Computer Vision

Class 11: The Shallow Basics of Deep Neural Networks

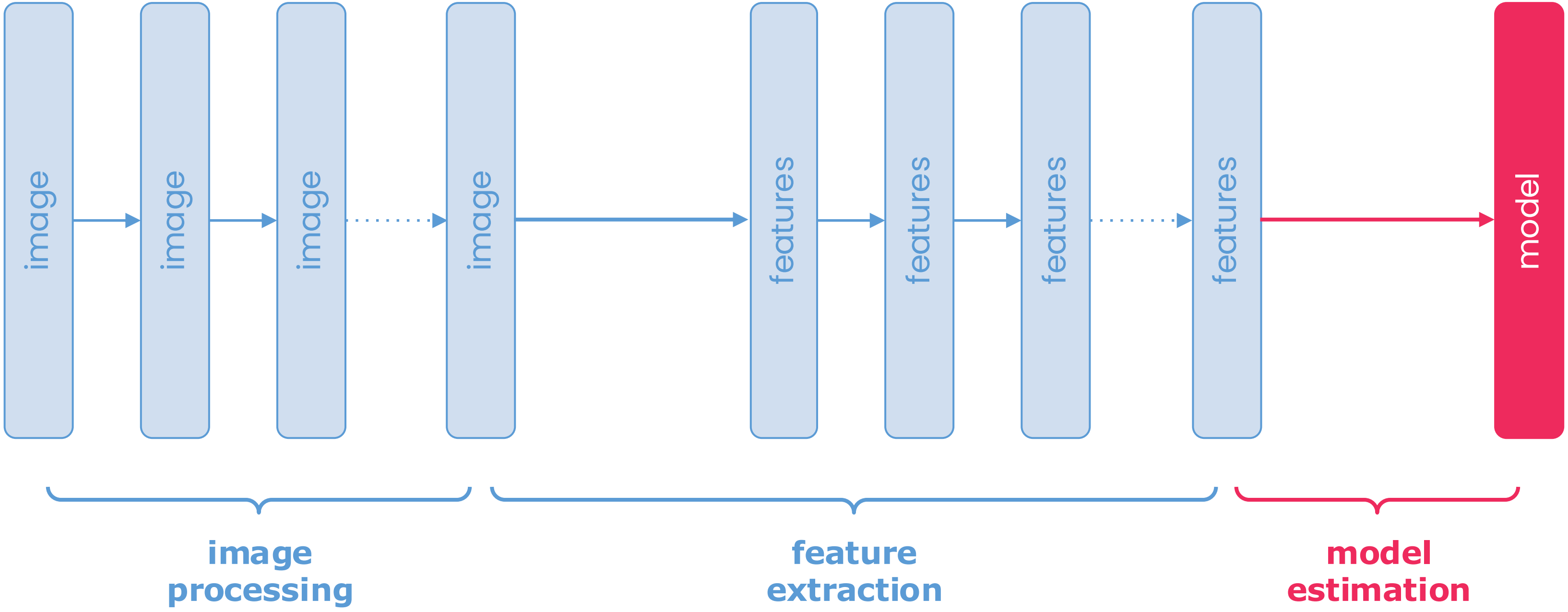
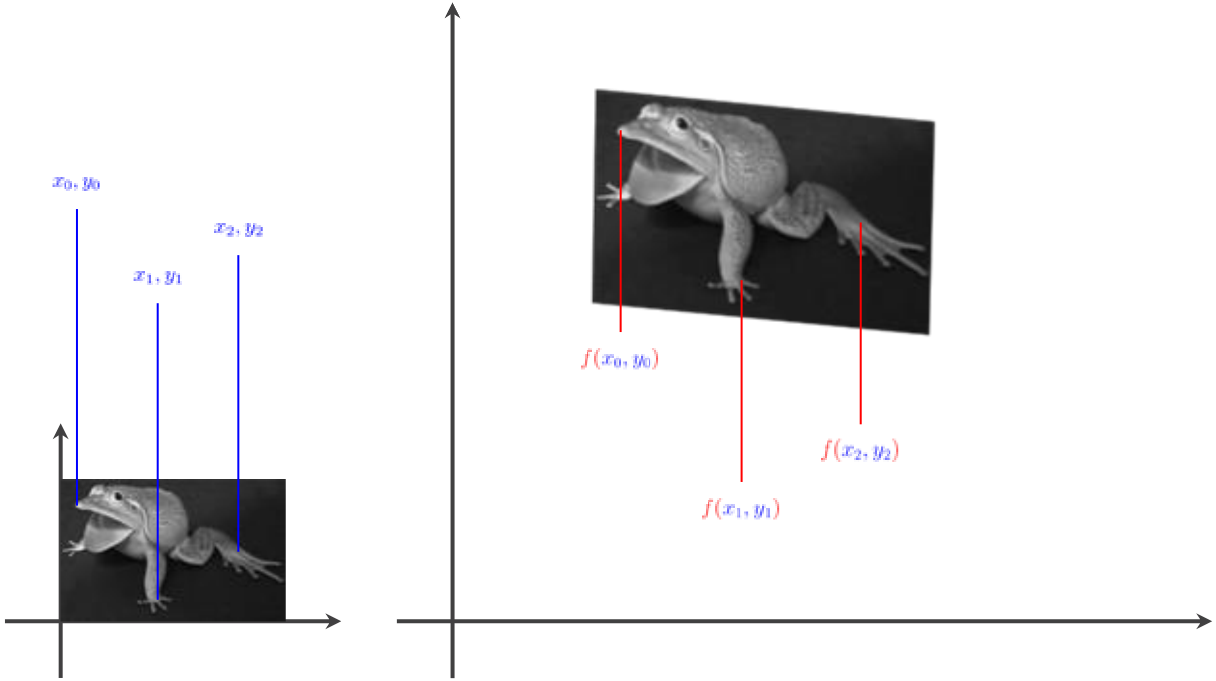


image
image
image
image
features
features
features
features
model
image processing
feature extraction
model estimation

The object position model

a geometric transformation that maps object coordinates to scene coordinates

$f : \mathbb{R}^2 \to \mathbb{R}^2$



A matrix multiplication can...

- ...scale.
- ...rotate.
- ...translate.
- ...warp.

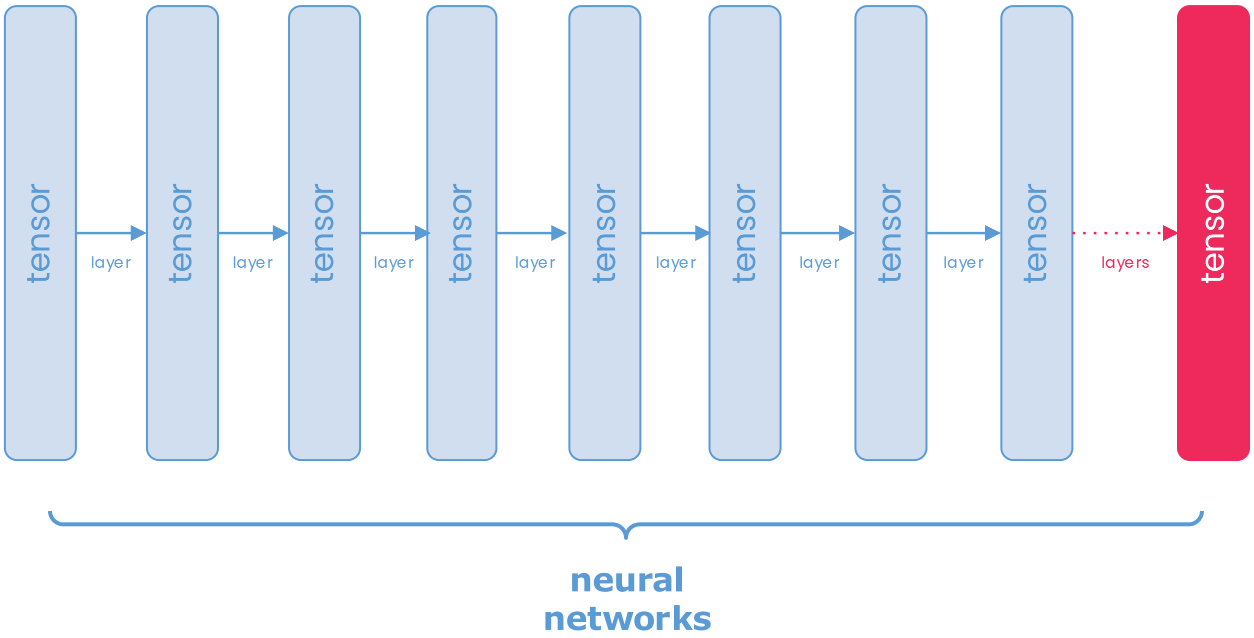
$$A_1 \cdot A_2 \cdots A_k \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

A matrix multiplication can...

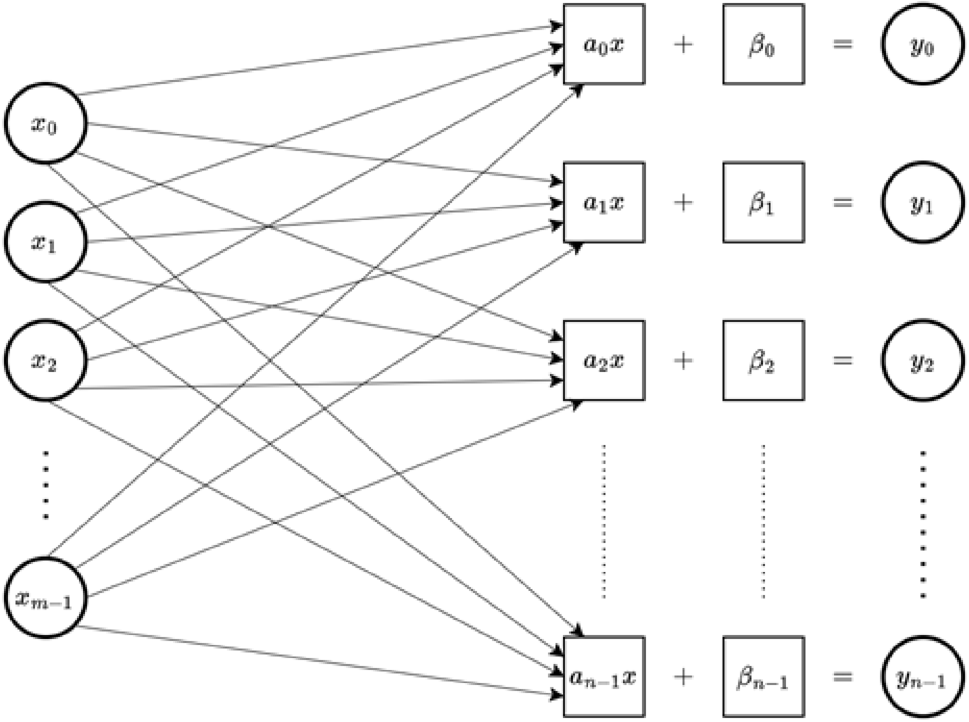
- ...scale.
- ...rotate.
- ...translate.
- ...warp.
- ...perform any combination of scales, rotations, translations, and warps, and in any order.

$$\overbrace{A_1 \cdot A_2 \cdots A_k}^{\text{can merge into a single matrix}} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

So, what else in computer vision can be modeled by matrix multiplication?

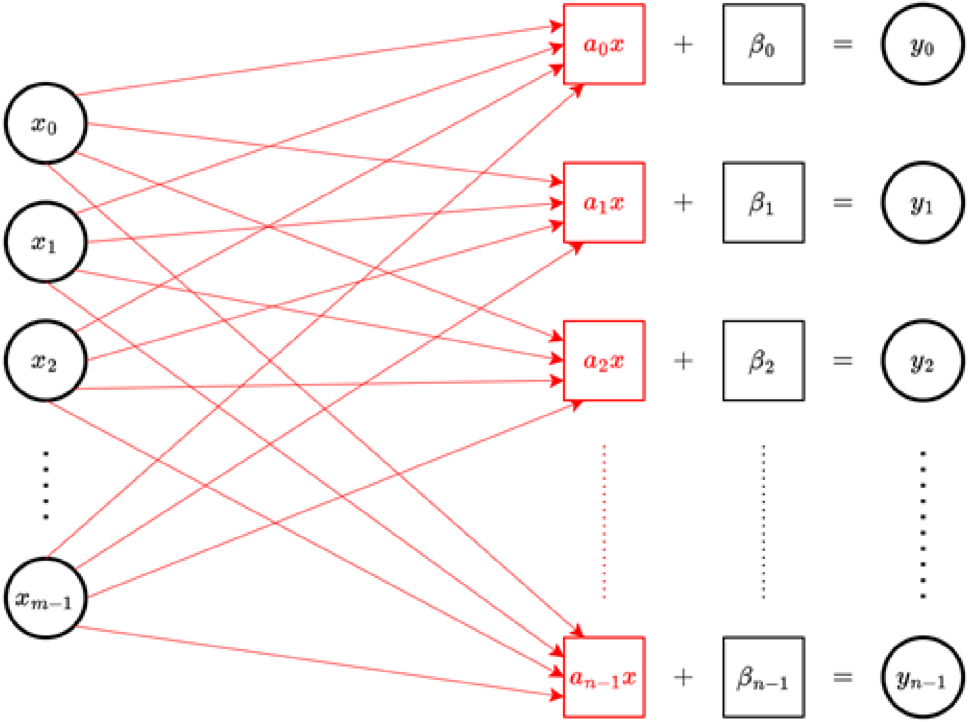

tensor
layer
tensor
layer
tensor
layer
tensor
layer
tensor
layer
tensor
layer
tensor
layer
tensor
layers
tensor
neural
networks

Dense layer:
multiplicative weights and additive biases



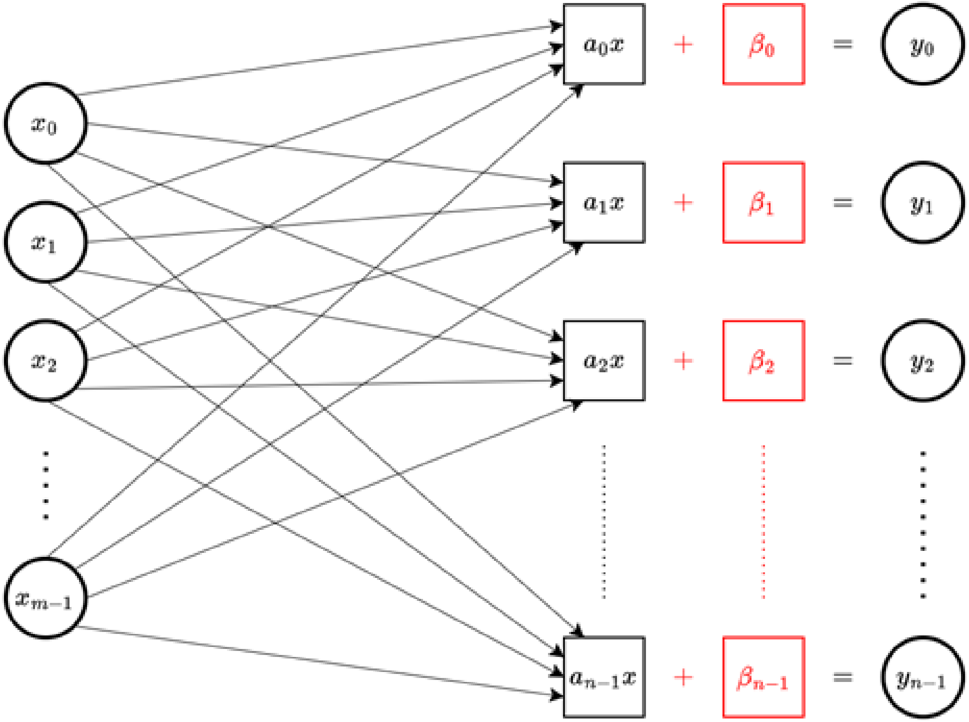
Dense layer:

multiplicative weights and additive biases

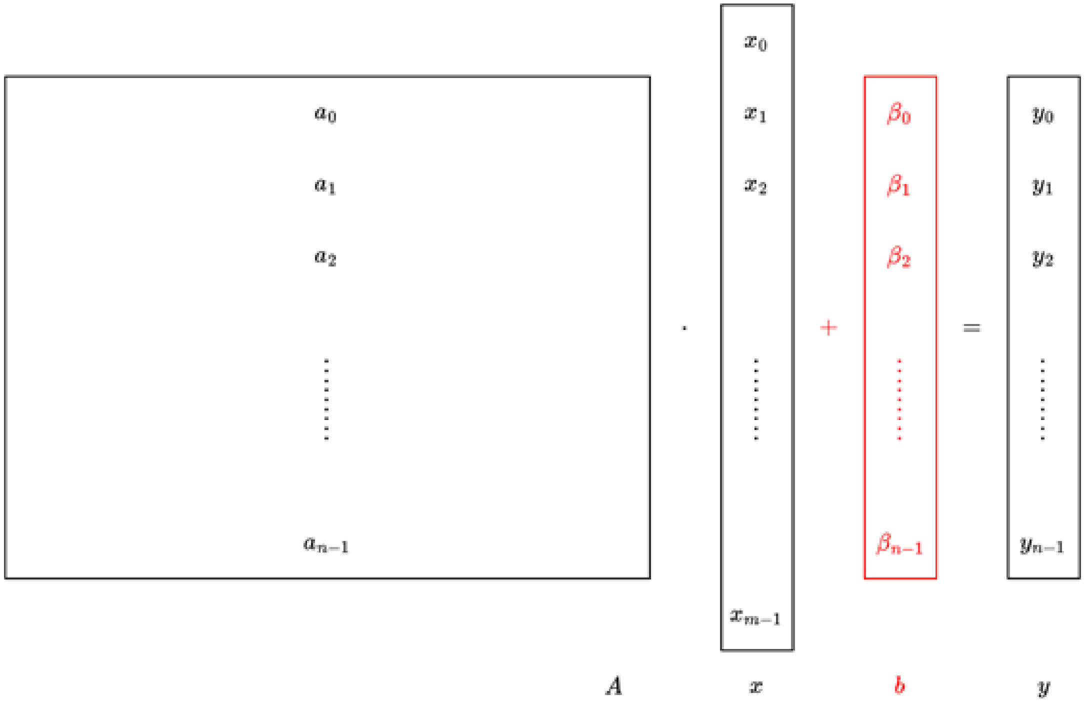


Dense layer:

multiplicative weights and additive biases



Dense layer as affine transformation:
matrix multiplication and array addition



Dense layer as affine transformation:

matrix multiplication and array addition

$$\underbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-1} \end{bmatrix}}_{A} \cdot \underbrace{\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{m-1} \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n-1} \end{bmatrix}}_{b} = \underbrace{\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{bmatrix}}_{y}$$

Dense layer as affine transformation:

matrix multiplication and array addition

$$\underset{A}{\begin{bmatrix} \alpha_{0,0} & \alpha_{0,1} & \alpha_{0,2} & \cdots \\ \alpha_{1,0} & \alpha_{1,1} & \alpha_{1,2} & \cdots \\ \alpha_{2,0} & \alpha_{2,1} & \alpha_{2,2} & \cdots \\ & \vdots & & \\ \alpha_{n,0} & \alpha_{n,1} & \alpha_{n,2} & \cdots \end{bmatrix}} \cdot \underset{x}{\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{m-1} \end{bmatrix}} + \underset{b}{\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n-1} \end{bmatrix}} = \underset{y}{\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{bmatrix}}$$

Dense layer as affine transformation:

matrix multiplication and array addition

$$
\underbrace{\begin{bmatrix}
\alpha_{0,0} & \alpha_{0,1} & \alpha_{0,2} & \cdots \\
\alpha_{1,0} & \alpha_{1,1} & \alpha_{1,2} & \cdots \\
\alpha_{2,0} & \alpha_{2,1} & \alpha_{2,2} & \cdots \\
 & \vdots & & \\
\alpha_{n,0} & \alpha_{n,1} & \alpha_{n,2} & \cdots
\end{bmatrix}}_{A}
\cdot
\underbrace{\begin{bmatrix}
x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{m-1}
\end{bmatrix}}_{x}
+
\underbrace{\begin{bmatrix}
\beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n-1}
\end{bmatrix}}_{b}
=
\underbrace{\begin{bmatrix}
y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1}
\end{bmatrix}}_{y}
$$

Since affine transformations can be modeled by matrix multiplication…

...dense layers can be seen as such.

handout

Toolkit

- **Language:** Mathematics
- **Library:** None
- **Platform:** Pencil and Paper

Instructions

1. Organize in groups of 2 or 3 members. No more, no less.

1. Take a copy of the handout, read it, and do the activities.

1. Register the member names and return it to the professor.

The word "deep" in "deep learning" refers to the number of layers through which the data is transformed. More precisely, deep learning systems have a substantial *credit assignment path* (CAP) depth. The CAP is the chain of transformations from input to output. CAPs describe potentially causal connections between input and output. For a feedforward neural network, the depth of the CAPs is that of the network and is the number of hidden layers plus one (as the output layer is also parameterized). For recurrent neural networks, in which a signal may propagate through a layer more than once, the CAP depth is potentially unlimited.[9] No universally agreed-upon threshold of depth divides shallow learning from deep learning, but most researchers agree that deep learning involves CAP depth higher than two. CAP of depth two has been shown to be a universal approximator in the sense that it can emulate any function.[10] Beyond that, more layers do not add to the function approximator ability of the network. Deep models (CAP > two) are able to extract better features than shallow models and hence, extra layers help in learning the features effectively.

https://en.wikipedia.org/wiki/Deep_learning

The word "deep" in "deep learning" refers to the number of layers through which the data is transformed. More precisely, deep learning systems have a substantial *credit assignment path* (CAP) depth. The CAP is the chain of transformations from input to output. CAPs describe potentially causal connections between input and output. For a feedforward neural network, the depth of the CAPs is that of the network and is the number of hidden layers plus one (as the output layer is also parameterized). For recurrent neural networks, in which a signal may propagate through a layer more than once, the CAP depth is potentially unlimited.[9] No universally agreed-upon threshold of depth divides shallow learning from deep learning, but most researchers agree that deep learning involves CAP depth higher than two. CAP of depth two has been shown to be a universal approximator in the sense that it can emulate any function.[10] Beyond that, more layers do not add to the function approximator ability of the network. Deep models (CAP > two) are able to extract better features than shallow models and hence, extra layers help in learning the features effectively.

https://en.wikipedia.org/wiki/Deep_learning

If all computer vision algorithms could be modeled by matrix multiplication...

$$A_1 \cdot A_2 \cdots A_k \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\overbrace{A_1 \cdot A_2 \cdots A_k}^{\text{can merge into a single matrix}} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

...what would be the point of having a neural network with many layers?

Neural network mysteries

- How can we separate a dataset in training data and testing data?
- ~~What is a hidden layer?~~
- ~~Does it matter if the input tensors are 3D, 2D, or 1D?~~
- ~~What is a dense layer?~~
- ~~How do we calculate the number of parameters in a dense layer?~~
- ~~How do we calculate the number of steps in a training process?~~
- ~~How do we calculate the number of steps in a testing process?~~
- Does the number of layers matter?
- ~~Does the size of a dense layer matter?~~
- What exactly `activation='relu'` does?
- ~~Can classic vision algorithms be represented by dense layers?~~
- ~~In a dense layer, are all the weights and biases always necessary?~~
- ~~How do we calculate the number of parameters in a convolutional layer?~~

Deep learning is not simply about adding extra layers.

Deep learning is about ensuring that the extra layers are relevant.

Next class:

- non-linear classic vision.

Credits

This material was based on the work of other professors, listed below.

- Fabio Miranda (fabiomiranda@insper.edu.br)
- Raul Ikeda (RaulIGS@insper.edu.br)
- Fabio Ayres (FabioJA@insper.edu.br)
- Igor Montagner (IgorSM1@insper.edu.br)
- Andrew Kurauchi (AndrewTNK@insper.edu.br)
- Luciano Silva (LucianoS4@insper.edu.br)
- Tiago Sanches (tiagoss4@insper.edu.br)

Well, except for the errors. Any errors you might find are probably my fault.

Images

Lowe, D. G. *Distinctive image features from scale-invariant keypoints.* International Journal of Computer Vision, 60. (2004)