**Inteligencja Obliczeniowa**

Praca domowa nr 4 – Przetwarzanie obrazu

Krzysztof Kulewski, 238149, grupa 1, 19.01.2019

## Opis zadania

Celem zadania było dokonanie klasyfikacji obrazów przy użyciu wybranych klasyfikatorów,  
a następnie zestawienie otrzymanych wyników i ich analiza.

## Wybrany zbiór danych i obróbka

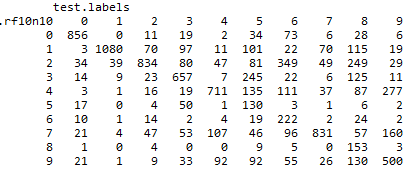
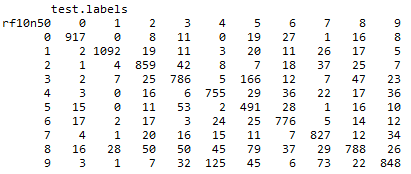
W projekcie postanowiłem posłużyć zbiorem MNIST, czyli zestawem cyfr od 0 do 9 pisanych odręcznie. W jego skład wchodzi 60 000 obrazów treningowych oraz 10 000 obrazów testowych, które zostały użyte do ewaluacji. Każdy z obrazów ma wielkość 28 x 28 pikseli, natomiast każdy piksel jest reprezentowany przez liczbę w zakresie 0 – 255, która opisuje nasycenie kanału alfa.  
Zbiór MNIST zdobył ogromną popularność i stał się wręcz kanoniczny dla zagadnienia klasyfikacji obrazów.   
  
Już na samym początku projektu okazało się, że praca na obrazach w formie plików jest bardzo niewygodna, stąd decyzja, by zamiast plików graficznych, użyć pliku CSV, w którego skład wchodzi 785 kolumn – pierwsza to klasa (cyfry od 0 do 9), a 784 pozostałe reprezentują wartość kanału alfa dla kolejnych pikseli obrazu. Wielkość pliku treningowego to około 110 MB, testowego – 10 MB.  
  
Dane zostały dodatkowo obrobione – wydzielono macierze danych i wektory klas oraz poprawiono nazwy kolumn tak, by mogły być przekazywane do klasyfikatorów:

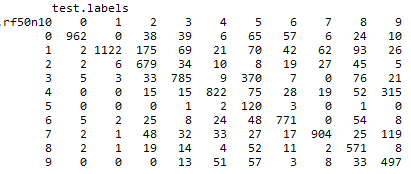
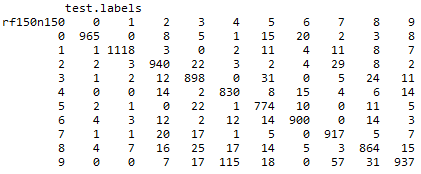
library(readr)  
train = read\_csv("data/mnist\_train.csv")  
test = read\_csv("data/mnist\_test.csv")  
  
train.labels = train[, 1]  
train.labels = as.factor(train.labels$label)  
test.labels = test[, 1]  
test.labels = as.factor(test.labels$label)  
  
train.data = as.matrix(train[, -1])  
test.data = as.matrix(test[, -1])  
  
names(train.data) = make.names(names(train.data))  
names(test.data) = make.names(names(test.data))

## Klasyfikator I – Lasy Losowe (Random Forest)

Pierwszy z użytych klasyfikatorów to lasy losowe. Lasy losowe są w pewnym sensie rozszerzeniem drzew decyzyjnych – zamiast jednego drzewa, tworzony jest „las” składający się z wielu różnych drzew. Wynikiem klasyfikacji jest ta cyfra, która została wybrana przez największą liczbę drzew.  
  
Postanowiłem uruchomić klasyfikator RandomForest z wartościami: 10 drzew, 10 węzłów.  
Model został użyty do klasyfikacji danych testowych. Stworzono również macierz błędu:

library(randomForest)  
model.rf10n10 = randomForest(train.data, train.labels, ntree = 10, maxnodes = 10)  
pred.model.rf10n10 = predict(model.rf10n10, test.data, type = "class")  
cm.rf10n10 = table(pred.model.rf10n10, test.labels)  
acc.rf10n10 = sum(diag(cm.rf10n10))/sum(cm.rf10n10)

**Macierz błędu RF dla 10 drzew, 50 węzłów:**  
  
  
**Dokładność to 59,74%.**  
  
Wartość taka świadczy o dwóch rzeczach:  
- z pewnością nie był to wybór losowy, gdyż w takim przypadku należałoby oczekiwać ok. 10%,  
- jest to wynik mizerny, gdyż dla zbioru MNIST istnieją klasyfikatory o dokładności 99,5%+  
Postanowiłem więc sprawdzić, jak zmiana liczby drzew oraz liczby węzłów wpłynie na uzyskane wyniki.   
  
**Macierz błędu RF dla 10 drzew, 50 węzłów:**  
  
**Dokładność: 81,39%**

**Macierz błędu RF dla 50 drzew, 10 węzłów:  
  
Dokładność: 72,33%.**Jak widać, zwiększenie liczby węzłów znacznie lepiej wpłynęło na jakość klasyfikacji.  
RF50N10 to liczny las, w którym poszczególne drzewa mają relatywnie niską dokładność.  
RF10N50 to natomiast mniejszy las, ale z dużo dokładniejszymi drzewami. Możemy spodziewać się, że w takim przypadku werdykt był bardziej jednomyślny.  
Warto odnotować również fakt, że cyfra „5” jest zdecydowanie najtrudniejsza w klasyfikacji, szczególnie dla lasów o małej liczbie węzłów.  
  
Ostatnia kombinacja miała na celu sprawdzenie jak bardzo jesteśmy w stanie polepszyć dokładność przez dalsze zwiększanie parametrów.   
  
**Macierz błędu RF dla 150 drzew, 150 węzłów:** **Dokładność: 91,43%**

Uzyskany wynik wydaje się być całkiem dobry. Tak jak poprzednio, największe problemy sprawia klasyfikacja cyfry „5”. Ciekawą obserwacją jest również to, iż cyfra „4” była często uznawana za „9”.

## Źródła

1. Materiały z laboratoriów  
2. Dokumentacja R

## Załączniki

1. Kod źródłowy w R (siec.txt)