Machine Intelligence 1

1.2 Connectionist Neurons

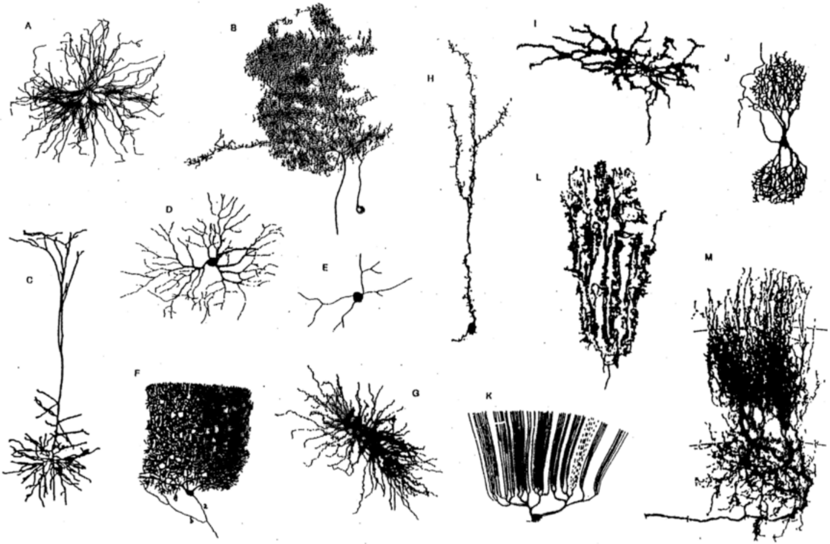
Prof. Dr. Klaus Obermayer

Fachgebiet Neuronale Informationsverarbeitung (NI)

WS 2017/2018

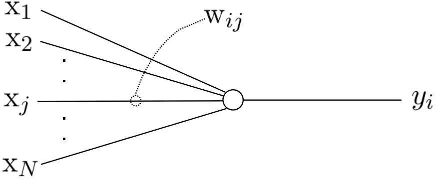
1.2.1 1. Input-Output Relationship

Biological neurons



Connectionist neurons

Input-output relationship: linear filter with a static non-linearity



$$\rightarrow \quad y_i = f\Big(\underbrace{\sum_j w_{ij} x_j - \theta_i}_{h_i} \Big)$$

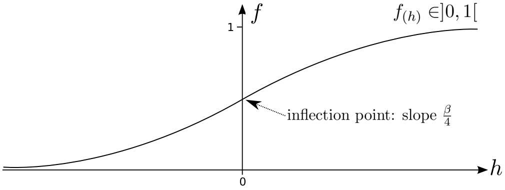
$\underline{\mathbf{x}}$:	input vector with components x_j
y_i :	scalar output of neuron i
$\underline{\mathbf{w}}_i$:	weight vector of neuron i with components w_{ij}
θ_i :	threshold of neuron i
h_i :	total input of neuron i
f :	transfer function

Examples: rate neurons, mean-field approximation, receptive field models, etc.

Typical transfer functions (1)

logistic function

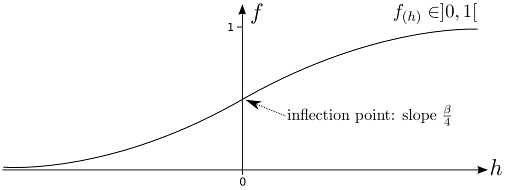
$$f_{(h)} = \frac{1}{1+\exp(-\beta h)}$$



Typical transfer functions (1)

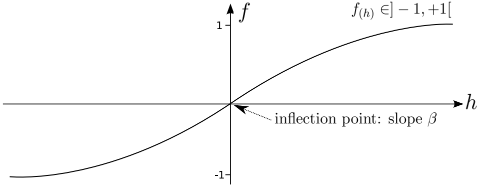
logistic function

$$f_{(h)} = \frac{1}{1+\exp(-\beta h)}$$



hyperbolic tangent

$$f_{(h)} = \tanh \beta h$$



Typical transfer functions (2)

$$
\begin{aligned}
\frac{1}{1+e^{-x}} &= \frac{e^{\frac{x}{2}}}{e^{\frac{x}{2}}+e^{-\frac{x}{2}}} \\
&= \frac{1}{2}\Bigg\{ \underbrace{\frac{e^{\frac{x}{2}} - e^{-\frac{x}{2}}}{e^{\frac{x}{2}} + e^{-\frac{x}{2}}}}_{\tanh \frac{x}{2}} + \underbrace{\frac{e^{\frac{x}{2}} + e^{-\frac{x}{2}}}{e^{\frac{x}{2}} + e^{-\frac{x}{2}}}}_{1} \Bigg\} \\
&= \frac{1}{2}\Big(\tanh \frac{x}{2} + 1 \Big)
\end{aligned}
$$

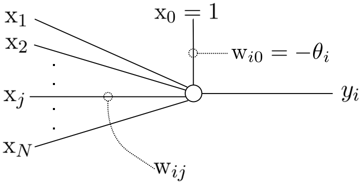
Typical transfer functions (2)

$$
\begin{aligned}
\frac{1}{1+e^{-x}} &= \frac{e^{\frac{x}{2}}}{e^{\frac{x}{2}}+e^{-\frac{x}{2}}} \\
&= \frac{1}{2}\left\{ \underbrace{\frac{e^{\frac{x}{2}} - e^{-\frac{x}{2}}}{e^{\frac{x}{2}} + e^{-\frac{x}{2}}}}_{\tanh \frac{x}{2}} + \underbrace{\frac{e^{\frac{x}{2}} + e^{-\frac{x}{2}}}{e^{\frac{x}{2}} + e^{-\frac{x}{2}}}}_{1} \right\} \\
&= \frac{1}{2}\left(\tanh \frac{x}{2} + 1 \right)
\end{aligned}
$$

transformation

1. scale all input weights w_{ij} or slope parameter β by 2
2. shift output by -1
3. scale output by 2

Shortcut notation for threshold / bias



$$y_i = f\left(\sum_j w_{ij} x_j\right) = f(\underline{\mathbf{w}}_i^T \underline{\mathbf{x}})$$

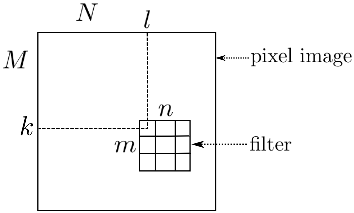
Note on notation

$\underline{\mathbf{w}}$ will be used for $\begin{pmatrix} w_1 \\ \vdots \\ w_N \end{pmatrix}$ as well as for $\begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_N \end{pmatrix}$

$\underline{\mathbf{x}}$ will be used for $\begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix}$ as well as for $\begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_N \end{pmatrix}$

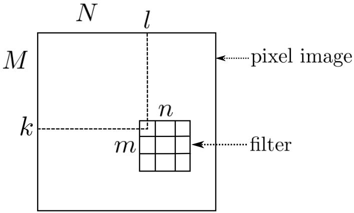
1.2.2 2. Feature Detection and Evaluation

Linear filters and feature detection



$$\underbrace{y_{kl}}_{\text{strength of feature}} = \sum_{i,j=0}^{m,n} \underbrace{\mathrm{w}_{ij}}_{\text{filter coefficients}} \mathrm{x}_{\underbrace{(k+i)(l+j)}_{\text{pixel value}}}$$

Linear filters and feature detection



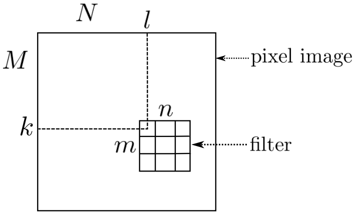
$$\underbrace{y_{kl}}_{\text{strength of feature}} = \sum_{i,j=0}^{m,n} \underbrace{w_{ij}}_{\text{filter coefficients}} \underbrace{x_{(k+i)(l+j)}}_{\text{pixel value}}$$

Example filters w for points and edges

-1	-1	-1
-1	+8	-1
-1	-1	-1

point filter

Linear filters and feature detection



$$\underbrace{y_{kl}}_{\text{strength of feature}} = \sum_{i,j=0}^{m,n} \underbrace{\mathrm{w}_{ij}}_{\text{filter coefficients}} \; \mathrm{x}_{\underbrace{(k+i)(l+j)}_{\text{pixel value}}}$$

Example filters w for points and edges

-1	-1	-1
-1	+8	-1
-1	-1	-1

point filter

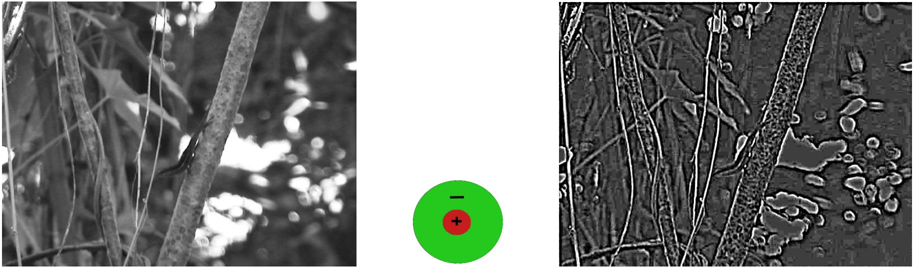
-1	0	+1
-2	0	+2
-1	0	+1

+1	+2	+1
0	0	0
-1	-2	-1

edge filter (Sobel filter)

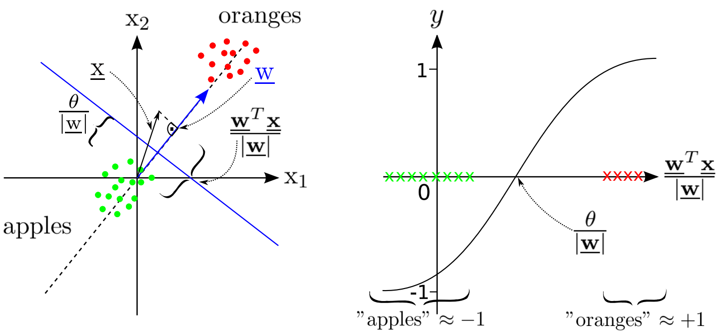
$\Rightarrow$ $\underline{\mathbf{w}}$ describes a feature and is often called receptive field.

Center-surround filters



from: http://www.union.edu/academic_depts/bioengineering/visual_motion/visual_tour/center_surround.php

Detection / evaluation of features



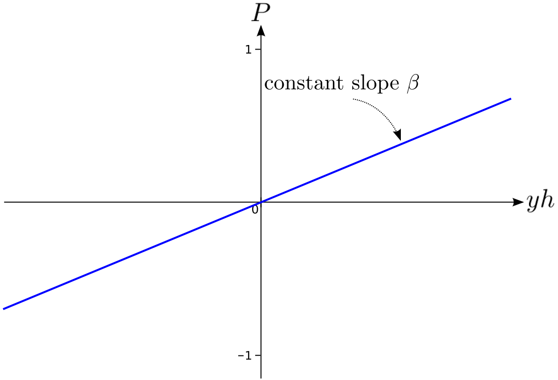
$$\underbrace{\underline{\mathbf{x}} \longrightarrow \underline{\mathbf{w}}^T\underline{\mathbf{x}}}_{\text{feature detection}} \underbrace{\longrightarrow f_{(\underline{\mathbf{w}}^T\underline{\mathbf{x}}-\theta)}}_{\text{evaluation}}$$

$\Rightarrow$ Partitioning of feature space into two half-spaces

1.2.3 3. Special Transfer Functions

Special transfer functions

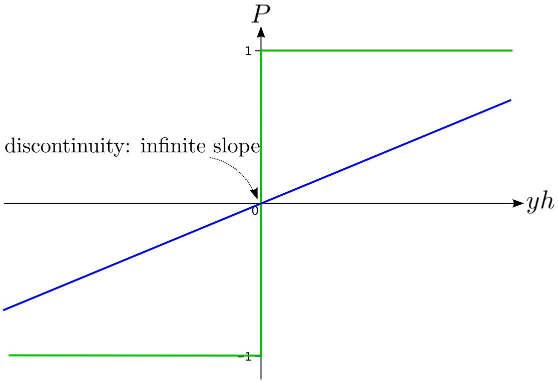
Linear neuron: $f(h) = \beta h$ extraction of linear features



Special transfer functions

Linear neuron: $f(h) = \beta h$ extraction of linear features

Binary neuron: $f(h) = \text{sign}(h)$ classification $\rightarrow$ perception



Special transfer functions

Linear neuron: $f(h) = \beta h$ extraction of linear features

Binary neuron: $f(h) = \mathrm{sign}(h)$ classification $\rightarrow$ perception

Stochastic binary neuron: $P(y \rightarrow -y) = \frac{1}{1+\exp(\beta y h)}$ noise parameter β