INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Ceará

HUAWEI

Convolutional Neural Networks

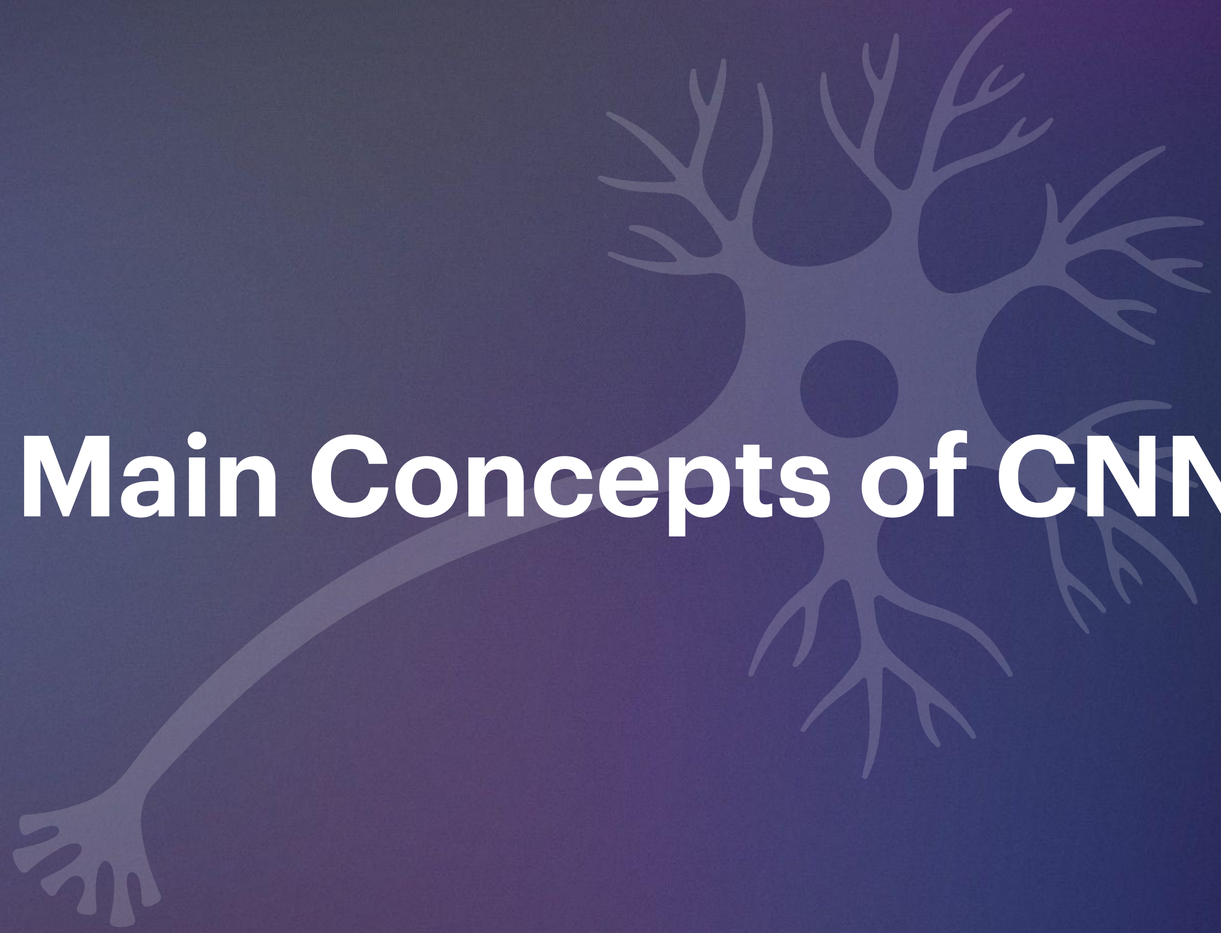
Prof. Me. Saulo A. F. Oliveira
saulo.oliveira@ifce.edu.br

Convolutional Neural Network

- A convolutional neural network (CNN) is a feedforward neural network. Its artificial neurons may respond to **surrounding units within the coverage range**. CNN excels at image processing. It includes a **convolutional layer**, a **pooling layer**, and a **fully connected layer**.
- In the 1960s, Hubel and Wiesel studied cats' cortex neurons used for local sensitivity and direction selection and found that their unique network structure could simplify feedback neural networks. They then proposed the CNN.
- Now, CNN has become one of the research hotspots in many scientific fields, especially in the pattern classification field. The network is widely used because it can avoid complex pre-processing of images and directly input original images.

01

Main Concepts of CNN



INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Ceará

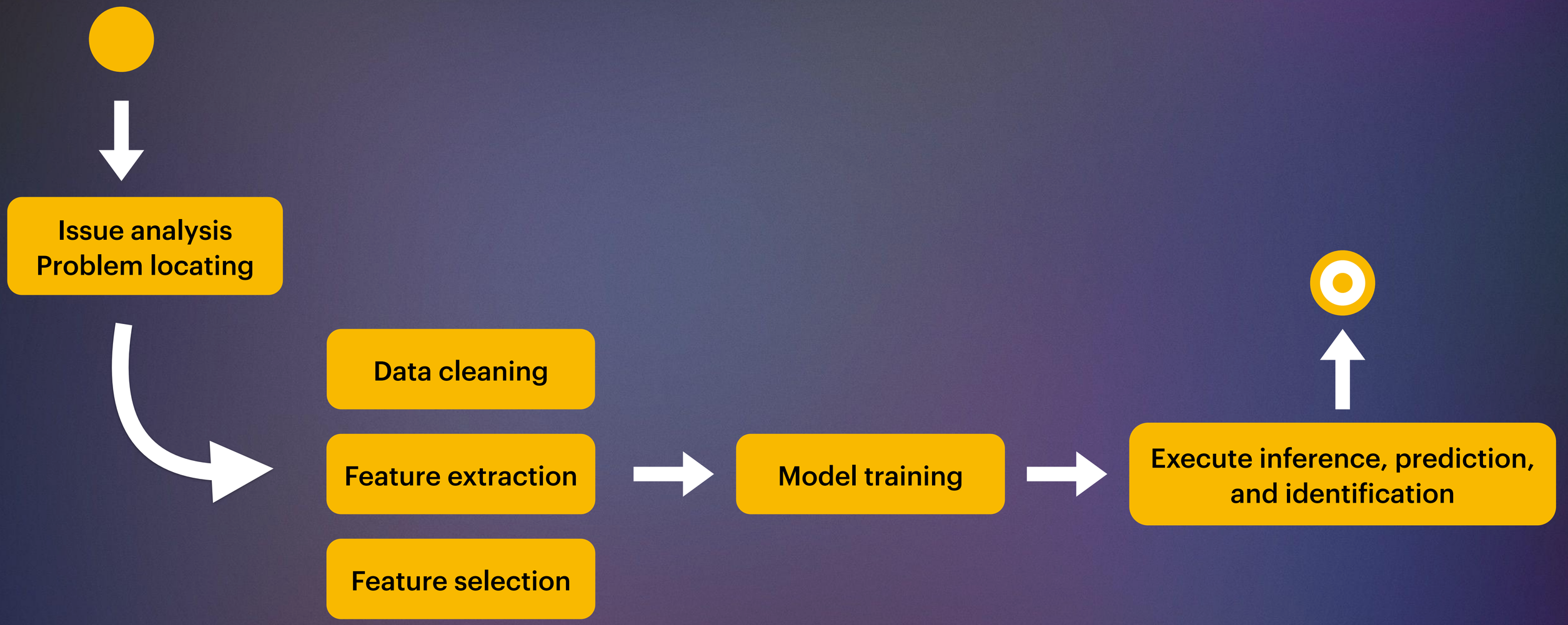
HUAWEI

Local Receptive Field and Parameter Sharing

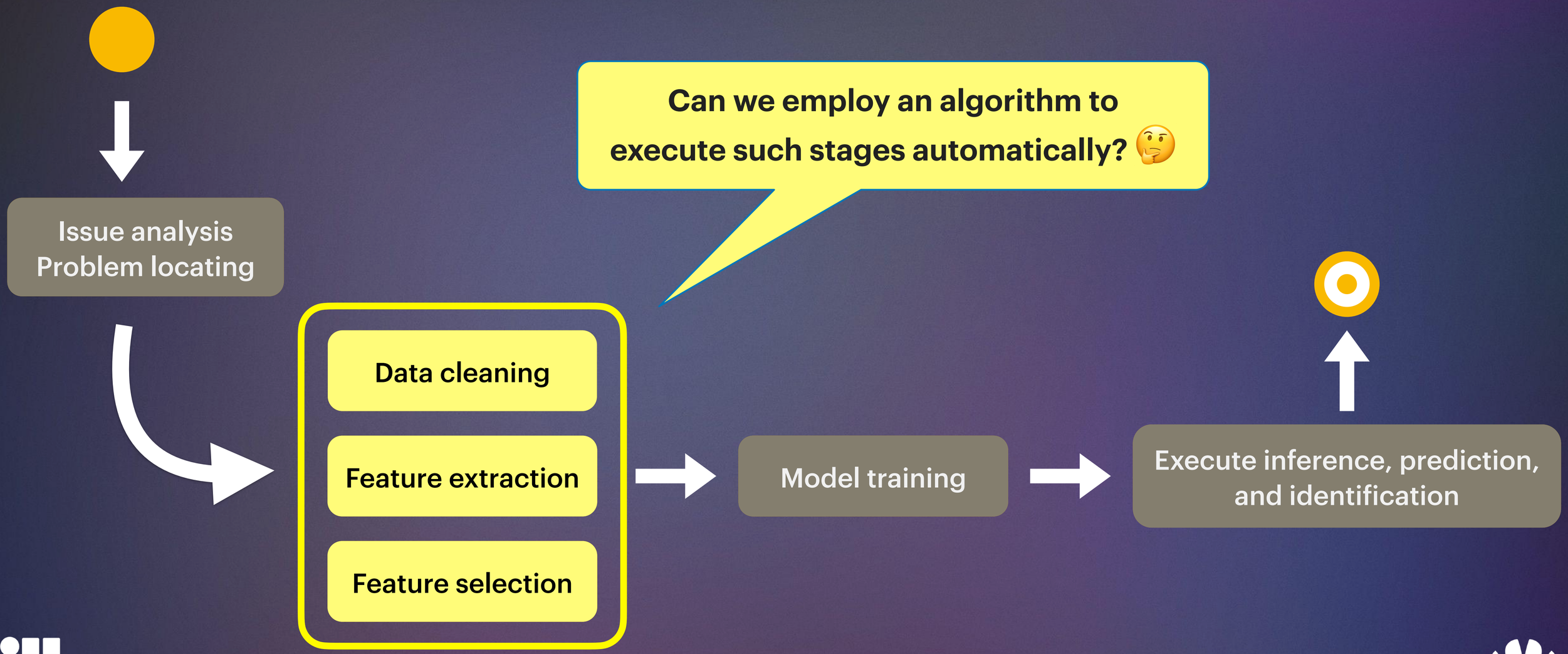
- **Local receptive field**: It is generally considered that human perception of the outside world is from local to global. **Spatial correlations among local pixels of an image are closer than those among distant pixels.** Therefore, each neuron does not need to know the global image. It only needs to know the local image. The local information is combined at a higher level to generate global information.

- **Parameter sharing**: One or more convolution cores may be used to scan input images. Parameters carried by the convolution cores are weights. In a layer scanned by convolution cores, each core uses the same parameters during weighted computation. Weight sharing means that when each convolution core scans an entire image, parameters of the convolution core are fixed.

Traditional Machine Learning vs Deep Learning



Traditional Machine Learning vs Deep Learning



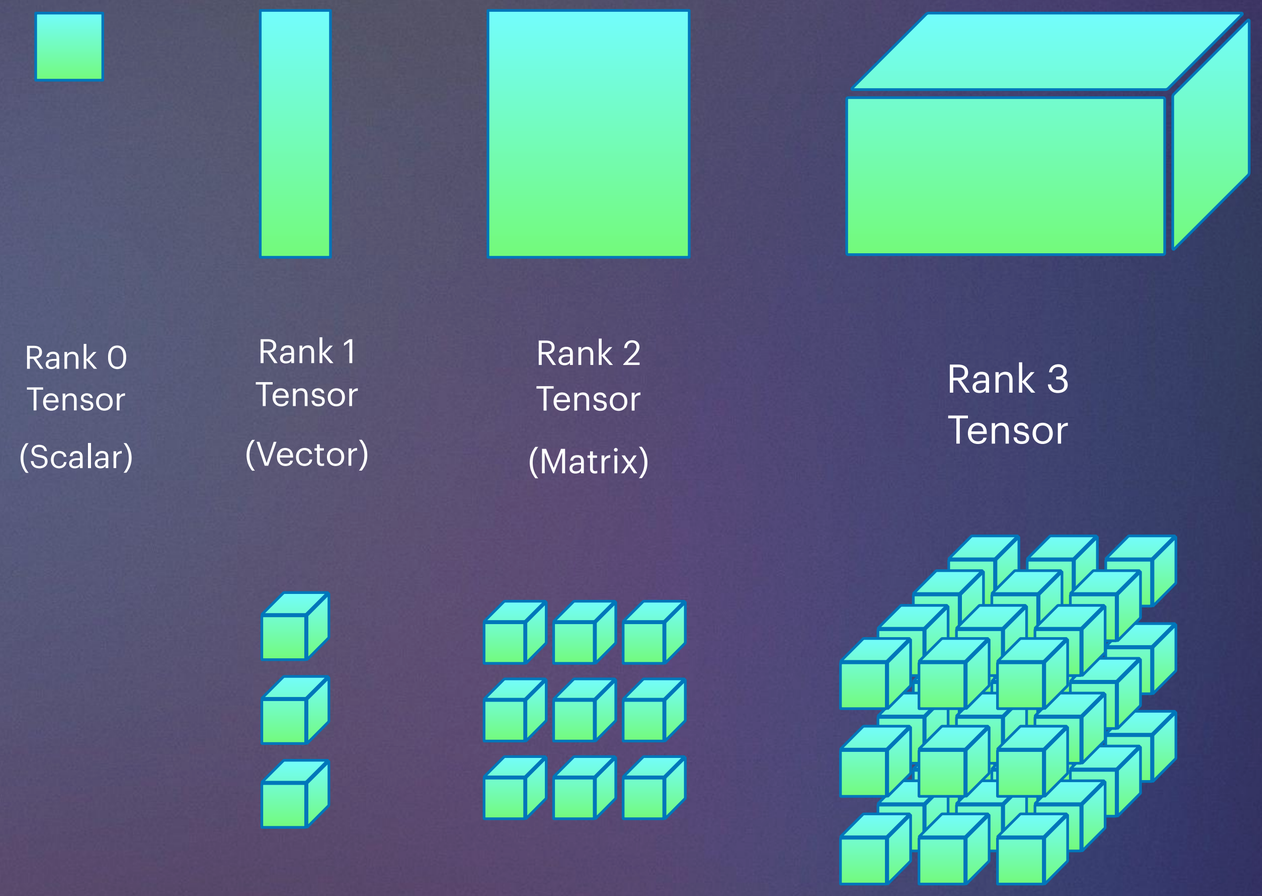
Tensors

- Tensors are the most basic data structures in Deep Learning Frameworks. All data is encapsulated in tensors.
- Tensor: a multidimensional array.

➡ A scalar is a rank-0 tensor.

➡ A vector is a rank-1 tensor.

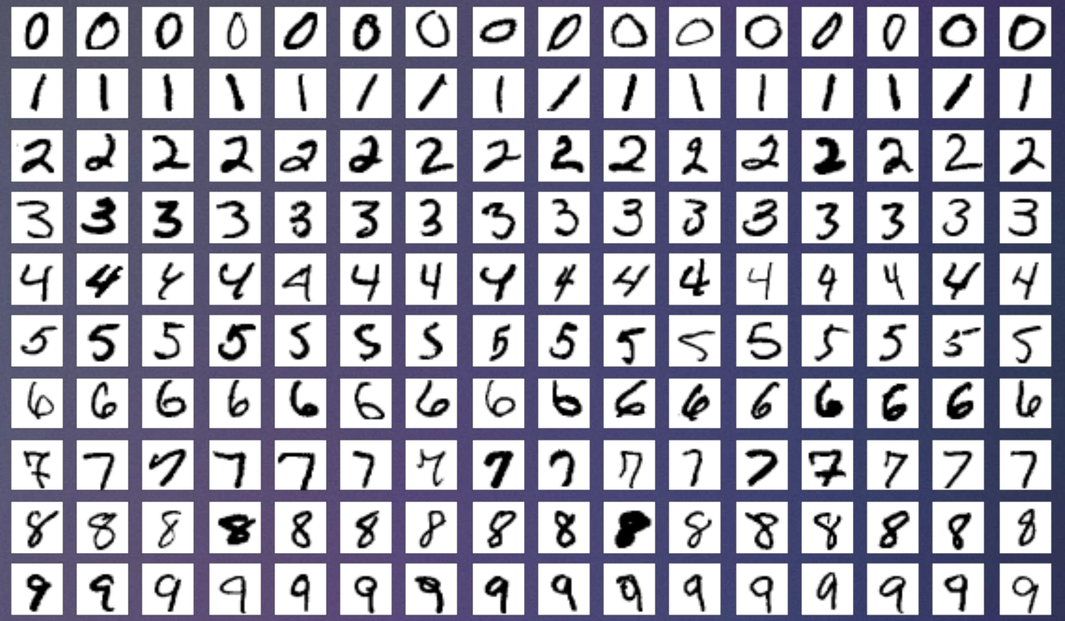
➡ A matrix is a rank-2 tensor.



MNIST dataset

- The **MNIST database** (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems.

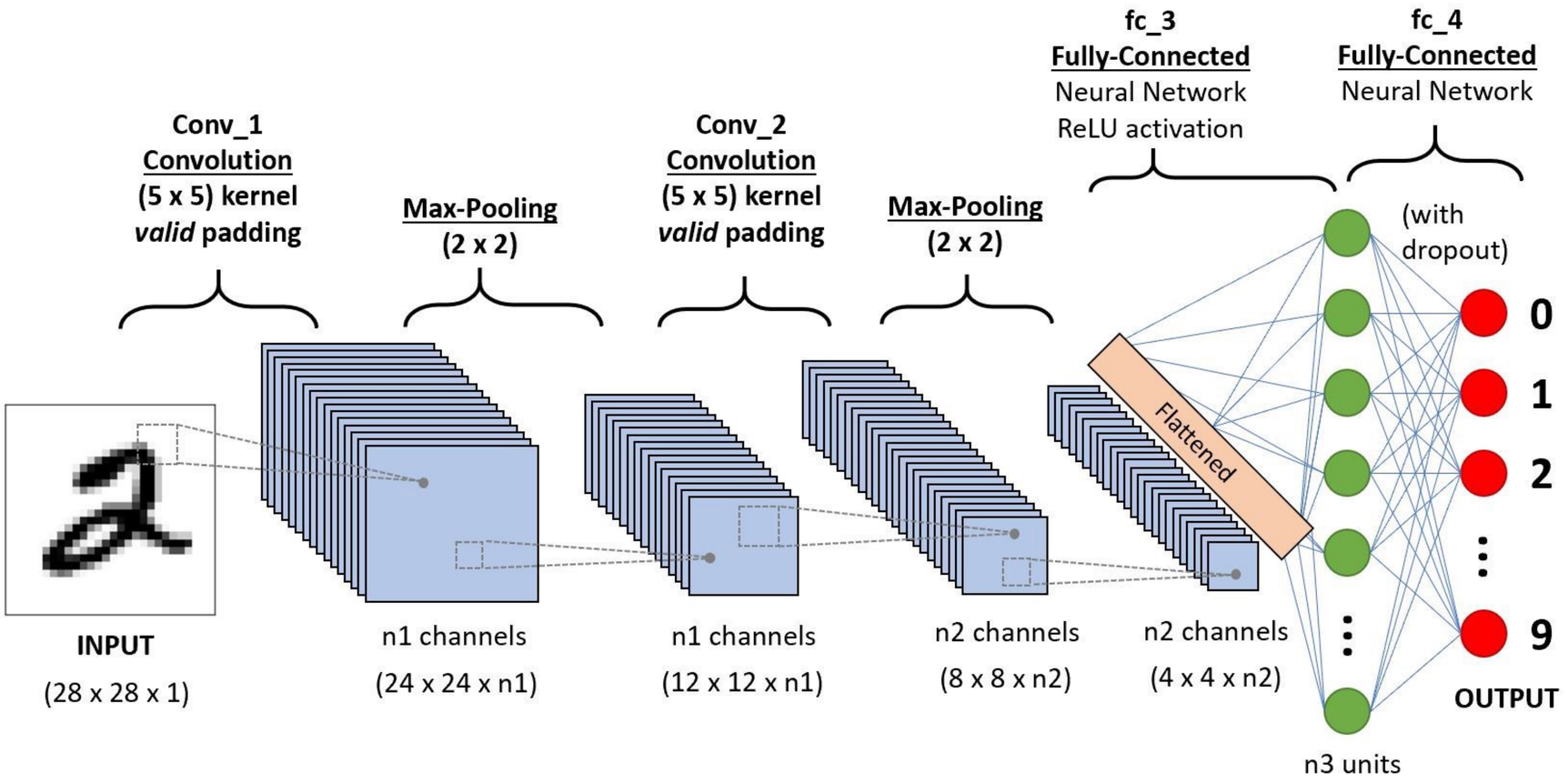
- The MNIST database contains $60{,}000$ training images and $10{,}000$ testing images.



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

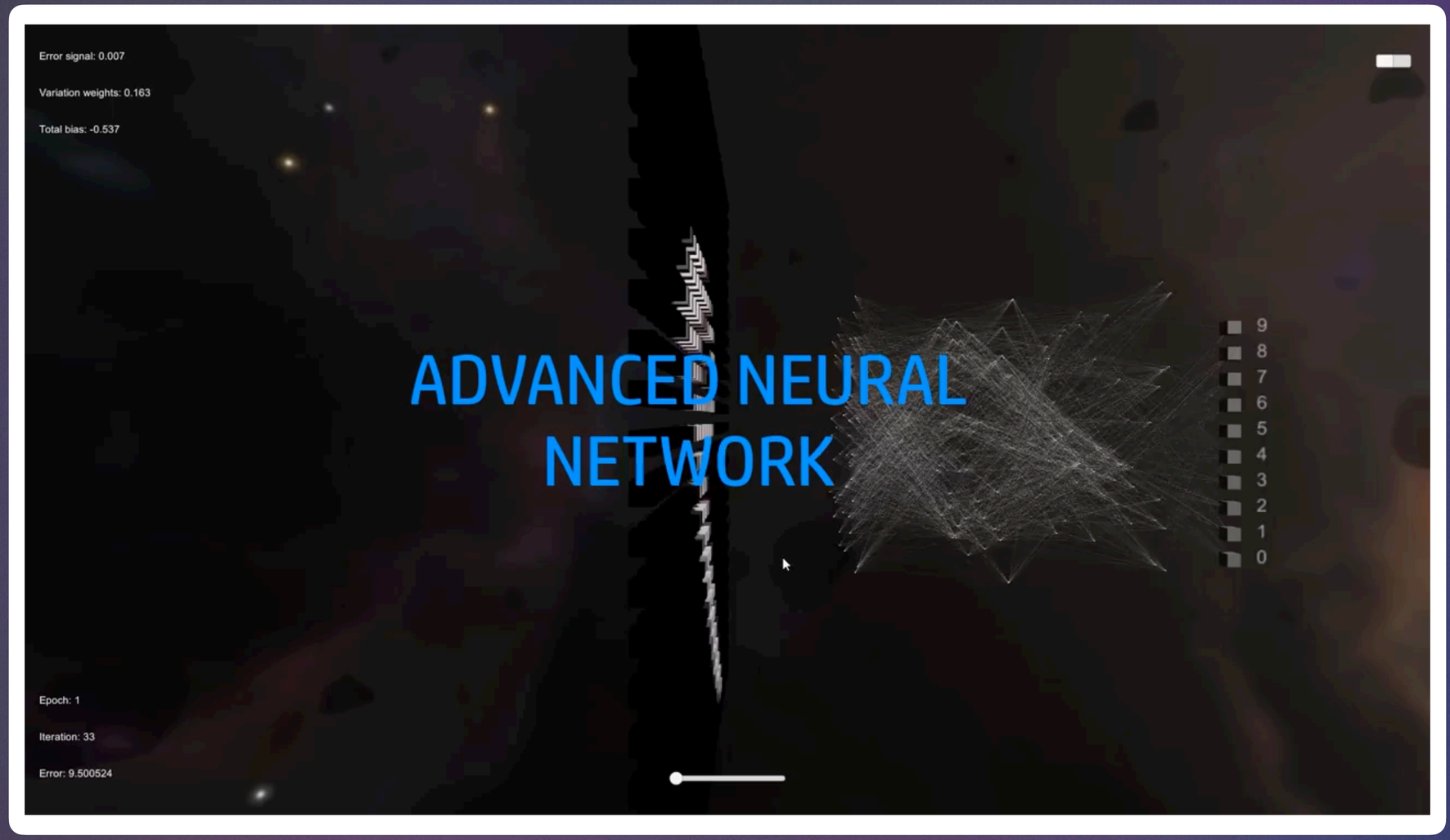
- Furthermore, the black and white images from NIST were normalized to fit into a 28×28 pixel bounding box and anti-aliased, which introduced grayscale levels.

Architecture of Convolutional Neural Network



https://aigents.co/blog/publication/introduction-to-convolutional-neural-networks-cnns

Deep Neural Network 3D Simulation

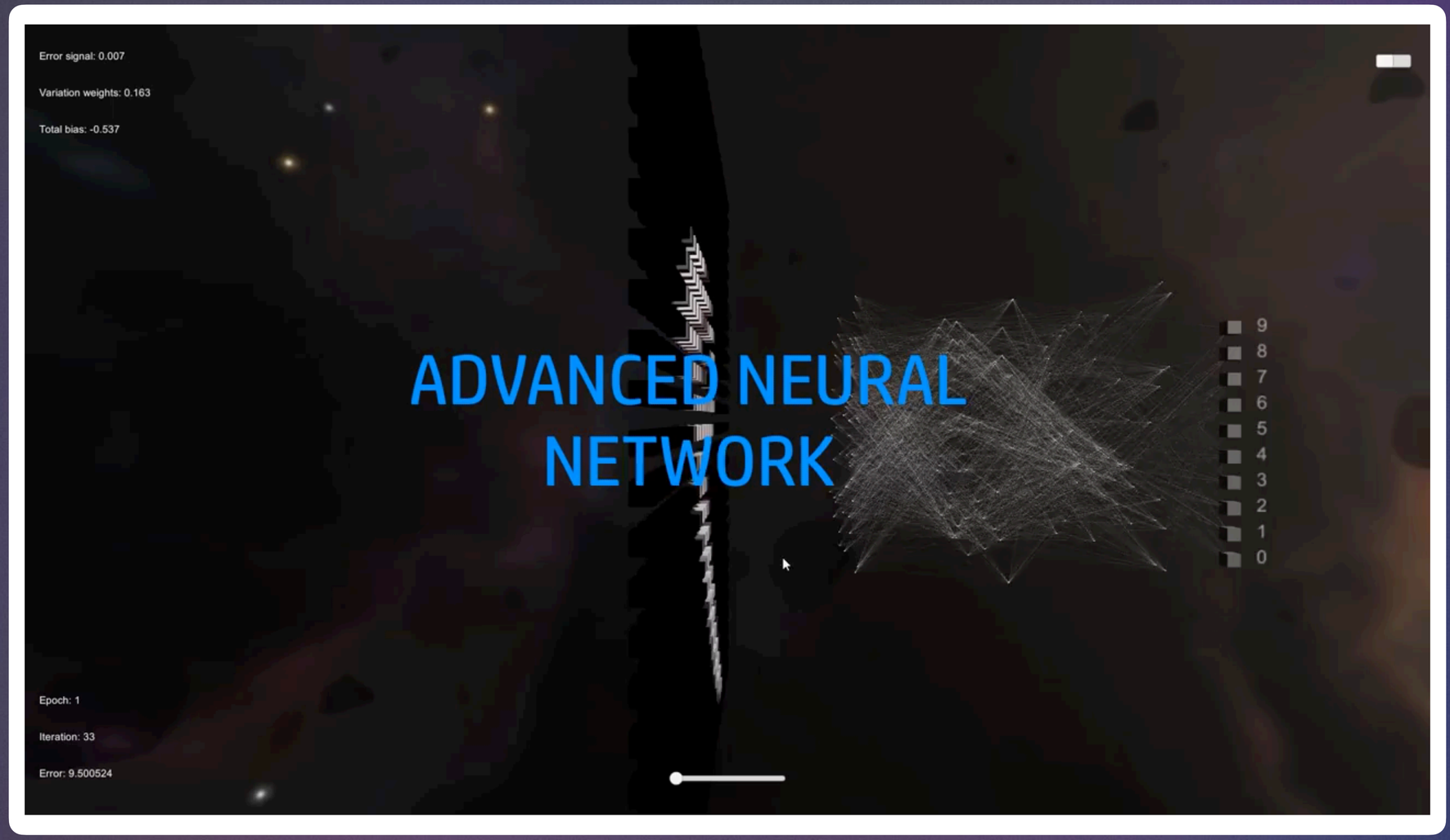


https://www.youtube.com/watch?v=2GYLpzZzqeg

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Ceará

HUAWEI

Deep Neural Network 3D Simulation



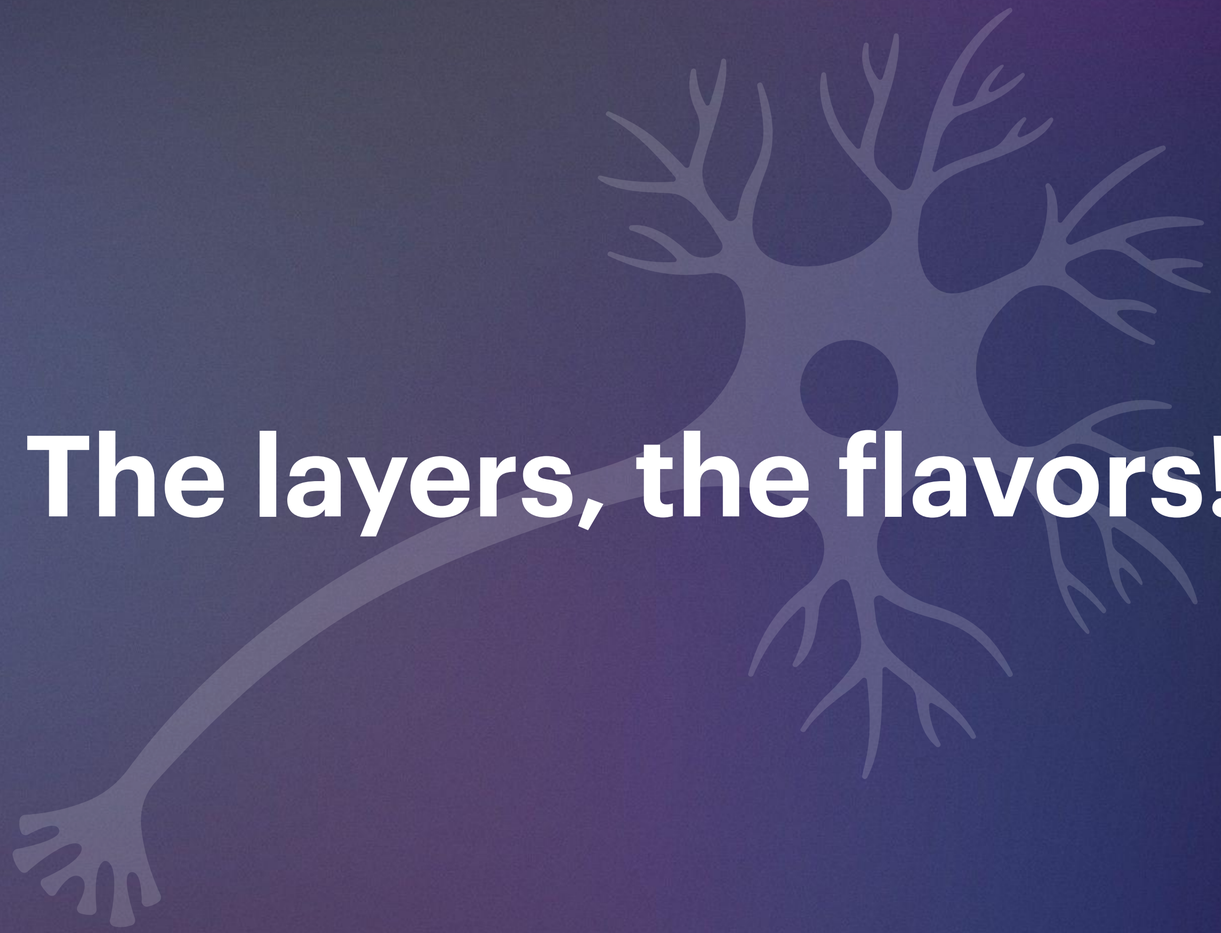
https://www.youtube.com/watch?v=2GYLpzZzqeg

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Ceará

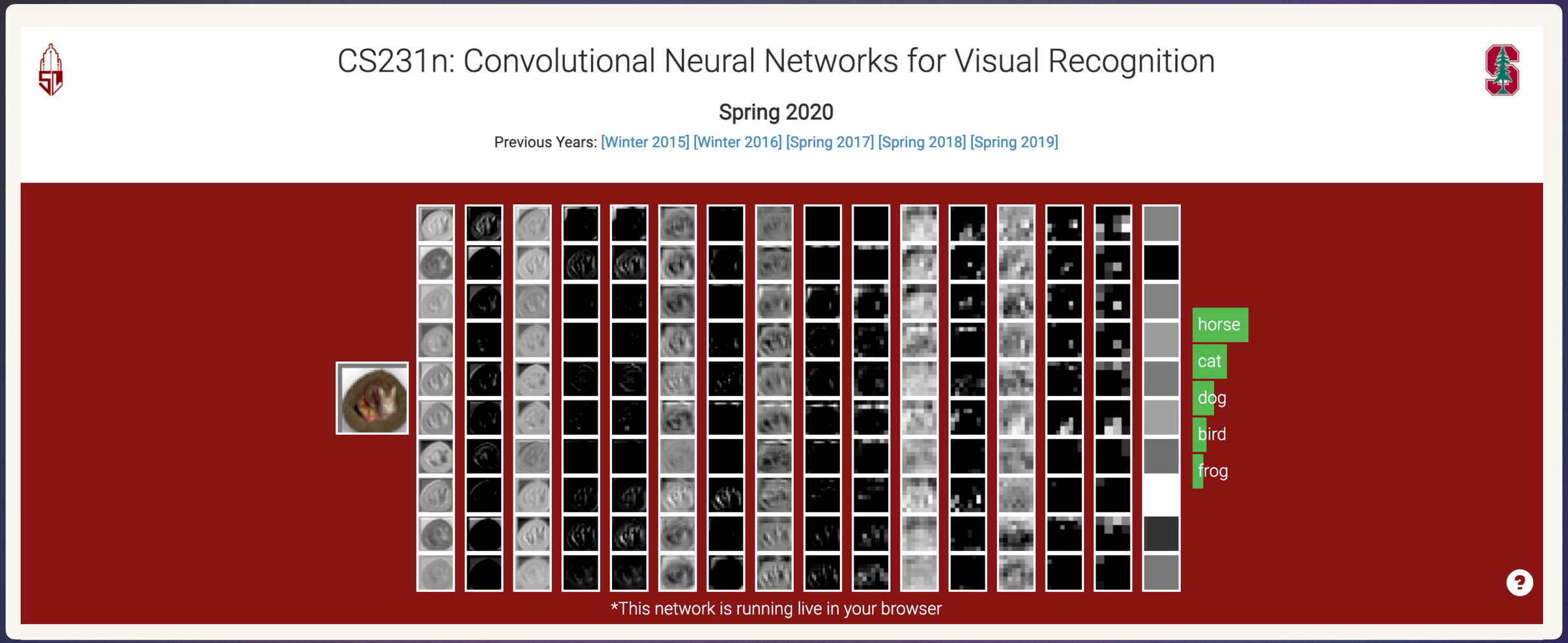
HUAWEI

02

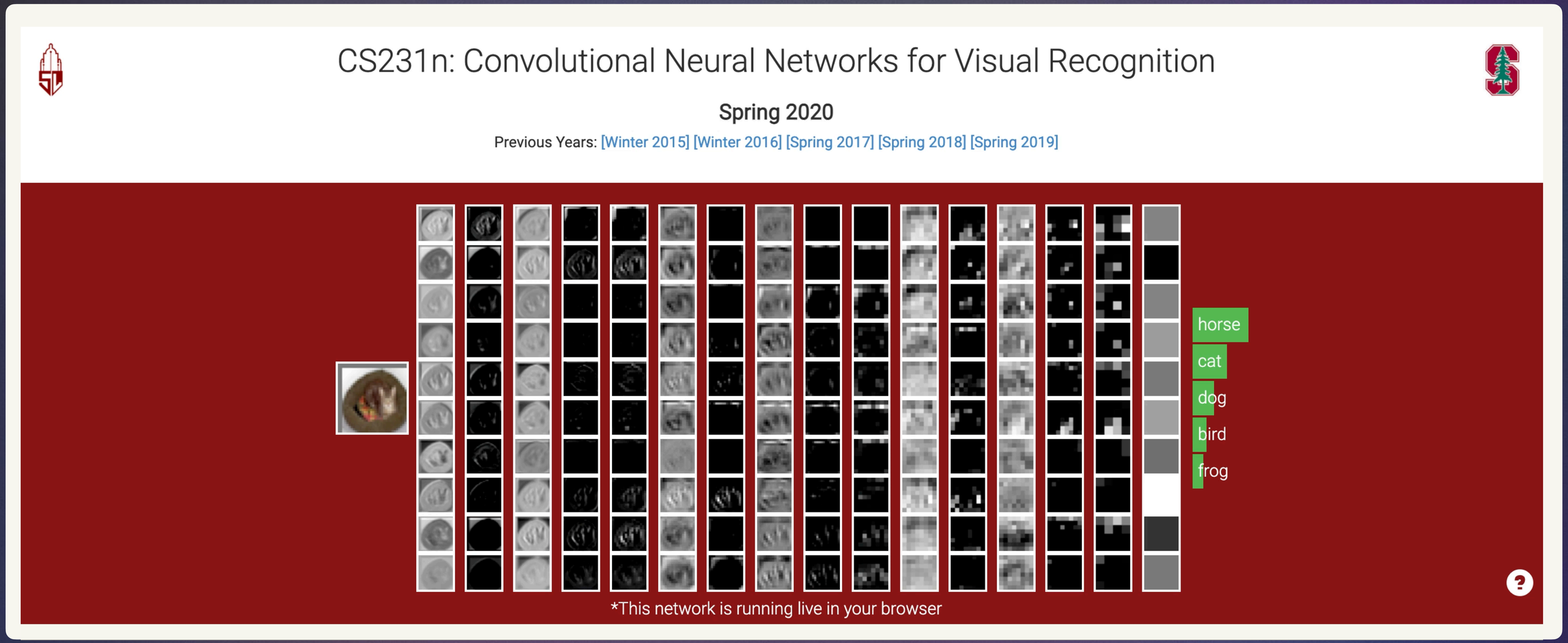
The layers, the flavors!



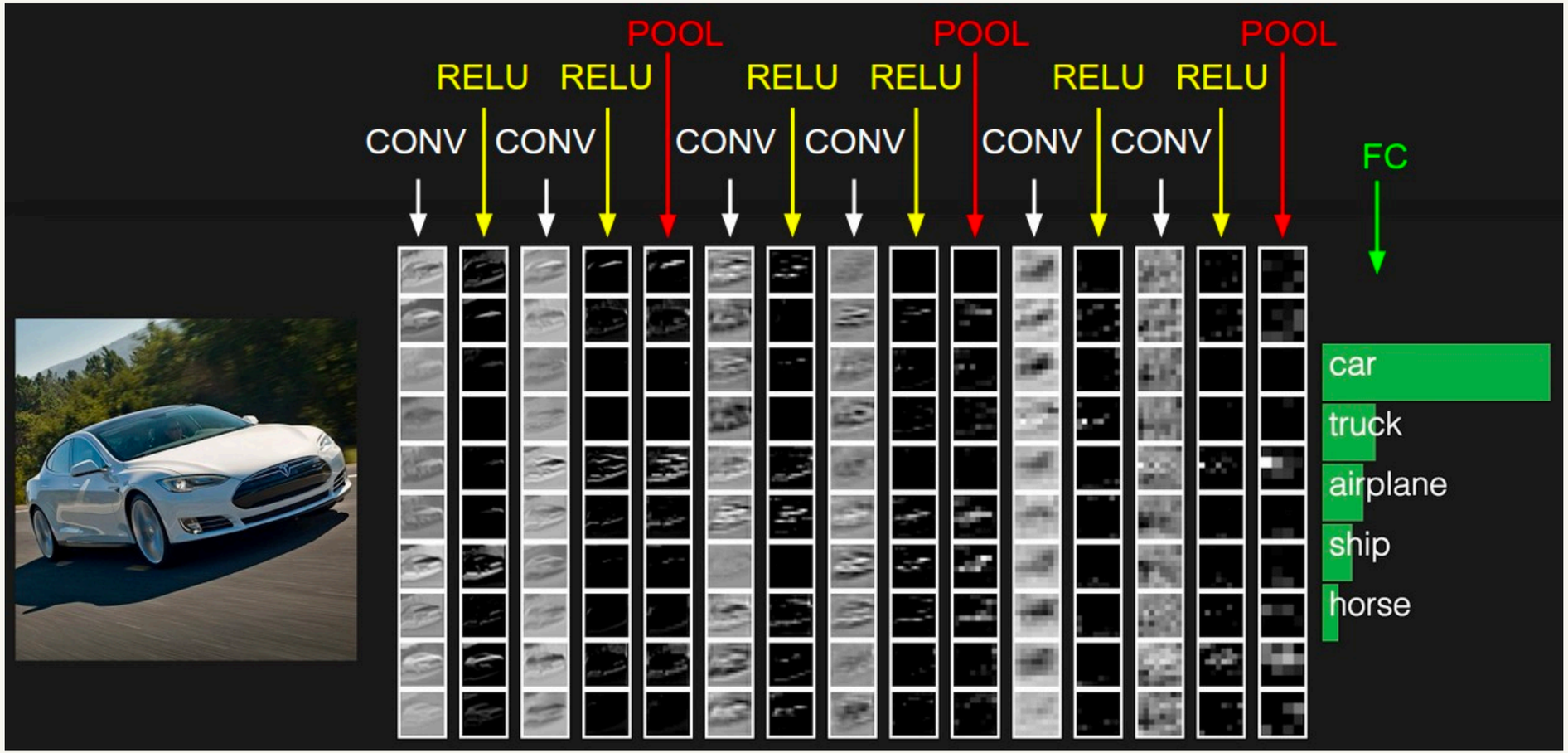
CNN in action! (1)



CNN in action! (1)



CNN in action! (2)



https://cs231n.github.io/convolutional-networks/

Convolutional layers

- The basic architecture of a CNN is multi-channel convolution consisting of multiple single convolutions. The output of the previous layer (or the original image of the first layer) is used as the input of the current layer. It is then convolved with the filter in the layer and serves as the output of this layer. The convolution kernel of each layer is the weight to be learned. Similar to FCN, after the convolution is complete, the result should be biased and activated through activation functions before being input to the next layer.

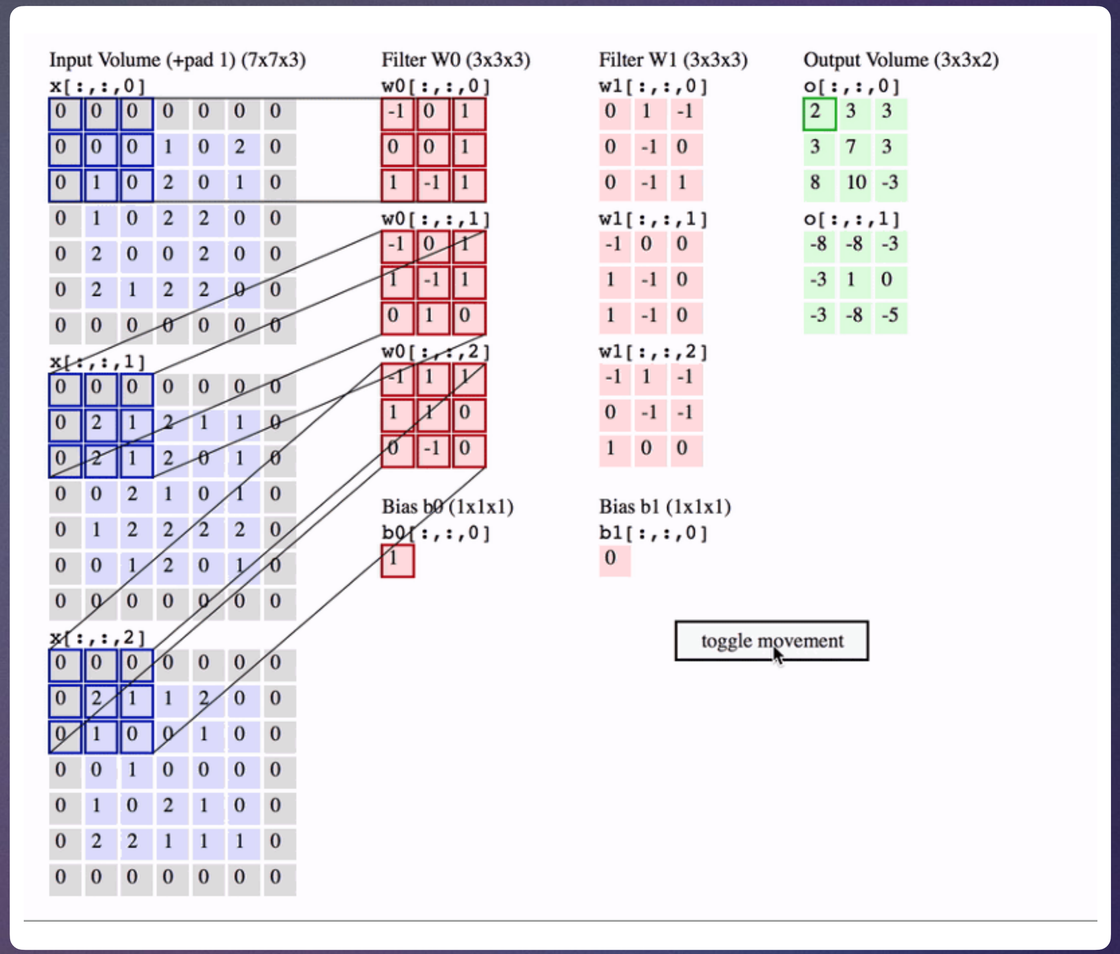
Summarizing the convolutional layers

- Accepts a volume of size $W_1 \times H_1 \times D_1$.
- Requires four hyper-parameters:
  - ➡ Number of filters K,
  - ➡ their spatial extent F
  - ➡ the stride S,
  - ➡ the amount of zero padding P.
- A common settings are $F = 3$, $S = 1$, $P = 1$.

- Produces a volume of size $W_2 \times H_2 \times D_2$ where:
  - ➡ $W_2 = (W_1 - F + 2P)/S + 1$
  - ➡ $H_2 = (H_1 - F + 2P)/S + 1$ (i.e. symmetry)
  - ➡ $D_2 = K$
- With parameter sharing, it introduces $F \cdot F \cdot D_1$ weights per filter, for a total of $(F \cdot F \cdot D_1) \cdot K$ weights and K biases.
- In the output volume, the d-th depth slice (of size $(W_2 \times H_2)$ is the result of performing a valid convolution of the d-th filter over the input volume with a stride of S, and then offset by d-th bias.

Convolution Demo

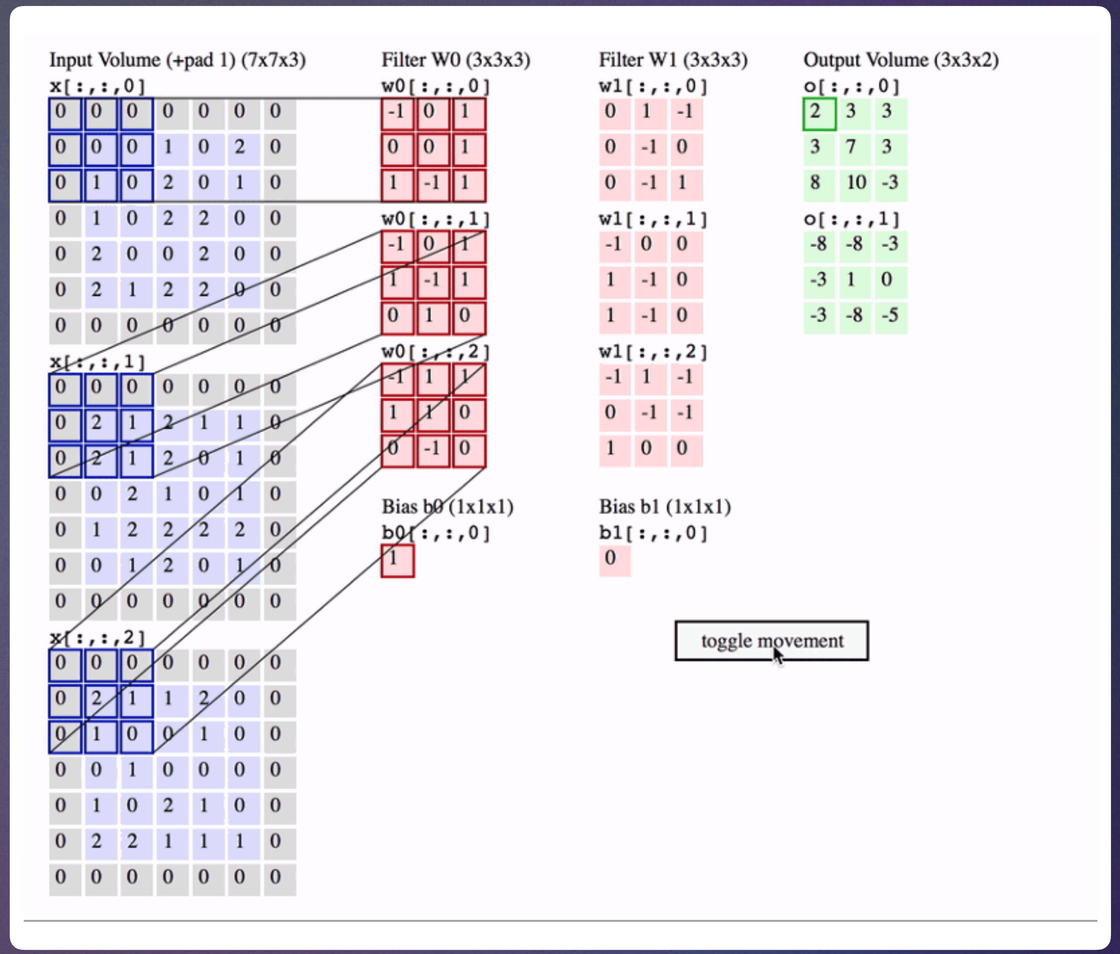
- Since 3D volumes are hard to visualize, all the volumes (the input volume (in blue), the weight volumes (in red), the output volume (in green)) are visualized with each depth slice stacked in rows. The input volume is of size $W_1 = 5$, $H_1 = 5$, $D_1 = 3$, and the CONV layer parameters are $K = 2$, $F = 3$, $S = 2$, and $P = 1$. That is, we have two filters of size 3×3, and they are applied with a stride of 2. Therefore, the output volume size has spatial size $(5 - 3 + 2)/2 + 1 = 3$.
- Moreover, notice that a padding of $P = 1$ is applied to the input volume, making the outer border of the input volume zero. The visualization below iterates over the output activations (green), and shows that each element is computed by element-wise multiplying the highlighted input (blue) with the filter (red), summing it up, and then offsetting the result by the bias.



https://cs231n.github.io/convolutional-networks/

Convolution Demo

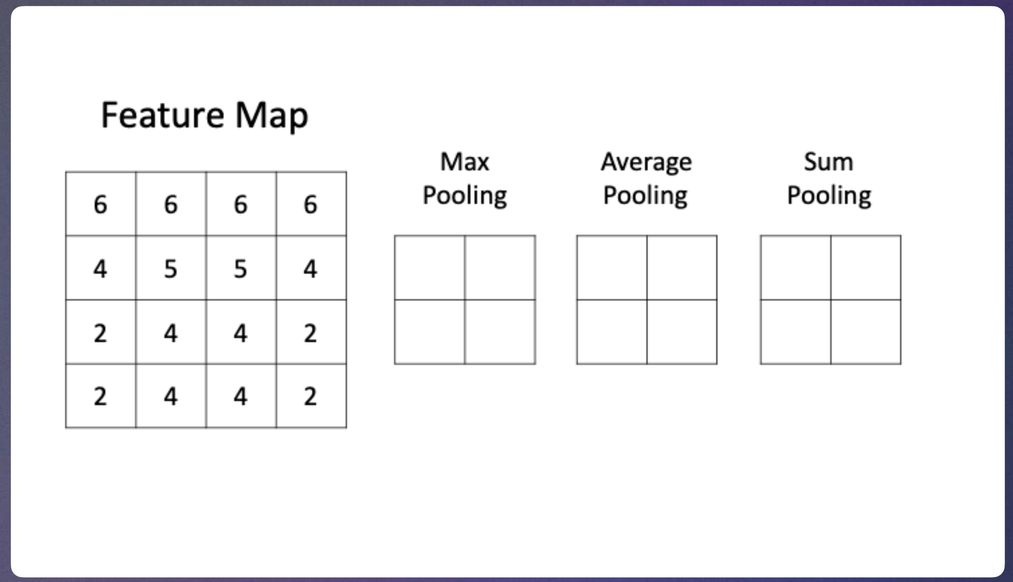
- Since 3D volumes are hard to visualize, all the volumes (the input volume (in blue), the weight volumes (in red), the output volume (in green)) are visualized with each depth slice stacked in rows. The input volume is of size $W_1 = 5$, $H_1 = 5$, $D_1 = 3$, and the CONV layer parameters are $K = 2$, $F = 3$, $S = 2$, and $P = 1$. That is, we have two filters of size 3×3, and they are applied with a stride of 2. Therefore, the output volume size has spatial size $(5 - 3 + 2)/2 + 1 = 3$.
- Moreover, notice that a padding of $P = 1$ is applied to the input volume, making the outer border of the input volume zero. The visualization below iterates over the output activations (green), and shows that each element is computed by element-wise multiplying the highlighted input (blue) with the filter (red), summing it up, and then offsetting the result by the bias.



https://cs231n.github.io/convolutional-networks/

Pooling layers

- Pooling combines nearby units to reduce the size of the input on the next layer, reducing dimensions. Common pooling includes max pooling and average pooling. When max pooling is used, the maximum value in a small square area is selected as the representative of this area, while the mean value is selected as the representative when average pooling is used. The side of this small area is the pool window size. The following figure shows the max pooling operation whose pooling window size is 2.

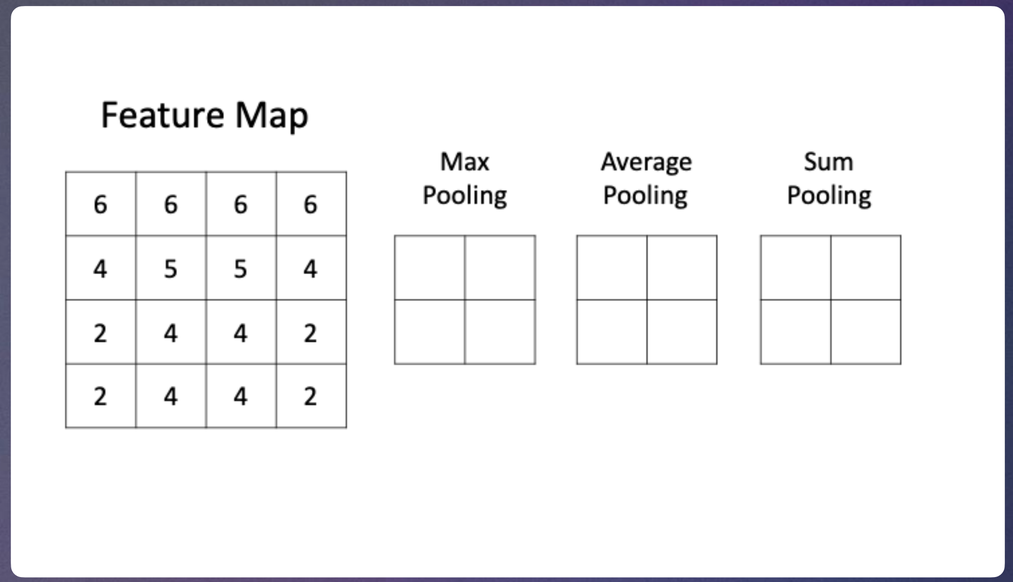


- The result of using a pooling layer and creating down sampled or pooled feature maps is a summarized version of the features detected in the input. They are useful as small changes in the location of the feature in the input detected by the convolutional layer will result in a pooled feature map with the feature in the same location. This capability added by pooling is called the model's invariance to local translation.

https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/#:~:text=A%20pooling%20layer%20is%20a,Convolutional%20Layer

Pooling layers

- Pooling combines nearby units to reduce the size of the input on the next layer, reducing dimensions. Common pooling includes max pooling and average pooling. When max pooling is used, the maximum value in a small square area is selected as the representative of this area, while the mean value is selected as the representative when average pooling is used. The side of this small area is the pool window size. The following figure shows the max pooling operation whose pooling window size is 2.



- The result of using a pooling layer and creating down sampled or pooled feature maps is a summarized version of the features detected in the input. They are useful as small changes in the location of the feature in the input detected by the convolutional layer will result in a pooled feature map with the feature in the same location. This capability added by pooling is called the model's invariance to local translation.

https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/#:~:text=A%20pooling%20layer%20is%20a,Convolutional%20Layer

Fully Connected layers

- The fully connected layer is essentially a classifier. The features extracted on the convolutional layer and pooling layer are straightened and placed at the fully connected layer to output and classify results.
- Generally, the $\mathrm{Softmax}$ function is used as the activation function of the final fully connected output layer to combine all local features into global features and calculate the score of each type.

$$\operatorname{Softmax}(\mathbf{x})_i = \frac{\exp(x_i)}{\sum_{j}^{K} \exp(x_j)}.$$

References

- Fei-Fei Li et al. CS231n: Convolutional Neural Networks for Visual Recognition. http://cs231n.stanford.edu/. 2020, Accessed on Feb 2021.
- Juan Cruz Martinez. Introduction to Convolutional Neural Networks CNNs. https://aigents.co/blog/publication/introduction-to-convolutional-neural-networks-cnns. 2020, Accessed on Feb 2021.
- HUAWEI. Deep Learning Overview. 2020, Accessed on Feb 2021.

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Ceará

HUAWEI

Thank you for your attention!

🙏☺️

Prof. Me. Saulo A. F. Oliveira
saulo.oliveira@ifce.edu.br