Deep Learning: Motivation

The curse of dimensionality

The curse of dimensionality is a term introduced by Richard Bellman to describe the rapid increase in volume as more dimensions are added to a mathematical space.

Leo Breiman describes exemplarily that 100 observations cover well the one-dimensional space of real numbers between 0 and 1. From these observations a histogram can be calculated and conclusions can be drawn. If comparatively in a 10-dimensional space of the same kind (each dimension can take values between 0 and 1) 100 samples are collected, these are isolated points which do not cover the space sufficiently to make meaningful statements about this space. To achieve a similar coverage as in one-dimensional space, $100^{10} = 10^{20}$ samples must be taken, which results in a much higher effort.

An often quoted formulation of the "curse" says that for different kinds of random distributions of the data sets, the difference between the smallest and the largest distance between data sets becomes arbitrarily small compared to the smallest distance if the dimensionality d increases (in other words, the smallest and largest distance differ only relatively little), [1] and therefore the results of distance functions (and algorithms based on them) are no longer useful for the distributions in question in high-dimensional spaces. This can be formalized as

$$\lim_{d\to\infty} \frac{dist_{max} - dist_{min}}{dist_{min}} = 0$$

However, current research results indicate that it is not the pure number of dimensions that is decisive, [2] as additional relevant information can better separate the data. Only additional dimensions that are "irrelevant" to the separation cause the effect. While the exact distance values become more similar, the resulting order then remains stable. Cluster analysis and outlier detection is still possible with suitable methods [3].

Another way to characterize the "curse" is to compare $V_{Sphere} = \frac{2r^d\pi^{d/2}}{d\,\Gamma(d/2)}$, where Γ describes the gamma-function. The Volume of the hyper-cube is defined by $V_{Cube} = (2r)^d$. If we now ignore the quotient, it is noticeable that the volume of the hypersphere becomes very small ("insignificant") compared to the volume of the hypercube with increasing dimension, because

$$\frac{V_{\text{sphere}}}{V_{\text{cube}}} = \frac{\frac{2r^d\pi^{d/2}}{d\,\Gamma(d/2)}}{(2r)^d} = \frac{\pi^{d/2}}{d2^{d-1}\Gamma(d/2)} \to 0$$

for $d \to \infty$.

This convergence can be shown by the estimation:

$$\frac{\pi^{d/2}}{d2^{d-1}\Gamma(d/2)} < \frac{(2^2)^{d/2}}{d2^{d-1}\Gamma(d/2)} = \frac{2^d}{d2^{d-1}\Gamma(d/2)} = \frac{2}{d\Gamma(d/2)} \to 0$$

for $d \to \infty$, where $\pi = 3.14... < 4 = 2^2$ and $\Gamma(x) > 0$ for $x > 0$.

The curse of dimensionality is a serious hurdle in machine learning problems that have to learn the structure of a high-dimensional space from a few sample elements.

Convolution of functions

In functional analysis, a branch of mathematics, convolution (from Latin convolvere "to roll up") describes a mathematical operator that returns for two function f and g a third function $f * g$.

Intuitively, the convolution $f*g$ means that each value of f is replaced by the weighted average of the values surrounding it. More precisely, for the mean value $f * g$ the function value $f(\tau)$ is weighted with $g(x-\tau)$. The resulting "superposition" between f and mirrored and shifted versions of g (this is also called a "smearing" of f) can be used, for example, to form a moving average.

Definition

convolution for functions on $\mathbb{R}^n :\to \mathbb{C}$ is defined by:

$$(f * g)(x) := \int_{\mathbb{R}^n} f(\tau)g(x-\tau)\mathrm{d}\tau$$

.

In order to keep the definition as general as possible, one does not limit the space of the admissible functions at first, and instead demands that the integral for almost all values of x is well defined.

For periodic functions f and g of a real variable with period $T > 0$ the convolution is defined as

$$(f * g)(t) = \frac{1}{T}\int_a^{a+t} f(\tau)g(t-\tau)\mathrm{d}\tau,$$

where the integration extends over any interval with period length T. It is $f * g$ again a periodic function with period T.

In the case of a limited definition range $\mathbb{D}$ as it is the case for convolutional networks, one can continue f and g to the entire space to perform the convolution. There are several approaches to this depending on the application.

"Continuation through zero": One continues the functions by definition outside the definition range by the null function: $f\Big|_{\mathbb{R}^n\setminus\mathbb{D}} \equiv 0$.

"Periodic continuation": One continues the functions outside the definition range (e.g. mirror or copy) and uses the convolution defined for periodic functions.

Algebraic properties

The convolution defines a product on the linear space of integrable functions.

- **Commutativity**: $f * g = g * f$
- **Associativity**: $f * (g * h) = (f * g) * h$
- **Distributivity**:$f * (g + h) = (f * g) + (f * h)$
- **Associativity with scalar multiplication**: $a(f * g) = (af) * g$ for any real (or complex) number a.
- **Derivative**: $\mathrm{D}(f * g) = (\mathrm{D}f) * g = f * \mathrm{D}g$

  Here $\mathrm{D}f$ is the distributional derivative of f. If f is (totally) differentiable, the distributional derivative and the (total) derivative are identical. Two interesting examples are:

  - $\mathrm{D}(f * \delta)(x) = (f * D\delta)(x) = Df(x)$, where $D\delta$ is the derivative of delta distribution. The derivative can thus be understood as a convolution operator.
  - $(f * \Theta)(x) = \int_{-\infty}^{x} f(t)\mathrm{d}t¿$, where Θ is the jump function, yields a stem function for f.

Some features of convolution are similar to cross-correlation: for real-valued functions, of a continuous or discrete variable, it differs from cross-correlation $(f * g)$ only in that either $f(x)$ or $g(x)$ is reflected about the y-axis; thus it is a cross-correlation of $f(x)$ and $g(-x)$, or $f(-x)$ and $g(x)$.

Why do I need this?

In convolutional neural networks, convolutions can be implemented through weight sharing, i.e., the weights of N spacial neighbours are identical as illustrated in the extremely simple example in Figure 1. These shared weights can be interpreted as weights of a filter function. The input tensor is convoluted with this filter function (c.p. sliding window filter) before the activation of a convolutional layer.

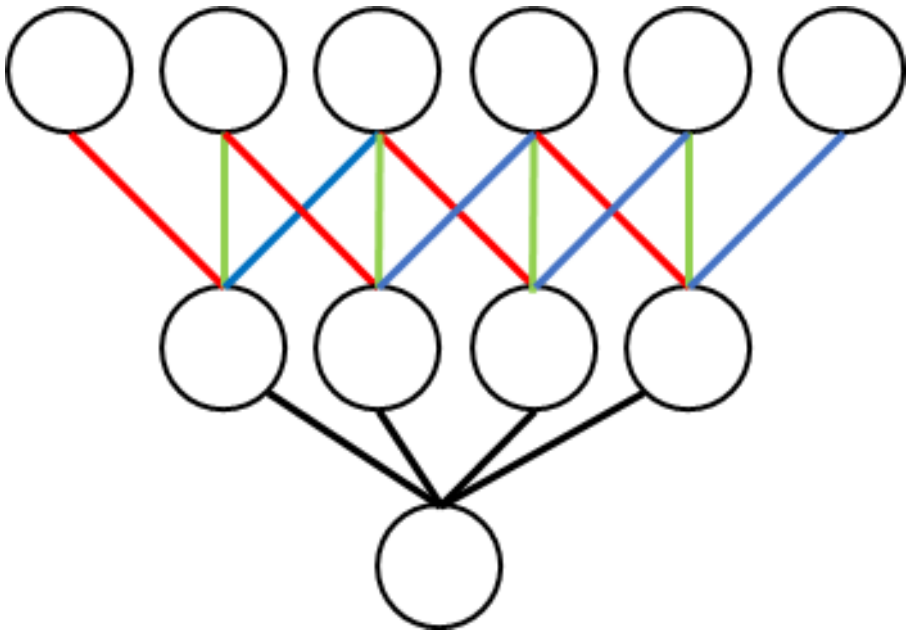

Figure 1: The weights of this network are sparse and spatially shared. red, green and blue are defining identical values.

References

[1] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In *International conference on database theory*, pages 217–235. Springer, 1999.

[2] Michael E Houle, Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Can shared-neighbor distances defeat the curse of dimensionality? In *International conference on scientific and statistical database management*, pages 482–500. Springer, 2010.

[3] Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(5):363–387, 2012.