THE STUDY OF ACTIVATION FUNCTIONS IN DEEP LEARNING FOR PEDESTRIAN DETECTION AND TRACKING

M. N. Favorskaya [1,*], V. V. Andreev [1]

[1] Reshetnev Siberian State University of Science and Technology, Institute of Informatics and Telecommunications, 31, Krasnoyarsky Rabochy ave., Krasnoyarsk, 660037 Russian Federation - favorskaya@sibsau.ru, jcjet88@gmail.com

Commission II, WG II/5

KEY WORDS: Deep Learning, Activation Function, Pedestrian Detection, Feature Extraction, Pedestrian Tracking

ABSTRACT:

Pedestrian detection and tracking remains a highlight research topic due to its paramount importance in the fields of video surveillance, human-machine interaction, and tracking analysis. At present time, pedestrian detection is still an open problem because of many challenges of image representation in the outdoor and indoor scenes. In recent years, deep learning, in particular Convolutional Neural Networks (CNNs) became the state-of-the-art in terms of accuracy in many computer vision tasks. The unsupervised learning of CNNs is still an open issue. In this paper, we study a matter of feature extraction using a special activation function. Most of CNNs share the same architecture, when each convolutional layer is followed by a nonlinear activation layer. The activation function Rectified Linear Unit (ReLU) is the most widely used as a fast alternative to sigmoid function. We propose a bounded randomized leaky ReLU working in such manner that the angle of linear part with the highest input values is tuned during learning stage, and this linear part can be directed not only upward but also downward using a variable bias for its starting point. The bounded randomized leaky ReLU was tested on Caltech Pedestrian Dataset with promising results.

1. INTRODUCTION

The issues of pedestrian detection and tracking have become an important area in computer vision since 1990s (Girshick et al., 2014). Various techniques were proposed in each of three main application fields of pedestrian detection, i.e. video surveillance, human-machine interaction, and analysis of captured motion data in different clinical studies. Generally, three fundamental steps, such as image acquisition, feature extraction, and classification, are involved into a process of vision-based human detection. The deep learning architectures inspired to the human visual cortex allow to remove the feature extraction step (Cao et al., 2016; Jiang et al., 2016). However, the extraction of features cannot be completely removed from the CNN workflow. These features are computed at different layers of abstraction that, on the one hand, revokes a handcrafted feature construction but, on the other hand, results in a longer training time of the CNN.

A lot of works dealing with pedestrian detection and tracking could be found in literature. First of all, a Region Of Interest (ROI) ought to be found that reduces the processing volume data significantly. The supervised learning of CNN requires a manual extraction of ROIs in images of frames. The unsupervised learning implies an automatic extraction of the ROIs with the required content. Note that ROI extraction depends principally from the both shooting (camera resolution, field of view) and environmental (luminance, weather) conditions.

In recent years, a multiple number of human detection and tracking algorithms were developed and the most of them are based on the following approaches:

1. Histograms of Oriented Gradients (HOG) is based on the idea that the local intensity gradients or edge directions distribution are described a local moving object (Dalal and Triggs, 2005). Each frame is divided into small regions, and the gradient direction based on 1D HOG or edge orientation is computed for each region. Numerous modifications of basic HOG are available in literature. However, in all versions a cascade of classifiers is used to discriminate each sub-region.
2. Haar-like features approach is a wavelet transform of the structural similarities between various instances (Zhang et al., 2014). The 2D Haar wavelets provide the basic functions, which detect changes in intensity along the horizontal, vertical, and two diagonals (or corners) directions. The obtained representations are utilized as an input to a classifier.
3. Viola–Jones features (Viola et al., 2003) are taken by the extended rectangle filters based on Haar wavelets. This approach considers both motion and intensity information in the consecutive frames.
4. Combination of low level features, such as texture, shape, colour, and contrast, is a quite simple approach based on their distributions in an image (Munder et al., 2008). Sometimes, these studies include additionally salient keypoints.
5. Local Binary Pattern (LBP) technique has become very popular due to its robustness respect to luminance and human pose variations. The LBP features are often combined with the HOG features for higher evaluations of pedestrian detection (Wang et al., 2009).
6. Deep learning is a recent approach, which is intensively developed in many issues of computer science including pedestrian tracking (Li P. et al., 2018).

In this paper, we consider a video surveillance through achievements of the deep learning methodologies, including CNNs.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-2-W12-53-2019 | © Authors 2019. CC BY 4.0 License.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 introduces the existing ReLU activation functions. Section 4 shows our modification of randomized ReLU activation function in order to decide two problems – the robustness to noise and overfitting of the CNN. Section 5 covers the experimental results and Section 6 concludes this paper.

2. RELATED WORK

Since the pedestrian detection and tracking have a high computational cost, the classifier models with low complexity, such as linear SVM or weak decision trees with low depths, are often employed. A deep learning framework for these tasks expends the classifiers to boosting, naive Bayes, multiple instance learning, metric learning, structured learning, latent variable learning, and correlation filter. The existing CNN trackers are categorized into generative and discriminative methods (Tian et al., 2015; Tomѐ et al., 2016; Xue et.al., 2016; Raza et al., 2018). Generative methods focus on searching for the ROI using handcrafted features, while discriminative methods interpret a tracking as a classification problem in a local surrounding background. According to this classification, deep networks are named as Feature Extraction Network (FEN), which extracts deep features and adopts the conventional method for learning, and End to End Network (EEN), which uses not only feature extraction but also a candidate evaluation. The outputs of the ENN can be probability map, heat map, candidate’s score, object position, and even bounding box. Note that attempts are being taken to a fusion between both discriminative and generative models (Pang et al., 2017).

Popular generative methods include kernel-based tracker, Gaussian mixture model-based tracker, subspace-based tracker, covariance-based tracker, low-rank and sparse representation-based trackers, and visual tracking decomposition. However, the most of publications in scope of pedestrian detection and tracking are devoted to the discriminative models. The proposed by Li H. et al. (Li H. et al., 2018) tracker was composed of two major components. A deep correlation filter adopted CNN to generate a robust representation of the context around the target. The online discriminative learning method trained the translating and scaling models to refine the coarse predictions of deep correlation filter. A combination of these two components allowed to create a precise tracker. Instead of using pre-trained VGG-19 Network (network invented by Visual Geometry Group from University of Oxford) directly, the authors modified its architecture and enhanced its representation ability via an offline learning strategy using ReLU layer.

Firstly, Tomѐ et al. (Tomѐ et al., 2016) trained ImageNet, which is successfully employed for object detection. Then, these authors exploited an annotated training dataset of positive and negative regions in order to fine tune the weights of the CNN and the classifier. An ad-hoc pedestrian detection algorithm called as Locally Decorrelated Channel Features (LDCF) was implemented (Nam et al., 2014). The output of LDCF included a large set of regions with the confidence scores. The higher the confidence score of a region, more likely such region contained a pedestrian. Threshold values of the confidence score allowed a trade-off-between precision and recall.

The advantages of deep learning and particle filtering were demonstrated by Qian et al. (Qian et al., 2018). They designed a substantially smaller CNN with two convolutional layers, two pooling operations, and one fully connected layer. For convolutional layers, the sigmoid activation function was adopted. The authors achieved good results by pre-training a simplified CNN using a large set of videos with tracking ground truths. Particle propagation was employed by a dynamic model considering the velocity and acceleration. The algorithm updated the tracking model from time to time in order to avoid shift and expensive computation.

In recent years, an aim of faster performance comparing to deep learning networks leads to lightweight solutions, for example SqueezeNet or MobieNet. Thus, inspired by the depthwise separable convolution and Single Shot multi-box Detector (SSD), a Lightweight CNN (L-CNN) was designed by Nikouei et al. (Nikouei et al., 2018). The SSD designed for object detection in real-time is faster than Region CNN (R-CNN), Fast R-CNN, and even Faster R-CNN (Liu et al., 2016). The L-CNN network architecture had 26 layers considering depthwise and pointwise convolutions as separate layers. The final classifier, softmax, and regression layers provided a bounding box around the detected object. A simple fully connected neural network classifier took the prior probabilities of each window of objects, identified the objects within the proposed window, and added the label for output bounding boxes at the end of the network.

The main idea of fusion both discriminative and generative models is to combine the motion object location with target verification. For this purpose, Pang et al. (Pang et al., 2017) proposed a Deep Framer Network (DFN) architecture initially based on AlexNet deep model because it was trained on small size images (227 × 227) concerted with the size of image patches in target tracking. First, a deep learning network was used to obtain a discriminative object location model in a keyframe. Second, the authors constructed a matching score function to verify, which object in the current frame matched the target object set in a keyframe. The ReLU activation function was implied due to its possibility to improve the learning speed and classification accuracy regarding the conventional sigmoid and tangent functions.

The CNN may be used not only as a tracker but also for automated pedestrian intention and behavior analysis (Raza et al., 2018). A supervised CNN was designed aiming a prediction of appearance-based pedestrian head-pose and full-body orientation under assumption that the head-pose and the direction of pedestrian’s movement are occasionally weakly correlated. Two CNN models, which included convolution, down-sampling, regularization, dropout, fully connected, and softmax layers, predicted the head position and full body orientation. The final step of the training phase consisted in a trained CNN classifier. Conventional ReLU function was applied in these CNNs.

3. ACTIVATION FUNCTIONS

The CNN mainly consists of three basic components: convolutional, pooling, and fully-connected layers, among which the convolutional layers have a significant influence on feature extraction. The activation functions introduce nonlinearities to the CNN, which are desirable to detect nonlinear features. Among the non-linear functions, ReLU is a popular activation function used in CNNs (LeCun et al., 2015). Since it is often rare to have the cortical neurons in their maximum saturation regime, it was argued that the ReLU is more biologically plausible than the standard sigmoid function (Glorot et al., 2011). Due to its faster learning speed, the ReLU is often employed, more than the smooth non-linear activation functions – sigmoid, hyperbolic tangent function, and logistic function. The definition of ReLU is given by equation 1:

$$f(x_k) = \begin{cases} x_k & \text{if } x_k > 0 \\ 0 & \text{if } x_k \leq 0 \end{cases} \quad (1)$$

where x_k = input of ReLU on *k*th channel
$f(x_k)$ = output of ReLU on *k*th channel

Equation 1 can be rewritten as:

$$f(x_k) = \max(x_k, 0)$$

As follows from equation 1, a non-linearity of ReLU is achieved by the hard threshold zero. Generally, the ReLU function is computed efficiently because it compares only two values. The ReLU has a sparse activation probability that creates sparse representation of data useful for classification. Also, the ReLU does not suffer from the gradient diffusion problem as much as sigmoid functions do. It is differentiable at any point except at the origin (piecewise differentiable).

A potential disadvantage of the ReLU is that it has zero gradients whenever the unit is not active. This may cause that some units, which were not active initially, will be never active as the gradient-based optimization will not adjust their weights. Additionally, a training process may slow down due to the constant zero gradients. To alleviate these problems, a Leaky ReLU (LReLU) function was proposed to allow a small, non-zero gradient value whenever the neuron is inactive (Maas et al., 2013). Thus, the LReLU obtained a small non-zero slope to the negative parts:

$$f(x_k) = \begin{cases} x_k & \text{if } x_k > 0 \\ \lambda x_k & \text{if } x_k \leq 0 \end{cases} \quad (2)$$

where λ = predefined parameter in range [0...1], usually $\lambda = 0.01$

Equation 2 can be rewritten as

$$f(x_k) = \max(x_k, 0) + \lambda \min(x_k, 0)$$

Compared with ReLU, the LReLU compresses the negative part that provides a small, non-zero gradient when the unit is not active.

Afterwards, He et al. (He et al., 2015) proposed so called Parametric ReLU (PReLU) aiming to learn the slope of the negative parts during the training stage:

$$f(x_k) = \begin{cases} x_k & \text{if } x_k > 0 \\ \lambda_k x_k & \text{if } x_k \leq 0 \end{cases} \quad (3)$$

where λ_k = the learned parameter for the *k*th channel

As the PReLU introduces a very small number of parameters, e.g. equaled to the number of channels of the whole network, there is no risk of overfitting and additional computational cost. The PReLU can be simultaneously trained with other parameters by back propagation algorithm.

Another variant of Leaky ReLU is Randomized Leaky ReLU (RReLU) (Xu et al., 2015). The RReLU can reduce overfitting due to its randomized nature. Formally, RReLU is defined as:

$$f\left(x_k^{(n)}\right) = \begin{cases} x_k^{(n)} & \text{if } x_k^{(n)} > 0 \\ \lambda_k^{(n)} x_k^{(n)} & \text{if } x_k^{(n)} \leq 0 \end{cases} \quad (4)$$

where n = *n*th example in training

Randomly Translational ReLU (RT-ReLU) has another meaning (Cao et al., 2018). It is found that the distribution of the ReLU inputs has a form of Gaussian distribution, and the most of the ReLU inputs are near zero. This makes the ReLU input very sensitive to the small jitter or the noise near zero. The ReLU and PReLU can be written in a view of randomly translation using equations 5 and 6:

$$f(x_k) = \begin{cases} x_k + a_k & \text{if } (x_k + a_k) > 0 \\ 0 & \text{if } (x_k + a_k) \leq 0 \end{cases} \quad (5)$$

$$f(x_k) = \begin{cases} x_k + a_k & \text{if } (x_k + a_k) > 0 \\ \lambda_k (x_k + a_k) & \text{if } (x_k + a_k) \leq 0 \end{cases} \quad (6)$$

where a_k = the shift of RT-ReLU along OX-axis

Clevert et al. (Clevert et al., 2016) introduced Exponential Linear Unit (ELU), which enabled faster learning of CNN and led to higher classification accuracies:

$$f(x_k) = \begin{cases} x_k & \text{if } x_k > 0 \\ \gamma\left(e^{x_k} - 1\right) & \text{if } x_k \leq 0 \end{cases} \quad (7)$$

where γ= predefined parameter for controlling the value, to which the ELU saturates for negative inputs, $\gamma > 0$

Equation 7 can be rewritten as:

$$f(x_k) = \max(x_k, 0) + \gamma \min\left(\left(e^{x_k} - 1\right), 0\right)$$

Also the basic ReLU was modified in the sense of the CNN robustness: Noisy ReLU (NReLU) (Nair and Hinton, 2010), Randomized ReLU (RReLU) (Xu et al., 2016), Noisy Activation Function (NAF) (Gulcehre et al., 2016).

In order to improve classification performances and training stability, Liew et al. (Liew et al., 2016) proposed the bounded variants of some ReLU functions. Thus, equations 1 and 2 have a view:

$$f(x_k) = \min(\max(x_k, 0), A) = \begin{cases} A & \text{if } x_k > A \\ x_k & \text{if } 0 < x_k \leq A \\ 0 & \text{if } x_k \leq 0 \end{cases} \quad (8)$$

$$f(x_k) = \begin{cases} \lambda x_k + (1 - \lambda) A & \text{if } x_k > A \\ x_k & \text{if } 0 < x_k \leq A \\ \lambda x_k & \text{if } x_k \leq 0 \end{cases} \quad (9)$$

where A = the maximum output value the function can produce

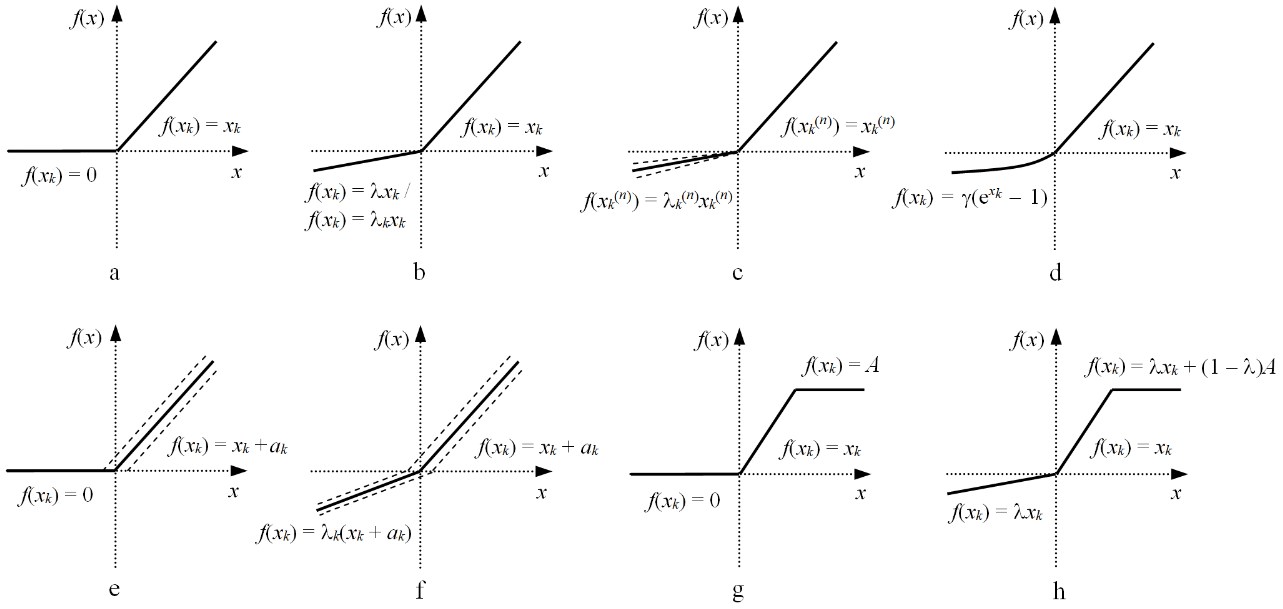

Figure 1. The ReLU activation functions: a ReLU, b LReLU/PReLU, c RReLU, d ELU e RT-ReLU, f RT-PReLU, g bounded ReLU, h bounded LReLU

A view of mentioned above ReLU activation functions are depicted in Figure 1. Note that a list of ReLU modifications can be further extended.

4. PROPOSED RELU MODIFICATION

As well-known, the CNN topology is the trade-off between the classification performance and training time. We try to enforce a classification performance using other components of CNN’s structure, particularly a view of activation functions. In Section 3, we examine the basic ReLU and its main modifications. However, the problem of overfitting, to which any deep architecture is easily prone, remains. In this study, we pay attention for two problems – the robustness to noise and overfitting. In this sense, we have proposed a bounded Randomized Leaky ReLU called as bounded RL-ReLU with a piecewise-linear structure described by equation 10:

$$f\left(x_k^{(n)}\right)=\begin{cases}\pm\lambda_k^{(n)}x_k^{(n)}+\left(1-\lambda_k^{(n)}\right)A & \text{if } x_k^{(n)}>A\\ x_k^{(n)} & \text{if } 0<x_k^{(n)}\le A\\ \lambda_k^{(n)}x_k^{(n)} & \text{if } x_k^{(n)}\le 0\end{cases}\qquad(10)$$

The bounded RL-ReLUs are depicted in Figure 2.

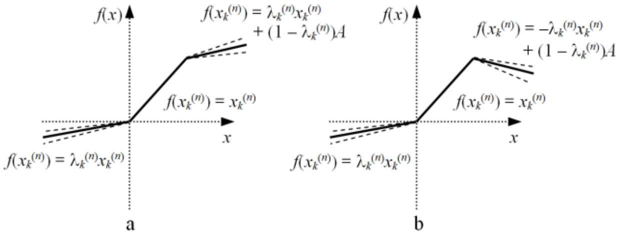

Figure 2. The bounded RL-ReLU with: a $+\lambda_k^{(n)}x_k^{(n)}$, b $-\lambda_k^{(n)}x_k^{(n)}$

We modify RReLU in such manner that the angle of linear part with the highest input values is tuned during learning stage, and this linear part can be directed not only upward but also downward using a variable bias for its starting point. Such modification allows to avoid an overfitting.

Noise adds nonlinearity in neural networks that may hide non-visible dependencies due to saturation of the activation function. One way is to inject noise into activation functions in their saturated regime (Gulcehre et al., 2016). We used another way, when a level of noise is adapted for each sample (see superscripts in equation 10) during a training mode. Then an averaged value of a level of noise is calculated and adopted with each sample during a recognizing mode. As a matter of fact, the nonlinearity improves the recognition results but makes the CNN more complicated. Thus, adding nonlinearity into the CNN linear layers ought to be reasonable and carefully done (Zhang and Wu, 2019).

5. EXPERIMENTAL RESULTS

Our proposition is verified by experiments with pedestrian detection and tracking in outdoor environment. In order to gauge performances, we used the public Caltech Pedestrian Dataset (Caltech Pedestrian Detection Benchmark, 2019). This dataset includes approximately 10 hours of 640×480 30Hz videos taken from a vehicle driving in an urban environment. About 250,000 annotated frames with pedestrians were divided into 137 approximately minute long segments. In total, 350,000 bounding boxes and 2,300 unique pedestrians were annotated. Each annotation includes temporal correspondence between bounding boxes and detailed occlusion labels.

The most popular architectures of CNN applied for pedestrian detection are the following: AlexNet (Tian et al., 2015), Multi-scale CNN (MS-CNN) (Cai et al., 2016), and Scale-Aware Fast Region CNN (SAF R-CNN) (Li et al., 2015). The architectures of these three CNN are represented in Tables 1-3. Architecture includes convolutional layers (Conv), polling layers (Pool and Max pool), fully-connected layers (FullyCon), Soft-max layer, and Region Of Interest (ROI) pooling (ROI Pool). The convolutional and fully-connected layers involve ReLU activation function.

The MS-CNN architecture contains the main branch and three additional branches with layer types Det-8, Det-16, Det-32, and Det-64 marked by Italic in Table 2.The layer Conv4-3 is connected with the layer Det-conv. The layer Conv5-3 is connected with the layer Det-16. The layer Conv6 is connected with the layer Det-32. The output maps from the layers Det-8, Det-16, Det-32, and Det-64 are combined in the layer ROI Pool.

The SAF R-CNN architecture represented in Table 3 uses the first seven convolutional layers and three max pooling layers of the VGG16 network and then is divided into Large-size sub-network and Small-size sub-network with following shared features.

Note that the AlexNet provides a class probability, while the MS-CNN and SAF R-CNN give Class probability and Bounding box.

Layer type	Kernel/Stride/ReLU/ Features	Output map size
Conv1	11×11/4/ReLU/96	55×55
Conv2	5×5/2/ReLU/256	27×27
Pool1	5×5/1	27×27
Conv3	3×3/2/ReLU/384	13×13
Pool2	3×3/1	13×13
Conv4	3×3/1/ReLU/384	13×13
Conv5	3×3/1/ReLU/256	13×13
Pool3	3×3/2	6×6
FullyCon1	6×6/1/ReLU/4096	1×1
FullyCon2	1×1/1/ReLU/4096	1×1
FullyCon3	1×1/1/ReLU/1000	1×1
Soft-max		
Output: Class probability		

Table 1. Architecture of AlexNet

Layer type	Kernel/Stride/ ReLU/Features	Layer type	Kernel/Stride/ ReLU/Features	Layer type	Kernel/Stride/ ReLU/Features	Layer type	Kernel/Stride/ ReLU/Features
			· · · · ·				
		Conv4-3	7×7/2/ReLU				
		Max pool1	7×7/1			Det-conv	3×3/1/ReLU/512
		Conv5-1	5×5/1/ReLU			*Det-8*	*5×3/1/ReLU/512*
		Conv5-2	5×5/1/ReLU				
		Conv5-3	5×5/1/ReLU				
Det-16	*5×3/1/ReLU/512*	Max pool2	5×5/1				
		Conv6	3×3/1/ReLU				
		Max pool3	3×3/1	*Det-32*	*5×3/1/ReLU/512*		
		Det-64	*5×3/1/ReLU/512*				
		ROI Pool	7×7/1				
		Conv	5×5/1/ReLU/512				
		FullyCon	1×1/1/ReLU/512				
Output: Class probability and Bounding box							

Table 2. Architecture of MS-CNN

Layer type	Kernel/Stride/ ReLU/Features	Layer type	Kernel/Stride/ ReLU/Features	Layer type	Kernel/Stride/ ReLU/Features
		Conv1	3×3/1/ReLU/64		
		Conv2	3×3/1/ReLU/64		
		Max pool1	3×3/1		
		Conv3	3×3/1/ReLU/128		
		Conv4	3×3/1/ReLU/128		
		Max pool2	3×3/1		
		Conv5	3×3/1/ReLU/256		
		Conv6	3×3/1/ReLU/256		
		Conv7	3×3/1/ReLU/256		
		Max pool3	3×3/1		
Conv	3×3/1/ReLU/512			Conv	3×3/1/ReLU/512
Conv	3×3/1/ReLU/512			Conv	3×3/1/ReLU/512
		ROI Pool	3×3/1		
		ROI Pool	3×3/1		
		FullyCon	1×1/1/ReLU/4096		
		FullyCon	1×1/1/ReLU/4096		
Output: Class probability and Bounding box					

Table 3. Architecture of SAF R-CNN

CNN architecture	Distance	Bounded LReLU			Proposed bounded RL-ReLU		
		mAP (%)	MR (%)	FP (%)	mAP (%)	MR (%)	FP (%)
AlevNet	far	44.36	18.60	37.77	46.81	12.92	40.91
MS-CNN	far	51.01	17.36	34.38	52.69	14.74	30.86
SAF R-CNN	far	48.53	13.95	37.73	46.78	14.06	33.99
AlevNet	middle	89.03	3.54	7.99	90.82	3.27	6.30
MS-CNN	middle	91.49	3.15	6.70	91.12	2.61	6.12
SAF R-CNN	middle	90.74	3.59	6.32	90.24	3.40	6.01
AlevNet	close	95.59	1.26	3.67	95.20	1.16	3.25
MS-CNN	close	97.02	0.97	1.64	97.38	0.76	1.75
SAF R-CNN	close	100.00	0.00	0.00	100.00	0.00	0.00

Table 4. Efficiency of CNNs with different types of ReLU activation functions

Since the training of CNN is very time consuming, two different approaches based on Transfer Learning and Feature Extractors are utilized in many applications. Feature extraction is more justified and perspective regarding promising results. Therefore, this approach was chosen during experiments. The conventional features for pedestrian detection and tracking, such as Haar features and AdaBoost classifiers, were tested using adaptive RReLU in the non-linear activation layers. Experiments were conducted in such manner that, first, the learning and testing processes were implemented with bounded LReLU activation function, which is the most close to the proposed bounded RL-ReLU, and, second, learning and testing processes were executed with the proposed bounded RL-ReLU.

The obtained results in the terms of mean Average Precision (mAP), Miss Rate (MR), and False Positives (FP) are grouped in Table 4. Values of average precision of pedestrian recognition for three distances (far, middle, and close) are very close for AlexNet, MS-CNN, and SAF R-CNN. The best coincidences achieved for MS-CNN and SAF R-CNN due to multiple use of ReLU activation function. As one can see from Table 4, the proposed bounded RL-ReLU has lesser values of errors for all CNN applications with far, middle, and close distances between camera and pedestrians. Some visual results are depicted in Figure 3. The green rectangle means the true positive example, the blue rectangle means the missing positive example, and the red rectangle means the false positive example.

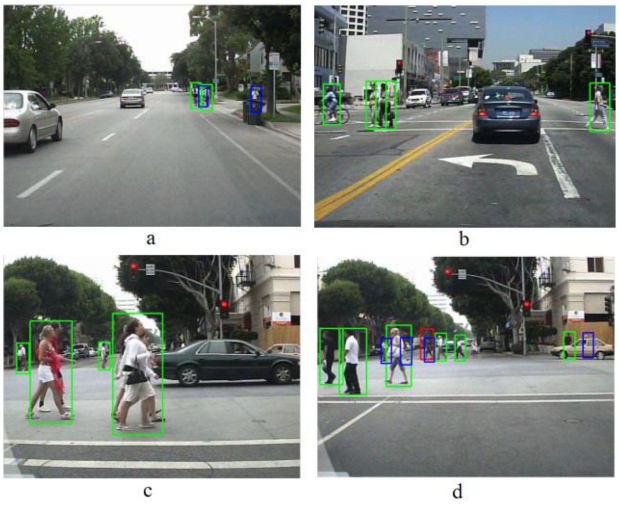

Figure 3. The obtained detection results using Caltech Pedestrian Detection dataset: a far distance, b middle distance, c close distance, d middle and far distances

Also, the bounded RL-ReLU was tested on noisy videos obtained by manual shooting. Our videos represent an urban environment with pedestrians. Pedestrian detection was executed using AlexNet, MS-CNN, and SAF R-CNN. In this case, the mAP, MR, and FP values using the proposed bounded RL-ReLU degraded on 10-15% respect to values from Table 4. At the same time, the mAP, MR, and FP values using the bounded L-ReLU showed worse results relative to the proposed bounded RL-ReLU on 24-30%. Experiments show that the overfitting became less sense problem.

6. CONCLUTIONS

In this paper, we have proposed a special ReLU activation function called as bounded RL-ReLU. The proposition was tested on the task of pedestrian detection in outdoor environment using the public Caltech Pedestrian Dataset. To this end, three architectures of CNN, such as AlexNet, MS-CNN, and SAF R-CNN, were employed. The proposed bounded RL-ReLU demonstrates lesser values of errors for all CNN applications with far, middle, and close distances between camera and pedestrians. However, the tested CNN architectures provide worse results with occlusions of visual objects due to their failure to consider the overlapping ROIs.

REFERENCES

Cai, Z., Fan, Q., Feris, R., Vasconcelos, N., 2016. A unified multi-scale deep convolutional neural network for fast object detection. In: *European Conference on Computer Vision*, Amsterdam, The Netherlands, Part IV, LNCS, Vol. 9908, pp. 354-370.

Caltech Pedestrian Detection Benchmark http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/ (9 March 2019).

Cao, J., Pang, Y., Li, X., 2016. Learning multilayer channel features for pedestrian detection. *IEEE Transactions on Image Processing*, 26(7), pp. 3210-3220.

Cao, J., Pang, Y,. Li, X., Liang, J., 2018. Randomly translational activation inspired by the input distributions of ReLU. *Neurocomputing*, 275, pp. 859-868.

Clevert, D.-A., Unterthiner, T., Hochreiter, S., 2016. Fast and accurate deep network learning by exponential linear units (ELUs). In: *The International Conference on Learning Representations*, San Juan, Puerto Rico, https://arxiv.org/abs/1511.07289 (8 January 2019).

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. In: *The 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 886-893.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *The 27th IEEE Conference on Computer Visual and Pattern Recognition*, pp. 580-587.

Glorot, X., Bordes, A., Bengio, Y., 2011. Deep sparse rectifier neural networks. In: *The 14th International Conference on Artificial Intelligence and Statistics*, pp. 315-323.

Gulcehre, C., Moczulski, M., Denil, M., Bengio, Y., 2016. Noisy activation functions. In: *The 33rd International Conference on International Conference on Machine Learning*, Vol. 48, pp. 3059-3068.

Jiang, X., Pang, Y., Li, X., Pan, J., 2016. Speed up deep neural network based pedestrian detection by sharing features across multi-scale models. *Neurocomputing*, 185, pp. 163-170.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers surpassing human-level performance on ImageNet classification. In: *The 2015 IEEE International Conference on Computer Vision*, pp. 1026-1034.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. *Nature*, 521, pp. 436-444.

Li J., Liang X., Shen S., Xu T., Yan S., 2015. Scale-aware fast R-CNN for pedestrian detection. arXiv:1510.08160.

Li, H., Wu, H., Lin, S., Luo, X., 2018. Coupling deep correlation filter and online discriminative learning for visual object tracking. *Journal of Computational and Applied Mathematics*, 329, pp. 191-201.

Li, P., Wang, D., Wang, L., Huchuan Lu, H., 2018. Deep visual tracking: Review and experimental comparison. *Pattern Recognition*, 76, pp. 323-338.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. SSD: Single shot multibox detector. In: *European Conference on Computer Vision*, pp. 21-37.

Liew, S.S., Khalil-Hani, M., Bakhteri, R., 2016. Bounded activation functions for enhanced training stability of deep neural networks on visual pattern recognition problems. *Neurocomputing*, 216, pp. 718-734.

Maas, A.L., Hannun, A.Y., Ng, A.Y., 2013. Rectifier nonlinearities improve neural network acoustic models. In: *The 30th International Conference on Machine Learning, Workshop on Deep Learning for Audio, Speech, and Language Processing*, Vol. 30, no. 1, pp. 1-6.

Munder, S., Schnorr, C., Gavrila, D.M., 2008. Pedestrian detection and tracking using a mixture of view-based shape–texture models. *IEEE Transactions on Intelligent Transportation Systems*, 9(2), pp. 333-343.

Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted Boltzmann machines. In: *The 27th International Conference on Machine Learning*, pp. 807-814.

Nam, W., Dollár, P., Han, J.H., 2014. Local decorrelation for improved pedestrian detection. *The 28th Annual Conference on Neural Information Processing Systems*, Vol. 1, pp. 424-432.

Nikouei, S.Y., Chen, Y., Song, S., Xu, R., Choi, B.-Y., Faughnan, T.R., 2018. Real-time human detection as an edge service enabled by a lightweight CNN. In: *The 2018 IEEE International Conference on Edge Computing* https://arxiv.org/abs/1805.00330 (8 January 2019).

Pang, S., del Coz, J.J., Yu, Z., Luaces, O., Díez, J., 2017. Deep learning to frame objects for visual target tracking. *Engineering Applications of Artificial Intelligence*, 65, pp. 406-420.

Raza, M., Chen, Z., Rehman, S.-U., Wang, P., Bao, P., 2018. Appearance based pedestrians' head pose and body orientation estimation using deep learning. *Neurocomputing*, 272, pp. 647-659.

Qian, X., Han, L., Wang, Y., Ding, M., 2018. Deep learning assisted robust visual tracking with adaptive particle filtering. *Signal Processing: Image Communication*, 60, pp. 183-192.

Tian, Y., Luo, P., Wang, X., Tang, X., 2015. Deep learning strong parts for pedestrian detection. In: *The 2015 IEEE International Conference on Computer Vision*, pp. 1904-1912.

Tomè, D., Monti, F., Baroffio, L., Bondi, L., Tagliasacchi, M., Tubaro, S., 2016. Deep convolutional neural networks for pedestrian detection. *Signal Processing: Image Communication*, 47, pp. 482-489.

Wang, X., Han, T.X., Yan, S., 2009. An HOG-LBP human detector with partial occlusion handling. *In: The IEEE 12th International Conference on Computer Vision*, pp. 32-39.

Viola, P., Jones, M.J., Snow, D., 2003. Detecting pedestrians using patterns of motion and appearance. *In: The 9th IEEE International Conference on Computer Vision*, Vol. 2, pp. 734-741.

Xue, H., Liu, Y., Cai, D., He, X., 2016. Tracking people in RGBD videos using deep learning and motion clues. *Neurocomputing*, 204, pp. 70-76.

Xu, B., Wang, N., Chen, T., Li, M., 2016. Empirical evaluation of rectified activations in convolution network. In: *The 33rd International Conference on Machine Learning, Workshop on Deep Learning* https://arxiv.org/abs/1505.00853 (8 January 2019).

Xu, B., Wang, N., Chen, T., Li, M., 2015. Empirical evaluation of rectified activations in convolutional network. *The International Conference on Machine Learning*, https://arxiv.org/abs/1505.00853 (8 January 2019).

Zhang, C.-L., Wu, J., 2019. Improving CNN linear layers with power mean non-linearity. *Pattern Recognition*, 89, pp. 12–21.

Zhang, S., Bauckhage, C., Cremers, A.B., 2014. Informed Haar-like features improve pedestrian detection. In: *The 27th IEEE Conference on Computer Vision and Pattern Recognition*, pp. 947-954.