

ANALIZA DIVERȘILOR ALGORITMI PENTRU CREAREA UNUI MODEL EFICIENT DE RECUNOAȘTERE A SCRISULUI DE MÂNĂ

Cristian CIOBANU¹
Angela BEREGOI*¹
Sergiu SCROB²
Inga LISNIC²

¹ Universitatea Tehnică a Moldovei, Facultatea Calculatoare, Informatică și Microelectronică, Departamentul Ingineria Software și Automatică, grupa TI-172, Chișinău, Moldova

² Universitatea Tehnică a Moldovei, Facultatea Calculatoare, Informatică și Microelectronică, Departamentul Ingineria Software și Automatică, Doctorand, Chișinău, Republica Moldova

*Autorul corespondent: Beregoi Angela, angela.beregoi@ati.utm.md

Rezumat: În multe domenii diferite, există o cerere abundentă pentru stocarea informațiilor în format digital din informația disponibilă în documente, imagini tipărite sau scrise de mână pentru a reutiliza ulterior aceste informații cu ajutorul calculatoarelor. Recunoașterea optică a caracterelor (OCR) este o zonă de cercetare care încearcă să dezvolte un sistem informațional ce are capacitatea de a extrage și prelucra textul din imagini în mod automat. Obiectivul OCR este de a realiza modificarea sau conversia oricărei forme de text într-un format digital editabil, cum ar fi text scris, imagini de text tipărite sau scanate, pentru o prelucrare mai profundă și ulterioară. În această lucrare cercetăm și analizăm cea mai bună arhitectură a unei rețele convoluționale pentru un model OCR. În primul rând, analizăm fazele generale ale unui sistem OCR. Apoi, evidențiem cei mai buni algoritmi de construire a unei rețele neuronale și construim trei arhitecturi diferite pentru a obține cele mai bune rezultate.

Cuvinte cheie: OCR, model, acuratețe, antrenare, caractere.

Introducere

Tehnologia de recunoaștere optică a caracterelor permite unei mașini să recunoască automat textul în astfel de documente. Performanța și acuratețea OCR depind direct de calitatea documentelor de intrare. Întrucât cercetarea OCR este un domeniu activ și important în problemele generale de recunoaștere a șabloanelor, datorită progresului rapid, sunt necesare revizuirii complete ale domeniului în mod regulat pentru a urmări noile progrese. Această lucrare încearcă să găsească cea mai bună arhitectură pentru crearea unui model pentru OCR. În Figura 1 este reprezentată organigrama sistemului folosit pentru instruirea modelului de recunoaștere a textului. Toate arhitecturile construite în cadrul sistemului de OCR, constau dintr-o rețea neuronală în diferite straturi, construită cu ajutorul bibliotecilor TensorFlow, Keras, Numpy și Scipy. A fost utilizat un set de date, MNIST, ce face parte dintr-un set mai mare, disponibil de la NIST (National Institute of Standards and Tehnology). Setul conține litere scrise de mână, sub forma unui set de antrenare de 60. 000 de exemple, precum și un set de test de 10. 000 de exemple [1]. Setul de date CoMNIST reprezintă o bază de date care conține peste 26. 000 de imagini. Pentru antrenarea arhitecturilor s-a folosit setul de 26 de litere din alfabetul latin. Un alt set de date utilizat este Kaggle, care conține peste 100, 000 de simboluri matematice. Din acest set au fost extrase pentru antrenare și testare literele latin. Pentru antrenarea modelului imaginile s-au adus la același format: au dimensiunile de 28x28 pixeli, tipul este .png, și sunt setate pe 3 canale de ieșire - RGB. Pentru convertirea imaginilor se folosește biblioteca Pillow, metodele thumbnail, convert și resize. Folosirea lui Adam este o alegere mai fiabilă, deoarece alege rata de învățare pentru fiecare parametru într-un mod adaptiv și, prin urmare, nu reduce rețeaua de detectare pentru a explora la fel de radical ca în cazul utilizării SGD [2-3].

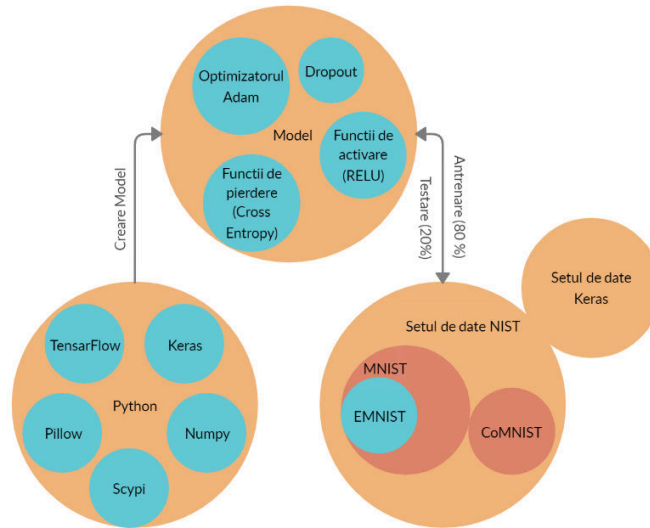


Figura 1. Organigrama modelului construit

Descrierea modelelor construite

Primele două arhitecturi ale modelului de OCR, constau la nivel fundamental dintr-o rețea neuronală din două straturi ascunse a câte 512 neuroni fiecare, pe când a treia are doar un strat. Numărul de exemplare de imagini propagate în rețea o singură dată (batch_size) are mărimea de 128. Kernel-ul are mărimea 2. Neuronii de intrare primesc ca date inițiale un vector unidimensional. Tabelul 1 descrie diferența și punctele comune dintre arhitecturi.

Tabelul 1

Diferența și punctele comune dintre arhitecturi

	Prima arhitectură	A doua arhitectură	A treia arhitectură
Straturi	<ul style="list-style-type: none"> • 2 straturi ascunse 	<ul style="list-style-type: none"> • 2 straturi ascunse • 2 straturi Dropout 	<ul style="list-style-type: none"> • 1 strat ascuns • 3 straturi Dropout • 2 straturi Convolutional • 2 straturi Max pooling
Algoritm de optimizare	Adam	Adam	Adam
Metrică	Acuratețe	Acuratețe	Acuratețe
Funcții de activare	RELU, Softmax	RELU, Softmax	RELU, Softmax
Numărul de epoci	40	40	40
Numărul de imagini	135376	135376	12824
Funcția Loss	CrossEntropy	CrossEntropy	CrossEntropy
Nr. de neuroni în strat	512	512	512

Prima arhitectură a modelului e reprezentată în Figura 2. Tipul conectării neuronilor este Dense (fiecare neuron este conectat cu ceilalți).

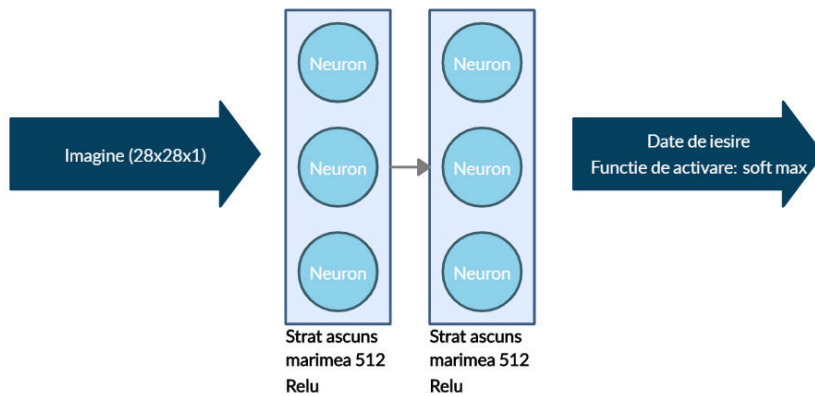


Figura 2. Prima arhitectură

A doua arhitectură, reprezentată în Figura 3, este un MLP (Multi-layer Perceptron) cu Dropout. Tipul conectării neuronilor este Dense. La fiecare strat ascuns se aplică principiul Dropout.

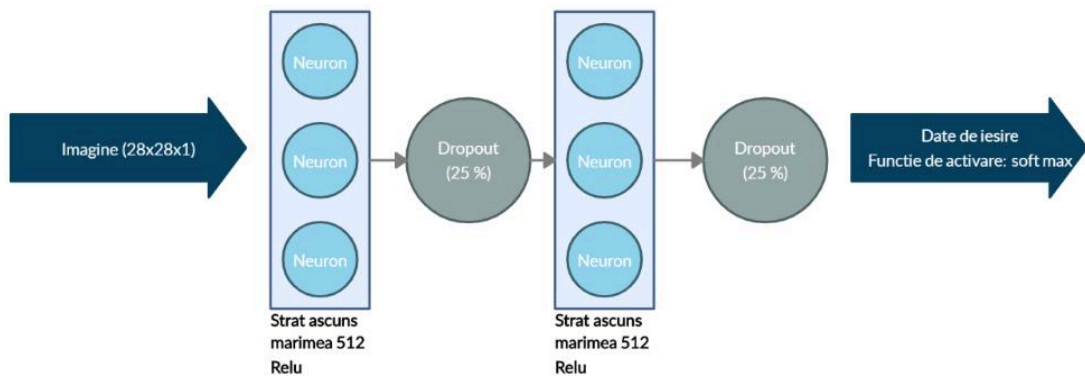


Figura 3. A doua arhitectură

A treia arhitectură, un CNN (Convolutional Neural Network) cu Dropout, e reprezentată în Figura 4. La neuronii de intrare se aplică principiul convoluțional, după care se aplică un strat de Dropout. La neuronii din stratul Dropout se conectează un strat de Max pooling.

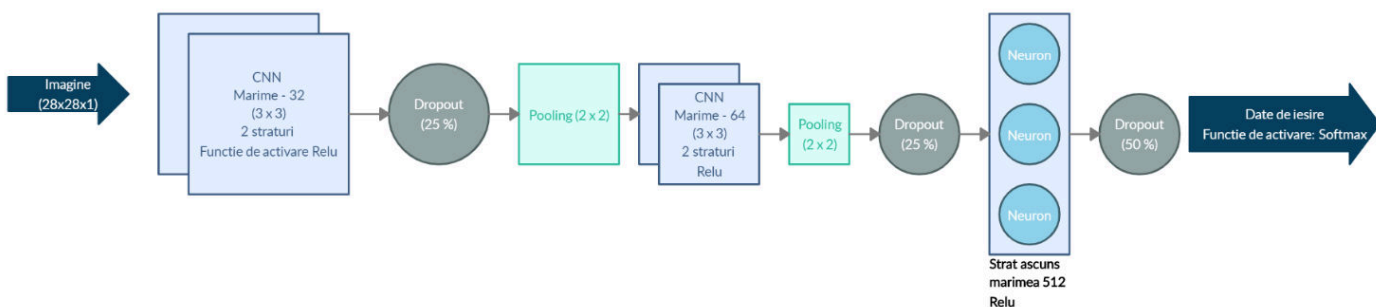


Figura 4. A treia arhitectură

Din nou se aplică principiul Convolutional- Max pooling. Tipul conectării neuronilor după stratul Max pooling este Dense. La ultimul strat, cel ascuns, se aplică principiul Dropout.

Rezultatele experimentale în urma simulării

În figura 5 sunt prezentate rezultatele experimentale a celor 3 arhitecturi. Corectitudinea testelor se verifică în baza metricii de acuratețe. Este o metrică provocatoare, deoarece e necesar ca rezultatul recunoașterii caracterului să fie identic cu însăși caracterul.

Prima arhitectură

A doua arhitectură

A treia arhitectură

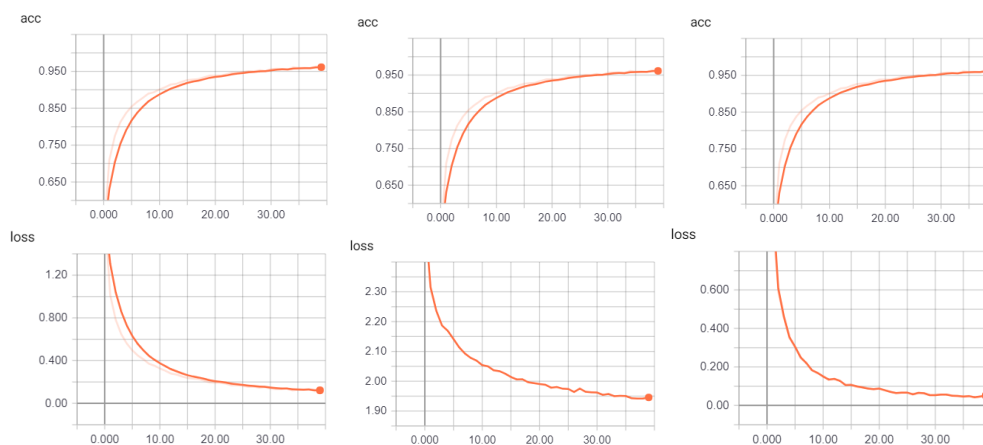


Figura 5. Graficele metricei de acuratețe și funcției de pierdere pentru toate structurile

Concluzii

O eventuală aplicație construită pe baza acestui model se poate dovedi de un real ajutor când nu dorim să tastăm un text și să deschidem numeroase aplicații pentru a obține informații despre conținutul lui, ci doar să realizăm o captură, iar de aici să avem posibilități multiple de distribuire și stocare a rezultatelor. Acest proiect ar putea fi dezvoltat pe viitor într-un OCR care ar recunoaște cuvinte, care s-ar conecta la o bază de date de cuvinte a unei limbi și ar putea recunoaște texte. Mai apoi, o dezvoltare posibilă ar fi un OCR multilingv. Pe baza acestei arhitecturi eficiente [4] se pot construi modele, care pot rezolva probleme de învățare super-vizată și nesupervizată, rezolvând problema extragerii și prelucrării textului scris de mână în mod automat. Din analiza figurii 5 putem observa că prima arhitectură este afectată de re-instruire. În mulțimea testelor cunoscute, se poate observa o acuratețe de 98%. Pe când pe teste necunoscute, acuratețea lui scade până la 50%. A doua arhitectură deja nu este afectat de re-instruire, dar din cauza numărului mic de teste și iterații acuratețea lui ajunge până la 44%. A treia arhitectură este lipsită de overfitting, iar acuratețea lui ajunge până la 99%. Această arhitectură a fost antrenată pe un număr mai mic de teste din cauza vitezei mici de antrenare.

Referințe

5. GOLBAN, I., MELINTE, T., ONIȚA, D. M.. *Recunoașterea cifrelor scrise de mână folosind rețele neuronale adânci* [online]. [accesat 17.02.2020]. Disponibil: http://adrianabirlutiu.uab.ro/sesiuni_studenti/2017_Golban.pdf
6. An overview of gradient descent optimization algorithms. [online]. [accesat 15.02.2020]. Disponibil: <https://ruder.io/optimizing-gradient-descent/>
7. DIEDERIK, P. K., JIMMY, L. B.. Adam: A method for stochastic optimization. In: *ICLR - 2015*. San Diego, CA, USA, pp. 7-9. 2015.