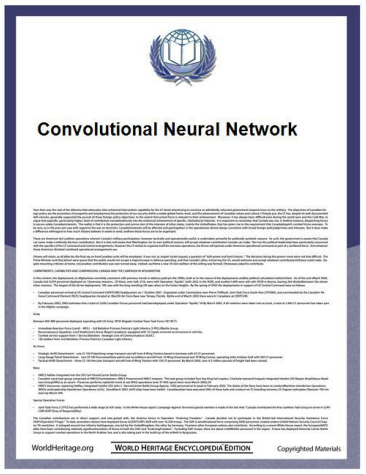


CONVOLUTIONAL NEURAL NETWORK

Article Id: WHEBN0040409788
Reproduction Date:

Title: *Convolutional neural network*
Author: *World Heritage Encyclopedia*
Language: *English*
Subject: *Deep learning, Yann LeCun, MNIST database, Artificial neural network, Computer vision*
Collection: *Artificial Neural Networks, Computational Neuroscience*
Publisher: *World Heritage Encyclopedia*
Publication Date:

CONVOLUTIONAL NEURAL NETWORK

In machine learning, a **convolutional neural network** (**CNN**, or **ConvNet**) is a type of feed-forward artificial neural network where the individual neurons are tiled in such a way that they respond to overlapping regions in the visual field.[1] Convolutional networks were inspired by biological processes[2] and are variations of multilayer perceptrons which are designed to use minimal amounts of preprocessing.[3] They are widely used models for image and video recognition.

OVERVIEW

When used for image recognition, convolutional neural networks (CNNs) consist of multiple layers of small neuron collections which look at small portions of the input image, called receptive fields. The results of these collections are then tiled so that they overlap to obtain a better representation of the original image; this is repeated for every such layer. Because of this, they are able to tolerate translation of the input image.[4] Convolutional networks may include local or global pooling layers, which combine the outputs of neuron clusters.[5][6] They also consist of various combinations of convolutional layers and fully connected layers, with pointwise nonlinearity applied at the end of or after each layer.[7] It is inspired by biological processes. To avoid the situation that there exist billions of parameters if all layers are fully connected, the idea of using a convolution operation on small regions has been introduced. One major advantage of convolutional networks is the use of shared weight in convolutional layers, which means that the same filter (weights bank) is used for each pixel in the layer; this both reduces required memory size and improves performance.[3]

Some time delay neural networks also use a very similar architecture to convolutional neural networks, especially those for image recognition and/or classification tasks, since the "tiling" of the neuron outputs can easily be carried out in timed stages in a manner useful for analysis of images.[8]

Compared to other image classification algorithms, convolutional neural networks use relatively little pre-processing. This means that the network is responsible for learning the filters that in traditional algorithms were hand-engineered. The lack of a dependence on prior-knowledge and the existence of difficult to design hand-engineered features is a major advantage for CNNs.

CONTENTS

Machine learning and data mining

Problems
Classification | Clustering | Regression | Anomaly detection | Association rules | Reinforcement learning | Structured prediction | Feature engineering | Feature learning | Online learning | Semi-supervised learning | Unsupervised learning | Learning to rank | Grammar induction

Supervised learning **(classification • regression)**
Decision trees | Ensembles (Bagging, Boosting, Random forest) | *k*-NN | Linear regression | Naive Bayes | Neural networks | Logistic regression | Perceptron | Relevance vector machine (RVM) | Support vector machine (SVM)

Clustering
BIRCH | Hierarchical | *k*-means | Expectation-maximization (EM) | DBSCAN | OPTICS | Mean-shift

Dimensionality reduction
Factor analysis | CCA | ICA | LDA | NMF | PCA | t-SNE

Structured prediction
Graphical models (Bayes net, CRF, HMM)

Anomaly detection
k-NN | Local outlier factor

Neural nets
Autoencoder | Deep learning | Multilayer perceptron | RNN | Restricted Boltzmann machine | SOM | **Convolutional neural network**

Theory
Bias-variance dilemma | Computational learning theory | Empirical risk minimization | PAC learning | Statistical learning | VC theory

Machine learning portal
Computer science portal
Statistics portal

HISTORY

The design of convolutional neural networks follows the discovery of visual mechanisms in living organisms. In our brain, the visual cortex contains lots of cells. These cells are responsible for detecting light in small, overlapping sub-regions of the visual field, called receptive fields. These cells act as local filters over the input space. The more complex cells have larger receptive fields. A convolution operator is created to perform the same function by all of these cells.

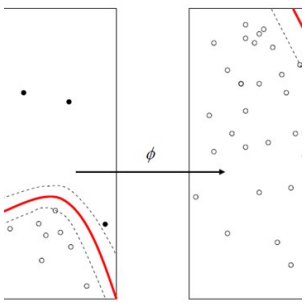
The neocognitron, a predecessor to convolutional networks,[9] was introduced in a 1980 paper by Kunihiko Fukushima.[7][10] In 1988 they were separately developed, with explicit parallel and trainable convolutions for temporal signals, by Toshiteru Homma, Les Atlas, and Robert J. Marks II.[11] Their design was later improved in 1998 by Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner,[12] generalized in 2003 by Sven Behnke,[13] and simplified by Patrice Simard, David Steinkraus, and John C. Platt in the same year.[14] The famous LeNet-5 network can classify digits successfully, which is applied to recognize checking numbers. However, given more complex problems the breadth and depth of the network will continue to increase which would become limited by computing resources. The approach used by LeNet did not perform well with more complex problems.

A different convolutional neural network design was proposed in 1988 by Daniel Graupe, Ruey Wen Liu and George S Moschytz.[15] for applications to decomposition of one-dimensional EMG signals, This design was further modified in 1989 to other convolution-based designs by Daniel Graupe, Boris Vern, Greg Gruener, Aaron S. Field and Qiu Huang in [16] and by Qiu Huang, Daniel Graupe and Yih Fang Huang in.[17]

With the rise of efficient GPU computing, it has become possible to train larger networks. In 2006 several publications described more efficient ways to train convolutional neural networks with more layers.[18][19][20] In 2011, they were refined by Dan Ciresan et al. and were implemented on a GPU with impressive performance results.[5] In 2012, Dan Ciresan et al. significantly improved upon the best performance in the literature for multiple image databases, including the MNIST database, the NORB database, the HWDB1.0 dataset (Chinese characters), the CIFAR10 dataset (dataset of 60000 32x32 labeled RGB images),[7] and the ImageNet dataset.[21]

DETAILS

BACKPROPAGATION



When doing propagation, the momentum and weight decay values are chosen to reduce oscillation during stochastic gradient descent. See Backpropagation for more.

DIFFERENT TYPES OF LAYERS

CONVOLUTIONAL LAYER

Unlike a hand-coded convolution kernel (Sobel, Prewitt, Roberts), in a convolutional neural net, the parameters of each convolution kernel are trained by the backpropagation algorithm. There are many convolution kernels in each layer, and each kernel is replicated over the entire image with the same parameters. The function of the convolution operators is to extract different features of the input. The capacity of a neural net varies, depending on the number of layers. The first convolution layers will obtain the low-level features, like edges, lines and corners. The more layers the network has, the higher-level features it will get.

CATEGORIES

COMPUTER SCIENCE
ENCYCLOPEDIA ARTICLE
Cryptography, Artificial intelligence, Software engineering, Science, Machine learning
READ MORE

STATISTICS
ENCYCLOPEDIA ARTICLE
Probability theory, Regression analysis, Mathematics, Observational study, Calculus
READ MORE

SUGGESTIONS

DEEP LEARNING
ENCYCLOPEDIA ARTICLE
Machine learning, Artificial intelligence, Jürgen Schmidhuber, Sepp Hochreiter, Restricted Boltzmann machine
READ MORE

YANN LECUN
ENCYCLOPEDIA ARTICLE
Computer Vision, New York City, Authority control, Machine learning, Paris
READ MORE

RELU LAYER

ReLU is the abbreviation of Rectified Linear Units. This is a layer of neurons that use the non-saturating activation function f(x)=max(0,x). It increases the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the convolution layer.

Other functions are used to increase nonlinearity. For example the saturating hyperbolic tangent f(x)=tanh(x), f(x)=|tanh(x)|, and the sigmoid function f(x)=(1+e^{-x})^{-1}. Compared to tanh units, the advantage of ReLU is that the neural network trains several times faster.[22]

POOLING LAYER

In order to reduce variance, pooling layers compute the max or average value of a particular feature over a region of the image. This will ensure that the same result will be obtained, even when image features have small translations. This is an important operation for object classification and detection.

DROPOUT METHOD

Since a fully connected layer occupies most of the parameters, it is prone to overfitting. The dropout method [23] is introduced to prevent overfitting. At each training stage, individual nodes are either "dropped out" of the net with probability 1-p or kept with probability p, so that a reduced network is left; incoming and outgoing edges to a dropped-out node are also removed. Only the reduced network is trained on the data in that stage. The removed nodes are then reinserted into the network with their original weights.

In the training stages, the probability a hidden node will be retained (i.e. not dropped) is usually 0.5; for input nodes the retention probability should be much higher, intuitively because information is directly lost when input nodes are ignored.

At testing time after training has finished, we would ideally like to find a sample average of all possible 2^n dropped-out networks; unfortunately this is infeasible for large n. However, we can find an approximation by using the full network with each node's output weighted by a factor of p, so the expected value of the output of any node is the same as in the training stages. This is the biggest contribution of the dropout method: although it effectively generates 2^n neural nets, and as such allows for model combination, at test time only a single network needs to be tested.

By avoiding training all nodes on all training data, dropout decreases overfitting in neural nets. The method also significantly improves the speed of training. This makes model combination practical, even for deep neural nets.

LOSS LAYER

It can use different loss functions for different tasks. Softmax loss is used for predicting a single class of K mutually exclusive classes. Sigmoid cross-entropy loss is used for predicting K independent probability values in [0,1]. Euclidean loss is used for regressing to real-valued labels [-inf,inf]

HIERARCHICAL COORDINATE FRAMES

Pooling in convolutional networks loses the precise spatial relationships between high-level parts (such as nose and mouth in a face image). The precise spatial relationships are needed for identity recognition. Overlapping the pools so that each feature occurs in different pools, helps retain the information about the position of a feature. But convolutional nets that just use translation cannot extrapolate their understanding of geometric relationships to a radically new viewpoints, like different orientations or different scales. On the other hand, people are very good at extrapolating, after seeing a new shape once they can recognize it from different viewpoint.[24]

Currently the common way to deal with this problem is to train the convolutional nets on transformed data in different orientations, scales, lighting etc. so that the network can cope with these variations which is extremely computationally intensive process on large datasets. The alternative is to use a hierarchy of coordinate frames and to use a group of neurons to represent a conjunction of the shape of the feature and its pose relative to the retina. The pose relative to retina is the relationship between the coordinate frame of the retina and the intrinsic coordinate frame of the feature.[25]

So in order to represent something we have to embed the coordinate frame within it. Once we've done that we can recognize large features by using the consistency of poses of their parts (e.g. nose and mouth poses make consistent prediction of the pose of the whole face). Using this approach we can tell if the higher level entity (e.g. face) is present if the lower level visual entities (e.g. nose and mouth) can agree on its prediction of the pose. The vectors of neurons activity that represent pose ("pose vectors") allow spatial transformations modeled as linear operations that make it easy to learn the hierarchy of visual entities and generalize across viewpoints. This corresponds to the way human visual system imposes coordinate frames in order to represent shapes.[26]

APPLICATIONS

IMAGE RECOGNITION

Convolutional neural networks are often used in image recognition systems. They have achieved an error rate of 0.23 percent on the MNIST database, which as of February 2012 is the lowest achieved on the database.[7] Another paper on using CNN for image classification reported that the learning process was "surprisingly fast"; in the same paper, the best published results at the time were achieved in the MNIST database and the NORB database.[5]

When applied to facial recognition, they were able to contribute to a large decrease in error rate.[27] In another paper, they were able to achieve a 97.6 percent recognition rate on "5,600 still images of more than 10 subjects".[2] CNNs have been used to assess video quality in an objective way after being manually trained; the resulting system had a very low root mean square error.[8]

The ImageNet Large Scale Visual Recognition Challenge is a benchmark in object classification and detection, with millions of images and hundreds of object classes. In the ILSVRC 2014, which is large-scale visual recognition challenge, almost every highly ranked team used CNN as their basic framework. The winner GoogLeNet[28] (the foundation of DeepDream) increased the mean average precision of object detection to 0.439329, and reduced classification error to 0.06656, the best result to date. Its network applied more than 30 layers. Performance of convolutional neural networks, on the ImageNet tests, is now close to that of humans.[29] The best algorithms still struggle with objects that are small or thin, such as a small ant on a stem of a flower or a person holding a quill in their hand. They also have trouble with images that have been distorted with filters, an increasingly common phenomenon with modern digital cameras. By contrast, those kinds of images rarely trouble humans. Humans, however, tend to have trouble with other issues. For example, they are not good at classifying objects into fine-grained categories such as the particular breed of dog or species of bird, whereas convolutional neural networks handle this with ease.

In 2015 a many-layered CNN demonstrated the ability to spot faces from a wide range of angles, including upside down, even when partially occluded with competitive performance. The network trained on a database of 200,000 images that included faces at various angles and orientations and a further 20 million images without faces. They used batches of 128 images over 50,000 iterations.[30]

VIDEO ANALYSIS

Video is more complex than images since it has another (temporal) dimension. The common way is to fuse the features of different convolutional neural networks, which are responsible for spatial and temporal stream.[31][32]

NATURAL LANGUAGE PROCESSING

Convolutional neural networks have also seen use in the field of natural language processing or NLP. Like the image classification problem, some NLP tasks can be formulated as assigning labels to words in a sentence. The neural network trained raw material fashion will extract the features of the sentences. Using some classifiers, it could predict new sentences.[33]

PLAYING GO

Convolutional neural networks have been used in computer Go. In December 2014, Christopher Clark and Amos Storkey published a paper showing a convolutional network trained by supervised learning from a database of human professional games could outperform Gnu Go and win some games against Monte Carlo tree search Fuego 1.1 in a fraction of the time it took Fuego to play.[34] Shortly after it was announced that a large 12-layer convolutional neural network had correctly predicted the professional move in 55% of positions, equalling the accuracy of a 6 dan human player. When the trained convolutional network was used directly to play games of Go, without any search, it beat the traditional search program GNU Go in 97% of games, and matched the performance of the Monte Carlo tree search program Fuego simulating ten thousand playouts (about a million positions) per move.[35]

FINE-TUNING

For many applications, only a small amount of training data is available. Convolutional neural networks usually require a large amount of training data in order to avoid overfitting. A common technique is to train the network on a larger data set from a related domain. Once the network parameters have converged an additional training step is performed using the in-domain data to fine-tune the network weights. This allows convolutional networks to be successfully applied to problems with small training sets.

COMMON LIBRARIES

Caffe: Caffe (replacement of Decaf) has been the most popular library for convolutional neural networks. It is created by the Berkeley Vision and Learning Center (BVLC). The advantages are that it has cleaner architecture and greater speed. It supports both CPU and GPU, easily switching between them. It is developed in C++, and has Python and MATLAB wrappers. In the developing of Caffe, protobuf is used to make researchers tune the parameters easily as well as adding or removing layers.
Torch (www.torch.ch): A scientific computing framework with wide support for machine learning algorithms, written in C and lua. The main author is Ronan Collobert, and it is now widely used at Facebook AI Research, Google DeepMind and Twitter, among others.
OverFeat: A pre-trained feature extractor by Pierre Sermanet.
Cuda-convnet: A convnet implementation in CUDA
MatConvnet
Theano: written in Python with an API largely compatible with the popular Numpy library. Allows user to write symbolic mathematical expressions, then automatically generates their derivatives, saving the user from having to code gradients or backpropagation. These symbolic expressions are automatically compiled to CUDA code for a fast, on-the-GPU implementation.
Deeplearning4j: Deep learning in Java and Scala on GPU-enabled Spark

SEE ALSO

Neocognitron
Convolution
Deep learning
Time delay neural network

EXTERNAL LINKS

A demonstration of a convolutional network created for character recognition
Caffe
Matlab toolbox
MatConvnet
Theano
UFLDL Tutorial
Deeplearning4j's Convolutional Nets