# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

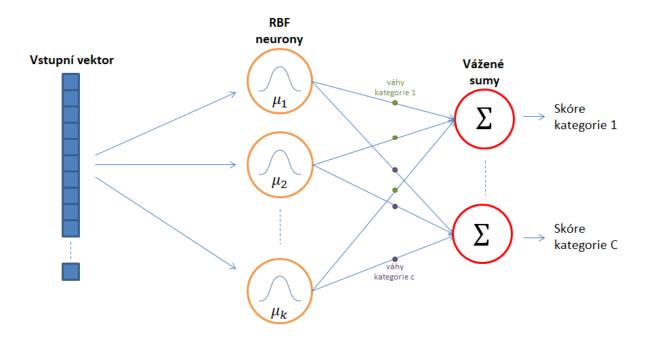
SFC – Soft Computing

Demonstrace učení sítě RBF

## 1 RBF (Radial Basis Function) neural network

Jedním z typů sítí, které zle využívat pro klasifikaci dat, jsou RBF sítě. Jedná se o typicky dvouvrstvé neuronové sítě, kde ve skryté vrstvě jsou obsaženy neurony s radiální bázovou funkcí a ve výstupní vrstvě se nachází neurony s lineární bázovou funkcí a lineární aktivační funkcí. [1]

### 1.1 Schéma propojení RBF sítě



Jak je ze schématu zřejmé, každý neuron skryté vrstvy je propojen se všemi neurony výstupní vrstvy a naopak, nejsou však propojeny v rámci stejné vrstvy a vstupní vektory dat se přikládají zásadně na vstupy skryté vrstvy, nikoliv výstupní vrstvy.

Jak je ve schématu také naznačeno, neurony v tomto typu neuronové sítě typicky nevyužívají "předpětí" (bias). [2] Výsledkem je vítězný neuron s nejvyšším skóre, který určuje třídu pro vektor vstupních dat.

## 2 Konstrukce a učení RBF sítě

Sít se nejprve zkonstruuje za pomoci trénovacích dat a s již pevně danou strukturou teprve následně probíhá učení. Učení skryté vrstvy je součástí konstrukce RBF sítě, po jejím učení proběhne odhad parametru σ pro každý neuron skryté vrstvy. Učení sítě probíhá po jejím zkonstruování opakovaným přikládáním vektorů dat na vstup sítě a následnou úpravou vah mezi skrytou a výstupní vrstvou.

## 2.1 Konstrukce a učení skryté vrstvy RBF sítě pomocí K-means

- 1. Inicializace k vektorů  $\vec{w}_k$  reprezentující shluky  $C_k$  náhodnými vektory z trénovací množiny.
- 2. Nastavení indexu p na první prvek trénovací množiny (p = 1).
- 3. Přiřazení vektoru  $\vec{l}_p$  do shluku  $C_{k^*}$ , od nějž má vektor  $\vec{w}_k$  nejmenší vzdálenost:

$$\|\vec{l}_{p} - \vec{w}_{k*}\| \le \|\vec{l}_{p} - \vec{w}_{k}\|, \quad k = 1..q$$

- 4. Inkrementace indexu p, pokud je p větší, než je počet vektorů trénovací množiny P, tak pokračujte, jinak se vraťte na bod 3.
- 5. Přepočítání hodnoty vektorů  $\vec{w}_k$  tak, aby reprezentovaly těžiště jednotlivých shluků:

$${}^{1}\vec{w}_{k} = \frac{1}{n_{C_{k}}} \sum_{i_{l} \in C_{k}} \vec{i}_{l}$$

- 6. Pokud došlo ke změně přiřazení nějakého vektoru (z jednoho shluku do jiného), vraťte se na krok 2, jinak pokračujte.
- 7. Výpočet parametru  $\sigma$  pro každý shluk  $C_k$ :

$$^{1}\sigma_{k} = \sqrt{\frac{1}{n_{C_{k}} - 1} \sum_{i_{l} \in C_{k}} \|\vec{i}_{p} - {}^{1}\vec{w}_{k}\|^{2}}, \quad k = 1..q$$

## 2.2 Konstrukce a učení výstupní vrstvy RBF sítě

- 1. Inicializace j neuronů výstupní třídy reprezentující j tříd a jejich propojení s k neurony skryté vrstvy.
- 2. Nastavení indexu p na první prvek trénovací množiny (p = 1).
- 3. Výpočet vnitřního potenciálu neuronů skryté vrstvy pro vektor  $\vec{l}_p$ .

$$^{1}u_{k} = \|\vec{i}_{p} - {}^{1}\vec{w}_{k}\| = \sqrt{\sum_{i=1}^{n} (i_{pi} - {}^{1}w_{ki})^{2}}$$

4. Výpočet výstupů neuronů skryté vrstvy pro vektor  $\vec{l}_n$ .

$${}^{1}y_{k} = e^{-\left({}^{1}u_{k}/_{1}\sigma_{k}\right)^{2}}$$

5. Výpočet výstupů výstupní vrstvy.

$$^{2}y_{j} = ^{2}u_{j} = \sum_{k=1}^{q} ^{2}w_{jk} ^{1}y_{k}$$

6. Modifikace vah výstupní vrstvy.

$$\Delta(^2w_{jk}) = \mu_2(d_j - ^2y_j)^1y_k$$

7. Pokud bylo dosaženo chtěného výsledku, algoritmus končí, jinak inkrementujte p o 1, pokud je p větší než P, nastavte p na 1, vraťte se na krok 2.

## 3 Implementace

Tato aplikace slouží pro demonstraci učení RBF sítě pomocí algoritmů v kapitole 2, z toho důvod byla navržena tak, aby byly výpočty snadno reprodukovatelné a uživatel tak mohl porovnávat výsledky učení pro vstupní data při různých parametrech. Pro tuto reprodukovatelnost tak bylo nutné zrušit náhodnost, která se vyskytuje například při inicializaci neuronů skryté vrstvy. Náhodnost však lze opět vynutit pomocí přepínače --rand.

Vzorce výše dále vedou v případě, že je ve shluku neuronu skryté vrstvy pouze jeden vektor dat k situaci, kdy bude nulová sigma, čili i výstup tohoto neuronu bude vždy nulový, tedy jako by v síti vůbec nebyl. Zkoušel jsem proto implementovat alternativní algoritmus, který by tento defekt neměl (Např. na základě rozptylu ostatních neuronů skryté vrstvy od konkrétního s myšlenkou, že čím vzdálenější jsou od neuronu q ostatní neurony, tím větší váhu by neuron q měl mít.), nicméně učení probíhalo celkově spíše hůře, a tak jsem od tohoto upustil a ponechal pouze základní variantu výpočtu parametru sigma.

### 3.1 Použití aplikace

Aplikace byla napsána v jazyce C++ ve standardu C++17. Lze ji přeložit pomocí příkazu *make* a několik ukázkových spuštění and přiloženými daty lze provést pomocí make *run*[1-4]. Program po jednotlivých krocích čeká na stisk klávesy enter pro pokračování a vypíná se pomocí *CTRL+C*.

V základní konfiguraci při spuštění pouze jako ./rbf je použit vstupní soubor input, u nějž a je předpokládán formát pro síť, kde vektor vstupních dat má dvě položky a data mají být klasifikována do tří tříd. K tomu bude využito 6 neuronů skryté vrstvy, koeficient učení výstupní vrstvy bude 0,5.

#### 3.1.1 Přepínače aplikace

random/-r	Randomizuje počáteční váhy neuronů, pro demonstraci jsou jinak využity údaje prvních vektorů dat.
	Počáteční váhy neuronů výstupní vrstvy budou pro demonstraci, že na nich příliš nezáleží náhodné v intervalu <-1;1>, jinak budou nulové.
class/-c X	Nastaví očekávaný počet tříd, a tedy počet neuronů výstupní vrstvy, implicitně 3.
dimension/-d X	Nastaví očekávanou dimenzi vstupních dat, implicitně 2.
file/-f FILENAME	Nastaví soubor, ze kterého se budou brát vektory vstupních dat, implicitně input.
mi/-m X	Nastaví koeficient učení výstupní vrstvy, implicitně 0,5.
neuron/-n X	Nastaví počet neuronů skryté vrstvy, implicitně 6.

#### 3.1.2 Spuštění aplikace

```
$ make run[1-4]
$ ./rbf [--random] [--class int] [--dimension int] [--file string] [--mi float] [--neuron int]
$ ./rbf [-r] [-c int] [-d int] [-f string] [-m float] [-n int]
```

## 4 Reference

Následující zdroje jsou primární zdroje, ze kterých jsem vycházel při snaze pochopit, jak má RBF síť principiálně fungovat, jak má být zkonstruována a jak probíhá učení.

- [1] F. Zbořil, "Privátní stránky předmětu SFC," [Online]. Available: https://www.fit.vutbr.cz/study/courses/SFC/private/20sfc\_5.pdf. [Přístup získán Listopad 2021].
- [2] D. Kriesel, "dkriesel," [Online]. Available: https://www.dkriesel.com/\_media/science/neuronalenetze-en-zeta2-2col-dkrieselcom.pdf. [Přístup získán Listopad 2021].
- [3] P. Kudova. [Online]. Available: http://www.cs.cas.cz/~petra/diplomka/text/node3.html. [Přístup získán Listopad 2021].