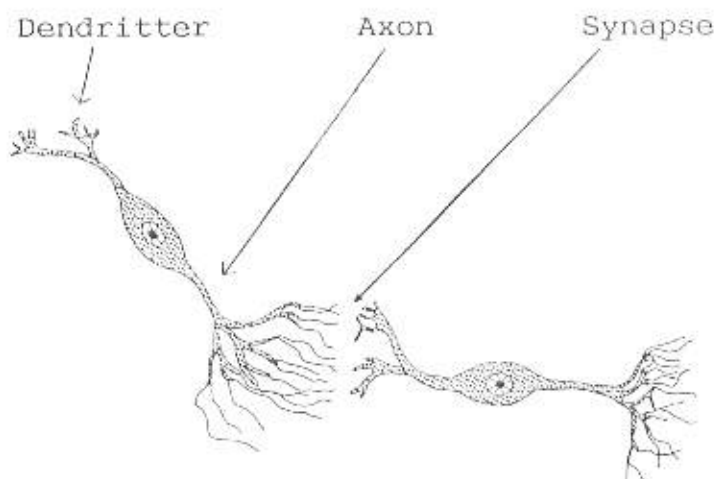


## **KAPITEL 2**

## 2. INTRODUKTION TIL NEURALE NETVÆRK

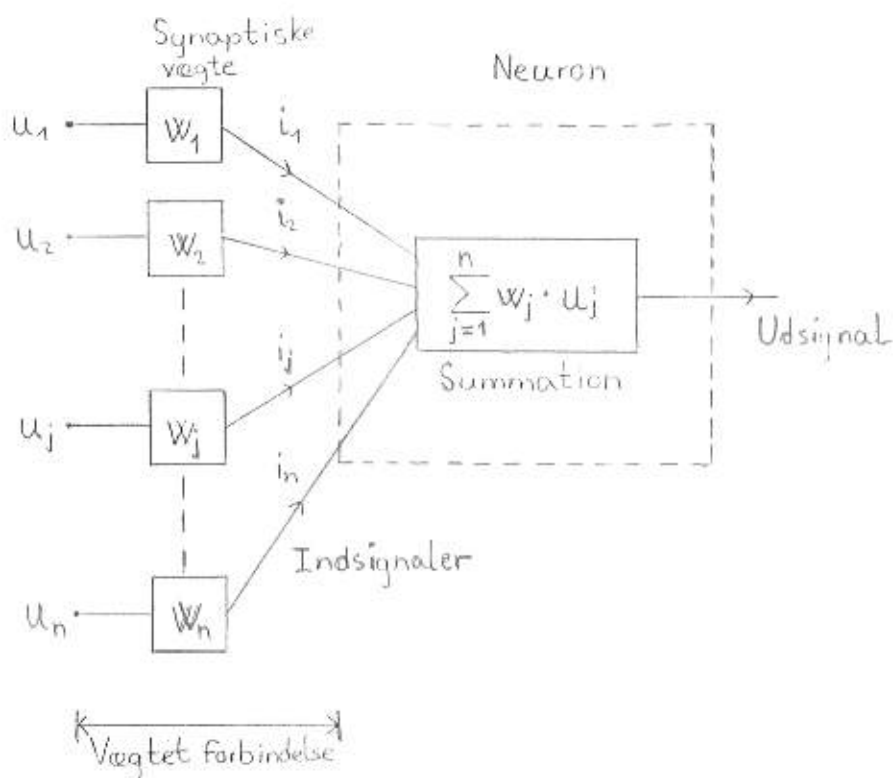
### 2.1 Indledning

Med neuron betegnes i den menneskelige hjerne en nervecelle, som har forbindelse til andre neuroner via sin "axon" og sine "dendritter". Axonen er neuronens sendekanal, hvorigennem det øvrige netværk informeres om neuronens øjeblikkelige tilstand. Dendritterne er modtagekanaler, gennem hvilke neuronens selv modtager information om tilstanden hos andre neuroner, den har forbindelse med. Som resultat af disse andre neuroners "pres", opdaterer neuronens løbende sit eget aktivitetsniveau. Se figur 2.1.



Figur 2.1 En biologisk neuron med dendritter, synapse og axon

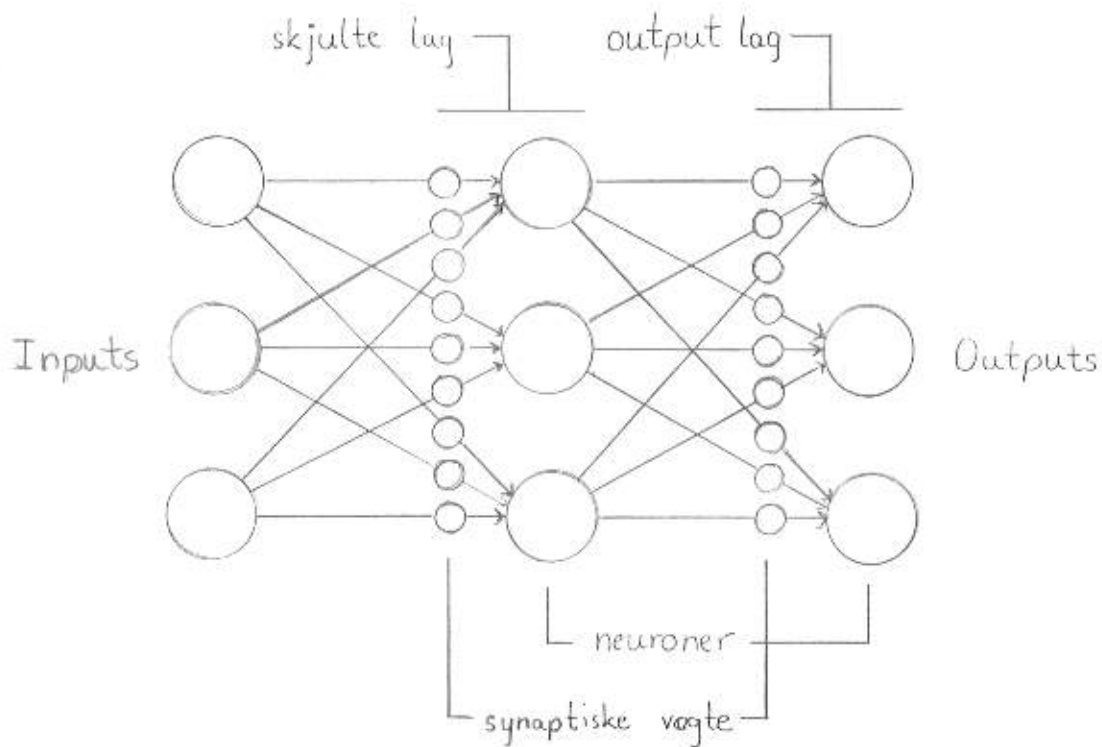
I kunstige neurale netværk simuleres synapserne mellem neuroner med et vist vægttal  $w_n$  - "en synaptisk vægt" - tillagt hver neuronforbindelse, hvormed udsignalet fra en afsender-neuron multipliceres, inden det bliver til indsignal for den modtagende neuron. I modtager-neuronen summeres de vægtede signaler. Forbindelsen mellem to forbundne neuroner benævnes "vægtet forbindelse". Se figur 2.2.



Figur 2.2 En simpel kunstig neuron med vægtede forbindelser

Efter behov kan en kunstig neuron tillægges en tærskelværdi, så den først "tænder", når summen af de vægtede indsignaler overstiger denne værdi, og den kan udstyres med en overføringsfunktion til transformation af den restværdi, hvormed summen af vægtede indsignaler overstiger tærskelværdien jfr. figur 2.7.

Kunstige neurale netværk, som herefter kaldes neurale netværk eller blot netværk, er defineret som parallelle lag af neuroner indbyrdes forbundet gennem vægtede forbindelser, se figur 2.3. Forbindelserne mellem neuronerne kan etableres på forskellig måde afhængigt af hvilken netværksmodel, der ønskes konstrueret.



Figur 2.3 Neuralt netværk

Med given neuron-bestykning af et neuralt netværk til løsning af en bestemt opgave står tilbage at fastlægge de synaptiske vægte, som bedst muligt afbilder input-data i tilsigtede output-data. Fastlæggelse af vægtene sker gennem såkaldt "træning" af netværket.

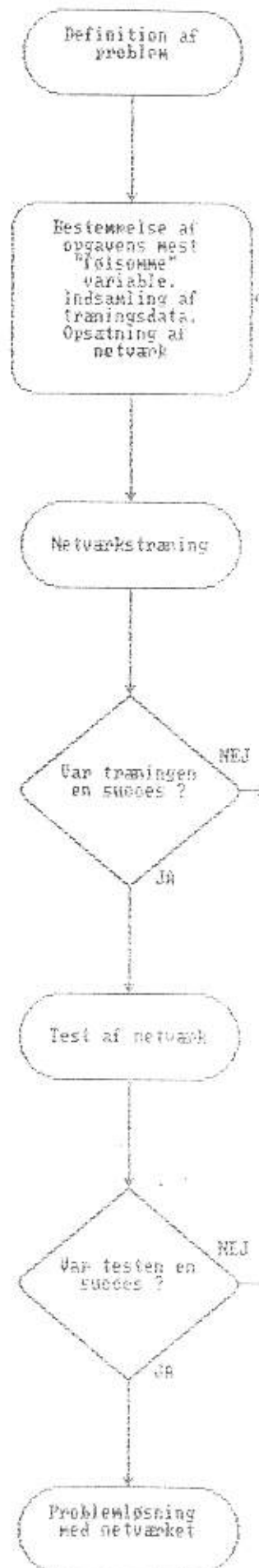
Der findes to principelt forskellige måder at træne et netværk på: "Overvåget" (supervised) træning og "uovervåget" (un-supervised) træning.

Ved overvåget træning indstilles netværkets synaptiske vægte først til "small random values" og input-laget påtrykkes et sæt input-data. De hertil svarende output-data fra netværket sammenholdes med de tilsigtede data, og forskellen bestemmes. Derefter justeres vægtene med henblik på at mindske afvigelsen mellem det

tilsigtede og det fundne output. Med de således bestemte synaptiske vægte som udgangspunkt påtrykkes netværket et nyt sæt træningsdata, og vægtene justres igen som beskrevet. Samme procedure gentages med øvrige træningsdata, indtil vægtekombinationen er bedst mulig, hvad alle de anvendte træningsdata angår.

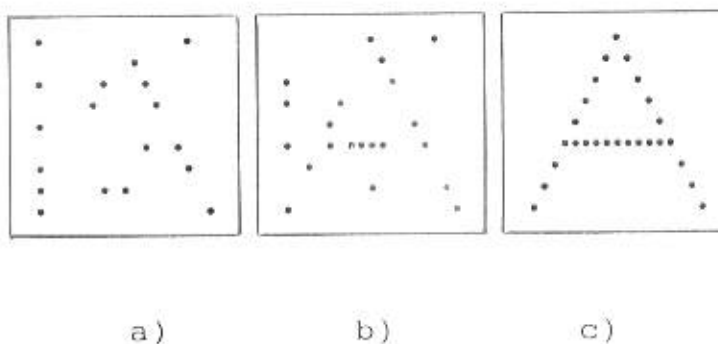
Ved uovervåget træning er der ikke samme mulighed for kontrol af netværkets indlæringsproces, som ved overvåget træning. I stedet for er lære-proceduren indrettet til at lede efter mønstre i de givne input data. Netværket forsøger - som ved overvåget træning - selv at korrigere afvigelser i fortolkningen af data ved at justere på vægtene, men der er ingen mulighed for at kontrollere, om det af netværket fundne resultat er korrekt.

Der findes i dag over 40 forskellige måder at udføre netværks-træning på, men mest udbredt er "Backpropagation of Error", som er en overvåget træningsmetode. Nærværende rapport omhandler kun denne metode, som anvendes i de tre netværksløsninger. I "Backpropagation of Error" føres fejlsignalet tilbage i netværket og anvendes til justering af de synaptiske vægte. Indlærings-proceduren gentages, til forskellen mellem alle træningsmønstrenes output-data og netværkets output-data når under en forudbestemt tolerance-værdi.



Figur 2.4 Træningsproces for et neuralt netværk

Som den menneskelige hjerne vil et optrænet netværk kunne danne en art "associationer" ud fra de erfaringer, det har gjort under træningen. Har et veltrænet netværk for eksempel lært sig bogstavet A (figur 2.5c) at kende under træning, vil det være i stand til at "associere" til dette bogstav, selv om det kun præsenteres for et brudstykke heraf (figur 2.5a).



Figur 2.5 Dynamisk udvikling for et neuralt netværk

En sådan stump af erindring om bogstavet A vil skabe et ustabil aktivitetmønster i netværket, som gradvist ændres af det neurale nets dynamik. Nettets neuroner udkæmper en "alle - mod - alle - kamp", der til sidst udmøntes i en stabil tilstand med mindst mulig frustration neuronerne imellem og en fuldstændig erindring om bogstavet A, figur 2.5a-c.

Træningsprocessen kan opfattes som en "oversættelse" af en ønsket erindring til synaptiske vægte. Omvendt vil en erindring kunne fremkaldes ved hjælp af samme synaptiske vægte.

Et neuralt netværk kan betragtes som et klitlandskab med "gryder", hvor gryderne afbilder netværkets erindringer. Hver gryde har et vist indfangningsområde, og ønskes en bestemt erindring genkaldt, må associeringsprocessen startes - "bolden kastes" - hen i den rigtige grydes indfangningsområde.

I et neuralt net kan en bestemt erindring ikke lokaliseres, idet den tilhørende informationen er fordelt over alle netværkets synaptiske vægte. Det er dette forhold, der gør det muligt, selv efter temmeligt store ødelæggelser i et neuralt netværk, at genkalde dets erindringer, uden at disse bliver væsentligt forringet.

Neurale net har adaptive egenskaber, hvilket muliggør, at netværket til stadighed kan træne, lære og minimere fejl. Da netværk tillige kan opbygges af neuroner med ulineære overføringsfunktioner, er det muligt også at behandle ulineære data.

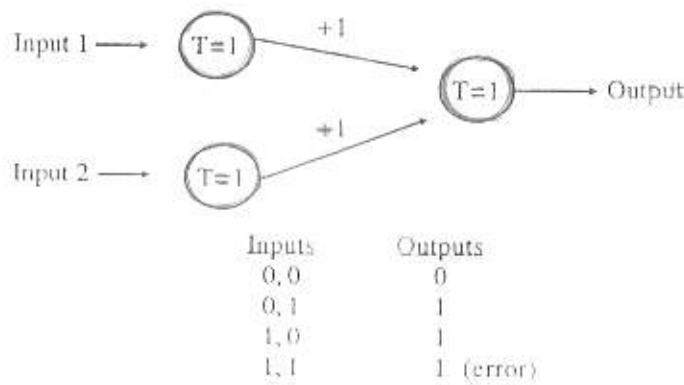
Det simplest tænkelige neurale netværk, er et to-lags netværk blot bestående af et input-lag og et output-lag. Sådanne netværk kaldes "enkeltlags-perceptroner". Det var med et sådant netværk, de to amerikanske forskere: Matematikeren Walter Pitts og neurofysiologen Warren McCullough i 1943 troede at de havde påvist muligheden for at kunne løse enhver opgave indenfor logiske funktioner med enkeltlags-perceptroner.

I 1960'erne tillagdes disse enkeltlags-perceptroner store fremtidsudsigter, men det viste sig snart, at deres muligheder var for begrænsede.

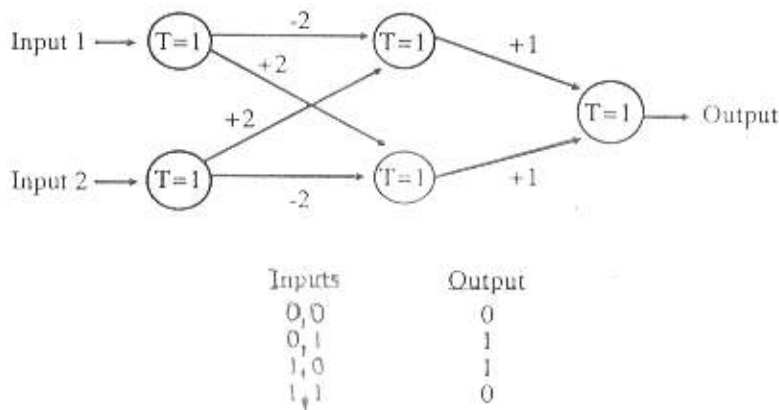
Begrænsningen kan for eksempel illustreres med figur 2.6a, hvor XOR-problemet søges løst ved en enkeltlags-perceptron med to neuroner i input-laget og én neuron i output-laget.  $T$  er neuronernes tærskelværdi, hvilket betyder at en neuron aktiveres, hvis summen af inputs er større eller lig med  $1$ . De anvendte synaptiske vægte er begge  $+1$ . Output  $0$  med Inputs  $1,1$  lader sig ikke realisere med dette meget simple netværk.

Ved at indskyde et skjult lag i netværket og anvende  $+1$  og  $+2$  som vægte bliver det imidlertid muligt at danne den logiske XOR-funktion, se figur 2.6b.





a)



b)

Figur 2.6 XOR-problemet

I 1969 fremførte to af den tids førende AI-folk Marvin Minsky og Seymour Papert i deres bog "Perceptrons", at neurale nets muligheder var stærkt begrænsede, og interessen herfor kølnedes. En anden medvirkende faktor var, at den serielle computer på samme tid havde fået "vind i sejlene".

Denne "dødsdom" over de neurale net i 1969 blev først ophævet i midten af 1980'erne, da Hinton udviklede den overvågede træning. Et nyt stort problem var imidlertid, at der ikke fandtes effektive træningsalgoritmer til mere komplicerede netværk, hvilket først fandt en tilfredsstillende løsning, da Rumelhart i 1986

udviklede "Backpropagation of Error" for træning af feed-forward netværk med skjulte neuroner, (se afsnit 2.3). Efter dette gennembrud har forskning og udvikling taget fart. Næste mål er udvikling af parallelle computere, og der arbejdes p.t. intensivt på en teknik, kaldet Analog VLSI, hvori beregninger udføres parallelt, med høj hastighed til følge.

Neurale netværk har løst opgaver på områder, som ikke tidligere har kunnet løses med brug af traditionelle computerprogrammer, for eksempel: Optical Character Recognition (OCR), tekst-til-tale omsætning, identifikation af cancerceller, fejlanalyse af maskiner, proceskontrol, forudsigelse af finansielle udviklinger etc.

## 2.2 Den kunstige neuron

Et neuralt netværk er opbygget af kunstige neuroner, hvor hver enkel neuron modtager flere indsignaler men afgiver kun ét udsignal, se figur 2.2.

Enhver neuron i de skjulte lag og output-laget modtager vægtede udsignaler fra alle tilkoblede neuroner på modtagersiden efter udtrykket:

$$(2.1) \quad x_i = \sum_{j=1}^n w_{ij} \cdot u_j - \theta_i$$

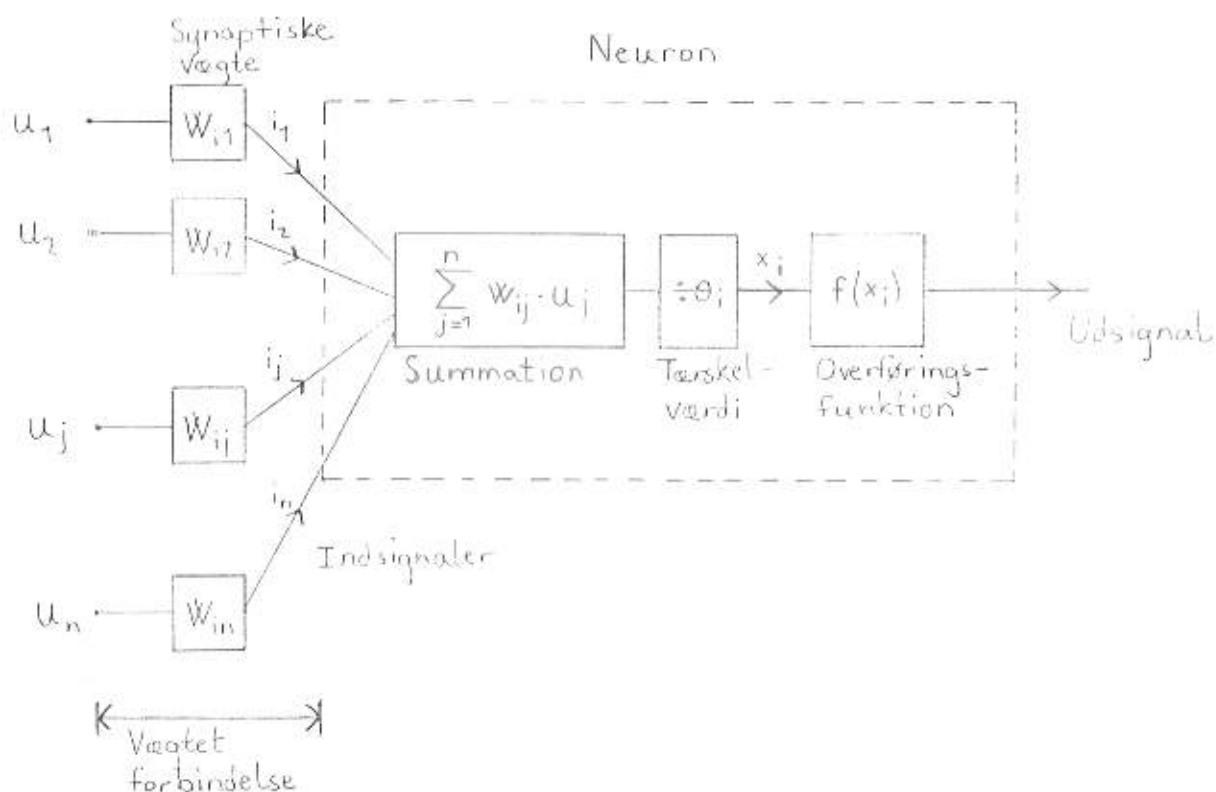
hvor  $i$  er neuronens nummer i netværket,  $n$  er antal neuroner, den har kontakt til på modtagersiden,  $j$  er summeringsvariablen,  $u_j$  er udsignal fra neuron nr.  $j$ ,  $w_{ij}$  er synaptisk vægt i forbindelsen mellem neuron nr.  $i$  og  $j$ , og  $\theta_i$  er tærskelværdien for neuron nr.  $i$ , se figur 2.7.

Neuronens eget udsignal er:

$$(2.2) \quad y_i = f(x_i)$$

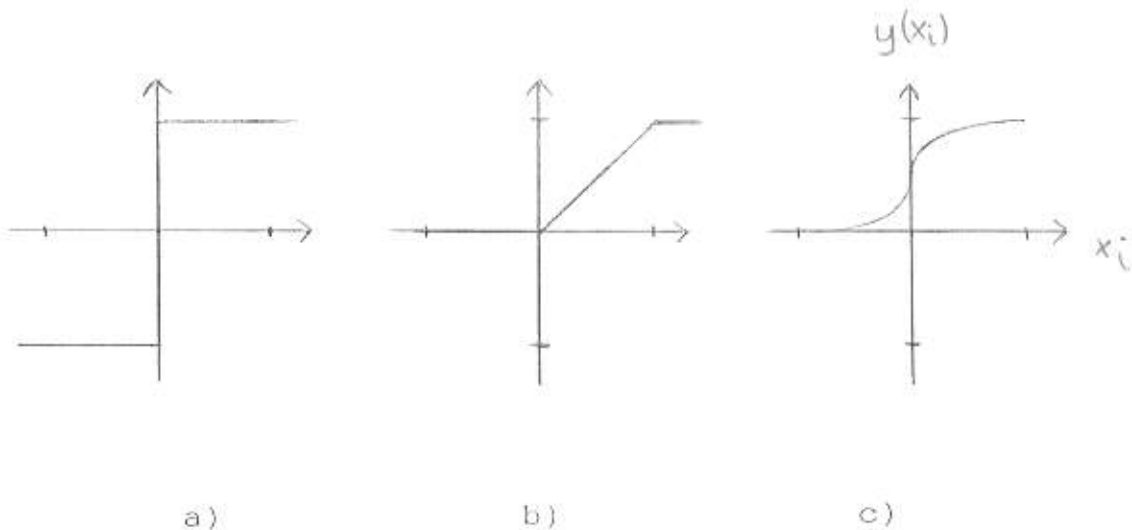
hvor funktionen  $f$  er neuronens overføringsfunktion.

Figur 2.7 illustrerer den kunstige neurons beregningsfunktion ved en matematisk model baseret på ligningerne 2.1 og 2.2.



Figur 2.7 En kunstig neurons beregningsfunktion

Kunstige neuroner opdeles efter deres overføringsfunktion, hvor de mest anvendte er vist nedenfor:



Figur 2.8 Overføringsfunktion for kunstige neuroner.  
a) Step b) Lineær c) Sigmoid

Programmet BrainMaker kan arbejde med disse tre overføringsfunktioner, men det tilrådes fra producenten at bruge Sigmoid (hyperbolsk tangens) overføringsfunktionen, fordi problemløsning ved hjælp af neurale netværk oftest fordrer neuroner med ikke-lineære og kontinuerligt differentiable overføringsfunktioner.

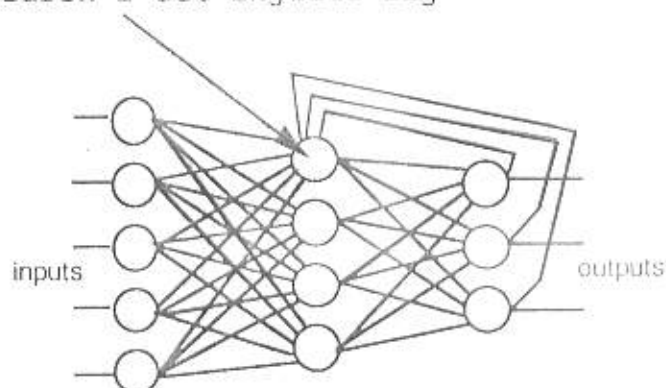
### 2.3 Netværket

Et neuralt netværk er en struktur af neuroner arrangeret i lag, med et input-lag, ét eller flere skjulte lag og et output-lag. Generelt kan neurale net opbygges på to forskellige måder:

- 1) Feedback-modellen
- 2) Feedforward-modellen

Feedback-modellen er karakteriseret ved, at udsignalet fra neuroner føres tilbage til neuroner i det samme eller forrige lag som indsignal (se figur 2.9).

Feedback fra outputlaget til  
en neuron i det skjulte lag



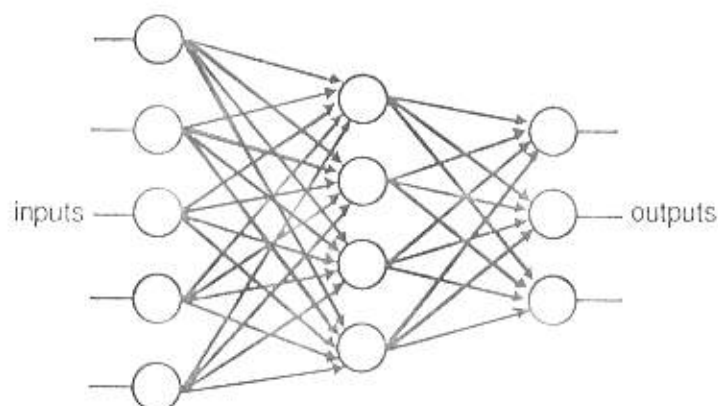
Figur 2.9 Feedback-modellen

Feedforward-modellen er karakteriseret ved, at signalretningen i netværket bestandigt er fra input-laget mod output-laget, altså ingen tilbagekobling mellem lagene. Feedforward-netværk er de oftest anvendte neurale netværk, blandt andet fordi de er lettere at håndtere i træningsfasen end feedback-netværk. De neurale netværk som anvendes i dette projekt er alle enkle feedforward netværk (se figur 2.10).

Et feedforward netværk approximerer en funktion, som afbilleder et antal variable (inputs) i nogen andre størrelser (outputs). Den approximerede bestemmes af netværkets opbygning, de anvendte neuron-typer og de synaptiske vægte. Med en valgt netværksopbygning og neuron-bestykning tilbage står at fastlægge de synaptiske vægte, så den ønskede funktion approximeres bedst mulig.

De synaptiske vægte fastlægges ved en iterativ træningsproces, hvori vægtene indledningsvis tildeles små, tilfældige værdier. Derefter påtrykkes netværkets input-lag en serie af træningsdatasæt, sæt for sæt, og de tilsvarende output-datasæt fra netværket bestemmes. Hvert af disse datasæt sammenlignes med de tilsligtede output-datasæt, og den resulterende afvigelse mellem realiserede og tilsligtede output anvendes til justering af de synaptiske vægte, så afvigelsen bliver mindre ved næste runde med samme trænings-datasæt.

En hyppigt anvendt metode til træning af feedforward-netværk er den såkaldte "Backpropagation of Error", en overvåget træningsmetode omtalt i afsnit 2.5.



Figur 2.10 Feedforward-modellen

## 2.4 Design af netværket

Når et neuralt netværk skal opstilles er der en del praktiske aspekter at tage stilling til:

- 1) Definition af det problem, der skal løses.
- 2) Valg af software- og/eller hardware-løsning.
- 3) Bearbejdning af træningsdata.
- 4) Opsætning af netværket.

I det følgende vil disse fire punkter blive belyst.

1) Definition af det problem, der skal løses.

Her er det vigtigt at formulere, hvad det ønskede mål er og at vælge hvilke data, som skønnes mest relevante for netværkets funktion. Dette er en ganske tidskrævende proces, da man skal sikre sig, at netværket får den nødvendige og tilstrækkelige information, der på den ene side dækker alle muligheder og på den anden side ikke indeholder for megen information.

2) Valg af software- og/eller hardware-løsning

Neurale netværk kan simuleres på traditionelle computere, hvor hastighedsfordelene ved den parallelle databehandling dog ikke lader sig realisere. Til gengæld er det relativt nemt at lave ændringer i netværket. På det seneste er der kommet et antal hardware-produkter på markedet, hvormed det er blevet nemmere og hurtigere end tidligere at implementere netværk i hardware. Til mindre og simple applikationer vil software-værktøjerne ofte være fuldt tilstrækkelige, men hvis netværkerne bliver store og komplekse, vil en hardware-løsning være det bedste valg.

I pkt. 2.6 beskrives en række af de produkter, som er tilgængelige på dagens marked.

3) Bearbejdning af træningsdata

Et væsentligt problem ved opsætning og træning af et neuralt netværk er at få de indsamlede træningsdata præsenteret på en måde, så netværket får bedst mulig konvergens. Til dette formål må træningsdata ofte bearbejdes før præsentation for netværket. Bearbejdelsen kan for eksempel omfatte bortlugning af overflødig information eller skalering af inputdata.

#### 4) Opsætning af netværket

En eksperimentel proces, hvortil der ikke findes klare regler eller matematisk sammenhæng. Nogle håndregler af teoretisk og/eller praktisk oprindelse tilsiger:

Antallet af træningsdatasæt bør ikke overstige 10 gange antallet af forbindelser i netværket.

Antallet af træningsdatasæt bør mindst være produktet af antallet af skjulte neuroner og den inverse af træningstolerancen.

Tillæg "støj" på inputdata, hvis der er mulighed for at disse data indeholder værdien nul (0). Dette gøres da  $0 \cdot w$  er 0 uafhængigt af den synaptiske vægts ( $w$ ) størrelse, hvilket med støj fører til at vægtens værdi ikke negligeres. Støjen, som skulle give netværket en bedre generaliseringsevne, bør vælges så den udgør 5-10% af inputområdet.

Start træningen med et beskedent antal neuroner i de skjulte lag og læg nye neuroner til, hver gang træning indenfor et vist interval ikke længere medfører forbedring.

Brug middelværdien af antallet af input- og output-neuroner som antallet af skjulte neuroner.

Hvis et netværk indeholder for mange skjulte neuroner, vil det ende med at huske de enkelte træningsdata i stedet for at lære at generalisere. Dette vil være tilfældet, når netværket træner godt, men tester dårligt. Indeholder netværket for få skjulte neuroner, og lægges der ikke automatisk nye neuroner til under træningen, vil det give sig udslag i, at netværket aldrig bliver færdigt med at træne.



## 2.5 Træning af netværket

Et neuralt netværk tilpasses ikke en foreliggende opgave ved at få foreskrevet instruktioner eller regler herfor, men ved at få præsenteret en serie input-/output-datasæt vedrørende opgaven, hvormed netværkets synaptiske vægte justeres, så dets afbildning af input-data bedst muligt kommer til at stemme overens med de tilsigtede output-data. Denne form for behandling af data kaldes træning eller oplæring.

Et neuralt netværk træner på en serie træningsdatasæt ved at gennemløbe disse et utal af gange, og prøver herved at finde de indstillinger af de synaptiske vægte, som giver det bedste samlede resultat. Vægtene bliver ændret lidt hver gang et netværks-output falder udenfor den fastlagte træningstolerance i forhold til ønskede output (Se iøvrigt kapitel 4).

Træningsmetoden "Backpropagation of Error", som betyder "tilbageudbredelse af fejl", er den mest almindelige træningsmetode for neurale netværk af feedforward modellen. Reglen er baseret på, at der til stadighed modificeres på de synaptiske styrker af neuronforbindelserne for at reducere afvigelsen (delta) mellem den ønskede output-værdi og den nuværende output-værdi af den enkelte neuron i netværket.

"Backpropagation of Error" består af to faser:

Første fase kaldes "forward-fasen", hvor et input bliver påtrykt netværket og udbredt fremad gennem dette for at bestemme den dertil svarende output-værdi for den enkelte neuron. For hver neuron er alle nuværende outputs sammenlignet med det ønskede output og afvigelsen (delta) udregnes.

I anden fase, som kaldes backward-fasen, bliver delta-værdien fra første fase fordelt tilbage gennem netværket ved hjælp af de synaptiske vægte. Først når disse to faser er afsluttet, kan et

nyt input-/output-sæt præsenteres for netværket.

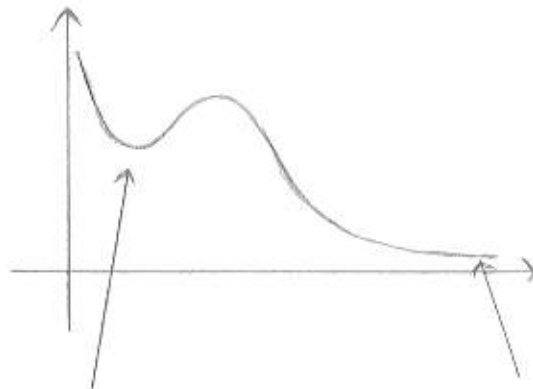
Teknikken bruges i netværk med tre eller flere lag. Ved hjælp af informationen fra forward-fasen er det muligt at bestemme hvor meget hver enkel neuron fra det forrige lag bidrager med til den samlede fejl.

Ved første gennemløb af netværket under træning anvendes tilfældigt genererede værdier for de synaptiske vægte inden for et vist interval og når den samlede afvigelse stabiliseres på et passende lavt niveau, indstilles træningen.

Forskellige versioner af Backpropagation-algoritmen bruges i mange kommercielt tilgængelige værktøjer, fordi metoden som regel finder gode løsninger, giver små netværk, og med kun to skjulte lag kan simulere ganske komplekse og ikke-lineære funktioner.

Backpropagation kræver megen træning, og træningstiden stiger eksponentielt med netværksstørrelsen. Metoden kan give anledning til oscillering, og der er risiko for at blive fanget i et lokalt minimum. Sidstnævnte kan forekomme, hvis systemet finder en output-afvigelse, der er mindre end de omkringliggende afvigelser, men som ikke er den mindst mulige afvigelse i netværket. Se figur 2.11.

Output-afvigelse



Lokalt minimum

Globalt minimum

Figur 2.11 Lokalt og globalt minimum.

## 2.6 Software og hardware tools.

Der findes en række software-værktøjer kommercielt tilgængeligt på markedet til opbygning, træning og test af netværk. Nogle af programpakkerne har en programkode med, som regel en C-kode, så kunden kan implementere sit trænede netværk i traditionel programmering. Nedenfor er angivet tekniske specifikationer for nogle af disse værktøjer:

Produkt: BrainMaker (v2.5)

Firma: California Scientific Software

Pris: 195 \$

Udgivelsesår: 1992

Kompatibel med: PC-MS/DOS, Apple Macintosh/System 7

Minimum RAM: 512 KB

Understøtter EMS: Nej

Understøtter coprocessor: Nej

Mus anbefales: Ja

Programmeringssprog: C

Mulighed for licensrettigheder: Nej

Tillader mønstergenkendelse og optræning fra data input. Design, opbygning, træning og test af neurale netværk.

Produkt: BrainMaker Professional (v2.5)

Firma: California Scientific Software

Pris: 795 \$

Udgivelsesår: 1992

Kompatibel med: PC-MS/DOS

Minimum RAM: 640 KB

Understøtter EMS: Ja

Understøtter coprocessor: Ja

Mus anbefales: Ja

Programmeringssprog: C

Mulighed for licensrettighed: Nej

Tillader mønstergenkendelse og træning ved hjælp af data input. Design, opbygning, træning og test af neurale netværk.

En bog om neurale netværk. Run-time fil i C-code. BrainMaker-pakken kan indlæse ASCII-filer, binære filer samt Lotus- og dBase-filer. Programmet er med pull-down menuer, dialog-bokse, farver og grafik.

Produkt: Braincel (v1.1)

Firma: Promised Land Technologies, Inc.

Pris: 249 \$

Udgivelsesår: 1991

Kompatibel med: Windows 3.X

Minimum RAM: 2 MB

Understøtter EMS: ?

Understøtter coprocessor: Ja

Mus anbefales: Ja

Programmeringssprog: C

Mulighed for licensrettighed: Ja

Et neuralt netværk for Microsoft Excel, som tillader import fra Windows-applikationer. Der er inkluderet 20 Excel function macros, expert mode og Mark Jurik's Back Percolation Algorithm.

Produkt: NeuroShell (v4.1)  
Firma: Ward Systems Group, Inc.  
Pris: 195 \$  
Udgivelsesår: 1991  
Kompatibel med: PC-MS/DOS  
Minimum RAM: 256 Kb  
Understøtter EMS: Ja  
Understøtter coprocessor: Ja  
Mus anbefales: Nej  
Programmeringssprog: ?  
Mulighed for licensrettighed: Ja  
En programskal der sætter neurale net til at træffe beslutninger.  
Mimics biologiske læreproces benyttes.

Produkt: NueX for Windows (v1.0n)  
Firma: Charles River Analytics, Inc.  
Pris: 295 \$  
Udgivelsesår: 1992  
Kompatibel med: Windows 3.X  
Minimum RAM: 4 MB  
Understøtter EMS: Ja  
Understøtter coprocessor: ?  
Mus anbefales: ?  
Programmeringssprog: ?  
Mulighed for licensrettighed: ?  
Neuralt netværk med grafisk editor og med manuel eller automatisk arkitektur-specifikation. Bruger Backpropagation træningsmetode.

Produkt: NeuralWorks Professionel II Plus

Firma: NeuralWare, Inc.

Pris: 1895 \$ til 4995 \$

Udgivelsesår: 1987

Kompatibel med: PC-MS/DOS, Sun/SunOS, Apple Macintosh Plus.

Minimum RAM: 1 MB

Understøtter EMS : Ja

Understøtter coprocessor: Ja

Mus anbefales: Ja

Video adapter (minimum): EGA

Programmeringssprog: C

Mulighed for licensrettighed: Ja

Udviklingsværktøj til neurale net. Inkluderet 25 netværkstyper og mulighed for at konstruere egne netværk. 14 læreregler, 10 forskellige overføringsfunktioner og 11 summationsfunktioner. Kan indlæse standard ASCII-filer produceret af Lotus 1-2-3, dBase III eller Excel.

Produkt: Neurocomputing Library

Firma: NeuroSym Corp.

Pris: 199 \$, 379 \$ (med source-kode)

Udgivelsesår: 1992

Kompatibel med: Windows 3.X

Minimum RAM: 2 MB

Understøtter EMS : Ja

Understøtter coprocessor : Nej

Mus anbefales: Ja

Programmeringssprog: C

Mulighed for licensrettighed: Nej

C-bibliotek med neurale netværksfunktioner indeholdende 12 forskellige netværkstyper. Programeksempler og mulighed for distribution af applikationer. Understøtter Turbo C (v2.0), Turbo C++ (v1.0), Microsoft C (v5.1) og Zortech C++:

Der findes også hardware værktøj, som gør det muligt at øge træningshastigheden af neurale netværk i PC. Nedenfor er angivet nogle af disse hardware-produkters tekniske specifikationer:

Produkt: NeuroBoard

Firma: Ward Systems Group, Inc.

Pris: 1695 \$ til 3295 \$

Lanceringsår: 1990

Funktion: Neural network coprocessor board

Kompatibel med: ISA (8-bit)

RAM: 64KB til 1 MB

Størrelse af board: Halv længde

Produkt: Professional Accelerator System

Firma: LORAL Space Information Systems.

Pris: 9750 \$ og 13000 \$

Lanceringsår: ?

Funktion: Neural network coprocessor board

Kompatibel med: AT/ISA, IBM AT, PS/2 el. kompatible med min.  
640 KB RAM.

RAM: 5 MB og 32 MB

Størrelse af board: Hel længde

Board'et har en hastighed, som er 57 gg hurtigere end en 486/33, og fås i to konfigurationer:

1) Maksimal konfiguration med 32 MB RAM, som kan behandle netværk med 3,6 millioner forbindelser og 4 millioner inputs\* facts (facts=trænings-input/output), for eksempel 200 inputs og 20000 facts eller 500 inputs og 8000 facts.

2) Minimum konfiguration med 5 MB RAM, som kan behandle netværk med 600.000 forbindelser og 250.000 inputs\*facts, for eksempel 100 inputs og 2500 facts eller 250 inputs og 1000 facts.