ISSN 2078–5097. Вісн. Львів. ун-ту. Сер. прикл. матем. та інф. 2018. Вип. 26. С. 90–104
Visnyk of the Lviv University. Series Appl. Math. and Informatics. Issue 26. P. 90–104

UDC 004.032.26::004.045+004.8+004.932.72

A DROPOUT TECHNIQUE STUDY FOR THE FASTER R-CNN DETECTORS WITH PRETRAINED CONVOLUTIONAL NEURAL NETWORKS FOR DETECTING VERY SIMPLE OBJECTS THAT CAN BE MASKED

V. Romanuke

Polish Naval Academy,
9 Śmidowicza Street, Gdynia, Poland, 81-127, e-mail: romanukevadimv@gmail.com

One of the best object detection methods, the Faster R-CNN, uses a pretrained convolutional neural network allowing to train the detector on small training sets typical in the object detection practice. Convolutional networks are prevented from overfitting by inserting DropOut layers. An open question is whether the DropOut technique improves much the object Faster R-CNN detector accuracy. Therefore, the goal is to show how the DropOut technique influences on the object detector performance. An original image classification dataset for pretraining a convolutional neural network is CIFAR-10. An appropriate convolutional network architecture for classifying CIFAR-10 images has a 50 % DropOut layer inserted in-between two fully-connected layers. Object detection tasks used for training and testing the Faster R-CNN detector are of monochrome images wherein small black rectangles are to be detected. Despite such objects are very simple, they can be masked around some dark localities so that detection would not be easy. One detection task is to detect the black rectangles in suburb house frontal views. Another one is to detect the rectangles in office room views. The suburb view dataset is divided into a training set of 120 images and a testing set of 121 images, every entry with a black rectangle. The office view dataset is divided likewise into a training set of 115 images and a testing set of 100 images, every entry with a black rectangle. Performance of the detector is studied against three training parameters: bounding box overlap ratio for positive training samples, minimum anchor box size, and anchor box pyramid scale factor. The performance is meant by the number of detected objects along with the intersection-over-union. However, neither graphs for the summed intersection-over-unions, nor graphs for the number of detected objects show that the DropOut technique influences on the Faster R-CNN object detector performance. Even for letting miss a few objects and decreasing an accuracy threshold, this influence is not significant. Therefore, a pretrained convolutional neural network to be included into the Faster R-CNN object detector should not contain a DropOut layer, especially if the network is trained much longer with the DropOut layer.

Key words: object detection, Faster R-CNN object detector, pretrained convolutional network, DropOut, monochrome image, training set, intersection-over-union, number of detected objects.

DROPOUT LAYERS FOR PREVENTING OVERFITTING

Overfitting is a poor effect of an analysis over data that involves too complex modeling. In machine learning, overfitting is especially likely for small training data sets and, additionally, when training is performed overlong. To prevent overfitting, a technique named dropout (or, stylistically, DropOut) is used [1, 2]. In convolutional neural networks (CNNs), DropOut is a layer, at which a part of nodes is randomly removed while training. The randomness of the remover is defined with a probability [2, 3]. This probability is commonly equal to 0.5, unless a very specific model is considered. Thus, the DropOut layer works as the remover at each training stage, dropping out a half of individual activations chosen randomly. After training, the removed nodes are reinserted into the network with their original weights.

A lot of studies considering DropOut exploitation exist. They have been trying to achieve high performance with DropOut rather than to explain reasons for inserting or canceling DropOut layers. Recently, the technique of DropOut was considered in [2] with CNNs for image classification, aiming at finding a rule of rationally allocating DropOut layers of 0.5 rate for maximizing performance. Two common network architectures having either 4 or 5 convolutional layers were used for that, benchmarking with CIFAR-10, EEACL26, and NORB datasets [4, 5]. A compromising rule was found as to non-compactly insert a few DropOut layers before the last convolutional layer. The work [2] factually claims that the rule “prefers” less number of DropOut layers. Eventually, the exemplary gain of the rule application was roughly between 10 % and 50 % signifying that the DropOut technique rationally applied is much influential in fine-tuning CNNs.

CONVOLUTIONAL NEURAL NETWORKS FOR OBJECT DETECTORS

CNNs are widely used for image classification and object detection. The R-CNN method is an early application of CNNs to object detection [6], wherein a pretrained CNN is included into the object detector. The pretrained CNN allows using small training sets that is typical in the object detection practice. An enhanced technique is the Fast R-CNN method [7], which is furthered to the Faster R-CNN method [8].

Selecting a pretrained CNN for including it into a Faster R-CNN detector architecture is a separate task. In fact, the pretrained CNN is used for the detector in the sense of the transfer learning workflow [9]. In transfer learning, a CNN used as the starting point to solve a new classification or detection task is to be trained on a dataset of diverse and heterogeneous images [2, 4, 5]. This is intended for that the pretrained CNN may learn a rich aggregate of image features that are applicable to a wide range of images. Deepness of the learning is still arguable, i. e. it is an open question whether the CNN should be trained to its top accuracy or not. It is quite important because a deeper training lasts longer and consumes more resources. Anyway, the learning is transferable to the new task by fine-tuning the pretrained CNN: the feature representations learned for the original task are slightly adjusted for supporting the new task (in this case, an object detection task). These slight adjustments may be executed by a small training set [9, 10], which is only available for the object detector.

An interesting question is, if the pretrained CNN has DropOut layers, does the DropOut technique improve much the object detector accuracy? If it does not, then training the CNN with DropOut layers would be senseless because training without DropOut layers is significantly faster and easier.

GOAL OF ARTICLE AND STAGES TO ACHIEVE IT

Generally, the goal is to show how the DropOut technique influences on the object detector performance considering the Faster R-CNN method, which is one of the best object detection methods. The object detector performance is not just an average accuracy of detection, but also stability of detection implying how badly accuracies related to individual objects are scattered. For achieving the said goal, the five stages are to be executed:

1. Substantiating an original image classification task (dataset) that is going to be used for pretraining a CNN.

2. Defining a CNN architecture for the substantiated task (dataset), with DropOut and without it.

3. Substantiating an object detection task (dataset) that is going to be used for training and testing the Faster R-CNN detector.

4. Comparing the Faster R-CNN detector performance by the CNN with DropOut and without it.

5. Discussing and concluding on how the DropOut technique influences on the Faster R-CNN object detector performance.

Both the original image classification dataset and object detection dataset should be as general as possible for covering plausibility and generalization of the detector's performance results. A method of measuring the detector's performance will be suggested at the fourth stage. At the last stage, unanswered points should be emphasized.

ORIGINAL IMAGE CLASSIFICATION DATASET FOR PRETRAINING A CNN

The original image classification dataset used for pretraining a CNN should be of very miscellaneous, diverse, and heterogeneous images.

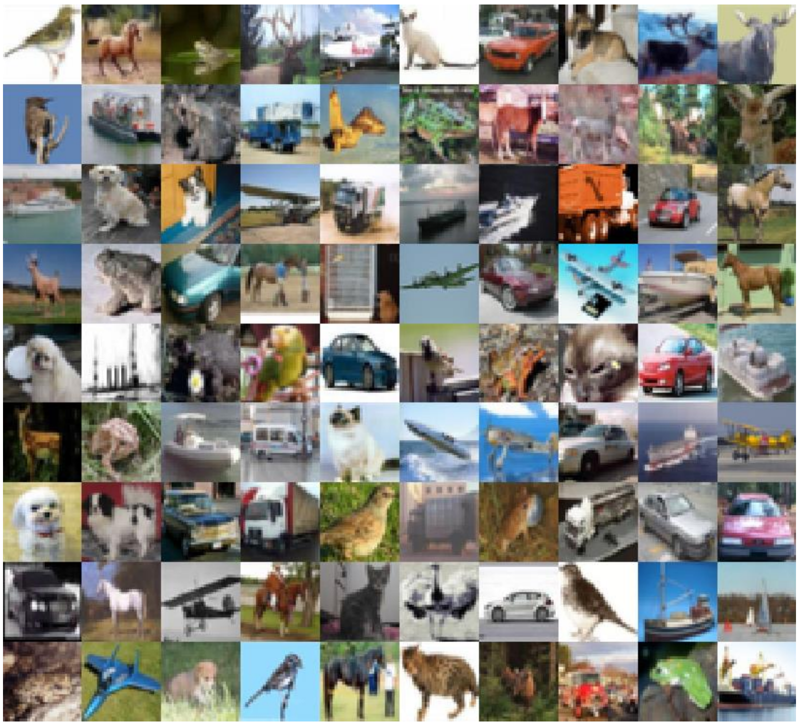

Fig. 1. A subset (90 randomly montaged images) of the CIFAR-10 dataset

One of the best and simplest examples of such dataset is the CIFAR-10 dataset (fig. 1). The CIFAR-10 dataset consists of 60,000 color $32 \times 32 \times 3$ images. These tiny images are represented as 32×32 matrix in each of the three color channels. The CIFAR-10 dataset, consisting of only 10 image categories ("airplane", "automobile", "bird", "cat", "deer", "dog", "frog", "horse", "ship", "truck"), is divided into 50,000 images intended for training and 10,000 images intended for validating and testing [2, 4, 5, 11, 12]. Thus, this is

6,000 images per image category, where 5,000 images per category are used to train, and 1,000 images per category are used to validate and test.

```
Image Input           32x32x3 images with 'zerocenter' normalization
Convolution           64 5x5x3 convolutions with stride [1  1] and padding [2  2  2  2]
ReLU                  ReLU
Max Pooling           3x3 max pooling with stride [2  2] and padding [0  0  0  0]
Convolution           128 5x5x64 convolutions with stride [1  1] and padding [2  2  2  2]
ReLU                  ReLU
Max Pooling           3x3 max pooling with stride [2  2] and padding [0  0  0  0]
Convolution           256 5x5x128 convolutions with stride [1  1] and padding [2  2  2  2]
ReLU                  ReLU
Max Pooling           3x3 max pooling with stride [2  2] and padding [0  0  0  0]
Fully Connected       64 fully connected layer
ReLU                  ReLU
Dropout               50% dropout
Fully Connected       10 fully connected layer
Softmax               softmax
Classification Output crossentropyex with 'airplane' and 9 other classes
```

Fig. 2. A CNN architecture for classifying CIFAR-10 images with a single DropOut layer, which is classically set at 50 % rate; max pooling layers have a purposely larger window and a greater stride by the zero padding, whereas the first three convolutional layers have a window of the same size by the minimal stride and a padding equal to 2

An appropriate CNN architecture for classifying CIFAR-10 images is shown in fig. 2, where a 50 % DropOut layer is inserted in-between two fully-connected layers. The starting number of filters in the first convolutional layer is 64, and then the number of filters is doubled through the layers [2]. Except for the final convolutional layer, every convolutional layer is followed by a rectified linear unit (ReLU), and this is close to the most appropriate versions of allocating ReLUs for classifying CIFAR-10 images [5]. All the max pooling layers have a 3×3 window and stride of 2, additionally preventing overfitting and ensuring good generalization. A CNN trained for 80 epochs by such architecture performs at 84.5 % accuracy rate, which is sufficient for including this CNN into the Faster R-CNN detector architecture.

If the DropOut layer in the architecture in fig. 2 is removed, the CNN is trained significantly faster: it takes only 40 epochs to achieve 82.6 % accuracy rate. This accuracy now being slightly less is also sufficient for the pretrained CNN.

OBJECT DETECTION DATASET FOR THE FASTER R-CNN DETECTOR

It is apparent that objects within color images are detected easier than dealing with monochrome images. This is so owing to colors and their gradients help in detecting. Henceforward, for making an object detection task harder, a set of monochrome images fits. The harder object detection task will allow to obtain magnified results of the DropOut layer influence.

Suburb house frontal views represented in 241 monochrome 220×330 images constitute a pretty hard origin for an object detection task dataset (fig. 3). The image resolution is about medium (although not as low as the resolution of CIFAR-10 images), which is suitable for obtaining the detector performance results faster (much faster in comparison to color images whose resolution is close to standards of 480×640, 600×800 and so on).

Fig. 3. A subset (165 randomly montaged images) of suburb house frontal views

Despite the images in fig. 3 appear simple, they have a lot of small localities where an object could be placed so that its detection would not be very easy. The localities are either darker or lighter, and the object must be respectively darker or lighter. Then an effect of masking the object may come. A very simple case, which, however, must not simplify the object detection task itself, is to detect small black rectangles in those suburb view images. Such black rectangles of sizes from 30×30 to 50×50 are randomly placed in them. Thus, the suburb view dataset (SubVDS) in fig. 3 is divided into a training set of 120 images with black rectangles and a testing set of 121 images with black rectangles. An assembled image of the suburb view testing set is shown in fig. 4.

Fig. 4. The object detector testing set of 121 suburb house frontal views, where house dark windows and shadows (and other darkened localities) in some images mask the object

Fig. 5. The object detector testing set of 100 office room views (widths of the images have been resized to 327), where dark screens of monitors in some images may confuse

For strengthening diversity, a supporting dataset should be formed with similar black rectangles. This is about office room views represented in 215 monochrome $220\times W$ images by $W\in\left[241;392\right]\cap\mathbb{N}$. Note that here the width of the office room view images is inconstant. This makes additional difficulties in detecting, although they are not much influential. Such image dataset is divided into a training set of 115 images with black rectangles and a testing set of 100 images with black rectangles. An assembled image of the office room view testing set is shown in fig. 5.

Training parameters for both datasets are the same, except for the minibatch size equal to the number of training images, which is 120 for the SubVDS and 115 for the office room view dataset (OffRVDS). The number of epochs is set at 2.

THE DETECTOR PERFORMANCE WITH DROPOUT AND WITHOUT IT

When an object is detected, the detection confidence is desired to be as much as greater. On the one hand, all the objects in the testing set should be detected. On the other hand, the minimum of the detection accuracy should be greater than 0.5, which is acceptable for such a detection accuracy parameter as the intersection-over-union (IoU) [6].

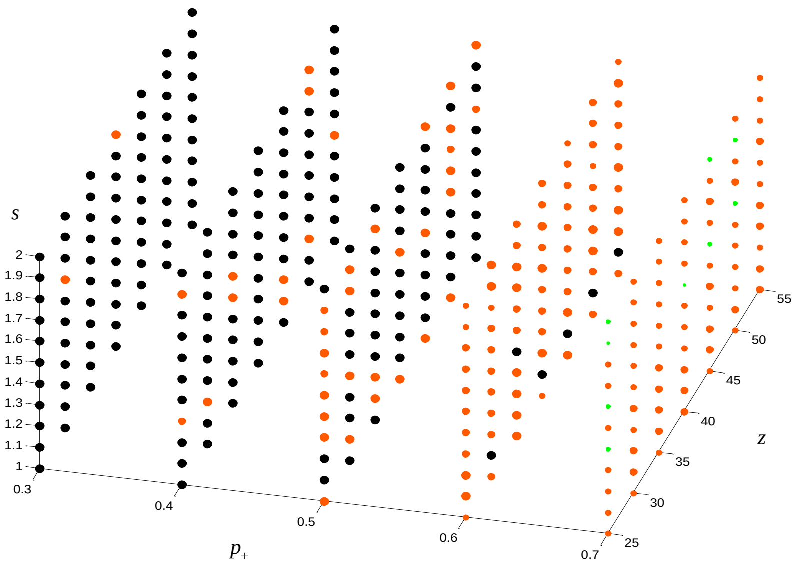

Fig. 6. The number of detected objects in the testing set of the SubVDS with using DropOut (see fig. 2)

While trained, three parameters of the detector training process are varied. They are bounding box overlap ratio p_+ for positive training samples, minimum anchor box size z, anchor box pyramid scale factor s. Performance of the detector will be shown below against these parameters. As the performance is a function of three variables, then it will be plotted by using a few colors and different thickness of dots dependent on the detection accuracy and number of detected objects. The number of detected objects is represented in fig. 6, 8, 10, 12, where the trickiest dots colored black correspond to the maximum of the detected objects (which is 121 for the SubVDS, and 100 for the OffRVDS). Dots colored lighter correspond to a lesser number of the detected objects (this is between 115 and 120 for the SubVDS, and is between 96 and 99 for the OffRVDS). The lightest dots correspond to a number of the detected objects less than 95 % (i. e., less than 115 for the SubVDS, and less than 96 for the OffRVDS). The sum of IoUs is represented in fig. 7, 9, 11, 13, where dots are plotted black only if the minimal IoU is greater than 0.5. If the minimal IoU is equal to 0, then the dot is plotted the lightest. The dot thickness is proportional to the sum.

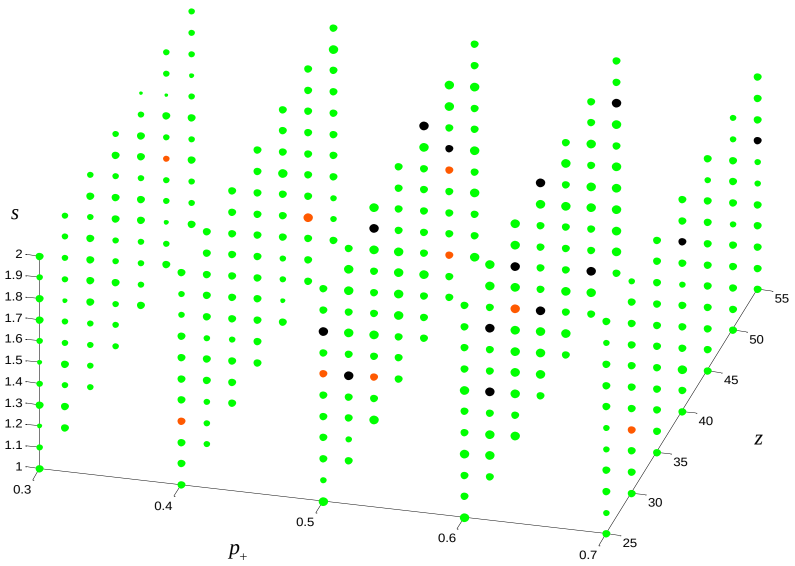

Fig. 7. The summed IoUs for the testing set of the SubVDS with using DropOut (see fig. 2)

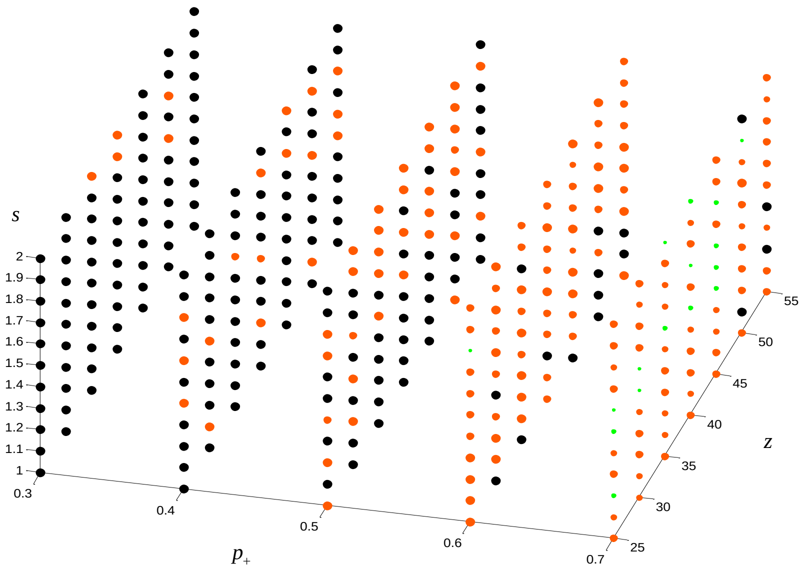

Fig. 8. The number of detected objects in the testing set of the SubVDS without using DropOut

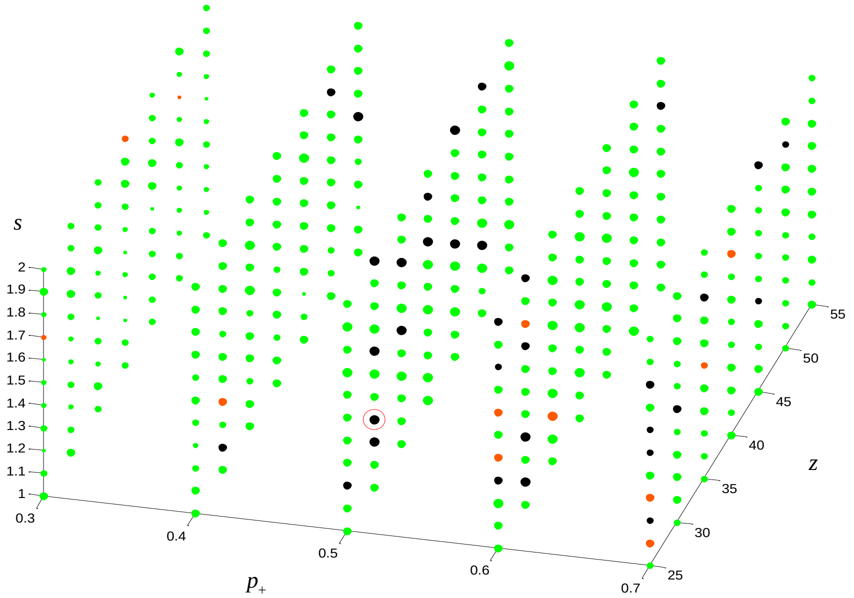

Fig. 9. The summed IoUs for the testing set of the SubVDS without using DropOut (the flawless dot is encircled)

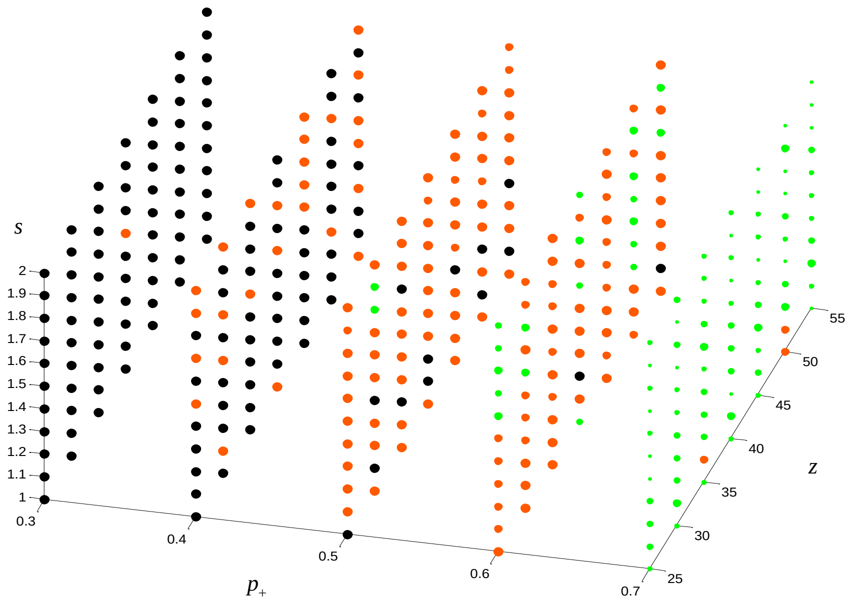

Fig. 10. The number of detected objects in the testing set of the OffRVDS with using DropOut (see fig. 2)

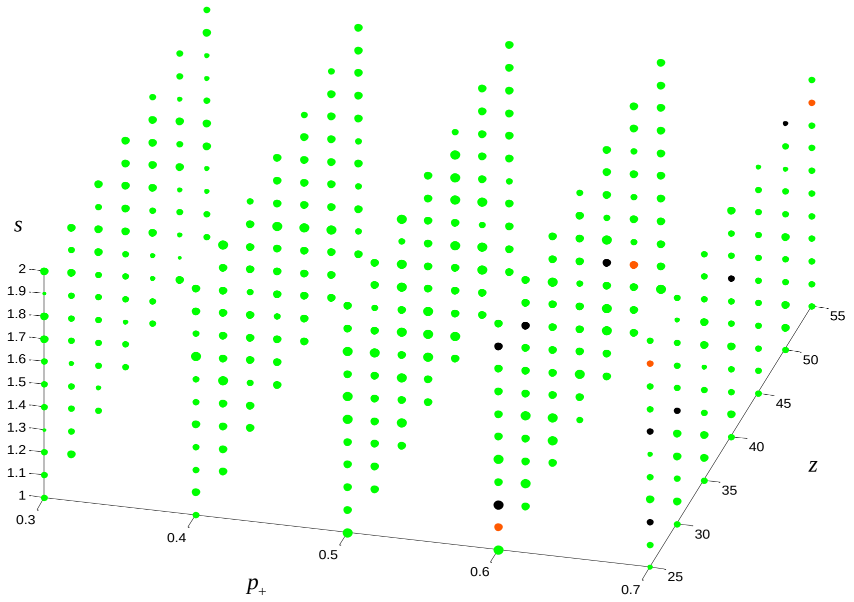

Fig. 11. The summed IoUs for the testing set of the OffRVDS with using DropOut (see fig. 2)

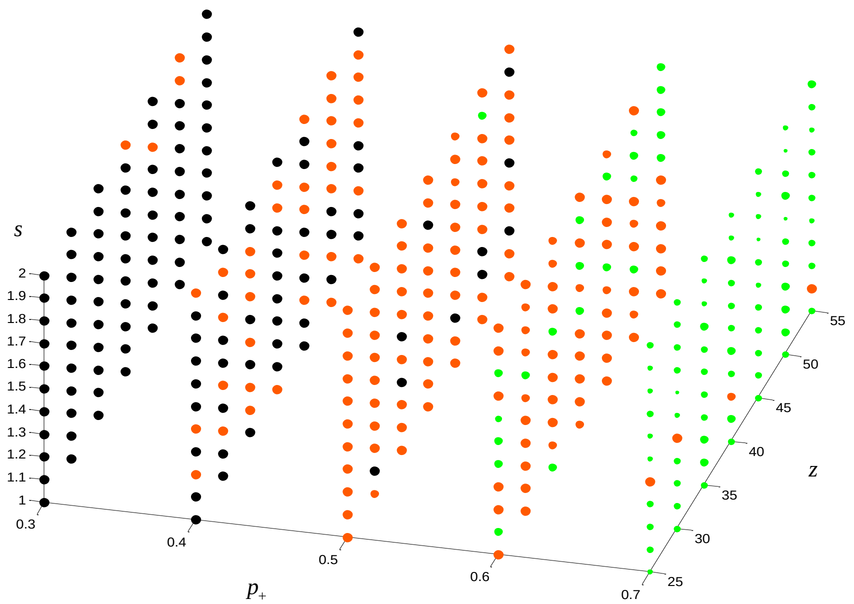

Fig. 12. The number of detected objects in the testing set of the OffRVDS without using DropOut

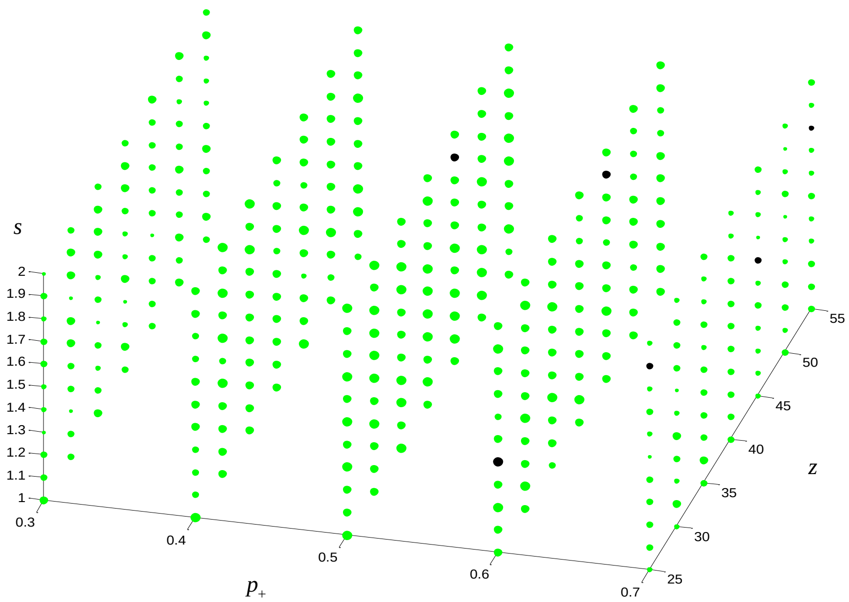

Fig. 13. The summed IoUs for the testing set of the OffRVDS without using DropOut

It is quite apparent that the performance results of detectors without DropOut are very similar to those of detectors used DropOut. The performance for the OffRVDS is poorer than that for the SubVDS (it is well seen if to compare fig. 10 to fig. 6 and fig. 12 to fig. 8, paying attention to faces at $p_+ = 0.7$). Detectors are trained faster without DropOut: on average, it is 14 % faster for the SubVDS and 19 % faster for the OffRVDS. Moreover, detectors are tested faster without DropOut as well: on average, it is 11 % faster for the SubVDS and 20 % faster for the OffRVDS.

There are 362 detectors (out of grand total 385 detectors) used DropOut for the SubVDS which have zero IoUs, whereas 340 detectors without DropOut for the SubVDS produced zero IoUs. The situation is a kind of reverse for the OffRVDS: 372 detectors used DropOut have zero IoUs, but there are 379 detectors produced zero IoUs without DropOut.

There is still only single detector (out of 385 detectors) which detected all 121 test objects of the SubVDS, without using DropOut, at a high accuracy rate where the minimal IoU was greater than 0.5 (see fig. 9). The high accuracy rate is meant to be not less than 97 % of the maximal sum of IoUs. However, the number of detectors against the maximal number of their object omissions does not seem better without using DropOut, for both the SubVDS (fig. 14) and OffRVDS (fig. 15). This is the most contradictory result, and it is going to be confirmed below.

If to loosen the requirement of detecting the whole testing set, at the same greater-than-0.5 minimal IoU and not-less-than-97 % of the maximal sum of IoUs, the number of flawless detectors increases. Aiming at detecting only 120 out of 121 objects in the testing set of the SubVDS, there are 4 and 5 flawless detectors, respectively using and without using DropOut. Such let-miss-one-object loosening does not affect the OffRVDS.

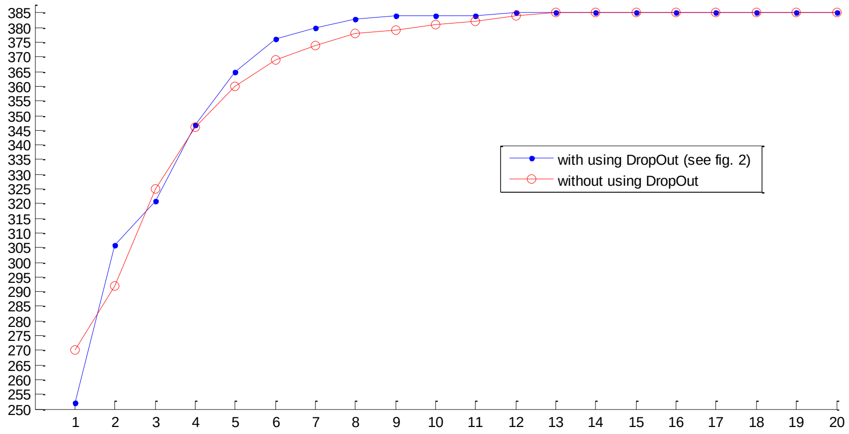

Fig. 14. SubVDS: the number of detectors against the maximal number of their object omissions (abscissa axis)

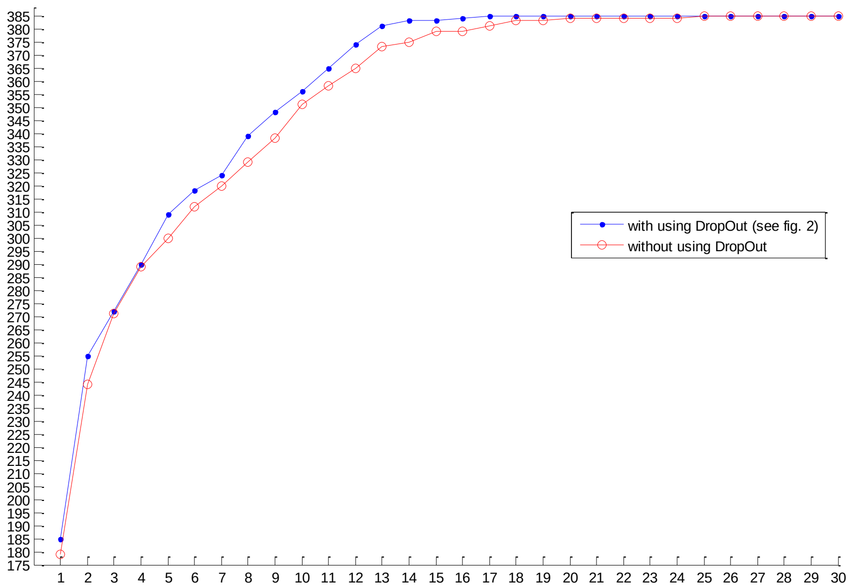

Fig. 15. OffRVDS: the number of detectors against the maximal number of their object omissions (abscissa axis)

Nonetheless, if to decrease that not-less-than-97 % down to 95 %, the detector without DropOut for the SubVDS gains more: 4 against 12. The number of flawless detectors with DropOut remains the same by decreasing it down to 90 %, whereas the number of flawless detectors without DropOut becomes 18. For the not-less-than-95 % and letting miss two objects, there are 11 and 14 flawless detectors of the SubVDS, respectively using and without using DropOut. For the not-less-than-93 %, this is 13 and 18. For the not-less-than-90 %, these respective numbers are 13 and 20, remaining the same by further decreasing.

The OffRVDS is much rougher: there is still only single detector which detected 98 OffRVDS test objects, without using DropOut, at not-less-than-95 % of the maximal sum of IoUs. The same result remains by decreasing it down to 90 % and looser, or letting miss even three objects. The further loosening reveals that here the detector with DropOut has its minimal gain. However, this gain is pretty insignificant because the numbers of flawless detectors are fewer. Namely, at the not-less-than-90 %, they are 2 and 1 by letting miss up to four objects, 3 and 2 by letting miss up to five objects, 3 and 3 by letting miss up to six objects. When the number of missed objects is seven or more, those respective numbers are 4 and 3.

DISCUSSION AND CONCLUSION

Neither the graphs for the summed IoUs (fig. 7, 9, 11, 13), nor the graphs for the number of detected objects (fig. 6, 8, 10, 12) have shown that the DropOut technique influences on the Faster R-CNN object detector performance. Letting miss a few objects, this influence is not significant for the SubVDS at the not-less-than-97 % (fig. 14), although detectors with using DropOut have a tiny advantage for the OffRVDS (fig. 15). Decreasing that not-less-than-97 % down has revealed that the DropOut technique can be “dropped out” for the SubVDS. Using the DropOut technique does not help much with the OffRVDS.

The object detection datasets used here have both similar and dissimilar features. Owed to a lot of images of these datasets mask the black rectangles, the performance results recapitulated above must be really perceived magnified. Consequently, the DropOut technique does not make any significant improvement of the Faster R-CNN object detector performance. Therefore, a pretrained CNN to be included into the Faster R-CNN object detector should not contain a DropOut layer, especially if the CNN is trained much longer with the DropOut layer. An exclusion may be for very simple original image classification datasets (like MNIST [13, 14] or EEACL26 [2, 4, 5]), where overfitting is prevented with two or even more DropOut layers and they do not retard the training process.

REFERENCES

1. *Hagiwara K.* Relation between weight size and degree of over-fitting in neural network regression / K. Hagiwara, K. Fukumizu // Neural Networks. – 2008. – Vol. 21, Iss. 1. – P. 48-58.
2. *Romanuke V.V.* Appropriateness of DropOut layers and allocation of their 0.5 rates across convolutional neural networks for CIFAR-10, EEACL26, and NORB datasets / V. V. Romanuke // Applied Computer Systems. – 2017. – Vol. 22. – P. 54-63.
3. *Elleuch M.* A new design based-SVM of the CNN classifier architecture with dropout for offline Arabic handwritten recognition / M. Elleuch, R. Maalej, M. Kherallah // Procedia Computer Science. – 2016. – Vol. 80. – P. 1712-1723.
4. *Romanuke V.V.* Appropriate number of standard 2×2 max pooling layers and their allocation in convolutional neural networks for diverse and heterogeneous datasets / V.V. Romanuke // Information Technology and Management Science. – 2017. – Vol. 20. – P. 1219.
5. *Romanuke V.V.* Appropriate number and allocation of ReLUs in convolutional neural networks / V.V. Romanuke // Research Bulletin of NTUU “Kyiv Polytechnic Institute”. – 2017. – No. 1. – P. 69-78.
6. *Girshick R.* Rich feature hierarchies for accurate object detection and semantic segmentation / R. Girshick, J. Donahue, T. Darrell, J. Malik // Proceedings of the 2014

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 23-28, 2014. – P. 580-587.
7. *Girshick R.* Fast R-CNN / R. Girshick // Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), December 7-13, 2015. – P. 1440-1448.
8. *Ren S.* Faster R-CNN: towards real-time object detection with region proposal networks / S. Ren, K. He, R. Girshick, J. Sun // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2017. – Vol. 39, Iss. 6. – P. 1137-1149.
9. *Han D.* A new image classification method using CNN transfer learning and web data augmentation / D. Han, Q. Liu, W. Fan // Expert Systems with Applications. – 2018. – Vol. 95. – P. 43-56.
10. *Khatami A.* A sequential search-space shrinking using CNN transfer learning and a Radon projection pool for medical image retrieval / A. Khatami, M. Babaie, H.R. Tizhoosh, A. Khosravi, T. Nguyen, S. Nahavandi // Expert Systems with Applications. – 2018. – Vol. 100. – P. 224233.
11. *Krizhevsky A.* ImageNet classification with deep convolutional neural networks / A. Krizhevsky, I. Sutskever, G. E. Hinton // Advances in Neural Information Processing Systems. – 2012. – Vol. 1. – P. 1097-1105.
12. *Date P.* Design index for deep neural networks / P. Date, J.A. Hendler, C.D. Carothers // Procedia Computer Science. – 2016. – Vol. 88. – P. 131138.
13. *Romanuke V.V.* Training data expansion and boosting of convolutional neural networks for reducing the MNIST dataset error rate / V. V. Romanuke // Research Bulletin of NTUU "Kyiv Polytechnic Institute". – 2016. – No. 6. – P. 29-34.
14. *Romanuke V.V.* Smooth non-increasing square spatial extents of filters in convolutional layers of CNNs for image classification problems / V.V. Romanuke // Applied Computer Systems. – 2018. – Vol. 23. – P. 52-62.

Стаття: надійшла до редколегії 23.08.2018
доопрацьовано 10.10.2018
прийнята до друку 31.10.2018

ДОСЛІДЖЕННЯ ТЕХНІКИ DROPOUT ДЛЯ ДЕТЕКТОРІВ FASTER R-CNN З НАПЕРЕД НАВЧЕНИМИ ЗГОРТКОВИМИ НЕЙРОННИМИ МЕРЕЖАМИ ДЛЯ ВИЯВЛЕННЯ ДУЖЕ ПРОСТИХ ОБ'ЄКТІВ, ЩО МОЖУТЬ БУТИ ЗАМАСКОВАНИМИ

В. Романюк

Військово-морська академія Польщі,
вул. Шмідовича, 69, Гдиня, Польща, 81-127, e-mail: romanukevadimv@gmail.com

Метод виявлення об'єктів Faster R-CNN використовує наперед навчену згорткову нейронну мережу для того, щоб навчати детектор на малих навчальних множинах, типових для практики виявлення об'єктів. Перенавчанню в згорткових мережах запобігають за допомогою вставки шарів DropOut. Відкритим питанням є те, чи техніка DropOut набагато поліпшує точність детектора об'єктів Faster R-CNN. Тому наша мета – з'ясувати, як техніка DropOut впливає на продуктивність детектора об'єктів. Первинним набором даних класифікації зображень для попереднього навчання згорткової нейронної мережі є CIFAR-10. Відповідна архітектура згорткової мережі для класифікації зображень CIFAR-10 має 50 % шар DropOut, вставлений між двома повнозв'язними шарами. Задачі виявлення об'єктів, використовувані для навчання і тестування детектора Faster R-CNN, складаються з монохромних зображень, на яких мають бути виявлені невеликі чорні прямокутники. Незважаючи на те, що такі об'єкти –

дуже прості, вони можуть бути замаскованими поблизу деяких темних ділянок так, щоб виявлення не було легким. Одна задача полягає у виявленні чорних прямокутників на фронтальних зображеннях приміських будинків. Інша – у виявленні таких прямокутників на зображеннях офісних кімнат. Набір даних приміських зображень поділений на навчальну множину зі 120 зображень і тестову множину зі 121 зображення, з чорним прямокутником у кожному зображенні. Набір даних офісних зображень поділений так само на навчальну множину зі 115 зображень і тестову множину зі 100 зображень, з чорним прямокутником у кожному зображенні. Продуктивність детектора досліджують за трьома параметрами навчання: співвідношення перекриття обмежувальних прямокутників для позитивних навчальних зразків, розмір мінімального прямокутника прив'язки і коефіцієнт пірамідального масштабування обмежувальних прямокутників. Під продуктивністю розуміють кількість виявлених об'єктів разом зі співвідношенням перетину й об'єднання. Однак ні графіки для сум співвідношень перетину й об'єднання, ні графіки для кількості виявлених об'єктів не продемонстрували, щоб техніка DropOut впливала на продуктивність детектора об'єктів Faster R-CNN. Навіть пропустивши кілька об'єктів та знижуючи поріг точності, такий вплив видається незначним. Відтак наперед навчена згорткова нейронна мережа для включення її у склад детектора об'єктів Faster R-CNN не повинна містити шару DropOut, особливо якщо ця мережа з шаром DropOut навчається набагато довше.

Ключові слова: виявлення об'єктів, детектор об'єктів Faster R-CNN, наперед навчена згорткова мережа, DropOut, монохромне зображення, навчальна множина, співвідношення перетину й об'єднання, кількість виявлених об'єктів.