteringen så kan ett antal startpunkter sparas så att t ex de 5 senaste finns tillgängliga. När ett slut hittas kontrolleras då de 5 segment som blir mellan de olika startpunkterna och den funna slutpunkten. Detta skulle skapa en tolerans för felaktiga startpunkter där antalet sparade startpunkter bestämmer toleransen (se Figur 12). Denna lösning har dock samma problem som skapas av felaktiga slutpunkter. Om startpunkterna ligger nära varandra så registreras gesten flera gånger då slutpunkten hittas.



Figur 12. En illustration av hur algoritmen blir mer tolerant mot felaktiga startpunkter. Den första starten är den menade, de andra två är felaktiga. Gesten hittas trots felaktiga startpunkter tack vare att ett antal startpunkter sparas undan och alla rörelsesegment kontrolleras vid en slutpunkt.

Ett sätt att göra algoritmen duglig skulle vara om startpunkter sparades undan som nämnts ovan. När kontrollen av rörelsesegmenten görs så börjas det alltid med den startpunkt som är längst tillbaka i tiden då idén är att göra segmenteringen tolerant mot felaktiga startpunkter under utförandet av en gest. Så fort en gest hittas så registreras denna och de sparade startpunkterna rensas. På så sätt registreras endast en gest och felaktiga startpunkter från föregående gest ligger inte kvar och förstör nästa gest som utförs. Denna lösning lider dock fortfarande av problemet att felaktiga slutpunkter kan få ett segment av den menade gesten att tolkas som något annat.

6.1.3 Handkontrollen till Nintendo Wii som input apparatur

En annan input apparatur vilken skulle kunna ha användning av den algoritmen vilken tagits fram i detta arbete om den skulle bli stabil är handkontrollen till *Nintendo Wii*. Denna kontroll ger accelerationen den rör sig i. Utifrån denna acceleration går det att utvinna vilken hastighet den rör sig i om det utgås ifrån att kontrollen vid start är

stillastående. Detta ger allt som behövs av algoritmen i detta arbete för att lokalisera när en gest utförs. För att sedan klassificera gesterna, vilka då kommer att vara av samma typ som de som tagits upp i detta arbete, behöver det endast bestämmas en utgångspunkt för rörelsen, t ex origo, och utifrån denna ta fram var positionerna hamnar m h a hastigheten. Detta att använda samma utgångspunkt för alla gester fungerar om datan från utförda gester bearbetas på ett likartat sätt med det som görs i Rabin (2006, kap 6.4), ett sätt vilket tar bort betydelsen av var i koordinatsystemet gesten utförs.

Att använda handkontrollen till *Nintendo Wii* för input skulle kunna ge ett bättre resultat än vad som gavs när *Gametrack* kontrollen användes. Anledningen till detta är att snörena vilka dras ut läses av som heltal, vilket ger artefakten att om snöret är utdraget så att värdet som ska ges är ett gränsvärde så växlar det värde som ges mellan de två angränsande värdena, vilket i sin tur ger accelerations- och hastighetsförändringar vilka inte finns. Handkontrollen till *Nintendo Wii* ger acceleration istället för position, vilket skulle kunna få bort denna artefakt då accelerationen inte längre räknas fram. Om utvärderingen skulle göras om på data samlad med *Nintendo Wii* kontrollen så skulle resultatet kunna bli bättre.

6.2 Framtida arbete

- I detta arbete användes ett tidsfönster på 10 tidssteg i data sparad 50 gånger per sekund. Argumentet till detta val var att ett tidigare arbete haft denna storlek på tidsfönstret och en ännu högre frekvens och fått ett lyckat resultat. Det är mycket möjligt att ett större tidsfönster skulle ge en bättre igenkänning då det ger de artificiella neurala nätverken mer att arbeta med. Det skulle därför vara intressant med en undersökning av hur resultatet förändras beroende på tidsfönstrets storlek.
- Utvärderingen av algoritmen visar att de flesta riktiga start- och slutpunkterna hittas (som sämst 79.49% av startpunkterna och 64.10% av slutpunkterna). Anledningen till att så få gester hittas (som bäst 2.56%) är att felaktiga startpunkter dyker upp under gesternas utförande. En metod för att utvinna kända gester av de segment som pekats ut av de olika start- och slutpunkterna, d v s använder alla de segment som hittas, skulle drastiskt öka antalet hittade gester (som bäst hittades 94.87% av startpunkterna och 88.46% av slutpunkterna).
- Det finns en möjlighet att rörelsedata insamlad med handkontrollen till *Nintendo Wii* skulle vara stabilare än den insamlad med *Gametrack* kontrollen. Med stabilare menas att *Gametrack* kontrollen har en tendens att skapa accelerations- och hastighetsförändringar vilka inte sker, vilket kanske inte *Nintendo Wii* kontrollen gör. Att samla in data med *Nintendo Wii* kontrollen, träna start- och slutletarnätverken med en del av denna data och utvärdera på en annan del av denna data skulle vara intressant för att se om anledningen till att resultatet blev dåligt i detta arbete berodde på den valda input apparaturen och inte på tidsfönstrets storlek.

Referenser

- Buckland, Mat (2002). *AI Techniques For Game Programming*. Cincinnati, Ohio: Premiere Press
- Fels, S. Sidney & Hinton, Geoffrey E. (1993). Glove-Talk: A neural network interface between a data-glove and a speech synthesizer. I IEEE Transaction on Neural Networks, vol. 4, nr. 1 (1993), s. 2-8.
- Harling, Philip A. (1993). [Elektronisk]. *Gesture Input using Neural Networks*. Report for BSc degree in Computer Science. Department of Computer Science at the University of York. Tillgänglig:
<http://citeseer.ist.psu.edu/cache/papers/cs/1424/ftp:zSzzSzftp.cs.york.ac.ukzSzpaper szSzgesture_NN.pdf/harling93gesture.pdf> [2008-02-26]
- Harling, Philip A. & Edwards, Alistair D.N. (1997). Hand tension as a gesture segmentation cue. I Harling, Philip A. & Edwards, Alistair D.N. (red.) *Progress in Gestural Interaction*, s. 75-88. London: Springer-Verlag.
- Rabin, S. (2006). *AI Game Programming Wisdom 3*. Boston, Massachusetts: Charles River Media.
- Viblis, M.K. & Kyriakopoulos, K.J. (1999). Gesture Recognition: The Gesture Segmentation Problem. I Journal of Intelligent and Robotic Systems, vol. 28 (2000), s. 151-158. Nederländerna: Kluwer Academic Publishers.
- Wickman, M. (2007). [Elektronisk]. *Undersökning av Wii Remote för utveckling av interaktiva rörelsekänsliga applikationer*. Examensarbete i Datateknik. Institutionen för Teknik vid Högskolan i Kalmar. Tillgänglig:
<<http://www.penvision.se/wic/rapport.pdf>> [2008-02-13]

Bilaga 1; Statestik för start- och stoppsökning i data från personerna som bidrog till träningsdatan

Tröskelvärde för start = 1.0, tröskelvärde för stopp = 0.5			
	Totalt	Kantiga	Icke kantiga
Antal gester	78	30	48
Hittade startpunkter	62	25	37
Hittade startpunkter %	79,4871795	83,3333333	77,0833333
Hittade slutpunkter	69	28	41
Hittade slutpunkter %	88,4615385	93,3333333	85,4166667
Hittade gester	2	2	0
Hittade gester %	2,56410256	6,6666667	0
	Totalt	Kantiga	Icke kantiga
Antal icke speciella tidpunkter	17034	7162	9872
Felaktigt utpekade startpunkter	1618	742	876
Felaktigt utpekade startpunkter %	9,49864976	10,3602346	8,87358185
Felaktigt utpekade slutpunkter	2207	1003	1204
Felaktigt utpekade slutpunkter %	12,9564401	14,004468	12,1961102

Tröskelvärde för start = 1.0, tröskelvärde för stopp = 0.75			
	Totalt	Kantiga	Icke kantiga
Antal gester	78	30	48
Hittade startpunkter	62	25	37
Hittade startpunkter %	79,4871795	83,3333333	77,0833333
Hittade slutpunkter	67	27	40
Hittade slutpunkter %	85,8974359	90	83,3333333
Hittade gester	2	2	0
Hittade gester %	2,56410256	6,6666667	0
	Totalt	Kantiga	Icke kantiga
Antal icke speciella tidpunkter	17034	7162	9872
Felaktigt utpekade startpunkter	1618	742	876
Felaktigt utpekade startpunkter %	9,49864976	10,3602346	8,87358185
Felaktigt utpekade slutpunkter	2135	977	1158
Felaktigt utpekade slutpunkter %	12,533756	13,6414409	11,7301459

Tröskelvärde för start = 1.0, tröskelvärde för stopp = 1.0			
	Totalt	Kantiga	Icke kantiga
Antal gester	78	30	48
Hittade startpunkter	62	25	37
Hittade startpunkter %	79,4871795	83,3333333	77,0833333
Hittade slutpunkter	50	23	27
Hittade slutpunkter %	64,1025641	76,6666667	56,25
Hittade gester	2	2	0
Hittade gester %	2,56410256	6,6666667	0
	Totalt	Kantiga	Icke kantiga
Antal icke speciella tidpunkter	17034	7162	9872
Felaktigt utpekade startpunkter	1618	742	876
Felaktigt utpekade startpunkter %	9,49864976	10,3602346	8,87358185
Felaktigt utpekade slutpunkter	1129	514	615
Felaktigt utpekade slutpunkter %	6,62792063	7,17676627	6,22974068

Tröskelvärde för start = 0.75, tröskelvärde för stopp = 0.5			
	Totalt	Kantiga	Icke kantiga
Antal gester	78	30	48
Hittade startpunkter	73	29	44
Hittade startpunkter %	93,5897436	96,6666667	91,6666667
Hittade slutpunkter	69	28	41
Hittade slutpunkter %	88,4615385	93,3333333	85,4166667
Hittade gester	0	0	0
Hittade gester %	0	0	0
	Totalt	Kantiga	Icke kantiga
Antal icke speciella tidpunkter	17034	7162	9872
Felaktigt utpekade startpunkter	2371	1070	1301
Felaktigt utpekade startpunkter %	13,9192204	14,9399609	13,1786872
Felaktigt utpekade slutpunkter	2194	996	1198
Felaktigt utpekade slutpunkter %	12,8801221	13,90673	12,1353323

Tröskelvärde för start = 0.75, tröskelvärde för stopp = 0.75			
	Totalt	Kantiga	Icke kantiga
Antal gester	78	30	48
Hittade startpunkter	73	29	44
Hittade startpunkter %	93,5897436	96,6666667	91,6666667
Hittade slutpunkter	67	27	40
Hittade slutpunkter %	85,8974359	90	83,3333333
Hittade gester	0	0	0
Hittade gester %	0	0	0
	Totalt	Kantiga	Icke kantiga
Antal icke speciella tidpunkter	17034	7162	9872
Felaktigt utpekade startpunkter	2371	1070	1301
Felaktigt utpekade startpunkter %	13,9192204	14,9399609	13,1786872
Felaktigt utpekade slutpunkter	2123	971	1152
Felaktigt utpekade slutpunkter %	12,4633087	13,5576655	11,6693679

Tröskelvärde för start = 0.75, tröskelvärde för stopp = 1.0			
	Totalt	Kantiga	Icke kantiga
Antal gester	78	30	48
Hittade startpunkter	73	29	44
Hittade startpunkter %	93,5897436	96,6666667	91,6666667
Hittade slutpunkter	50	23	27
Hittade slutpunkter %	64,1025641	76,6666667	56,25
Hittade gester	0	0	0
Hittade gester %	0	0	0
	Totalt	Kantiga	Icke kantiga
Antal icke speciella tidpunkter	17034	7162	9872
Felaktigt utpekade startpunkter	2371	1070	1301
Felaktigt utpekade startpunkter %	13,9192204	14,9399609	13,1786872
Felaktigt utpekade slutpunkter	1125	512	613
Felaktigt utpekade slutpunkter %	6,60443818	7,14884111	6,20948136

Tröskelvärde för start = 0.5, tröskelvärde för stopp = 0.5

	Totalt	Kantiga	Icke kantiga
Antal gester	78	30	48
Hittade startpunkter	74	29	45
Hittade startpunkter %	94,8717949	96,6666667	93,75
Hittade slutpunkter	69	28	41
Hittade slutpunkter %	88,4615385	93,3333333	85,4166667
Hittade gester	0	0	0
Hittade gester %	0	0	0
	Totalt	Kantiga	Icke kantiga
Antal icke speciella tidpunkter	17034	7162	9872
Felaktigt utpekade startpunkter	2421	1084	1337
Felaktigt utpekade startpunkter %	14,212751	15,135437	13,5433549
Felaktigt utpekade slutpunkter	2193	996	1197
Felaktigt utpekade slutpunkter %	12,8742515	13,90673	12,1252026

Tröskelvärde för start = 0.5, tröskelvärde för stopp = 0.75

	Totalt	Kantiga	Icke kantiga
Antal gester	78	30	48
Hittade startpunkter	74	29	45
Hittade startpunkter %	94,8717949	96,6666667	93,75
Hittade slutpunkter	67	27	40
Hittade slutpunkter %	85,8974359	90	83,3333333
Hittade gester	0	0	0
Hittade gester %	0	0	0
	Totalt	Kantiga	Icke kantiga
Antal icke speciella tidpunkter	17034	7162	9872
Felaktigt utpekade startpunkter	2421	1084	1337
Felaktigt utpekade startpunkter %	14,212751	15,135437	13,5433549
Felaktigt utpekade slutpunkter	2122	971	1151
Felaktigt utpekade slutpunkter %	12,4574381	13,5576655	11,6592382

Tröskelvärde för start = 0.5, tröskelvärde för stopp = 1.0

	Totalt	Kantiga	Icke kantiga
Antal gester	78	30	48
Hittade startpunkter	74	29	45
Hittade startpunkter %	94,8717949	96,6666667	93,75
Hittade slutpunkter	50	23	27
Hittade slutpunkter %	64,1025641	76,6666667	56,25
Hittade gester	0	0	0
Hittade gester %	0	0	0
	Totalt	Kantiga	Icke kantiga
Antal icke speciella tidpunkter	17034	7162	9872
Felaktigt utpekade startpunkter	2421	1084	1337
Felaktigt utpekade startpunkter %	14,212751	15,135437	13,5433549
Felaktigt utpekade slutpunkter	1125	512	613
Felaktigt utpekade slutpunkter %	6,60443818	7,14884111	6,20948136