

Zaawansowane programowanie w C++ (ZPR)

Dokumentacja wstępna

Projekt pod kierownictwem dr inż. Rafała Biedrzyckiego:

***„Zaimplementować algorytm indukcji klasyfikatora
bazujący na sieci neuronowej”***

1. Ogólna koncepcja funkcjonalności programu

Program bazując na sieci neuronowej będzie realizował rozpoznawanie dźwięków z kilku kategorii na podstawie ich widma, tworzonego za pomocą szybkiej transformaty Fouriera (FFT).

Próbki dla kilku kategorii z zestawów uczących i weryfikujących dostarczane będą jako posegregowane zbiory plików WAVE (*.wav).

Interfejs użytkownika będzie miał postać linii poleceń. Użytkownik będzie miał możliwość wskazać plik z dźwiękiem do rozpoznania. Wyniki podawane będą jako procentowe dopasowanie próbki do każdego z zestawu wyuczonych dźwięków.

2. Wybrany zestaw uczący

Docelowo sieć będzie uczona rozpoznawania jednego z 6 dźwięków pustych strun gitary strojonej do dźwięku Eb (Eb, Ab, Db, Gb, Bb, eb). W tym celu zostanie przygotowany szereg nagrań w postaci plików WAVE każdego z tych dźwięków.

Poziom abstrakcji programu powinien pozwalać na rozpoznawanie również innego zestawu dźwięków, w tym innej ilości kategorii, łatwo rozróżnialnych na podstawie ich widma. Jednakże taka funkcjonalność nie będzie weryfikowana w zakresie projektu (nie zostaną przygotowane dodatkowe zbiory uczące/testujące).

3. Moduły aplikacji

- Moduł do wczytywania dźwięków z nieskompresowanych plików WAVE, zawierający prostą funkcjonalność dostosowującą dane w dowolnym formacie (częstotliwość próbkowania, liczba bitów na próbkę, stereo/mono) do wymaganego formatu. Interfejs abstrakcyjny umożliwiający dołączenie modułu wczytującego dane także z innych źródeł, np. mikrofonu.
- Moduł wykonujący FFT.
- Moduł prostej 3-warstwowej, jednokierunkowej sieci neuronowej, uczonej za pomocą propagacji wstecznej błędu (zastrzegamy możliwość zmiany szczegółowego typu sieci w związku z uszczegółowieniem projektu aplikacji).
- Moduł interfejsu użytkownika, monitorujący przebieg uczenia się i prezentujący weryfikację próbek, z podawaniem wyników.

4. Cechy aplikacji wraz z proponowaną listą narzędzi i bibliotek

- Program w całości napisany obiektowo w C++
- Poprawna kompilacja i praca programu w systemie Windows (Visual Studio 2010) i Linux (Eclipse/g++)
- SCons do automatyzacji kompilacji wieloplatformowej
- Doxygen do stworzenia dokumentacji technicznej, na podstawie komentarzy w kodzie w odpowiednim formacie

- Wykorzystanie biblioteki boost, m.in. testy jednostkowe (boost::test_unit), zarządzanie pamięcią (sprytne wskaźniki), kontenery
- FFTW - multiplatformowa biblioteka open-source do FFT (www.fftw.org)
- Repozytorium kodu w oparciu o system kontroli wersji GIT