Deep Learning pentru descrierea automată a imaginilor în limbaj natural - *Image Captioning*

Anca Mihaela HOTĂRAN, Mihnea Horia VREJOIU

Institutul Naţional de Cercetare-Dezvoltare în Informatică – ICI Bucureşti

anca.hotaran@ici.ro, mihnea.vrejoiu@ici.ro

Rezumat: Sintagma *Image Captioning* (IC) în contextul vederii artificiale se referă la generarea automată de descrieri textuale asociate imaginilor digitale. Nu este vorba doar de recunoaşterea obiectelor din aceste imagini, ci şi de descrierea atributelor lor, precum şi a relaţiilor şi interacţiunilor dintre ele, totul exprimat textual în limbaj natural, corect din punct de vedere sintactic şi semantic. Sintetic, paşii principali în generarea automată de descrieri textuale asociate imaginilor sunt: a) – extragerea informaţiei vizuale din imagine şi, b) – „traducerea" acesteia într-un text adecvat şi semnificativ. Evoluţiile spectaculoase din domeniul reţelelor neuronale adânci şi *Deep Learning* din ultimii ani au condus la progrese absolut remarcabile şi în domeniul IC, calitatea textelor descriptive generate fiind îmbunătăţită substanţial. Reţelele neuronale convoluţionale (*Convolutional Neural Network* – CNN) au fost folosite în mod natural pentru obţinerea de reprezentări vectoriale esenţializate ale caracteristicilor (*features*) din imagini, iar reţelele neuronale recurente (*Recurrent Neural Network* – RNN), în particular de tip *Long Short-Term Memory* (LSTM), au fost utilizate pentru decodarea acestor reprezentări în fraze în limbaj natural. În lucrarea de faţă prezentăm o trecere în revistă a noilor tehnici şi metode bazate pe *Deep Learning* utilizate în domeniul IC, cu detalierea şi analizarea ca studiu de caz a uneia dintre cele mai performante dintre acestea, utilizând o arhitectură de tip *encoder-decoder* combinată cu un mecanism de focalizare a atenţiei vizuale pe regiunile corespunzătoare relevante din imagine la generarea fiecărui cuvânt nou din secvenţa de ieşire.

Cuvinte cheie: descriere textuală a imaginilor, învăţare automată, învăţare profundă, reţea neuronală adâncă, reţea convoluţională, reţea recurentă, LSTM, *encoder-decoder*, mecanism atenţional.

Deep Learning for automatically describing images in natural language - *Image Captioning*

Abstract: *Image Captioning* (IC) in Computer Vision context refers to the automatic generation of textual descriptions associated with digital images. It is not only the recognition of the objects in these images, but also the description of their properties, as well as the relationships and interactions between them, all expressed textually in natural language, syntactically and semantically correct. Synthetically, the main steps in the automatic generation of textual descriptions associated with the images are: a) – extracting the visual information from the image, and, b) – "translating" it into an adequate and meaningful text. The spectacular developments in the field of deep neural networks and *Deep Learning* in recent years have led to absolutely remarkable progress also in the field of IC, the quality of the generated descriptive texts being substantially improved. *Convolutional Neural Networks* (CNN) have been naturally used to obtain essentialized vectorial representations of the image *features*, and *Recurrent Neural Networks* (RNN), in particular *Long Short-Term Memory* (LSTM), were used to decode these representations into phrases in natural language. In this paper we present an overview of the new techniques and methods based on *Deep Learning* used in the IC field, while also detailing and analyzing, as a case study, one of the best performing ones, using an *encoder-decoder* architecture combined with a mechanism for focusing the visual attention on the appropriate relevant regions of the image when generating each new word in the output sequence.

Keywords: image captioning, machine learning, deep learning, deep neural network, convolutional network, recurrent network, LSTM, encoder-decoder, attentional mechanism.

1. Introducere

Generarea automată a unui text descriptiv asociat unei imagini (*Image Captioning* - IC), ca parte a obiectivului mai complex de înţelegere a scenelor în vederea artificială, este în sine o sarcină netrivială. Este vorba de mai mult decât de identificarea obiectelor care apar în imagini, respectiv de surprinderea şi exprimarea atributelor lor, precum şi a relaţiilor şi interacţiunilor dintre ele, textual în limbaj natural coerent, corect din punct de vedere sintactic şi semantic.

https://doi.org/10.33436/v30i1y202007

Aria de aplicabilitate este destul de largă, pornind de la adnotarea descriptivă automată a volumului tot mai mare de imagini postate de utilizatorii individuali pe reţelele de *social media* sau crearea de rezumate textuale descriptive pentru bazele de date de imagini pentru organizare şi/sau accesibilitate mai eficientă, asistarea persoanelor cu deficienţe vizuale majore prin descrierea textuală, combinată cu un sistem *text-to-speech* (TTS), a unor astfel de imagini, filme, sau chiar a mediului înconjurător în care se află sau se deplasează acestea, adnotarea cu descrieri ale celor observate în imagistica medicală de diferite tipuri pentru asistarea în punerea unor diagnostice şi/sau urmărirea unor evoluţii ş.a.m.d.

Înţelegerea conţinutului unei imagini depinde în mod esenţial de extragerea caracteristicilor specifice (*features*) din aceasta. Tehnicile de învăţare automată (*Machine Learning* – ML) utilizate în acest scop pot fi încadrate în două mari categorii: tehnici bazate pe metode tradiţionale şi, respectiv, tehnici noi bazate pe *Deep Learning*. Nu vom aborda în această lucrare aspecte legate de metodele şi tehnicile tradiţionale, cum sunt de exemplu cele utilizând şabloane (*templates*) de descrieri – în care sunt introduse în poziţii predefinite (*slot*-uri) denumirile obiectelor, atributele acestora sau relaţiile între ele identificate în imaginea analizată, sau cele bazate pe căutarea şi alegerea dintr-o mulţime predefinită a celei mai potrivite descrieri, asociate imaginii celei mai asemănătoare vizual cu cea analizată.

Problemele şi etapele principale în generarea automată de descrieri textuale asociate imaginilor cu metode şi tehnici bazate pe *Deep Learning* sunt: **a)** – extragerea informaţiei vizuale din imagine şi **b)** – transformarea acesteia într-un text adecvat şi semnificativ. În ultimii ani, odată cu evoluţiile spectaculoase din domeniul reţelelor neuronale adânci şi *Deep Learning* cu performanţe comparabile şi chiar superioare celor umane în domenii ca analiza de imagini şi traducerea automată, au fost înregistrate progrese absolut remarcabile şi în domeniul IC, calitatea descrierilor textuale generate fiind îmbunătăţită substanţial prin combinarea reţelelor neuronale convoluţionale (*Convolutional Neural Network* – CNN), folosite pentru obţinerea unor reprezentări vectoriale esenţializate ale caracteristicilor (*features*) din imagini, cu reţele neuronale recurente (*Recurrent Neural Network* - RNN), în particular de tip *Long Short-Term Memory* (LSTM), utilizate pentru traducerea acestor reprezentări în fraze în limbaj natural.

Tehnicile de IC bazate pe *Deep Learning* pot fi grupate în mai multe categorii [Hossain et al., 2019], după: modul de mapare a caracteristicilor – *feature mapping* – (în spaţiul vizual vs. într-un spaţiu multimodal); tipul învăţării (supervizată vs. nesupervizată); numărul de descrieri textuale şi aria lor de acoperire sau extinderea (descrieri „dense” vs. descrieri ale întregii scene); arhitectura sistemului (*encoder-decoder vs.* arhitecturi compoziţionale); modelul utilizat pentru partea de limbaj natural (RNN-LSTM vs. altele); alte abordări complementare utilizate (metode bazate pe atenţie vs. metode bazate pe concepte semantice vs. extinderea scenariilor generate vs. metode de generare a unor descrieri textuale stilizate).

Lucrarea de faţă îşi propune o prezentare succintă a tehnicilor şi metodelor bazate pe *Deep Learning* utilizate în domeniul IC, cu detalierea şi analizarea ca studiu de caz a uneia dintre cele mai performante dintre acestea, prima în care s-a utilizat şi un mecanism atenţional pe o arhitectură de tip *encoder-decoder*.

În continuare, lucrarea este structurată după cum urmează: Secţiunea 2 trece în revistă principalele aspecte privind noile metode şi tehnici bazate pe *Deep Learning* utilizate în generarea automată de descrieri textuale ale imaginilor. În secţiunea 3 este analizată ca studiu de caz una dintre metodele de referinţă în IC [Xu et al., 2015], utilizând o arhitectură de tip *encoder-decoder* combinată cu un mecanism de focalizare dinamică a atenţiei pe regiunile corespunzătoare relevante ale imaginii la generarea fiecărui cuvânt din secvenţa descriptivă de ieşire. Lucrarea se încheie cu secţiunea 4 în care sunt enunţate câteva concluzii şi comentarii.

2. Metode şi tehnici de *Image Captioning* utilizând *Deep Learning*

Problema tratată de IC se referă în esenţă la extragerea dintr-o imagine de intrare a unei reprezentări a caracteristicilor esenţiale şi generarea automată pe baza acestei reprezentări a unei descrieri textuale a conţinutului imaginii respective în limbaj natural. Aceasta se realizează în general ulterior unui proces corespunzător de învăţare automată (*machine learning* – ML).

După modul de mapare a caracteristicilor, metodele de IC pot fi grupate în metode cu mapare în spaţiul vizual sau într-un spaţiu multimodal. În cazul mapării în spaţiul vizual, caracteristicile imaginilor şi descrierile textuale corespunzătoare sunt transmise independent *decoder*-ului de limbaj la învăţare. În cazul spaţiului multimodal, la antrenare, sistemul învaţă un spaţiu comun obţinut din imagini şi descrierile asociate acestora, reprezentate în acest acelaşi spaţiu prin ieşirile generate de un *encoder* de imagini şi, respectiv, un *encoder* de limbaj, iar reprezentarea comună fiind transmisă *decoder*-ului.

Metodele de învăţare automată utilizate în IC pot fi supervizate sau nesupervizate. În cazul învăţării supervizate, datele de antrenare au ataşate etichetele de clasă care reprezintă ieşirea aşteptată, în timp ce, în cazul învăţării nesupervizate, datele sunt neetichetate. Dintre metodele de învăţare automată nesupervizată, cele mai folosite sunt *Generative Adversarial Networks* (GAN) şi învăţarea prin recompensă (*Reinforcement Learning* – RL). În această din urmă tehnică, învăţarea este realizată de un agent prin explorarea datelor pas cu pas, alegând câte o acţiune în fiecare stare şi primind „recompense" care evaluează acţiunea şi/sau noua stare, obiectivul fiind maximizarea recompensei cumulate pe termen lung. Reţelele de tip GAN învaţă caracteristici din date neetichetate aplicând un mecanism competitiv între două reţele, generatorul şi discriminatorul. Reţele neuronale bazate pe învăţare supervizată au fost folosite cu succes în domenii precum clasificarea imaginilor, detectarea de obiecte sau învăţarea atributelor, obţinând performanţe comparabile cu, sau chiar peste cele umane. Această experienţă a fost aplicată şi în domeniul generării de descrieri textuale asociate imaginilor. Arhitecturile utilizate pot fi: de tip *encoder-decoder*; compoziţionale; cu mecanisme bazate pe atenţie; bazate pe concepte semantice; cu metode de generare a unor descrieri textuale stilizate; cu extinderea scenariilor generate; cu generare de descrieri „dense".

Descrierile textuale ale imaginilor pot fi „dense" sau globale ale întregii scene. În cazul descrierii textuale dense sunt localizate regiunile importante ale imaginii şi se generează descrieri pentru fiecare dintre acestea. Paşii urmaţi sunt în general următorii: se generează propuneri de regiuni de interes (*regions of interest* – ROI) din imagini; o reţea neuronală convoluţională extrage caracteristicile din regiunile selectate; ieşirile generate sunt utilizate de un model al limbajului pentru a genera descrieri textuale pentru fiecare regiune. Descrierea globală a întregii imagini poate să fie subiectivă şi să nu ofere suficientă informaţie pentru înţelegerea acesteia. Descrierile asociate regiunilor de interes sunt mai obiective şi mai detaliate. Există desigur anumite dificultăţi şi provocări în generarea descrierilor dense. Printre acestea, menţionăm faptul că un obiect poate avea mai multe regiuni de interes care se suprapun, sau faptul că este dificil de recunoscut fiecare regiune pentru toate conceptele vizuale. În cazul generării de descrieri textuale corespunzătoare întregii imagini, metodele cele mai utilizate sunt: arhitecturi de tip *encoder-decoder*; arhitecturi compoziţionale; arhitecturi bazate pe semantică; arhitecturi bazate pe atenţie; descrieri textuale stilizate; extinderea scenariilor.

Cele mai multe modele se bazează pe arhitectura *encoder-decoder*, care este foarte flexibilă şi performantă. Această arhitectură a fost iniţial proiectată pentru traducerea automată a limbajului. În cazul IC putem vorbi de o traducere automată a unei imagini într-un text descriptiv corespunzător. Astfel, generararea automată a descrierilor textuale ale imaginilor necesită logic două modele: un model pentru extragerea caracteristicilor din imagini (*encoder*) şi un model al limbajului, care translatează caracteristicile şi obiectele furnizate de modelul de tratare a imaginilor într-o frază în limbaj natural (*decoder*). *Encoder*-ul este, cel mai adesea, o reţea neuronală convoluţională pentru extragerea caracteristicilor din imagini, detectarea obiectelor şi a relaţiilor dintre acestea. În lucrarea [Staniute & Sesok, 2019] este prezentată o analiză sistematică a literaturii de specialitate din domeniul IC publicate în ultimii patru ani. Sunt evidenţiate cele mai folosite tehnici şi cele mai importante provocări în domeniul IC. Au fost identificate cinci tipuri de reţele convoluţionale utilizate pentru extragerea de caracteristici din imagini, dintre care două majoritare: VGGNet şi ResNet. Acestea au fost raportate ca utilizate fiecare în câte 33 de articole din cele 78 analizate, primul fiind preferat pentru simplitatea modelului şi pentru puterea sa, iar al doilea pentru numărul mai mic de parametri şi eficienţa sa computaţională în comparaţie cu toate celelalte arhitecturi convoluţionale. *Decoder*-ul este în general o reţea neuronală recurentă

(*Recurrent Neural Network* - RNN), care permite reprezentarea contextului anterior, memorând secvenţa de cuvinte anterioare pentru a prezice următorul cuvânt din secvenţă. Arhitecturi particulare de RNN, precum *Long Short-Term Memory* (LSTM), rezolvă problema diminuării excesive a gradientului (*vanishing gradient*) într-un lanţ lung de transformări neliniare, învăţând astfel dependenţe pe termen lung în secvenţa de date. Utilizarea LSTM este considerată cea mai populară metodă în contextul IC. Cu toate că reţelele de tip LSTM posedă memorie în care se poate păstra informaţie despre secvenţa deja generată mai bine decât în cazul reţelele convoluţionale, este însă necesară actualizarea la fiecare pas, fiind un model consumator de memorie. De aceea au fost reconsiderate unele sisteme care utilizau combinaţia CNN-RNN, fiind propuse în locul acestora arhitecturi CNN-CNN. Reţelele convoluţionale pot fi antrenate mult mai rapid decât cele recurente, în principal din două motive: convoluţiile pot fi procesate în paralel, pe când recurenţele numai secvenţial; procesoarele grafice (GPU) pot fi folosite pentru a creşte viteza de antrenare a modelelor convoluţionale, în timp ce (deocamdată) nu există *hardware* care să accelereze antrenarea reţelelor recurente. Totuşi, rezultatele nu au fost cele dorite. Din cele 78 de sisteme considerate, 68 au revenit la LSTM. Primele sisteme de tip *encoder-decoder* au utilizat LSTM unidirecţionale pentru generarea textului, cuvântul următor fiind prezis pe baza contextului vizual şi cuvintelor generate anterior. În [Wang et al., 2016] este propus un model bidirecţional cu două reţele LSTM separate, care utilizează atât informaţia contextuală trecută cât şi cea viitoare pentru generarea descrierii textuale. Această metodă permite generarea unor descrieri mai bogate, atât semantic cât şi contextual. Viteza depinde nu numai de caracteristicile modelului, ci şi de dimensiunea vocabularului. În [Mishra & Liwicki, 2019] s-a încercat reducerea dicţionarului de cuvinte utilizate de la 10.000–40.000 de cuvinte la numai 248 de cuvinte. Deşi este o reducere drastică, rezultatele au fost încurajatoare.

Arhitecturile compoziţionale sunt alcătuite din mai multe blocuri funcţionale independente. Paşii urmaţi sunt: o reţea neuronală convoluţională extrage caracteristicile din imagine; conceptele vizuale (atribute) se obţin din caracteristici; se generează un set de descrieri textuale utilizând informaţiile din paşii anteriori; descrierile sunt reevaluate utilizând un model de similaritate multimodal, care selectează descrierile cele mai adecvate.

Utilizarea mecanismelor de atenţie vizuală a permis obţinerea de performanţe superioare. Acestea au la bază comportamentul uman natural: oamenii acordă o atenţie specială anumitor regiuni ale imaginii şi formulează explicaţii ale relaţiilor dintre obiectele ce apar în aceste regiuni. Astfel, mecanismele bazate pe atenţie au devenit foarte populare în *Deep Learning* pentru că permit depăşirea unor limitări prin focalizarea dinamică asupra diverselor regiuni din imaginea de intrare pe parcursul generării secvenţei de cuvinte la ieşire. Paşii urmaţi sunt: o reţea neuronală convoluţională extrage caracteristicile din întreaga imagine; modelul limbajului generează cuvinte şi fraze pornind de la aceste caracteristici; cu fiecare cuvânt generat, mecanismul de atenţie determină regiunea de interes din imagine corespunzătoare descrierii curente pentru generarea cuvântului următor; această regiune, împreună cu textul deja generat, sunt utilizate de modelul limbajului în generarea cuvântului următor. Majoritatea modelelor cu mecanisme atenţionale pentru generarea descrierilor textuale asociate imaginilor utilizează imaginea la fiecare pas, chiar şi atunci când se generează cuvinte de legătură, exprimând relaţii ş.a. care nu au un corespondent vizual în imagine. În [Lu et al., 2016] se propune introducerea unei „santinele” vizuale, care să stabilească dacă la un anumit moment modelul se bazează pe semnalul vizual sau pe modelul limbajului, implementând un model de atenţie adaptivă. Sistemul extinde modelul LSTM al *decoder*-ului cu o nouă poartă santinelă, care decide ponderea informaţiei extrase de *decoder* din imagine pentru a genera cuvântul următor. Mecanismele atenţionale pot fi de două tipuri: stocastic „*hard*” sau determinist „*soft*”. Pe de altă parte, abordările atenţionale pot fi de tip *top-down* sau *bottom-up*. [He et al., 2019] au confirmat recent că abordarea *top-down* este mai eficientă, rezultatele fiind similare cu cele din experimentele cu subiecţi umani. Sistemele de IC folosesc în general mecanisme atenţionale de tip *top-down*. În [Anderson et al., 2018] este descris un sistem care propune un mecanism atenţional ce combină metodele *top-down* şi *bottom-up* pentru stabilirea regiunilor de interes la nivel de obiecte sau alte zone ale imaginii. Mecanismul *bottom-up* utilizează o reţea convoluţională bazată pe regiuni (*region based*) optimizată – Faster R-CNN [Ren et al., 2015] – pentru a propune împărţirea imaginii în regiuni de interes şi generarea vectorilor de

caracteristici asociaţi fiecăreia dintre acestea, în timp ce mecanismul *top-down* utilizează LSTM pentru determinarea ponderilor caracteristicilor. În fine, nivelul limbajului este implementat la rândul său utilzând un al doilea LSTM.

În ultimul an, doi termeni apar în aproape fiecare articol din domeniul IC: extinderea scenariilor generate (*novel*) şi semantică [Hossain et al., 2019]. Aceşti termeni au apărut în legătură cu provocarea actuală cea mai importantă: generarea automată de descrieri textuale ale imaginilor care să fie cât mai apropiate de cele produse de subiecţi umani. Implementarea semantică facilitează o modalitate de a introduce sentimente în sistemul de generare automată de descrieri textuale. Extinderea scenariilor generate încearcă depăşirea limitărilor sistemelor actuale, datorate mai multor motive: în primul rând, modelele sunt antrenate pe seturi de date specifice, care nu acoperă toate scenariile posibile şi, de asemenea, dicţionarele de cuvinte sunt limitate ca număr de cuvinte şi combinaţii; în al doilea rând, modelele sunt în general gândite să execute o singură sarcină specifică, în timp de oamenii sunt capabili să efectueze mai multe sarcini simultan. Au fost propuse abordări pentru acest aspect al IC, care în principal propun un dialog cu utilizatorul, prin care sistemul pune întrebări specifice şi învaţă din răspunsuri. Această metodă este specifică domeniului *question-answering,* dar are un impact foarte însemnat pentru generarea automată de descrieri textuale asociate imaginilor. Cu toate că metodele de IC bazate pe *Deep Learning* au avut rezultate încurajatoare, acestea depind în mare măsură de seturile de date imagine-descrieri. Aceste metode pot genera descrieri numai pentru obiectele care au apărut în setul de date de antrenare şi, ca urmare, necesită seturi foarte mari de date. Metodele care extind contextul scenariilor pot genera descrieri ale obiectelor ce nu apar în setul de antrenare. În general, paşii urmaţi sunt: se antrenează separat un clasificator lexical şi un model al limbajului pe seturi de imagini şi texte necorelate; se antrenează un model de generare de descrieri textuale pe seturi de date imagine-descrieri; în final, ambele modele se combină pentru a fi antrenate împreună. Generarea de descrieri textuale pentru obiecte care nu au mai fost văzute este un subiect de cercetare recent. În lucrarea [Venugopalan et al., 2017] este introds *Novel Object Captioner* (NOC) pentru generarea de descrieri ale obiectelor noi utilizând surse externe pentru recunoaşterea acestor obiecte şi folosind cunoaştere semantică. Metodele bazate pe concepte semantice folosesc selectiv un set de propuneri de concepte semantice extrase din analiza imaginilor. Aceste concepte sunt apoi ataşate stărilor ascunse şi ieşirilor reţelei neuronale recurente. Etapele acestor metode sunt: o reţea neuronală convoluţională *encoder* extrage caracteristicile imaginilor şi conceptele semantice; aceste caracteristici sunt transmise ca intrări unui model de generare a limbajului; conceptele semantice sunt adăugate unor stări ascunse ale modelului limbajului; reţeaua responsabilă cu generarea limbajului produce descrierile textuale cu concepte semantice.

Majoritatea sistemelor de IC se bazează strict pe caracteristicile extrase din imagini, rezultând o descriere factuală a acestora. Descrierile stilizate pot fi însă mult mai expresive şi atractive decât o descriere factuală. Astfel de descrieri sunt apreciate în multe aplicaţii din lumea reală. De exemplu, oamenii încarcă o mulţime de fotografii în diverse sisteme *social media*. Aceste imagini necesită descrieri stilizate şi atractive. Aceste metode folosesc, în general, următorii paşi: o reţea convoluţională extrage caracteristicile imaginii; este folosit un corpus separat de text pentru a extrage diverse concepte stilistice (de ex. stil romantic, umoristic ş.a.) din datele de antrenare; din informaţiile de la paşii anteriori modelul limbajului poate genera descrieri textuale stilizate.

Cele mai multe sisteme s-au limitat la a folosi exclusiv seturile de date specifice pentru IC, restrângând astfel categoriile de obiecte pentru care pot fi generate descrieri textuale. Metoda propusă de [Venugopalan et al., 2017] poate utiliza suplimentar surse de date externe, precum imagini etichetate din seturi de date folosite la clasificarea/recunoaşterea de obiecte, sau informaţii semantice extrase din texte neadnotate.

3. *Image Captioning* utilizând o arhitectură *encoder-decoder* cu mecanism atenţional

În cele ce urmează prezentăm ca studiu de caz o metodă de descriere automată a imaginilor (*Image Captioning* – IC) având la bază ideile şi rezultatele prezentate de [Xu et al., 2015]. Este

vorba despre o abordare *Deep Learning* ***encoder-decoder***, îmbunătăţită însă prin introducerea unui **mecanism atenţional** care permite focalizarea dinamică pe o anumită regiune a reprezentării imaginii de intrare pentru predicţia următorului cuvânt din secvenţa descriptivă de ieşire.

Arhitectura de tip *encoder-decoder* a fost iniţial utilizată pentru învăţarea de tip secvenţă la secvenţă (*sequence to sequence*) cu reţele neuronale recurente utilizată în traducerea automată [Cho et al., 2014; Bahdanau et al., 2014; Sutskever et al., 2014]. Generarea descrierii unei imagini este privită analog unei „traduceri" a imaginii într-o frază descriptivă a acesteia. Astfel, la intrarea modelului (a *encoder*-ului) este aplicată imaginea $\boldsymbol{x}$, iar la ieşirea acestuia (a *decoder*-ului) este generată o secvenţă de cuvinte în limbaj natural asociate respectivei imagini: $\boldsymbol{y} = \{y_1, y_2, \dots, y_C\}$, $y_i \in \mathbf{R}^K$, unde K este dimensiunea vocabularului de cuvinte disponibile, iar C este lungimea secvenţei de cuvinte care descrie imaginea.

Metode anterioare (de ex. [Vinyals et. al, 2014]) s-au bazat în general pe o unică reprezentare statică comprimată, obţinută în stratul final complet conectat al unei reţele convoluţionale [Vrejoiu, 2019a] utilizate ca *encoder*, care codifică imaginea de intrare integrală şi este utilizată ca intrare de *decoder* doar o singură dată, la primul pas. Totuşi, astfel se poate pierde informaţie potenţial utilă pentru descrieri mai cuprinzătoare. Prin utilizarea unor reprezentări dintr-un nivel convoluţional precedent poate fi valorificată inclusiv astfel de informaţie, fiind necesară însă şi „direcţionarea" modelului către ceea ce este important în acestea la generarea fiecărui nou cuvânt la fiecare pas. Inspirată din funcţionarea sistemului vizual uman [Rensink, 2000; Corbetta & Shulman, 2001], focalizarea selectivă prin mecanisme atenţionale permite caracteristicilor semnificative dintr-o imagine să fie puse în evidenţă în mod dinamic contextual. Implementarea analizată de noi utilizează un mecanism atenţional determinist de tip „*soft*" (introdus de [Bahdanau et al., 2014]), antrenabil prin metode standard folosind retropropagarea erorii (*backpropagation*).

Partea de *encoder* este reprezentată de o **reţea neuronală convoluţională** (*Convolutional Neural Network* – CNN) fără straturile finale de agregare şi complet conectate, aceasta terminându-se cu o hartă de caracteristici (*feature map*) cu D „canale" (straturi de neuroni), de dimensiuni spaţiale $W \times H$, corespunzătoare aplicării ultimelor D filtre de convoluţie. Pentru fiecare imagine de intrare, este generat la ieşire un set de $L = W \times H$ vectori numiţi **vectori de adnotare** (*annotation vectors*), fiecare de dimensiune D şi reprezentând câte o porţiune din respectiva imagine, localizată în poziţia i: $a = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_L\}$, $\mathbf{a}_i \in \mathbf{R}^D$. Fiecare componentă $\mathbf{a}_i[j]$, ($j = 1 \div D$) a unui vector $\mathbf{a}_i$ reprezintă un indicator pentru detectarea (în mai mare sau mai mică măsură) a caracteristicii j în locaţia i ($i = 1 \div L$) din imagine. În figura 1 este reprezentată intuitiv structura ultimului nivel convoluţional considerat (harta de caracteristici – *feature map*) de dimensiuni $L \times D$, ($L = W \times H$) şi ieşirea *encoder*-ului (vectorii de adnotare $\mathbf{a}_i$), fiecare secţiune transversală corespunzând unei caracteristici j, ($j = 1 \div D$).

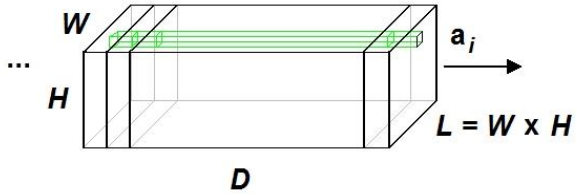

Figura 1. Ultimul nivel (convoluţional) şi ieşirea *encoder*-ului

Această ieşire a *encoder*-ului este utilizată mai departe de către *decoder*. Prin utilizarea ieşirii unui strat convoluţional anterior în locul unuia final complet conectat, se asigură posibilitatea *decoder*-ului de a se putea focaliza dinamic selectiv pe anumite porţiuni din imagine prin alegerea unui subset din toţi vectorii de adnotare.

Partea de *decoder* este reprezentată de o **reţea neuronală recurentă** (*Recurrent Neural Network* – RNN) de tip **LSTM** (*Long Short-Term Memory*) [Hochreiter & Schmidhuber, 1997; Vrejoiu, 2019b], care pentru fiecare reprezentare de imagine aplicată la intrare produce o descriere textuală. La fiecare pas de timp t este generat câte un cuvânt, $\mathbf{y}_t$ pe baza cuvântului generat la pasul anterior, $\mathbf{y}_{t-1}$, a stării ascunse (*hidden state*) obţinute la pasul anterior, $\mathbf{h}_{t-1}$, precum şi a unui **vector de context** vizual, $\hat{\mathbf{z}}_t$, care surprinde informaţia extrasă din imagine relevantă pentru generarea

noului cuvânt. Trebuie precizat faptul că pentru cuvinte este utilizată o reprezentare codificată numeric (*embedding*) $\mathbf{Ey}_{t-1}$, respectiv $\mathbf{Ey}_t$. Toate aceste trei mărimi vectoriale sunt aplicate ca intrare, modulate în fiecare caz prin ponderi antrenabile, la toate porţile celulei LSTM: de intrare, de uitare/ştergere şi de ieşire. Astfel, celula LSTM învaţă cum să pondereze fiecare componentă de intrare (*input gate*), învăţând totodată cum să moduleze respectiva contribuţie la memorie (*input modulator*). De asemenea sunt învăţate ponderi pentru ştergerea memoriei celulei LSTM (*forget gate*) şi ponderi pentru controlul modului în care această memorie trebuie să emită (*output gate*). Funcţionarea LSTM este descrisă la fiecare pas de timp *t* după cum urmează:

$$\mathbf{i}_t = \sigma(Wy_i\,\mathbf{Ey}_{t-1} + Wh_i\,\mathbf{h}_{t-1} + Wz_i\,\hat{\mathbf{z}}_t + \boldsymbol{b}_i) \quad (1)$$

$$\mathbf{f}_t = \sigma(Wy_f\,\mathbf{Ey}_{t-1} + Wh_f\,\mathbf{h}_{t-1} + Wz_f\,\hat{\mathbf{z}}_t + \boldsymbol{b}_f) \quad (2)$$

$$\mathbf{o}_t = \sigma(Wy_o\,\mathbf{Ey}_{t-1} + Wh_o\,\mathbf{h}_{t-1} + Wz_o\,\hat{\mathbf{z}}_t + \boldsymbol{b}_o) \quad (3)$$

$$\mathbf{g}_t = tanh(Wy_c\,\mathbf{Ey}_{t-1} + Wh_c\,\mathbf{h}_{t-1} + Wz_c\,\hat{\mathbf{z}}_t + \boldsymbol{b}_c) \quad (4)$$

unde: $Wy_{m,n}$, $Wh_{n,n}$ şi $Wz_{D,n}$ reprezintă matricile de ponderi antrenabile, iar **b** *bias*-urile globale asociate fiecărei porţi (*i, f, o*), respectiv memoriei celulei LSTM (*c*), iar:

$$\mathbf{c}_t = \mathbf{f}_t \cdot \mathbf{c}_{t-1} + \mathbf{i}_t \cdot \mathbf{g}_t \quad (5)$$

$$\mathbf{h}_t = \mathbf{o}_t \cdot \tanh(\mathbf{c}_t). \quad (6)$$

Vectorii $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$, $\mathbf{c}_t$, $\mathbf{h}_t \in \mathbf{R}^n$, de dimensiune *n* fiecare, reprezintă „semnalul" prin porţile de intrare, uitare/ştergere şi ieşire, respectiv starea memoriei şi starea ascunsă pentru reţeaua LSTM, iar $\mathbf{g}_t \in \mathbf{R}^n$ modulează poarta de intrare la momentul *t*. Vectorul $\hat{\mathbf{z}}_t \in \mathbf{R}^D$ este vectorul de context conţinând informaţia vizuală asociată unei locaţii particulare din intrare la fiecare moment *t*. $\mathbf{E} \in \mathbf{R}^{m \times K}$ reprezintă matricea de *embedding* corespunzătoare vocabularului de *K* cuvinte disponibile, *m* şi *n* reprezintă dimensiunea de *embedding* şi respectiv dimensiunea LSTM, iar *σ* şi *tanh* reprezintă funcţiile de activare de tip sigmoidă şi respectiv tangentă hiperbolică şi · semnifică înmulţirea element cu element.

Vectorul de context $\hat{\mathbf{z}}_t$ oferă o reprezentare dinamică a porţiunii relevante din imaginea de intrare pentru fiecare moment *t*. Acesta se calculează utilizându-se un mecanism *Φ* pe baza vectorilor de adnotare $\mathbf{a}_i$, $(i = 1 \div L)$ corespunzători caracteristicilor extrase pentru diferite locaţii *i* din imagine. Pentru fiecare locaţie, este generată o pondere pozitivă, subunitară, α_{ti}, care – în cazul mecanismului atenţional determinist „*soft*" utilizat – indică importanţa relativă ce trebuie acordată locaţiei *i* la momentul *t*. Ponderea α_{ti} a fiecărui vector de adnotare $\mathbf{a}_i$ este calculată utilizându-se un **model atenţional** f_{att} reprezentat de un perceptron multistrat (*Multi-Layer Perceptron* – MLP), pe baza stării ascunse anterioare a reţelei LSTM, $\mathbf{h}_{t-1}$. Această stare evoluează pe măsură ce reţeaua recurentă avansează în secvenţa de cuvinte de ieşire, locaţia unde reţeaua trebuie să caute mai departe depinzând astfel de secvenţa de cuvinte deja generate.

$$e_{ti} = f_{att}(\mathbf{a}_i, \mathbf{h}_{t-1}) = W_{att}\,ReLU(T_a\,\mathbf{a}_i + T_h\,\mathbf{h}_{t-1}), \quad (7)$$

unde: *ReLU* este funcţia de activare de tip *Rectified Linear Unit*, T_a şi T_h reprezintă ponderile a două straturi liniare separate asociate transformărilor aplicate fiecărui vector de adnotare $\mathbf{a}_i$ de dimensiune *D* şi, respectiv, stării ascunse anterioare a LSTM, $\mathbf{h}_{t-1}$, de dimensiune *n* pentru a fi combinate în spaţiul (dimensionalitatea) modelului atenţional, iar W_{att} reprezintă ponderile stratului final, cu câte 1 neuron pentru fiecare pereche de intrare ($\boldsymbol{a}_i$, $\boldsymbol{h}_{t-1}$), care furnizează la ieşire valoarea e_{ti}, ponderi care sunt comune pentru fiecare locaţie spaţială *i*. Ponderile α_{ti} sunt obţinute prin aplicarea funcţiei *softmax* peste cele *L* ieşiri e_{ti} ale modelului atenţional corespunzătoare aplicării celor *L* vectori de adnotare $\mathbf{a}_i$ (fiecare în conjuncţie cu starea anterioară $\mathbf{h}_{t-1}$), la intrarea acestuia:

$$\alpha_{ti} = \exp(e_{ti}) / \sum_{k \le L} \exp(e_{tk}). \quad (8)$$

Cu ponderile α_{ti} astfel calculate ($0 \le \alpha_{ti} \le 1$ şi $\sum_{i \le L} \alpha_{ti} = 1$), poate fi obţinut vectorul de context la pasul *t*, $\hat{\mathbf{z}}_t$, care „integrează" contribuţiile ponderate ale tuturor vectorilor de adnotare. Acesta nu mai depinde de locaţie, iar valorile celor *D* componente ale sale pun în evidenţă acele caracteristici/canale (reprezentând eventual obiecte) care sunt cele mai reprezentative contextual

pentru predicţia următorului cuvânt la pasul t, regăsite spaţial în locaţiile cu valori α_{ti} mari:

$$\hat{\mathbf{z}}_t = \Phi(\{\mathbf{a}_i\}, \{\alpha_{ti}\}), \quad (9)$$

unde Φ este o funcţie care întoarce un singur vector pe baza unui set de vectori de adnotare şi a ponderilor asociate acestora. În cazul nostru, pentru mecanismul atenţional determinist „soft”:

$$\hat{\mathbf{z}}_t = \sum_{i \le L} \alpha_{ti}\, \mathbf{a}_i. \quad (10)$$

Starea memoriei reţelei LSTM la momentul iniţial $t = 0$ şi respectiv starea ascunsă la ieşirea acesteia se iniţializează pe baza mediei vectorilor de adnotare $\mathbf{a}_i$, fiecare prin câte o reţea de tip perceptron multistrat (MLP) cu un strat ascuns, $f_{init,c}$ şi respectiv $f_{init,h}$, care realizează şi adaptarea la dimensionalitatea n a celulei LSTM:

$$\mathbf{c}_0 = f_{init,c}(1/\mathrm{L} \sum_{i \le L} \mathbf{a}_i), \quad (11)$$

$$\mathbf{h}_0 = f_{init,h}(1/\mathrm{L} \sum_{i \le L} \mathbf{a}_i). \quad (12)$$

Probabilitatea următorului cuvânt în secvenţa de ieşire a *decoder*-ului este obţinută printr-un strat de ieşire adânc [Pascanu et al., 2014] şi depinde de starea ascunsă la pasul de timp curent t ($\mathbf{h}_t$), de vectorul de context ($\hat{\mathbf{z}}_t$) şi de cuvântul generat la pasul anterior (ca *embedding*, $\mathbf{E}\mathbf{y}_{t-1}$):

$$p(\mathbf{y}_t \mid \mathbf{a}, \mathbf{y}_1^{t-1}) \sim \exp(\mathbf{L}_0 (\mathbf{E}\mathbf{y}_{t-1} + \mathbf{L}_h \mathbf{h}_t + \mathbf{L}_z \hat{\mathbf{z}}_t)), \quad (13)$$

unde parametrii $\mathbf{L}_0 \in \mathbf{R}^{K \times m}$, $\mathbf{L}_h \in \mathbf{R}^{m \times n}$, $\mathbf{L}_z \in \mathbf{R}^{m \times D}$ şi $\mathbf{E} \in \mathbf{R}^{m \times K}$ (matricea de *embedding*) sunt matrici de ponderi antrenabile, iniţializate aleator.

La antrenarea modelului atenţional determinist „*soft*” este aplicată o **regularizare dublu stocastică**, fiind încurajată şi condiţia $\sum_{t \le T} \alpha_{ti} \approx 1$ pe lângă cea impusă prin construcţie, $\sum_{i \le L} \alpha_{ti} = 1$, ceea ce asigură acordarea aceleiaşi atenţii fiecărei porţiuni a imaginii în cursul generării secvenţei de ieşire. Acest tip de regularizare permite obţinerea de performanţe îmbunătăţite atât cantitativ cât şi calitativ (descrieri mai bogate). În plus, modelul atenţional „*soft*” utilizat prezice la fiecare pas t şi un scalar β_t pe baza stării ascunse anterioare $\mathbf{h}_{t-1}$, astfel încât:

$$\hat{\mathbf{z}}_t = \Phi(\{\mathbf{a}_i\}, \{\alpha_{ti}\}) = \beta_t \sum_{i \le L} \alpha_{ti}\, \mathbf{a}_i, \quad (14)$$

unde: $\beta_t = \sigma(f_\beta(\mathbf{h}_{t-1}))$, funcţia f_β fiind reprezentată de un strat liniar. Prin introducerea acestui scalar, *decoder*-ul poate decide la fiecare pas dacă să pună accent mai cu seamă pe modelarea limbajului, sau pe context. Valori mai mari sunt asociate cu punerea mai pregnantă în evidenţă a obiectelor în imaginile de intrare.

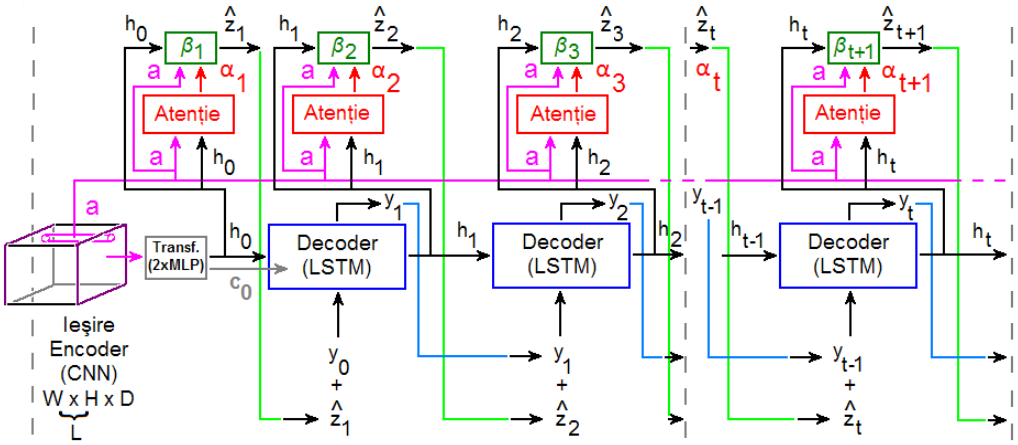

Figura 2. Structura funcţională desfăşurată a *decoder*-ului LSTM

Modelul atenţional este neted şi derivabil putând fi antrenat integral (*end-to-end*) cu *backpropagation* standard prin minimizarea următoarei funcţii *loss*:

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \sum_{i \le L} (1 - \sum_{t \le C} \alpha_{ti})^2. \quad (15)$$

Se utilizează metoda gradientului descendent stocastică (*stochastic gradient descent* – SGD) şi algoritmi cu rată de învăţare adaptivă şi optimizare RMSProp [Tieleman & Hinton, 2012] sau Adam (*Adaptive Moment Estimation*) [Kingma & Ba, 2014], precum şi regularizare de tip *dropout* [Srivastava et al., 2014].

Implementarea experimentală testată

Pentru experimentarea metodei prezentate mai sus s-a utilizat o implementare PyTorch [Vinodababu, 2018] dezvoltată pe baza celei iniţiale în Theano a [Xu et al., 2015] şi un sistem Dual Xeon 2,4GHz, cu 128GB DDR4 şi placă GPU nVidia GeForce GTX 1080 Ti 11GB. A fost utilizat setul de date *Microsoft Common Objects in COntext* (MSCOCO), cu 91 de categorii de obiecte în 328.000 de imagini adnotate cu un total 2.500.000 de descrieri. Au fost folosite efectiv 123.287 imagini, 82.783 de antrenare şi 40.504 de validare şi testare, cu câte 5 descrieri textuale fiecare.

Ca şi în cazul VGG şi pentru ResNet imaginile de intrare de rezoluţii variate sunt iniţial redimensionate astfel încât latura cea mai mică să aibă 256 pixeli, cu păstrarea factorului de formă (*aspect ratio*). Intrarea *encoder*-ului este dată de cele 3 canale RGB obţinute din decuparea regiunii centrale de 224 × 224 pixeli a acestei imagini. Valorile pixelilor sunt aduse în intervalul de valori reale [0, 1] şi apoi normalizate pe baza valorii medii şi a deviaţiei standard pentru fiecare canal (care se obţin utilizându-se tot setul de imagini de care se dispune). Astfel, PyTorch folosind convenţia NCHW (N = dimensiunea mini-*batch*-ului, C = numărul de canale, H = înălţimea şi W = lăţimea imaginii), la intrarea modelului sunt aplicaţi tensori de dimensiuni N, 3, 256, 256, conţinând valori reale din intervalul [0, 1].

Pentru partea de *encoder* care furnizează vectorii de adnotare $\mathbf{a}_i$ utilizaţi de *decoder* s-a folosit arhitectura convoluţională ResNet-101 [He et al., 2015], câştigătoare a competiţiei *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) 2015, mai performantă decât VGG-19 [Simonyan & Zisserman, 2014] utilizată de [Xu et al., 2015]. În experimente s-a utilizat practic implementarea PyTorch din modulul *torchvision* a modelului ResNet-101, preantrenat pe setul de date ImageNet 2014, fără ultimele două niveluri (de *average pooling* şi liniar final) deoarece se doreşte doar codificarea imaginilor şi nu clasificarea acestora. *Encoder*-ul poate fi de asemenea antrenat de la început, sau doar ajustat fin (*fine-tuned*) odată cu restul modelului. În cel din urmă caz, trebuie ajustate fin numai ponderile pentru ultimele trei niveluri convoluţionale „*bottleneck*" ale ResNet-101, deoarece în primele niveluri s-au învăţat practic numai caracteristici simple fundamentale (linii, muchii, curbe etc.), care nu mai trebuie afectate în vreun fel. În cazul ResNet-101, numărul de canale din ultimul strat al ultimului nivel convoluţional „*bottleneck*" (Conv 5), care reprezintă ieşirea *encoder*-ului, este $D = 2048$, la o rezoluţie spaţială $W \times H = 7 \times 7$, faţă de $D = 512$ şi $W \times H = 14 \times 14$ la VGG-19. Astfel, pentru a păstra o rezoluţie analoagă, se aplică o redimensionare adaptivă asupra acestei ieşiri la o rezoluţie impusă de 14 × 14. Pentru creşterea vitezei, performanţei şi stabilităţii, fiecare strat convoluţional din *encoder* este urmat de aplicarea unui strat de normalizare pe lot (*batch normalization*).

Şirurile descriptive asociate imaginilor (*captions*) constituie atât intrări cât şi ieşiri ţintă pentru *decoder*, fiecare cuvânt fiind utilizat pentru predicţia celui următor. Primul cuvânt generat este întotdeauna unul standard care marchează începutul descrierii, y_0 = <start>, iar finalul descrierii este marcat întotdeauna prin generarea unui cuvânt standard corespunzător, y_{C+1} = <end>, deoarece *decoder*-ul trebuie să înveţe să prezică sfârşitul unui şir descriptiv (*caption*) pentru a se putea opri procesul inferenţial de decodare odată prezis acesta: <start> y_1 y_2 ... y_C <end>. Totodată, deoarece în PyTorch şirurile descriptive sunt manipulate ca tensori de dimensiuni fixe, este necesar ca acestea să fie completate, când este necesar, până la o aceeaşi lungime maximă stabilită, C_{max}, cu un cuvânt standard dedicat, y_j = <pad>, ($j = C+2 \div C_{max}$), astfel: <start> y_1 y_2 ... y_C <end> <pad> <pad> ... <pad>.

Pe de altă parte, se crează o tabelă asociativă de indexare a cuvintelor din vocabularul utilizat, prin care fiecăruia dintre acestea, inclusiv celor trei standard: <start>, <end> şi <pad>, le

este asociat câte un indice în această tabelă. Utilizând acest index prin care este astfel codat asociativ fiecare cuvânt prin indicele respectiv, este posibilă şi asocierea *embedding*-ului corespunzător fiecăruia şi identificarea locului ocupat în scorurile cuvintelor prezise.

Astfel, descrierile textuale ale imaginilor (*captions*) sunt tensori de întregi de dimensiuni N_c, L_c, unde L_c este lungimea completată cu <pad>-uri până la sfârşit. Lungimea şirurilor descriptive (*caption lengths*) inclusiv cuvintele standard de început şi sfârşit (2 + lungimea efectivă C) este de asemenea menţinută într-un tensor de întregi de dimensiune N_c. Astfel este posibilă procesarea unei secvenţe doar până la această lungime fără a se mai pierde timp de calcul inutil pentru cuvintele de completare <pad> şi de asemenea pot fi construite grafuri dinamice cu PyTorch.

Modelul atenţional este unul simplu, compus doar din straturi liniare şi câteva activări. Astfel, două straturi liniare separate transformă fiecare din cei $14 \times 14 = 196$ vectori de adnotare $\mathbf{a}_i$, cu câte 2048 componente fiecare, obţinuţi la ieşirea *encoder*-ului şi respectiv starea ascunsă (*hidden state*) obţinută la ieşirea *decoder*-ului la momentul anterior, $\mathbf{h}_{t-1}$, de dimensiune 512, la o aceeaşi dimensiune atenţională, 512. Rezultatele acestor transformări liniare sunt adunate şi apoi rezultatul sumării este trecut printr-o activare de tip ReLU. În fine, un al treilea strat liniar transformă acest din urmă rezultat la dimensiunea 1 – e_{ti} din relaţia (7) – pentru fiecare pereche ($\mathbf{a}_i$, $\mathbf{h}_{t-1}$). În fine, asupra tuturor acestor L valori e_{ti}, de dimensiune 1, se aplică funcţia *softmax* pentru a fi generate ponderile α_{ti} – date de relaţia (8), pentru fiecare locaţie spaţială i la pasul t.

Decoder-ul primeşte la intrare ieşirea *encoder*-ului reprezentată de imaginea codificată în vectorii de adnotare $\mathbf{a}_i$, care este liniarizată la dimensiunile N, 196, 2048. Iniţializarea stării ascunse, $\mathbf{h}_0$ şi a memoriei celulei LSTM, $\mathbf{c}_0$, se realizează, prin două straturi liniare separate, pe baza mediei vectorilor de adnotare $\mathbf{a}_i$ furnizaţi la ieşirea *encoder*-ului – relaţiile (11) şi (12).

Deoarece implementarea necesită la învăţare durate proporţionale cu lungimea descrierii celei mai lungi la fiecare actualizare a ponderilor, antrenarea pe grupuri aleatorii de descrieri este evident ineficientă ca şi consum de timp. Pentru a se evita aceasta, s-a efectuat o preprocesare prin care s-a construit mai întâi o tabelă (un dicţionar) care asociază o anumită lungime subsetului corespunzător de descrieri de acea lungime. Apoi, la antrenare se eşantionează aleator o anumită astfel de lungime şi se compun mini loturi (*mini-batches*) pe baza respectivei lungimi la fiecare pas, ceea ce îmbunătăţeşte semnificativ viteza de convergenţă fără a afecta negativ performanţa. Cele N imagini şi descrierile (*captions*) asociate din fiecare lot (*batch*) sunt ordonate descrescător pe baza lungimii descrierilor respective, astfel încât să poată fi efectuate procesări numai pentru paşii de timp utili, respectiv să nu fie procesate şi eventualele cuvinte de completare <pad>. Astfel, se poate itera la fiecare pas de timp numai pentru lungimea efectivă considerată la momentul respectiv, ceea ce determină dimensiunea efectivă N_t a lotului la acel pas. Ordonarea permite ca primele N_t imagini la fiecare pas să fie aliniate cu ieşirile de la pasul anterior (primele N_t imagini sunt procesate utilizându-se primele N_t ieşiri de la pasul anterior). Această iteraţie este efectuată explicit pas cu pas, individual, într-o buclă *for* aplicându-se mecanismul atenţional între fiecare doi paşi consecutivi de decodare.

Ponderile α_{ti} şi codificarea ponderată atenţional la fiecare pas de timp sunt calculate de reţeaua atenţională. Vectorul de context $\hat{\mathbf{z}}_t$ se obţine ca sumă ponderată cu coeficienţii α_{ti} a vectorilor de adnotare $\mathbf{a}_i$, multiplicată apoi cu scalarul β_t obţinut prin aplicarea unei transformări liniare pe starea ascunsă anterioară a *decoder*-ului urmată de activare printr-o funcţie de tip sigmoid(ă) – relaţia (14). Această codificare ponderată atenţional $\hat{\mathbf{z}}_t$ este concatenată cu *embedding*-ul cuvântului anterior (<start> la început) $\mathbf{E}\mathbf{y}_{t-1}$, formând intrarea de dimensiune 2048 + 512 = 2560 a LSTM, pe baza căreia este generată ca ieşire noua stare ascunsă $\mathbf{h}_t$ de dimensiune 512. Un strat liniar transformă această nouă stare ascunsă în scoruri de predicţie asociate prin indice fiecărui cuvânt din vocabularul indexat disponibil, de dimensiune 9490, scoruri care sunt memorate. De asemenea sunt memorate şi ponderile atenţionale α_{ti} la fiecare pas t spre a putea fi utilizate ulterior pentru vizualizarea pe imagini a regiunilor de focalizare dinamică a atenţiei la generarea fiecărui cuvânt $\mathbf{y}_t$ din secvenţa de ieşire.

La antrenare, după fiecare epocă, poate fi salvat un fişier *checkpoint* cu valorile parametrilor modelului obţinute până la momentul respectiv. Astfel, antrenarea poate fi reluată de la oricare

stadiu anterior al învăţării cu parametrii modelului stocaţi într-un fişier *checkpoint* corespunzător. La sfârşitul fiecărei epoci de antrenare se execută o validare. Funcţia obiectiv (*loss*) utilizată în cazul nostru de generare a unor secvenţe de cuvinte este de tip *cross entropy*. Aceasta efectuează operaţiile necesare pentru calcularea *softmax* şi *log* pe baza scorurilor brute furnizate de stratul final al *decoder*-ului. La calcularea *loss*-ului, la fiecare pas de timp sunt ignorate regiunile completate cu <pad>. În plus, aşa cum am menţionat mai sus, este utilizată o „regularizare dublu stocastică”, implicând şi o a doua funcţie *loss*, pentru modelul atenţional. Astfel, se impune şi ca suma tuturor ponderilor atenţionale α_{ti} ($0 \le \alpha_{ti} \le 1$ şi $\sum_{i \le L} \alpha_{ti} = 1$) pentru fiecare locaţie i să fie 1 pe totalitatea paşilor de timp T ($\sum_{t \le T} \alpha_{ti} \approx 1$), ceea ce impune modelului să se focalizeze pe fiecare pixel în cursul generării întregii secvenţe descriptive de ieşire. Prin urmare se caută şi minimizarea diferenţei între 1 şi suma ponderilor asociate pixelilor de-a lungul tuturor paşilor de timp – relaţia (15).

Pentru evaluarea performanţei pe setul de validare şi alegerea modelului (dar şi pentru testare) se utilizează metrica automatizată BLEU (*BiLingual Evaluation Understudy*) 1 – 4. Aceasta a constituit multă vreme un standard de evaluare în raportările din literatura de specialitate în domeniul generării automate de fraze la traducerea automată dintr-un limbaj natural în altul, dar şi de descrieri textuale asociate imaginilor. Scorul BLEU indică gradul de potrivire – considerând secvenţe de 1 până la 4 cuvinte – cu un text de referinţă corespunzător creat de om şi poate avea valori reale în intervalul [0, 1], unde valoarea 1 indică potrivirea perfectă, iar valoarea 0 totală nepotrivire. Algoritmul BLEU tinde prin construcţie să acorde scorurile cele mai mari textelor generate cu lungimi apropiate de cele ale textelor de referinţă. În cazul nostru, pentru fiecare descriere generată, sunt utilizate toate cele *N_c* descrieri disponibile ca referinţă pentru respectiva imagine. După un anumit număr de epoci se constată apariţia unei fracturi de corelare între continuarea încercărilor de minimizare a funcţei *loss* de tip probabilitate logaritmică şi evoluţia scorului BLEU. Astfel, antrenarea trebuie oprită anticipat (*early stopping*) atunci când scorul BLEU începe să se degradeze chiar dacă *loss*-ul continuă să scadă. Această oprire anticipată a învăţării pe baza scorului BLEU este singura strategie de regularizare utilizată înafară de *dropout*. A fost utilizată implementarea BLEU din modulul de resurse şi instrumente Python pentru limbaj natural NLTK (*Natural Language Tool Kit*). Deoarece există destule critici aduse scorului BLEU, constatându-se că nu este corelat întotdeauna bine cu modul de gândire uman, [Xu et al., 2015] au utilizat de asemenea ca metrică alternativă scorul METEOR [Denkowski & Lavie, 2014], mult mai potrivit din acest punct de vedere, dar acesta nu a fost folosit în implementarea experimentată. Ca strategie de antrenare este recomandată abordarea învăţării în etape. Astfel, se porneşte mai întâi cu antrenarea *decoder*-ului fără ajustarea fină (*fine-tuning*) a *encoder*-ului, cu o dimensiune a loturilor (*batch size*) mai mare, pentru un număr de epoci M, observându-se evoluţia scorului BLEU-4. Pentru optimizare se utilizează *optimizer*-ul Adam cu rată de învăţare (*learning rate*) iniţială de $4e^{-4}$. Se alege un *checkpoint* aferent unei epoci M_1 anterioare M, pentru care scorul BLEU-4 a atins o valoare de circa 22,5-23,5. Pornind de la *checkpoint*-ul corespunzător acestei epoci M_1, se continuă antrenarea *encoder*-ului permiţând şi *fine-tuning*, alegând însă de această dată o dimensiune mai mică a loturilor, deoarece dimensiunea modelului include acum şi gradienţii *encoder*-ului. Se observă că la câteva epoci odată scorul BLEU-4 creşte cu peste o unitate.

Atât la antrenare cât şi la validare este aplicată tehnica *Teacher Forcing*, care constă în furnizarea ca intrare la fiecare nou pas de decodare a cuvântului real din descrierea curentă asociată, în locul cuvântului care tocmai a fost generat la pasul anterior. Utilizarea acestei tehnici este comună în timpul antrenării pentru accelerarea convergenţei modelului. Totuşi, acesta ar putea să devină astfel prea dependent de indicarea răspunsului corect şi să ajungă să manifeste o oarecare instabilitate în practică. Ideal ar fi să se utilizeze *Teacher Forcing* la antrenare doar uneori, pe bază probabilistică (*scheduled sampling*). Pe de altă parte, în timpul validării ar trebui reproduse în cât mai mare măsură condiţiile reale pentru inferenţă, deoarece cu *Teacher Forcing* şi la validare, scorul BLEU calculat pentru descrierile astfel generate nu reflectă de fapt performanţa reală a modelului. Practic, oprirea anticipată pe baza scorului BLEU ar trebui să fie utilizată doar dacă pentru validare este implementată inferenţa utilizând cuvântul anterior efectiv generat de model.

Mai trebuie menţionat faptul că, pentru ajustarea fină a parametrilor modelului la *Transfer Learning*, este recomandat să se utilizeze o rată de învăţare substanţial mai mică decât cea utilizată la antrenarea modelului original, pentru ca modificările parametrilor deja optimizaţi ai acestuia să nu fie prea grosiere şi mai mult să strice. Optimizarea *encoder*-ului se face de asemenea cu Adam, dar cu o rată de învăţare (*learning rate*) de $1e^{-4}$, ceea ce reprezintă o zecime din valoarea implicită pentru acest *optimizer*. Antrenarea unei epoci cu ajustarea fină a parametrilor/ponderilor arhitecturii (*fine-tuning*) poate dura în medie de circa 2,5-3 ori mai mult decât fără ajustare fină.

Prin vizualizarea efectului mapării ponderilor atenţionale α_{ti} pe diferite regiuni ale imaginilor de intrare la generarea fiecărui nou cuvânt în şirul descriptiv de ieşire la momentul *t*, se poate observa focalizarea dinamică selectivă a atenţiei şi faptul că aceasta corespunde în mare măsura intuiţiei umane, adăugându-se astfel un nivel suplimentar de interpretabilitate a ieşirii modelului. Se poate observa faptul că mecanismul atenţional „*soft*" permite focalizarea dinamică inclusiv pe regiuni semnificative care nu reprezintă obiecte.

4. Concluzii

Înţelegerea scenelor reprezintă încă un obiectiv ambiţios în domeniul vederii artificiale. În acest context se încadrează şi *Image Captioning* (IC) – generarea automată a unei descrieri textuale asociate unei imagini – cu identificarea obiectelor care apar în imagini şi a atributelor lor, precum şi a relaţiilor şi interacţiunilor dintre ele şi exprimarea tuturor acestora în text în limbaj natural semnificativ şi coerent.

Exemple de aplicaţii sunt: adnotarea automată cu texte descriptive a volumului tot mai mare de imagini postate pe reţelele de *social-media*; crearea de rezumate asociate imaginilor din diferite baze de date pentru organizare şi/sau accesibilitate mai eficientă; asistarea persoanelor cu deficienţe vizuale majore prin descrierea textuală, combinată cu un sistem *text-to-speech* (TTS), a unor astfel de imagini, filme, sau chiar a mediului înconjurător în care se află sau se deplasează acestea; adnotarea cu descrieri ale celor observate în imagistica medicală de diferite tipuri pentru asistarea în punerea unor diagnostice şi/sau urmărirea unor evoluţii ş.a.m.d.

Odată cu evoluţiile recente spectaculoase din domeniul reţelelor neuronale adânci şi *Deep Learning*, cu performanţe comparabile şi chiar peste cele umane în domenii ca analiza de imagini şi traducerea automată, au fost înregistrate progrese absolut remarcabile şi în domeniul IC, calitatea textelor descriptive generate automat pentru imagini fiind îmbunătăţită substanţial. În general, mai întâi se obţin reprezentări vectoriale esenţializate ale caracteristicilor (*features*) din imagini folosindu-se reţele neuronale convoluţionale, apoi aceste reprezentări sunt decodate în fraze în limbaj natural utilizându-se reţele neuronale recurente, în particular de tip LSTM.

Provocările actuale cele mai importante în domeniul IC se referă la generarea automată de descrieri textuale ale imaginilor cât mai apropiate de cele produse de subiecţi umani, cât mai nuanţate şi bogate în informaţii utile şi care să se refere inclusiv la obiecte şi situaţii nemaiîntâlnite anterior la învăţare. Există deja abordări şi soluţii cu rezultate promiţătoare inclusiv pe aceste problematici. În lucrarea de faţă au fost prezentate sintetic principalele tehnici şi metode bazate pe *Deep Learning* utilizate în ultimii ani în domeniul IC, cu detalierea şi analizarea ca studiu de caz a uneia dintre cele mai performante dintre acestea, prima în care s-a utilizat (şi) un mecanism atenţional pe o arhitectură de tip *encoder-decoder* şi pentru care s-a experimentat o implementare PyTorch.

Menţiuni

Prezenta lucrare are la bază parte din activităţile şi rezultatele etapei a II-a a proiectului PN 1937-0601 derulat la ICI Bucureşti (2019) în cadrul Programului naţional nucleu „SMARTIC" finanţat de Ministerul Cercetării şi Inovării.

BIBLIOGRAFIE

1. Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S. & Zhang, L. (2018). Bottom-up and top-down attention for image captioning and visual question answering. *CVPR2018*.
2. Bahdanau, D.; Cho, K. & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473*.
3. Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bougares, F.; Schwenk, H. & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, Oct. 2014.
4. Corbetta, M. & Shulman, G. L. (2002). Control of goaldirected and stimulus-driven attention in the brain. *Nature reviews neuroscience, 3*(3):201–215.
5. Denkowski, M. & Lavie, A. (2014). Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.
6. He, K.; Zhang, X.; Ren, S. & Sun, J. (2015). Deep Residual Learning for Image Recognition. Tech Report. *arXiv:1512.03385*.
7. He, S.; Tavakoli, H. R.; Borji, A. & Pugeault, N. (2019). A synchronized multi modal attention-caption dataset and analysis. *arXiv:1903.02499*.
8. Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8):1735–1780.
9. Hossain, M. D.; Sohel, F.; Shiratuddin, M. F. & Laga, H. (2019). A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CSUR), 51*(6):118.
10. Kingma, D. P. & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv:1412.6980*.
11. Lu. J.; Xiong, C.; Parikh, D. & Socher, R. (2016). Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning. *IEEE Conference on Computer Vision and Pattern Recognition – CVPR 2017*.
12. Mishra, A. & Liwicki, M. (2019). Using deep object features for image descriptions. *arXiv:1902.09969*.
13. Pascanu, R.; Gulcehre, C.; Cho, K. & Bengio, Y. (2014). How to construct deep recurrent neural networks. In *ICLR 2014*.
14. Ren, S.; He, K.; Girshick, R. & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv:1506.01497*.
15. Rensink, R. A. (2000). The dynamic representation of scenes. *Visual cognition, 7*(1-3):17–42.
16. Simonyan, K. & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR, abs/1409.1556*.
17. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I. & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *JMLR 15*.
18. Staniute, R. & Sesok, D. (2019). A Systematic Literature Review on Image Captioning. *Applied Sciences, 9*(10):2024.
19. Sutskever, I.; Vinyals, O. & Le, Quoc V.V. (2014). Sequence to sequence learning with neural networks. In *NIPS 2014*:3104–3112.
20. Tieleman, T. & Hinton, G. (2012). Lecture 6.5 - RMSprop. *Technical report*.
21. Venugopalan, S.; Hendricks, L. A.; Rohrbach, M.; Mooney, R. J.; Darrell, T. & Saenko, K. (2017). Captioning Images with Diverse Objects. *Proc. of IEEE Conference on Computer Vision and Pattern Recognition – CVPR 2017*:1170-1178.

22. Vinodababu, S. (2018). PyTorch Tutorial to Image Captioning. *https://github.com/sgrvinod/a-PyTorch-Tutorial-to-Image-Captioning* (accesat 2019)
23. Vinyals, O.; Toshev, A.; Bengio, S. and Erhan, D. (2014). Show and tell: A neural image caption generator. *arXiv:1411.4555*.
24. Vrejoiu, M. H. (2019a). Reţele neuronale convoluţionale, Big Data şi Deep Learning în analiza automată de imagini. *Revista Română de Informatică şi Automatică, 29*(1):91-114.
25. Vrejoiu, M. H. (2019b). Neural Networks and Deep Learning in Cyber Security. *Romanian Cyber Security Journal, 1*(1):69-86.
26. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhutdinov, R.; Zemel, R. & Bengio, Y. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In Proceedings of the 32nd International Conference on Machine Learning, *Proc. of ML Research, 37*:2048-2057. (Preprint *arXiv:1502.03044*).
27. Wang, C.; Yang, H.; Bartz, C. & Meinel, C. (2016). Image captioning with deep bidirectional LSTMs. *Proc. of ACM on Multimedia Conference – ACM 2016*:988–997.

Anca Mihaela HOTĂRAN este cercetător ştiinţific gradul III în Departamentul „Sisteme inteligente distribuite intensive ca date” din ICI Bucureşti. Domeniile şi subiectele sale principale de expertiză şi interes cuprind: reprezentarea cunoştinţelor (ontologii, semantică), analiza avansată a datelor şi inteligenţa artificială (sisteme expert, învăţare automată).

Anca Mihaela HOTĂRAN is a 3rd grade scientific researcher in the Department of “Distributed Data Intensive Intelligent Systems” at ICI Bucharest. Her main areas and topics of expertise and interest include: Knowledge Representation (ontologies, semantics), advanced Data Analysis and Artificial Intelligence (Expert Systems, Machine Learning).

Mihnea Horia VREJOIU este cercetător ştiinţific gradul III în Departamentul „Sisteme inteligente distribuite intensive ca date” din ICI Bucureşti. Domeniile şi subiectele sale principale de expertiză şi interes cuprind: vedere artificială (prelucrare şi analiză de imagini, recunoaşterea formelor, recunoaşterea optică de caractere – OCR, recunoaşterea numerelor de înmatriculare – LPR) şi învăţare automată (clasificatoare, memorii asociative).

Mihnea Horia VREJOIU is scientific researcher 3rd grade in the “Distributed Data Intensive Intelligent Systems” Department at ICI Bucharest. His main areas and topics of expertise and interest cover: Artificial Vision (Image Processing and Analysis, Pattern Recognition, Optical Character Recognition – OCR, License Plate Recognition – LPR) and Machine Learning (classifiers, associative memories).