Backprop

Symbolische KI — Konnektionismus - Muster - } K-NN Clustering

Logik

(sichere Information)

Muster → Aussage

Probabilistische KI

unsichere Informationen

a → b [0,9]

[0,7] b → c Konfidenz

.....

a → c ? f(0,7; 0,9)

Bayes-Netze

Parameter

l_1: Länge, Winkel, Öffnung

l_2: " " "

Regel:

1) If öffnung ≤ 0,2

Then

{ teste Null }

if

if

:

else

=

Symbolische KI

→ Programmieren ⇒ Regelwerk

"Logik"

Konnektionistische KI

→ Automatisch

(Machine learning)

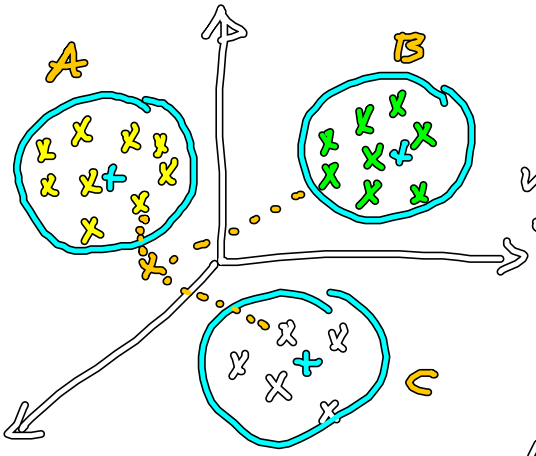
"lernen"

Backprop_

Lernen

Objekte ⟶ Features ⟶ Erkennung

$\mathbb{R}^n$



Rep. 1

Alle Daten

Unbekannter → KNN

Nachbarn

Rep. 2

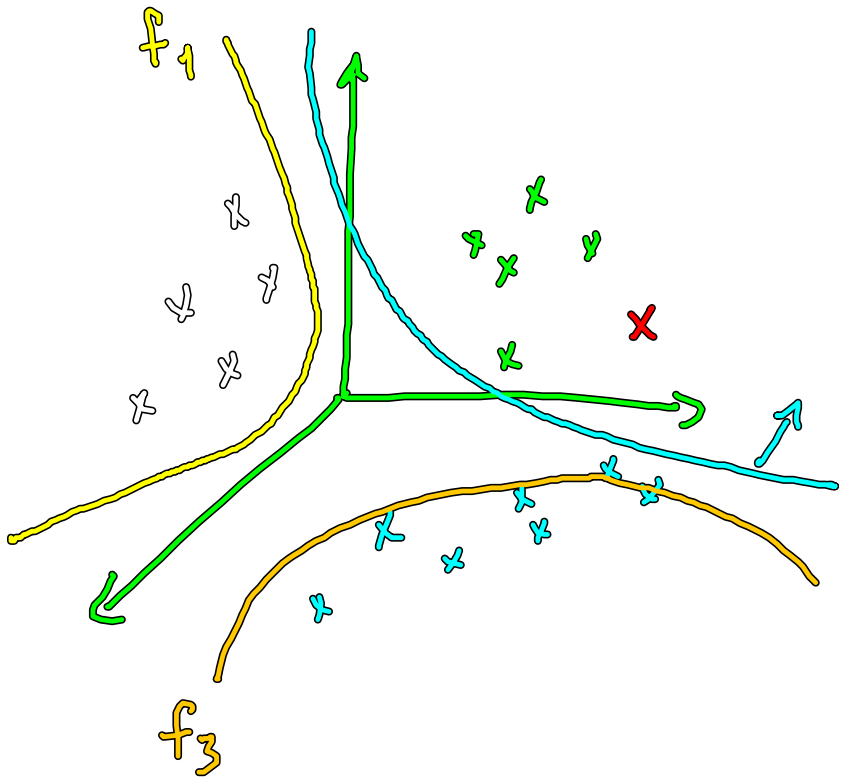
Clusterzentren

Radius

Abstand ≤ Radius

Effizient

Radius < Zentrum



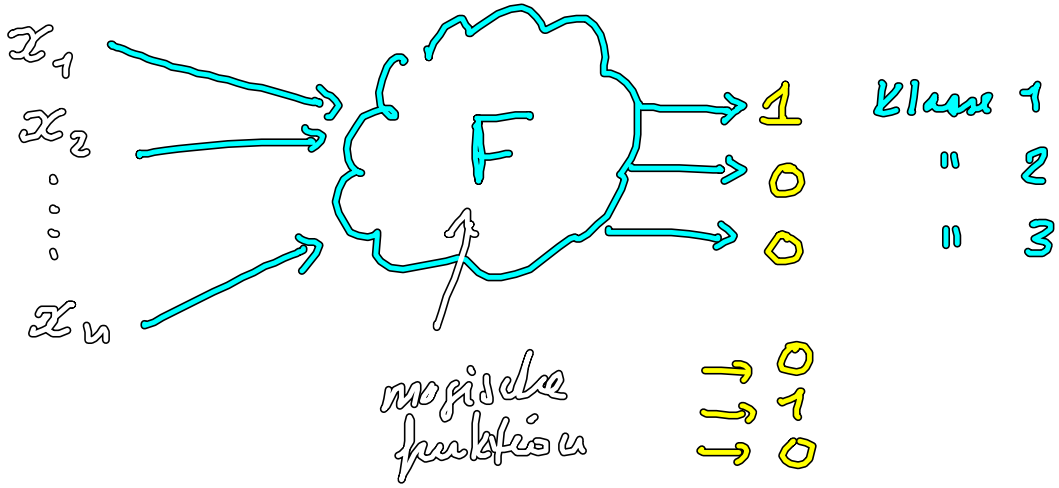
Trennungs-flächen

- einfach

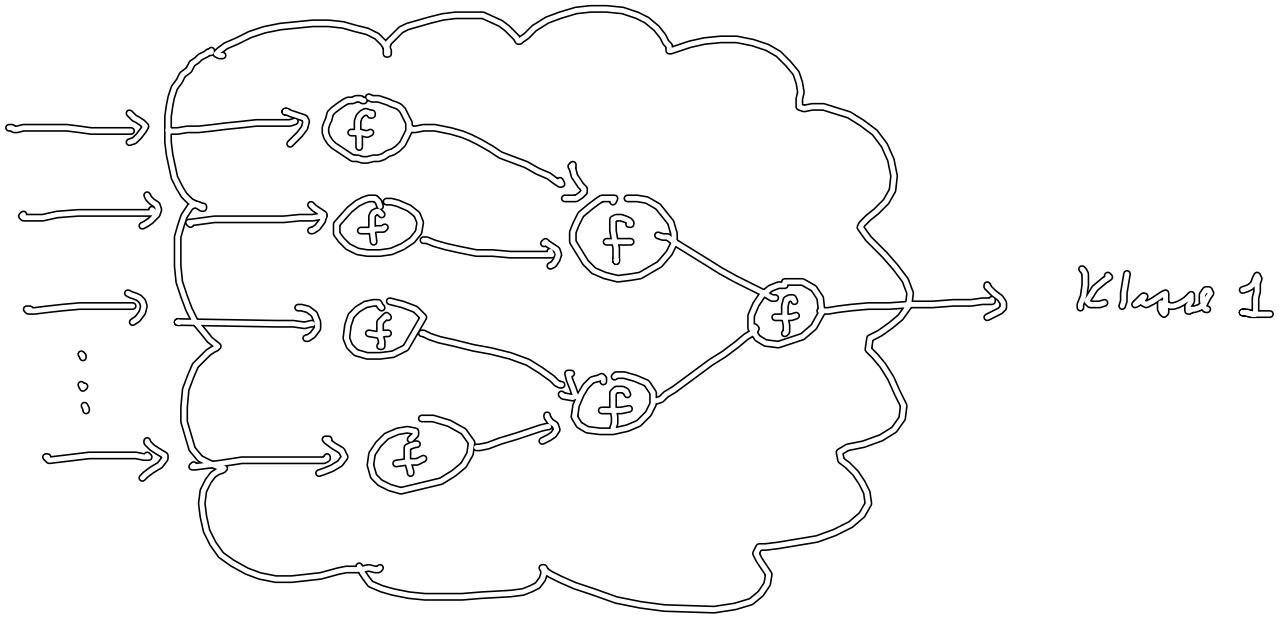
Netz - Neuronal

Funktional

Merkmale



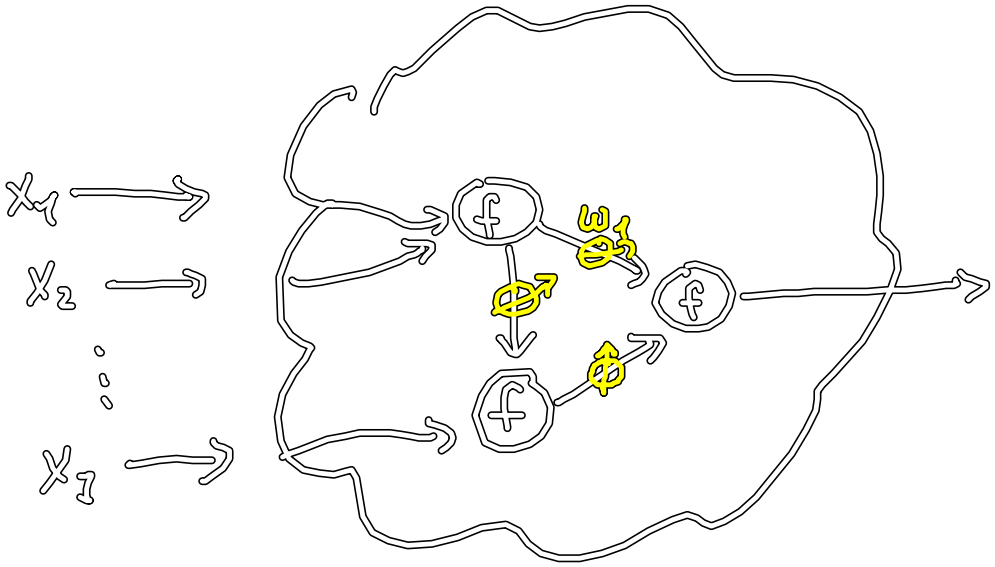
→ 0,7 Klasse 1
→ 0,3 " 2
→ 0,0 " 3

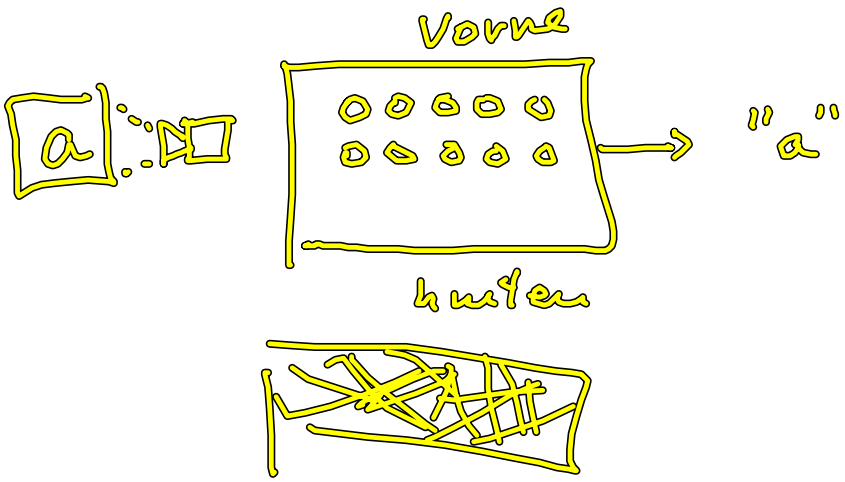


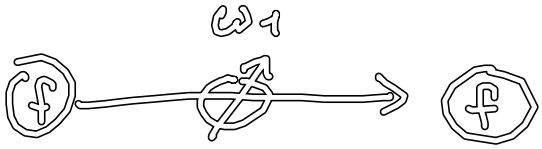
Einschränkung:

- Elementarfunktionen sind gleich
- Topologie ist fest
- keine Zyklen
- freie Parameter sind Gewichte der Kanten (modulieren)

$0{,}7 \xrightarrow{0{,}2} 0{,}14 = 0{,}7 \times 0{,}2$

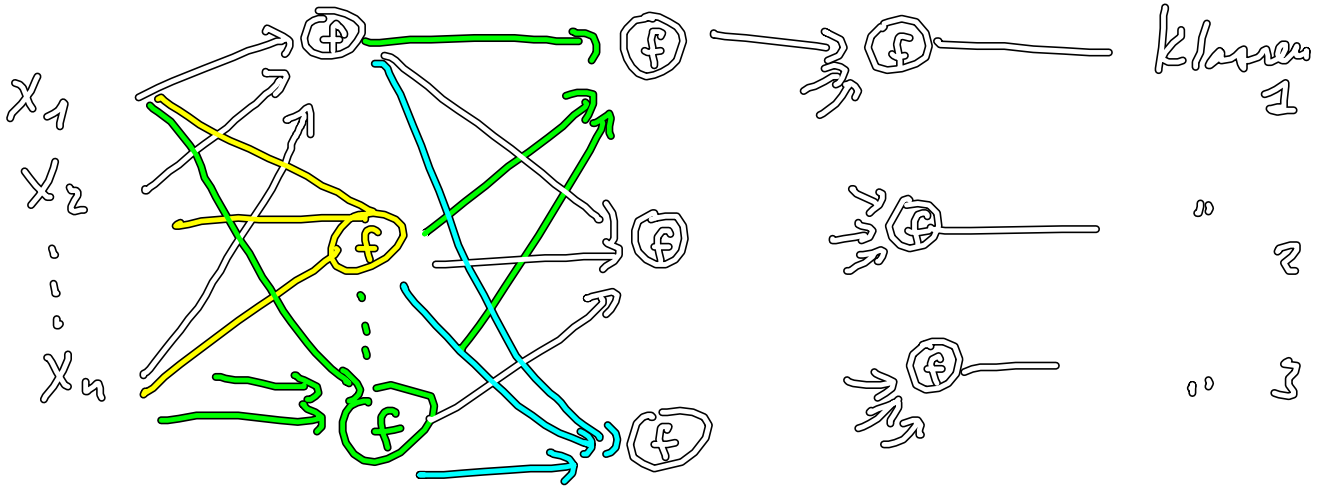


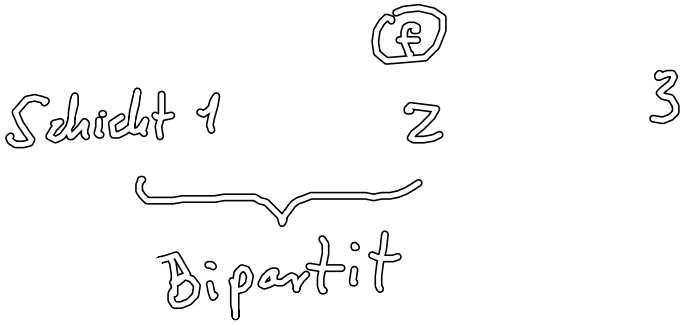




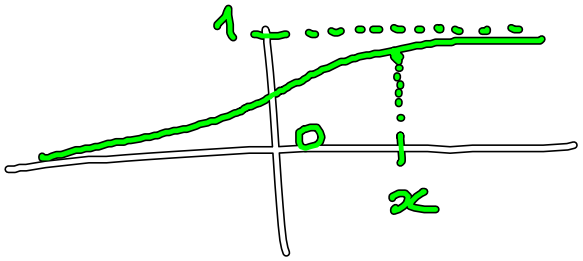
$w_1 = 0$

Topologie





Elementarfunktion

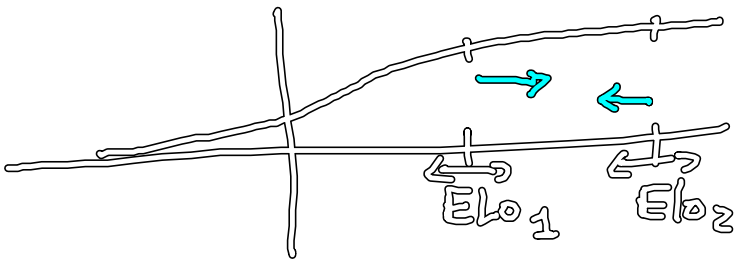


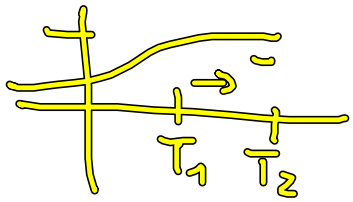
Sigmoide

logistische Funktion

Vorschlag FIFA

ELO-Rating



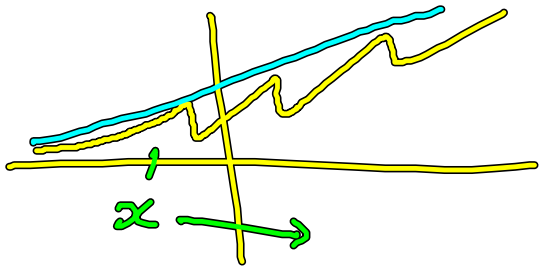


$$f(x) = \frac{1}{1 + e^{-x}}$$

$f(0) = \frac{1}{2}$

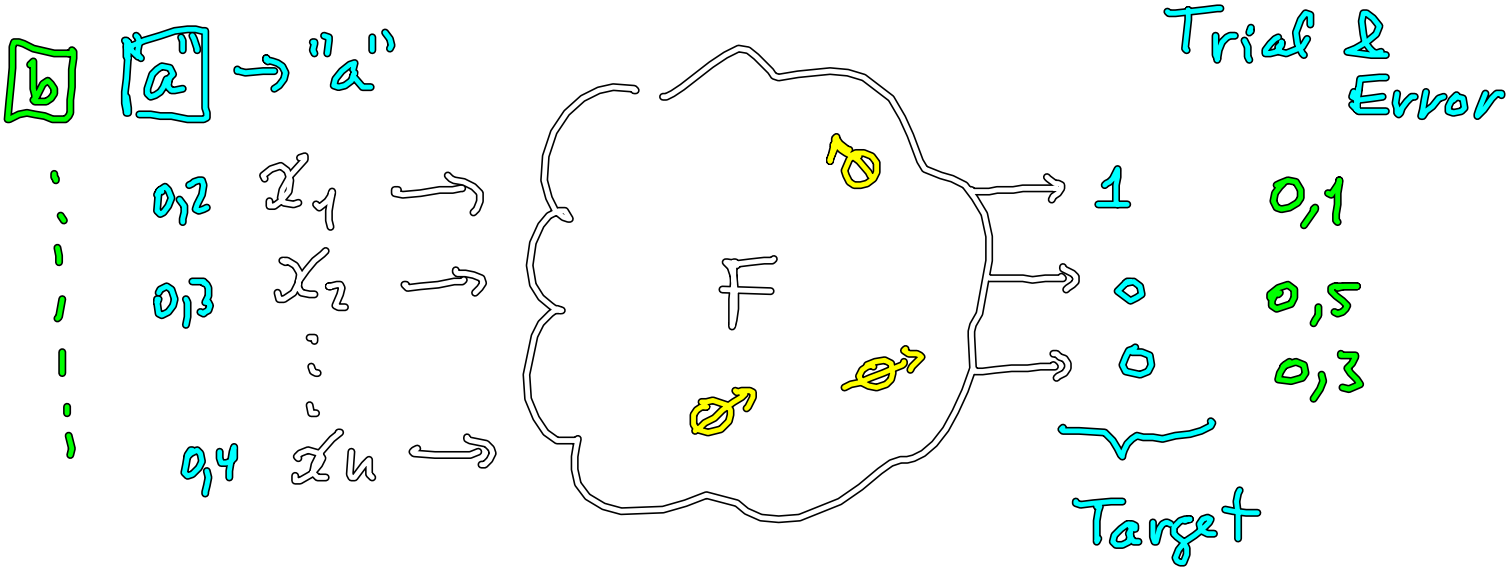
$f(\infty) = 1$

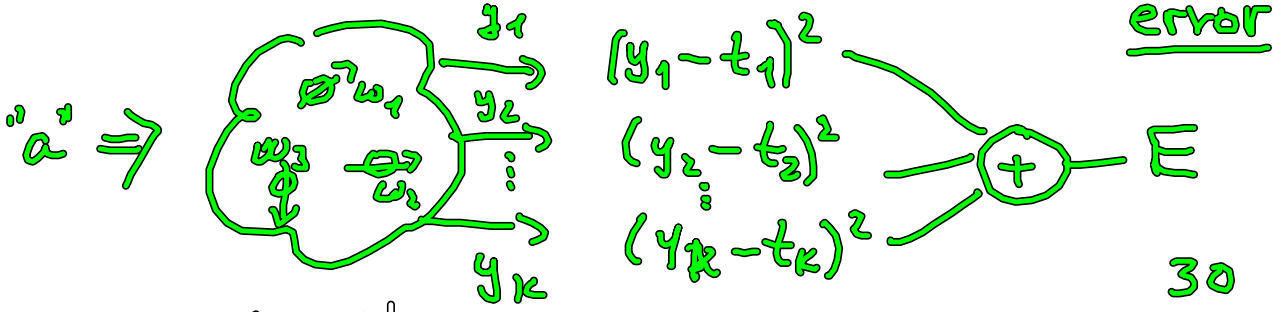
$f(-\infty) = 0$

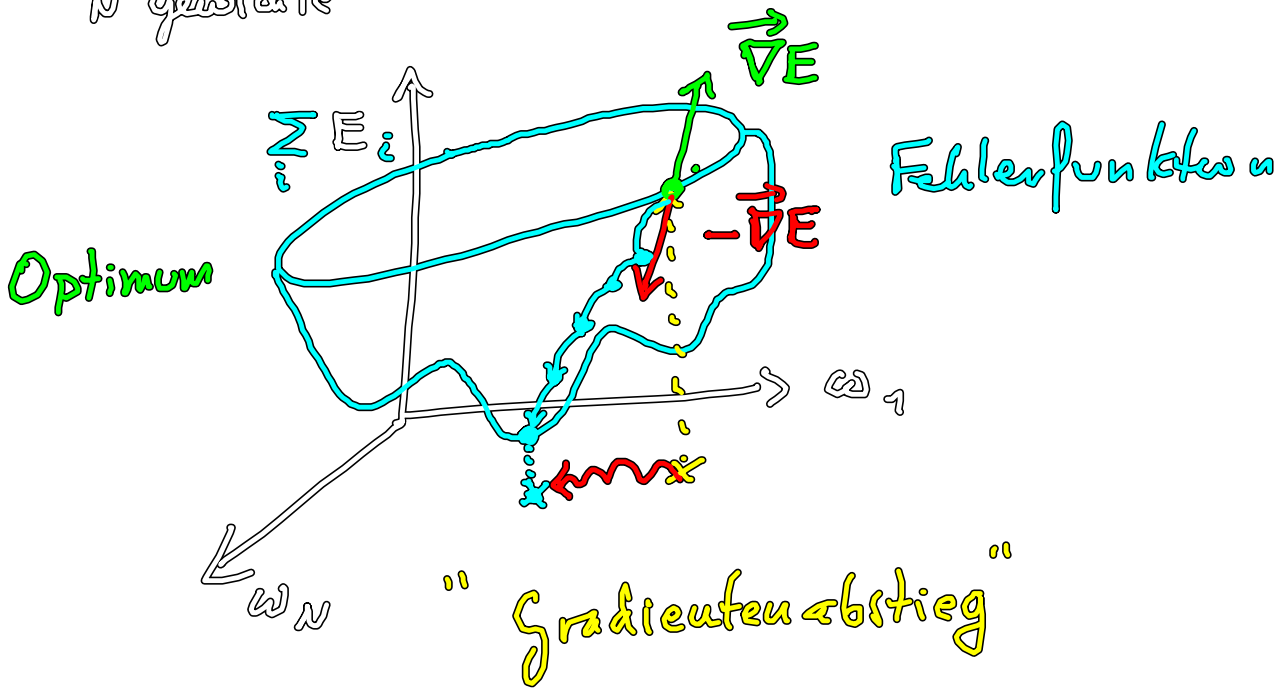


$$f'(x) = \underbrace{f(x)}\left(1 - \underbrace{f(x)}\right)$$

$f(x)$ $\quad$ $f(x)$







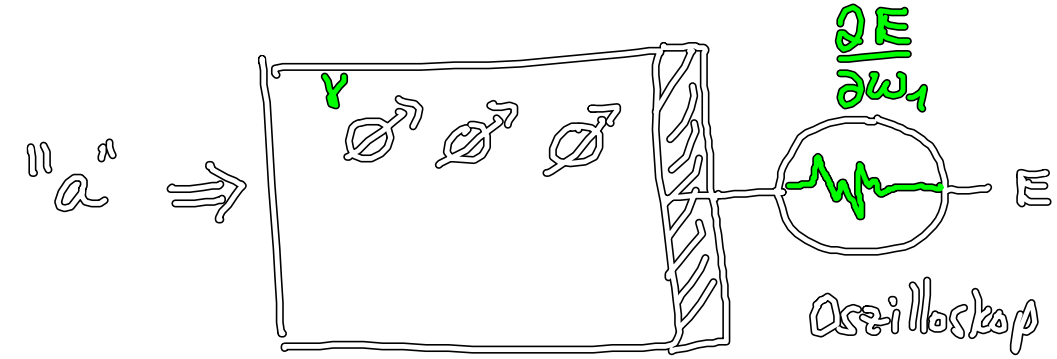
$$\vec{\nabla}E = \left(\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \dots, \frac{\partial E}{\partial w_N} \right)$$

$$w_1 := w_1 - \gamma \frac{\partial E}{\partial w_1}$$

$$w_2 := w_2 - \gamma \frac{\partial E}{\partial w_2}$$

$$\vdots$$

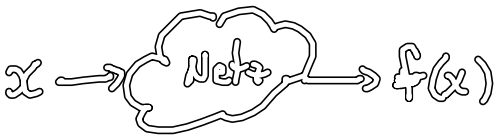
$$w_N := w_N - \gamma \frac{\partial E}{\partial w_N}$$



Ableitungen

kleines Problem

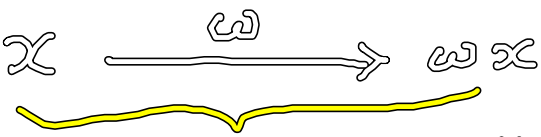
$f'(x)$?



Backprop-Algorithmus

Fälle

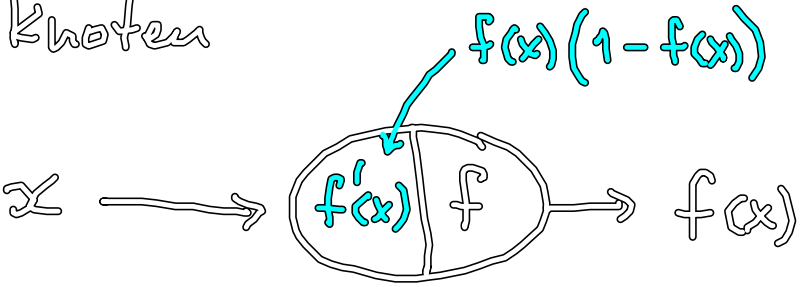
1) Kante



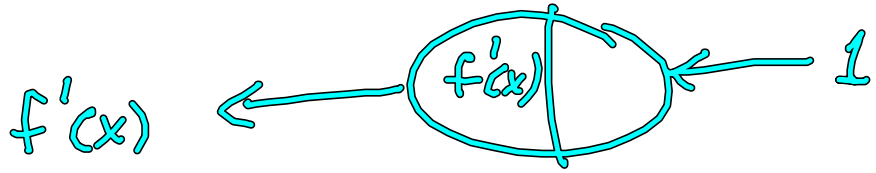
$$\frac{d(wx)}{dx} = w \xleftarrow{\text{Backprop}} 1$$

Traversierungs-weit

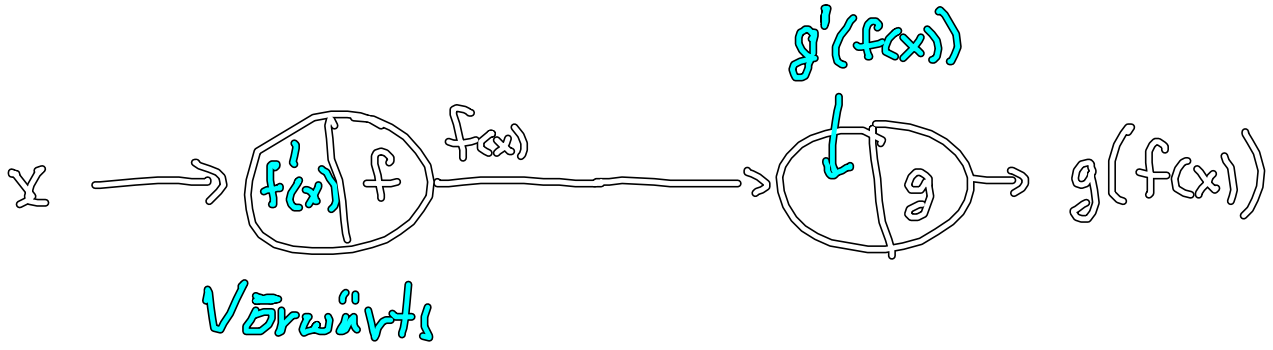
2) Ein Knoten



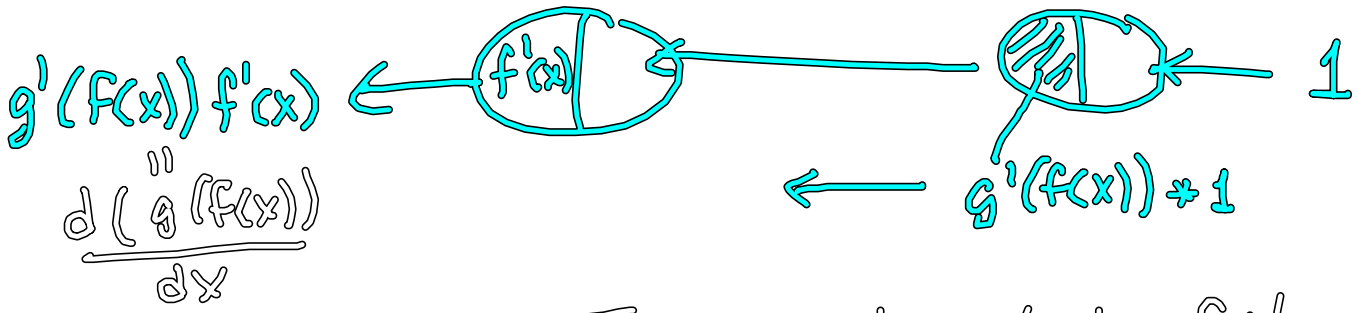
Vorwärts



3) Funktionskomposition



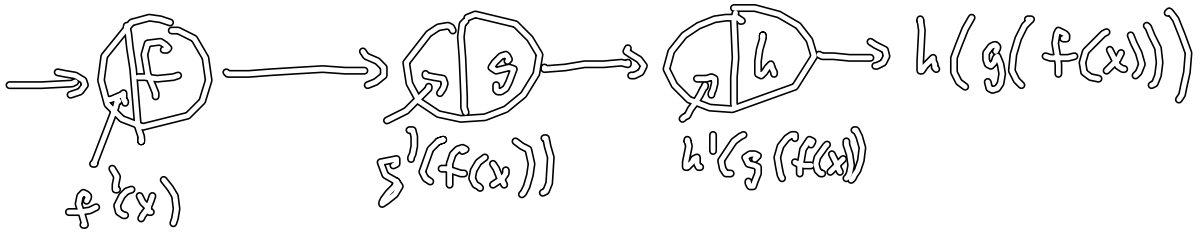
Vorwärts



$\frac{d(g(f(x)))}{dx}$ = $g'(f(x)) f'(x)$

← Trave.wert * linke Seite

4) Beliebig Funktionskomp.



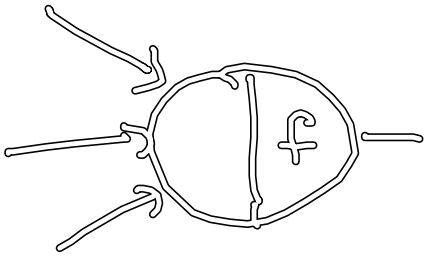
$h'(g(f(x))) \cdot g'(f(x)) \cdot f'(x)$

=

$\frac{d}{dx}(h(g(f(x))))$

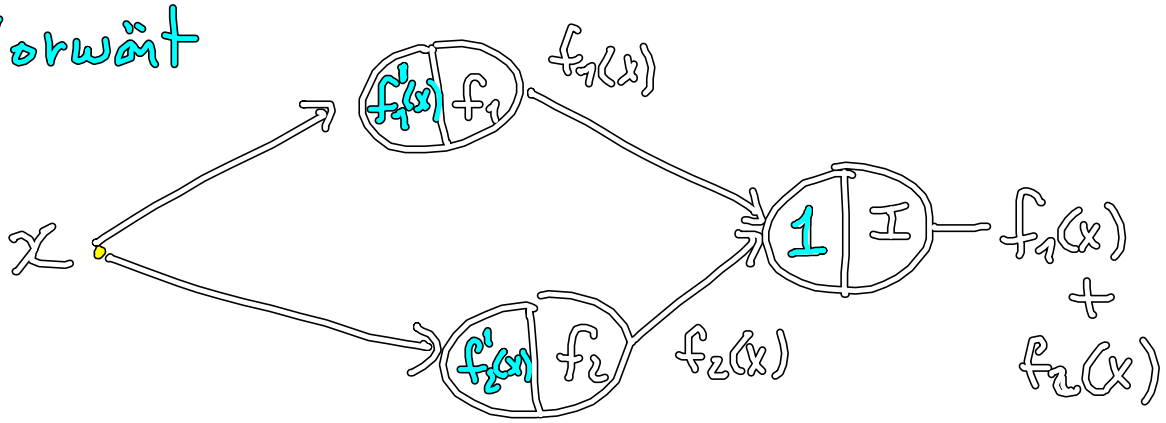
5) Addition von Funktionen

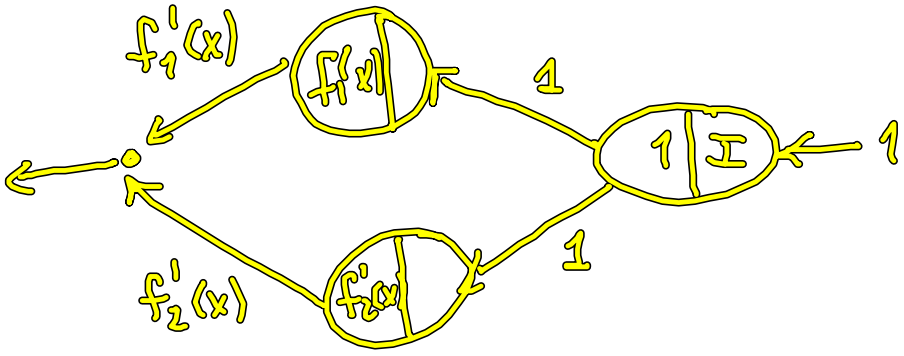
Vereinbarung



addiert Input

Vorwärt

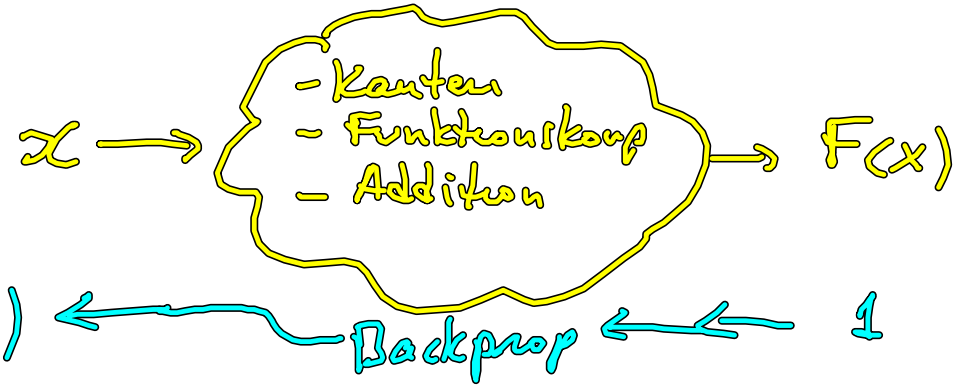




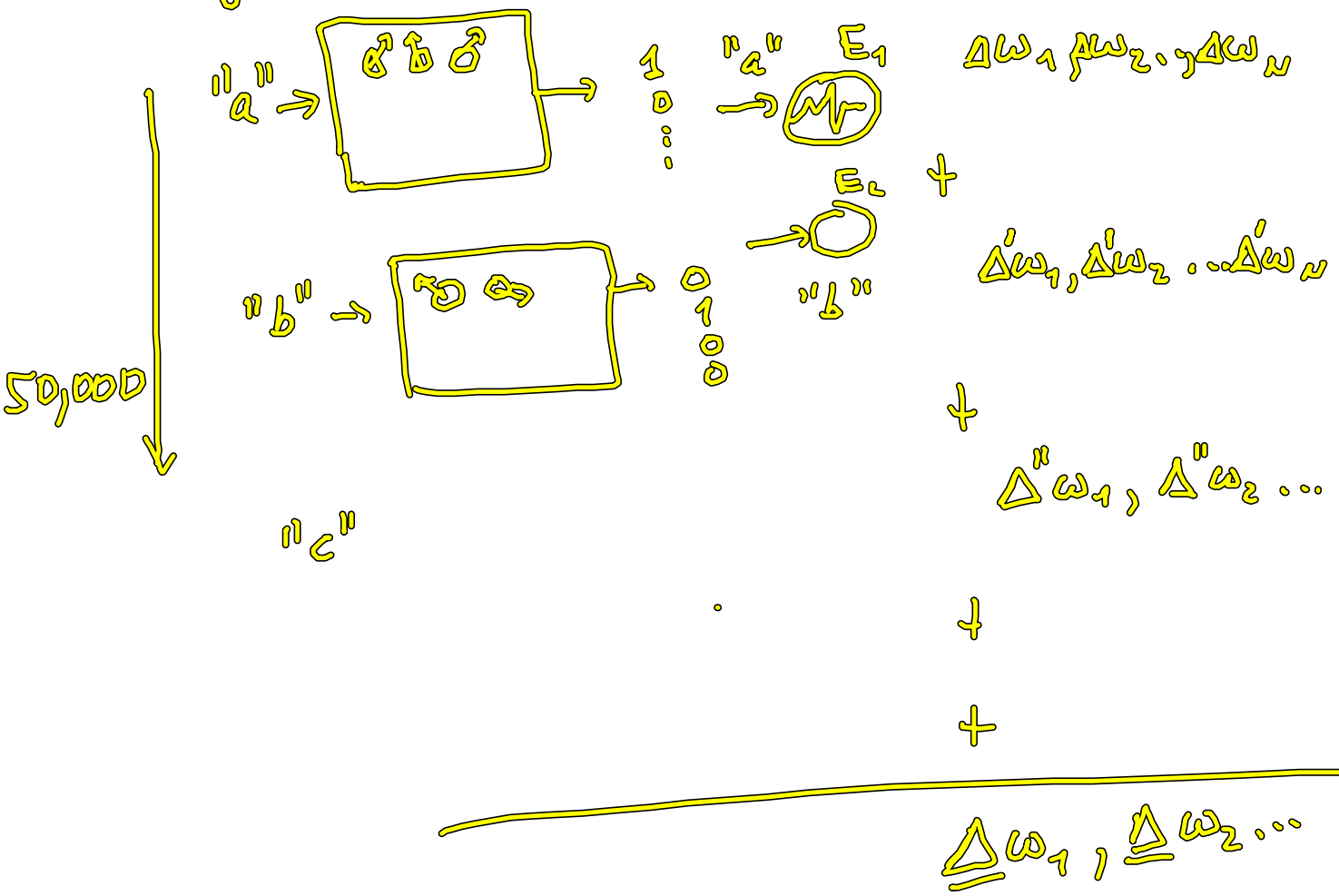
$f_1'(x)$

$f_1'(x)$

" $\frac{d}{dx}(f_1(x) + f_2(x)$

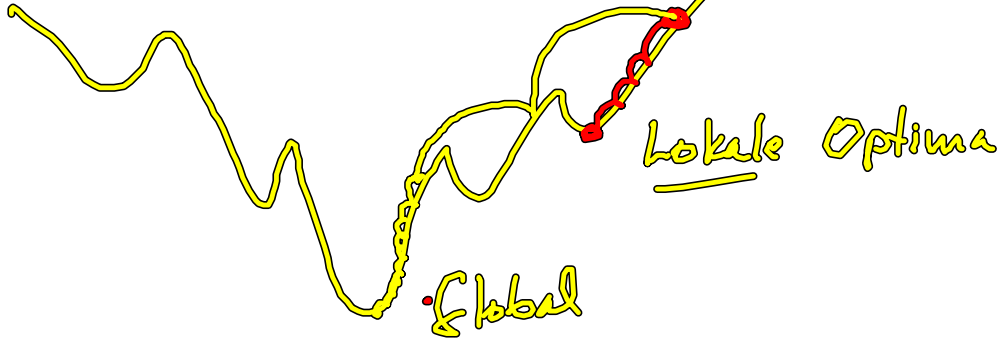


Trainingsmenge

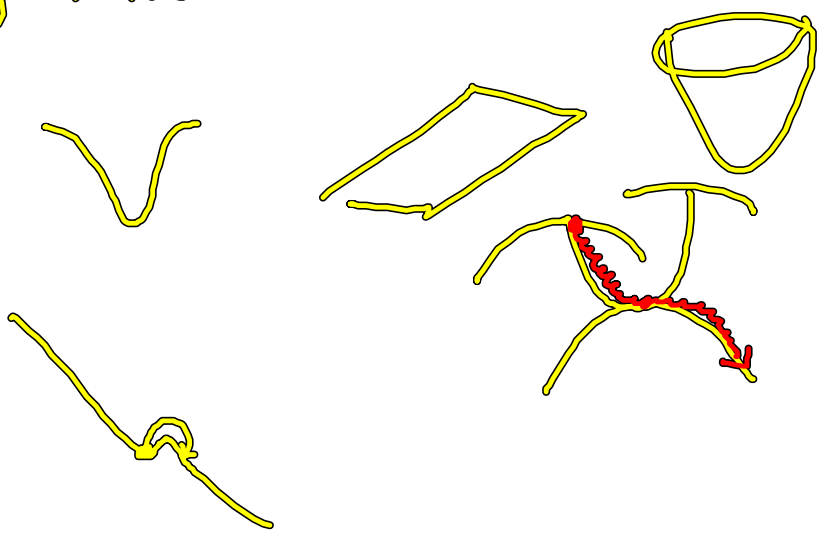


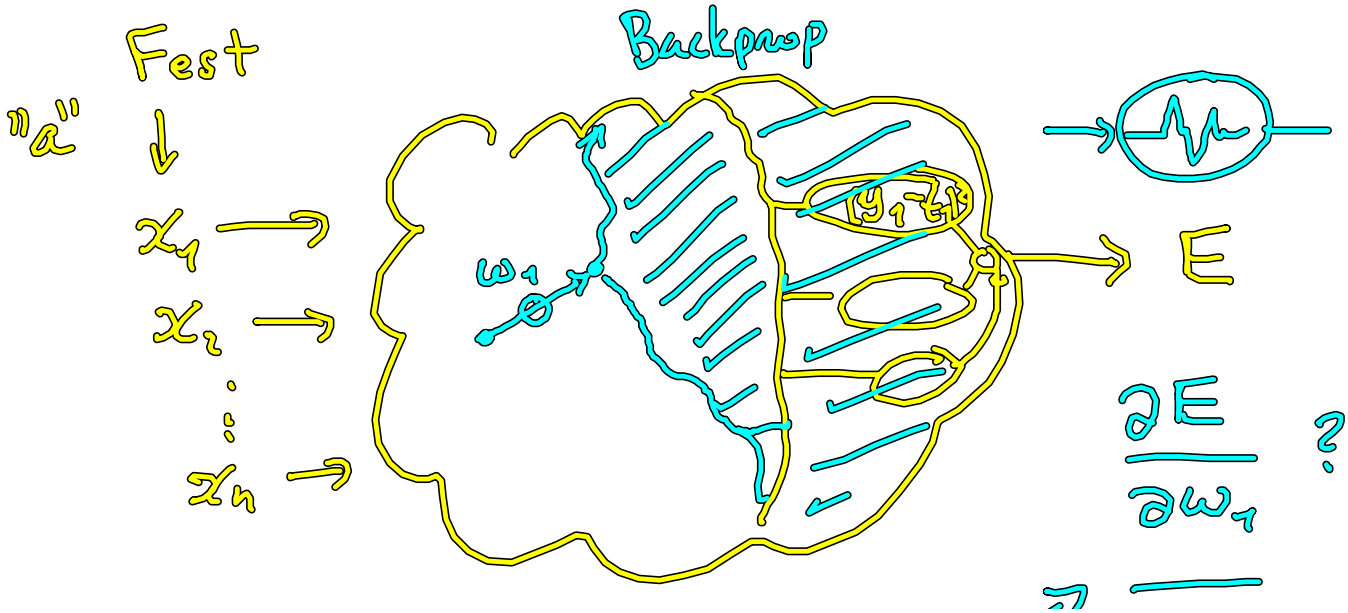
Gradientenabstieg

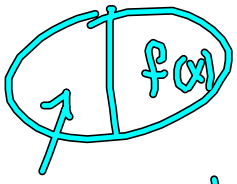
Simulated Annealing



1000 Gewichte







$f(x)(1-f(x))$

ω_1

1
1,1

E
2

Backprop von der Ausgabe bis

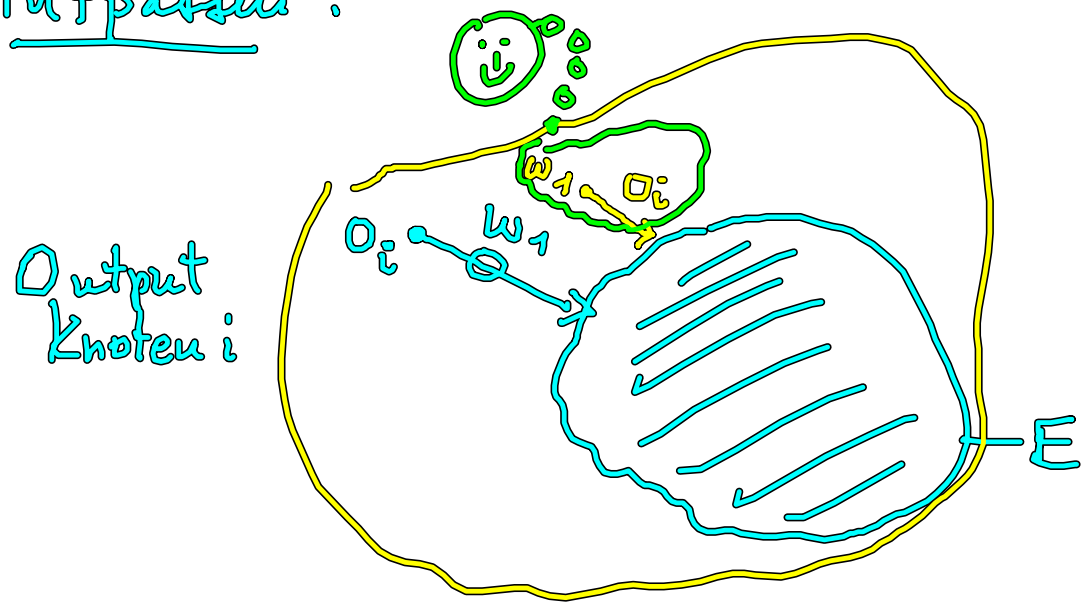
2,2 ω_1

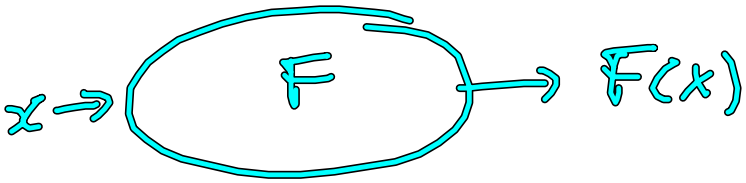
$\Delta E = 0{,}2$

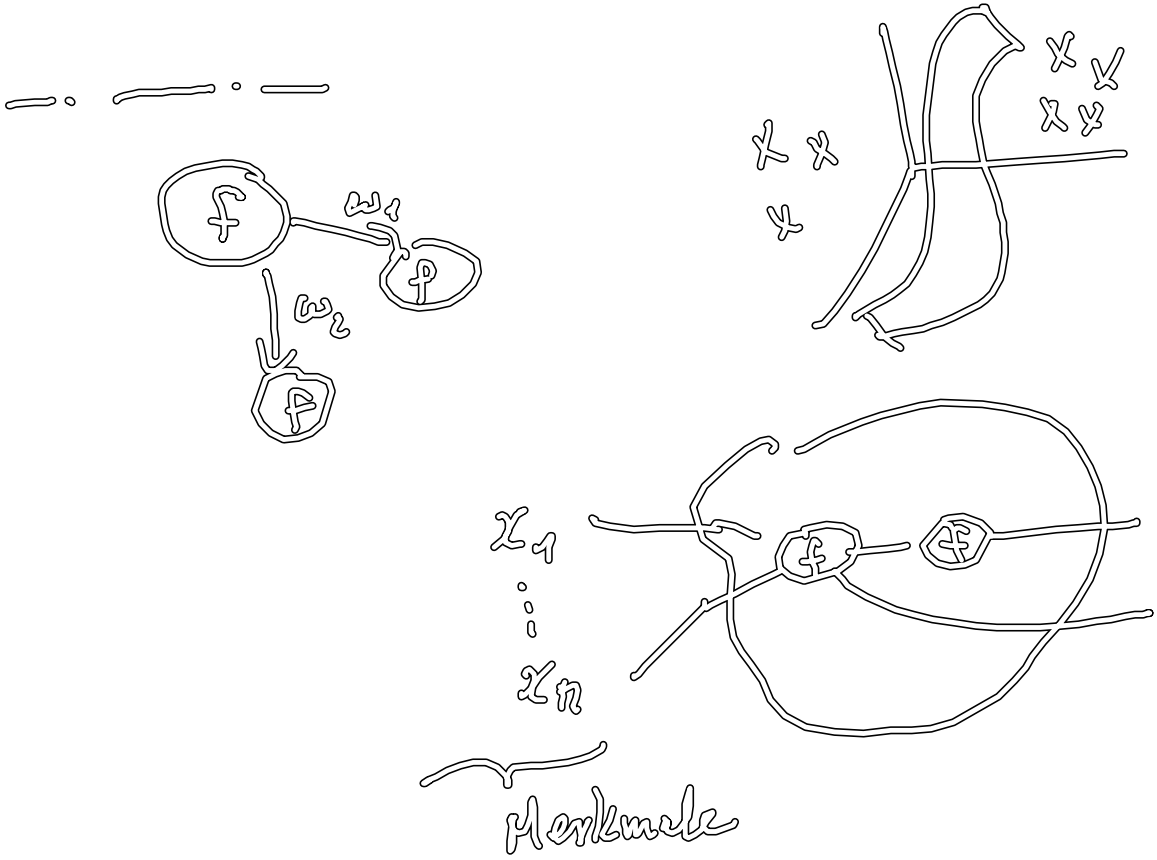
$\Delta \omega_1 = 0{,}1$

$\frac{\Delta E}{\Delta \omega_1} = 2$

Aufpassen!






f
w_1
p
w_2
f
x_1
x_n
Merkmale