Deep Learning course

Edgar **Francisco** Roman-Rangel
edgar.roman@alumni.epfl.ch

Session 3 – Background

CInC-UAEM. Cuernavaca, Mexico. September 8th, 2018.

Outline

Branch of mathematics that deals with vectors and their transformation from one vector space to another vector space.

In deep learning we deal with multidimensional data and their manipulation.

Scalar

A quantity that has only magnitude (and no direction), e.g., height or weight, x or y.

Vector

An array of scalars: $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$,

e.g., house pricing, where,
$x_1 =$ area of the house,
$x_2 =$ number of bedrooms,
$x_3 =$ number of bathrooms, and
$x_4 =$ population density of the locality.

Vectors define a vector space.

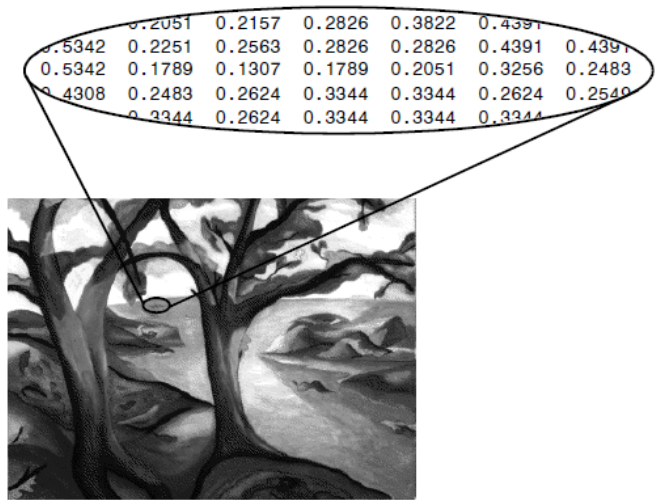
Matrix

A two dimensional array of scalars, arranged in rows and columns.

$$\mathbf{X} = \begin{bmatrix} x_{1,1}, & x_{1,2}, & \dots, & x_{1,m} \\ x_{2,1}, & x_{2,2}, & \dots, & x_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1}, & x_{n,2}, & \dots, & x_{n,m} \end{bmatrix}$$

$(n \times m)$ with n rows and m columns.

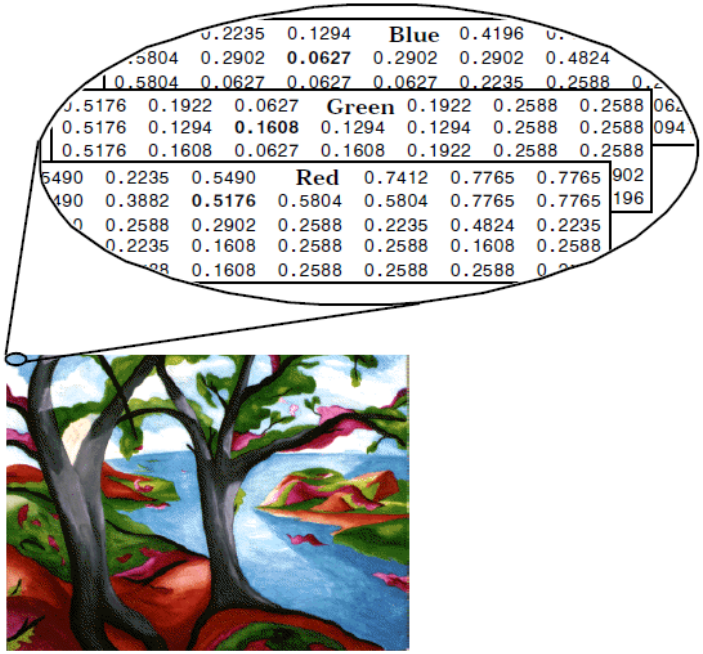
e.g., gray-scale image.

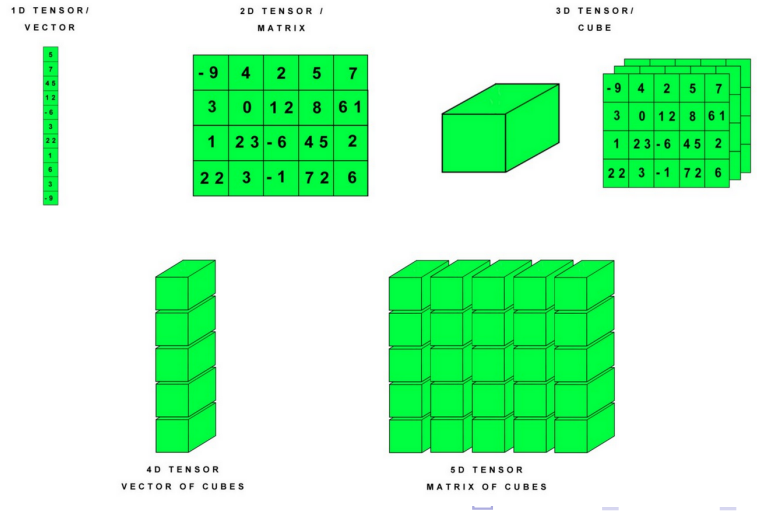


Tensor

A multidimensional array of scalars. e.g.,

- Color images (3D tensor).
- Collection of gray-scale images (3D Tensor).
- Collection of color images (4D Tensor).
- Video (4D or higher-order Tensor).
- 3D Video (5D Tensor).





The deep learning way

Tensors store and process data, e.g.,

Multi-variate data	2D	$[example \times feature]$.
Sequences	2D	$[sequence \times sample]$.
Sequences	3D	$[sequence \times sample \times feature]$.
Gray-scale images	3D	$[height \times width \times example]$.
Color images	4D	$[height \times width \times channel \times example]$.

Transpose

$$(\mathbf{A}^T)_{i,j} = \mathbf{A}_{j,i}$$

$$\mathbf{A} = \begin{bmatrix} a_{1,1}, & a_{1,2} \\ a_{2,1}, & a_{2,2} \\ a_{3,1}, & a_{3,2} \end{bmatrix} \implies \mathbf{A}^T = \begin{bmatrix} a_{1,1}, & a_{2,1}, & a_{3,1} \\ a_{1,2}, & a_{2,2}, & a_{3,2} \end{bmatrix}$$

Addition and Multiplication with scalars

$$b + \mathbf{A} = \begin{bmatrix} b + a_{1,1}, & b + a_{1,2} \\ b + a_{2,1}, & b + a_{2,2} \\ b + a_{3,1}, & b + a_{3,2} \end{bmatrix} \qquad b \cdot \mathbf{A} = \begin{bmatrix} b \cdot a_{1,1}, & b \cdot a_{1,2} \\ b \cdot a_{2,1}, & b \cdot a_{2,2} \\ b \cdot a_{3,1}, & b \cdot a_{3,2} \end{bmatrix}$$

Matrix addition

$$\mathbf{C} = \mathbf{A} + \mathbf{B}$$

$$C_{ij} = A_{ij} + B_{ij}$$

$\mathbf{A}$ and $\mathbf{B}$ must have the same shape (except **broadcasting**).

Matrix product

$$\mathbf{C} = \mathbf{AB}$$

$$C_{i,j} = \sum_k A_{i,k} B_{k,j}$$

Rule: multiply $[n \times m][m \times p] = [n \times p]$, i.e., $\mathbf{A}$ must have the same number of columns as $\mathbf{B}$ has rows.

– Different from $\mathbf{A} \odot \mathbf{B}$.

Questions

- What is the size of the result of the product of two vectors?

$$\mathbf{v} \in \mathbb{R}^{[1 \times n]} \cdot \mathbf{u} \in \mathbb{R}^{[n \times 1]}$$

Questions

- What is the size of the result of the product of two vectors?

$$\mathbf{v} \in \mathbb{R}^{[1\times n]} \cdot \mathbf{u} \in \mathbb{R}^{[n\times 1]}$$

A: a scalar.

Questions

- What is the size of the result of the product of two vectors?

$$\mathbf{v} \in \mathbb{R}^{[1\times n]} \cdot \mathbf{u} \in \mathbb{R}^{[n\times 1]}$$

  A: a scalar.

- What is the size of the result of multiplying?

$$\mathbf{A} \in \mathbb{R}^{[m\times n]} \cdot \mathbf{x} \in \mathbb{R}^{[n\times 1]}$$

Questions

- What is the size of the result of the product of two vectors?

$$\mathbf{v} \in \mathbb{R}^{[1\times n]} \cdot \mathbf{u} \in \mathbb{R}^{[n\times 1]}$$

A: a scalar.

- What is the size of the result of multiplying?

$$\mathbf{A} \in \mathbb{R}^{[m\times n]} \cdot \mathbf{x} \in \mathbb{R}^{[n\times 1]}$$

A: a vector $\in \mathbb{R}^{[m\times 1]}$.

Properties

Distributive:

$$\mathbf{A}(\mathbf{B}+\mathbf{C}) = \mathbf{AB}+\mathbf{AC}$$

Associative (observe product constrain):

$$\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}$$

Multiplication is not commutative:

$$\mathbf{AB} \neq \mathbf{BA}$$

Simplification of matrix product:

$$(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$$

Identity matrix

Matrix whose product does not change any vector $\mathbf{x}$.

$$\mathbf{I}_n \in \mathbb{R}^{[n \times n]}$$

Identity matrix of order n (square with n rows and n columns).

$$\forall \mathbf{x}, \mathbf{I}_n \mathbf{x} = \mathbf{x}$$

"All its values along the main diagonal are 1, and 0 elsewhere".

Matrix inverse

The matrix inverse of $\mathbf{A}$, denoted $\mathbf{A}^{-1}$, is the matrix that yields,

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$$

If we apply a linear transformation to the space with $\mathbf{A}$, it is possible to go back with $\mathbf{A}^{-1}$.

System of equations

$$\mathbf{Ax} = \mathbf{b}$$

where $\mathbf{A}$ and $\mathbf{b}$ are known, and we want to solve for $\mathbf{x}$.

System of equations

$$\mathbf{Ax} = \mathbf{b}$$

where $\mathbf{A}$ and $\mathbf{b}$ are known, and we want to solve for $\mathbf{x}$.

$$\mathbf{A}^{-1}\mathbf{Ax} = \mathbf{A}^{-1}\mathbf{b}$$

System of equations

$$\mathbf{Ax} = \mathbf{b}$$

where $\mathbf{A}$ and $\mathbf{b}$ are known, and we want to solve for $\mathbf{x}$.

$$\mathbf{A}^{-1}\mathbf{Ax} = \mathbf{A}^{-1}\mathbf{b}$$

$$\mathbf{Ix} = \mathbf{A}^{-1}\mathbf{b}$$

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

s.t. we are able to compute $\mathbf{A}^{-1}$ (square and with all its column being linearly independent).

System of equations

$$\mathbf{Ax} = \mathbf{b}$$

where $\mathbf{A}$ and $\mathbf{b}$ are known, and we want to solve for $\mathbf{x}$.

$$\mathbf{A}^{-1}\mathbf{Ax} = \mathbf{A}^{-1}\mathbf{b}$$

$$\mathbf{Ix} = \mathbf{A}^{-1}\mathbf{b}$$

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

s.t. we are able to compute $\mathbf{A}^{-1}$ (square and with all its column being linearly independent).

Often we use a pseudo-inverse approximation.

Orthogonal vectors

Two vectors $\mathbf{x}$ and $\mathbf{y}$ are orthogonal to each other if $\mathbf{x}^T\mathbf{y} = 0$, i.e., they are at 90 degrees to each other.

Orthonormal vectors

Besides being orthogonal, they have unit norm:

$$\|\mathbf{x}\|_2 = 1$$

where, $\|\cdot\|$ denotes Euclidean norm.

Orthogonal matrix

Square matrix whose rows are mutually orthonormal and whose columns are mutually orthonormal:

$$\mathbf{A}^T\mathbf{A} = \mathbf{A}\mathbf{A}^T = \mathbf{I}$$

which implies,

$$\mathbf{A}^{-1} = \mathbf{A}^T$$

Symmetric matrix

$$\mathbf{A} = \mathbf{A}^T$$

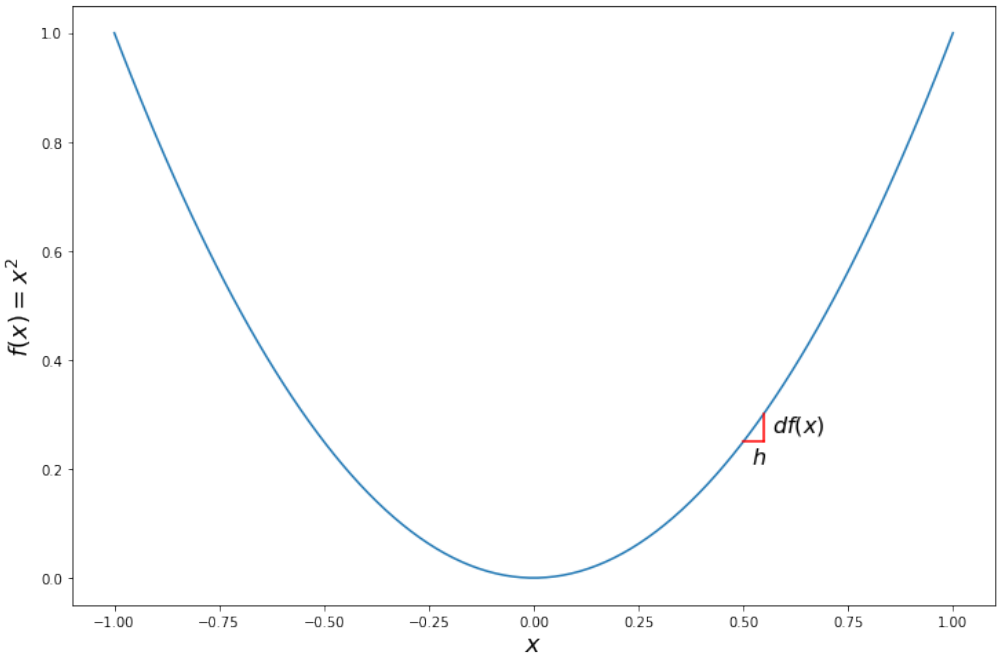
Do you wanna know more?

- Goodfellow's, Deep Learning Book.
- https://hadrienj.github.io/posts/Deep-Learning-Book-Series-2.1-Scalars-Vectors-Matrices-and-Tensors/
- https://www.youtube.com/watch?v=fNk_zzaMoSs&list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab

Outline

Derivative

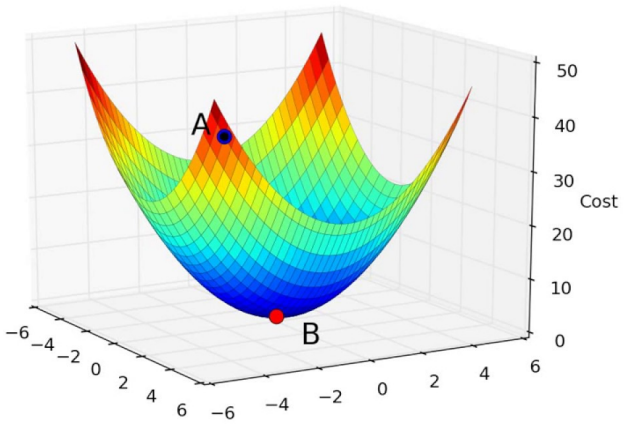
$$\frac{df}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$



Partial derivative

For a function with two variables,

$$z = f(x, y)$$



Partial derivative of z with respect to x:

$$\frac{\partial z}{\partial x} = \lim_{h \to 0} \frac{f(x+h, y) - f(x, y)}{h}$$

Similarly,

$$\frac{\partial z}{\partial y} = \lim_{h \to 0} \frac{f(x, y+h) - f(x, y)}{h}$$

Gradient

The vector of partial derivatives.

$$\nabla(z) = \left[\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}\right]^T$$

Chain rule

"If a variable z depends on the variable y, which itself depends on the variable x, so that y and z are therefore dependent variables, then z, via the intermediate variable of y, depends on x as well". [Wikipedia]

$$\frac{dz}{dx} = \frac{dz}{dy} \cdot \frac{dy}{dx}$$

Outline

Approximations and inference

In Deep Learning, we want to solve

$$y = f(\mathbf{x}; \omega)$$

but often, we can only estimate an approximation $\hat{y}$,

$$\hat{y} = f(\mathbf{x}; \omega)$$

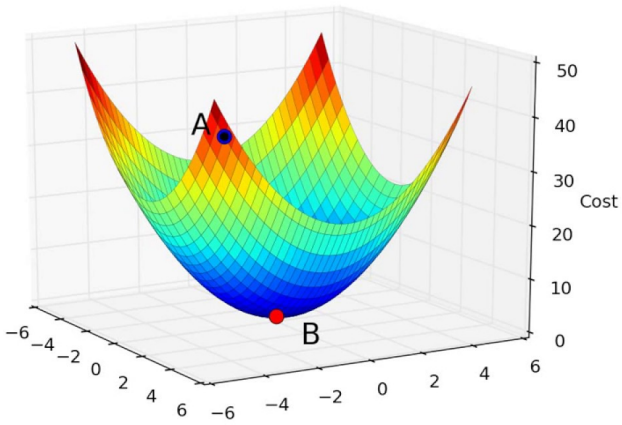
So, we define a cost function $E(\hat{y}, \hat{y})$ to measure our performance, e.g., *mean squared error* (mse),

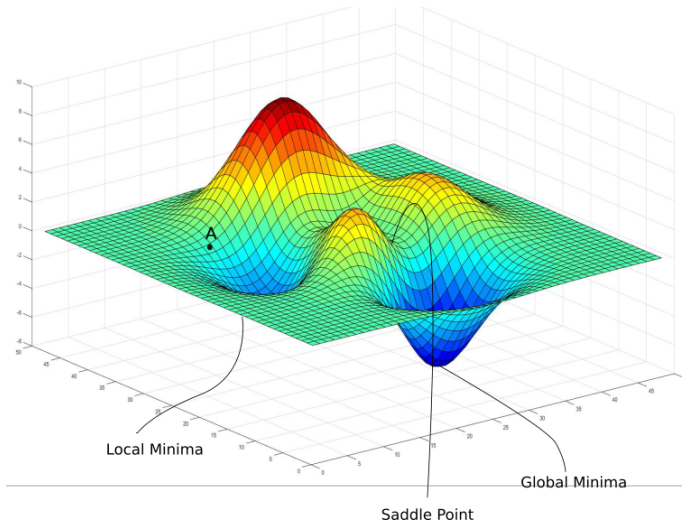
$$E = \|\hat{y} - y\|_2^2$$

which we wish to minimize.

Minimization

- Can/should we reach $E = 0$?
- Linear regression has closed-form solution $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$.
- Most complex problems have no closed-form solution.
- Iterative approaches reach fairly good approximations.
- Risk of getting trapped in local minima.

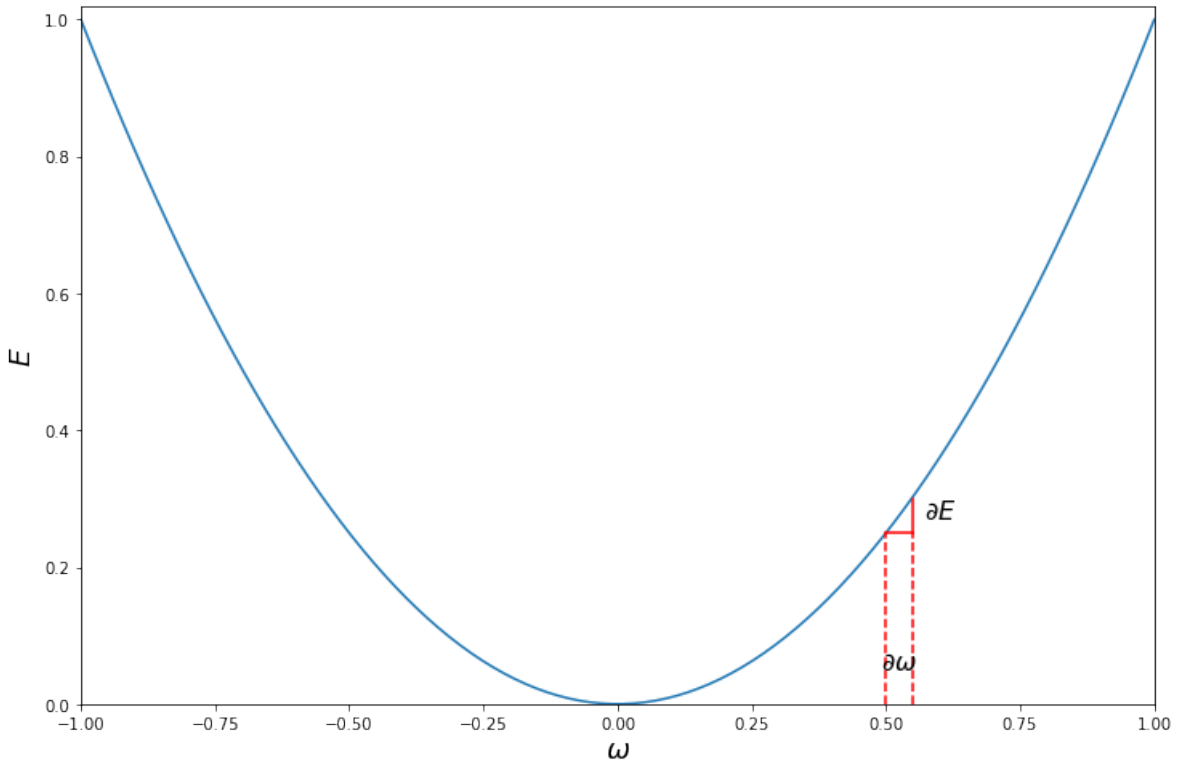




Approximation by:

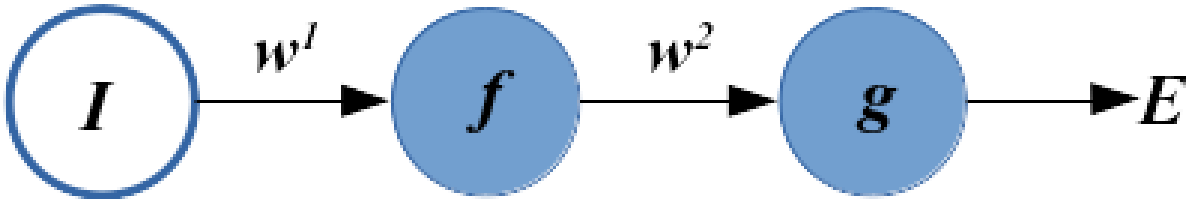
Estimating the gradient of the cost (E) with respect to the model parameters (ω)

$$\frac{\partial E}{\partial \omega}$$



Propagating by:

Chain rule for complex functions (Deep Neural Networks).



$$\frac{\partial E}{\partial \omega^1} = \frac{\partial E}{\partial g} \cdot \frac{\partial g}{\partial \omega^2} \cdot \frac{\partial \omega^2}{\partial f} \cdot \frac{\partial f}{\partial \omega^1}$$

Thank you.

Q&A