Title: Tensor (machine learning)

URL: https://en.wikipedia.org/wiki/Tensor_(machine_learning)

PageID: 73050688

Categories: Category:Arrays, Category:Machine learning, Category:Tensors

---

In machine learning , the term tensor informally refers to two different concepts (i) a way of organizing data and (ii) a multilinear (tensor) transformation. Data may be organized in a multidimensional array (M -way array), informally referred to as a "data tensor"; however, in the strict mathematical sense, a tensor is a multilinear mapping over a set of domain vector spaces to a range vector space. Observations, such as images, movies, volumes, sounds, and relationships among words and concepts, stored in an M -way array ("data tensor"), may be analyzed either by artificial neural networks or tensor methods . [1] [2] [3] [4] [5]

Tensor decomposition factorizes data tensors into smaller tensors. [1] [6] Operations on data tensors can be expressed in terms of matrix multiplication and the Kronecker product . [7] The computation of gradients, a crucial aspect of backpropagation , can be performed using software libraries such as PyTorch and TensorFlow . [8] [9]

Computations are often performed on graphics processing units (GPUs) using CUDA , and on dedicated hardware such as Google 's Tensor Processing Unit or Nvidia 's Tensor core . These developments have greatly accelerated neural network architectures, and increased the size and complexity of models that can be trained.

History

A tensor is by definition a multilinear map. In mathematics, this may express a multilinear relationship between sets of algebraic objects. In physics, tensor fields , considered as tensors at each point in space, are useful in expressing mechanics such as stress or elasticity . In machine learning, the exact use of tensors depends on the statistical approach being used.

In 2001, the field of signal processing and statistics were making use of tensor methods. Pierre Comon surveys the early adoption of tensor methods in the fields of telecommunications, radio surveillance, chemometrics and sensor processing. Linear tensor rank methods (such as, Parafac/CANDECOMP) analyzed M-way arrays ("data tensors") composed of higher order statistics that were employed in blind source separation problems to compute a linear model of the data. He noted several early limitations in determining the tensor rank and efficient tensor rank decomposition. [10]

In the early 2000s, multilinear tensor methods [1] [11] crossed over into computer vision, computer graphics and machine learning with papers by Vasilescu or in collaboration with Terzopoulos, such as Human Motion Signatures, [12] [13] TensorFaces [14] [15] TensorTextures [16] and Multilinear Projection. [17] [18] Multilinear algebra, the algebra of higher-order tensors, is a suitable and transparent framework for analyzing the multifactor structure of an ensemble of observations and for addressing the difficult problem of disentangling the causal factors based on second order [14] or higher order statistics associated with each causal factor. [15]

Tensor (multilinear) factor analysis disentangles and reduces the influence of different causal factors with multilinear subspace learning. [19] When treating an image or a video as a 2- or 3-way array, i.e., "data matrix/tensor", tensor methods reduce spatial or time redundancies as demonstrated by Wang and Ahuja. [20]

Yoshua Bengio, [21] [22] Geoff Hinton [23] [24] and their collaborators briefly discuss the relationship between deep neural networks and tensor factor analysis [14] [15] beyond the use of M-way arrays ("data tensors") as inputs. One of the early uses of tensors for neural networks appeared in natural language processing . A single word can be expressed as a vector via

Word2vec . [5] Thus a relationship between two words can be encoded in a matrix. However, for more complex relationships such as subject-object-verb, it is necessary to build higher-dimensional networks. In 2009, the work of Sutskever introduced Bayesian Clustered Tensor Factorization to model relational concepts while reducing the parameter space. [25] From 2014 to 2015, tensor methods become more common in convolutional neural networks (CNNs). Tensor methods organize neural network weights in a "data tensor", analyze and reduce the number of neural network weights. [26] [27] Lebedev et al. accelerated CNN networks for character classification (the recognition of letters and digits in images) by using 4D kernel tensors. [28]

Definition

Let F {\displaystyle \mathbb {F} } be a field such as the real numbers R {\displaystyle \mathbb {R} } or the complex numbers C {\displaystyle \mathbb {C} } . A tensor T ∈ F I 0 × I 2 × … × I C {\displaystyle {\mathcal {T}}\in {\mathbb {F} }^{I_{0}\times I_{2}\times \ldots \times I_{C}}} is a multilinear transformation from a set of domain vector spaces to a range vector space:

T : { F I 1 × F I 2 × … F I C } ■ F I 0 {\displaystyle {\mathcal {T}}:\{{\mathbb {F} }^{I_{1}}\times {\mathbb {F} }^{I_{2}}\times \ldots {\mathbb {F} }^{I_{C}}\}\mapsto {\mathbb {F} }^{I_{0}}}

Here, C {\displaystyle C} and I 0 , I 1 , … , I C {\displaystyle I_{0},I_{1},\ldots ,I_{C}} are positive integers, and (C + 1) {\displaystyle (C+1)} is the number of modes of a tensor (also known as the number of ways of a multi-way array). The dimensionality of mode c {\displaystyle c} is I c {\displaystyle I_{c}} , for 0 ≤ c ≤ C {\displaystyle 0\leq c\leq C} . [14] [15] [29] [5]

In statistics and machine learning, an image is vectorized when viewed as a single observation, and a collection of vectorized images is organized as a "data tensor". For example, a set of facial images { d i p , i e , i l , i v ∈ R I X } {\displaystyle \{{\mathbb {d} }_{i_{p},i_{e},i_{l},i_{v}}\in {\mathbb {R} }^{I_{X}}\}} with I X {\displaystyle I_{X}} pixels that are the consequences of multiple causal factors, such as a facial geometry i p (1 ≤ i p ≤ I P) {\displaystyle i_{p}(1\leq i_{p}\leq I_{P})} , an expression i e (1 ≤ i e ≤ I E) {\displaystyle i_{e}(1\leq i_{e}\leq I_{E})} , an illumination condition i l (1 ≤ i l ≤ I L) {\displaystyle i_{l}(1\leq i_{l}\leq I_{L})} , and a viewing condition i v (1 ≤ i v ≤ I V) {\displaystyle i_{v}(1\leq i_{v}\leq I_{V})} may be organized into a data tensor (ie. multiway array) D ∈ R I X × I P × I E × I L × V {\displaystyle {\mathcal {D}}\in {\mathbb {R} }^{I_{X}\times I_{P}\times I_{E}\times I_{L}\times V}} where I P {\displaystyle I_{P}} are the total number of facial geometries, I E {\displaystyle I_{E}} are the total number of expressions, I L {\displaystyle I_{L}} are the total number of illumination conditions, and I V {\displaystyle I_{V}} are the total number of viewing conditions. Tensor factorizations methods such as TensorFaces and multilinear (tensor) independent component analysis factorizes the data tensor into a set of vector spaces that span the causal factor representations, where an image is the result of tensor transformation T {\displaystyle {\mathcal {T}}} that maps a set of causal factor representations to the pixel space.

Another approach to using tensors in machine learning is to embed various data types directly. For example, a grayscale image , commonly represented as a discrete 2-way array D ∈ R I R X × I C X {\displaystyle {\mathbf {D} }\in {\mathbb {R} }^{I_{RX}\times I_{CX}}} with dimensionality I R X × I C X {\displaystyle I_{RX}\times I_{CX}} where I R X {\displaystyle I_{RX}} are the number of rows and I C X {\displaystyle I_{CX}} are the number of columns. When an image is treated as 2-way array or 2nd order tensor (i.e. as a collection of column/row observations), tensor factorization methods compute the image column space, the image row space and the normalized PCA coefficients or the ICA coefficients.

Similarly, a color image with RGB channels, D ∈ R N × M × 3 . {\displaystyle {\mathcal {D}}\in \mathbb {R} ^{N\times M\times 3}.} may be viewed as a 3rd order data tensor or 3-way array.--------

In natural language processing, a word might be expressed as a vector v {\displaystyle v} via the Word2vec algorithm. Thus v {\displaystyle v} becomes a mode-1 tensor

The embedding of subject-object-verb semantics requires embedding relationships among three words. Because a word is itself a vector, subject-object-verb semantics could be expressed using mode-3 tensors

In practice the neural network designer is primarily concerned with the specification of embeddings, the connection of tensor layers, and the operations performed on them in a network. Modern

machine learning frameworks manage the optimization, tensor factorization and backpropagation automatically.

As unit values

Tensors may be used as the unit values of neural networks which extend the concept of scalar, vector and matrix values to multiple dimensions.

The output value of single layer unit y m {\displaystyle y_{m}} is the sum-product of its input units and the connection weights filtered through the activation function f {\displaystyle f} :

where

If each output element of y m {\displaystyle y_{m}} is a scalar, then we have the classical definition of an artificial neural network . By replacing each unit component with a tensor, the network is able to express higher dimensional data such as images or videos:

This use of tensors to replace unit values is common in convolutional neural networks where each unit might be an image processed through multiple layers. By embedding the data in tensors such network structures enable learning of complex data types.

In fully connected layers

Tensors may also be used to compute the layers of a fully connected neural network, where the tensor is applied to the entire layer instead of individual unit values.

The output value of single layer unit y m {\displaystyle y_{m}} is the sum-product of its input units and the connection weights filtered through the activation function f {\displaystyle f} :

The vectors x {\displaystyle x} and y {\displaystyle y} of output values can be expressed as a mode-1 tensors, while the hidden weights can be expressed as a mode-2 tensor. In this example the unit values are scalars while the tensor takes on the dimensions of the network layers:

In this notation, the output values can be computed as a tensor product of the input and weight tensors:

which computes the sum-product as a tensor multiplication (similar to matrix multiplication).

This formulation of tensors enables the entire layer of a fully connected network to be efficiently computed by mapping the units and weights to tensors.

In convolutional layers

A different reformulation of neural networks allows tensors to express the convolution layers of a neural network. A convolutional layer has multiple inputs, each of which is a spatial structure such as an image or volume. The inputs are convolved by filtering before being passed to the next layer. A typical use is to perform feature detection or isolation in image recognition.

Convolution is often computed as the multiplication of an input signal g {\displaystyle g} with a filter kernel f {\displaystyle f} . In two dimensions the discrete, finite form is:

where w {\displaystyle w} is the width of the kernel.

This definition can be rephrased as a matrix-vector product in terms of tensors that express the kernel, data and inverse transform of the kernel. [31]

where A , B {\displaystyle {\mathcal {A}},{\mathcal {B}}} and C {\displaystyle {\mathcal {C}}} are the inverse transform, data and kernel. The derivation is more complex when the filtering kernel also includes a non-linear activation function such as sigmoid or ReLU .

The hidden weights of the convolution layer are the parameters to the filter. These can be reduced with a pooling layer which reduces the resolution (size) of the data, and can also be expressed as a tensor operation.

Tensor factorization

An important contribution of tensors in machine learning is the ability to factorize tensors to decompose data into constituent factors or reduce the learned parameters. Data tensor modeling

techniques stem from the linear tensor decomposition (CANDECOMP/Parafac decomposition) and the multilinear tensor decompositions (Tucker).

Tucker decomposition

Tucker decomposition , for example, takes a 3-way array X ∈ R I × J × K {\displaystyle {\mathcal {X}}\in \mathbb {R} ^{I\times J\times K}} and decomposes the tensor into three matrices A , B , C {\displaystyle {\mathcal {A,B,C}}} and a smaller tensor G {\displaystyle {\mathcal {G}}} . The shape of the matrices and new tensor are such that the total number of elements is reduced. The new tensors have shapes

Then the original tensor can be expressed as the tensor product of these four tensors:

In the example shown in the figure, the dimensions of the tensors are

The total number of elements in the Tucker factorization is

The number of elements in the original X {\displaystyle {\mathcal {X}}} is 144, resulting in a data reduction from 144 down to 110 elements, a reduction of 23% in parameters or data size. For much larger initial tensors, and depending on the rank (redundancy) of the tensor, the gains can be more significant.

The work of Rabanser et al. provides an introduction to tensors with more details on the extension of Tucker decomposition to N-dimensions beyond the mode-3 example given here. [5]

Tensor trains

Another technique for decomposing tensors rewrites the initial tensor as a sequence (train) of smaller sized tensors. A tensor-train (TT) is a sequence of tensors of reduced rank, called canonical factors . The original tensor can be expressed as the sum-product of the sequence.

Developed in 2011 by Ivan Oseledts, the author observes that Tucker decomposition is "suitable for small dimensions, especially for the three-dimensional case. For large d it is not suitable." [32] Thus tensor-trains can be used to factorize larger tensors in higher dimensions.

Tensor graphs

The unified data architecture and automatic differentiation of tensors has enabled higher-level designs of machine learning in the form of tensor graphs. This leads to new architectures, such as tensor-graph convolutional networks (TGCN), which identify highly non-linear associations in data, combine multiple relations, and scale gracefully, while remaining robust and performant. [33]

These developments are impacting all areas of machine learning, such as text mining and clustering, time varying data, and neural networks wherein the input data is a social graph and the data changes dynamically. [34] [35] [36] [37]

Hardware

Tensors provide a unified way to train neural networks for more complex data sets. However, training is expensive to compute on classical CPU hardware.

In 2014, Nvidia developed cuDNN , CUDA Deep Neural Network, a library for a set of optimized primitives written in the parallel CUDA language. [38] CUDA and thus cuDNN run on dedicated GPUs that implement unified massive parallelism in hardware. These GPUs were not yet dedicated chips for tensors, but rather existing hardware adapted for parallel computation in machine learning.

In the period 2015–2017 Google invented the Tensor Processing Unit (TPU). [39] TPUs are dedicated, fixed function hardware units that specialize in the matrix multiplications needed for tensor products. Specifically, they implement an array of 65,536 multiply units that can perform a 256x256 matrix sum-product in just one global instruction cycle. [40]

Later in 2017, Nvidia released its own Tensor Core with the Volta GPU architecture. Each Tensor Core is a microunit that can perform a 4x4 matrix sum-product. There are eight tensor cores for each shared memory (SM) block. [41] The first GV100 GPU card has 108 SMs resulting in 672 tensor cores. This device accelerated machine learning by 12x over the previous Tesla GPUs. [42] The number of tensor cores scales as the number of cores and SM units continue to grow in each

new generation of cards.

The development of GPU hardware, combined with the unified architecture of tensor cores, has enabled the training of much larger neural networks. In 2022, the largest neural network was Google's PaLM with 540 billion learned parameters (network weights) [43] (the older GPT-3 language model has over 175 billion learned parameters that produces human-like text; size isn't everything, Stanford's much smaller 2023 Alpaca model claims to be better, [44] having learned from Meta/Facebook's 2023 model LLaMA , the smaller 7 billion parameter variant). The widely popular chatbot ChatGPT is built on top of GPT-3.5 (and after an update GPT-4) using supervised and reinforcement learning.

References