The Functions of Deep Learning

By Gilbert Strang

Suppose we draw one of the digits 0, 1, ..., 9. How does a human recognize which digit it is? That neuroscience question is not answered here. How can a computer recognize which digit it is? This is a machine learning question. Probably both answers begin with the same idea: learn from examples.

So we start with $M$ different images (the training set). An image is a set of $p$ small pixels — or a vector $v=(v_1,\ldots,v_p)$. The component $v_i$ tells us the "grayscale" of the $i$th pixel in the image: how dark or light it is. We now have $M$ images, each with $p$ features: $M$ vectors $v$ in $p$-dimensional space. For every $v$ in that training set, we know the digit it represents.

In a way, we know a function. We have $M$ inputs in $\mathbb{R}^p$, each with an output from 0 to 9. But we don't have a "rule." We are helpless with a new input. Machine learning proposes to create a rule that succeeds on (most of) the training images. But "succeed" means much more than that: the rule should give the correct digit for a much larger set of images (the test set). An image is a set of $M$ pixels - or a vector $v=(v_1,\ldots,v_M)$, each with an output from 0 to 9.

Linear finite elements start with a triangular mesh. But specifying many individual nodes in $\mathbb{R}^p$ is expensive. It will be better if those nodes are the intersections of a smaller number of lines (or hyperplanes). Please note that a regular grid is too simple.

Figure 1 is a first construction of a piecewise linear function of the data vector $v$. Choose a matrix $A$ and vector $b$. Then set to 0 (this is the nonlinear step) all negative components of $Av+b$. Then multiply by a matrix $C$ to produce the output $w=F(v)=C(Av+b)_+$. That vector $(Av+b)_+$ forms a "hidden layer" between the input $v$ and the output $w$.

The nonlinear function called ReLU $x_+=\max(x,0)$ was originally smoothed into a logistic curve like $1/(1+e^{-x})$. It was reasonable to think that continuous derivatives would help in optimizing the weights $A, b, C$. That proved to be wrong.

The graph of each component of $(Av+b)_+$ has two half-planes (one is flat, from the 0s where $Av+b$ is negative). If $A$ is $q$ by $p$, the input space $\mathbb{R}^p$ is sliced by $q$ hyperplanes into $r$ pieces. We can count those pieces! This measures the "expressivity" of the overall function $F(v)$.

The formula from combinatorics is

$$r(q,p) = \binom{q}{0} + \binom{q}{1} + \cdots + \binom{q}{p}.$$ 

This number gives an impression of the graph of $F$. But our function is not yet sufficiently expressive, and one more idea is needed.

Here is the indispensable ingredient in the learning function $F$. The best way to create complex functions from simple functions is by composition. Each $F_i$ is a linear (or affine) followed by the nonlinear ReLU $x_+=\max(x,0)$, which is convenient-closed under addition and maximization and composition. The magic is that the learning function $F(A, b, v)$ gives accurate results on images $v$ that $F$ has never seen before.

This article is published with very light edits.

Gilbert Strang teaches linear algebra at the Massachusetts Institute of Technology. A description of the January 2019 textbook Linear Algebra and Learning from Data is available at math.mit.edu/learningfromdata.