

TRANSFER LEARNING APMĀCĪBAS PAĀTRINĀŠANAI TRANSFER LEARNING FOR TRAINING ACCELERATION

Authors: **Ilmārs APEINĀNS**, e-mail: ia16024@edu.rta.lv

Vitālijs ŽUKOVŠ, e-mail: vz16020@edu.rta.lv

Scientific supervisors: **doc., Dr.sc.ing. Sergejs KODORS**, e-mail: sergejs.kodors@rta.lv

doc., Dr.sc.ing. Imants ZAREMBO, e-mail: imants.zarembo@rta.lv

Rēzeknes Tehnoloģiju akadēmija

Atbrīvošanas aleja 115, Rēzekne, Latvija

Abstract. In this work, authors compare training time of standard convolution neuron network model with model trained using transfer learning. Both models are based on Alexnet architecture. CNN model training from scratch included full model, but using transfer learning, some layers of model were frozen for learning acceleration considering transfer learning methodology.

Keywords: Alexnet, CNN, Convolution Neuron Network, Model, Transfer learning.

Ievads

Konvolūciju neironu tīkli (*convolutional neural networks*, tālāk *CNN*) [1] ir neironu tīklu modeļi, kuru pārsvarā pielieto vizuālu attēlu apstrādei. Tā kā *CNN* ir izstrādāti, lai darbotos, imitējot cilvēka smadzeņu neironu savienojumu modeļi, kad atsevišķi neironi reaģē uz stimuliem tikai ierobežotā redzes laukā, kas pazīstams kā “uztverošais lauks”. Šie lauki pārklājas, lai aptvertu visu redzes laukumu.

Transfer learning jeb pārneses mācīšanās ir mašīnmācības pētniecisks risinājums vienas problēmas risināšanai sagatavota konvolūciju neironu tīkla pielietošanai, citas, bet vienlaikus saistītas problēmas risināšanai [2]. Kā piemēram konvolūciju neironu tīkls kurš ir apmācīts atpazīt mašīnas var tik izmantots priekš kravas automašīnu atpazīšanai.

Šī pētījuma **mērķis** ir eksperimentāli salīdzināt divu konvolūciju neironu tīklu apmācības metožu, standartu un *transfer learning*, apmācības ātrdarbību.

Materiāli un metodes

Darba ietvaros modelis, kas tika izmantots priekš konvolūciju neironu tīklu apmācīšanas salīdzinājuma ir *Alexnet* [3] struktūras modelis.

Standarta modeļa apmācība tika balstīta uz *Fruits360* [4] datu kopas ar 120 klasēm, bet priekš *transfer learning* pielietojuma vispirms tika izmantota *CIFAR100* [5] datu kopa kas satur 100 klases un vēlāk tika pielietota standarta *Fruit360* datu kopa, kas satur pamata 120 klases un modificēta *Fruit360* datu kopa, kura tika pārveidota tikai, lai saturētu 3 klases: “apple”, “pears” un “noise”, kas tika modificēts, lai klases būtu pēc izmēra aptuveni vienādas, ~400 attēli.

Konvolūciju neironu tīklu izstrādei tika izmantots *Tensorflow 2.0*, *Keras* un *Jupyter Notebooks*.

Eksperiments

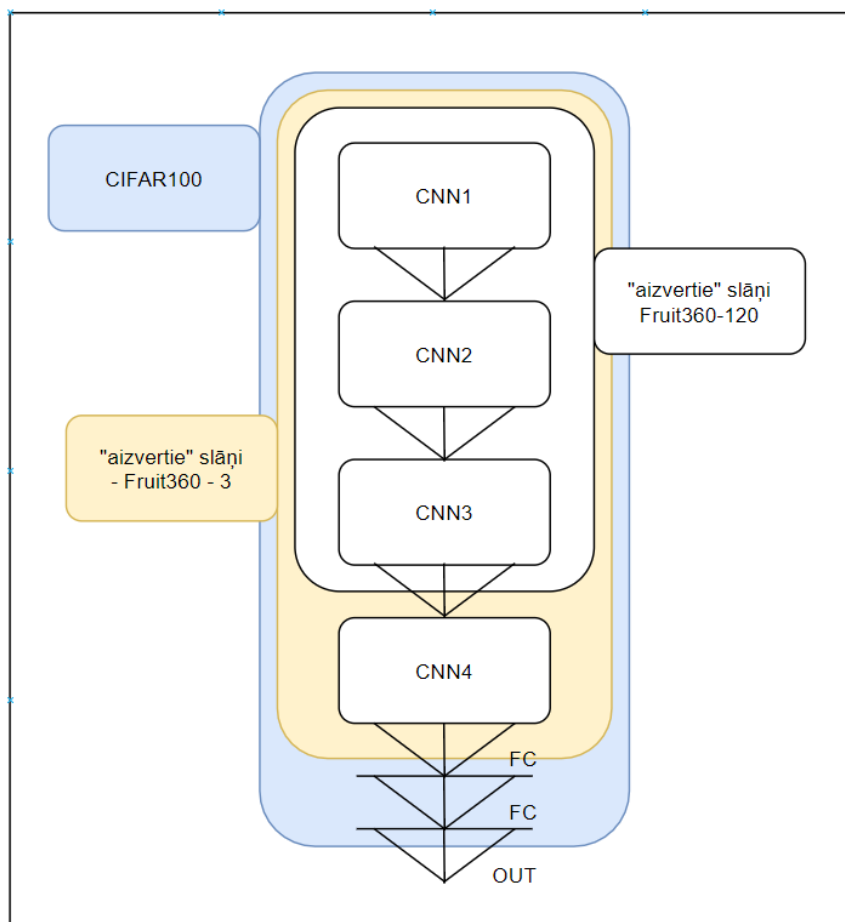
Darba eksperimenta pirmais solis bija konvolūciju neironu tīkla apmācība, izmantojot *Alexnet* struktūras modeļi (skat. 1. tab.). Apmācība tika veikta 10 reizes un visi apmācības laiki tika fiksēti.

Alexnet arhitektūras modelis

	Layer (type)	Output Shape (platums, augstums, filteru skaits)	Param.
CNN1	<i>conv2d_1</i>	(48, 48, 16)	1216
	<i>batch_normalization_1</i>	(48, 48, 16)	64
	<i>max_pooling2d_1</i>	(24, 24, 16)	0
CNN2	<i>conv2d_2</i>	(22, 22, 32)	4640
	<i>batch_normalization_2</i>	(22, 22, 32)	128
	<i>max_pooling2d_2</i>	(11, 11, 32)	0
CNN3	<i>conv2d_3</i>	(9, 9, 64)	18496
CNN4	<i>conv2d_4</i>	(7, 7, 128)	73856
	<i>max_pooling2d_3</i>	(3, 3, 128)	0
FC	<i>dense_1</i>	512	590336
FC	<i>dense_2</i>	512	262656
OUT	<i>dense_3</i>	3	1539

Konvolūciju neironu tīklu apmācība notiek vairākos posmos (skat. 1. att.):

1. Apmācība ar *CIFAR100* datu kopu;
2. Apmācība ar *Fruits360* datu kopu un visām 120 klasēm;
3. Apmācība, izmantojot *Fruits360* datu kopu, bet tikai 3 klases, “apple”, “pears” un “noise”.



1. attēls. “Aizvērti” pirmie trīs CNN slāņi

Pirmajā posmā tika apmācīts konvolūciju neironu tīkls uz *CIFAR100* datu kopas pamata. Dotā datu kopa satur sevī 100 dažādas klases ar 600 attēliem katrai klasei. *CNN* tika apmācīts pilnā mērā. *Nākošais solis*, kas tika sperts, bija iesaldēti pirmie trīs *Alexnet* modeļa slāņi (skat. 2. att.) un tika apmācīts modelis no jauna, tikai apmācības laikā netika veiktas nekādas manipulācijas ar pirmo trīs konvolūciju neironu tīkla slāņu svariem, saglabājot iepriekš apmācītos neironu svarus.

```
# Load pretrained model
model = tf.keras.models.load_model( cifar100 )

# Freeze pretrained layers
model.get_layer('sconv2d_1').trainable = False
model.get_layer('sconv2d_2').trainable = False
model.get_layer('sconv2d_3').trainable = False
```

2. attēls. “Aizvērti” pirmie trīs *CNN* slāņi

Pēc otrās apmācīšanas, jaunais modelis saglabāja apmācītos neironu svarus pirmajos trijos slāņos un apmāca atlikušos modeļa slāņus. Pabeidzot apmācību, izmantojot *Fruits360* datu kopu, kas satur 120 klases ar augļiem, *nākošais* un *pēdējais* solis ir veikt *pēdējo* apmācības posmu, kur tiek saglabāti iepriekš “aizvērtie” slāņi un tiek “aizvērts” ceturtais *Alexnet* modeļa slānis (skat. 3. att.) un tiek sagatavots izvadslānis, lai tas atbilstu datu kopas klašu struktūrai, kas tika izmantota apmācības procesā.

```
# Freeze and replace output layer
new_model = keras.models.Sequential()
for layer in model.layers[:-1]:
    layer.trainable = False
new_model.add( layer )
```

3. attēls. “Aizvērts” ceturtais *CNN* slānis

Priekš *pēdējās* apmācības tiek izmantota datu kopa, kas ir balstīta uz *Fruits360* datu kopas, bet ir tikai trīs klases, kas ir, “apple”, “pears” un “noise”. No pamata datu kopas tika izņemtas visas klases, kas saturēja nosaukumā “apple” vai “pears” un tika izveidotas jaunas klases kas saturēja visu saturu no pamata klasēm. Un tad tika izveidota klase “noise”, kas satur attēlus tiek izmantoti kā nepareizas atbildes *CNN* apmācībai. Klases tika veidotas aptuveni ar vienādu izmēru, lai neironu tīkls neatdotu priekšrocību vienai klasei.

Rezultāti

Veicot apmācību, izmantojot standarta *CNN* apmācības metodi, tika veikti modeļu apmācība 10 reizes balstoties uz *Fruits360* datu kopu ar 3 klasēm. 2. tabulā ir fiksēti rezultāti, kas tika iegūti pēc *CNN* apmācības. Tabulā ir divas iedaļas, kas satur laiku minūtēs (*Train time (min)*) un iegūto apmācības precizitāti (*Accuracy*). Pēc 10 apmācību veikšanas tika aprēķināts, ka vidēji vienas apmācības veikšanai nepieciešams ~104 minūtes un *CNN* precizitāte iegūta ir 95%.

2. tabula

Standarta apmācības nepieciešamais laiks

	<i>Apmācības laiks (min)</i>	<i>Precizitāte</i>
1	117	0,95119
2	104	0,95006
3	60,66666667	0,95423
4	117	0,94917
5	82,33333333	0,95261
6	91	0,93892
7	156	0,95286
8	130	0,95116
9	99,66666667	0,95611
10	99,66666667	0,94785
<i>Vidēji</i>	104,7878788	0,95019

Veicot *transfer learning* metodes pielietošanu tika veikta apmācība trīs etapos, kur pirmais etaps tika veikts tikai vienu reizi. Pirmajā apmācības etapā tika apmācīts CNN atpazīt dažādus objektu attēlus. Tas aizņēma lielāko apmācības laiku, kas bija vairāk kā 24 stundas (skat 3. tab.), bet precizitāte kas tika iegūta bija tikai 47%.

3. tabula

Pirmā apmācības etapa rezultāti

<i>Apmācības laiks (min)</i>	<i>Precizitāte</i>
1449,7	0,47080

Pēc pirmā apmācības etapa pabeigšanas, tika aizvērti pirmie trīs iepriekš apmācītā modeļa slāņi un tika uzsākta modeļa pārāpmācīšana, kur tika apmācīti atlikušie modeļa slāņi. Kā rezultātā apmācības laiks salīdzinājumā ar pirmo etapu samazinājās 4,5 reizēs (skat 4. tab.)

4. tabula

Otrā apmācības etapa rezultāti

<i>Apmācības laiks (min)</i>	<i>Precizitāte</i>
310	0,98681

Pēc otra apmācības etapa pabeigšanas modelis ir sagatavots pēdējam apmācības etapam kur tam tiks iemācīts atpazīt ābolus un bumbierus, bet visu kas neatbilst iepriekšminētajā klasēm definēt kā troksni un ignorēt. Pirms apmācības uzsākšanas tika aizvērts arī ceturtais modeļa slānis, atstājot tikai divus pēdējos slāņus apmācībai. Apmācība tika veikta 10 reizes uz rezultāti tika piefiksēti 5. tabulā. Var redzēt, ka vidējais apmācības laiks samazinājās līdz 8,5 minūtēm jeb 12 reizēs salīdzinājumā ar apmācību pilnīgi no jauna.

Transfer learning pielietotais laiks

	<i>Apmācības laiks (min)</i>	<i>Precizitāte y</i>
1	8,133333333	0,96176
2	10,53333333	0,96976
3	10,8	0,96754
4	6	0,95820
5	10,8	0,97421
6	5,866666667	0,95643
7	8	0,96665
8	5,733333333	0,95776
9	8,933333333	0,96665
10	10,53333333	0,96843
Average	8,53333333	0,96474

Secinājumi

Eksperimenta veikšanas laikā tika noskaidros, ka vidēji, apmācot *CNN* pilnīgi no jauna, viena neirontīkla apmācības laiks ir ~104 minūtes, kas nav slikti, ja nav plānots uzlabot modeļa darbību nākotnē. Bet, izmantojot *transfer learning* metodi modeļa apmācībai sagataves darbi, kas ir jāveic, lai nonāktu pie pielietojama modeļa aizņem daudz vairāk laika. Pirmais apmācības etaps aizņēma vairāk nekā 24 stundas, bet otrs etaps vairāk kā 5 stundas. Toties pēdējais etaps, kur tika modelis apmācīts atpazīt ābolus un bumbierus atbilstoši sagatavotajai datu kopai vidēji aizņēma tikai 8,5 minūtes, kas ir aptuveni 12 reizēs ātrāk. Jāatzīmē, ka vidēja precizitāte *transfer learning* gadījumā izauga par 1,5%.

Ja uzdevums, kur tiek pielietots *CNN* ir tikai izveidot vienu modeli un neņemt vērā tālāku tā uzlabošanu, tad var izmantot standarta metodi bez priekšapmācības. Tomēr, ja modeļa darbības mērķis pieprasa tālāku tā uzlabošanu, izmantojot jaunus datus, tad daudz izdevīgāk ir izmantot *transfer learning* sagatavotu modeli, jo tālākai modeļa apmācībai nav nepieciešams veikt visus trīs apmācības etapus, bet tikai trešo, kas padara to daudz izdevīgāku salīdzinājumā ar standarta metodi.

Summary

During the experiment, it was found that the average training time from scratch per CNN model is ~104 minutes, which is not bad, if there are no plans to improve the performance of the model in the future. But using the transfer learning method for model training, the preparatory work that needs to be done to get to the applicable model takes much more time. The first phase of training lasted approximately 24 hours and the second phase little bit more than 5 hours.

However, the last stage, where the model was trained to recognize apples and pears according to the prepared data set, took on average 8.5 minutes, that is about 12 times faster. It must be mentioned, that average accuracy is improved by 1.5%

If the task, where CNN is used is to create only one model and ignore further improvements, then you can use the training scratch. However, if the model requires further improvement using new data, then it is much more advantageous to use a model prepared by transfer learning, because for further model training it is not necessary to perform all 3 training stages, but only the last, which makes it much more advantageous compared to the training from scratch.

Acknowledgement

Funding institution: Latvian Council of Science

Funding number: lzp-2019/1-0094

Acronym: FLPP-2019-1

Funding text: This research is funded by the Latvian Council of Science, project “Application of deep learning and datamining for the study of plant-pathogen interaction: the case of apple and pear scab”, project No. lzp-2019/1-0094

Literatūra

[1] Konvolucionāli neironu tīkli vizuālai atpazīšanai [tiešsaiste], [atsauce uz 07.08.2020.]. Pieejams:

<https://cs231n.github.io/convolutional-networks/>

[2] Transfer learning pamatideja [tiešsaiste], [atsauce uz 18.08.2020.]. Pieejams:

<https://towardsdatascience.com/transfer-learning-with-convolutional-neural-networks-in-pytorch-dd09190245ce>

[3] Alexnet arhitektūra [tiešsaiste], [atsauce uz 12.08.2020.]. Pieejams: <https://medium.com/@smallfishbigsea/a-walk-through-of-alexnet-6cbd137a5637>

[4] H. Mureşan & O. Mihai. Fruit recognition from images using deep learning. Acta Universitatis Sapientiae, Informatica, vol. 10, pp. 26-42, 2018.

[5] CIFAR100 datu kopa [tiešsaiste], [atsauce uz 12.08.2020.]. Pieejams: