

Laboratorium Rozpoznawania Obrazów – Ćwiczenie #5 & #6

Rozpoznawanie cyfr z wykorzystaniem sieci neuronowych

Termin oddawania: **18.05.2015, 21.05.2015**

W kolejnym ćwiczeniu pozostaniemy przy rozpoznawaniu cyfr, zmieniając jedynie typ klasyfikatora używanego do tego zadania. Ponieważ wszystkie informacje o zbiorze danych zostały podane w materiałach poprzedniego ćwiczenia, nie będę ich tutaj powtarzał.

Rozwiązaniem referencyjnym, z którym można porównywać osiągi swojego rozwiązania jest klasyfikator najbliższego sąsiedztwa (w wersji 5-NN z głosowaniem absolutną większością) działający na cechach otrzymanych bezpośrednio z obrazów znaków metodą składowych głównych (PCA). W testach użyłem 40 składowych głównych. Wyniki klasyfikacji podsumowuje poniższa tabela.

	Zbiór testowy MNIST			„Polski” zbiór testowy		
	OK.	Błąd	Odrzucenie	OK.	Błąd	Odrzucenie
Jakość klasyfikacji	97.21%	2.02%	0.77%	79,96%	15,70%	4,34%

Proszę zwrócić uwagę, na dramatycznie niską jakość rozpoznawania cyfr pisanych przez studentów naszego Wydziału (w „programie obowiązkowym” nie zajmujecie się Państwo tym zbiorem – te wyniki mają tylko ilustrować, że cyfry arabskie można różnie pisać...).

Klasyfikacja cyfr pisanych odręcznie będzie wykonana przez wielowarstwowe sieci neuronowe (z **jedną** warstwą ukrytą) uczonymi algorytmem ze wsteczną propagacją błędów. Dwa podstawowe elementy tego ćwiczenia to:

1. Przygotowanie referencyjnej implementacji uczenia sieci ze wsteczną propagacją błędów (do uzyskania maksymalnej oceny jest konieczna jakość **lepsz**a, niż klasyfikatora 5-NN powyżej).

Realizacja tego punktu będzie wymagać zapewne od Państwa nieco eksperymentowania z architekturą sieci (liczba neuronów w warstwie ukrytej) i parametrami uczenia (wartość i/lub sposób modyfikacji stałej uczenia).

2. Przeczytanie załączonego artykułu Yanna LeCuna i spółki „*Efficient BackProp*” i wybranie z niego modyfikacji sieci (np. postaci funkcji aktywacji) lub algorytmu uczenia (np. metody modyfikacji stałej uczenia, kolejności prezentacji próbek...) i przeprowadzenie podobnej, jak w przypadku sieci referencyjnej, serii eksperymentów w poszukiwaniu najlepszego rozwiązania.

Jako podstawowy sposób uczenia przyjmijmy:

1. Uczenie przyrostowe, czyli wagi aktualizowane po prezentacji każdej próbki.
2. Współczynnik uczenia zmniejszany po epoce uczenia.
3. Kryterium zatrzymania uczenia na podstawie jakości klasyfikacji zbioru walidacyjnego (żeby nie komplikować, przyjmijmy, że zbiorem walidacyjnym będzie zbiór testowy; to oznacza, że będziemy mieć nieco lepsze niż faktycznie wyniki klasyfikacji).

Pewnego komentarza wymaga wybór wyniku klasyfikacji. Rozwiązaniem najprostszym, które jest w zupełności wystarczające dla potrzeb tego ćwiczenia, jest wybór klasy odpowiadającej neuronowi z warstwy wyjściowej z maksymalną odpowiedzią (zakładam,

że jest stosowane kodowanie 1 z N, czyli dla cyfr w warstwie wyjściowej jest 10 neuronów). Opcjonalne jest dodanie decyzji wymijającej (np. jeśli maksymalna odpowiedź nie przekracza ustalonego progu lub gdy różnica między dwoma neuronami o największych sygnałach na wyjściu jest zbyt mała).

W sprawozdaniu proszę zamieścić:

1. Opis metody uczenia ze szczególnym uwzględnieniem wariantu niestandardowego.
2. Dane dotyczące liczby epok i czasów uczenia.
3. Macierze pomyłek i ich analizę dla rekomendowanej sieci.
4. Oczywiście oprócz sprawozdania przesyłacie Państwo kod oraz **parametry generatora liczb losowych** (patrz niżej) – chodzi o to, żebym miał możliwość wygenerowania dokładnie takich samych sieci, jak opisane w sprawozdaniu.

Do pobrania stanu generatora losowego należy użyć polecenia:

```
rstate = rand("state");
```

I tak otrzymany wektor zapisać:

```
save rnd_state.txt rstate
```