Nadaljevalno učenje s superpozicijo v transformerjih

Marko Zeman, Jana Faganeli Pucer, Igor Kononenko, Zoran Bosnić
Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, Ljubljana, Slovenija
{marko.zeman, jana.faganeli, igor.kononenko, zoran.bosnic}@fri.uni-lj.si

Izvleček

V mnogih aplikacijah strojnega učenja se novi podatki nenehno zbirajo, npr. v zdravstvenem varstvu, za vremenske napovedi itd. Raziskovalci si pogosto želijo sistem, ki bi omogočal nadaljevalno učenje novih informacij. To je izjemnega pomeni tudi v primeru, ko vseh podatkov ni mogoče shranjevati v nedogled. Največji izziv pri nadaljevalnem strojnem učenju je težnja nevronskih modelov, da po določenem času pozabijo prej naučene informacije. Da bi zmanjšali pozabljanje modela, naša metoda nadaljevalnega učenja uporablja superpozicijo z binarnimi konteksti, ki zavzemajo zanemarljiv dodaten pomnilnik. Osredotočamo se na nevronske mreže v obliki transformerjev, pri čemer smo naš pristop primerjali z več vidnimi metodami nadaljevalnega učenja na nizu klasifikacijskih nalog obdelave naravnega jezika. V povprečju smo dosegli najboljše rezultate: 4,6% izboljšavo pri ploščini pod krivuljo ROC (angl. AUROC - area under the receiver operating characteristic) in 3,0% izboljšavo pri ploščini pod krivuljo PRC (angl. AUPRC - area under the precision-recall curve).

Ključne besede: globoko učenje, nadaljevalno učenje, strojno učenje, superpozicija, transformer, klasifikacija besedil

Continual learning with superposition in transformers

Abstract

In many machine learning applications, new data is continuously collected, e.g., in healthcare, for weather forecasting etc. Researchers often want a system that allows for continuous learning of new information. This is extremely important even in the case when not all data can be stored indefinitely. The biggest challenge in continual machine learning is the tendency of neural models to forget previously learned information after a certain time. To reduce model forgetting, our continual learning method uses superposition with binary contexts, which require negligible additional memory. We focus on transformer-based neural networks, comparing our approach with several prominent continual learning methods on a set of natural language processing classification tasks. On average, we achieved the best results: 4.6% and 3.0% boost in AUROC (area under the receiver operating characteristic) and AUPRC (area under the precision-recall curve), respectively.

Keywords: deep learning, continual learning, machine learning, superposition, transformer, text classification

1 UVOD

Ljudje imamo naravno sposobnost nenehnega pridobivanja in razvijanja znanja ter veščin. Ta sposobnost, imenovana nadaljevalno ali vseživljenjsko učenje, je mogoča zaradi našega dolgoročnega spomina in enostavnega prenosa znanja med podobnimi nalogami. Vendar pa nadaljevalno učenje še vedno predstavlja velik izziv pri strojnem učenju. Trenutno najbolj priljubljeni modeli strojnega učenja so globoke nevronske mreže, ki pogosto trpijo zaradi obsežnega pozabljanja modela [1, 7]. To pomeni, da modeli ponavadi pozabljajo predhodno naučene informacije in si zapomnijo le nedavno opazovane vzorce [10]. V naši problemski domeni novi podatki postanejo na voljo v obliki novih nalog, in sicer na zaporedni način.

Da bi se soočili z omenjenimi izzivi, raziskovalci poskušajo najti načine za ublažitev pozabljanja modelov in prilagoditev modela za novo nalogo tako pomnilniško kot hitrostno učinkovito.

Najenostavnejša rešitev je naučiti različne naloge v ločenih globokih nevronskih mrežah [8]. Vendar pa je v tem primeru glavna težava velika poraba

pomnilnika, saj se število globokih nevronskih mrež povečuje linearno s številom nalog.

Številni obstoječi pristopi zahtevajo veliko pomnilnika ali arhitekturne spremembe, ki jih je pogosto težko izvesti. Eden najpreprostejših in najbolj pomnilniško učinkovitih pristopov uporablja superpozicijo, ki omogoči učenje več nalog v eni nevronski mreži z omejenim pozabljanjem in majhno porabo pomnilnika na nalogo [2]. Superpozicija se je že izkazala za koristen pristop v polno povezanih in konvolucijskih nevronskih mrežah v domeni računalniškega vida [2, 12].

Predlagamo novo rešitev, kjer uporabljamo superpozicijo v transformerjih [11], ki dosegajo najboljše rezultate na področju obdelave naravnega jezika (ONJ). V strojnem učenju transformer predstavlja specifično obliko nevronske mreže, ki poskuša razumeti povezave med zaporednimi entitetami, npr. besedami v stavkih. Transformerji s pomočjo mehanizmov pozornosti odkrivajo, kako oddaljeni elementi v nekem zaporedju vplivajo drug na drugega [11]. Pri transformerjih pridobimo na zmogljivosti ob ohranjanju pozitivnih učinkov superpozicije v polno povezanih mrežah.

2 SORODNA DELA

Ponavljalne metode temeljijo na shranjevanju dela učnih primerov iz prejšnjih nalog, ki se nato med učenjem modela ponovno uporabijo. Lopez-Paz in Ranzato [4] sta razvila pristop *Gradient Episodic Memory* (GEM). Ta zmanjša pozabljanje, saj omogoča koristen prenos znanja iz prejšnje naloge z malo dodatnega pomnilnika in preprečuje, da bi vrednost funkcije izgube iz preteklih nalog naraščala. Ker se učni primeri hranijo za vsako nalogo in se občasno ponavljajo, se računske in pomnilniške zahteve povečujejo sorazmerno s številom naučenih nalog.

Arhitekturne metode zmanjšujejo pozabljanje z uporabo sprememb v arhitekturi mreže in uvedbo parametrov, specifičnih za nalogo. Običajno večji del mreže ostane fiksiran, manjši del pa se prilagodi na novo nalogo [6, 14].

Regularizacijske metode se zanašajo na en sam model in manjšajo pozabo z uvedbo omejitev za posodobitev uteži nevronske mreže. Kirkpatrick in sod. [3] so predlagali metodo EWC (*angl.* Elastic Weight Consolidation), ki kaznuje razliko med starimi in novimi parametri naloge. Natančneje, EWC zmanjša pozabljanje z uravnavanjem funkcije izgube, kar upočasni spreminjanje parametrov, pomembnih za predhodne naloge. Poleg tega so Schwarz in sod. [9] predlagali spremembo, imenovano sprotni EWC (*angl.* Online EWC), ki ne presega linearne rasti računskih zahtev. Poleg tega so Zenke in sod. [13] predlagali ublažitev pozabljanja tako, da posameznim sinapsam (tj. parametrom) omogočijo, da ocenijo svojo pomembnost. Podobno kot [3], ta pristop kaznuje spremembe najpomembnejših sinaps, tako da se lahko nove naloge naučijo z minimalnim pozabljanjem starih.

Superpozicija [2] je drugačna oblika metod za nadaljevalno učenje, ki jo podrobneje predstavljamo v razdelku 3.

3 SUPERPOZICIJA

Pri globokem učenju superpozicijski pristop omogoča učenje več nalog v eni sami nevronski mreži z minimalnim prepletanjem med nalogami. Glavni navdih prihaja iz dela Cheunga in sod. (PSP, [2]), kjer avtorji predstavljajo način za izkoriščanje odvečnih parametrov, da se naučijo več nalog v eni mreži, hkrati pa zmanjšajo pozabljanje modela.

Splošna ideja metode superpozicije je, da se N različnih nalog uči zaporedoma z uporabo algoritma vzratnega razširjenja napake v eni sami mreži z L nivoji. Matrike uteži (parametrov, ki jih je mogoče naučiti) so označene z W_1, W_2, . . . , W_{L-1} in se spreminjajo skozi učenje vseh nalog. Za omogočanje uporabe superpozicijie uporabljamo strukturo, imenovano *kontekst*, ki je predstavljena z množico binarnih vektorjev. Kontekst se najprej uporabi med učenjem nalog in se kasneje uporabi za obnovitev ustreznih uteži mreže za specifično nalogo.

Konteksti: V polno povezanih nevronskih mrežah so konteksti predstavljeni v obliki kontekstnih matrik, ki so kvadratne in diagonalne. Omenjene kontekstne matrike služijo za prehod med nalogami, in sicer se množijo z matrikami uteži. Matrike uteži se nenehno spreminjajo glede na fiksne kontekstne matrike. Konteksti delujejo le kot ključ za odklepanje predhodno naučenih nalog in se med učenjem ne spreminjajo. Vse kontekstne matrike vključujejo na diagonali samo elemente, ki so naključno izbrani med {−1, 1} (kot je predlagano v [2]), ostali elementi pa so enaki 0.

Učenje: Posamezno nalogo učimo, dokler ni dosežena želena točnost na validacijski množici po-

datkov. Nato se matrike uteži posodobijo z uporabo kontekstov. Preko vseh nivojev mreže izvedemo množenje matrik uteži s kontekstnimi matrikami. Ta postopek posodobitve uteži se ponovi za vsako novo nalogo z ustreznim naborom kontekstnih matrik.

Testiranje: Ko model naučimo vseh N nalog, lahko pridobimo ustrezne uteži modela za določeno nalogo z rahlo izgubo predhodnega znanja. Tudi tokrat je posodobitev uteži potrebno izvesti nad vsemi matrikami uteži. Na primer, če želimo ekstrahirati primerne uteži za tretjo nalogo, moramo trenutne uteži pomnožiti z inverznimi kontekstnimi matrikami od predzadnje naloge (po zadnji nalogi ne množimo s konteksti) do tretje naloge. Konteksti za prvo in drugo nalogo v tem primeru niso pomembni. Uporaba takega načina množenja zagotavlja pridobitev mrežnih uteži, ki so primerne za določeno nalogo, ob tem pa rahlo izgubimo na zmogljivosti modela.

3.1 Superpozicija v transformerjih

V transformerjih je naš kontekst v obliki kontekstnih vektorjev, ki so označeni s $C_1, C_2, \ldots, C_{L-1}$. Število kontekstnih vektorjev je enako številu matrik uteži za posamezno nalogo. Vendar ima vsaka naloga, razen zadnje, svoj nabor $L - 1$ kontekstov, kar pomeni, da imamo skupaj $(N - 1)$ $(L - 1)$ kontekstnih vektorjev. Konteksti se uporabljajo za transformacijo matrik uteži W_i. Pred uporabo med nalogami se kontekstni vektor predhodno pretvori v kontekstno matriko, kjer se vrednosti iz vektorja kopirajo v kontekstne matrike.

Razlikujemo dve vrsti matrik: (1) *polne* kontekstne matrike in (2) *redke* kontekstne matrike. Polne matrike vsebujejo le elemente iz množice {−1, 1} in so enake velikosti kot pripadajoče matrike uteži. V primeru polnih matrik izvajamo množenje po elementih. Redke matrike pa so predstavljene le z diagonalnimi binarnimi elementi iz {−1, 1}, medtem ko so ostali elementi 0. V tem primeru je pogoj, da so matrike kvadratne in diagonalne, z matrikami uteži pa jih matrično množimo. Ko se kontekstne matrike pomnožijo z matrikami uteži, se dimenzije slednjih v nobenem primeru ne spremenijo. Zaradi želje po pomnilniško učinkoviti metodi uporabljamo polne kontekstne matrike le v primeru majhnih pripadajočih matrik uteži, redke kontekstne matrike pa uporabimo za večje pripadajoče matrike uteži.

V polno povezanih mrežah kontekstne matrike uporabljamo tudi v prvem nivoju mreže, kar povzroči, da indirektno vplivajo tudi na vhodne podatke. Nasprotno pa v transformerjih apliciramo kontekste šele v mehanizmu pozornosti, kjer ne vplivamo na vhodne podatke.

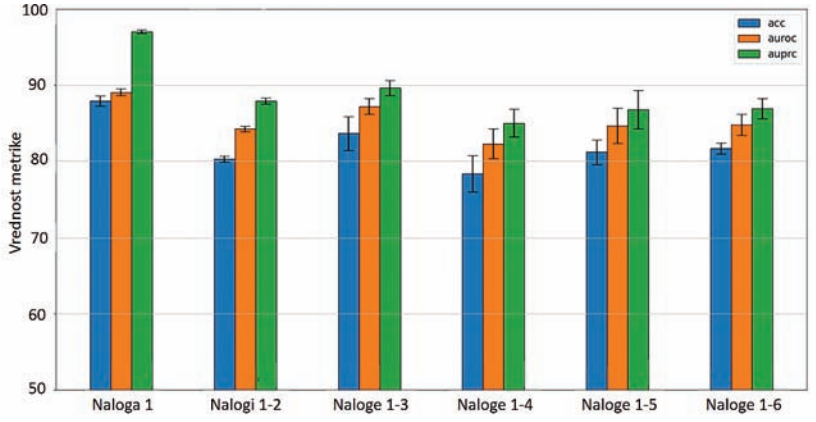
4 REZULTATI

4.1 Način vrednotenja

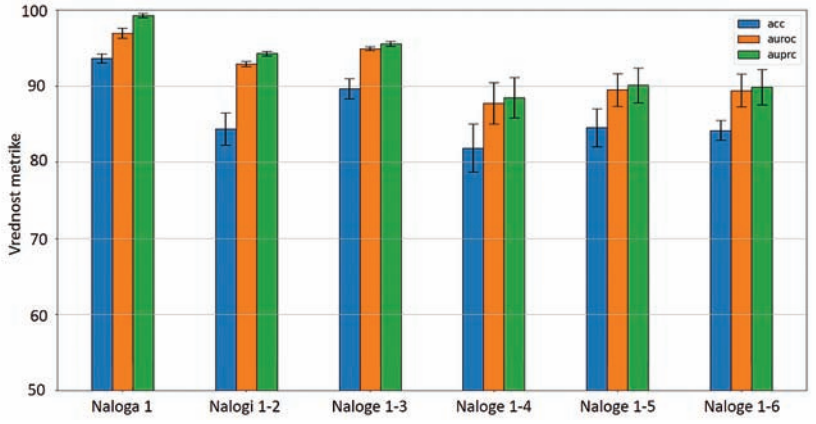
Naš pristop ocenjujemo z učenjem šestih nalog s področja obdelave naravnega jezika. Pri vseh nalogah gre za binarno klasifikacijo, in sicer zaznavanje sovražnega govora, analiza razpoloženja pri IMDb (angl. Internet Movie Database) komentarjih, zaznavanje neželenih sporočil, analiza komentarjev na platformah Amazon in Yelp, zaznavanje vab za klike in zaznavanje humorja. Vseh šest nalog ima podatke v obliki tekstovnih primerov, kjer ima posamezen primer na vhodu zaporedje besed ali stavkov, na izhodu pa binarno labelo (v primeru sovražnega govora labela pove, ali gre za sovražni govor ali ne). Tekstovni primeri so predprocesirani z algoritmom *Word2Vec* [5], ki posamezne besede spremeni v številske vektorje velikosti 32. Uspešnost merimo pri vseh naučenih nalogah. Vrstni red nalog je izbran naključno in je enak pri vseh metodah. Pri eksperimentih merimo točnost, AUROC (ploščino pod krivuljo ROC - angl. receiver operating characteristic) in AUPRC (ploščino pod krivuljo PRC - angl. precision-recall curve). Učenje trenutne naloge prenehamo, ko se AUROC na validacijski množici te naloge preneha izboljševati. Naš model sledi arhitekturi, ki je bila predlagana v [11] in vsebuje en nivo transformerskega kodirnika, ki mu sledita dva polno povezana nivoja (s 64 in 2 nevroni).

4.2 Primerjalne metode

Najprej predstavljamo primerjavo uporabe superpozicije v polno povezanih mrežah (PPM) v primerjavi z transformerskimi mrežami glede na vse tri metrike (tj. točnost, AUROC, AUPRC). Na sliki 1 prikazujemo, kako se vrednosti naših ocenjenih meritev spreminjajo med učenjem šestih nalog. Ko je učenje vsake zaporedne naloge končano, izračunamo povprečno vrednost metrike za vse do sedaj naučene naloge. Navpični stolpci pri nalogi i predstavljajo povprečne vrednosti nalog 1, ..., i. Ker se naše naloge razlikujejo po težavnosti, lahko opazimo, da povprečne vrednosti med učenjem nihajo. Iz grafov je razvidno, da je uporaba superpozicije v transformerju boljša od prve do zadnje naloge glede na vsa tri merila.



(a) Polno povezana mreža



(b) Transformer (naš pristop)

Slika 1: **Primerjava povprečnih vrednosti evalvacijskih metrik do i-te naloge z uporabo superpozicije v (a) polno povezani mreži in (b) transformerju.**

Za strnjeno predstavitev rezultatov iz vseh drugih primerjalnih metod v nadaljevanju prikažemo le vrednosti ocenjevalnih metrik po tem, ko so vse naloge naučene. To je enako zadnjim trem stolpcem (z desne strani) s slike 1. Ker so nekatere naše naloge neuravnotežene pri distribuciji ciljnih razredov, poročamo o AUROC in AUPRC metrikah. V tabeli 1 primerjamo našo metodo s tremi priljubljenimi pristopi nadaljevalnega učenja: *EWC* (angl. Elastic Weight Consolidation) [3], *Online EWC* (angl. Online Elastic Weight Consolidation) [9] in superpozicijo v polno povezanih mrežah [2]. Poleg tega primerjamo naš pristop z metodo, kjer vsako nalogo učimo v ločeni mreži, tako da ne more priti do pozabljanja modela (s tem torej dobimo zgornjo mejo uspešnosti). Ta pristop pričakovano dosega najboljše rezultate, vendar je izjemno pomnilniško neučinkovit. Kot je prikazano v tabeli 1, je naša metoda s superpozicijo v transformerjih boljša od drugih metod nadaljevalnega učenja glede na AUROC in AUPRC. Od druge najuspešnejše metode je naš pristop v povprečju šestih nalog boljši za 4,6 % pri AUROC in 3,0 % pri AUPRC.

Tabela 1: **Primerjalna analiza metod po naučenih šestih klasifikacijskih nalogah. Rezultati predstavljajo povprečje vseh šestih nalog. Najboljša rezultata (z neupoštevanjem ločenih mrež) sta krepko označena.**

Metode	AUROC	AUPRC
Ločene mreže (transformer)	94.0 ± 0.1	94.5 ± 0.1
Ločene mreže (PPM)	90.3 ± 0.1	91.3 ± 0.1
EWC [3]	74.4 ± 1.4	74.2 ± 4.1
Online EWC [9]	70.7 ± 2.0	73.1 ± 0.5
Superpozicija v PPM	84.8 ± 2.3	86.9 ± 2.0
Superpozicija v transformerjih	**89.4 ± 2.4**	**89.9 ± 2.7**

5 ZAKLJUČEK

Predstavili smo novo metodo nadaljevalnega učenja, kjer uporabljamo superpozicijski pristop znotraj transformerske arhitekture nevronskih mrež. Naša rešitev zmanjša pozabljanje modela med učenjem več nalog in doseže najboljšo zmogljivost med primerjanimi metodami. Glavna omejitev našega dela je ta, da so vse naloge vezane na isto globoko nevronsko mrežo in zato naš pristop služi le nalogam, ki jih je mogoče naučiti s podobno mrežno arhitekturo. Naše delo bi lahko dodatno izboljšali z možnostjo učenja nalog z različnimi velikostmi vhoda ali izhoda. V prihodnosti želimo razširiti našo metodo, da bo primerna za različne velikosti podatkov, pa tudi za različne mrežne arhitekture z združevanjem metode z drugimi pristopi regularizacije. Naše delo širi uporabnost superpozicijskega principa in ker smo prvi, ki smo omogočili uporabo superpozicije v transformerjih, verjamemo, da lahko naše delo ustvari novo vejo raziskav na področju nadaljevalnega učenja.

LITERATURA

[1] Magdalena Marta Biesialska, Katarzyna Biesialska, and Marta Ruiz Costa-jussà. Continual lifelong learning in natural language processing: a survey. In *COLING 2020, The 28th International Conference on Computational Linguistics: December 8-13, 2020, Barcelona, Spain (online): proceedings of the conference*, pages 6523–6541. Association for Computational Linguistics, 2020.

[2] Brian Cheung, Alexander Terekhov, Yubei Chen, Pulkit Agrawal, and Bruno Olshausen. Superposition of many models into one. *Advances in neural information processing systems*, 32, 2019.

[3] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in ne- ural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 114(13):3521–3526, 2017.

[4] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6470–6479, 2017.

[5] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[6] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? *Advances in neural information processing systems, 33:512–523, 2020.*

[7] German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

[8] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098, 2017.

[9] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, pages 4528–4537. PMLR, 2018.

[10] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, 2017.

[12] Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in Superposition. *NIPS*, (NeurIPS), 2020.

[13] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. *34th International Conference on Machine Learning, ICML 2017, 8:6072–6082, 2017.*

[14] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE,* 109(1):43–76, 2021.

▪

Marko Zeman je magistriral iz računalništva in informatike na Univerzi v Ljubljani, Fakulteti za računalništvo in informatiko leta 2020. Trenutno je raziskovalec in doktorski študent na Fakulteti za računalništvo in informatiko v Laboratoriju za kognitivno modeliranje. Njegova raziskovalna zanimanja so predvsem globoko učenje, nevronske mreže in metode nadaljevalnega učenja.

▪

Jana Faganeli Pucer je docentka na Fakulteti za računalništvo in informatiko. Njeno raziskovalno delo je osredotočeno na strojno učenje, predvsem na aplikacijo metod strojnega učenje v okoljskih znanostih. Več let sodeluje z Agencijo Republike Slovenije za okolje na področju kakovosti zraka.

▪

Igor Kononenko je doktor računalniških znanosti in redni profesor na Fakulteti za računalništvo in informatiko Univerze v Ljubljani ter predstojnik Laboratorija za kognitivno modeliranje. Njegova raziskovalna področja so umetna inteligenca, strojno učenje, nevronske mreže in kognitivno modeliranje. Je (so)avtor 225 člankov na teh področjih ter 13 učbenikov (dve knjigi izšli v Angliji).

▪

Zoran Bosnić je profesor na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Raziskovalno se ukvarja z umetno inteligenco, zlasti s strojnim učenjem. Osredotoča se pretežno na učenje iz podatkovnih tokov in na interdisciplinarne aplikacije strojnega učenja. Na tem področju je tudi (so)avtor okoli 70 znanstvenih člankov.