

NEIRONU TĪKLU ATPAZĪŠANAS PROCESA VIZUALIZĀCIJAS METODES

VISUALIZATION METHODS OF IMAGE CLASSIFICATION PROCESS IN NEURAL NETWORKS

Autori: **Kaspars VOGULIS**, e-pasts: kaspars.vogulis@gmail.com,
Valdis PLATONOVŠ, e-pasts: valdisp3@inbox.lv,
Edgars JUDOVIČS, e-pasts: edgars.judovics@gmail.com
Zinātniskā darba vadītājs: **Dr.sc.ing., docents Sergejs KODORS**,
Rēzeknes Tehnoloģiju akadēmija,
Atbrīvošanas aleja 115, Rēzekne, Latvija

Abstract. *The aim of the work was to find and describe ways of visualising layers of neural networks, which analyse images and classify them. By visualising network layers, scientists and developers could see features, which influence on results of neural network mapping, training and overall result. In this paper, authors demonstrate different visualization methods, which can be applied by machine learning engineers.*

Keywords: *image classification, neural networks, layers.*

Ievads

Neironu tīklu adaptīvā pieeja un iespēja apmācīties padara tos par lielisku rīku, kā risināt problēmas, pielietojot statistisko informāciju un lielas datu kopas. Neironu tīkli tiek pielietoti dažādiem uzdevumiem: elektroenerģijas efektivitātes noteikšana ēkām [1], vēža šūnu noteikšana organismā [2], ēdiena atpazīšana [3], sejas atpazīšana, izmantojot novērošanas kameras [4], ugunsgrēka monitorings [5] un citi uzdevumi.

Attēlu atpazīšanā neironu tīklu izmantošana ir efektīva un lēta pieeja, jo neirontīklu apmācība nepieprasa mašīnāpmācības inženierus manuāli definēt atpazīšanas īpašības.

Tomēr neironu tīkli strādā pēc melnās kastes (*blackbox*) principa – var tikai gaidīt vēlamu rezultātu, bet viennozīmīgi pārliecināties par korektu neirontīkla darbu nav iespējams, tāpēc neirontīklu darbs ir neprognozējams.

Pats attēlu atpazīšanas process notiek, izmantojot vairākus neirontīkla slāņus, kur katrs slānis atbild par sava veida vizuālo datu apstrādi. Tāpēc, lai varētu noteikt kādas īpašības tiek pielietotas klasifikācijā, ir izstrādāti dažādi vizualizācijas palīgriķi. Apskatot aktivizācijas kartes vairākiem slāņiem, var noteikt, kādi parametri ir jāpaslēpj, bet kuriem jāpievērš lielāka uzmanība.

Darba mērķis ir izpētīt kādas atpazīšanas slāņu vizualizācijas metodes pastāv un kā šīs metodes var pielietot.

Darba uzdevumi:

1. Eksperimentāli izmēģināt uzmanības reģionu vizualizācijas metodi;
2. Eksperimentāli izmēģināt uzmanības īpašību vizualizācijas metodi.

Materiāli un metodes

Pētījumā tika pielietotas šādas metodes:

- Zinātniskās literatūras analīze, lai iegūtu informāciju par neironu tīklu darba principiem un eksistējošiem risinājumiem.
- Eksperimentu veikšana ar slāņu vizualizāciju, izmantojot apmācītus neironu tīkla modeļus.

Eksistē dažādi veidi, kā vizualizēt neironu tīklus: gan neironu tīkla struktūras vizualizācijas metodes, kad tiek attēlotas saites starp slāņiem un neironiem, gan metodes, kas

attēlo kā neironu tīkls apstrādā ievades datus, ko neirontīkls redz, un kādiem elementiem pievērš lielāku uzmanību.

Eksperimenta laikā tiek izmantoti šādi rīki:

1. *Tensorflow* - brīvs atvērta koda *Python* bibliotēku apvienojums, ar kura palīdzību var izveidot, apmācīt un darbināt neironu tīklus.
2. *Keras* – atvērta koda neironu tīklu bibliotēka, kas papildina *Tensorflow* funkcionalitāti.
3. *tf-keras-vis toolkit* – rīku apvienojums, kas apstrādā *Keras* veidotus modeļus, kas ļauj vizualizēt atpazīšanas procesus.
4. *Google Collab* – uzņēmuma *Google* mākoņpakalpojums, kas ļauj palaist kodu uz uzņēmuma serveriem.

Darbā tika apskatītas 2 metodes:

1. Uzmanības reģionu vizualizācija – tiek noteikti punkti, pēc kuriem neironu tīkls atpazīst objektus attēlā;
2. Uzmanības īpašību vizualizācija – metode, kas tiek pielietota neirontīklu slāņiem, lai attēlotu atpazīšanas īpašības, ar kurām strādā filtrs.

Eksperimentā tiek pielietots apmācīts neironu tīkla *VGG16* modelis [6] uz *Imagenet* datu kopas, kuru piedāvā *Keras* bibliotēka.

Rezultāti un diskusija

Uzmanības reģiona vizualizācija

Viena no vizuālācijas metodēm ir uzmanības reģiona attēlošana (*saliency map*), kad ar dažādu krāsu gradācijām, topogrāfijas stilā, attēlo vietas, kurām neironu tīkls pievērš vairāk vai mazāk uzmanības.

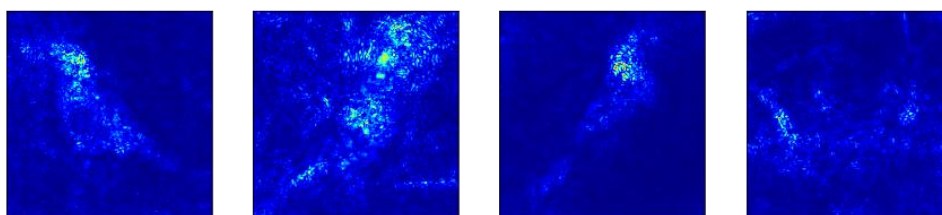
Izmantojot uzmanības kartes, var viegli un vizuāli noteikt, ko neirona tīkls meklē ievadītajos attēlos, kuriem reģioniem pievērš vairāk vai mazāk uzmanības, kur var būt iespējami meklēšanas trūkumi, utt. Izmantojot šo informāciju ir iespējams turpināt apmācīt neironu tīklu, ņemot vērā trūkumus un daļas, kuras neirontīklam ir grūti atpazīt, lai klasificētu attēlu.

Uzmanības karšu veidošanai tika izmantotas *tensorflow*, *keras* un *tf-keras-vis python* bibliotēkas. Demonstrācijas nolūkiem tika izmantoti četri attēli (skat. 1. attēlu): vārna, gulbis, gulbja rotaļlieta un lidmašīna.



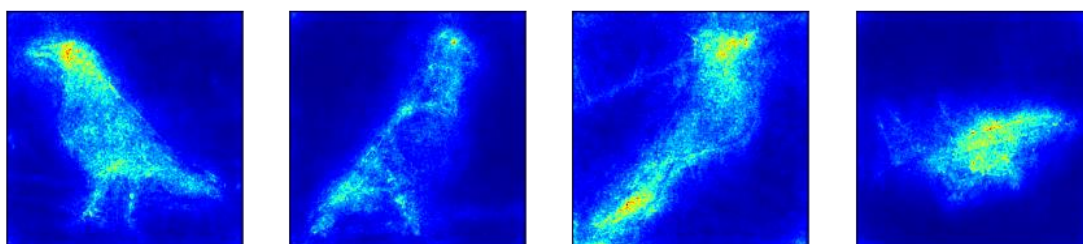
1.att. Ievades dati

Cilvēks uzreiz, apskatot šos attēlus, spēj atpazīt, kas ir attēlots. Datora redzei tā ir liela problēma un liels izaicinājums. Lai uzzinātu, vai neirontīkls saprot, kur atrodas attēla objekts, izmanto uzmanības kartes. 2. attēlā ir redzami neirontīklu uzmanības reģioni, kuros var saskatīt attēlu objektus, bet tie ir diezgan abstrakti, it īpaši lidmašīnas gadījumā. Ir grūti saprast, kur atrodas lielāki neirona tīkla fokusa punkti.



2. att. Neirontīkla uzmanības punkti

Lai labāk saskatītu uzmanības reģionus, tos var pastiprināt, izpildot attēla normalizāciju (skat. 3. attēlu). Rezultātā var labāk saskatīt apstrādātus objektus, gan putnus, gan arī lidmašīnu. Kā redzams paraugos, neironu tīkls pārsvarā ignorē attēlu fonu un fokusējas uz objekta – galvenokārt, uz putnu galvas, acis un astes.



3. att. Normalizētas neirontīkla uzmanības reģionu kartes

Pēc uzmanības kartēm, var secināt, ka dators spēj atšķirt objektu no tā fona un lielākoties spēj atrast putnu raksturīgās īpašības. Baloža gadījumā, var redzēt, ka tīklam ir grūtības atpazīt baloža spārnu.

Uzmanības īpašību vizualizācija

Konvolūcijas slānis (*convolution layer*) ir plaši izmantota attēlu atpazīšanas neirontīkla struktūra. To izmanto, lai atpazītu formas un attēla īpašības. Slānis saglabā attiecības starp pikseliem, aplūkojot bildi caur noteikta izmēra logiem (*windows*). Tad, apskatot attēlu caur filtru, tiek iegūta attiecību vērtība noteiktā attēla apgabalā, ar ko var izteikt bildes parametrus uz šo apgabalu.

Tā kā bilde sastāv no bitu vērtībām 1 un 0, izmantojot matricu reizināšanu var iegūt vērtību atšķirības. Pēc apstrādes ar slāni, tiek samazināts attēla izmērs, bet netiek pazaudēta informācija.

Ja bilde nav melnbalta, bet krāsaina, tiek apstrādātas 3 bildes – katra savai krāsai (sarkana, zaļa un zila). Ievada slānim tiek padota bilde, kurai ir atdalītas krāsas, noteikts augstums un platums. Pats filtrs arī satur krāsu slāņus, un izmēra parametrus. Rezultātā iegūta bilde atspoguļo izmaiņas un atšķirības starp blakus esošiem pikseliem.

4. attēlā ir redzama konvolūcijas slāņa darba loģika, kur attēla matrica tiek sareizināta ar filtra matricu, lai iegūtu rezultātu.

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |

*

| | | |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |

5 x 5 – Image Matrix

3 x 3 – Filter Matrix

4.att. Konvolūcijas slāņa darba princips

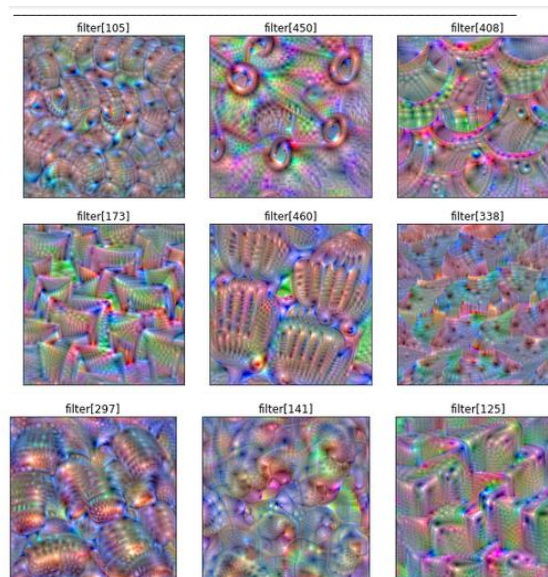
Konvolūcijas slānī var mainīt filtrus, lai iegūtu dažādus rezultātus.

Lai eksperimentāli apskatītu konvolūcijas slāņu filtru vizualizācijas metodi, tika izmantots putna attēls (skat. 5. attēlu).



5.att. Putna attēls

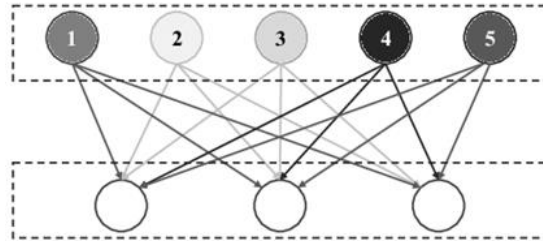
Lai vizualizētu filtrus, tika izvēlēti filtri pēc nejaušības principa. 6. attēlā var redzēt filtru vizualizācijas rezultātu.



6.att. Filtru vizualizācija

Rezultātā var redzēt dažādu konvolūcijas slāņu izmantoto attēlu rezultātu. Apskatot, var noteikt, kuri filtri atbilst vajadzīgajam atpazīšanas īpašībām, bet kuri - nē.

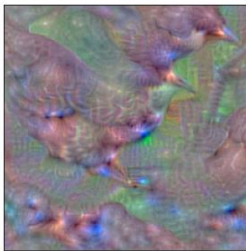
1D slānis ir viens no vienkāršākajiem slāņu tipiem, kuram visi neironi ir savienoti ar visiem neironiem nākamā slānī. Katrs slāņa neironš saņem ievadi no visiem iepriekšēja slāņa neironiem - tādējādi tie ir blīvi savienoti. Lai izveidotu pilnīgi saistītu 1D slāni, *Keras* bibliotēka piedāvā *DenseLayer*. 7. attēlā ir redzams *DenseLayer* piemērs. Ieejas neironu krāsas norāda aktivizācijas vērtības. Tumšāka krāsa apzīmē stiprāku signālu.



7.att. *DenseLayer* piemērs

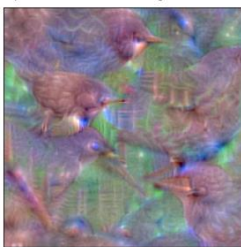
Praktiskā daļā tika apskatīta *DenseLayer* darbība. Lai redzētu kā mainās attēli un iegūtie dati no apmācības ilguma, tika mainīts apmācības soļu skaits. Jo vairāk soļu, jo vairāk reizi tika veikta attēla apstrāde. Eksperimentā tika apstrādāta putna bilde (skat. 5. attēlu) un parādīti rezultāti slāņiem, kuriem tika konstatēti blīvāki normalizēti uzmanības reģioni (skat. 8., 9. un 10. attēlus).

Steps: 100 Losses: [113.66600799560547], Regularizations: [[('TotalVariation', 83.50137329101562), ('L2Norm', 0.025623245164752007)]]
 Steps: 200 Losses: [180.3651885986328], Regularizations: [[('TotalVariation', 95.44664001464844), ('L2Norm', 0.025486545637249947)]]



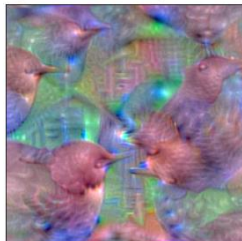
8.att. 200 soļu rezultāts

Steps: 100 Losses: [103.49851989746094], Regularizations: [[('TotalVariation', 88.19293975830078), ('L2Norm', 0.02561434730887413)]]
 Steps: 200 Losses: [109.63790893554688], Regularizations: [[('TotalVariation', 90.4698486328125), ('L2Norm', 0.02551199682056904)]]
 Steps: 300 Losses: [199.9986114501953], Regularizations: [[('TotalVariation', 95.8156509399414), ('L2Norm', 0.025409458205103874)]]



9.att. 350 soļu rezultāts

Steps: 100 Losses: [125.67552185098594], Regularizations: [[('TotalVariation', 84.12543487548828), ('L2Norm', 0.02561759389936924)]]
 Steps: 200 Losses: [194.90782165527344], Regularizations: [[('TotalVariation', 93.90827178955078), ('L2Norm', 0.02547607812093067)]]
 Steps: 300 Losses: [234.0313720703125], Regularizations: [[('TotalVariation', 103.06727600097656), ('L2Norm', 0.025368813425302505)]]
 Steps: 400 Losses: [233.4376678466797], Regularizations: [[('TotalVariation', 108.03016662597656), ('L2Norm', 0.02529124729335308)]]
 Steps: 500 Losses: [305.28594970703125], Regularizations: [[('TotalVariation', 108.28776550292969), ('L2Norm', 0.025196414440870285)]]



10.att. 500 soļu rezultāts

Pēc rezultātiem ir redzams, ka iterāciju skaits padara iegūto attēlu daudz tuvāku vajadzīgajiem datiem.

Secinājumi

Darba laikā tika secināts, ka eksistē vairāki rīki un gatavi risinājumi, kas var vizualizēt neironu tīkla attēlu atpazīšanas procesu. Eksperimentāli tika apskatītas: uzmanības reģionu vizualizācijas un uzmanības īpašību vizualizācijas metodes.

Neironu tīklu vizualizācijas metodes un rīki ir ļoti noderīgi instrumenti, izstrādājot viedo risinājumu. Uzmanības reģionu vizualizācijas metode ir viens no paņēmieniem, kā vizuāli attēlot neirona tīkla analīzes fokusu. Izmantojot šo metodi var redzēt, kādām attēla daļām neirona tīkls pievērš vai nepievērš uzmanību, tādējādi dodot iespēju to uzlabot vai modificēt tam paredzētajiem nolūkiem.

Konvolūcijas slānis apstrādā attēlu, pielietojot redzes logus. Izmantojot uzmanības īpašību vizualizāciju, var pārliecināties vai pareizus objektus neironu tīkls izvēlas atpazīšanas procesā. Katram filtram ir sava matrica un parametri, tāpēc, izmantojot abas metodes, var atrast pašu efektīvāko filtru.

Uzmanības īpašību vizualizācija ir atkarīga no apmācības ilguma, jo ilgāk notiek apmācība, jo vieglāk saskatīt atpazīšanas īpašības.

Summary

During the work it was concluded that there are several tools and ready-made solutions that can visualize one or more processes of neural network image recognition methods. Desirability, convolution, and dense-layer visualizations were performed to see how the neural network processes images.

Neural network visualization methods and tools are very useful tools for developing computer intelligence models and the efficiency of the analysis of their intended tasks. The method of visualization of desirability is one of the ways to visually depict the thought process of the neural network in a human-understandable form. Using this method, it is possible to see which parts of the image the neural network pays attention to or ignores, thus allowing it to be improved or modified for its intended purposes.

The convolution layer processes the image with a specific filter or filters and obtains the pixel ratio of the image in a specific (small) area. Using convolution layer visualization, the effect of filters on the desired result can be determined. Each filter has its own matrix and parameters, so using the obtained filter activation and desirability maps, you can find the most efficient filter.

The dense layer is used as a linear operation, where each input is connected to each output. The results of Dense layer visualization mostly depend on the number of steps or iterations to be performed, the more steps, the more detailed the image remains. The "loss factor" is like the second parameter that affects the results - if it is increased, the image is less detailed.

Literatūras saraksts

1. Chae Y.T., Horesh R., Hwang Y., Lee Y.M., *Artificial neural network model for forecasting sub-hourly electricity usage in commercial buildings.* [Piekluve:12.04.2020]
2. Huang M., Hung Y., Lee W., Li R. K., Wang T.. *Usage of Case-Based Reasoning, Neural Network and Adaptive Neuro-Fuzzy Inference System Classification Techniques in Breast Cancer Dataset Classification Diagnosis* [Piekluve:14.04.2020]
3. Kagaya H., Aizawa K. , Ogawa M. . *Food Detection and Recognition Using Convolutional Neural Network.* [Piekluve:12.04.2020]
4. Zhang, H., Zou, Z., Li, J. et al. *Flame image recognition of alumina rotary kiln by artificial neural network and support vector machine methods.* J. Cent. South Univ. Technol. 15, 39–43 (2008). [Piekluve:14.04.2020]

5. Sinha P. *A symmetry perceiving adaptive neural network and facial image recognition*
[Piekluve:14.04.2020]
6. VGG16 – Convolutional Network for Classification and Detection [Tiešsaiste] Pieejams:
<https://neurohive.io/en/popular-networks/vgg16/>