JOURNAL OF COMPUTER SCIENCES INSTITUTE

JCSI 15 (2020) 109-113
Received: 25 October 2019
Accepted: 31 October 2019

Sieci neuronowe w rozpoznawaniu pisma odręcznego

Neural networks in recognition of handwriting

Bernadetta Michalik*, Marek Miłosz

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

Artificial neural networks consist of many simple elements capable of processing data. These are tools inspired by the construction of the human brain, used in machine learning. The aim of the research was to analyze the occuracy of the created neural network in the process of handwriting recognition. The article presents the results obtained during the learning and testing of a convolution network with a different number of hidden layers. Each time learning and testing the network was carried out using the same set of images (taken from the publicly available IAM database) depicting handwritten words in English.

Keywords: handwriting; artificial neural network; word recognition

Streszczenie

Sztuczne sieci neuronowe składają się z wielu prostych elementów zdolnych do przetwarzania danych. To narzędzia inspirowane budową ludzkiego mózgu, stosowane w uczeniu maszynowym. Celem badań była analiza dokładności odpowiedzi stworzonej sieci neuronowej w procesie rozpoznawania pisma odręcznego. W artykule przedstawiono wyniki uzyskane podczas nauki i testowania sieci konwolucyjnej o różnej liczbie warstw ukrytych. Każdorazowo uczenie i testowanie sieci realizowane było za pomocą tego samego zbioru obrazów (zaczerpniętych z ogólnodostępnej bazy IAM Handrwiting Database) przedstawiających słowa pisane odręcznie w języku angielskim.

Słowa kluczowe: sztuczne sieci neuronowe; rozpoznawanie słów

*Corresponding author

Email address: **bernadetta768@gmail.com** (B. Michalik)

1. Wstęp

Sztuczne sieci neuronowe (SSN) są narzędziami do modelowania obliczeniowego, które pojawiły się w wielu dyscyplinach w zakresie kształtowania złożonych problemów w świecie rzeczywistym. SSN można zdefiniować jako struktury złożone z gęsto połączonych prostych elementów przetwarzających (zwanych sztucznymi neuronami lub węzłami), które są zdolne do wykonywania masowo równoległych obliczeń w celu przetwarzania danych i reprezentacji wiedzy [1]. Mimo że SSN są abstrakcjami odpowiedników biologicznych, ich ideą nie jest replikacja działania systemów biologicznych, ale wykorzystanie wiedzy o funkcjonowaniu sieci biologicznych do rozwiązywania złożonych problemów. Atrakcyjność SSN wynika z ich niezwykłych właściwości takich jak nieliniowość, wysoka równoległość, odporność na awarie i uszkodzenia, zdolność do obsługi nieprecyzyjnych i rozmytych informacji oraz ich zdolność do generalizowania [2].

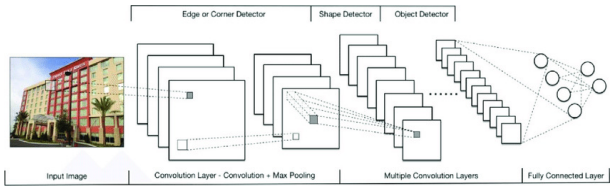
Obecnie coraz więcej osób korzysta z obrazów do reprezentowania i przesyłania informacji. Popularne jest również wydobywanie ważnych informacji z obrazów. Rozpoznawanie obrazu jest ważnym obszarem badawczym ze względu na jego szerokie zastosowania. W stosunkowo młodej dziedzinie komputerowego rozpoznawania wzorców jednym z trudniejszych zadań jest dokładne rozpoznawanie ludzkiego pisma, ponieważ istnieje znaczna różnorodność pisma między każdym z osobna. Chociaż człowiek zazwyczaj nie ma z tym żadnych problemów, trudniej jest nauczyć komputery rozpoznawać odręczne pismo. Jednym z rozwiązań jest użycie optycznego rozpoznawania znaków (OCR) do konwersji zeskanowanych dokumentów papierowych na formę do odczytu maszynowego (ASCII). Technologia ta umożliwia rozróżnianie drukowanych lub odręcznych znaków tekstowych wewnątrz cyfrowych obrazów fizycznych dokumentów. Odbywa się to poprzez zrobienie najpierw zdjęcia dokumentu lub zeskanowanie go. Tworzy to obraz rastrowy złożony z danych, które komputer rozumie, a przez specjalnie zaprogramowane algorytmy, z których większość jest wykorzystywana w dziedzinie sztucznej inteligencji, komputer rozpoznaje wzorce na obrazie. Następnie program tworzy lub wyprowadza kody znaków, zwykle ASCII, które są równoważne rozpoznanym znakom z obrazu wejściowego. Większość programów OCR musi zostać przeszkolonych, aby mogły lepiej rozpoznawać znaki [3].

2. Głębokie uczenie

Technologia uczenia maszynowego zasila wiele aspektów współczesnego społeczeństwa: od wyszukiwania w sieci przez filtrowanie treści w sieciach społecznościowych po rekomendacje na stronach e-commerce i jest coraz bardziej obecna w produktach konsumenckich, takich jak aparaty fotograficzne i smartfony. Systemy uczenia maszynowego służą do identyfikacji obiektów na obrazach, transkrypcji mowy na tekst, dopasowania wiadomości, postów lub produktów do zainteresowań użytkowników oraz wyboru odpowied-

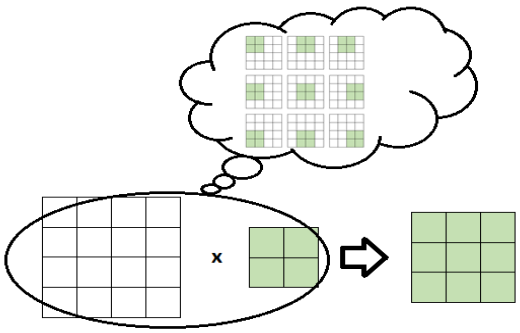
nich wyników wyszukiwania. Coraz częściej aplikacje te korzystają z klasy technik zwanych głębokim uczeniem się, które potrafią przetwarzać surowe dane do wektora cech. Uczenie się reprezentacji to zestaw metod, które pozwalają zasilać maszynę surowymi danymi i automatycznie odkrywać reprezentacje potrzebne do wykrywania lub klasyfikacji [4]. Metody dogłębnego uczenia się to metody uczenia się reprezentacji z wieloma poziomami reprezentacji, uzyskiwane przez złożenie prostych, ale nieliniowych modułów, z których każdy przekształca reprezentację na jednym poziomie w reprezentację na wyższym, nieco bardziej abstrakcyjnym poziomie . Dzięki wystarczającej liczbie takich transformacji można nauczyć się bardzo złożonych funkcji. W przypadku zadań klasyfikacyjnych wyższe warstwy reprezentacji wzmacniają aspekty danych wejściowych, które są ważne dla dyskryminacji i eliminują nieistotne zmiany [5].

W ostatnich latach coraz częściej wykorzystuje się konwolucyjne sieci neuronowe (CNN, ConvNets), narzędzia do głębokiego uczenia się. Są one szczególnie odpowiednie dla obrazów jako danych wejściowych. Na wejściu mogą się również pojawić się dane w postaci tekstu, sygnałów i inne. CNN składają się z trzech rodzajów warstw. Są to warstwy splotowe, warstwy pulujące i warstwy w pełni połączone [6]. Uproszczona architektura CNN została zilustrowana na rysunku 1.



Rysunek 1: Schemat konwolucyjnej sieci neuronowej [7]

Warstwa splotowa jest miejscem, gdzie wykonuje się operacje matematyczne na matrycach, z których jedna z nich jest zbiorem danych wejściowych a druga jądrem.



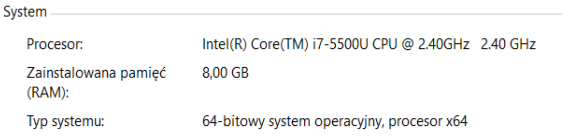
Rysunek 2: Procedura mnożenia danych wejściowych i wag

Po lewej stronie diagramu (rysunek 2) znajduje się matryca obrazu (tablica 4x4 w kolorze białym), pośrodku znajduje się jądro (tablica 2x2 w kolorze zielonym), a po prawej wynik splotu. Szczegóły przejścia jądra przez matrycę obrazu przedstawiono w chmurze na diagramie. Oto, co się stało: filtr odczytuje kolejno, od lewej do prawej i od góry do dołu wszystkie piksele w obszarze działania jądra i pomnożył wartość każdego z nich przez odpowiednią wartość jądra. Wynikiem działań warstwy splotowej jest mapa aktywacyjna [8].

Głównym zadaniem warstwy łączącej jest próbkowanie w dół w celu zmniejszenia złożoności kolejnych warstw, a tym samym dalsze zmniejszanie liczby parametrów i złożoności obliczeniowej modelu. W dziedzinie przetwarzania obrazu można to uznać za podobne do procesu zmniejszania rozdzielczości. Warstwy pulujące wprowadza się do modelu, przeciwdziałając przeuczeniu [9]. Ostatnim elementem jest w pełni połączona warstwa, która zawiera neurony bezpośrednio połączone z neuronami w dwóch sąsiadujących warstwach, bez połączenia z żadnymi warstwami w nich zawartymi. Jest to analogiczne do sposobu, w jaki neurony są ułożone w tradycyjnych formach sztucznych sieci neuronowych [10].

3. Realizacja badań

Badania zaprezentowanego poniżej modelu zostały przeprowadzone na urządzeniu o parametrach przedstawionych na rysunku 3. Dodatkowo, aby nie zakłócać przebiegu testów, wszystkie niepotrzebne procesy zostały zamknięte.

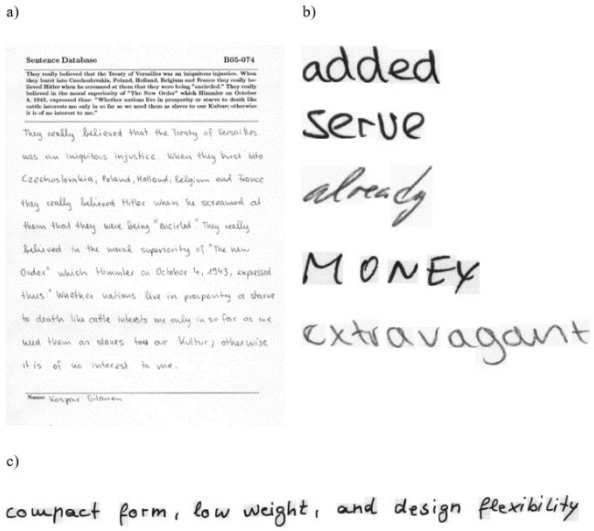


Rysunek 3: Parametry urządzenia.

3.1. Zestaw danych

W badaniach wykorzystano zasoby bazy IAM (rys. 4). Jest to zbiór zdjęć przedstawiających odręcznie napisane teksty w języku angielskim. Pierwsza publikacja bazy IAM ukazała się w 1999 roku. Zbiór wykorzystywany jest głównie do nauki rozpoznawania pisma odręcznego jak również do badań nad identyfikacją autora. Do utworzenia bazy przyczyniło się ponad 600 autorów. Składa się z 1539 stron odręcznie pisanego tekstu, 5685 wyodrębnionych zdań, 13353 linii z tekstem i 115320 słów. Zasoby bazy to zeskanowane dokumenty zapisane w formacie png, które posiadają 256 poziomów szarości i rozdzielczość 300 dpi [11].

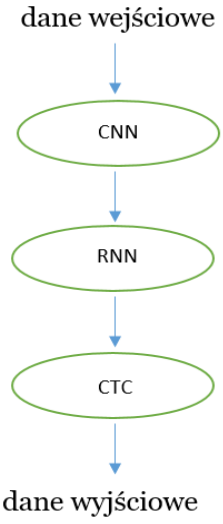
Na potrzeby pracy wykorzystano jedynie zbiór wyodrębnionych słów, które podzielono na trzy zestawy: treningowy, walidacyjny i testowy. Pierwszy z nich stanowi 70 % całego zbioru pojedynczych słów, drugi to 20 %, natomiast reszta (10 %) jest zbiorem testowym.



Rysunek 4: Przykład zasobów bazy IAM: strona (a), wiersz (c) i pojedyncze słowa (b) [11].

3.2. Model sieci neuronowej

Skonstruowano model sieci neuronowej, który stanowi połączenie trzech warstw: CNN (convolutional neural network), RNN (recurrent neural network) i CTC (connectionist temporal classification). Uproszczony schemat sieci przedstawiono na rysunku 5.



Rysunek 5: Schemat modelu

Na wejściu sieci konwolucyjnej znajduje się zbiór, który składa się z szarych obrazów. Ze względu na różną długość wyrazów przyjęto jedną (uniwersalną) wielkość, która wynosi 128 x 32. Obrazy o innych rozmiarach zostały przekształcone do rozmiarów uniwersalnych.

Listing 1: Definicja sieci konwolucyjnej z pięcioma warstwami ukrytymi

```
def conv_net(self):
conv = tf.nn.conv2d(self.input,
filter=tf.Variable(tf.random.truncated_normal([
3, 3, 1, 16], stddev=0.5)), padding='SAME',
strides=(1,1,1,1))
relu = tf.nn.relu(conv)
pool = tf.nn.max_pool2d(relu, (1, 2, 2, 1), (1,
2, 2, 1), 'SAME')
conv2 = tf.nn.conv2d(pool,
filter=tf.Variable(tf.random.truncated_normal([
3, 3, 16, 32], stddev=0.5)), padding='SAME',
strides=(1,1,1,1))
relu2 = tf.nn.relu(conv2)
pool2 = tf.nn.max_pool2d(relu2, (1, 2, 2, 1),
(1, 2, 2, 1), 'SAME')
conv3 = tf.nn.conv2d(pool2,
filter=tf.Variable(tf.random.truncated_normal([
3, 3, 32, 64], stddev=0.5)), padding='SAME',
strides=(1,1,1,1))
relu3 = tf.nn.relu(conv3)
pool3 = tf.nn.max_pool2d(relu3, (1, 1, 2, 1),
(1, 1, 2, 1), 'SAME')
conv4 = tf.nn.conv2d(pool3,
filter=tf.Variable(tf.random.truncated_normal([
3, 3, 64, 128], stddev=0.5)), padding='SAME',
strides=(1,1,1,1))
relu4 = tf.nn.relu(conv4)
pool4 = tf.nn.max_pool2d(relu4, (1, 1, 2, 1),
(1, 1, 2, 1), 'SAME')
conv5 = tf.nn.conv2d(pool4,
filter=tf.Variable(tf.random.truncated_normal([
3, 3, 128, 256], stddev=0.5)), padding='SAME',
strides=(1,1,1,1))
relu5 = tf.nn.relu(conv5)
pool5 = tf.nn.max_pool2d(relu5, (1, 1, 2, 1),
(1, 1, 2, 1), 'SAME')
```

Listing 1 opisuje sieć konwolucyjną zawierającą pięć warstw ukrytych. Składa się z warstwy splotowej, ReLU (rectified linear Units) oraz warstwy pulującej. Dane z wyjścia sieci splotowej są jednocześnie danymi wejściowymi sieci RNN, która składa się z dwóch warstw (listing 2).

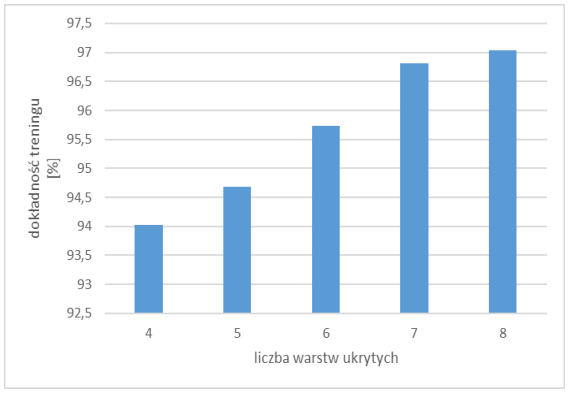
Listing 2: Definicja sieci rekurencyjnej

```
def rnn_net(self):
rnn_size = 256
cell1 = tf.contrib.rnn.LSTMCell(rnn_size)
cell2 = tf.contrib.rnn.LSTMCell(rnn_size)
cell = tf.contrib.rnn.MultiRNNCell([cell1,
cell2], state_is_tuple=True)
([encoder_outputs, encoder_state], _) =
tf.nn.bidirectional_dynamic_rnn(cell_fw=cell,
cell_bw=cell, inputs=tf.squeeze(self.pool5,
axis=[2]), dtype=tf.float32)
concat =
tf.expand_dims(tf.concat([encoder_outputs,
encoder_state], 2), 2)
   self.rnnOutput =
tf.squeeze(tf.nn.atrous_conv2d(value=concat,
filters=tf.Variable(tf.random.truncated_normal(
[1, 1, rnn_size * 2, len(self.charList) + 1],
stddev=0.5)), rate=1, padding='SAME'))
```

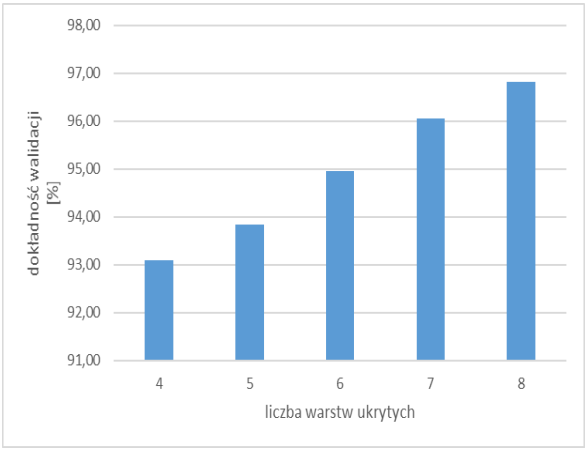
4. Wyniki

Na rysunku 5, 6 i 7 przedstawiono dokładność uzyskaną podczas poszczególnych etapów pracy sieci neuronowej przy różnej liczbie warstw ukrytych. W każdym przypadku zauważono, że wraz ze wzrostem liczby warstw ukrytych rośnie dokładność modelu. Najwyższe wyniki uzyskano podczas działań na zestawie treningowym, czyli na danych służących dopasowaniu wag. Nieco niższe wyniki, ale nadal zadowalające, uzyskano podczas walidacji. Rzeczywistą dokładność modelu przed-

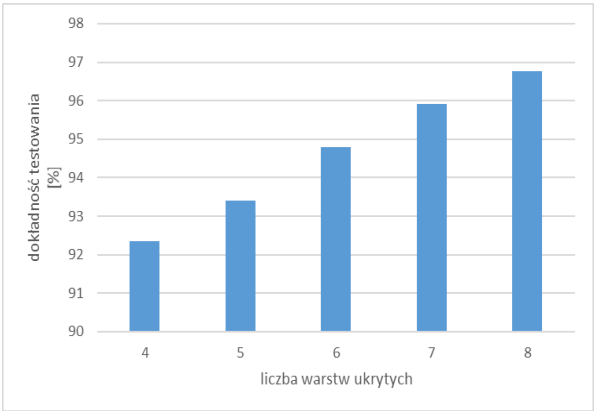
stawia rysunek 10. Nauczona sieć jest sprawdzana z wykorzystaniem zbioru testującego. Wszystkie wyniki przekraczają próg 90 %. Najwyższe wyniki uzyskano dla modelu z siedmioma (95,92 %) i ośmioma (96,77 %) warstwami ukrytymi sieci konwolucyjnej.
Szczegółowe wyniki przeprowadzonych badań przedstawiono w tabeli 1.



Rysunek 6: Wyniki uzyskane podczas treningu sieci w zależności od ilości warstw ukrytych



Rysunek 7: Wyniki uzyskane podczas walidacji w zależności od ilości warstw ukrytych



Rysunek 8: Wyniki uzyskane podczas testowania sieci w zależności od ilości warstw ukrytych

Tabela 1: Zbiorcze zestawienie wyników

Liczba warstw ukrytych	Dokładność trenowania [%]	Dokładność walidacji [%]	Dokładność testowania [%]
4	94,02	93,10	92,34
5	94,69	93,84	93,41
6	95,73	94,96	94,80
7	96,81	96,05	95,92
8	97,03	96,83	96,77

5. Wnioski

Sztuczne sieci neuronowe są coraz częściej wykorzystywane przez naukowców i programistów ze względu na ich bardzo dobre wyniki. W powyższych badaniach sprawdzano jak zmienia się dokładność odpowiedzi sieci wraz ze zmianą jej parametrów- w tym przypadku zmianie podlegała liczba warstw ukrytych. Wyniki wykazały wzrost dokładności modelu wraz ze wzrostem liczby warstw ukrytych sieci konwolucyjnej. Nie zanotowano znaczących różnic pomiędzy kolejnymi wynikami. Już przy czterech warstwach ukrytych wyniki modelu przekroczyły 90 %, a najlepsze rezultaty zaobserwowano przy siedmiu i ośmiu warstwach (wyniki przekroczyły 95 %).

Literatura

[1] R. Tadeusiewicz, M. Szaleniec, Leksykon sieci neuronowych. Wydawnictwo Fundacji "Projekt Nauka" (2015).

[2] K. Różanowski, Sztuczna inteligencja rozwój, szanse i zagrożenia. Zeszyty Naukowe Warszawskiej Wyższej Szkoły Informatyki (2007).

[3] R. Mithe, S. Indalkar, N. Divekar, Optical character recognition. International journal of recent technology and engineering (IJRTE), 2 (2013) 72-75.

[4] H. Li, Z. Lin, X. Shen, J. Brandt, G. Hua, A convolutional neural network cascade for face detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, (2015) 5325-5334.

[5] Y. LeCun, Y. Bengio, G. Hinton, Deep learning. Nature, 521 (2015) 436-444.

[6] M. Liang, H. Hu, Recurrent convolutional neural network for object recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, (2015) 3367-3375.

[7] Y. Ma, Z. Xiang, Q. Du, W. Fan, Effects of user-provided photos on hotel review helpfulness: An analytical approach with deep leaning. International Journal of Hospitality Management, 71 (2018) 120-131.

[8] Y. Hou, H. Zhao, Handwritten digit recognition based on depth neural network. International Conference on Intelligent Informatics and Biomedical Sciences, Okinawa, 2017.

[9] S. Albawi, T. A. Mohammed, S. Al-Zawi, Understanding of a convolutional neural network. In 2017 International Conference on Engineering and Technology (ICET), (2017) 1-6 .

[10] T. N. Sainath, A. R. Mohamed, B. Kingsbury, B. Ramabhadran, Deep convolutional neural networks for LVCSR. In 2013 IEEE international conference on acoustics, speech and signal processing, (2013) 8614-8618.

[11] http://www.fki.inf.unibe.ch/databases/iam-handwriting-database [05.02.2019]