

Smurto atpažinimas

Juozas Vainauskas

Informatikos institutas

Matematikos ir informatikos fakultetas
Vilnius, Lietuva
juozas.vainauskas@mif.stud.vu.lt

Justas Vitkauskas

Informatikos institutas

Matematikos ir informatikos fakultetas
Vilnius, Lietuva
justas.vitkauskas@mif.stud.vu.lt

Vilius Macijauskas

Informatikos institutas

Matematikos ir informatikos fakultetas
Vilnius, Lietuva
vilius.macijauskas@mif.stud.vu.lt

Santrauka—Pasitelkdami apmokyta konvoluciinių neuroninių tinklą (ResNet50) ir rekurentinių neuroninių tinklų sukūrėme architektūrą, gebančią aptikti smurtą vaizdo įrašuose. Sukurta architektūra geba aukštą tikslumą lygiu klasifikuoti vaizdo įrašus iš dvių klasės: vaizdo įrašus su smurto požymiais ir vaizdo įrašus, kuriuose nėra smurto požymiai.

Raktažodžiai: CNN, RNN, CNN-RNN, video klasifikavimas, smurto atpažinimas

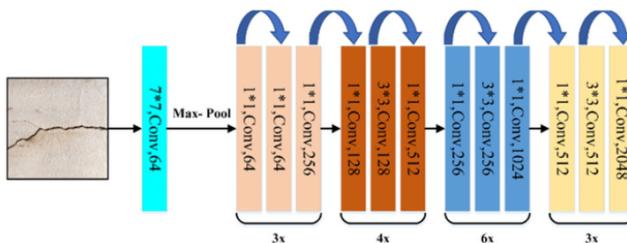
I. IVADAS

Vaizdo įrašų klasifikavimas yra vienas iš kompiuterinės regos (angl. "Computer vision") uždavinii. Vaizdo įrašai – kadrai, kurie yra išdėlioti eilės tvarka. Kiekvienas vaizdo įrašo kadas turi atitinkamą erdvinę informaciją, o jų seka suteikia papildomą laiko erdvės informaciją. Būtent dėl šios priežasties paprasti konvoluciiniai neuroniniai tinklai yra neefektyvūs sprendžiant vaizdo įrašų klasifikavimo uždavinį, kadangi jie įvertina kiekvieną kadrą atskirai, neatsižvelgdami į visą kadru seką. Šiam uždavinui spręsti siūlome naudoti konvoluciinio neuroninio tinklo ir rekurentinio neuroninio tinklo kombinaciją. Tokiu būdu, naudodamiesi konvoluciiniu neuroniniu tinklo modeliu, mes išgauname reikiamas kiekvieno kadro savybes, o naudodamiesi rekurentinio neuroninio tinklo modeliu galime klasifikuoti vaizdo įrašą, atsižvelgdami į kadru seką, o ne pavienius kadrus.

II. NAUDOJAMI NEURONINIŲ TINKLŲ MODELIAI

A. Konvoluciinis neuroninio tinklo modelis

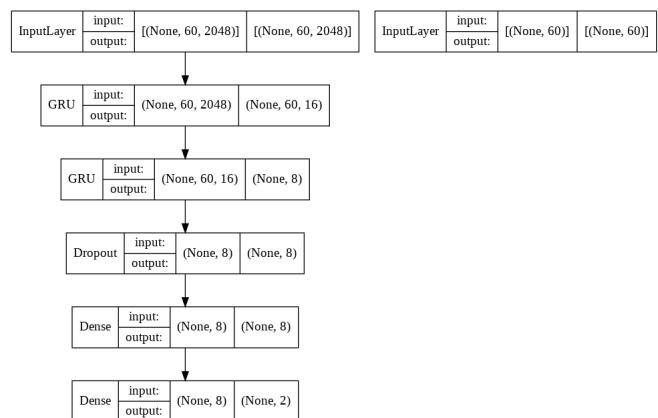
Naudojame konvoluciinių, 48 konvoluciinių sluoksnų neuroninio tinklo modelį ResNet50, kuris buvo ištreniruotas su ImageNet duomenų rinkiniu, kurį sudaro daugiau nei 1 mln. paveikslėlių. Kadangi norime išgauti tik mums aktualias kadru savybes, paskutinį ResNet50 modelio sluoksnį atmetame.



1 pav. – Modifikuoto ResNet50 struktūra

B. Rekurentinis neuroninio tinklo modelis

Naudojame rekurentinį, 7 sluoksnį neuroninio tinklo modelį, kurį treniruojame su duomenimis, gautais iš konvoluciinio neuroninio tinklo modelio.



2 pav. – Rekurentinio neuroninio tinklo modelio struktūra

III. PRIIMTI TECHNOLOGINIAI SPRENDIMAI

Kadangi klasikiniai RNN modeliai negali efektyviai apdrožoti ilgų duomenų sekų dėl gradientų nykimo, nuspindėme naudoti GRU sluoksnius. Tokį sprendimą priemėme dėl to, nes GRU sluoksnio architektūra yra paprastesnė, GRU skaičiavimų greitis yra spartesnis bei šiaisiai laikais GRU tampa vis populiaresnė alternatyva LSTM sluoksniams.

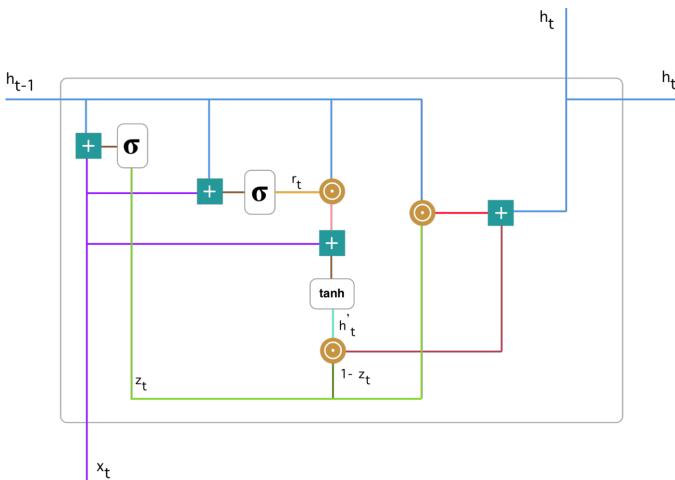
Siekiant išspręsti gradientų nykimo problemą, GRU naudoja atnaujinimo ir atstatymo vektorius (angl. "update gate", "reset gate"). Šie vektoriai nulemia kokia informacija turi išeiti iš GRU sluoksnio. Pagrindinis šio sluoksnio privalumas yra tas, jog GRU sluoksnis gali išlaikyti reikiama informaciją, surinktą iš ilgos duomenų sekos ar išmesti informaciją, kuri galiapti neberekalinga spėjimui atligli.

A. Atnaujinimo vektorius

Atnaujinimo vektorius z_t laiko momentui t skaičiuojamas pagal formulę:

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}) \quad (1)$$

Kai jeit is x_t patenka į GRU sluoksnį, jis yra padauginamas iš savo svorio $W^{(z)}$. Tas pats yra atliekama ir su praėjusio



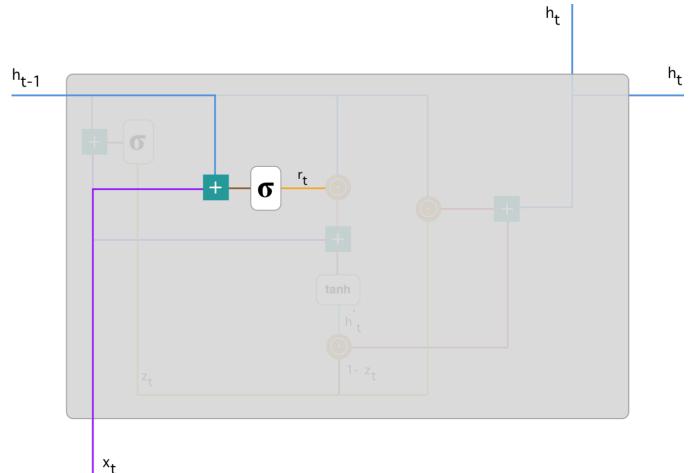
3 pav. – GRU sluoksnio architektūra



4 pav. – GRU sluoksnio apibrėžimai

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1}) \quad (2)$$

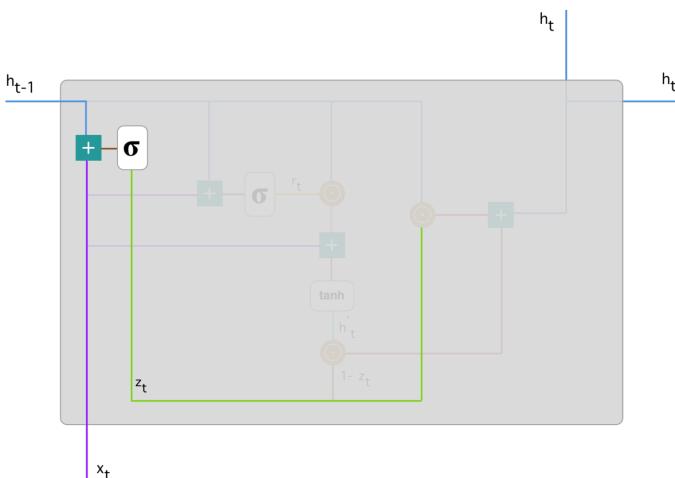
Atstatymo vektorius GRU sluoksnyje naudojamas neberekalingos informacijos apskaičiavimui ir išmetimui. Pateiktoje diagramoje galime pamatyti atstatymo vektoriaus apskaičiavimo procesą:



6 pav. – GRU sluoksnio architektūra

laiko momento paslėpta būsena (angl. hidden state). h_{t-1} yra padauginamas iš savo svorio $U^{(z)}$. Gauti rezultatai yra sudedami ir gautam rezultatui yra panaudojama sigmoido funkcija, jog rezultatas tilptų į intervalą $[0;1]$. Atnaujinimo vektorius padeda modeliui nuspresti kiek informacijos iš praėjusių laiko momentų perduoti toliau. Dėl šios priežasties atnaujinimo vektorius yra labai svarbus šioje architektūroje, kadangi jis gali nulemti, jog visa informacija iš praeities būtų išsaugoma ir tokiu būdu būtų išspręsta gradientų nykimo problema.

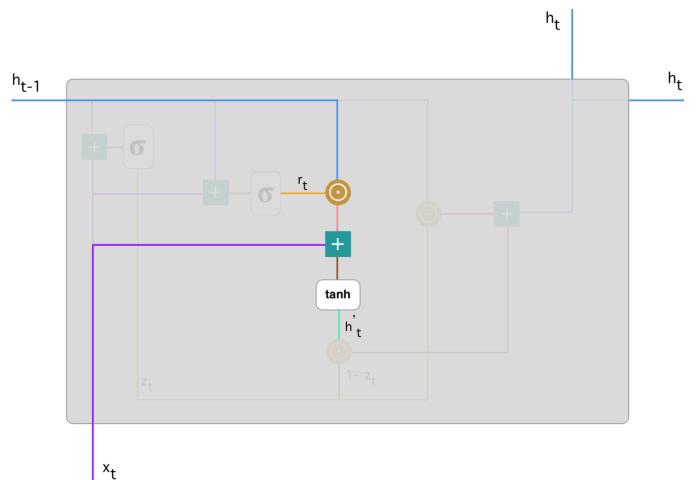
Apskaičiavus atnaujinimo vektorių ir pritaikius sigmoido funkciją, mūsų diagrama atrodo šitaip:



5 pav. – GRU sluoksnio architektūra

B. Atstatymo vektorius

Atstatymo vektorius yra skaičiuojamas pagal tą pačią formulę, kaip ir atnaujinimo vektorius, tik naudojami kiti svoriai:



7 pav. – GRU sluoksnio architektūra

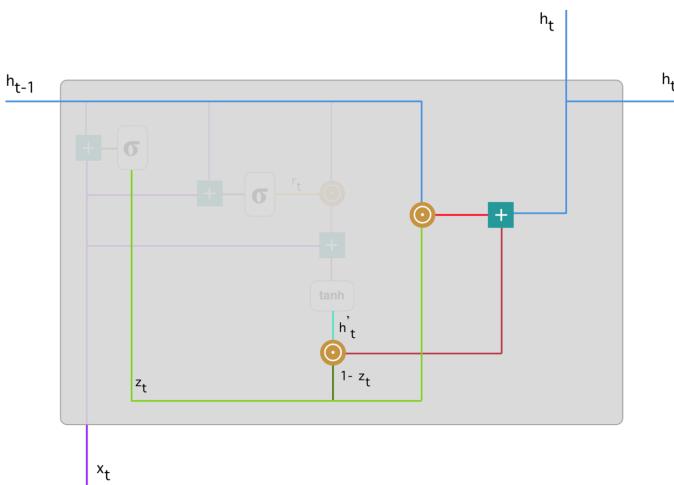
D. Paslépta būsena

Paskutinis GRU sluoksnio žingsnis yra pasléptos būsenos h_t apskaičiavimas. Norint apskaičiuoti pasléptą būseną, reikalingas atnaujinimo vektorius. Jis nusprendžia kokius duomenis paimti iš tarpinės pasléptos būsenos bei iš būsenų, kurios buvo praeityje. Apskaičiuoti pasléptą būseną galime su formule:

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h'_t \quad (4)$$

Įsivaizduokime, jog apdorojame vaizdo įrašą, kuriame smurta užfiksotas įrašo pradžioje. Tokiu atveju modelis gali nustatyti atnaujinimo vektoriaus reikšmę, artimą vienetui ir išsaugoti didžiąją dalį informacijos iš vaizdo įrašo pradžios. Kadangi z_t yra artimas vienetui, $1-z_t$ bus artimas 0, kas reikš vaizdo įrašo pabaigos duomenų ignoravimą, kurie mūsų spėjimui yra neaktualūs.

Paslēptos būsenos skaičiavimo žingsnius galime pamatyti pateiktoje diagramoje:



8 pav. – GRU sluoksnio architektūra

IV. DUOMENYS

Smurto atpažinimo vaizdo įrašuose užduočiai naudojame „Vision-based Fight Detection From Surveillance Cameras“ duomenų rinkinį. Šis duomenų rinkinys yra surinktas iš internetinės platformos „Youtube“ vaizdo įrašų, kuriuose yra smurto apraiškų. Taip pat, duomenų rinkinyje yra ir vaizdo įrašų iš stebėjimo kamерų, kuriuose smurto neaptikta. Iš viso duomenų rinkinyje yra 300 vaizdo įrašų, iš kurių 150 turi smurto požymį, o 150 smurto požymį neturi. Visi vaizdo įrašai yra 2 sekundžių ilgio. Vaizdo įrašai, kuriuose yra smurto požymiu, yra iškarptyti taip, jog visas vaizdo įrašas apimtų tik smurto veiksmus. Vaizdo įrašai yra padaryti skirtingose vietose (kavinėje, gatvėje, autobuse ir pan.). Taip pat, smurto vaizdo įrašai turi skirtinges smurto atvejus kaip smūgiavima objektu, spardymą, smūgiavimą rankomis, grumtyne.



9 pav. – Pavyzdiniai kadrai iš duomenų rinkinio

V. REZULTATAI

A. Treniravimo rezultatai

Rekurentinio neuroninio tinklo modelį treniravome 20 epochų, naudodami Adam optimizatorių ir kategorinės kryžminės entropijos nuostolių funkciją. Pabaigus treniravimą, su treniravimo duomenimis pasiekėme 95% tikslumą.

Epoch	Step	loss	accuracy		
9/9	[=====]	10s 39ms/step	0.5393		
Epoch 2/20	9/9	[=====]	0s 39ms/step	0.6554	accuracy: 0.5929
Epoch 3/20	9/9	[=====]	0s 37ms/step	0.5741	accuracy: 0.6821
Epoch 4/20	9/9	[=====]	0s 38ms/step	0.5450	accuracy: 0.7357
Epoch 5/20	9/9	[=====]	0s 38ms/step	0.5252	accuracy: 0.7321
Epoch 6/20	9/9	[=====]	0s 37ms/step	0.5004	accuracy: 0.7357
Epoch 7/20	9/9	[=====]	0s 39ms/step	0.5218	accuracy: 0.7321
Epoch 8/20	9/9	[=====]	0s 37ms/step	0.4536	accuracy: 0.8179
Epoch 9/20	9/9	[=====]	0s 38ms/step	0.4381	accuracy: 0.8000
Epoch 10/20	9/9	[=====]	0s 38ms/step	0.4266	accuracy: 0.8143
Epoch 11/20	9/9	[=====]	0s 38ms/step	0.3602	accuracy: 0.8536
Epoch 12/20	9/9	[=====]	0s 41ms/step	0.3418	accuracy: 0.8714
Epoch 13/20	9/9	[=====]	0s 39ms/step	0.3572	accuracy: 0.8679
Epoch 14/20	9/9	[=====]	0s 37ms/step	0.3069	accuracy: 0.8893
Epoch 15/20	9/9	[=====]	0s 38ms/step	0.3231	accuracy: 0.8750
Epoch 16/20	9/9	[=====]	0s 37ms/step	0.2697	accuracy: 0.9071
Epoch 17/20	9/9	[=====]	0s 38ms/step	0.2479	accuracy: 0.9250
Epoch 18/20	9/9	[=====]	0s 37ms/step	0.2288	accuracy: 0.9429
Epoch 19/20	9/9	[=====]	0s 38ms/step	0.2067	accuracy: 0.9536
Epoch 20/20	9/9	[=====]	4s 26ms/step	0.1894	accuracy: 0.9464
			Test accuracy: 97.5%		

10 pav. – Treniravimo statistika

B. Testavimo rezultatai

Norėdami patikrinti savo modelio tikslumą, jį ištetestavome su treniravimo metu modeliui nematytais vaizdo įrašais. Testavimui panaudojome 10% savo duomenų rinkinio. Testavimo metu sugebėjome pasiekti 97% tikslumą.

11 pav. – Smurtas (Nustatyta tikimybė - 92.24%)

12 pav. – Nėra smurto (Nustatyta tikimybė - 60.18%)

VI. IŠVADOS

Konvoluciinių ir rekurentinių neuroninių tinklų kombinacija padeda sėkmingai išspręsti vaizdo įrašų klasifikavimo problemą. Svarbu paminėti, jog šis sprendimo būdas gali būti pritaikytas ne tik smurto atpažinimo, bet ir kitoms panašaus pobūdžio problemoms.

LITERATŪRA

- [1] Ş. Aktı, G.A. Tataroğlu, H.K. Ekenel *Vision-based Fight Detection from Surveillance Cameras*, IEEE/EURASIP 9th International Conference on Image Processing Theory, Tools and Applications. İstanbul, Turkey, November 2019.