JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

RESEARCH ARTICLE/REVIEW

The organized Survey on Object Detection System Using CNN:A Theoretical Approach.

Chandrashekhar S, Basavaprasad*
1 Department of Computer Science, Govt. First Grade College Raichur, India
2. Department of Computer Science, Govt. Degree College Yadgiri, India,

Abstract: One of the important developments that sparked the deep neural network renaissance in computer vision, a subset of machine learning, was the development of CNNs. Typically, convolutional, pooling, and dense layers are combined to create a CNN. The moment we perceive an image, the human brain begins processing a massive amount of data. Every neuron has a distinct receptive field and is coupled to other neurons so that they collectively cover the whole visual field. Each neuron in a CNN processes data only in its receptive field, similar to how each neuron in the biological vision system responds to stimuli only in the constrained area of the visual field known as the receptive field. Simpler patterns like lines and curves are detected initially by the layers, followed by more intricate patterns like faces and objects. One can enable sight to computers by employing a CNN. The foundational component of the CNN is the convolution layer. It carries the majority of the computational load on the network. In order to increase the precision and energy efficiency of the detection process, this research examines algorithms created for real-time object detection applications by fusing Convolutional Neural Networks (CNN) with Scale Invariant Feature Transform. The scientific community has been interested in object detection for many years and has made great progress in this field. There is a vast array of applications that could benefit from more advancement in the field of object detection. The efforts in this area have been complimented by the field of machine learning's rapid development, and in recent years, the research community has made significant contributions to real-time object detection. Real-time object detection has been used in the current work, and the authors have worked to increase the detection mechanism's precision.

Keywords: Kernel, Pattern, Machine learning, Object Detection, Deep Learning, CNN

Foreword

CNNs (Convolution Neural Networks) are a subset of deep learning algorithms that can take in a sample image as input and perform convolution operations to extract features from the image and be able to distinguish between individual objects.

1. Introduction

Artificial intelligence and machine learning have advanced at an exponential rate in recent years, which has improved accuracy while lowering human effort and failure rates. This advancement has made a notable contribution to processing time reduction, which has further improved net productivity and resulted in a reduction in cost.

Various Approaches to Object Detection Problem (Hernandez-Penaloza et al., 2017)

- Naïve way

The image can be divided into four sections: the upper left-hand corner, the upper right-hand corner, the bottom left-hand corner, and the lower right-hand corner. Let's say we're trying to locate the pedestrian in the picture. Feed a classifier with each of the four components. Whether a pedestrian is present in the image will be shown by the output. Mark that patch in the image if it is discovered.

- Increase number of divisions

A more effective strategy than a naive one is to increase the number of patches or divisions. The sole drawback is that numerous bounding boxes are needed for essentially the same thing.

- Perform structural divison

It gets around the drawback of the second strategy. grid of 10x10 pixels around the image.Define each patch's centroid. Take various patches and run them through the classifier for each centroid.

. • To make it even more efficient

Instead of using three patches, use additional patches on a larger grid. However, expanding patches will make it more tough for the classifier, therefore we should only use some of them.

- Using deep learning

Deep learning is an extension of machine learning that focuses on algorithms that are motivated by how the brain functions (Zhang et al., 2020; Esteva et al., 2017; Bali et al., 2020). Since Deep Learning performs the best of all algorithms in terms of object detection, we use it. In deep learning, the entire image is sent to the neural network for dimension reduction instead of only sampling portions of it. To recommend specific patches and provide predictions as close as feasible to the original bounding box, neural networks may be useful..

- Challenges for Object Detection using Machine Learning Algorithms

Although classifying items with the human eye is a simple process, doing so with a computer is technically difficult because the human eye is capable of classifying objects in two-dimensional imagery as well. Object detection algorithms face a number of difficulties in real time.

- Localization

Finding the location of a single object within a picture is challenging for algorithms (Zhao et al., 2019; Ouadiay et al., 2018).

- Instance Segmentation

The algorithm then needs to segment, partition, or separate the object from other objects in the image after it has been located (Hu et al., 2018).

- Classification of object in different categories

Images used as input should have good quality and resolution. Other important factors are lighting, angle, object size, and orientation (Ard et al., 2019; Klette, 2014).

- Occlusion

It happens in the situation of two or more things coming too close to each other and either merging or combining altogether (Chandel & Vatta, 2015).

- Mirroring

Mirror images of any object must be recognized by object identification systems (Owen & Chang, 2019).

It frequently draws on the disciplines of image processing, deep learning, and computer vision. Object detection is a multidisciplinary study area. For object recognition in a real-time context, we have incorporated computer vision and machine learning ideas in our recent experimental work. Sliding windows, support vector machines, principal component analysis (Mishra et al., 2017), and SSD (Single Shot Multibox Detector) (Redmon et al., 2016 and Phadnis et al., 2018) are some of the concepts used in our work. TensorFlow and OpenCV are employed for the purposes of the experimental study. There is a wide range of potential uses for real-time object detection that are now becoming essential to human activities. Facial recognition, industrial quality control, autonomous vehicles, optical character recognition, robotics, and real-time disease diagnosis are only a few of the applications (Emami & Suciu, 2012; Alie et al., 2017; Memon et al., 2016; Hamad & Kaya, 2016; Aggarwal et al., 2019).

Objective:

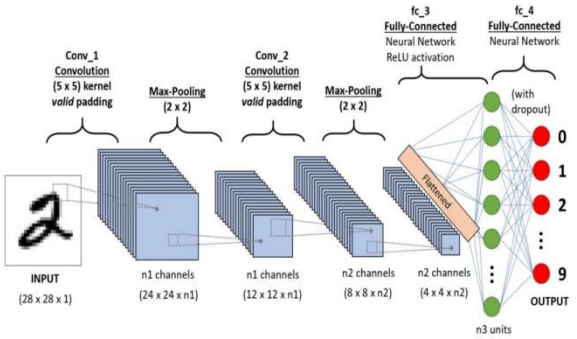
This study aims to investigate and evaluate a cluster of numerous deep convolutional neural networks and a hybrid CNN-SIFT object tracking system. The ultimate objective of the current object detection research is to accept raw photos as inputs, precisely locate the object in the given image, and then mask or classify the object with the relevant categories.

Convolutional Neural Networks, a unique kind of neural network that substantially mimics human vision, were at the core of AlexNet. Since CNNs are now a crucial component of many computer vision applications, they are included in every online course on computer vision. Consequently, let's examine how CNNs operate.

Around the 1980s, CNNs were first created and put to use. At the time, a CNN could only recognize handwritten numbers to a certain extent. To read zip codes, pin numbers, etc., it was mostly utilized in the postal industry. The most crucial thing to keep in mind about any deep learning model is that it needs a lot of computational power and data to train. Because of this significant disadvantage at the time, CNNs were restricted to the postal industry and were unable to enter the machine learning field.

”

Figure 1. CNN Model for Object Detection



2. Literature Review

A CNN network's structural layout (Fig. 1) resembles the connection pattern of the neurons in the human brain. The Convolution Layer is the first layer of a CNN design, and the kernel/filter is the object in charge of convolution (Zhang et al., 2017). The mapping of an image's significant features is done by this layer. Rectified Linear, also known as Relu, is used as a piecewise activation function after the convolution function to perform a non-linear transformation on the input. After that, pooling is carried out (Scherer et al., 2010) to condense the size of the feature map. Pooling plays a crucial role in reducing the amount of computational resources required to analyse the data by reducing the number of dimensions and further limiting the characteristics that are extracted to those that are dominant. There are two main types of pooling. If the convolved feature is 4*4 and we need 2*2 blocks, we use max pooling (Christlein et al., 2020). To do this, we divide the convolved feature map into 2*2 blocks and then locate the biggest element in each block. To determine the average of each block, we can utilize average pooling. Classifying the objects is the final stage. CNN might only be able to anticipate the type of object; it cannot pinpoint where the thing is. To get around this, it becomes necessary to choose a lot of different regions (Tran et al., 2020; Sedghi et al., 2019). Other filtering techniques exist that can be useful for feature recognition and censoring (Shang, 2020), however in this work, the authors have developed a hybrid technique.

R-CNN: Researcher Ross Girshick presented a method including selective search methods for extracting "2000" regions of the image that are addressed as the region suggestions in order to overcome the constraint caused by the need to choose a large number of regions (Ren, He, Girshick, & Sun, 2017). "2000" regions are taken out of the image and wrapped in a square before being given to CNN, which acts as an extractor of features. Support Vector Machine (SVM) performs object classification in order to assess the region suggestion. It consumes a substantial amount of time for network training. Selective search meets the requirements to be a fixed algorithm. There may be a chance that bad region proposals will be made. (Lakhal et al. 2018)

Fast R-CNN: The training process was altered to make the R-CNN faster (Ren, He, Girshick, & Sun, 2017). Instead of feeding the region proposals, this model uses CNN to create a convolutional feature map from the input image and then identify the region of proposals before wrapping it into a square. Through ROI pooling, fixed-size regions of proposals are shaped (Qin et al., 2016). The maximum pooling technique is referred to as ROI pooling and is used to translate the image features into the matching region of the picture of dimension h*w into a small static window of size H*W. The area of Input is divided into H*W grids, followed by the creation of subwindows, and finally, maxpooling is applied to each grid.

Faster R-CNN: Faster R-CNN, which uses a detection pipeline idea, uses R-CNN and Fast CNN to accelerate the detection process (Ren, He, Girshick, & Sun, 2017).

RPN (Region Proposal Network) in Faster R-CNN (Zhou et al., 2018): The object proposals are invoked using this method. A classifier comes first in an RPN, then a regressor. While a regressor's job is to regress the coordinates that correspond to the proposals, a classifier's job is to estimate the likelihood that a candidate proposal would contain the object. The anchor is the center point of the sliding window via which the involved featured map travels.

YOLO-You Only Look Once: An image is given as the input to YOLO (Redmon et al., 2016), which is then divided up into a grid and some bounding boxes are formed inside the grid. The network generates a probability of the class and the bounding box offset quantum values for each bounding box. Higher levels of probability are represented by a bounding box. Compared to the speedier R-CNN, the approach analyzes data substantially faster.

Mask R-CNN: Faster R-CNN and fixed level picture segmentation make up Mask R-CNN (He et al., 2020). Over the ROI pooling layer, Mask R-CNN improvised and called it the RoI Align layer. It corrects the positioning issue brought on by the ROI pooling layer.

SSD(Single Shot Multi-Box Detector): Because of its accuracy and increased speed, this method is widely used for object recognition in real-time (Liu et al., 2016) (Fig. 2). To assess the model's accuracy, the mean average of precision (mAP) may be important. One drawback of YOLO was that it could miss very minute objects in an image. While SSD may also be used to find little items. The SSD object detection system has two parts. The first one, called VGG 16, is

in charge of extracting feature maps, while the second one, called Conv4_3 layer (convolution), is in charge of identifying objects.

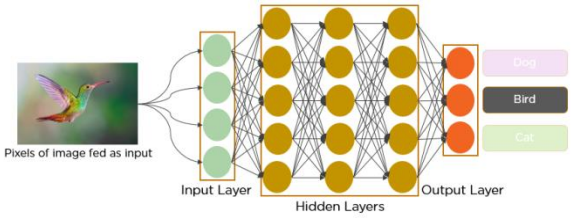
Since each layer of SSD can identify and categorize items, accuracy rises. Rightmost layers are capable of detecting huge things, whereas the leftmost layers, or feature maps, with higher levels of resolution, are responsible for differentiating little items. 300*300*3 is the size of each input image segment. The image is then made convoluted by a number of different convolution layers.

Example: The output of the convolution layer will be 5*5 if the feature map is 10*10 and the convolution layer is 1*1 with a two-stride. The term "stride" alludes to the quantity of pixel shifts. Conv4_3 is assumed to be 38*38. It provides '4' predictions of the items, one for each cell. The boundary box is a component of each forecast. There are 21 scores total, with the top score for each class being chosen. Creating several forecasts with boundary boxes and confidence results in a multi-box procedure. After VGG16, SSD adds six convolutional layers. These layers use stride2 and a 1*1 convolution. Six predictions are made in those layers. SSD generates 8732 predictions in total. Only the top N predictions are left when low confidence bounding boxes are eliminated. By doing this, noisy forecasts are removed.

Deep Learning has established itself as a very potent tool over the last few decades due to its capacity for handling massive amounts of data. Hidden layer technology is much more popular than conventional methods, particularly for pattern recognition. Convolutional Neural Networks are among the most widely used deep neural networks. Researchers have struggled to create a system that can comprehend visual input ever since the 1950s, the early years of AI. This area of study eventually became known as computer vision. When a team of researchers from the University of Toronto created an AI model that significantly outperformed the best image recognition algorithms in 2012, computer vision experienced a quantum leap.

The 2012 ImageNet computer vision competition was won by the AI system, known as AlexNet (after its principal designer, Alex Krizhevsky), with an astounding 85 percent accuracy. The test result for the runner-up was a modest 74 percent.

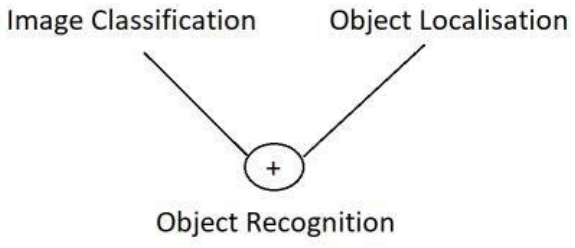
Figure 2
Example of Process



2.1 Theoretical Framework

The Inception SSD model is employed in the current experimental effort. Prior to inception, the majority of CNNs just stacked convolution layers deeper to improve performance, which was computationally expensive and subject to overfitting. Inception SSD (Szegedy et al., 2016) is a complicated network and is used to gain higher performance, both in terms of speed and performance. The Inception model comes in a variety of variations. There are 27 levels in The Inception1 (Szegedy et al., 2015), a CNN. There are several Inception layers in the layered architecture of Inception1. The first layer is made up of a mushroomed pool of several layers (some examples include 1*1, 3*3, or 5*5 convolutional layers, etc.), each of which has an output filter bank coupled to a single output vector that serves as the feed for the subsequent stage. Analyze the issue that inceptionv1 attempted to fix. The excessive dimensional reduction issue that Inception1 identified was addressed in Inceptionv2, which also discovered that factoring lessens complexity. Due of the sluggish and expensive nature of 5*5 convolutions, this inception approach factors it into two 3*3 convolutions. As a result, Inceptionv2's performance was altered. By combining convolutions of dimensions 1*n and n*1 with segmentation of convolutions of the filter dimension n*n. This was even better than before. Here, wider banks are used as filter banks rather than deeper banks.

Figure 2
Theory of Object etection



3. Research Methodology

3.1. Research Design

In contrast to the image classification challenge, which focuses on categorizing the photos, in object recognition our objective is to locate all of the objects and arrange them in the boxes that are most appropriate for each one.

Bounding Box — ROI (Region of interest): We need to introduce a new phrase for object recognition. Bounding boxes are used in our attempts to recognize items so that they may be detected. Later, we shall discover how to get boxes that are as near to the observed object as is humanly possible.

To resume again these 3 different tasks:

Image Classification: Determine the class of an object in a picture using image classification.

Object Localization: Locate the presence of objects in an image and use a bounding box to pinpoint their locations.

Object detection: Use bounding boxes to find the presence of objects and identify the classes of those items.

There are two primary categories of object recognition neural network architectures that have been developed so far: Detectors: Single-Stage vs. Multi-Stage.

Multi-Stage Detectors

1. RCNN 2014
2. Fast RCNN 2015
3. Faster RCNN 2015

Single-Stage Detectors

1. SSD 2016
2. YOLO 2016
3. YOLOv2 2016, YOLOv3 2018, YOLOv4 2020, YOLOv5 2020.

3.2. Proposed Concepts

Process flow of the proposed model

With the use of the internet, raw photographs were gathered as the first step in the experimental procedure to create the dataset (Bashiri et al., 2018). We have gathered a sizable number of related photographs and categorized them. Figure 3 depicts the consecutive steps that were taken to model the current system. Images were captured from a variety of viewpoints, and care was given with regard to brightness, scale, lightning conditions, and angles. The format of every image used is.jpg, and a total of 150 photographs from each category were collected; these images ultimately provided sufficient performance in detection. The directory created for storage contained all of the gathered photographs in a suitable manner. Images from the entire collection were divided into two groups: a training group and a testing group. In this work, a ratio of 90:10—which may possibly be increased to 80:20—was assumed.

Using a labeling image annotator named LabelImg, images were appropriately labeled by hand throughout the next step. According to

Fiedler et al. (2019), this annotator offers a user-friendly GUI and saves label files in the PascalVOC format, which can be beneficial in the future. After the picture dataset has been annotated, it is customary to retain the remaining data for evaluation purpose and only use a subset of it for training. The xml files for each captioned image are then created. The intrinsic nature of XML, The easiest way to express an ordered data format like XML is through a tree. According to Tu et al. (2004), ElementTree refers to the entire XML document as a tree, with each element represented by a single tree node. Every interaction with the document is done at the ET level, and all communication within XML components is done at the Element level. Additionally, all files are converted from XML to CS because most users cannot read data in XML format (Mitlohner et al., 2016). Values are separated in CSV format using delimeters or commas. Using the ElementTree library that is a built-in part of Python, XML files can be provided. Figure 4 displays a screenshot of the csv file that was made for the images. We further transformed the CSV files to TensorFlow Records (also known as TF records) in the next phase (Smith et al., 2016). This method converts any available data into a format that is supported.

A new way of viewing the world has been made possible by various applications that use object detection in real time utilizing contemporary machine learning algorithms. An SSD Model interacts with our data with all the knowledge and resources to enable our experimental work. Finally, when we run our model, a different window that can simultaneously identify several objects is opened. Figure 5 shows how it simultaneously recognizes a mobile phone and a remote using an active webcam. We conducted tests using a dataset created for approximately 100 things, including phones, bags, remote controls, books, chairs, and tables, and approximately 100–150 photographs for each object, for a total database of approximately 10,000 images, of which approximately 9,001 images trained the model and the remaining 1,000 images tested it. An average success rate of 92.4% was attained. One of the better examples is a cell phone with an accuracy rate of above 97%. The application updates the video window with a new frame at regular intervals of 0.25 and 0.5 seconds, indicating an average of 2–4 FPS.The model's speed and general accuracy are superior than those of other models because it was created using SSD. The quantity of crucial features provided into the algorithm and the size of the database affect how quickly an object can be identified. Currently, the total recognition time on a single processor for the 6-object database is around 20 seconds, and for the 24-object database, it is 2 minutes.

3.3. Instruments

As the first member of this family, we will see the base methodology and the improvement of its disadvantages in further versions. The methodology of this model is as follows:

1. Region Proposal Extraction from Input Image using Selective Search

The segmentation algorithm is used by the selective search algorithm to detect blobs in an image that might represent objects. Selective search creates 2,000 areas to be examined by iteratively combining these groupings of regions into larger ones.

Being Selective Search in the same category, it applies the following steps to segment the image:

• The first step is to calculate the similarities between all nearby regions.

• New similarities are determined between the resulting region and its neighbors after grouping the two most similar regions together..

• Once the entire object is coated in one area, this process is repeated.

2. Feature Extraction using CNN on each ROI comes from the previous step

After extracting almost 2000 possible boxes which may have an object according to the segmentation, CNN is applied to all these boxes one by one to extract the features to be used for classification at the next step

3. SVM-based classification using bounding boxes

Finally, the model gives us the final bounding boxes along with the detected classes utilizing SVM (support vector machine) for classification and a bounding box regressor, where the bounding box regressor's role is simply to enhance the proposed box to encircle the item.

4. Conclusion

CNNs deliver in-depth results despite their immense power and complicated resource requirements. Simply identifying patterns and nuances that are so minute and subtle that the human eye misses them is what it all boils down to. Deep learning techniques have been continually improved since CNNs were invented in order to create a more reliable pedestrian detector. Different Deep Neural Network architectures have been developed to address various problems encountered in real-world settings in order to enhance robotic vision.

There are numerous object detection models available at the moment for detecting the items. SSD has been applied in the current experimental attempt. Choosing the best model is a very common and difficult issue because certain models may have better accuracy but not the best processing speed, and vice versa.

Acknowledgement

This work is carried out with thorough survey on object detection and counting. We would like to thank all the authors which we have mentioned in the reference section without them this work would be impossible.

References

[1] A Bengio, Y.; Courville, A.; Vincent, P. (2013). "Representation Learning: A Review and New Perspectives". IEEE Transactions on Pattern Analysis and Machine Intelligence. 35 (8): 1798–1828. arXiv:1206.5538. doi:10.1109/tpami.2013.50. PMID 23787338. S2CID 393948.

[2] Dasiopoulou, Stamatia, et al. "Knowledge-assisted semantic video object detection." IEEE Transactions on Circuits and Systems for Video Technology 15.10 (2005): 1210–1224.

[3] Ling Guan; Yifeng He; Sun-Yuan Kung (1 March 2012). Multimedia Image and Video Processing. CRC Press. pp. 331–. ISBN 978-1-4398-3087-1.

[4] Alsanabani, Ala; Ahmed, Mohammed; AL Smadi, Ahmad (2020). "Vehicle Counting Using Detecting-Tracking Combinations: A Comparative Analysis". 2020 the 4th International Conference on Video and Image Processing. pp. 48–54. doi:10.1145/3447450.3447458. ISBN 9781450389075. S2CID 233194604.

[5] Wu, Jianxin, et al. "A scalable approach to activity recognition based on object use." 2007 IEEE 11th international conference on computer vision. IEEE, 2007.

[6] Bochkovskiy, Alexey (2020). "Yolov4: Optimal Speed and Accuracy of Object Detection". arXiv:2004.10934 [cs.CV].

[7] Dalal, Navneet (2005). "Histograms of oriented gradients for human detection" (PDF). Computer Vision and Pattern Recognition. 1.

[8] Ross, Girshick (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation" (PDF). Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE. pp. 580–587. arXiv:1311.2524. doi:10.1109/CVPR.2014.81. ISBN 978-1-4799-5118-5. S2CID 215827080.

[9] Girschick, Ross (2015). "Fast R-CNN" (PDF). Proceedings of the IEEE International Conference on Computer Vision. pp. 1440–1448. arXiv:1504.08083. Bibcode:2015arXiv150408083G.

[10] Shaoqing, Ren (2015). "Faster R-CNN". Advances in Neural Information Processing Systems. arXiv:1506.01497.

[11] Pang, Jiangmiao; Chen, Kai; Shi, Jianping; Feng, Huajun; Ouyang, Wanli; Lin, Dahua (2019-04-04). "Libra R-CNN: Towards Balanced Learning for Object Detection". arXiv:1904.02701v1 [cs.CV].

[12] LeCun, Yann; Bengio, Yoshua; Hinton, Geoffrey (28 May 2015). "Deep learning". Nature. 521 (7553): 436–444. Bibcode:2015Natur.521..436L. doi:10.1038/nature14539. PMID 26017442. S2CID 3074096.

[13] Schmidhuber, J. (2015). "Deep Learning in Neural Networks: An Overview". Neural Networks. 61: 85–117. arXiv:1404.7828. doi:10.1016/j.neunet.2014.09.003. PMID 25462637. S2CID 11715509.
[14] Shigeki, Sugiyama (12 April 2019). Human Behavior and Another Kind in Consciousness: Emerging Research and Opportunities: Emerging Research and Opportunities. IGI Global. ISBN 978-1-5225-8218-2.
[15] Bengio, Yoshua; Lamblin, Pascal; Popovici, Dan; Larochelle, Hugo (2007). Greedy layer-wise training of deep networks (PDF). Advances in neural information processing systems. pp. 153–160. Archived (PDF) from the original on 2019-10-20. Retrieved 2019-10-06.
[16] Hinton, G.E. (2009). "Deep belief networks". Scholarpedia. 4 (5): 5947. Bibcode:2009SchpJ...4.5947H. doi:10.4249/scholarpedia.5947.
[17] Cybenko (1989). "Approximations by superpositions of sigmoidal functions" (PDF). Mathematics of Control, Signals, and Systems. 2 (4): 303–314. doi:10.1007/bf02551274. S2CID 3958369. Archived from the original (PDF) on 10 October 2015.
[18] Hornik, Kurt (1991). "Approximation Capabilities of Multilayer Feedforward Networks". Neural Networks. 4 (2): 251–257. doi:10.1016/0893-6080(91)90009-t.
[19] Haykin, Simon S. (1999). Neural Networks: A Comprehensive Foundation. Prentice Hall. ISBN 978-0-13-273350-2.
[20] Hassoun, Mohamad H. (1995). Fundamentals of Artificial Neural Networks. MIT Press. p. 48. ISBN 978-0-262-08239-6.
[21] Lu, Z., Pu, H., Wang, F., Hu, Z., & Wang, L. (2017). The Expressive Power of Neural Networks: A View from the Width Archived 2019-02-13 at the Wayback Machine. Neural Information Processing Systems, 6231-6239.
[22] Orhan, A. E.; Ma, W. J. (2017). "Efficient probabilistic inference in generic neural networks trained with non-probabilistic feedback". Nature Communications. 8: 138. doi:10.1038/s41467-017-00181-8. PMID 28743932.
[23] Murphy, Kevin P. (24 August 2012). Machine Learning: A Probabilistic Perspective. MIT Press. ISBN 978-0-262-01802-9.
Sonoda, Sho; Murata, Noboru (2017). "Neural network with unbounded activation functions is universal approximator". Applied and Computational Harmonic Analysis. 43 (2): 233–268. arXiv:1505.03654. doi:10.1016/j.acha.2015.12.005. S2CID 12149203.