

MATLAB (Version 5.3.1 (R11.1))

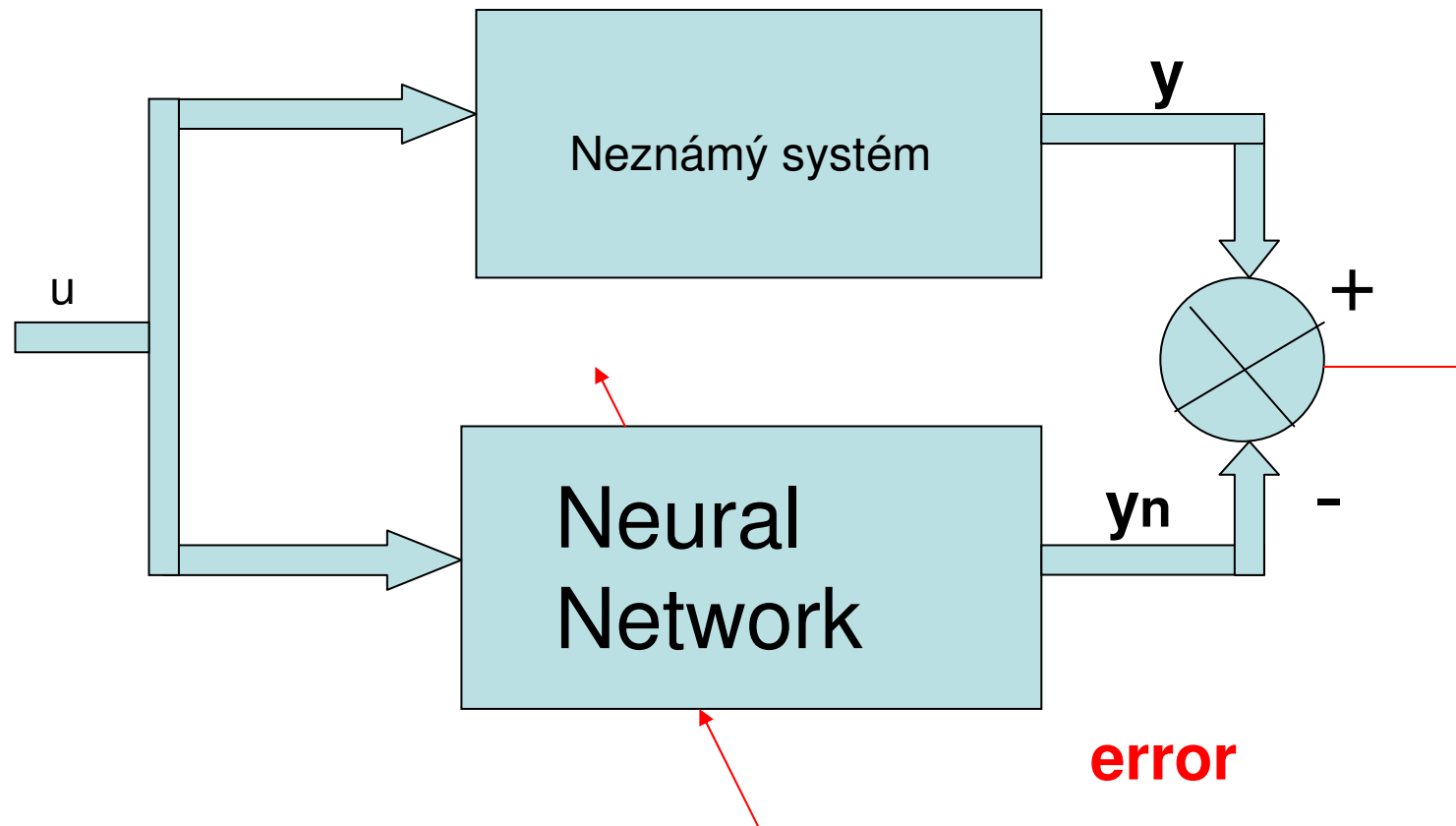
Neural Network Toolbox (3.0.1)

Co je třeba (také) vědět pokud navrhujeme NN

- Je třeba vědět jaký typ sítě je pro danou aplikaci vhodný, typ přenosové funkce $f(*)$, zda se jedná o klasifikaci nebo identifikaci, adaptabilní řízení, zda řešíme ‘lineární’ nebo nelineární problém, a další ...

Jak pracuje NN síť

Příklad identifikace neznámého systému

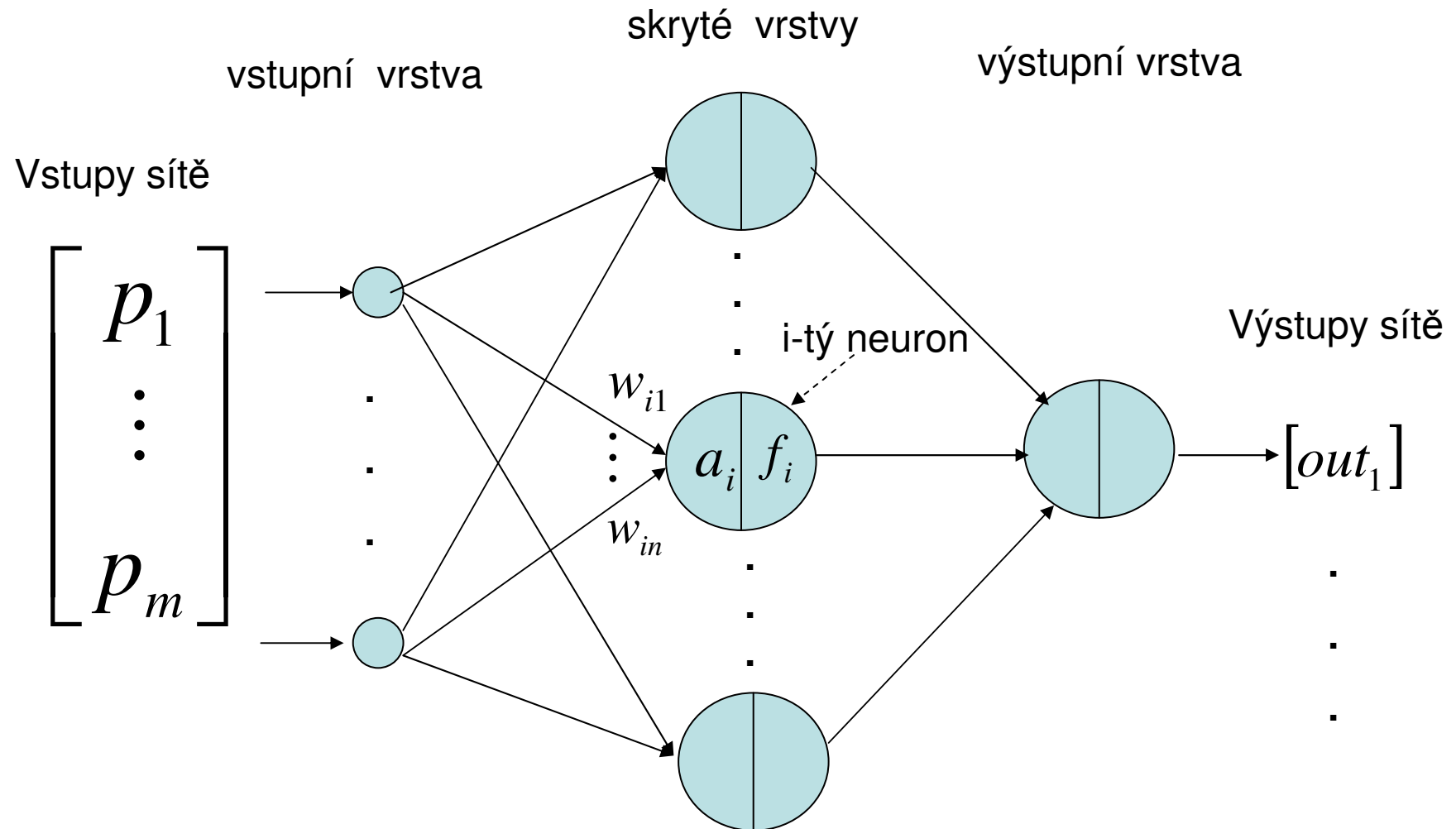


Váhy neuronů v síti se ladí podle zvoleného pravidla učení (Delta, Hebb, Back Propagation,...)

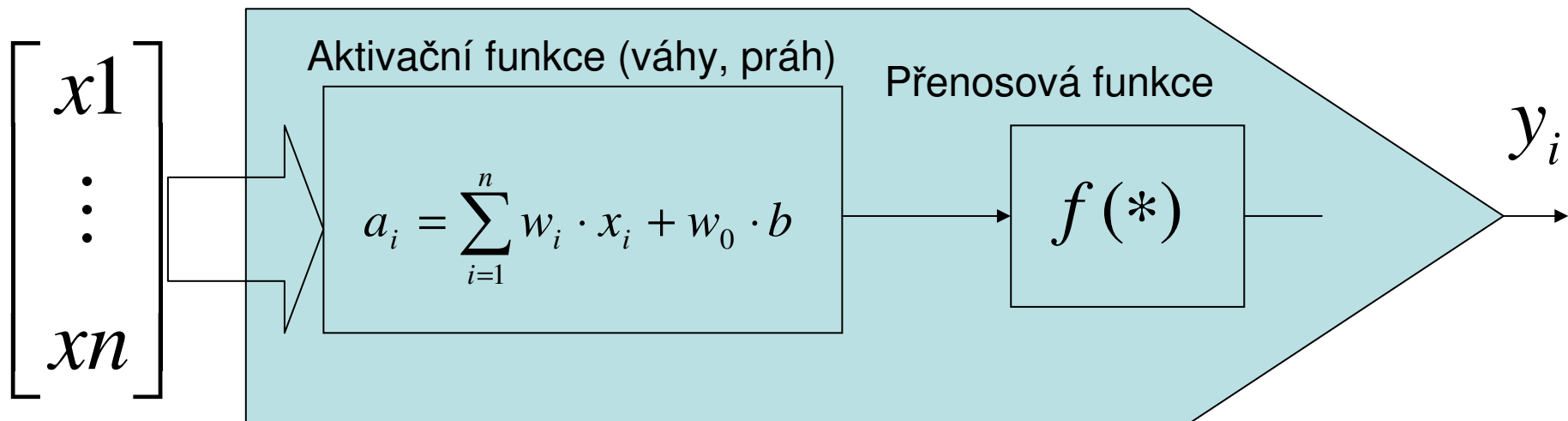
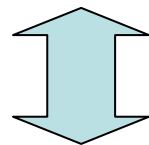
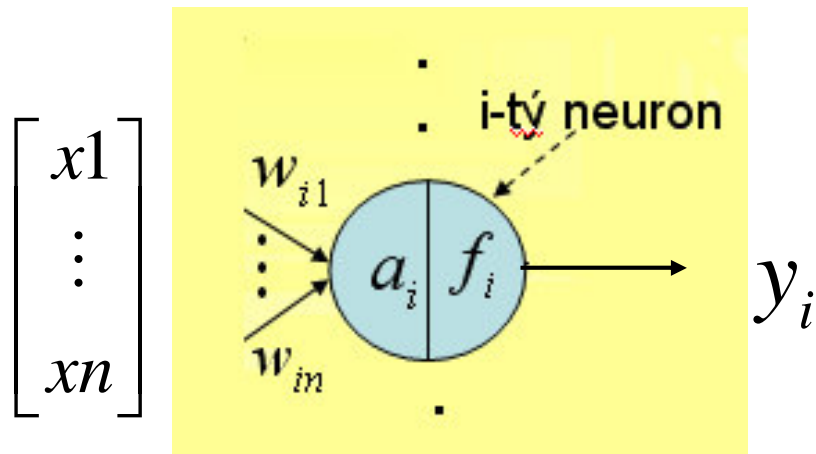
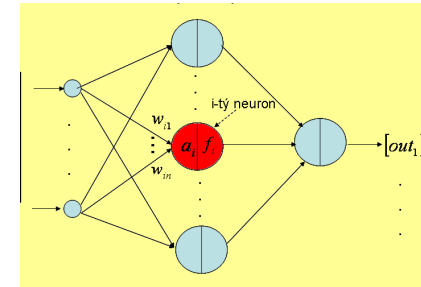
Jak pracuje NN síť pokr.

- NN síť pracuje v režimech:
 - **režim učení**, tj. ladění vah neuronů pro dosažení minimální chyby : a) adaptace
b) trénování
 - **režim provozu**: a) síť je už naučená a může se použít s pevně nastavenými váhami (např. rozpoznávání písma).
b) síť je už naučená, může se používat ale váhy se stále adaptují (např. adaptabilní regulátor)

Neuronové sítě typu:
MULTILAYER FEED-FORWARD NN
(FFNN)

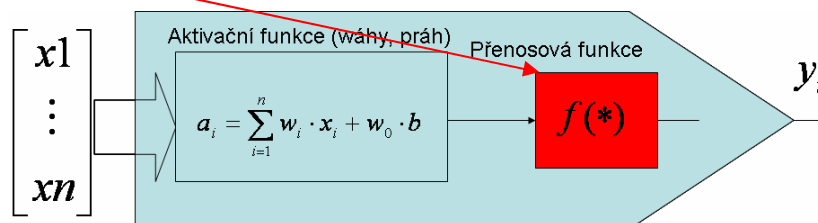


i-tý UMĚLÝ NEURON



Typy FFNN v NN-Toolboxu

- Důležitými hledisky je příhodnost daného typu NN sítě pro danou aplikaci a typ implementované přenosové funkce $f(*)$.



např:

- Síť 1-Vrstvý perceptron: jednovrstvá síť s výstupní funkcí typu hardlim, vhodná pro klasifikační problémy.

Práce v NN-Toolboxu 3.0.1.

- NN síť v NN-Toolboxu se vytváří na příkazovém řádku Matlab commanderu, tj. nejlépe zápisem do *.m souboru a pak jeho spuštěním.
- Síť se z Matlab commanderu musí i učit.
- Naučenou síť lze vygenerovat i jako blok do Simulinku a tam dále používat.

Práce v NN-Toolboxu 3.0.1 pokr.

Vytvoření obecné prázdné struktury NN sítě námi nazvanou „net“:

```
net = NETWORK(...)
```

Příkazy které vytvoří před-definovanou prázdnou strukturu NN sítě námi nazvanou „net“:

```
net = NEWP([min max; ...;min max],n);
```

```
net = NEWLIND(P,T);
```

```
net = NEWLIN([min max; ...;min max],n);
```

```
net = NEWFF ([min max; ...;min max],[ n1 n2],{'tansig','purelin'},'trainlm');
```

Práce v NN-Toolboxu 3.0.1 pokr.

Některé atributy sítě které můžete chtít nastavit:

- `net.layers{i}.initFcn = 'initwb';`
- `net.inputWeights{i,j}.initFcn='rands';`
- `net.layerWeights{i,j}.initFcn='rands';`
- `net.biases{i}.initFcn='rands';`
- `net.trainParam.epochs = 1000;`
- `net.trainParam.goal = .001^2;`
- `net.trainParam.show = 40;`

Práce v NN-Toolboxu 3.0.1 pokr.

- Inicializace sítě:

`net = INIT(net);`

- Trénování sítě:

`[net , e] = TRAIN(net,P,T);` (vrací vektor výstupu a chybu)

- Spuštění provozu sítě v Matlab commanderu:

`y=SIM(net,P);` (vrací vektor výstupu)

- Generování sítě do Simulinku:

`GENSIM(net,dt);`

Příkazy pro učení sítě

Trénování (Batch Training)

Váhy a prahy neuronů se nastavují po proběhnutí celé sekvence všech vektorů vstupů

net = TRAIN(net,P,T,Pi,Ai).

net – Network, P - Network inputs, T - Network targets.

Větší možnosti metod učení

Adaptace (Incremental Training)

Váhy a prahy neuronů se nastavují po každé jednotlivé prezentaci vektoru vstupu

[net,Y,...] = ADAPT(net,P,T,...)

Vhodné pro průběžnou identifikaci, adaptabilní regulátory,...

Pravidlo učení

- Jedny z nejpoužívanějších pravidel vycházejí z modifikací metody Back-Propagation a kde přírůstek váhy je úměrný velikosti chyby a působí v opačném směru gradientu přírůstku “Performance Indexu”

$$\text{např : } \Delta w_i = -\mu \cdot \frac{\partial J}{\partial w_i} = -\mu \cdot 2e \frac{\partial e}{\partial w_i}$$

kde μ ... learning rate

$$J = \int e^2(t) dt \quad \dots \text{Performance Index}$$

$$e = (y_{target} - y_{neuron}) \quad \dots \text{error}$$

Příkaz NEWLIND:Linear System

Design pokr.

```
P = [1 2 3];
```

```
T= [2.0 4.1 5.9];
```

```
net = newlind(P,T);
```

```
Y = sin(net,P)
```

```
Y =
```

```
2.0500    4.0000    5.9500
```

NN sítě s neurony s lineární přenosovou funkcí $f(*)$ jsou vhodné hlavně pro lineární či lineárně separovatelné problémy.

Pro nelineární problémy alespoň minimalizují střední kvadratickou chybu (MSE), ale při učení nekonvergují k přesnému řešení, jinými slovy, síť musí obsahovat dané řešení jako podmnožinu své struktury, aby k ní vůbec mohla konvergovat...

Multilayer Feed Forward NN

- Nejvíce používané sítě
- Přenosové funkce ve skrytých vrstvách, „tansig“ nebo „logsig“
- Přenosové funkce ve výstupní vrstvě jsou nejčastěji lineární:

Příkaz GENSIM

GENSIM(net,Ts) generuje navrženou NN síť
„net“ do Simulinku

