# Umělé neuronové sítě

L. Kocman[1], L. Kupilík[2], P. Mikeš[3], M. Rejman[4], B. Rosička[5]
[1]Gymnazium Bučovice,
[2]VOŠ a SPŠE Plzeň,
[3]SOŠ Blatná,
[4]Gymnazium, Jablonec nad Nisou,
[5]Gymnazium Broumov,
martin.rejman@centrum.cz

**Abstrakt:**

Umělé neuronové sítě jsou populární, a proto si zaslouží pozornost. Zde jsou ukázány různé metody strojového učení s důrazem na umělé neuronové sítě, jejich výhody a podrobně ro příklad aplikace na funkci XOR.

# 1 Úvod

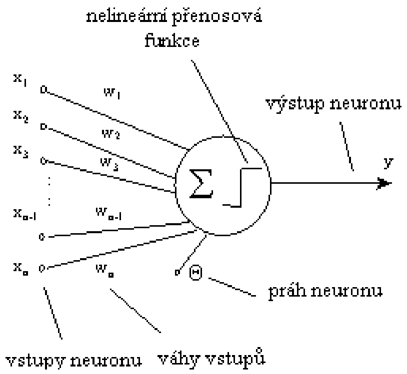
Miniprojekt si kladl za úkol seznámit účastníky s problematikou neuronových sítí jako součásti strojového učení. Silnou motivací pro studium umělých neuronových sítí je výkon biologických neuronových sítí např. u člověka, jež posloužily jako příklad a paralelizace výpočtu.

# 2 Neuronové sítě jako součást strojového učení

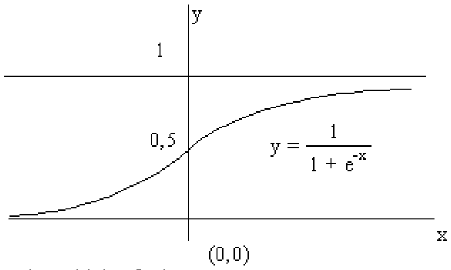
Neuronové sítě jsou jednou z metod učení strojů. Základním problémem strojového učení (tedy i neuronových sítí) je **klasifikace** (rozpoznávání) prvků do **tříd** podle známých údajů, **příznaků**, a na trénovací množině, skupině dat se známou klasifikací, naučit počítač klasifikovat předkládané prvky. Postupů, jak to lze udělat, je několik:

- Statistické metody - využívají pravděpodobnost : NN (nearest neighbour – nejbližší soused : prvek určí stejně, jako nejbližší sousedící prvek), *k*-NN (příslušnost objektu určí podle převahy typů u *k* nejbližších sousedů) a Bayesův klasifikátor (odhaduje pravděpodobnost, že prvek patří k určité třídě).
- Genetické algoritmy – na základě vývoje objektů a vnášení mutací postupně v generacích posiluje žádoucí znaky či vlastnosti
- Rozhodovací stromy a lesy – podle příznaků postupně určuje klasifikaci, pro větší věrohodnost výsledků se užívá kombinace rozhodovacích stromů, tj. *rozhodovací lesy*
- Neuronové sítě – viz. dále

Volba postupu závisí na řešiteli, určitá metodika má však i své “speciality”, problémy, které umí řešit lépe než ostatní.

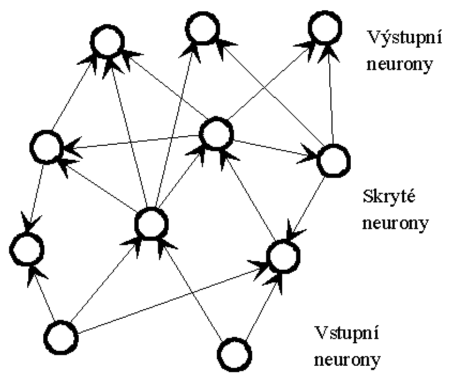



Aby bylo možné problém zjednodušit, je třeba vytvořit model biologického neuronu. Ten existuje již od roku 1958, nazývá se **perceptron** a užívá se s obměnami dodnes. Tato “buňka“ má více vstupů $x_i$ a jeden výstup y, jenž může být předán více neuronům jako vstup. Význam vstupu je určen synaptickou vahou vstupu (spoje) $w_i$. Práh neuronu Θ určuje, při jaké hodnotě bude neuron aktivován. Výstupní hodnota neuronu je určena funkcí $y = \Phi\left(\sum_{i=1}^{n} x_i w_i - \Theta\right)$, tedy součtem vstupů vynásobených vahami a zmenšeným o práh, a na tento výsledek se aplikuje libovolná aktivační (přenosová funkce) Φ[fí], např. čistě binární funkce (x < 0, pak f(x) = 0; x ≥ 0, pak f(x) = 1), nebo sigmoidální funkce (viz. obr., příznakový prostor dělí přímkami) a RBF (Radial Basis Function) funkce (ty dělí prostor elipsoidy, např. $y = e^{-\frac{1}{2}x^2}$ ). Výstup je buď konečným výsledkem, nebo je předán dalším neuronům.



sigmoidální funkce

**Neuronová síť** je struktura složená z neuronů spojených orientovanými spoji (vedou od někud někam). **Právě hrany (=spoje) určují schopnost učit se a spolu s neurony skladují nabyté dovednosti**. Síť je tedy určena množinou neuronů a množinou orientovaných hran. Obecné (tj. nepravidelně, náhodně propojená, viz. obr.) síťě ještě nejsou dostatečně prozkoumány kvůli značné složitosti dějů a paralelizaci výpočtu. Učení probíhá na základě trénovací množiny Proto jsme se dále zabývali **vrstevnatými** sítěmi.



Obecná neuronová síť

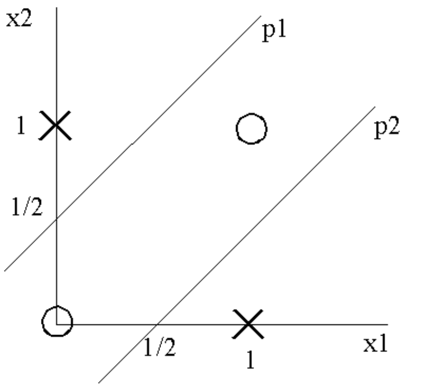
Ty se vyznačují zjednodušením : neurony jsou uspořádány do vrstev a spoje vedou pouze do vyšších vrstev, tj. od vstupní směrem k výstupní, což umožnilo systematickou minimalizaci chybové funkce sítě (její **učení**), která indikuje naučenost komplexu. Jednou z postupů je tzv. back-propagation (zpětné šíření chyby sítě, tj. od výstupu ke vstupu). Chyba je rozdíl očekávaného výsledku (známý z trénovací množiny) a zjištěného výstupu. Celková chyba sítě se vypočítá $E = \frac{1}{2}\sum_{i=1}^{n}\left\|F(X_i) - y_i\right\|$, kde funkce *F* je funkce realizovaná neuronovou sítí, $X_i$ trénovací vzor a $y_i$ výstup pro daný vzor. Problémem tohoto postupu je, že najdeme jen lokální minimum chybové funkce, tj. nenajdeme nejlepší řešení.

**Příkladem užití** neuronové sítě je realizace logické funkce XOR (nonekvivalence, viz. vpravo). Chceme rozdělit příznakový prostor tak, abychom oddělili žádoucí (křížky) a nežádoucí prvky (kolečka, obr.1). Problém je v tom, že příznakový prostor XORu není lineárně separabilní, a proto nelze řešení realizovat jedním perceptorem. Potřebujeme dva perceptrony, dvě přímky (každý perceptron znamená jednu přímku).O hodnotě výstupu perceptronu rozhoduje přímka $0 = x1.w1 + x2.w2 - h$. Je-li daný “bod” nad ní, dostaneme 1, jinak 0. Získáme dělicí přímky (zvolíme, jak přímkami rozdělit daný prostor) **p1:** $-x1 + x2 - 1/2 = 0$ a **p2:** $-x1 + x2 + 1/2 = 0$. Nyní spočteme výstup (tj. levou stranu rovnic přímek) těchto perceptronů dosazením učebních vzorů na vstup (x1 a x2 z tab.1). Je-li větší nebo roven nule, výstup je 1, jinak 0. Výstupy z prvních dvou perceptronů (p1,p2) přecházejí na vstup další vrstvy, jejíž příznakový prostor je na obr. 2. Problém je vyřešen, jelikož tento prostor lze již rozdělit jedinou přímkou, užijeme jeden preceptron p3. Zvolíme dělicí přímkou, zapíšeme rovnici : p3: $-x1 + x2 - 1/2 = 0$. Vstupem pro tuto rovnici (a tento perceptron) je výstup první vrstvy. Který označit křížkem a který kolečkem? Kolečka jsou ty výstupy první vrstvy, které vznikly ze vzoru první vrstvy a u nichž je výsledkem funkce XOR 0, tedy výstup (0,1). Dosadíme tedy (0,1) a zjistíme, jestli dostaneme 1 nebo 0. Výsledkem dosazení je kladné hodnota, tedy výstup jedna, ale zkoumaný bod tam nepatří, funkci tedy obrátíme, tj. vynásobíme mínus jedničkou. Známe tedy rovnice, nyní je zapíšeme do neuronové sítě. Pro přepis do neuronové sítě vezmeme koeficienty z rovnic: pro p1: váha z x1 = -1, z x2 =1, práh – 1/2; pro p2 : váha z x1 = -1, z x2 = 1, práh 1/2. Obdobně pro p3. Na obrázku vpravo je znázorněno řešení, u šipek jsou váhy, v kostičkách prahy.

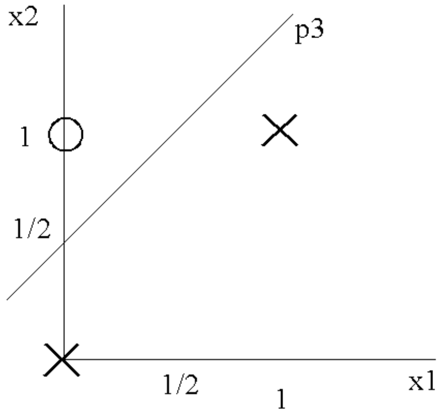
| x1 | x2 | x1 XOR x2 |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

Tab.1.: XOR

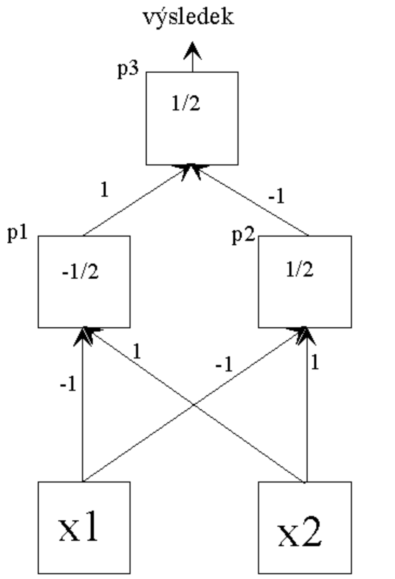
| vzor | | výstup | |
|---|---|---|---|
| $x_1$ | $x_2$ | $p_1$ | $p_2$ |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 |



Obr. 1: Příznakový prostor funkce XOR



Obr. 2: Příznakový prostor 2.vrstvy



Řešení problému XOR

# 3 Shrnutí

Neuronové sítě jsou bezesporu zajímavým řešením problémů, zejména proto, že nemusíme znát alogoritmus pro řešení problému, je tedy jakousi černou skříňkou. Nevýhodou tohoto postupu je, že je obtížné najít správné propojení neuronů a pomalé učení sítě. Při dobrém naučení však poskytuje přesnější výsledky než konkurenční metody. Neprozkoumané, složitější sítě jsou výzvou do budoucna.

# Poděkování

Poděkování fakultě jaderné a fyzikálně inženýrské ČVUT a supervizorovi ing. Emilu Kotrčovi z katedry matematiky.

# Reference:

[1] Hakl, Fr. – Holeňa, M. : Úvod do teorie neuronových sítí, Ediční středisko ČVUT, Praha 1997. (www.cs.cas.cz/~hakl)