Wymagania dotyczące projektu z Sieci Neuronowych

March 13, 2013

1 I etap

Symulator sieci neuronowych ma spełniać następujące wymagania:

- Sieć neuronowa o dowolnej liczbie warstw i liczbie neuronów w warstwie (ustawia użytkownik).
- Typ neuronu do wyboru przez użytkownika z funkcjami aktywacji: liniową, progową, sigmoidalna (suma ważona sygnału jako potencjał synaptyczny).
- Możliwość wczytywania wag z pliku.
- Możliwość testowania sieci (wprowadzamy sygnał wejściowy i otrzymujemy wektor odpowiedzi).

Na tak przygotowanej sieci pokazujemy działanie sieci dla funkcji AND:

1. funkcja skokowa: wagi losowe, wagi nauczone w czasie ćwiczeń,
2. funkcja sigmoidalna:(1/1+e(-x)): wagi losowe, wagi nauczone w czasie ćwiczeń.

To samo dla XOR (warstwa ukryta i wyjściowa - funkcje sigmoidalne) wagi losowe, wagi nauczone w czasie ćwiczeń.

2 II etap

Symulator należy rozbudować o następującą funkcjonalność:

- sąsiedztwo (1D i 2D),
- algorytm uczenia Kohonena (z rywalizacją i sąsiedztwem),
- zmniejszanie parametrów uczenia w trakcie uczenia,
- inicjacja wag wartościami zerowymi lub przypadkowymi z zadanego przedziału.

3 III etap

Ten etap projektu polega na implementacji sieci typu Counter Propagation do symulatora sieci neuronowych. Proszę zbudować sieć typu CounterPropagation będącą w stanie rozwiązać problem parzystości trzech bitów (na trzech wejściach mogą pojawiać się zera i jedynki - na wyjściu powinno pojawić się 1 gdy ilość jedynek na wejściu jest nieparzysta, a 0 gdy jest parzysta). Sieć należy przetestować stworzywszy odpowiedni ciąg uczący (wszystkie przypadki). Należy pamietać o zmianie parametrów sąsiedztwa i szybkości uczenia Alfa.

Problemy do analizy:

- Proszę wyjaśnić w jaki sposób sieć CounterPropagation rozwiązuje problem parzystości. Przy jakiej najmniejszej liczbie neuronów w warstwie Kohonena taka sieć jest w stanie rozwiązać problem parzystości?
- Spróbuj wykasować ze zbioru uczącego dwa elementy np. 010 i 111. Stwórz z nich zbiór testowy. Jeszcze raz naucz sieć i przetestuj najpierw zbiorem uczącym, a potem testowym. Czy sieć generalizuje problem? Dlaczego? Przygotuj się na zaprezentowanie działania swojego symulatora z problemem parzystości.

4 IV etap

W swoim symulatorze zaimplementujcie algorytm wstecznej propagacji błędów (wzory na stronie internetowej (http://home.agh.edu.pl/~asior/stud/stud_pl_10.html sieci bp). Sprawdźcie czy w swoim programie otrzymaliście podobne wyniki uczenia dla wybranych ikonek. W swoim raporcie umieście test programu dla takich ikonek, których sieć bez elementu bias nie potrafi się nauczyć. Pokażcie błąd uczenia i wagi. Podłączcie biasy do neuronów warstwy wyjściowej i nauczcie sieć, pokażcie wagi neuronów wraz z wartością wag biasów. Przetestujcie swój program problemem XOR (dwa neurony wejściowe i ukryte i jeden neuron na wyjściu). Warstwa wejściowa liniowa, w pozostałych funkcje sigmoidalne. Przeanalizujcie wpływ parametrów uczenia na przebieg procesu (błąd końcowy i liczbę epok potrzebnych do osiągnięcia akceptowalnego błędu). Proszę powtarzać próby uczenia przy tych samych współczynnikach dla różnych wag początkowych, a próby z różnymi współczynnikami uczenia powtarzać dla tych samych wag początkowych. Zbudujcie sieć dla problemu parzystości trzech bitów. W warstwie ukrytej wstępnie ustawcie 20 neuronów (parametry uczenia: 0.2, 0.5). Czy udało się nauczyć sieć? Przy ewentualnej porażce zróbcie eksperymenty: przy ilu neuronach w warstwie ukrytej (dwóch warstwach ukrytych) i przy dobrze ustawionych parametrach uczenia, sieć nauczyła się problemu na pamięć? Przy sukcesie próbujcie redukować liczbę neuronów. Skomentujcie wyniki .