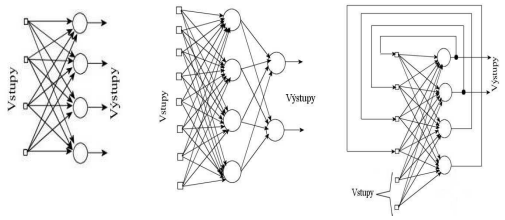
Vícevrstvé sítě I

ML-perceptron

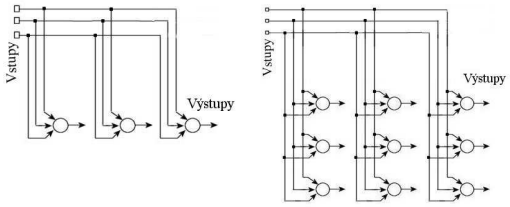
Vícevrstvé sítě - co to znamená?

- Jednovrstvé x vícevrstvé,
- skrytá vrstva,
- dopředné x zpětné šíření,
- rekurzivní (rekurentní) sítě.



Jednovrstvá Dvouvrstvá

Síť se vstřícným šířením signálu Rekurentní síť



Jedno- (1-D) a dvou-dimenzionální (2-D) jednovrstvá síť se vstřícným šířením signálu

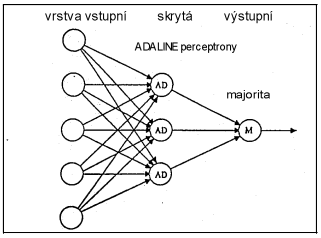
MADALINE (**M**ultiple **Ada**ptive **Lin**ear **E**lement)

- Zpracovává binární signály,
- pracuje s bipolárním kódováním,
- jedna skrytá, jedna výstupní vrstva,
- učí se s učitelem,
- autorem je opět B. Widrow.



Bernard Widrow

MADALINE - architektura sítě



Pozn.: učí se pouze synapse na vstupech ADALINE perceptronů

Aplikační možnosti - nevalné

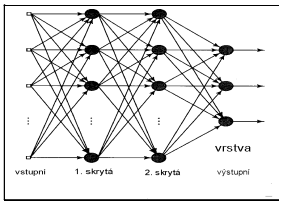
- Přestože je vícevrstvá (dvouvrstvá), neporadí si s lineárně neseparovatelnými vstupními množinami,
- význam pouze metodický.

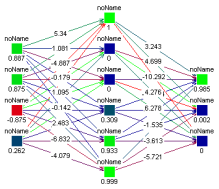
- Inspirace:
- Předpověď počasí, v ní ale zejména kódování vstupu **lineárně nezávislými kódy**.

Vícevrstvé sítě II

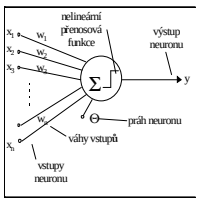
Vícevrstvý perceptron - síť
Back-propagation

Architektura sítě Back-propagation





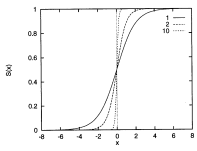
Jak vypadá aktivní prvek BP-sítě?



To je přece známý McCulloch-Pittsův perceptron!

Ano. Zvláštností je jen nelineární funkce S.

Nelineární přenosová funkce BP-neuronu



$$S(\varphi) = \frac{1}{1 + e^{-\gamma\varphi}}$$

Sigmoida zobrazená pro různé koeficienty ϕ

Namísto sigmoidy může být přenosovou (aktivační) funkcí

- hyperbolická tangenta,
- sinus, či
- jiná nelineární funkce splňující jisté (viz dále) požadavky.

Funkci umělé neuronové sítě můžeme chápat jako transformaci *T* vstupního signálu $\vec{X}$

na výstupní signál

$$\vec{Y} = T\left(\vec{X}\right)$$

Úvahy o tom, jaké transformace *T* je možno pomocí neuronových sítí realizovat, jsou předmětem zájmu již od počátku rozvoje jejich poznání.

Minsky-Papertův omyl

- Oba autoři matematicky dokázali, že se tato neuronová síť (rozuměj **perceptron**) nehodí pro implementaci jednoduché logické funkce typu **XOR**.
- Svůj správný poznatek však neoprávněně zobecnili na neuronové sítě vůbec. Jak se dnes s odstupem let často konstatuje, nesou díky tomu vinu za pozdržení vývoje, které nastalo od konce šedesátých až do poloviny let osmdesátých.

Transformační funkce *T* obecně:

Aplikace **Kolmogorovova teorému** na problematiku neuronových sítí vede k poznatku, že k tomu, aby bylo transformační funkcí *T* neuronové sítě možno aproximovat libovolnou funkci *f* postačí, aby příslušná neuronová síť měla alespoň tři vrstvy o odpovídajících počtech neuronů (výkonných prvků) v jednotlivých vrstvách.

23 problémů matematiky

- International Congress of Mathematicians in Paris (1900)
- http://babbage.clarku.edu/~djoyce/hilbert/



David Hilbert

Třináctý Hilbertův problém

Třináctý Hilbertův problém se týká možnosti reprezentace spojitých funkcí *n* proměnných pomocí konečného součtu a superpozice spojitých funkcí jedné proměnné. Hilbert vyslovil domněnku, že kořeny rovnice

$$x^7 + ax^3 + bx^2 + cx + 1 = 0$$

nelze vyjádřit jako funkce koeficientů *a*, *b*, *c* konečnými součty spojitých funkcí pouze dvou proměnných. Teprve po 54 létech dokázal Kolmogorov, že tato domněnka nebyla správná.

Kolmogorovovův teorém (1954)

Každou reálnou spojitou funkci *f*, mající *n* proměnných definovanou v *n*-rozměrné krychli o hranách (0,1) lze vyjádřit jako

$$f(x_1,\ldots,\ x_n) = \sum_{q=1}^{2n+1} \psi_q \left(\sum_{p=1}^{n} \phi_{pq}(x_p)\right)$$

kde ψ_q a ϕ_{pq} jsou spojité funkce jedné proměnné a $p = 1, \ldots, n$ a $q = 1, \ldots, (2n + 1)$.

Přitom pro danou funkci *f* jsou specifické pouze funkce ψ_q, zatímco funkce ϕ_{pq} jsou pro dané *n* a *f* nezávislé.

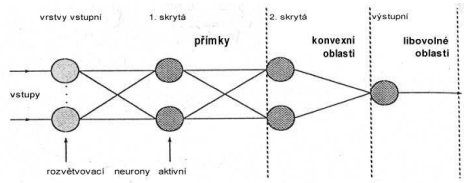
Hecht-Nielsen:

Hecht-Nielsen ukázal, že tuto univerzální vlastnost funkcí ϕ_{pq} lze využít též pro reprezentaci funkcí s hodnotami v prostoru vyšších rozměrů.

Důsledky pro umělé neuronové sítě:

Aplikace Kolmogorovova teorému na problematiku neuronových sítí vede k poznatku, že k tomu, aby bylo transformační funkcí *T* neuronové sítě možno aproximovat libovolnou funkci *f* **postačí**, aby příslušná neuronová síť měla **alespoň tři vrstvy** o odpovídajících počtech neuronů (výkonných prvků) v jednotlivých vrstvách.

Tato důležitá otázka jinak: kolik skrytých vrstev?



Další parametry BP sítě - počet neuronů

Síť s **jednou skrytou vrstvou** (tedy v naší terminologii síť dvouvrstvá):

$$N_{skryt} = \sqrt{N_{vst} . N_{výst}}\ ,$$

Síť se **dvěma skrytými vrstvami**:

$$N_{skryt1} = N_{výst} \left(\sqrt[3]{\frac{N_{vst}}{N_{výst}}} \right)^2, \qquad N_{skryt2} = N_{výst} \sqrt[3]{\left(\frac{N_{vst}}{N_{výst}} \right)}.$$

Poznámky ke klmgrvv. teorému

- R. Hecht-Nielsen, **Kolmogorov's Mapping Neural Network Existence Theorem**, IEEE International Conference on Neural Networks, San Diego, SOS Printing, 1987, Vol. 2, pp. 11--14.
- F. Girosi and T. Poggio Representation Properties of Networks: **Kolmogorov's Theorem is Irrelevant**, *Neural Computation*, 1, 465-469, 1989.
- V. Kurkova, ***Kolmogorov's Theorem Is Relevant***, Neural Computation, 1991, Vol. 3, pp. 617--622


R. Hecht-Nielsen
UCSD Jacobs S.E.


Tomaso A. Poggio
MIT


V. Kůrková
CAS

Algoritmus zpětného šíření

Co že se to zpětně šíří?

Chyba!

Kam se zpětně šíří?

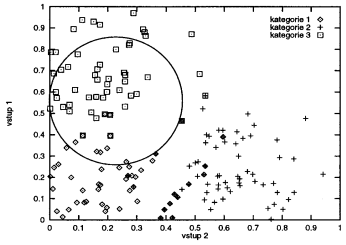
Z výstupu přes skrytou/té vrstvu/y na vstup!

Zpětnému šíření se anglicky říká **Back-propagation**.

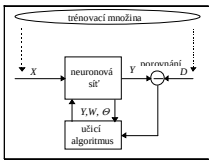
Trénovací a testovací množina

- Množina všech nebo vybraných uspořádaných dvojic vstupní vektor - výstupní vektor. V druhém případě musí **reprezentativním** způsobem popisovat prostor vstupních dat.
- V případě, že je množina vstupních vektorů příliš veliká, vybere se (viz výše) podmnožina a kvalita naučení se kontroluje množinou testovací.

Příklad nevhodně zvolené trénovací množiny



Učení BP-sítě



náhodná_inicializace_vah;
repeat
  repeat
    vyber_vzor_z_trénovací_množiny;
    přilož_vybraný_vzor_na_vstupy_sítě;
    vypočti_výstupy_sítě;
    porovnej_výstupy_s_požadovanými_hodnotami;
    modifikuj_váhy
  until vybrány_všechny_vzory_z_trénovací_množiny;
until globální_chyba < kritérium

Poznámky k průběhu učení

Výběr vzorů z trénovací množiny nemusí být jen sekvenční, často se používá např. náhodný výběr. Volba strategie není pevně dána a spíše patří do kategorie "umění učit" neuronové sítě. V průběhu učení můžeme určité vzory předkládat častěji k učení, jiné zase méně často atp. Časovému intervalu, během něhož síti předložíme každý vzor z trénovací množiny alespoň jednou, říkáme **epocha**. K naučení sítě obvykle potřebujeme stovky až tisíce epoch. Vhodnou strategií můžeme rychlost a úspěšnost učení značně ovlivnit.

Kdy je síť naučená?

Když je splněna předem zadaná podmínka.

Tou může být

- dosažení předepsaného počtu trénovacích epoch,
- pokles **globální chyby** pod předem stanovenou mez,
- atp.

Globální chyba

Globální chybou je v BP algoritmu míněna např. střední kvadratická odchylka (RMS - Root Mean Squared), počítaná přes všechny vzory trénovací množiny:

$$E_{RMS} = \sum_m \sqrt{\sum_n (y_{mn} - d_{mn})^2},$$

kde n je počet neuronů výstupní vrstvy a m mohutnost trénovací množiny.

Poznámka:

Někdy se hodnota střední kvadratické odchylky vyjadřuje vztažená na jeden vzor. Dělí se tedy mohutností (počtem členů) trénovací množiny. Na vlastní význam chyby nemá tato změna (multiplikativní člen) žádný významný vliv.

Energie sítě

Algoritmus zpětného šíření je založen na minimalizaci energie neuronové sítě. Energie je mírou naučenosti, tedy odchylkou mezi skutečnými a požadovanými hodnotami výstupů neuronové sítě pro danou trénovací množinu.

Pro neuronovou síť typu Back-propagation se energetická funkce definuje takto:

$$E = \frac{1}{2} \sum_{i=1}^{n} (y_i - d_i)^2$$

kde n je počet výstupů sítě, y_i je i-tý výstup a d_i je i-tý požadovaný výstup.

