**Implementing Computer Vision with Neural Networks**

Lee Yieng Cassandra (1155015264)

Abstract

Neural networks have been deployed in computer vision to give more effective results than traditional computer vision techniques. They are computational infrastructure organized in a similar way as biological neural networks. Different types of neural networks are used for computer vision tasks of various natures, such as convolutional neural networks and recurrent neural networks. The basic architecture of neural networks will be explained in this paper. This paper will concentrate on the neural network implementations of classification, detection, tracking and computer vision geometry methods. Most computer vision neural network applications use convolutional neural networks, but there are exceptions for image sequence-based methods.

Introduction

According to [1], the neural network approach to computer vision produces more accurate results than traditional approaches such as SIFT. Here, a brief survey of the use of neural networks in computer vision is presented.

Advanced computer vision applications can be produced through neural networks. Examples include image classification, object detection, video classification, object tracking, edge detection, human pose estimation, video captioning, image generation, and even answering image-based queries. [2] This paper will mainly discuss classification, detection, tracking and computer vision geometry methods using neural networks.

Theory and background

A neural network is a set of connected computational units called neurons that, like biological neurons, are activated when signals between them are strong enough. These activated neurons then pass the signals on to other neighboring neurons. A special feature of neural networks is that these systems learn to generalize from a set of training examples implicitly, unlike explicitly delineated rules as in traditional computational techniques. These neurons are comprised of an activation function which takes the strength of the incoming signals as the input. Common activation functions include the hyperbolic tangent function tanh(*x*) and the ramp function max(0,*x*). The neurons using the latter are called ReLU for “rectified linear unit”. These neurons are them separated into layers from which input is processed and passed on to the next layer, until it has gone through a sufficient number of passes. Max pooling layers down-sample the inputs and outputs of the previous neuron layers so that there are fewer dimensions for the neural network to consider when building the model. There may also be a dropout layer to prevent over-fitting. This is because the power of neural networks enables them to find the best fit of the data, likely at the expense of being too specific.

Neural networks applicable to computer vision come in different flavors. For example, convolutional neural networks contain a convolutional layer, where the inputs are convolved with a filter which is to be determined. Recurrent neural networks continually feed output model parameters of the neural network back into itself. There are also residual neural networks, which are similar to recurrent neural networks, except that instead of the output model, the feature differences between layers are fed back into the network.

To compare the performance of the neural network model and the actual labels of the data (if provided), cost functions are defined. For example, a simple cost function would be the absolute square of the difference of the predicted labels and the actual labels. The cost function is to be minimized while the neural network constructs the model. Neural network libraries come with optimization routines using techniques such as gradient descent, and such routines are deployed in the hope that the model eventually converges and the error introduced through the neural network abstraction is not too large.

Existing implementation and procedures

In computer vision, once the image data has been pre-processed, the neural networks used to perform various tasks on them are fairly straightforward. The neural networks have often been used with GPUs for speedy processing.

For image or object classification purposes, convolutional neural networks are used. [3] [4] This type of neural network has the advantage that it can pick up abstract features readily. For some models, the deeper (more layered) the convolutional neural network, the higher the accuracy. [4] This may be attributed to normalization of the input/output data. [5] In suitably deep models, the number of layers reach a critical point after which the trend reverses and the accuracy lowers when more layers are added, and so some have implemented a residual neural network with good performance. [5] Existing convolutional neural networks have the drawback of only admitting fixed-length input vectors and producing fixed-length output vectors. [6]

For object detection or video classification, recurrent neural networks have a role to play because they can be used to identify local or important features of a sequence of images. This is because recurrent neural networks accept sequences of inputs and generate output sequences, which is a major improvement over their convolutional counterparts. [6]

Convolutional neural networks are also used for image detection and object tracking. For image detection, they require an additional edge-aware filter function for it to identify the edges. [7] In this case, the neural network identifies the regions in the image highlighted by this function and updates the convolutional weights associated with the model. For object tracking, the neural networks may need to be pre-trained and other machine learning methods such as support vector machine may need to be used because the neural network may overgeneralize and ignore the non-target information (such as background) of the image, which contains spatial information about the target being tracked. [8] To overcome such a shortcoming, the convolutional neural networks for object tracking differ from the other neural networks mentioned in this paper in that features from all layers of the model are extracted to form the model, rather than aiming for convergence in the final layer and using the final layer as the ultimate output. [8]

More recently, neural networks were used for basic computer vision geometry. [9] In brief, the neural network model layers make use of mathematical properties to analyze pairs of consecutive images and determine the geometric transformations that transformed the former image to the latter.

Discussion and conclusion

In general, neural networks are a superior method for computer vision tasks. They can process huge amounts of data and are scalable. Different neural networks may be designed for different computer vision tasks.

A drawback of neural networks is that the model is basically a “black box” and the actual decisions in the models for computer vision could not be measured or studied in depth. It would be instructive if there were attempts at understanding the decision-making operations that are used to create the models, so that errors could be dissected and analyzed, and that the features identified by human beings can be matched or compared with those that the neural network identified.

Nevertheless, it is a viable and effective, as well as often efficient, method for various computer vision tasks. The application of neural networks on computer vision has prompted much research, and it is anticipated that more studies in this field will be done.

References

[1] Andy Lee, *Comparing Deep Neural Networks and Traditional Vision Algorithms in Mobile Robotics*, <https://www.cs.swarthmore.edu/~meeden/cs81/f15/papers/Andy.pdf>

[2] Awesome Deep Vision <https://github.com/kjw0612/awesome-deep-vision>

[3] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*, NIPS, 2012.

[4] Karen Simonyan and Andrew Zisserman, *Very Deep Convolutional Networks for Large-Scale Visual Recognition*, ICLR, 2015.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, *Deep Residual Learning for Image Recognition*, arXiv:1512.03385.

[6] Feng Wang, David M. J. Tax, *Survey on the attention based RNN model and its applications in computer vision*, <https://arxiv.org/abs/1601.06823>

[7] Li Xu, Jimmy SJ. Ren, Qiong Yan, Renjie Liao, Jiaya Jia, *Deep Edge-Aware Filters*, ICML, 2015. <http://proceedings.mlr.press/v37/xub15.pdf>

[8] Chao Ma, Jia-Bin Huang, Xiaokang Yang and Ming-Hsuan Yang, *Hierarchical Convolutional Features for Visual Tracking*, ICCV 2015 <http://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Ma_Hierarchical_Convolutional_Features_ICCV_2015_paper.pdf>

[9] Ankur Handa, Michael Bloesch, Viorica Patraucean, Simon Stent, John McCormac, Andrew Davison, *gvnn: Neural Network Library for Geometric Computer Vision*, arXiv:1607.07405v3 [cs.CV]