Opis projektu “backprop-mnist”

Damian Trojnar II UWr

Jest to implementacja sieci neuronowej mającej za zadanie nauczenie się rozpoznawania 28x28 pikselowych obrazków cyfr.

Dane do nauki sieci są pobierane z bazy MNIST(http://yann.lecun.com/exdb/mnist/), która zawiera pliki:

- train-images-idx3-ubyte (60k obrazków)
- train-labels-idx1-ubyte (60k cyfr do tych obrazków)
- są też analogiczne dane to testowania sieci(10k obrazków)

Format tych plików jest podany w komentarzach(w pliku Mnist.hs) nad funkcjami wczytującymi.

Jest to bardzo popularny zbiór danych, dla którego najlepsze mają błąd < 2%.
Moja sieć ma błąd około 9% przy maksymalnej ilości danych treningowych.
Pliki zawierają czarne litery na białym tle w kodowaniu 8 bitowym.

Sieć zbudowana jest z 3 warstw jednej o wymiarze 28*28, czyli każdy znormalizowany piksel traktujemy jako wejscie, potem mamy jedną warstwę ukrytą o rozmiarze 20 oraz warstwe wyjściową o rozmiarze 10.

Sieć trenowana jest z wykorzystaniem algorytmu propagacji wstecznej(ang. backpropagate). Jest to przykład nauki nadzorowanej(ang. supervised learning).
W skrócie chodzi o to, że sieć traktujemy jako funkcje o dziedzinie rozmiaru 28*28 oraz przeciwdziedzinie 10, w trakcie nauki sprawdzamy czy najbardziej prawdopodobny według naszej sieci jest oczekiwany wynik, jeśli nie to modyfikujemy wagi w zależności od tego jaki był ich wpływ na wynik, tak by następnym razem uzyskać lepsze przybliżenie naszej idealnej funkcji.
Po więcej informacji odsyłam do http://neuralnetworksanddeeplearning.com/chap2.html

Program podzielony jest na 7 plików:
Main.hs - n/c
Mnist.hs - odczyty danych binarnych MNIST
Macierze.hs - brakujące metody dla macierzy oraz definicje WektoraKolumnowego i metod pomocniczych
SiecNeuronowa.hs - definicja klasy typu SiecNeuronowa
PropagacjaWsteczna.hs - implementacja algorytmu propagacji wstecznej na macierzach
Cwicz.hs - funkcje uczące sieć oraz wczytyjące dane z folderu mnist
Obrazki.hs - funkcje do wczytywania jpg

Uruchomienie

Najpierw trzeba zainstalować następujące paczki:
sudo apt-get install haskell-platform # haskell i cabal
sudo apt-get install libghc-hmatrix-dev # operacja na macierzach z fortrana
sudo apt-get install libgd-dev # do grafiki

Aby skompilować projekt wystarczy wpisać "cabal install", ewentualnie "cabal install --force-reinstalls". Domyślnie plik wykonywalny pojawi się nam w “/home/USER/.cabal/bin”

Należy uważać, gdyż program zużywa dużo pamięci RAM(10K obrazków - nawet 1GB RAM). 10 tysięcy jest wystarczające do “poprawnego” rozpoznawania cyfr.

Używanie

Program na początku prosi nas o podanie ilości danych na jakich chcemy testować. Potem jeszcze możemy stestować wydajność naszej sieci, po czym program prosi nas o podanie relatywnej ścieżki do jakiegoś pliku .jpg, który zawiera cyfrę. Program normalizuje te obrazki do odcienia szarości i rozmiaru 28*28 i próbuje odgadnąć co za cyfra jest na obrazku.