

Title: Convolutional layer

URL: https://en.wikipedia.org/wiki/Convolutional_layer

PageID: 59211466

Categories: Category:Artificial neural networks, Category:Computer vision, Category:Deep learning

Source: Wikipedia (CC BY-SA 4.0).

Supervised learning

Unsupervised learning

Semi-supervised learning

Self-supervised learning

Reinforcement learning

Meta-learning

Online learning

Batch learning

Curriculum learning

Rule-based learning

Neuro-symbolic AI

Neuromorphic engineering

Quantum machine learning

Classification

Generative modeling

Regression

Clustering

Dimensionality reduction

Density estimation

Anomaly detection

Data cleaning

AutoML

Association rules

Semantic analysis

Structured prediction

Feature engineering

Feature learning

Learning to rank

Grammar induction

Ontology learning

Multimodal learning

Apprenticeship learning

Decision trees

Ensembles Bagging Boosting Random forest

Bagging

Boosting

Random forest

k -NN

Linear regression

Naive Bayes

Artificial neural networks

Logistic regression

Perceptron

Relevance vector machine (RVM)

Support vector machine (SVM)

BIRCH

CURE

Hierarchical

k -means

Fuzzy

Expectation–maximization (EM)

DBSCAN

OPTICS

Mean shift

Factor analysis

CCA

ICA

LDA

NMF

PCA

PGD

t-SNE

SDL

Graphical models Bayes net Conditional random field Hidden Markov

Bayes net

Conditional random field

Hidden Markov

RANSAC

k -NN

Local outlier factor
Isolation forest
Autoencoder
Deep learning
Feedforward neural network
Recurrent neural network LSTM GRU ESN reservoir computing
LSTM
GRU
ESN
reservoir computing
Boltzmann machine Restricted
Restricted
GAN
Diffusion model
SOM
Convolutional neural network U-Net LeNet AlexNet DeepDream
U-Net
LeNet
AlexNet
DeepDream
Neural field Neural radiance field Physics-informed neural networks
Neural radiance field
Physics-informed neural networks
Transformer Vision
Vision
Mamba
Spiking neural network
Memtransistor
Electrochemical RAM (ECRAM)
Q-learning
Policy gradient
SARSA
Temporal difference (TD)
Multi-agent Self-play
Self-play
Active learning
Crowdsourcing
Human-in-the-loop

Mechanistic interpretability

RLHF

Coefficient of determination

Confusion matrix

Learning curve

ROC curve

Kernel machines

Bias–variance tradeoff

Computational learning theory

Empirical risk minimization

Occam learning

PAC learning

Statistical learning

VC theory

Topological deep learning

AAAI

ECML PKDD

NeurIPS

ICML

ICLR

IJCAI

ML

JMLR

Glossary of artificial intelligence

List of datasets for machine-learning research List of datasets in computer vision and image processing

List of datasets in computer vision and image processing

Outline of machine learning

v

t

e

In artificial neural networks , a convolutional layer is a type of network layer that applies a convolution operation to the input. Convolutional layers are some of the primary building blocks of convolutional neural networks (CNNs), a class of neural network most commonly applied to images, video, audio, and other data that have the property of uniform translational symmetry . [1]

The convolution operation in a convolutional layer involves sliding a small window (called a kernel or filter) across the input data and computing the dot product between the values in the kernel and the input at each position. This process creates a feature map that represents detected features in the input. [2]

Concepts

Kernel

Kernels, also known as filters, are small matrices of weights that are learned during the training process. Each kernel is responsible for detecting a specific feature in the input data. The size of the kernel is a hyperparameter that affects the network's behavior.

Convolution

For a 2D input x and a 2D kernel w , the 2D convolution operation can be expressed as: $y[i, j] = \sum_{m=0}^{k_h-1} \sum_{n=0}^{k_w-1} x[i+m, j+n] \cdot w[m, n]$ where k_h and k_w are the height and width of the kernel, respectively.

This generalizes immediately to nD convolutions. Commonly used convolutions are 1D (for audio and text), 2D (for images), and 3D (for spatial objects, and videos).

Stride

Stride determines how the kernel moves across the input data. A stride of 1 means the kernel shifts by one pixel at a time, while a larger stride (e.g., 2 or 3) results in less overlap between convolutions and produces smaller output feature maps.

Padding

Padding involves adding extra pixels around the edges of the input data. It serves two main purposes:

Preserving spatial dimensions: Without padding, each convolution reduces the size of the feature map.

Handling border pixels: Padding ensures that border pixels are given equal importance in the convolution process.

Common padding strategies include:

No padding/valid padding: This strategy typically causes the output to shrink.

Same padding: Any method that ensures the output size same as input size is a same padding strategy.

Full padding: Any method that ensures each input entry is convolved over for the same number of times is a full padding strategy.

Common padding algorithms include:

Zero padding: Add zero entries to the borders of input.

Mirror/reflect/symmetric padding: Reflect the input array on the border.

Circular padding: Cycle the input array back to the opposite border, like a torus.

The exact numbers used in convolutions is complicated, for which we refer to (Dumoulin and Visin, 2018) [3] for details.

Variants

Standard

The basic form of convolution as described above, where each kernel is applied to the entire input volume.

Depthwise separable

Depthwise separable convolution separates the standard convolution into two steps: depthwise convolution and pointwise convolution. The depthwise separable convolution decomposes a single standard convolution into two convolutions: a depthwise convolution that filters each input channel independently and a pointwise convolution (1×1 convolution) that combines the outputs of the depthwise convolution. This factorization significantly reduces

computational cost. [4]

It was first developed by Laurent Sifre during an internship at Google Brain in 2013 as an architectural variation on AlexNet to improve convergence speed and model size. [4]

Dilated

Dilated convolution , or atrous convolution , introduces gaps between kernel elements, allowing the network to capture a larger receptive field without increasing the kernel size. [5] [6]

Transposed

Transposed convolution , also known as deconvolution , fractionally strided convolution , and upsampling convolution , is a convolution where the output tensor is larger than its input tensor. It's often used in encoder-decoder architectures for upsampling. It's used in image generation, semantic segmentation , and super-resolution tasks.

History

The concept of convolution in neural networks was inspired by the visual cortex in biological brains. Early work by Hubel and Wiesel in the 1960s on the cat's visual system laid the groundwork for artificial convolution networks. [7]

An early convolution neural network was developed by Kunihiko Fukushima in 1969. It had mostly hand-designed kernels inspired by convolutions in mammalian vision. [8] In 1979 he improved it to the Neocognitron , which learns all convolutional kernels by unsupervised learning (in his terminology, " self-organized by 'learning without a teacher'"). [9] [10]

During the 1988 to 1998 period, a series of CNN were introduced by Yann LeCun et al., ending with LeNet-5 in 1998. It was an early influential CNN architecture for handwritten digit recognition, trained on the MNIST dataset , and was used in ATM . [11]

(Olshausen & Field, 1996) [12] discovered that simple cells in the mammalian primary visual cortex implement localized, oriented, bandpass receptive fields, which could be recreated by fitting sparse linear codes for natural scenes. This was later found to also occur in the lowest-level kernels of trained CNNs. [13] : Fig 3

The field saw a resurgence in the 2010s with the development of deeper architectures and the availability of large datasets and powerful GPUs. AlexNet , developed by Alex Krizhevsky et al. in 2012, was a catalytic event in modern deep learning . [13] [14] In that year's ImageNet competition , the AlexNet model achieved a 16% top-five error rate, significantly outperforming the next best entry, which had a 26% error rate. The network used eight trainable layers, approximately 650,000 neurons, and around 60 million parameters, highlighting the impact of deeper architectures and GPU acceleration on image recognition performance. [14]

From the 2013 ImageNet competition, most entries adopted deep convolutional neural networks, building on the success of AlexNet. Over the following years, performance steadily improved, with the top-five error rate falling from 16% in 2012 and 12% in 2013 to below 3% by 2017, as networks grew increasingly deep. [14]

See also

Convolutional neural network

Pooling layer

Feature learning

Deep learning

Computer vision

References

v

t

e

Differentiable programming

Information geometry

Statistical manifold

Automatic differentiation

Neuromorphic computing

Pattern recognition

Ricci calculus

Computational learning theory

Inductive bias

IPU

TPU

VPU

Memristor

SpiNNaker

TensorFlow

PyTorch

Keras

scikit-learn

Theano

JAX

Flux.jl

MindSpore

Portals Computer programming Technology

Computer programming

Technology