

POLITECHNIKA WARSZAWSKA  
WYDZIAŁ MATEMATYKI I NAUK INFORMACYJNYCH



## SIECI NEURONOWE

---

**Rozpoznawanie i klasyfikacja pisanych  
cyfr przy użyciu modeli  
matematycznych - raport**

*Autorzy:*

Anna ZAWADZKA  
Piotr WASZKIEWICZ

17 stycznia 2017

## 1 Opis problemu badawczego

Problem badawczy przedstawiony na stronie <https://www.kaggle.com/c/digit-recognizer> polega na rozpoznawaniu i klasyfikacji ręcznie pisanych cyfr poprzez przetwarzanie i analizę obrazów przedstawiających odpowiednie symbole. Zbiory danych zostały zaczerpnięte z publicznej bazy danych MNIST[1].

00112233445566778899  
00112233445566778899

## 2 Cel badań

Projekt zakładał realizację zadania poprzez zbadanie różnych metod, ze szczególnym uwzględnieniem różnych modeli sieci neuronowych. Zbadane zostały dwa rodzaje takich sieci - Backpropagation oraz SoftMax. Wykorzystane zostały również jedne z najpopularniejszych obecnie klasyfikatorów: maszyny wektorów podpierających (SVM)[3], Lasy Losowe[2], kNN[4].

Celem badań było porównanie jakości klasyfikacji dla różnych modeli klasyfikatorów i wskazanie najskuteczniejszego pod względem czasu uczenia, wydajności i jakości udzielanych odpowiedzi. Oprócz tego badania miały na celu rozszerzenie istniejącego wektora cech o nowe, unikalne wartości które, jak przypuszczano, polepszyłyby jakość klasyfikacji. W trakcie obliczeń podjęta została próba odrzucenia tych cech które przeszkadzają lub pogarszają działanie modeli.

## 3 Opis danych

Zbiory danych treningowych oraz testowych pochodzą z publicznej bazy danych MNIST[1]. Każdy element ze zbioru treningowego jest opisany 785 wartościami. Pierwsza liczba określa zakodowaną cyfrę (wartość z przedziału [0, 9]), kolejne 784 wartości są z przedziału [0, 255] i opisują kolory pikseli zeskanowanej cyfry w skali szarości dla obrazka o wymiarach 28x28 pikseli. Zbiór testowy w przeciwieństwie do treningowego nie zawiera informacji o reprezentowanej klasie. Zbiór treningowy i testowy zawierają odpowiednio 42,000 i 28,000 elementów.

## 4 Operacje graficzne

Jednym z założeń dotyczącym zbioru danych było podjęcie próby zmiany wektora cech poprzez dodanie do niego nowych wartości. Zaproponowane zostały cztery nowe cechy którymi były: liczba punktów startowych cyfry, liczba punktów przecięcia cyfry, wektor przecięć cyfry (opisany dokładniej w podrozdziale 4.6) a także informacja o liczbie czarnych pikseli. Wszystkie wymienione wyżej

cechy zostały wyliczone na podstawie szkieletu litery znajdującej się na obrazku, będącego wynikiem operacji zwanej szkieletyzacją.

## 4.1 Dylatacja

Dylatacja to operacja morfologiczna która służy do zamykania małych otworów oraz zatok we wnętrzu symbolu. Obiekty zwiększają swoją objętość i jeśli dwa lub więcej obiektów położonych jest blisko siebie, zrastają się w jedną, większą całość.

## 4.2 Erozja

Erozja jest operacją odwrotną do dylatacji. Jej działanie polega na obcinaniu brzegów obiektu na obrazie.

## 4.3 Szkieletyzacja

Szkieletyzacja, zwana również operacją ścieniania, służy do odchudzania graficznego symbolu tak aby jako wynik otrzymać ten sam symbol narysowany linią o grubości jednego piksela. Obecnie istnieje wiele różnych algorytmów szkieletyzacji różniących się podejściem do zagadnienia, stosowalnością (bywają symbole lepiej ścieniane przez jeden algorytm podczas gdy inne mogą być przez niego odchudzane niepoprawnie) jak i złożonością. Do najbardziej znanych zaliczyć można algorytm K3M, algorytm Zhang-Suen'a, algorytm Guo-Hall'a a także algorytm KMM.

## 4.4 Znajdowanie punktów startowych cyfry <sup>1</sup>

Zazwyczaj pisząc symbole rozpoczynamy i kończymy tę czynność w pewnych szczególnych miejscach. Są to najczęściej niepołączone zakończenia linii które zmuszają do oderwania pióra. Punkty te nazywane są punktami startowymi (choć równie dobrze mogłyby nazywać się punktami końcowymi) a ich liczba pomaga w identyfikacji narysowanego symbolu.

## 4.5 Znajdowanie przecięć w cyfrze <sup>2</sup>

Punktami przecięcia w narysowanym symbolu nazywane są te miejsca w których następuje rozwidlenie ścieżek. Przykładem może być daszek litery T która swój punkt przecięcia posiada w miejscu złączenia daszka i nogi litery. Stosując algorytm opisany w artykule zaimplementowana została funkcjonalność liczenia punktów przecięć w cyfrach.

---

<sup>1</sup>Na podstawie artykułu: <https://arxiv.org/pdf/1202.3884.pdf>

<sup>2</sup>Na podstawie artykułu: <https://arxiv.org/pdf/1202.3884.pdf>

## 4.6 Wektor przecięć cyfry

Wektor przecięć cyfry zawiera sześć elementów. Przechowuje on wartości informujące o liczbie przecięć narysowanej cyfry z liniami prostymi przechodzącymi przez obraz w określonych miejscach, to znaczy w 30% 50% i 70% jego szerokości i wysokości.

## 5 Opis wykorzystanych klasyfikatorów

### 5.1 Backpropagation

### 5.2 SoftMax

### 5.3 SVM

### 5.4 Lasy losowe

### 5.5 kNN

## 6 Opis wyników

Wyniki zaprezentowane w tym rozdziale zostały uzyskane dla dwóch sposobów otrzymywania wektorów cech. Pierwszy z nich zakładał wykorzystanie obrazów dostępnych w ramach zbiorów testowych i traktowanie ich jako jedynej wejścia dla klasyfikatorów. Drugi sposób rozszerzał wektor o zbiór wyekstrahowanych cech, które zostały opisane w rozdziale 4. Dla obu podejść dokonano pełnych testów dla każdego z przygotowanych klasyfikatorów. Poniżej zamieszczone zostały tabelki zawierające wyniki. Każda kolumna oznaczona cyfrą oznacza wyniki dla obiektów z danych testowych reprezentujących tę cyfrę. Wartości w komórkach oznaczają jakość klasyfikacji dla poszczególnych klasyfikatorów (ich skuteczność). Wartości w kolumnie oznaczonej jako *error* przedstawiają ogólny procent błędów dla każdego z klasyfikatorów. Wiersz *summary* zawiera wartości oznaczające skuteczność rozpoznawania każdej z cyfr przez klasyfikatory łącznie.

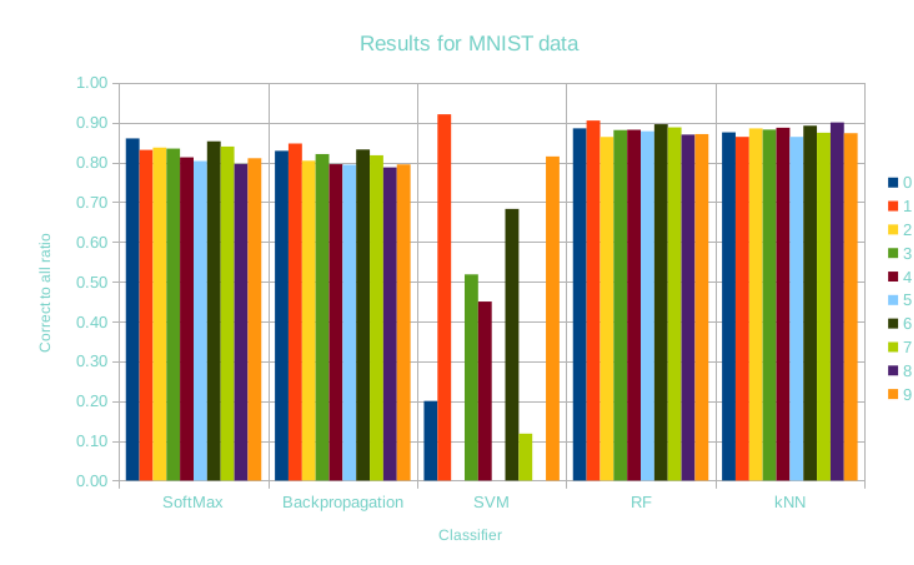
Tabela 1: Wyniki dla danych z podstawowym wektorem cech

	0	1	2	3	4	5	6	7	8	9	error
SoftMax	0.86	0.83	0.84	0.83	0.81	0.80	0.85	0.84	0.80	0.81	0.17
Backpropagation	0.83	0.85	0.80	0.82	0.80	0.79	0.83	0.82	0.79	0.79	0.19
SVM	0.20	0.92	0.00	0.52	0.45	0.00	0.68	0.12	0.00	0.81	0.80
RF	0.89	0.91	0.86	0.88	0.88	0.88	0.90	0.89	0.87	0.87	0.12
kNN	0.88	0.86	0.89	0.88	0.89	0.86	0.89	0.87	0.90	0.87	0.12
summary	0.73	0.87	0.68	0.79	0.77	0.67	0.83	0.71	0.67	0.83	

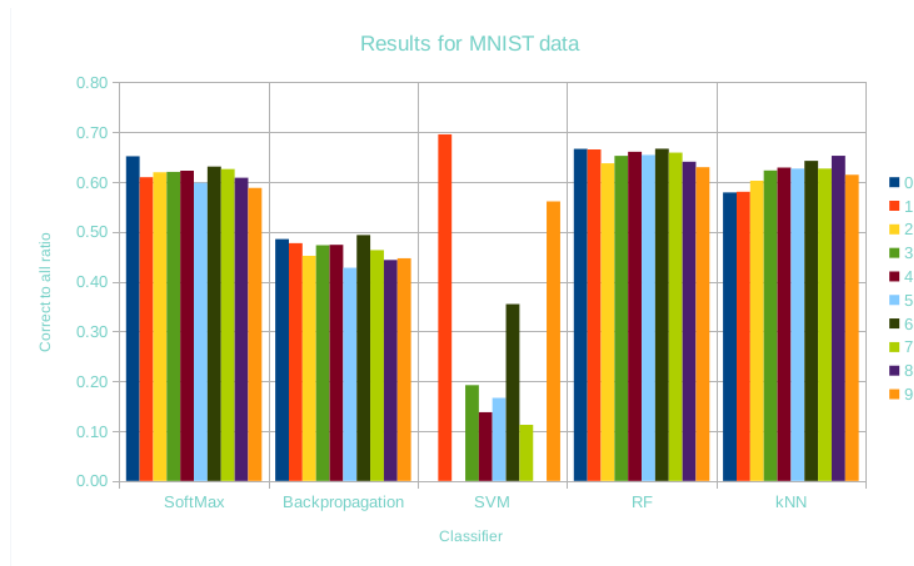
## 7 Podsumowanie

Tabela 2: Wyniki dla danych z rozszerzonym wektorem cech

	0	1	2	3	4	5	6	7	8	9	error
<b>SoftMax</b>	0.65	0.61	0.62	0.62	0.62	0.60	0.63	0.63	0.61	0.59	0.38
<b>Backpropagation</b>	0.49	0.48	0.45	0.47	0.47	0.43	0.49	0.46	0.44	0.45	0.54
<b>SVM</b>	0.00	0.70	0.00	0.19	0.14	0.17	0.35	0.11	0.00	0.56	0.78
<b>RF</b>	0.67	0.67	0.64	0.65	0.66	0.65	0.67	0.66	0.64	0.63	0.35
<b>kNN</b>	0.58	0.58	0.60	0.62	0.63	0.63	0.64	0.63	0.65	0.61	0.38
<b>summary</b>	0.48	0.61	0.46	0.51	0.50	0.49	0.56	0.50	0.47	0.57	



Rysunek 1: Wyniki dla danych z podstawowym wektorem cech przedstawione na wykresie



Rysunek 2: Wyniki dla danych z rozszerzonym wektorem cech przedstawione na wykresie

## Literatura

- [1] LeCun, Y., Cortes, C., and Burges, C., *The MNIST database of handwritten digits*, in: <http://yann.lecun.com/exdb/mnist>.
- [2] Breiman, L., *Random Forests*. Machine Learning 45 (1), 2001
- [3] Cortes, C., Vapnik, V., *Support-vector networks*. Machine Learning 20 (3), 1995.
- [4] Altman N. S., *An introduction to kernel and nearest-neighbor nonparametric regression*. The American Statistician 46 (3), 1992.
- [5] Scholkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J., Platt, J., *Support Vector Method for Novelty Detection*, Advances in Neural Information Processing Systems 12, 1992.
- [6] Wang, Y., Casasent, D., *A Support Vector Hierarchical Method for multi-class classification and rejection*, Proc. of Int. Joint Conf. on Neural Networks, 2009.
- [7] Dinesh Dileep *A feature extraction technique based on character geometry for character recognition*, <https://arxiv.org/pdf/1202.3884.pdf>
- [8] [http://scikit-learn.org/stable/modules/grid\\_search.html](http://scikit-learn.org/stable/modules/grid_search.html)