Intrinsic dimension and density profile of neural representations

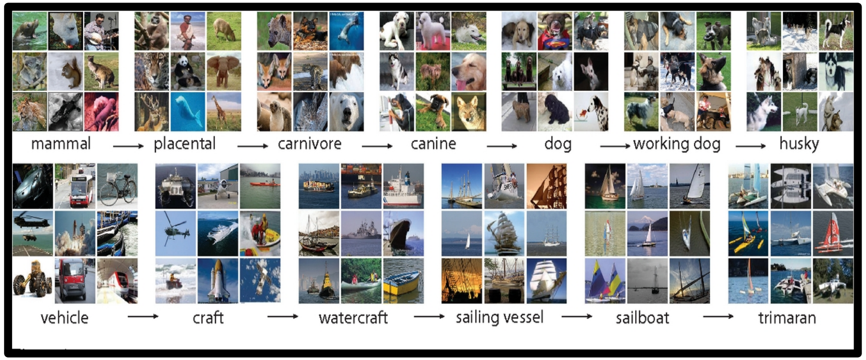
Diego Doimo and Aldo Glielmo

International School for Advanced Studies (SISSA), Trieste

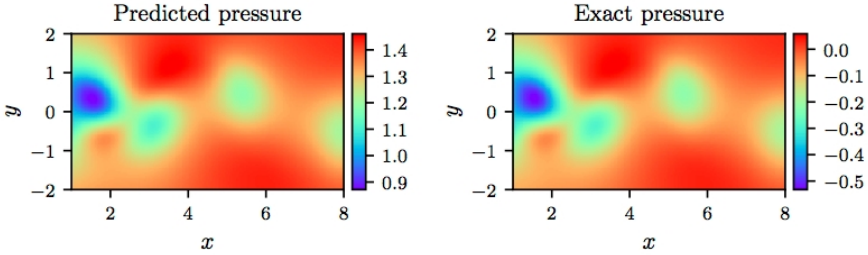
UniTS, 27 April 2021

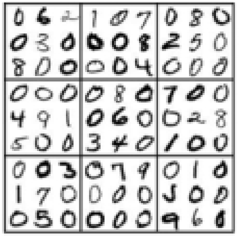
What can we do with deep learning models?

Image classification



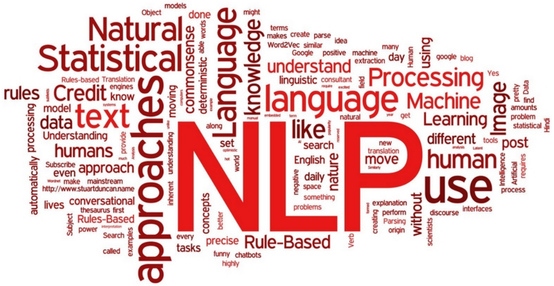
Modeling physical systems

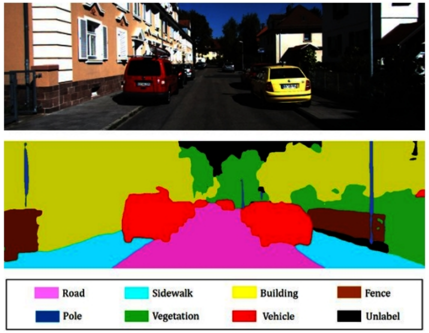




Solving constrained optimization

Translation, test generation, sentiment analysis, …



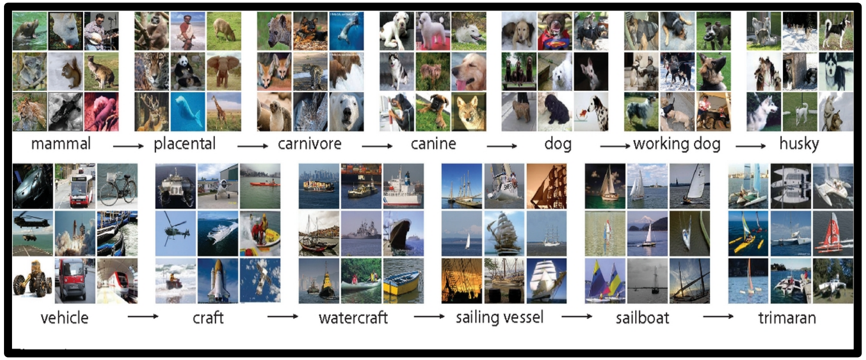


Semantic segmentation

Generative models

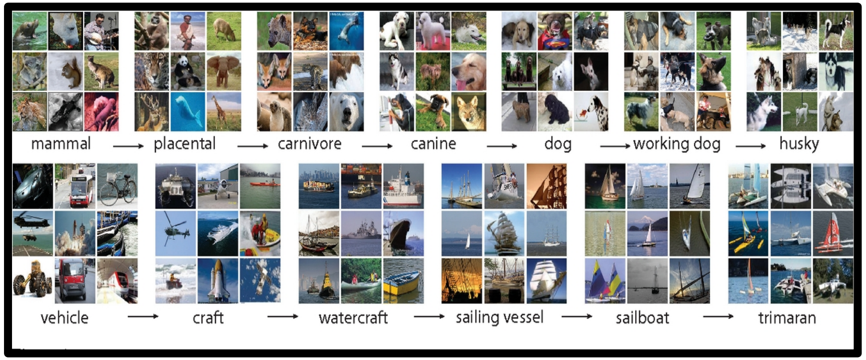
What do we learn with deep learning models?

Image classification

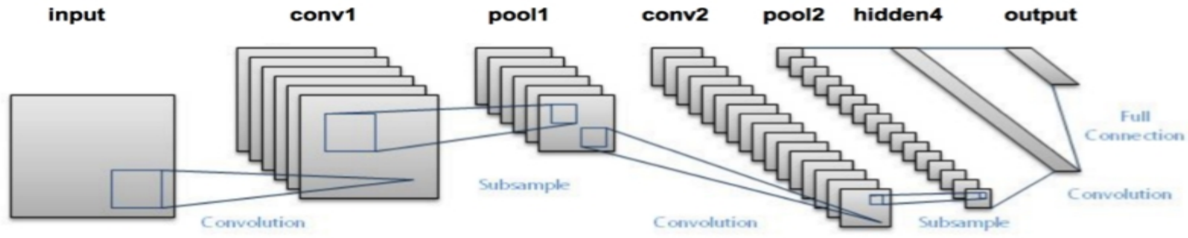


What do we learn with deep learning models?

Image classification

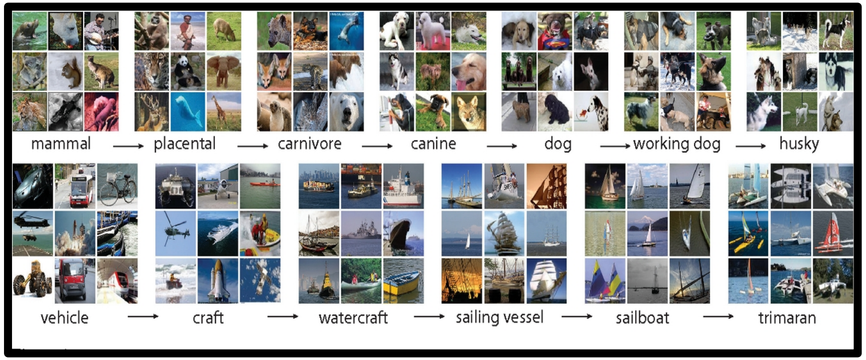


Convolutional neural networks

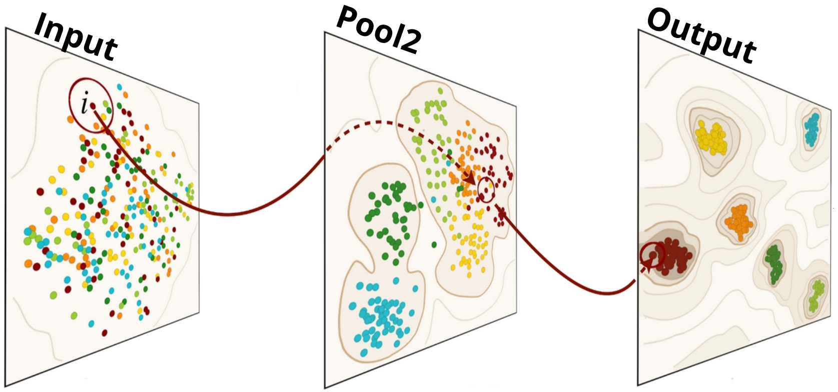


What do we learn with deep learning models?

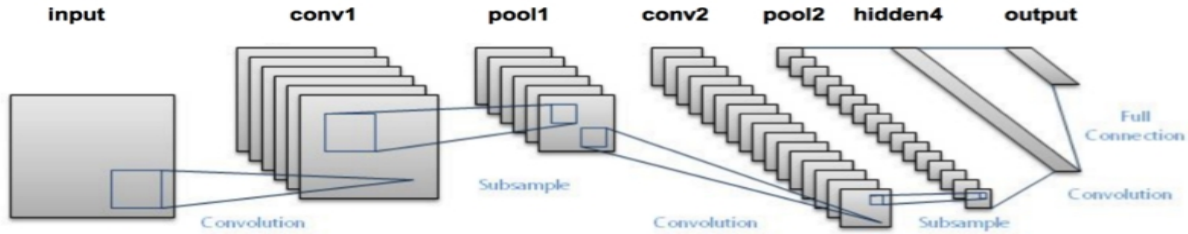
Image classification



Representations



Convolutional neural networks



The importance of representations in neural networks

Representations arise automatically → Need to understand their meaning

- To make NN more interpretable

  Q1: When / How do interpretable representations arise ?

- To transfer efficiently the learned concepts

  Q2: Which information is encoded in a given representation?

- To improve the architecture design

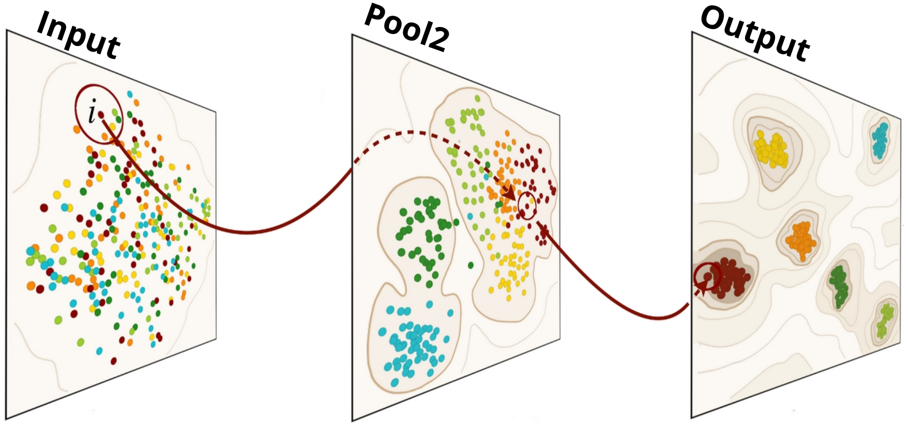
  What is the depth required to achieve a given performance?

1) **Intrinsic dimension** [Ansuini et al., NeurIPS 2019]

2) **Probability density** [Doimo et al., NeurIPS 2020]

Colab:

https://colab.research.google.com/drive/1fTxE0GWb5BobZhL3j6G6Ra5hBj__c9X-#scrollTo=VrIL_J3FLQab



What is a representation?

=

-1.61
-1.54
-1.53
-1.51
-1.47
-1.54
-1.56

Representation = function of the input data

$$X_L = f(X_0)$$

f = neural network

0.00
0.00
0.00
0.95
0.00
0.00
0.00

What is a representation?

=

•
•

-1.61
-1.54
-1.53
-1.51
-1.47
-1.54
-1.56

•
•

Input representation → X_0

Representation = function of the input data

$$X_L = f(X_0)$$

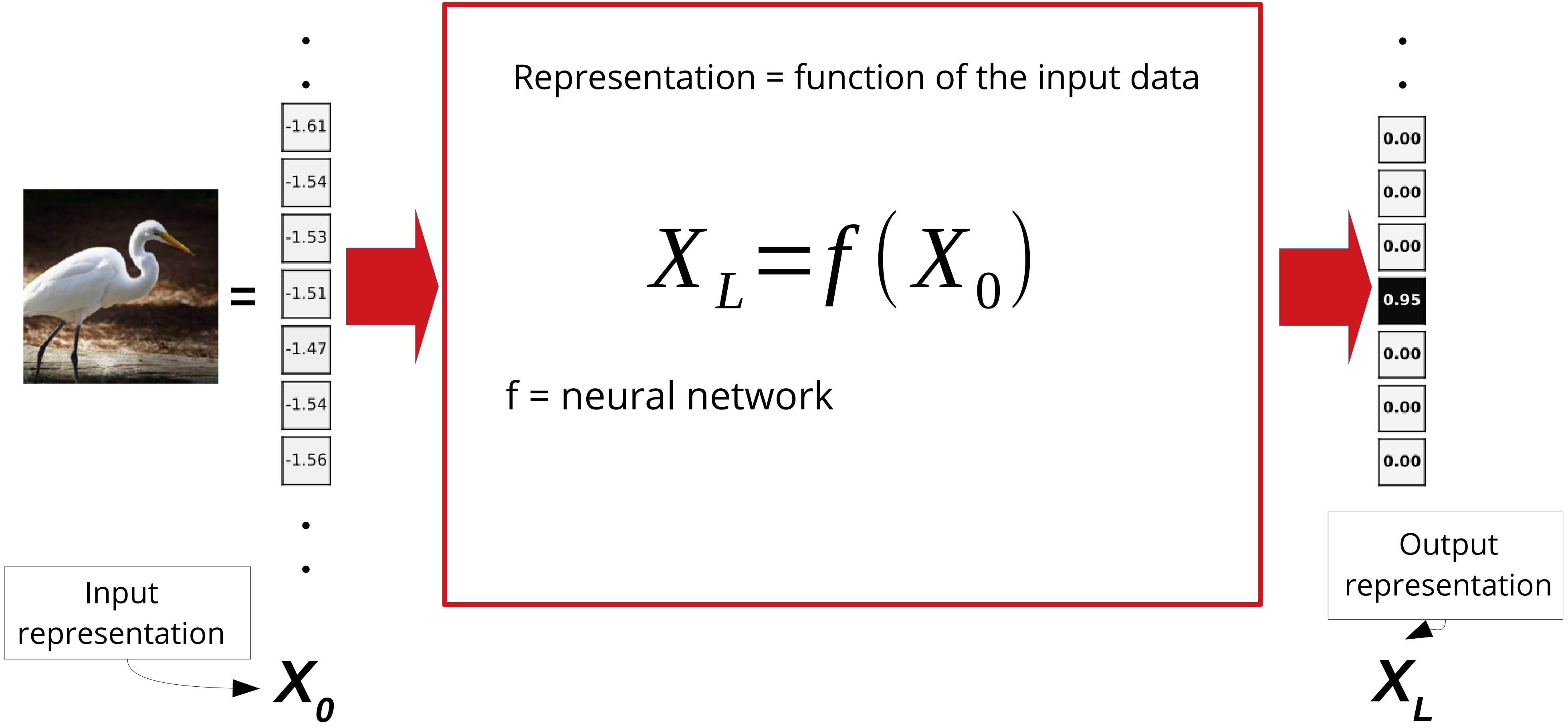
f = neural network

•
•

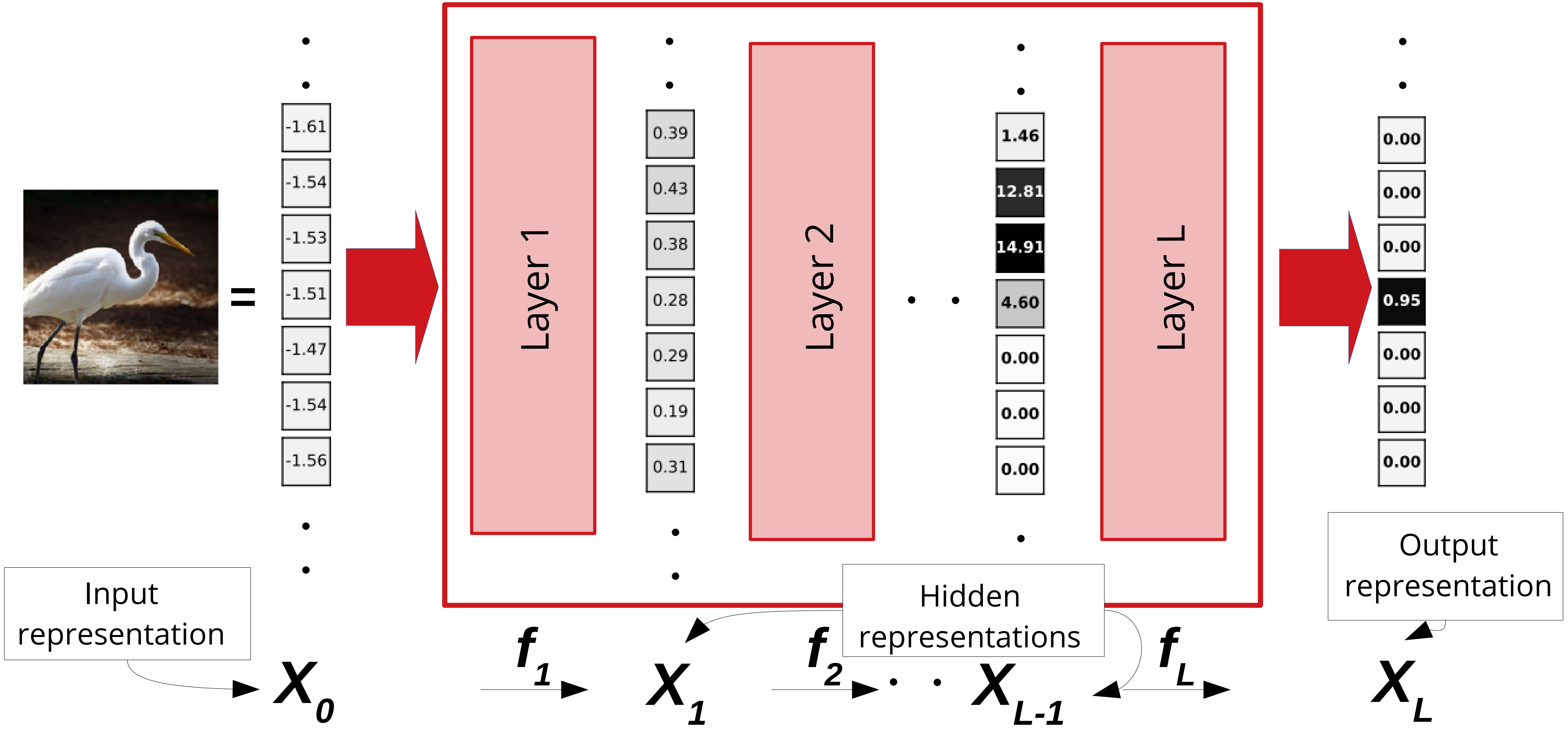
0.00
0.00
0.00
0.95
0.00
0.00
0.00

•
•

What is a representation?

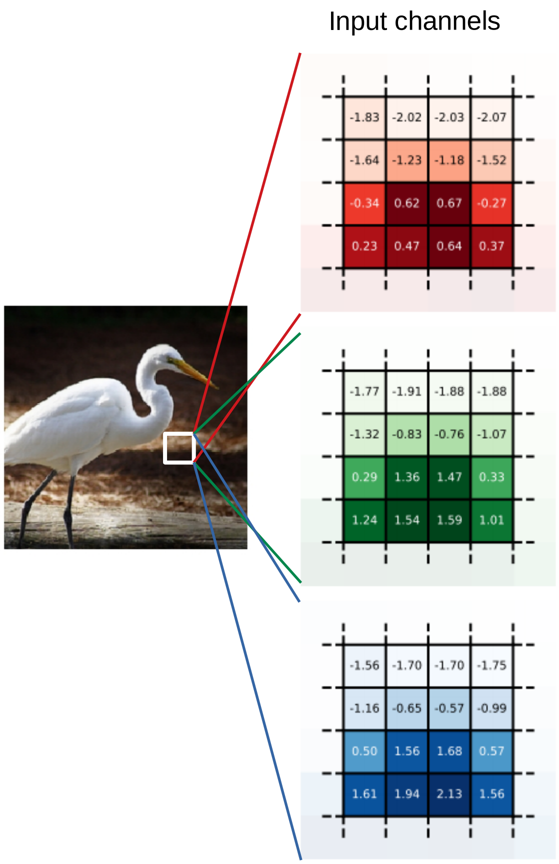


What is a representation?

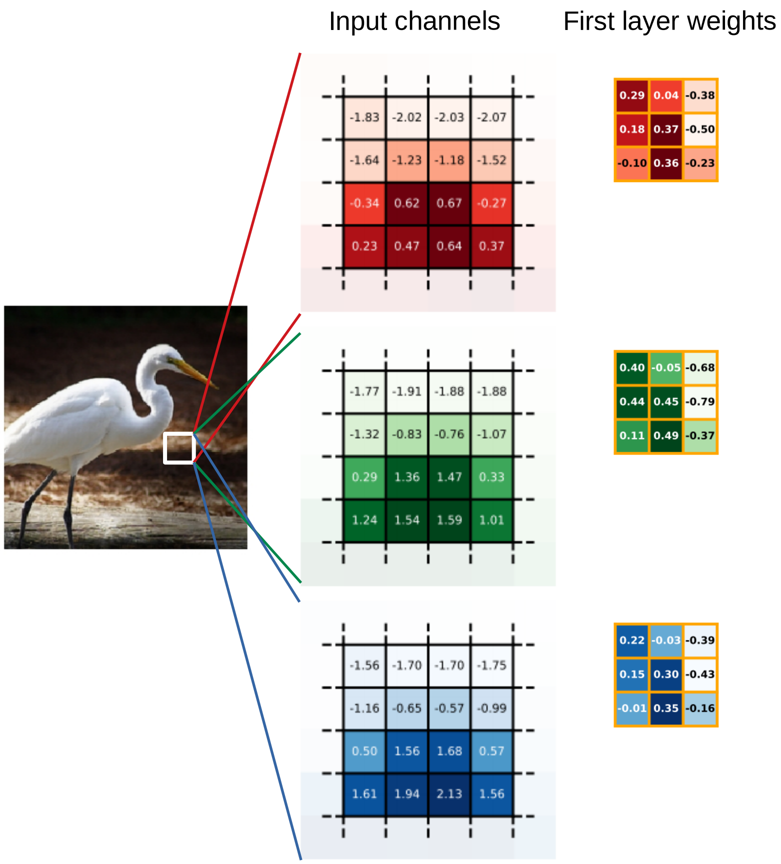


What is a representation in a convolutional neural network?

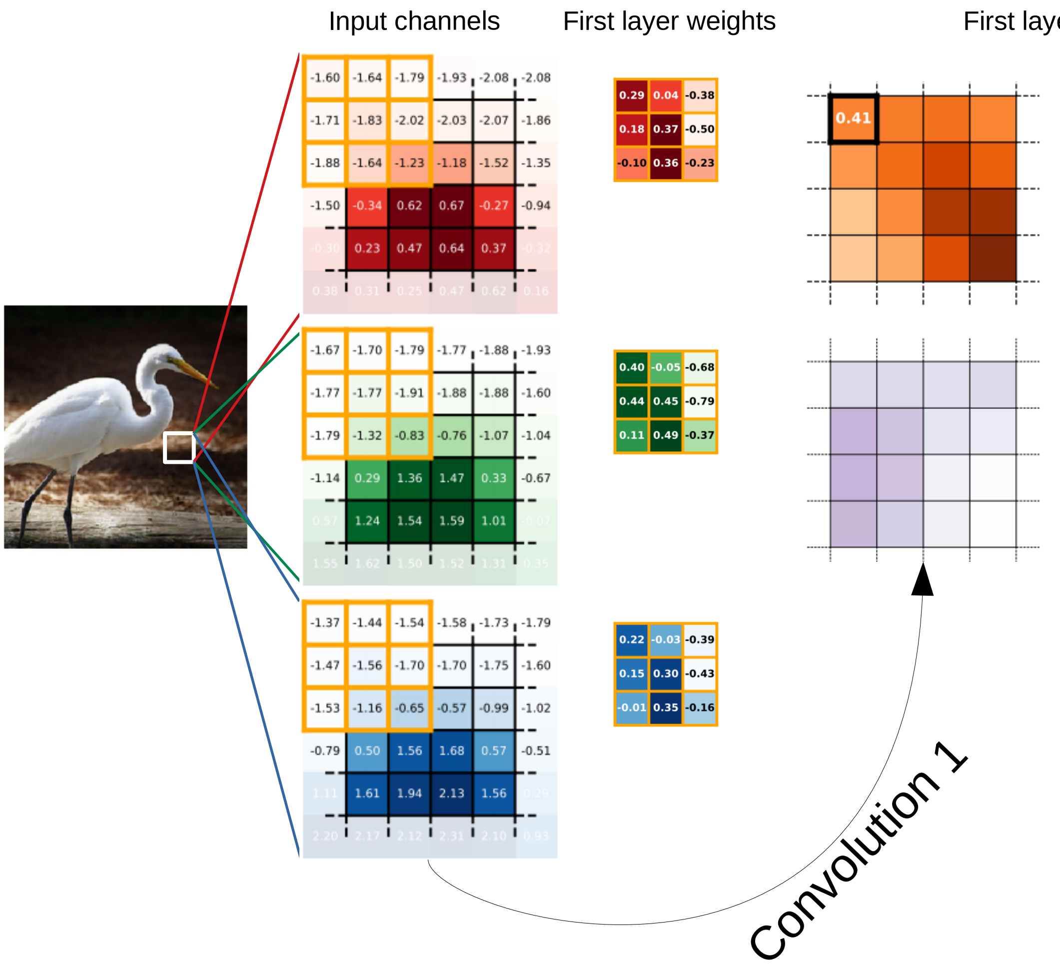
What is a representation in a convolutional neural network?



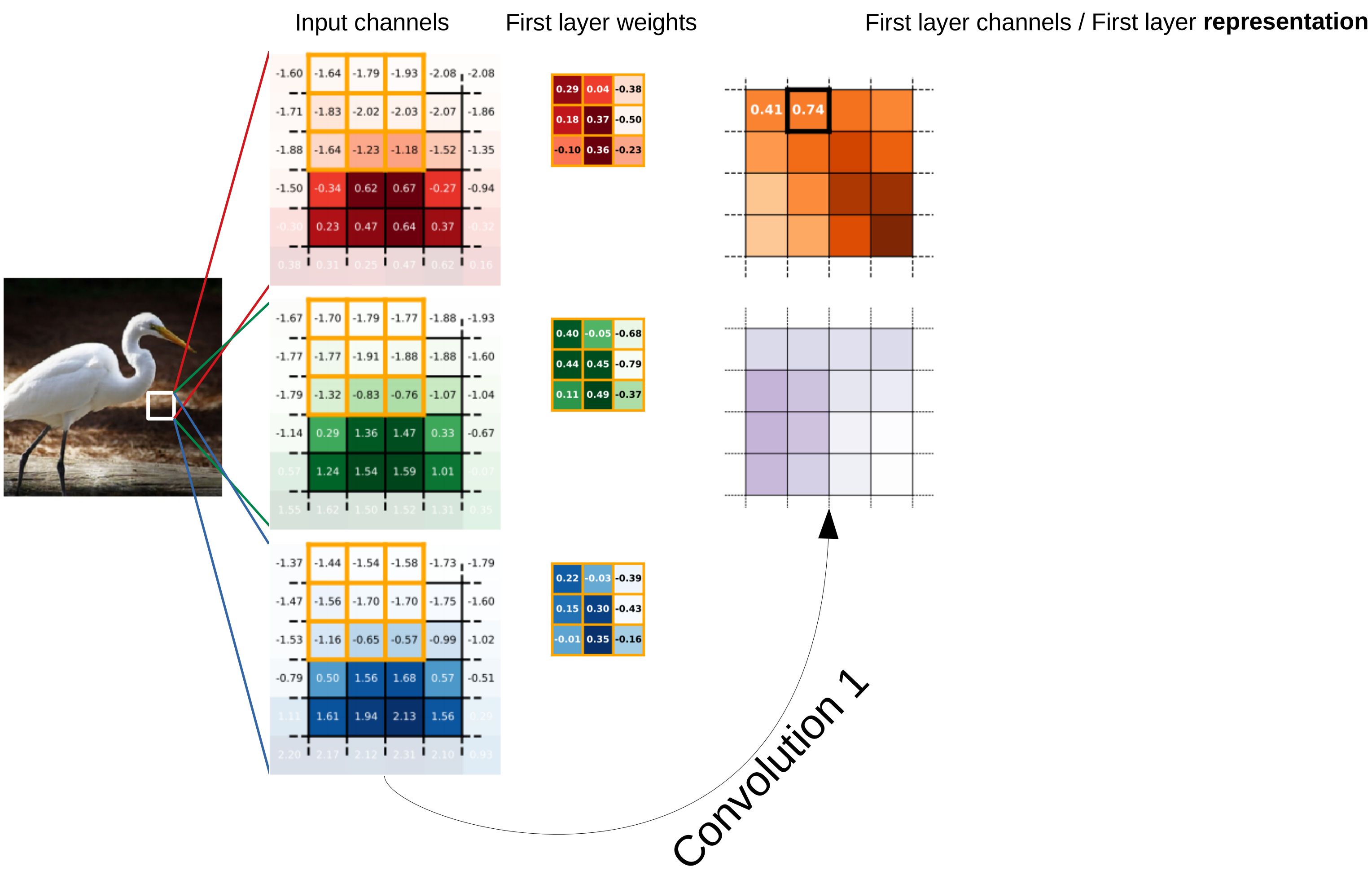
What is a representation in a convolutional neural network?



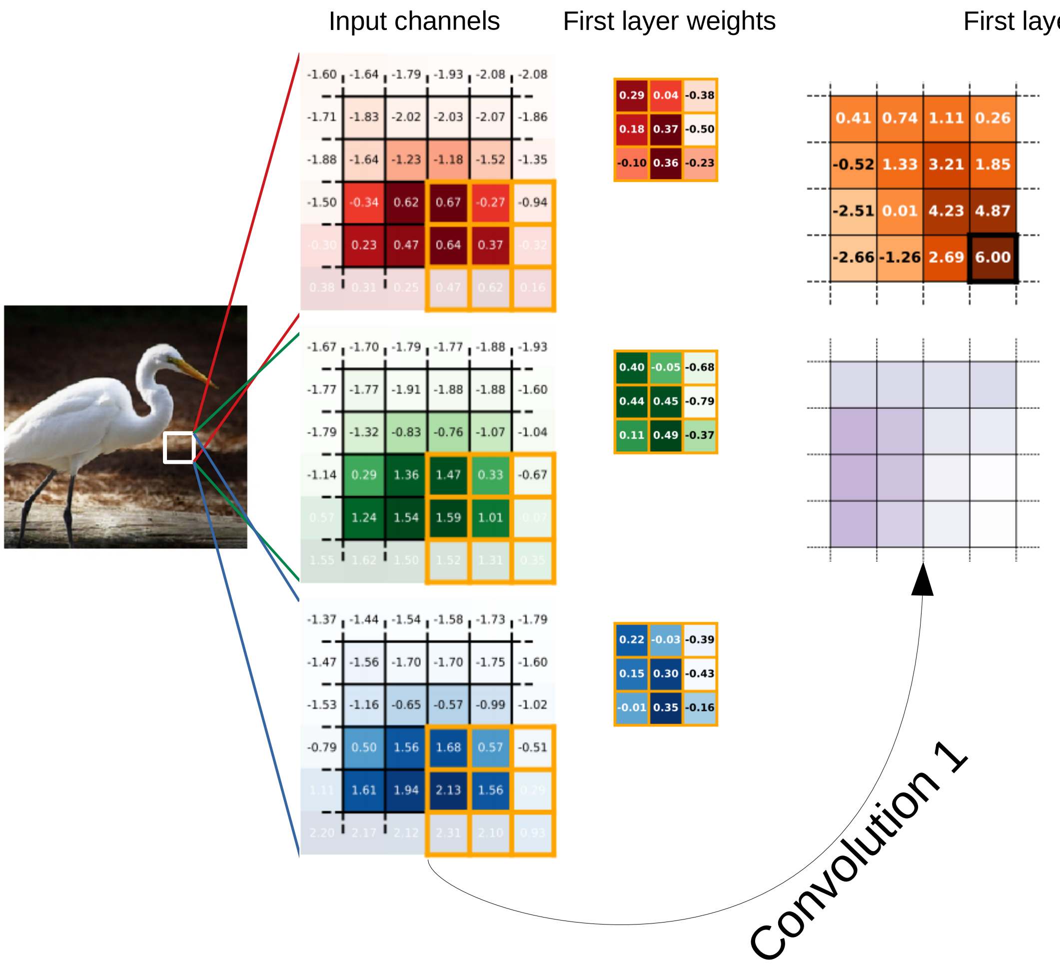
What is a representation in a convolutional neural network?



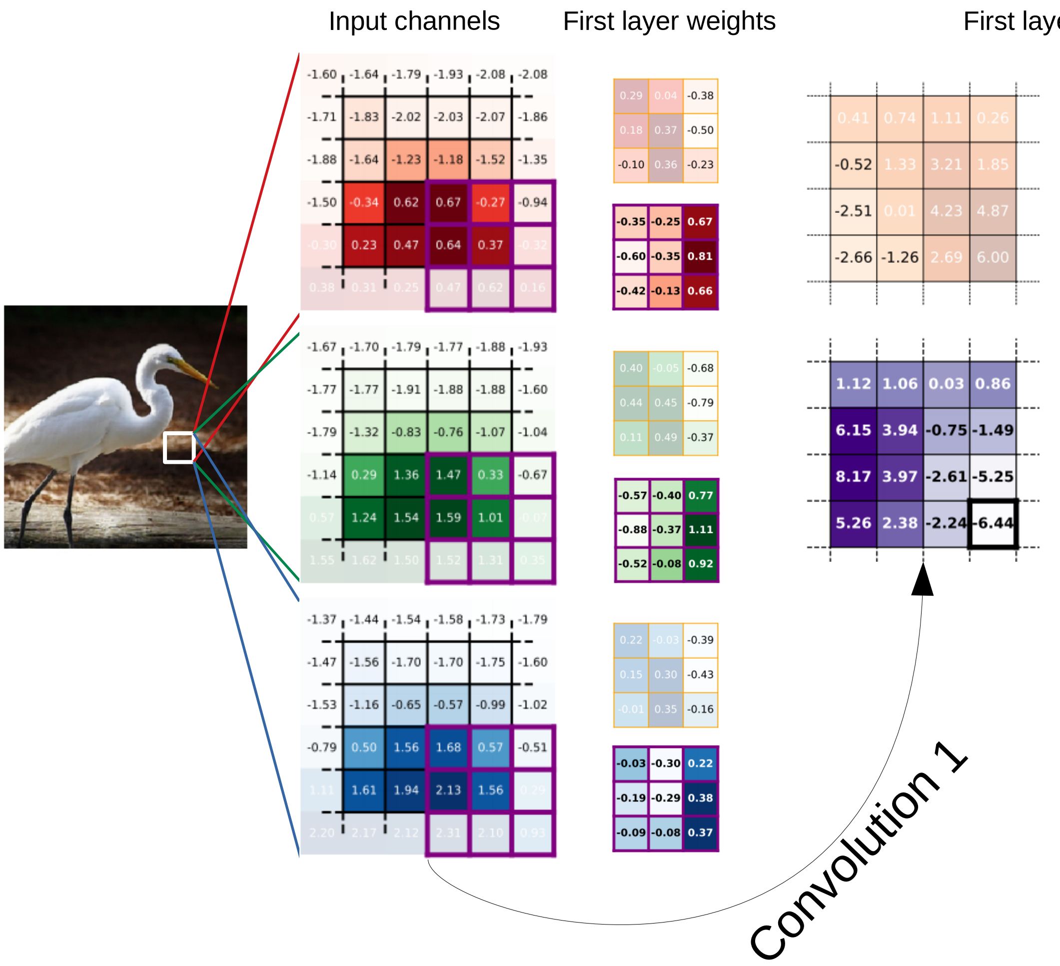
What is a representation in a convolutional neural network?



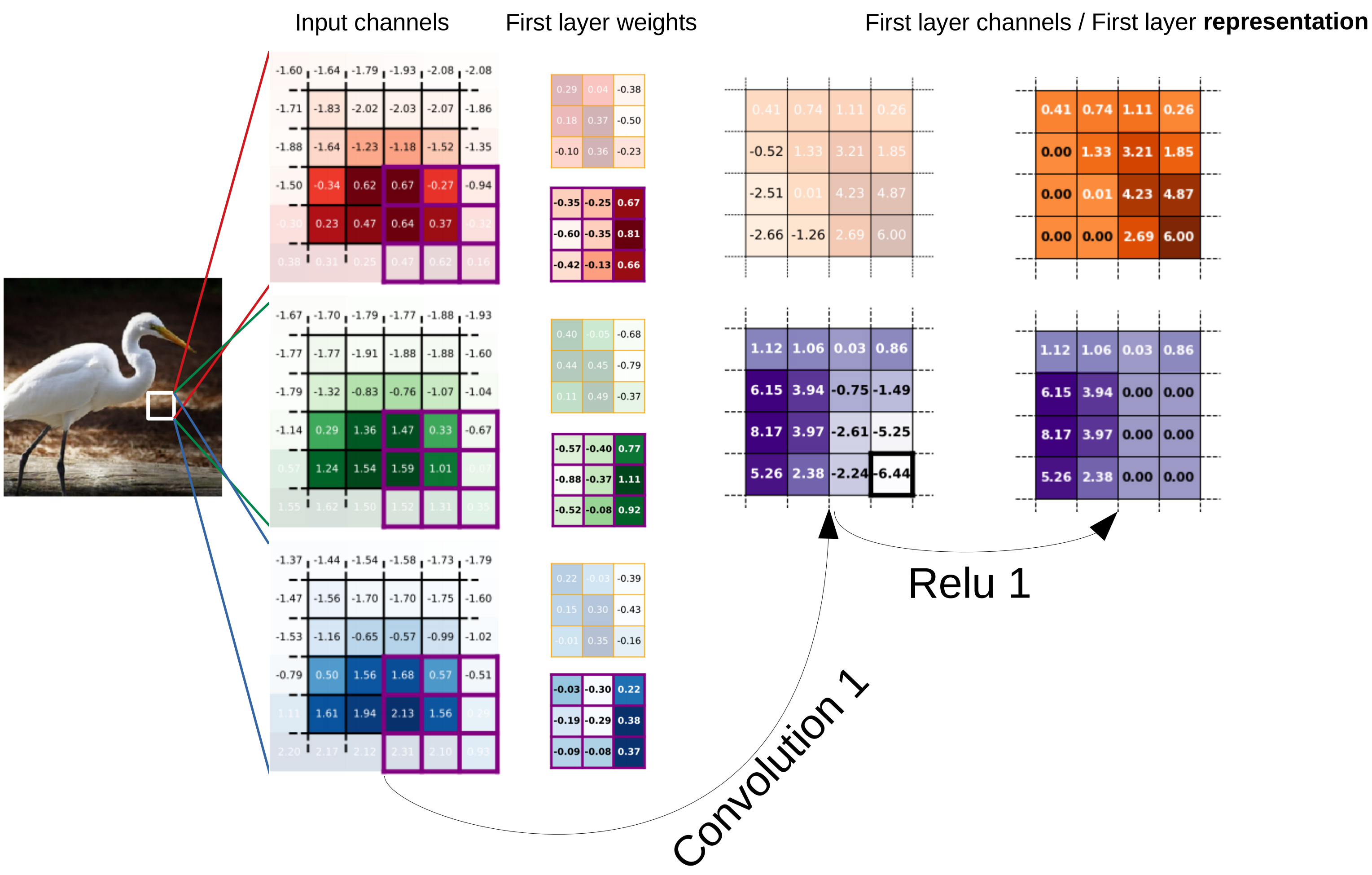
What is a representation in a convolutional neural network?



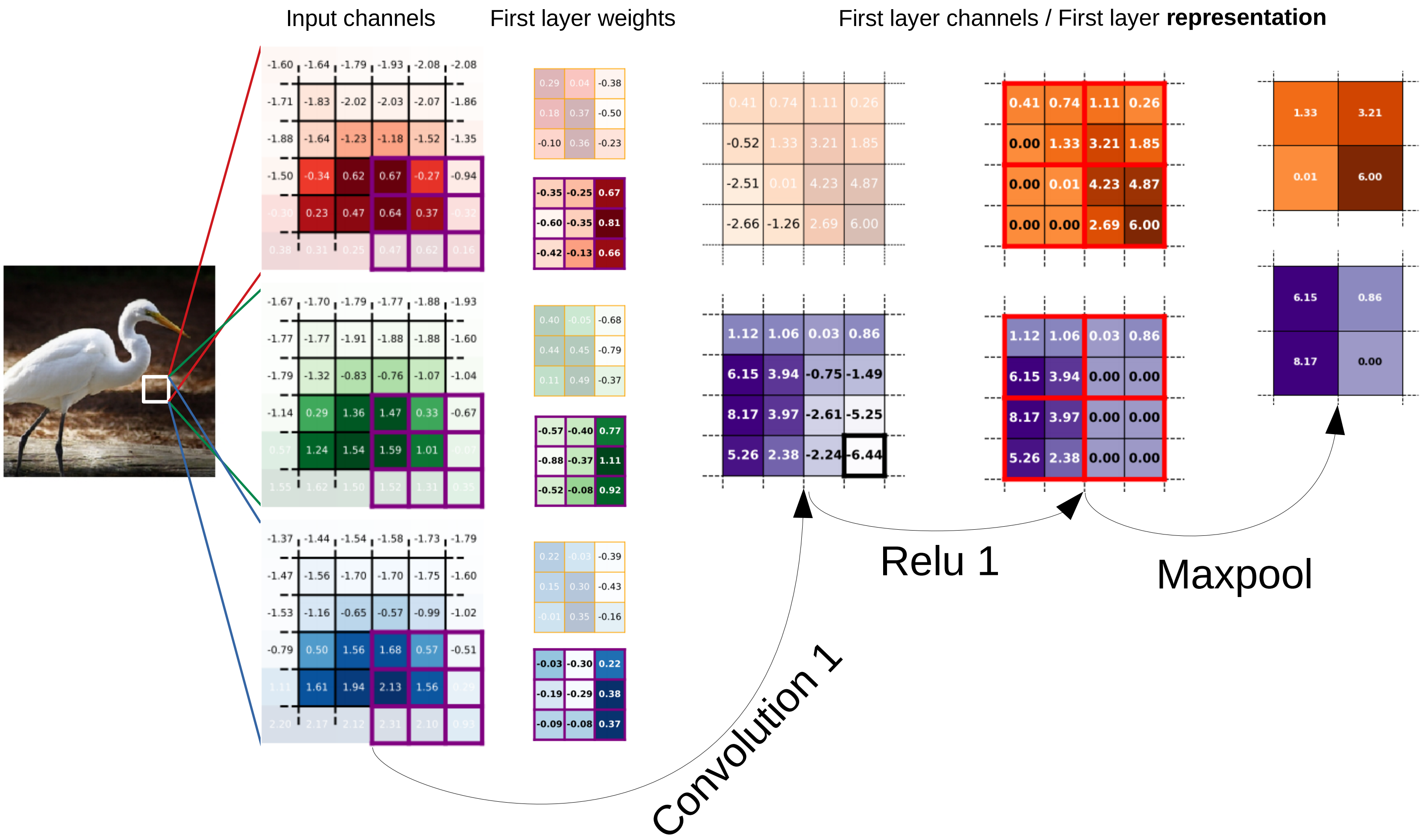
What is a representation in a convolutional neural network?



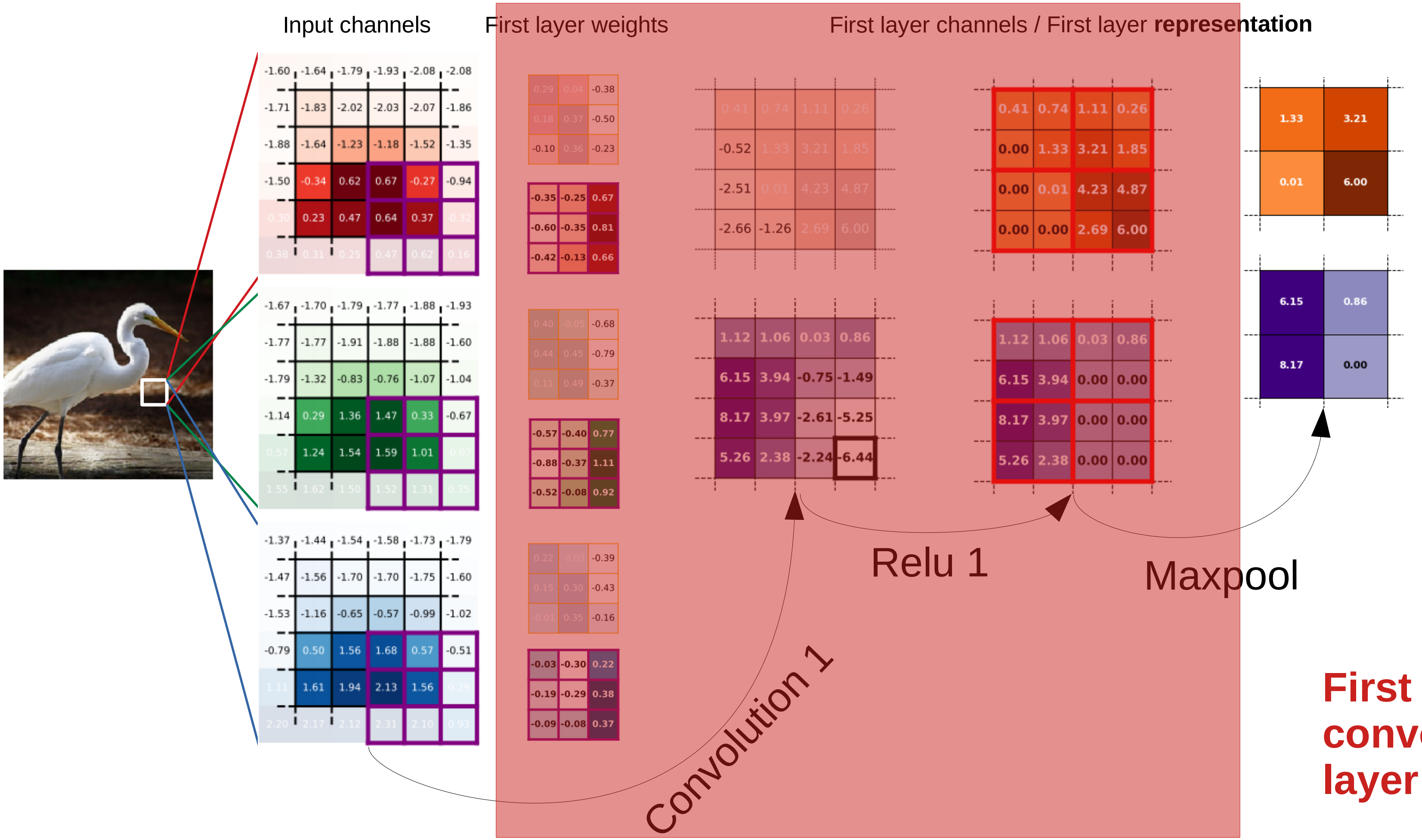
What is a representation in a convolutional neural network?



What is a representation in a convolutional neural network?

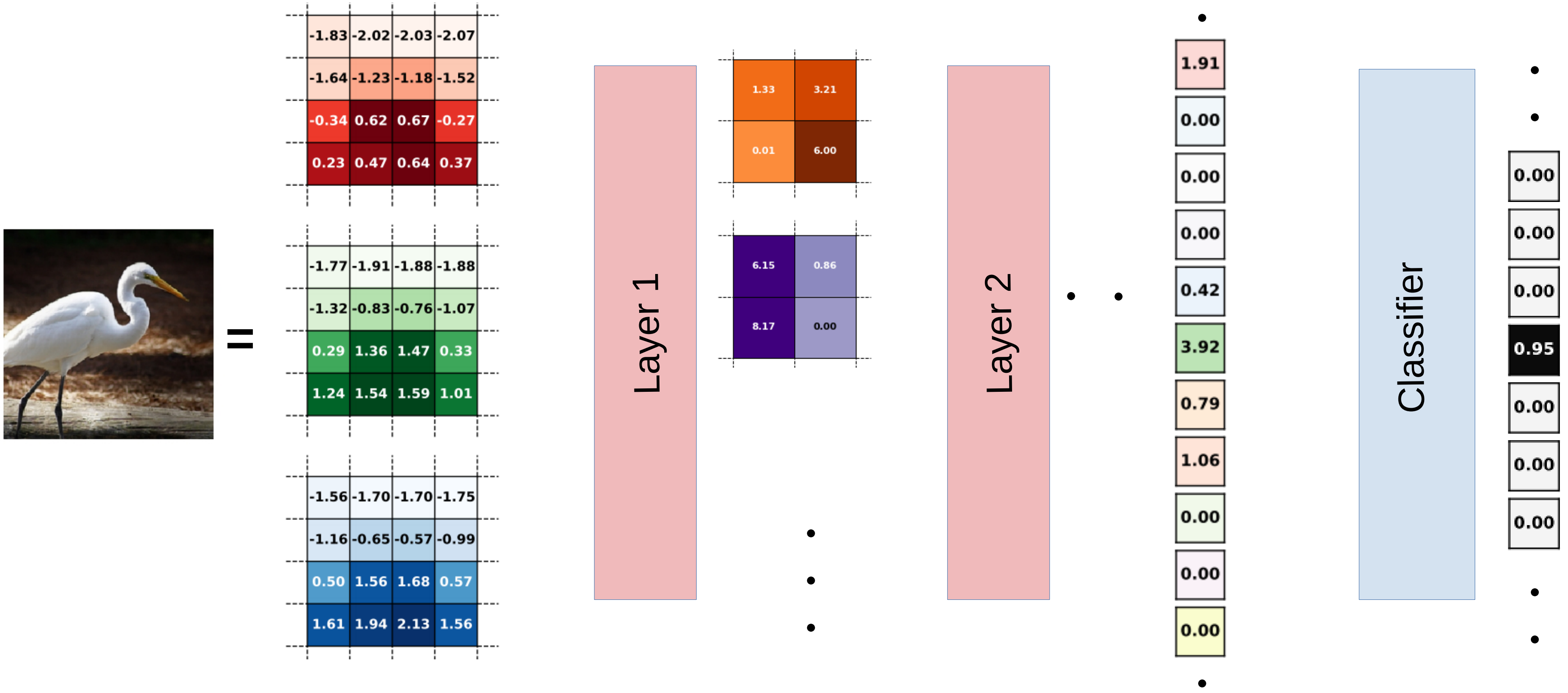


What is a representation in a convolutional neural network?



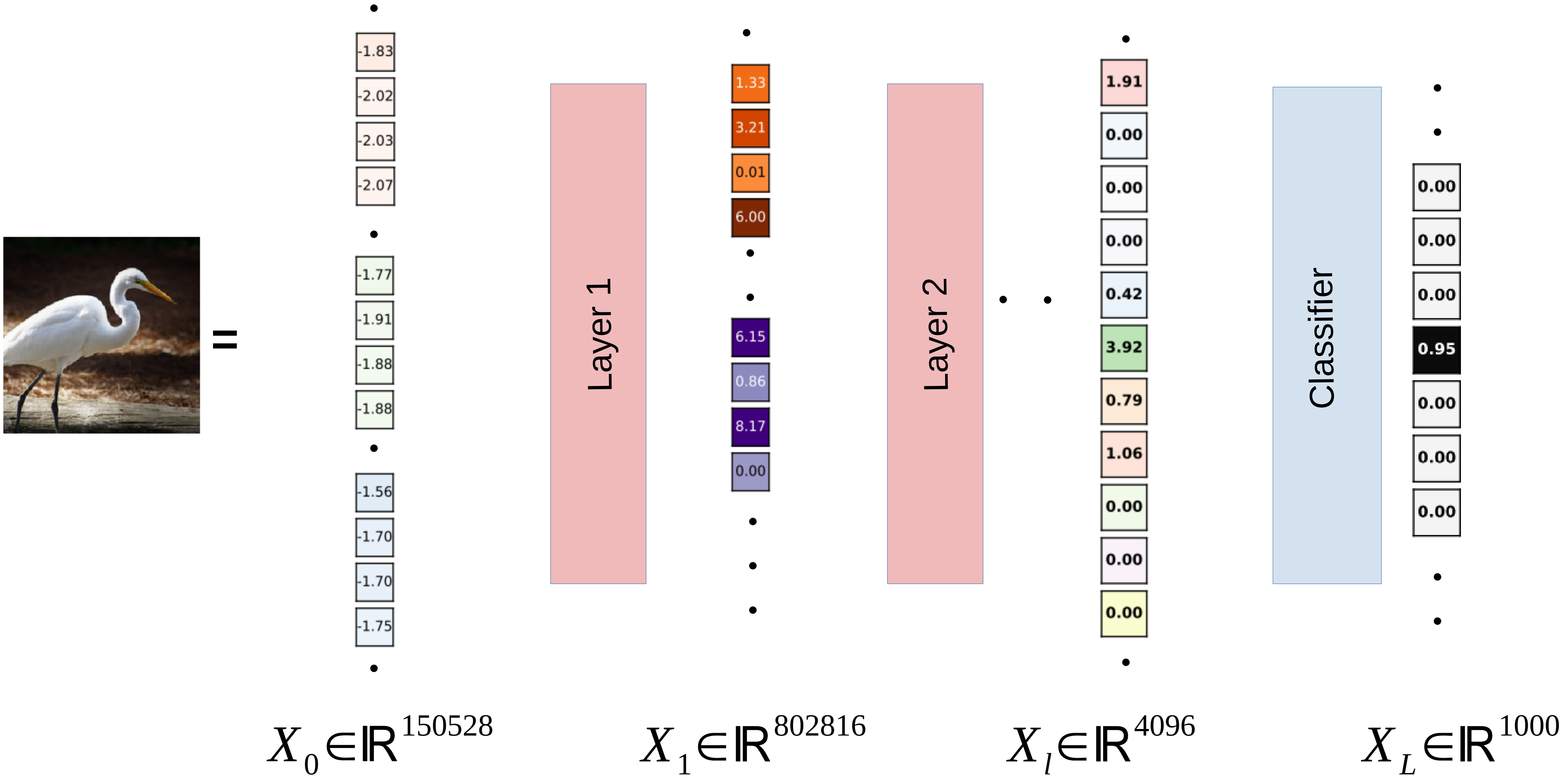
First convolutional layer

What is a representation in a convolutional neural network?



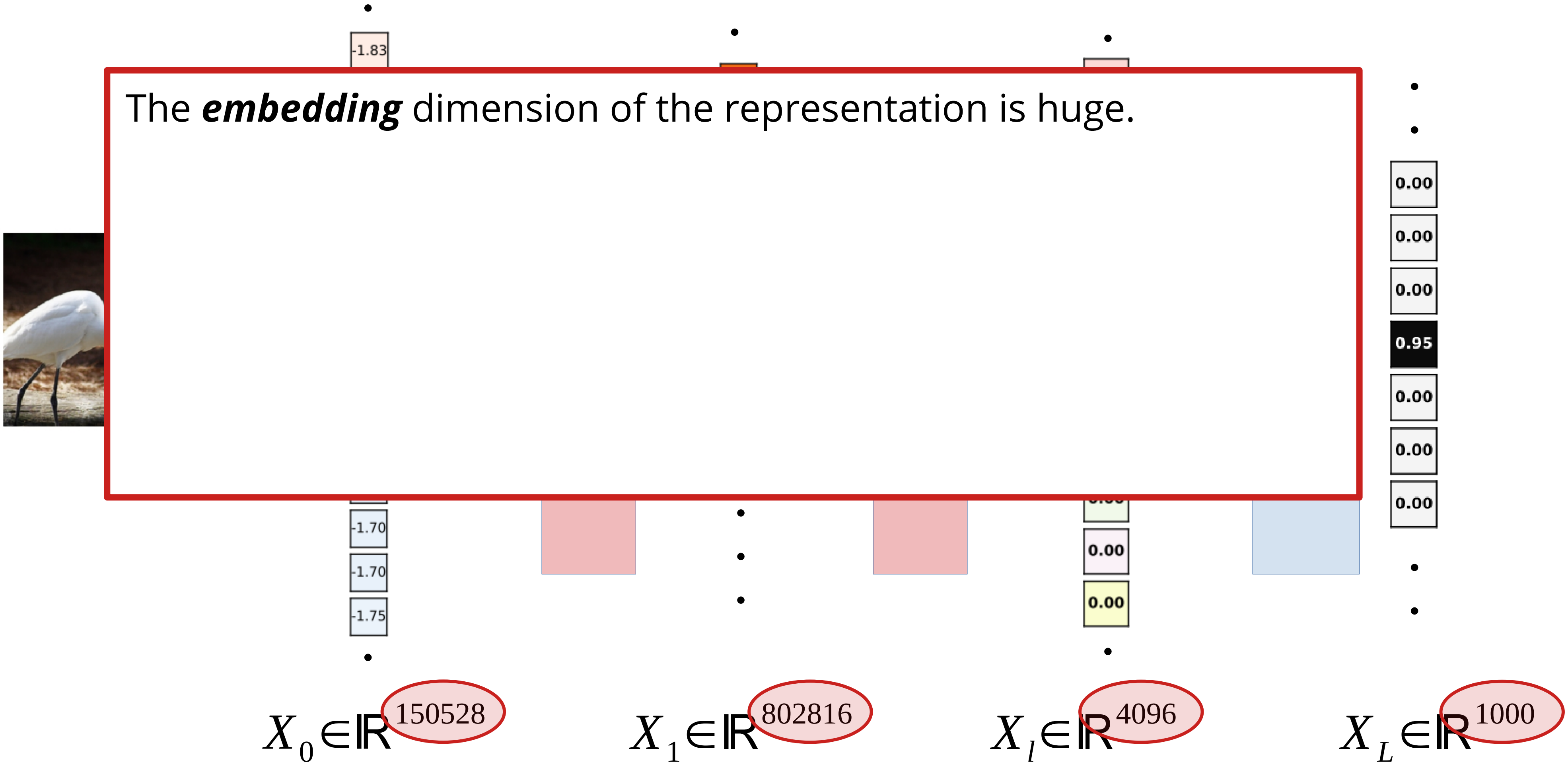
$X_0 \in \mathbb{R}^{3 \times 224 \times 224}$ $X_1 \in \mathbb{R}^{64 \times 112 \times 112}$ $X_l \in \mathbb{R}^{4096 \times 1 \times 1}$ $X_L \in \mathbb{R}^{1000}$

What is a representation in a convolutional neural network?

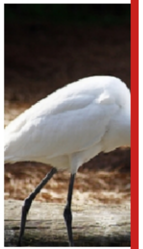


What is a representation in a convolutional neural network?

The ***embedding*** dimension of the representation is huge.



What is a representation in a convolutional neural network?

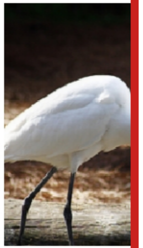


The ***embedding*** dimension of the representation is huge.

a) Any density-based analysis seems hopeless when D > 100

$X_0 \in \mathbb{R}^{150528}$ $X_1 \in \mathbb{R}^{802816}$ $X_l \in \mathbb{R}^{4096}$ $X_L \in \mathbb{R}^{1000}$

What is a representation in a convolutional neural network?



The ***embedding*** dimension of the representation is huge.

a) Any density-based analysis seems hopeless when D > 100

b) Neural networks take advantage of the low dimensional structure of the data.

This is ***not*** true for other classification approaches (kernels,…)

Chizat & Bach, *Implicit bias of gradient descent…* Conference on Learning Theory (2020)

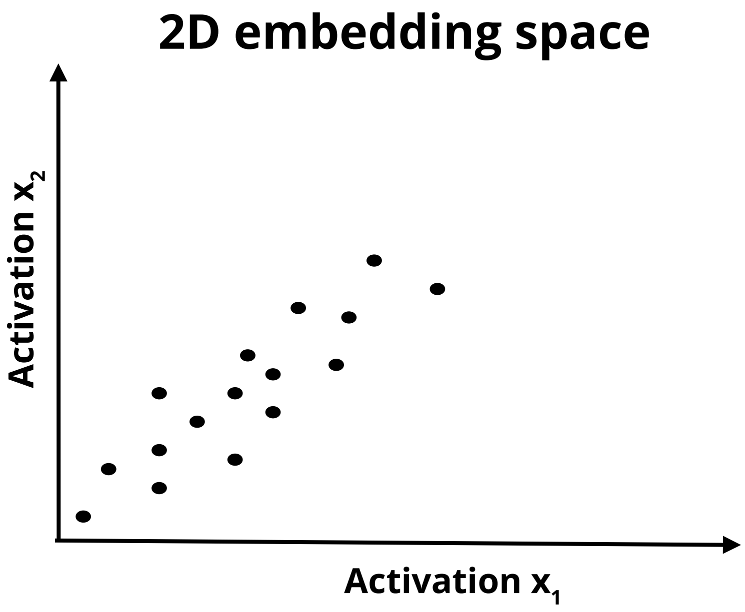
Ghorbani et al, *When do neural networks outperform kernel methods?* NeurIPS (2020)

$X_0 \in \mathbb{R}^{150528}$ $X_1 \in \mathbb{R}^{802816}$ $X_l \in \mathbb{R}^{4096}$ $X_L \in \mathbb{R}^{1000}$

Intrinsic dimension estimation

Intrinsic dimension of a data representation:
minimum number of coordinates to describe the data without significant information loss

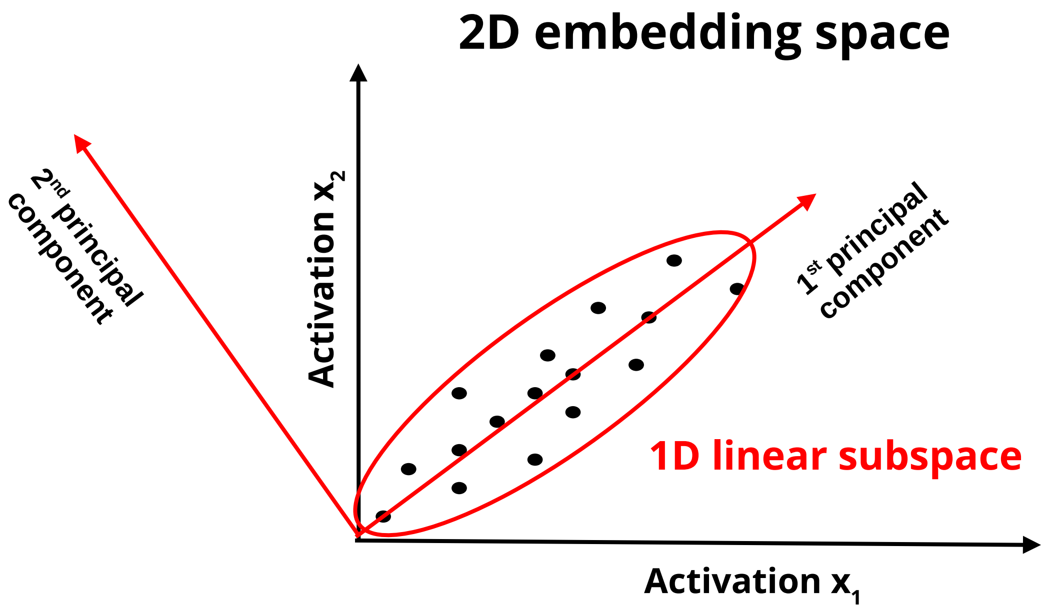
Linear case: Principal Component Analysis (PCA)



Intrinsic dimension estimation

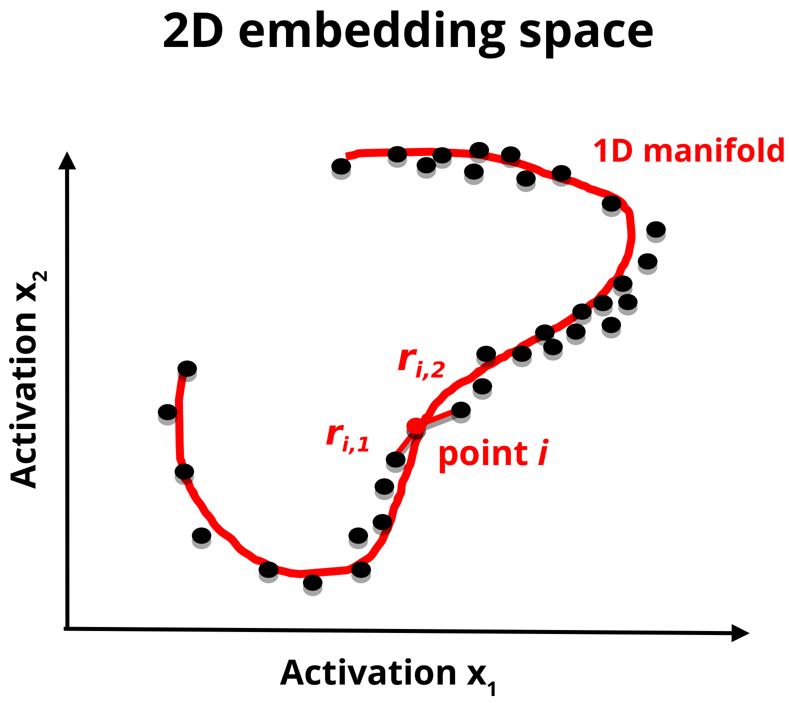
Intrinsic dimension of a data representation:
minimum number of coordinates to describe the data without significant information loss

Linear case: Principal Component Analysis (PCA)



Intrinsic dimension estimation

The general (non linear) case: TwoNN (Facco et al, 2017)

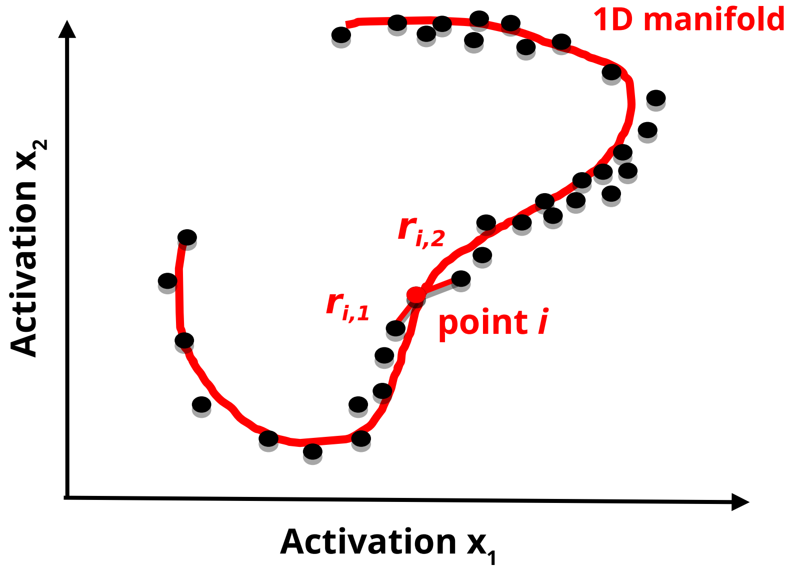


Intrinsic dimension estimation

The general (non linear) case: TwoNN (Facco et al, 2017)

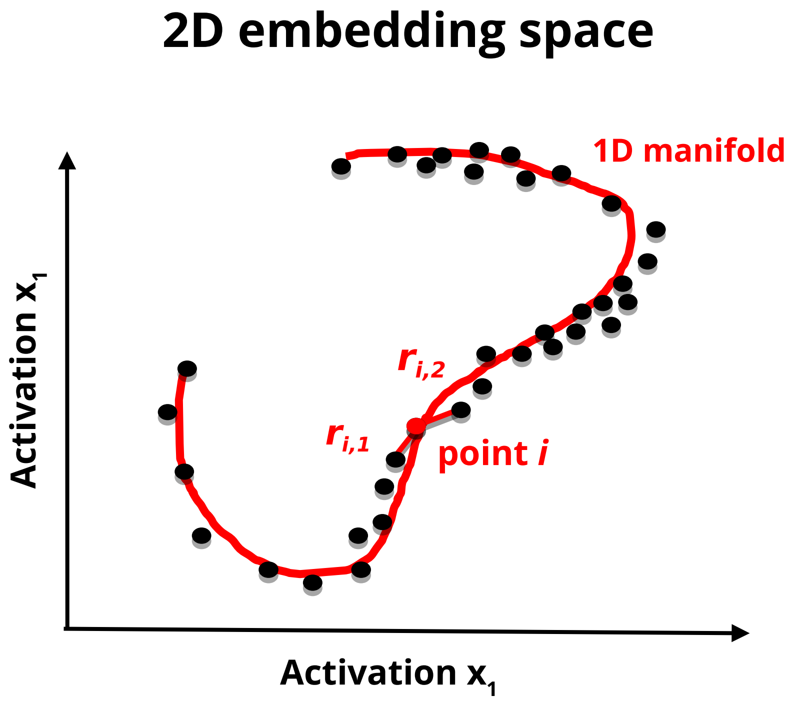
1) For each data point *i* compute the distance to its first and second neighbors ($r_{i,1}$ and $r_{i,2}$)

2D embedding space



Intrinsic dimension estimation

The general (non linear) case: TwoNN (Facco et al, 2017)

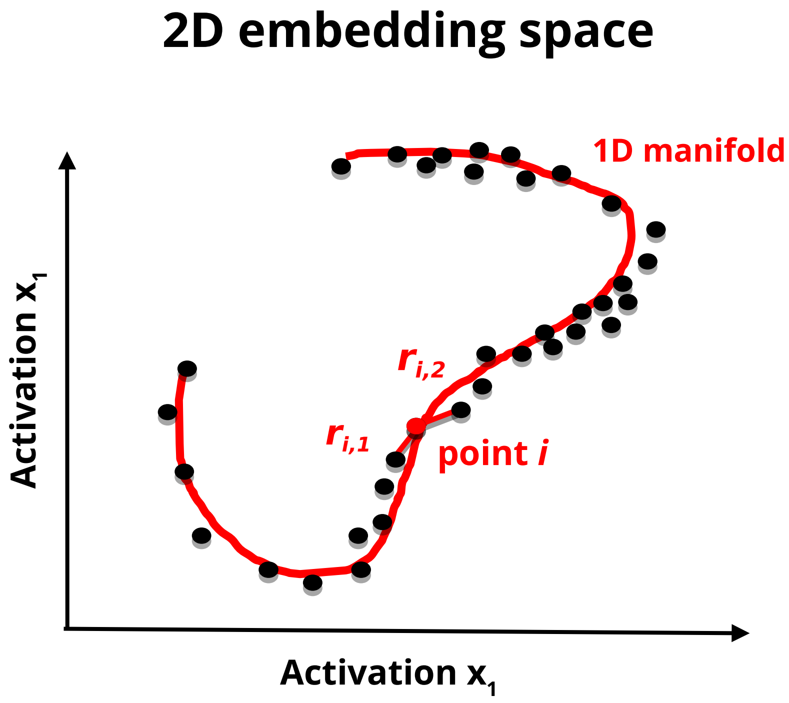


1) For each data point *i* compute the distance to its first and second neighbors ($r_{i,1}$ and $r_{i,2}$)

2) For each *i* compute $\mu_i = \frac{r_{i,2}}{r_{i,1}}$

Intrinsic dimension estimation

The general (non linear) case: TwoNN (Facco et al, 2017)



1) For each data point *i* compute the distance to its first and second neighbors ($r_{i,1}$ and $r_{i,2}$)

2) For each *i* compute $\mu_i = \frac{r_{i,2}}{r_{i,1}}$

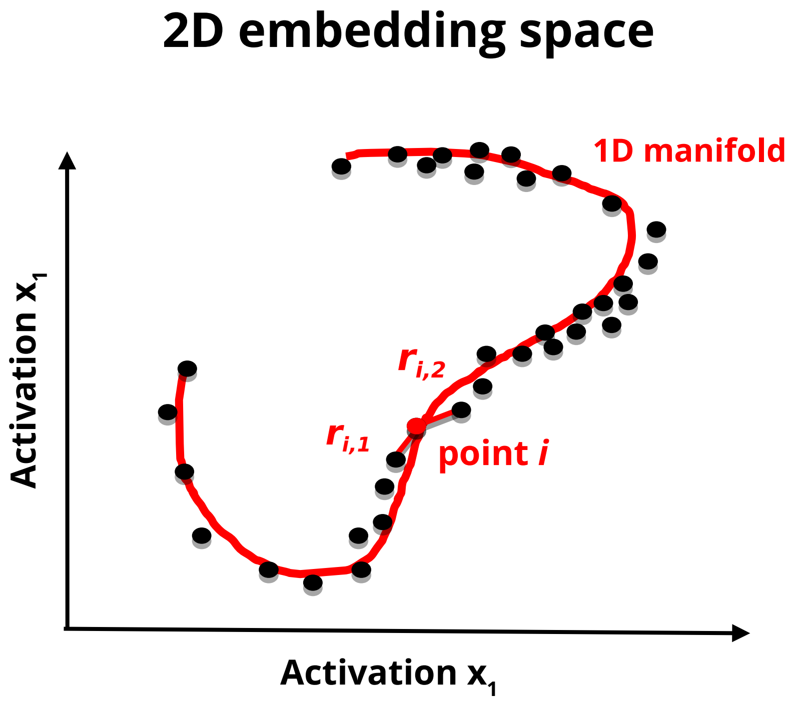
The probability distribution of μ is

$$p(\mu|d) = \frac{d}{\mu^{d+1}}$$

where *d* is the ID.

Intrinsic dimension estimation

The general (non linear) case: TwoNN (Facco et al, 2017)



1) For each data point *i* compute the distance to its first and second neighbors ($r_{i,1}$ and $r_{i,2}$)

2) For each *i* compute $\mu_i = \frac{r_{i,2}}{r_{i,1}}$

The probability distribution of μ is

$$p(\mu|d) = \frac{d}{\mu^{d+1}}$$

where *d* is the ID.

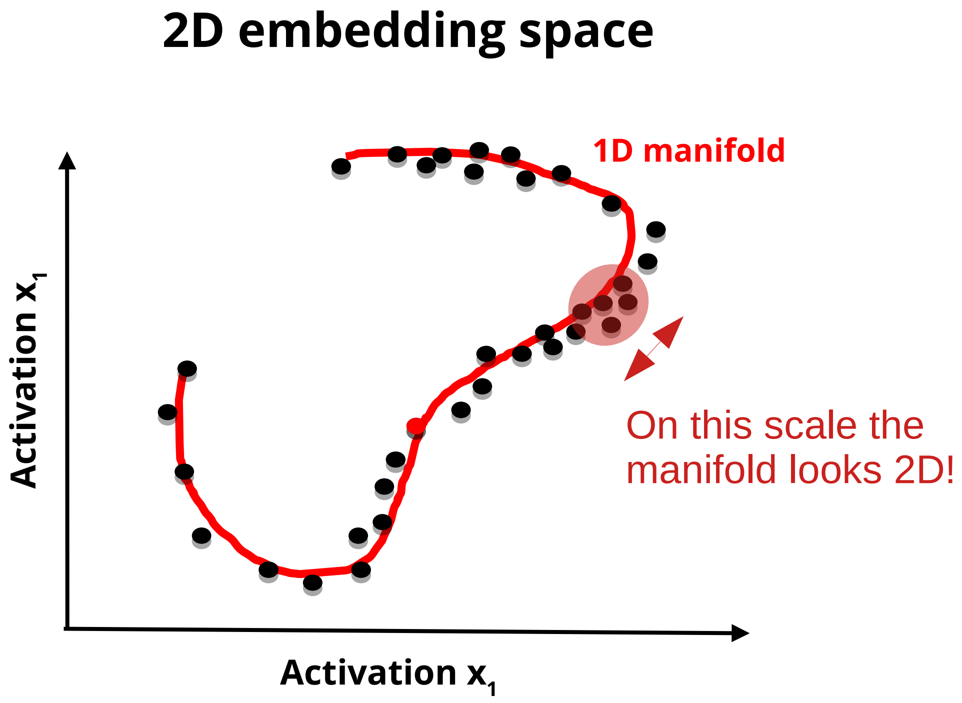
3) Infer *d* e.g via maximum likelihood

$$L(\mu_i|d) = \log \prod^{N} p(\mu_i|d)$$

$$\partial_d L(\mu_i|d) = 0 \rightarrow \hat{d} = \frac{N}{\sum \log \mu_i}$$

Intrinsic dimension estimation of a noisy manifold

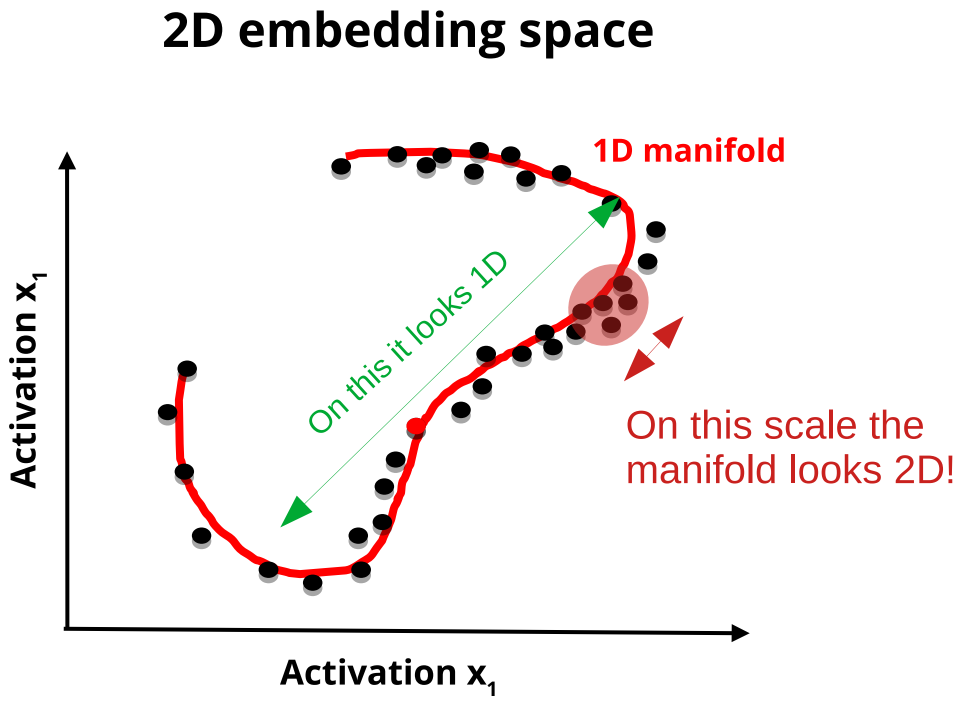
When the data are noisy TwoNN can overestimate the ID due to its **local** nature



Intrinsic dimension estimation of a noisy manifold

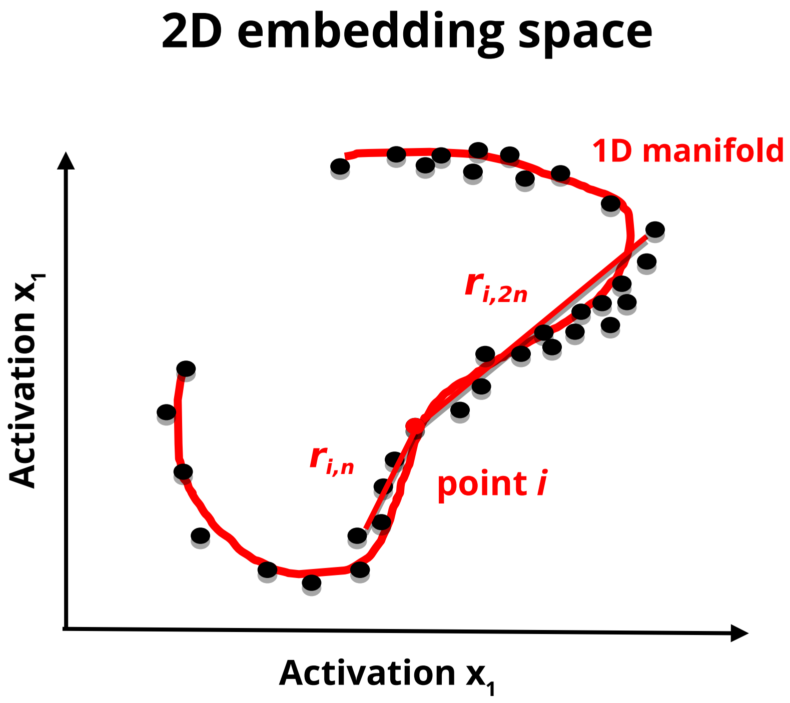
When the data are noisy TwoNN can overestimate the ID due to its **local** nature

Enlarge the neighborhood range to find the actual ‘soft directions’ of the data



Intrinsic dimension estimation of a noisy manifold

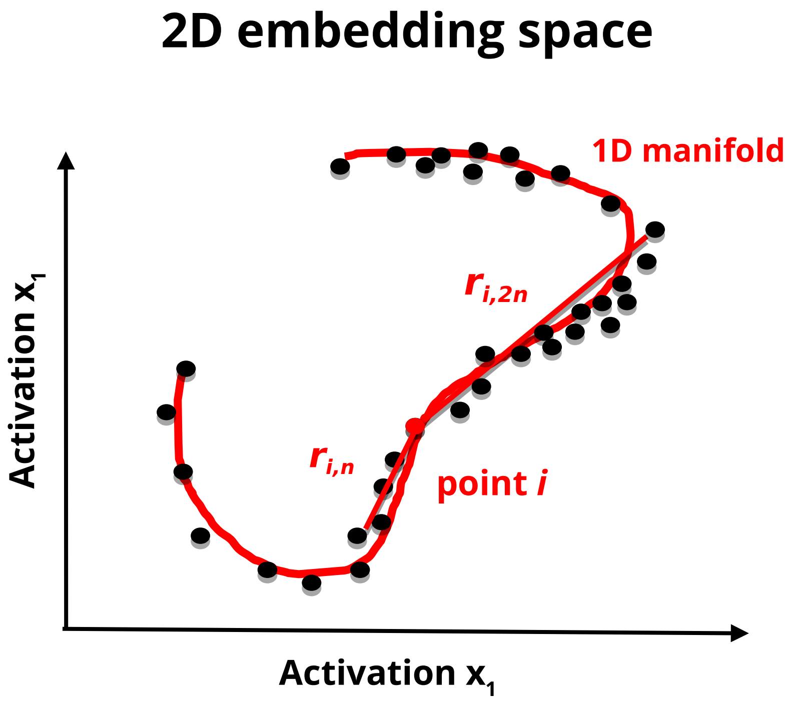
When the data are noisy TwoNN can overestimate the ID due to its **local** nature



1) For each data point *i* compute the distance to its **nth** and **2*nth** neighbors ($r_{i,n}$ and $r_{i,2n}$)

Intrinsic dimension estimation of a noisy manifold

When the data are noisy TwoNN can overestimate the ID due to its **local** nature

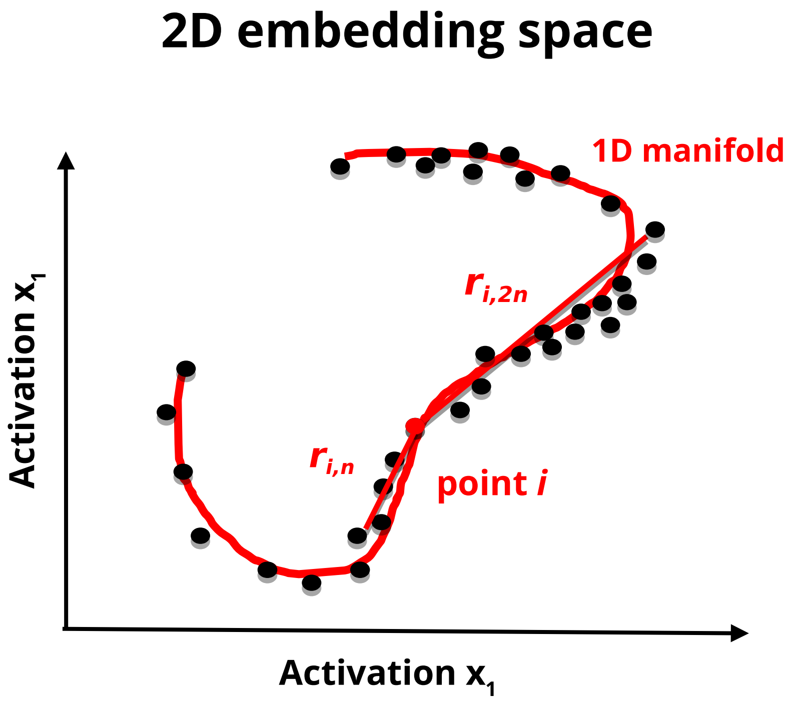


1) For each data point *i* compute the distance to its **nth** and **2*nth** neighbors ($r_{i,n}$ and $r_{i,2n}$)

2) For each *i* compute $$\mu_{n,i} = \frac{r_{i,2n}}{r_{i,n}}$$

Intrinsic dimension estimation of a noisy manifold

When the data are noisy TwoNN can overestimate the ID due to its **local** nature



1) For each data point *i* compute the distance to its **nth** and **2*nth** neighbors ($r_{i,n}$ and $r_{i,2n}$)

2) For each *i* compute $\mu_{n,i} = \frac{r_{i,2n}}{r_{i,n}}$

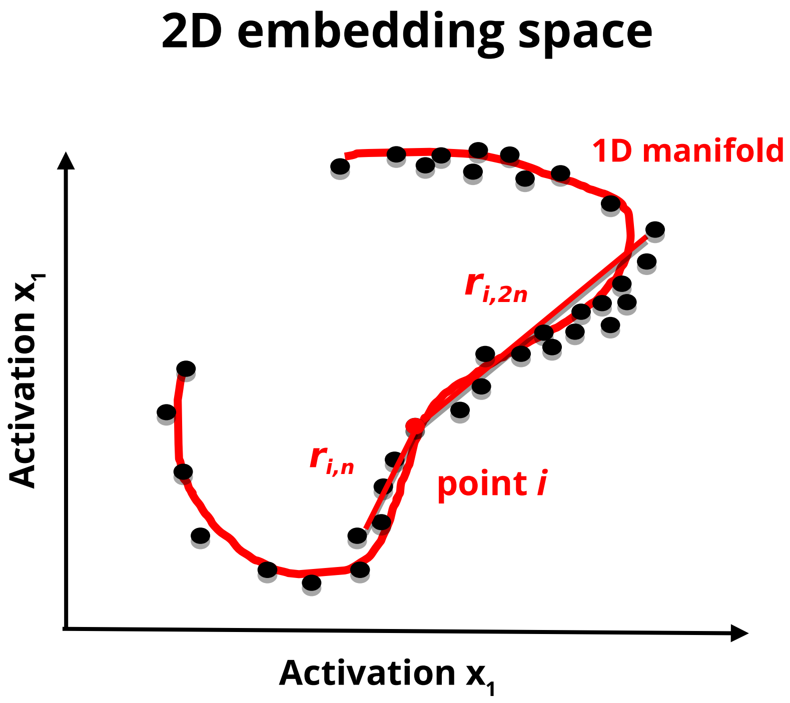
The probability distribution of μ_n is

$$p(\mu_n|d) \propto \frac{d(\mu^d - 1)^{n-1}}{\mu^{(2n-1)d+1}}$$

where *d* is the ID.

Intrinsic dimension estimation of a noisy manifold

When the data are noisy TwoNN can overestimate the ID due to its **local** nature



1) For each data point *i* compute the distance to its **nth** and **2*nth** neighbors ($r_{i,n}$ and $r_{i,2n}$)

2) For each *i* compute $\mu_{n,i} = \frac{r_{i,2n}}{r_{i,n}}$

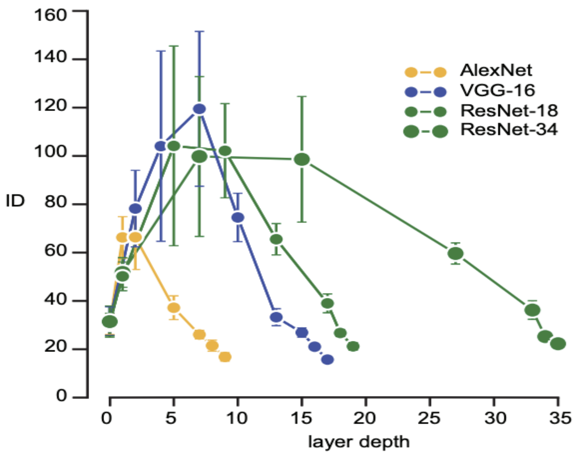
The probability distribution of μ_n is

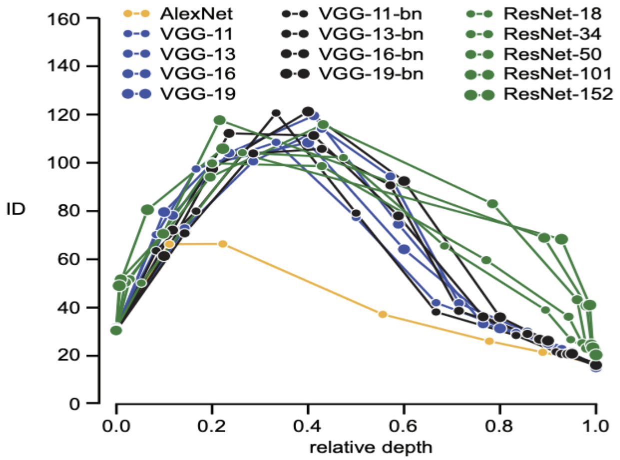
$$p(\mu_n|d) \propto \frac{d(\mu^d - 1)^{n-1}}{\mu^{(2n-1)d+1}}$$

where *d* is the ID.

3) Infer *d* via maximum likelihood

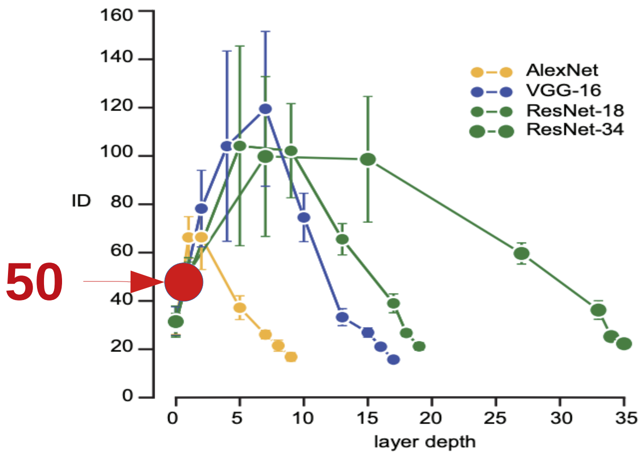
Expansion and compression of the ID

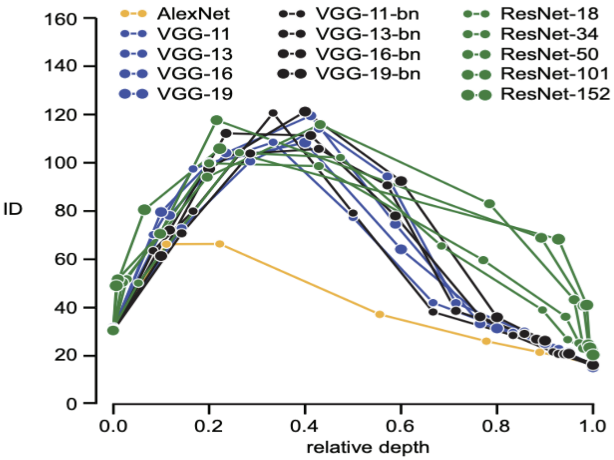


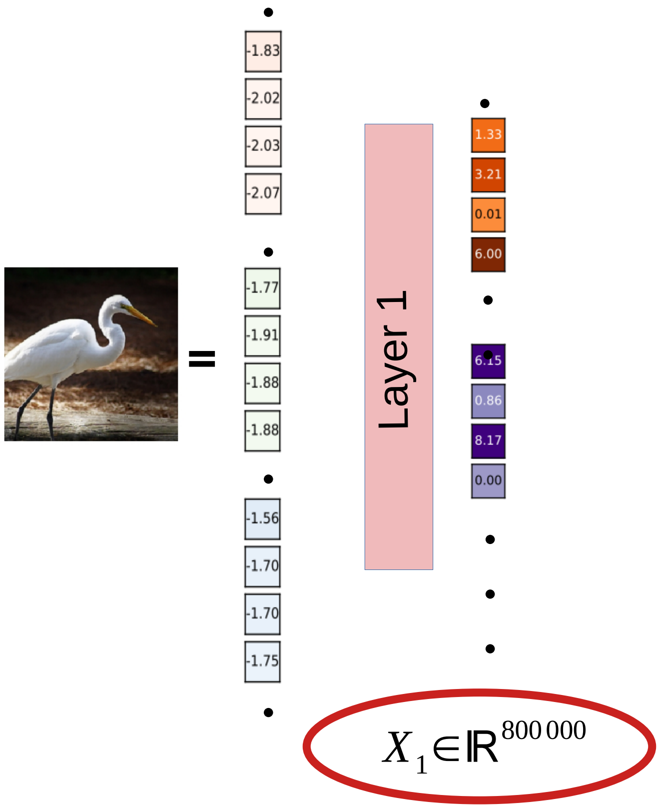


The ID is always much smaller than the embedding dimension

Expansion and compression of the ID

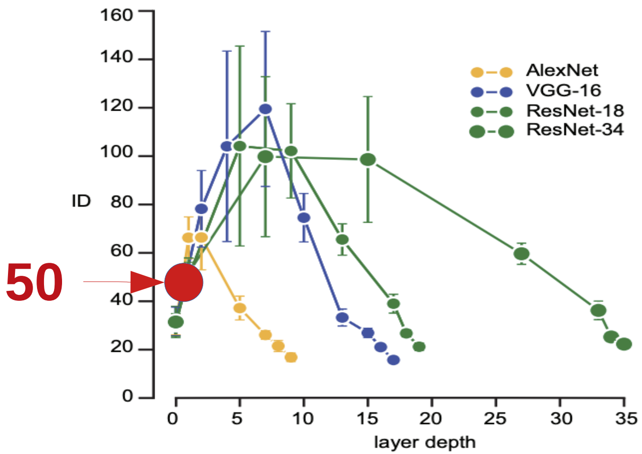


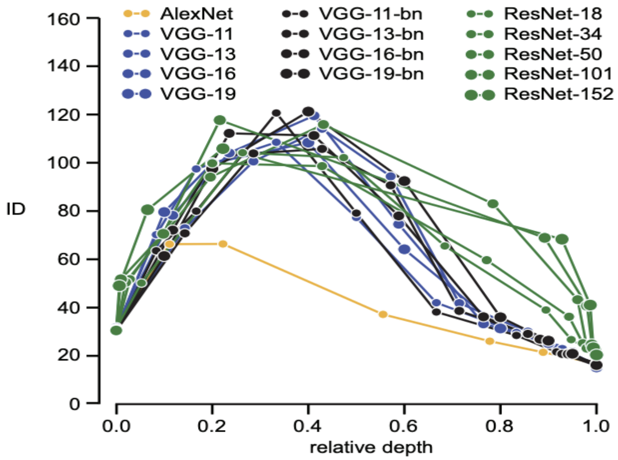


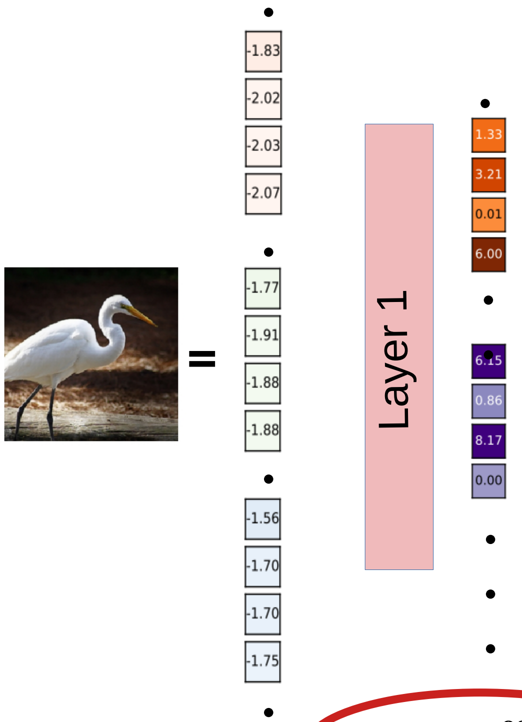


The ID is always much smaller than the embedding dimension

Expansion and compression of the ID







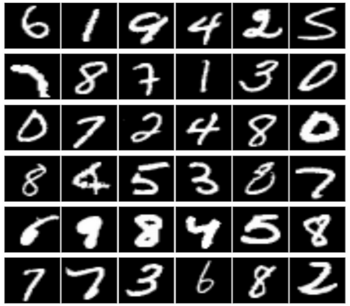
$X_1 \in \mathbb{R}^{800\,000}$

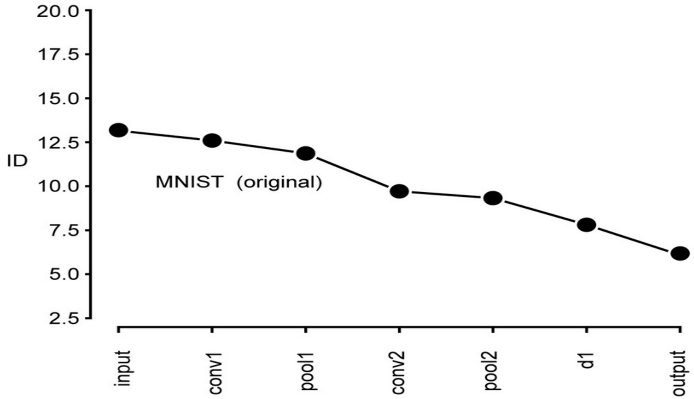
The ID is always much smaller than the embedding dimension

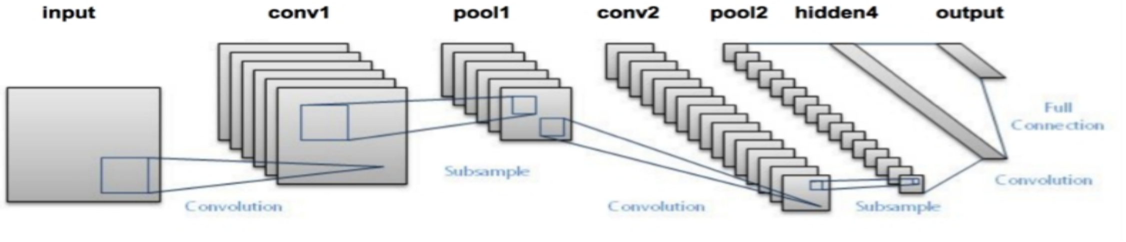
ID evolution across layer has a hunchback shape

Discarding useless features

MNIST (original)

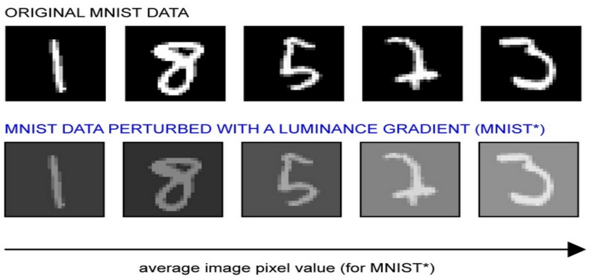


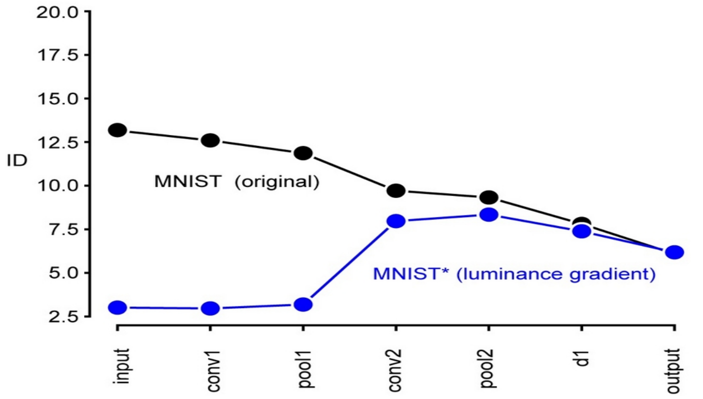


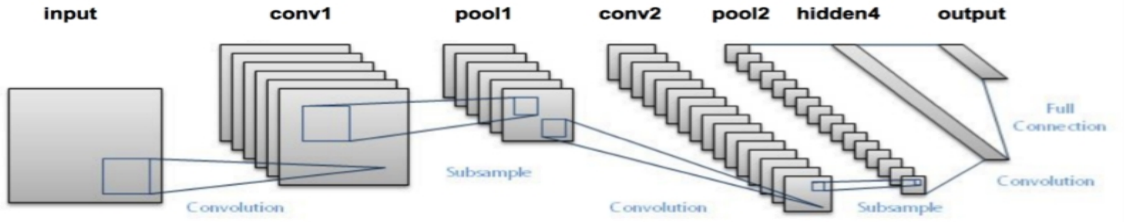


Discarding useless features

MNIST* (luminance gradient)







In a trained network, the initial ID expansion reflects the pruning of low-level visual features that carry no information about the correct labeling