Technická dokumentácia NN

# Požiadavky

Minimálne požiadavky pre správny beh programu:

* Nainštalovaný programovací jazyk C++
* Stiahnutá a nainštalovaná knižnica FANN-2.2.0-Source
* Nainštalovaný Operačný systém
* Dostatok voľnej pamäte RAM

# Inštalácia

FANN knižnicu je potrebné nainštalovať a nastaviť pre prácu vo MS Visual Studio.

Postup inštalácie:

1. Rozbaliť knižnicu **FANN-2.2.0-Source** do MS Visual projektu:

C:\Documents\Visual Studio 2013\Projects\projekt\projekt\

1. V projekte vyhľadať rozbalenú knižnicu \Projects\projekt\projekt\**FANN\VS2010** a spustíme **fann.sln**(MVS solution file) vo MS Visual Studio
2. Vo Visual Studio **BUILD->Batch Build… , Select all** a stlačiť **Rebuild**.

Postup nastavenia projektu pre prácu s FANN:

1. Otvor projekt vo Visual Studio a **Project -> Properties**
2. **C/C++** -> Additional Include Directories -> **\FANN\src\include**
3. **Linker** -> General -> Additional Library Directories -> **\FANN\bin**
4. **Linker** -> Input -> Additional Dependencies -> **fanndouble.lib** ALEBO **fannfloat.lib**
5. Skopíruj všetky **.dll** súbory z **\FANN\bin** do projektu
6. Pridať knižnice do kódu:

#include "fann.h"

#include "floatfann.h"

# Premenné (global)

Tieto premenné boli zadefinované ako globálne z dôvodu rýchlejšieho a efektívnejšieho použitia. V aplikácií sa takto tieto parametre nastavujú oveľa jednoduchšie a uľahčí aj prípadné vyhľadávanie v kóde. Sú to v podstate základné parametre pre trénovanie neurónovej siete.

unsigned int num\_input; - počet vstupov

unsigned int num\_output; - počet výstupov

unsigned int num\_layers; - počtu vrstiev

unsigned int num\_neurons\_hidden1; - počet neurónov v skrytej vrstve

unsigned int max\_epochs; - maximálny počet epoch

unsigned int epochs\_between\_reports; - epochy medzi reportami

float desired\_error; - definovanie chyby

struct fann \*ann; - štruktúra neurónovej siete

# Funkcie

void setFunctions(){} – funkcia pre nastavenie rôznych funkcií pre sieť

1. fann\_set\_activation\_function\_hidden(struct fann \*ann, FANN\_LINEAR);
2. fann\_set\_activation\_function\_output(struct fann \*ann, FANN\_SIGMOID);
3. fann\_set\_train\_stop\_function(struct fann \*ann, FANN\_STOPFUNC\_MSE);
4. fann\_set\_training\_algorithm(struct fann \*ann, FANN\_TRAIN\_SARPROP);
5. fann\_set\_bit\_fail\_limit(struct fann \*ann, 0.035);
6. fann\_set\_learning\_rate(struct fann \*ann, 0.7f);
7. nastaví aktivačnú funkciu pre skrytú vrstvu siete ann
8. nastaví aktivačnú funkciu pre výstupnú vrstvu siete ann
9. nastaví zastavovaciu funkciu pre siet ann
10. nastaví algoritmus trénovania pre sieť ann
11. nastaví bit fail limit pre sieť ann
12. nastaví learning rate pre sieť ann

void setNetworkParameters(){} – funkcia pre nastavenie parametrov siete.

Nastaví hodnoty pre vopred definované premenné ako je

počet vstupov, výstupov atď.

void trainNetwork(){} – volanie funkcií pre nastavenie siete a

vytvorenie/natrénovanie siete.

* setNetworkParameters();
  + popis vyššie, nastaví parametre
* struct fann \*ann = fann\_create\_standard(num\_layers, num\_input,num\_neurons\_hidden1, num\_output);
  + vytvorí nadefinovanú sieť s potrebnými parametrami:
    - Počet vstupov
    - Počet výstupov
    - Počet vrstiev
    - Počet neurónov
* setFunctions();
  + popis vyššie, nastaví funkcie pre sieť
* fann\_train\_on\_file(ann, "Fann\_1.train", max\_epochs,epochs\_between\_reports, desired\_error);
  + natrénovanie siete s nastavenými parametrami:
    - štruktúra fann – sieť
    - názov siete ako sa má uložiť
    - maximálny počet epoch
    - epochy medzi reportami
    - min chyba siete
* fann\_save(ann, "Fann\_1.net");
  + funkcia na uloženie natrénovanej siete

void testNetwork(){} – funkcia pre testovanie natrénovanej siete

int main(){} – hlavná funkcia pre prácu s natrénovanou sieťou, testovanie a ukladanie výsledkov

* struct fann \*test\_ann = fann\_create\_from\_file("Fann\_1.net");
  + zo súboru (bol predtým natrénovaný) načíta a vytvorí sieť na prácu
* fann\_type \*output = fann\_run(struct fann \*test\_ann, control);
  + spustenie siete (beh ANN), beh siete na hľadanie markantov
* fann\_destroy(test\_ann);
  + zničí/zatvorí sieť