Introduction to Deep Learning

Alvaro Soto

Computer Science Department, PUC

Deep Learning

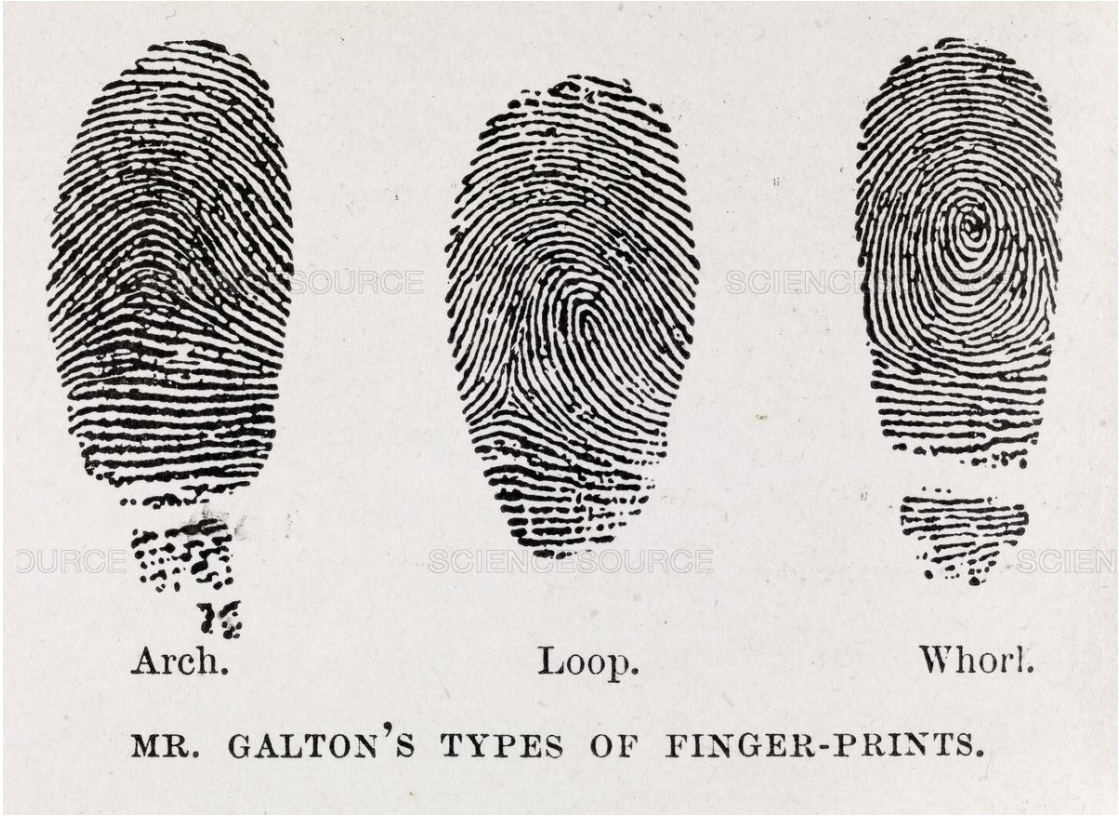
Two main ingredients

1. Feature learning
2. A hierarchical compositional representation

Ingredient 1: Feature learning

Any relevant feature to recognize each individual ?

Any relevant feature to recognize each individual ?



Any relevant feature to recognize each individual ?

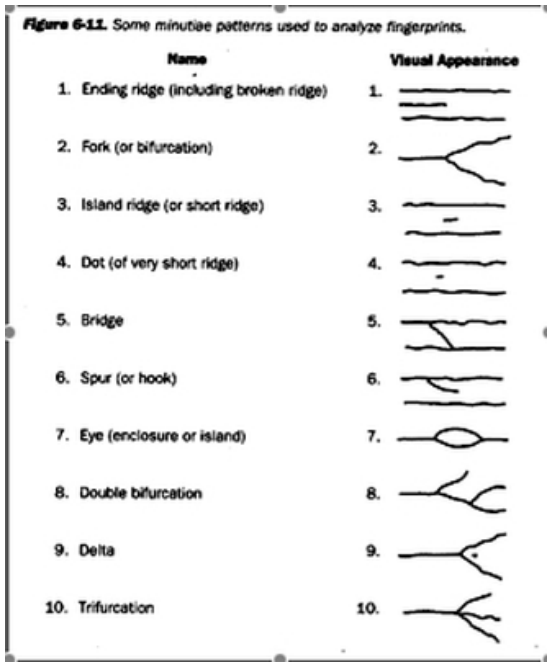
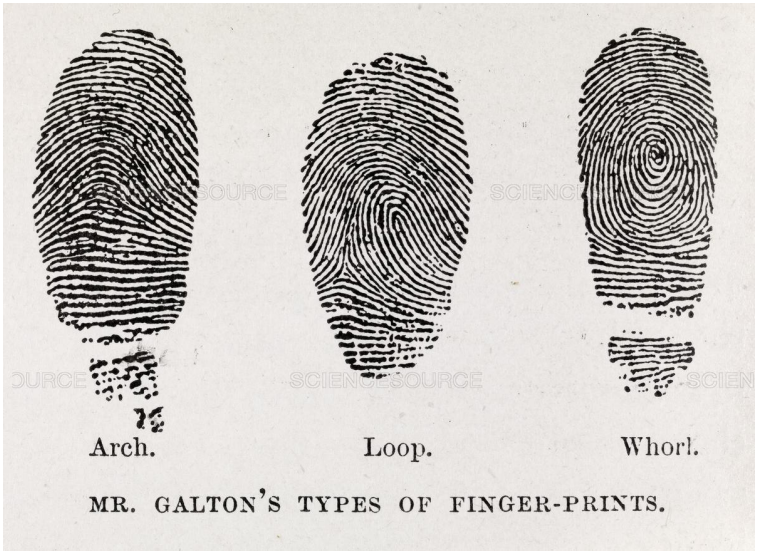


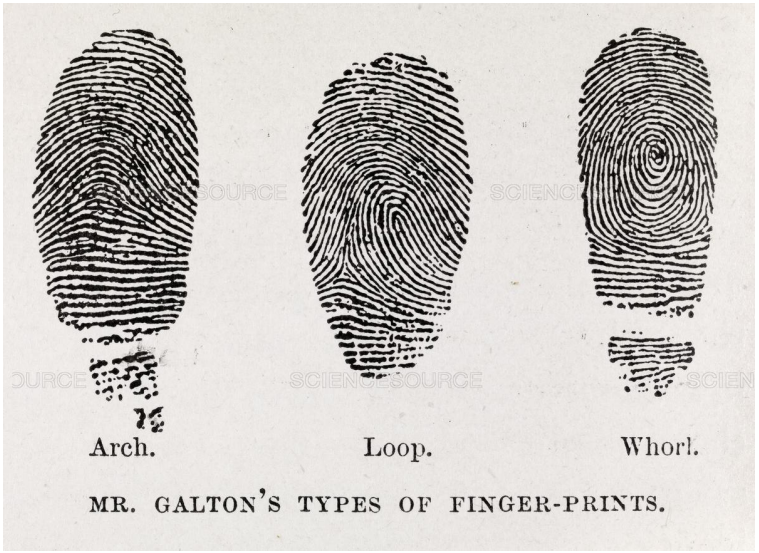
Figure 6-11. *Some minutiae patterns used to analyze fingerprints.*

Name	Visual Appearance
1. Ending ridge (including broken ridge)	1.
2. Fork (or bifurcation)	2.
3. Island ridge (or short ridge)	3.
4. Dot (of very short ridge)	4.
5. Bridge	5.
6. Spur (or hook)	6.
7. Eye (enclosure or island)	7.
8. Double bifurcation	8.
9. Delta	9.
10. Trifurcation	10.

Any relevant feature to recognize each individual ?



Any relevant feature to recognize each individual ?



Good features are key

Feature based machine learning

Handcrafted Features

Feature based machine learning

Handcrafted Features

Ex. Bag of Word Representation

d1 : Mary loves Movies, Cinema and Art
d2 : John went to the Football game

d3 : Robert went for the Movie Delicatessen

Class 1 : Arts
Class 2 : Sports

Class : Arts

	Mary	Loves	Movies	Cinema	Art	John	Went	to	the	Delicatessen	Robert	Football	Game	and	for
d1	1	1	1	1	1									1	
d2						1	1	1	1			1	1		
d3			1				1		1		1				1

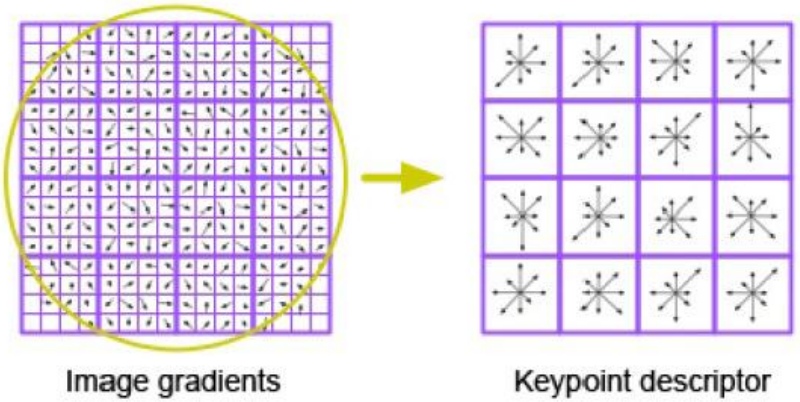
Feature based machine learning

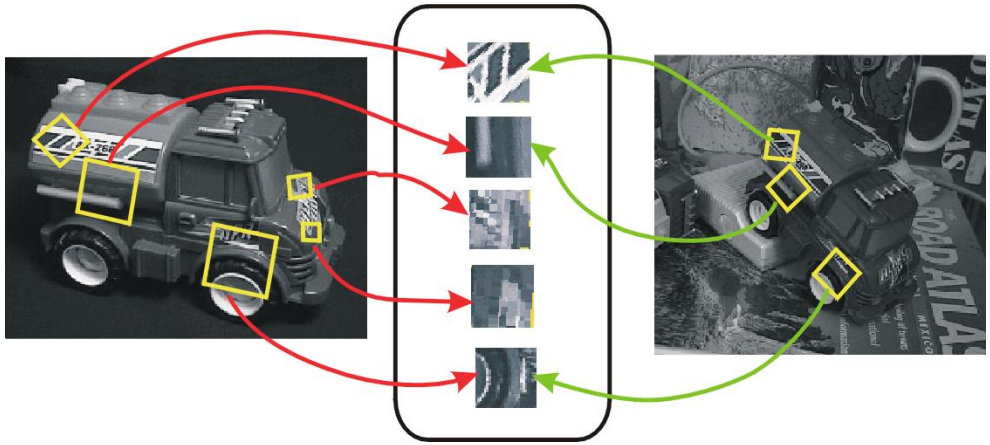
Handcrafted Features

Feature based machine learning

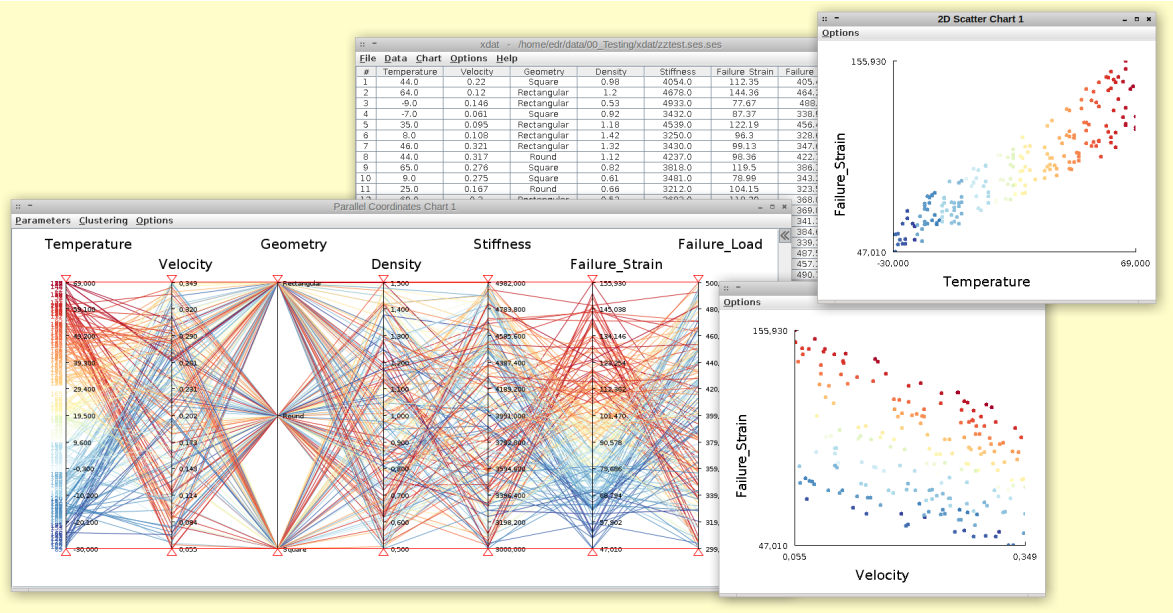
Handcrafted Features

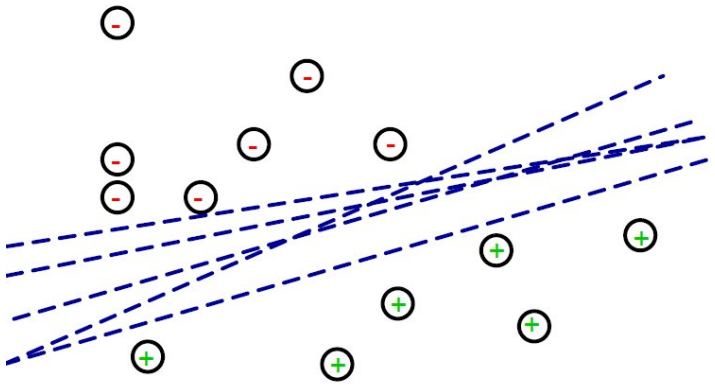
Ex. SIFT Based Representation





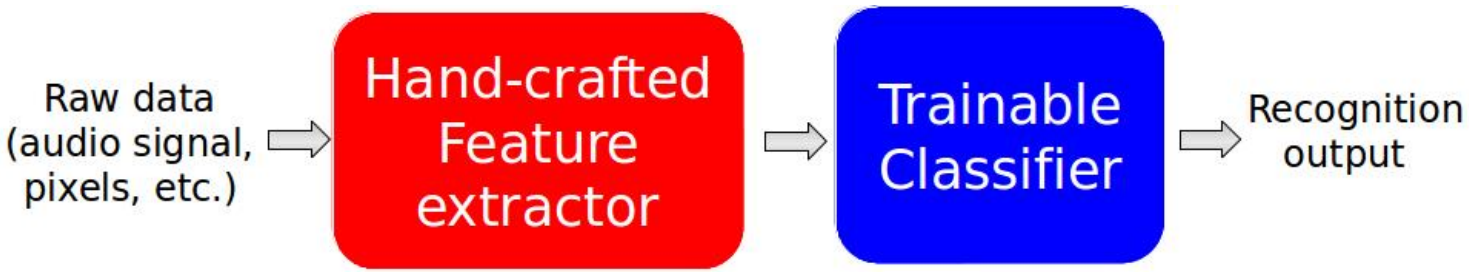
Feature space



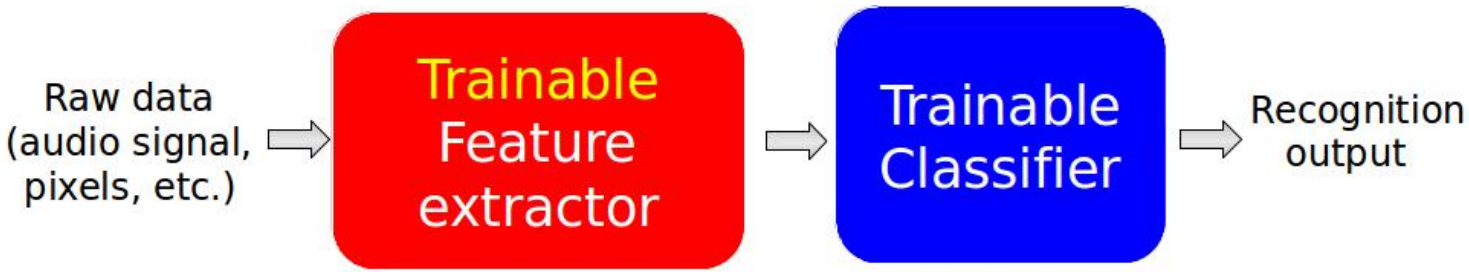


From Handcrafted to Learned Features

- Traditional pattern recognition approach since the 50's:



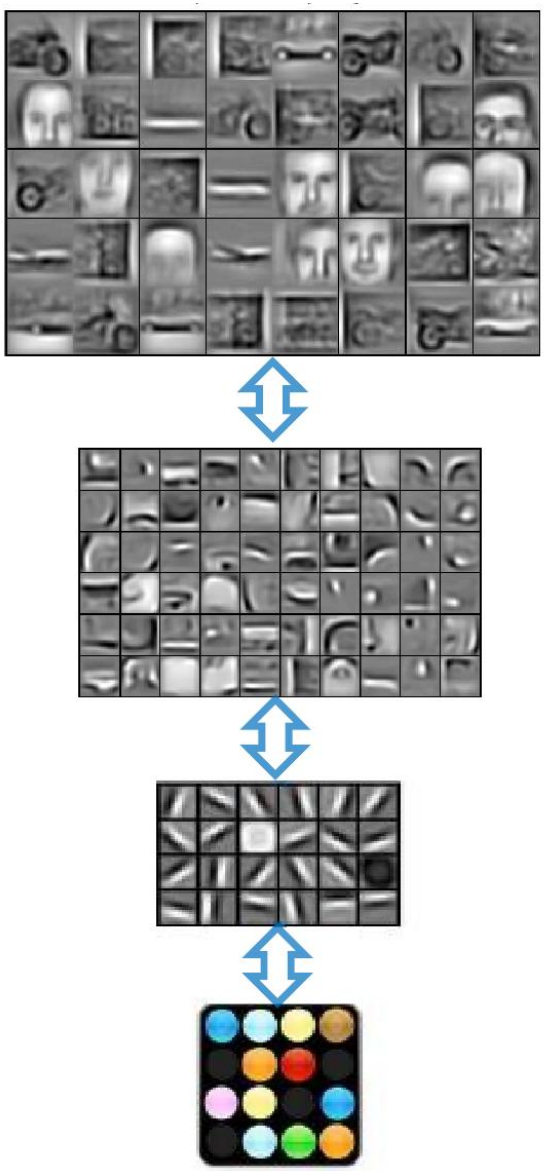
- Feature learning approach:



Ingred. 2: A hierarchical compositional repres.

Hierarchical and Composable Feat Represent.

- Hierarchical representations organized from low to high levels of abstraction are common in nature.
- Compositionality is key. Language is compositional, vision is compositional, etc.
- Examples:
  - Sounds $\rightarrow$ phonemes $\rightarrow$ syllables $\rightarrow$ words.
  - Pixels $\rightarrow$ edges $\rightarrow$ parts $\rightarrow$ objects
  - Characters $\rightarrow$ words $\rightarrow$ word groups $\rightarrow$ clauses $\rightarrow$ sentences.
  - Nucleotides $\rightarrow$ genes $\rightarrow$ proteins $\rightarrow$ cells $\rightarrow$ organs.



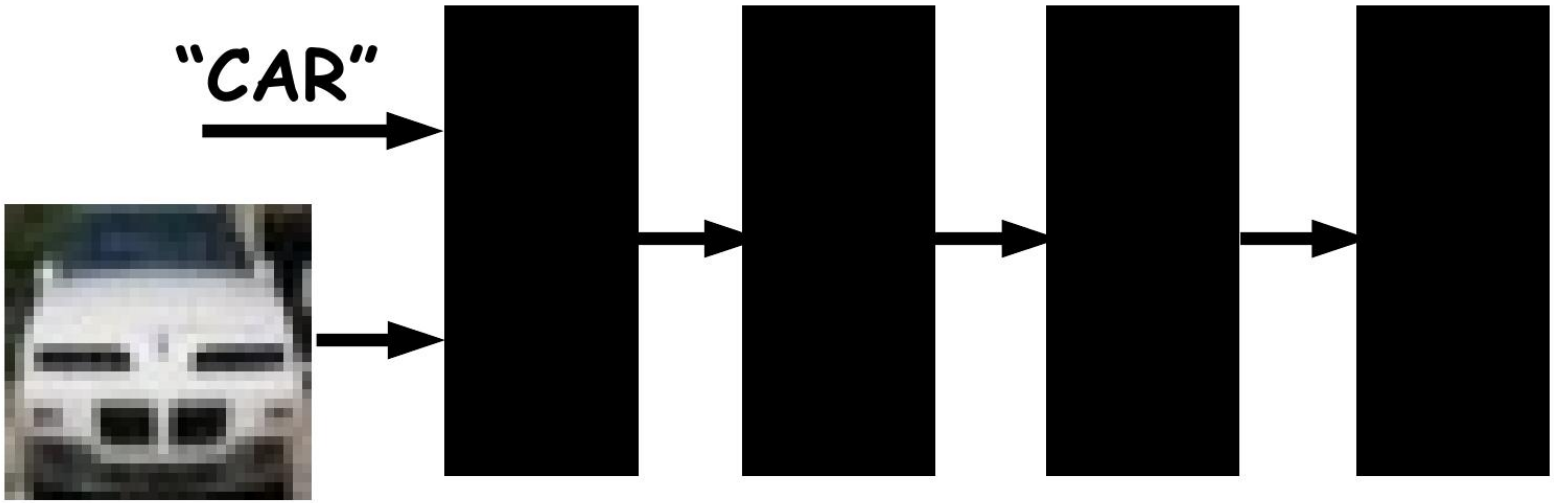
3rd layer
“Objects”

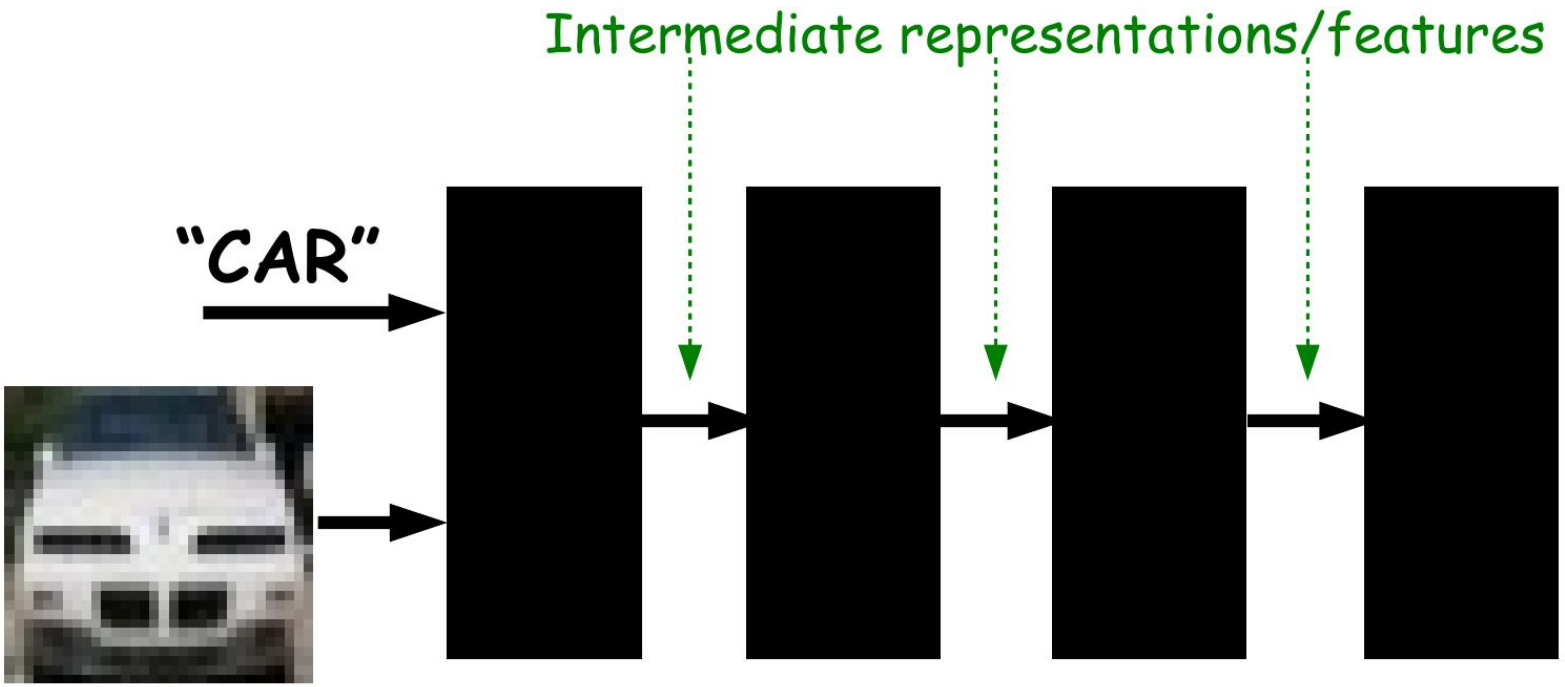
2nd layer
“Object parts”

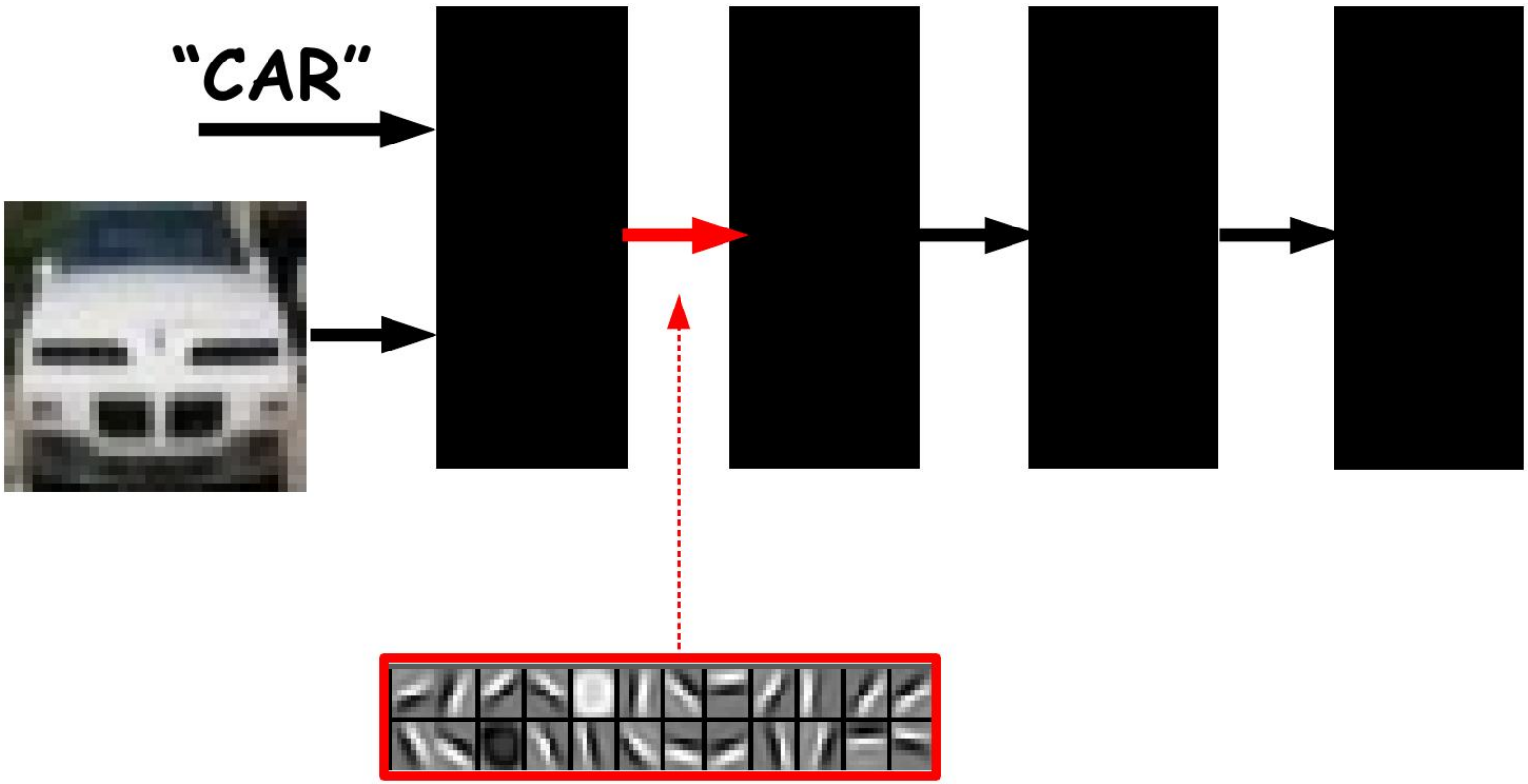
1st layer
“edges”

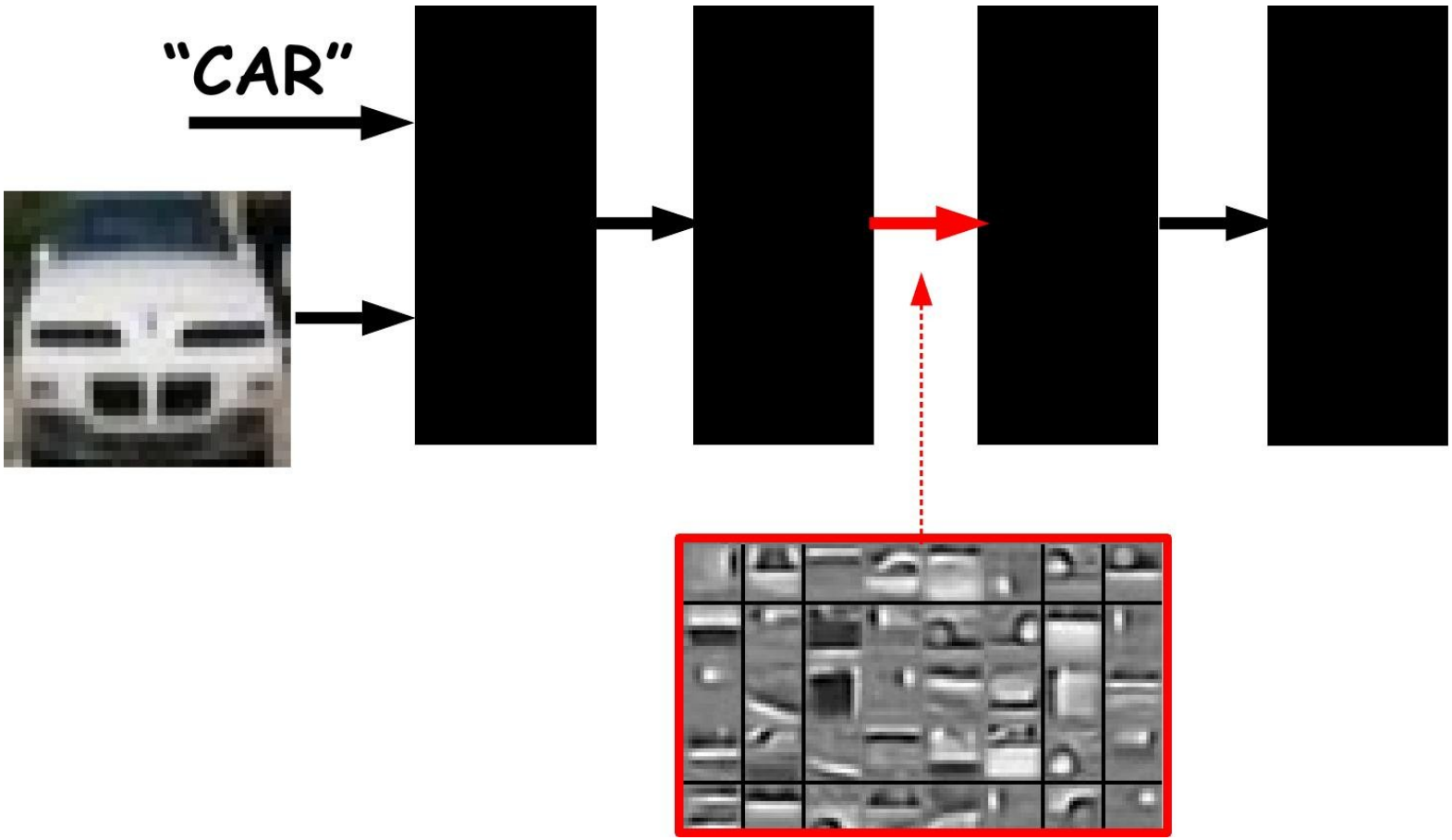
Input

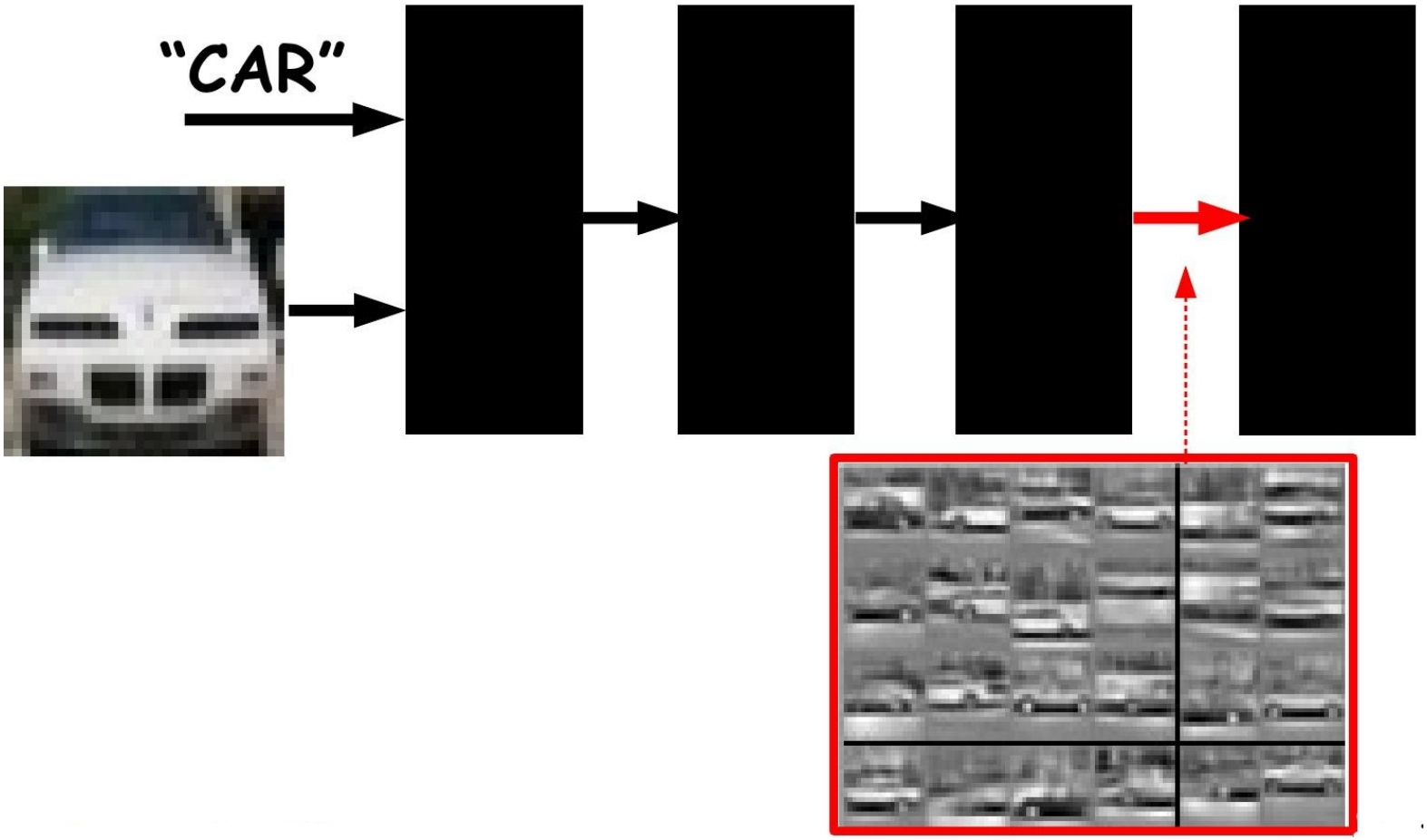

"CAR"


"CAR"

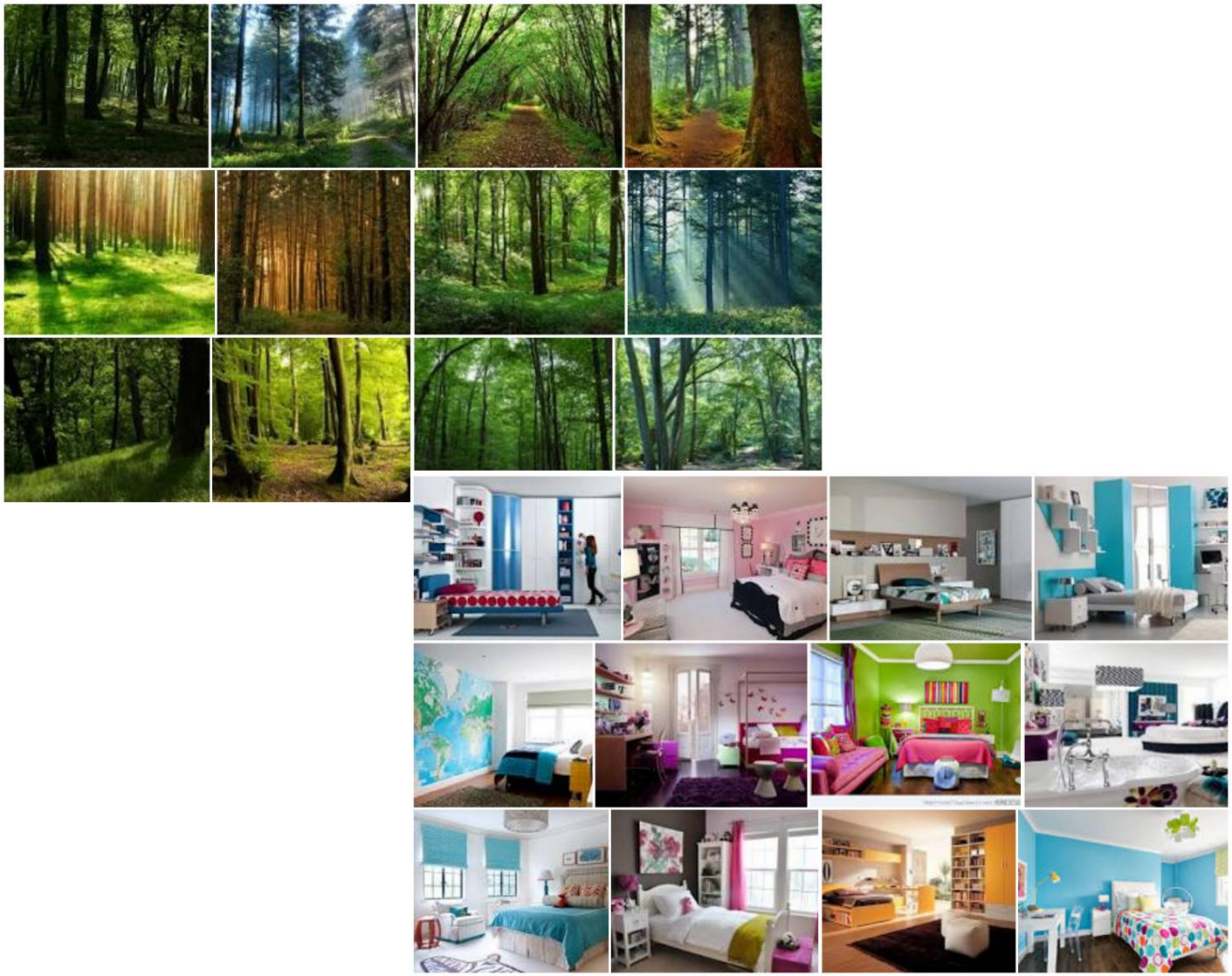

Intermediate representations/features
"CAR"


"CAR"

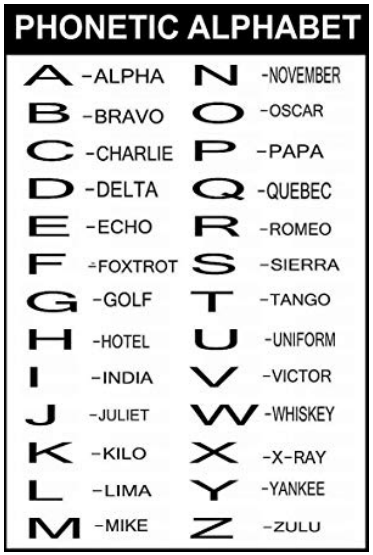

"CAR"


"CAR"

Compositionality is Key: Holistic vs Local Reps.



What is the right level of compositionality?



Aoccdrnig to a rscheearch at an Elingsh uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht frist and lsat ltteer is at the rghit pclae. The rset can be a toatl mses and you can sitll raed it wouthit porbelm. Tihs is bcuseae we do not raed ervey lteter by it slef but the wrod as a wlohe

Now our honor guest:

Deep Learning

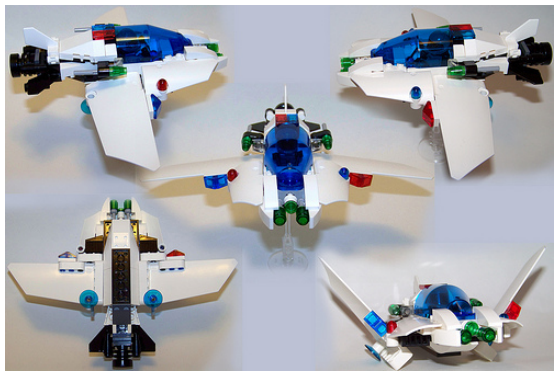
Deep Learning

- Deep learning: Neural network (NN) models that learn hierarchical compositional representations (hierarchies of feats).
- In general, NNs with several hidden layers.
- Today, the most successful learning framework. They hold records for best recognition performance on several difficult tasks, such as, voice, handwritting, and object recognition.
- What is the secret?

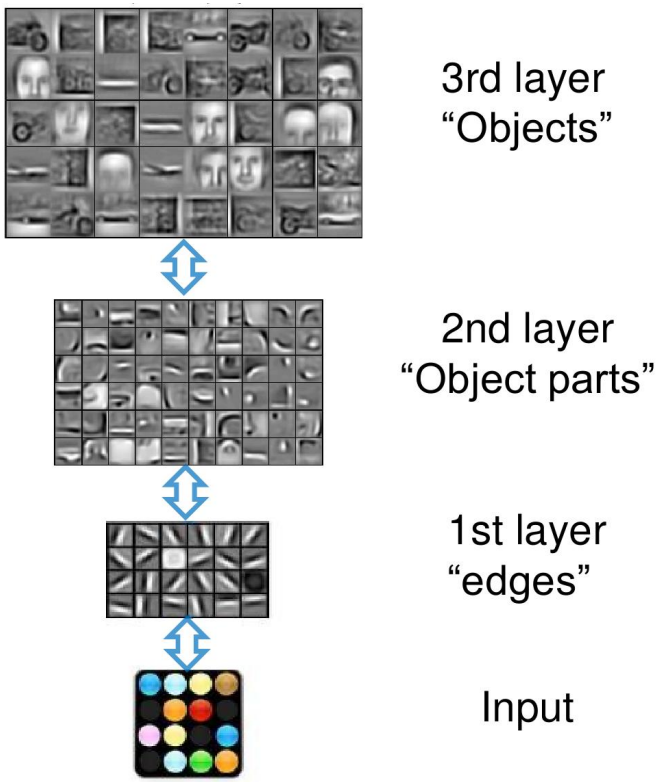
The magic is learning suitable hierarchical and composable feature representations

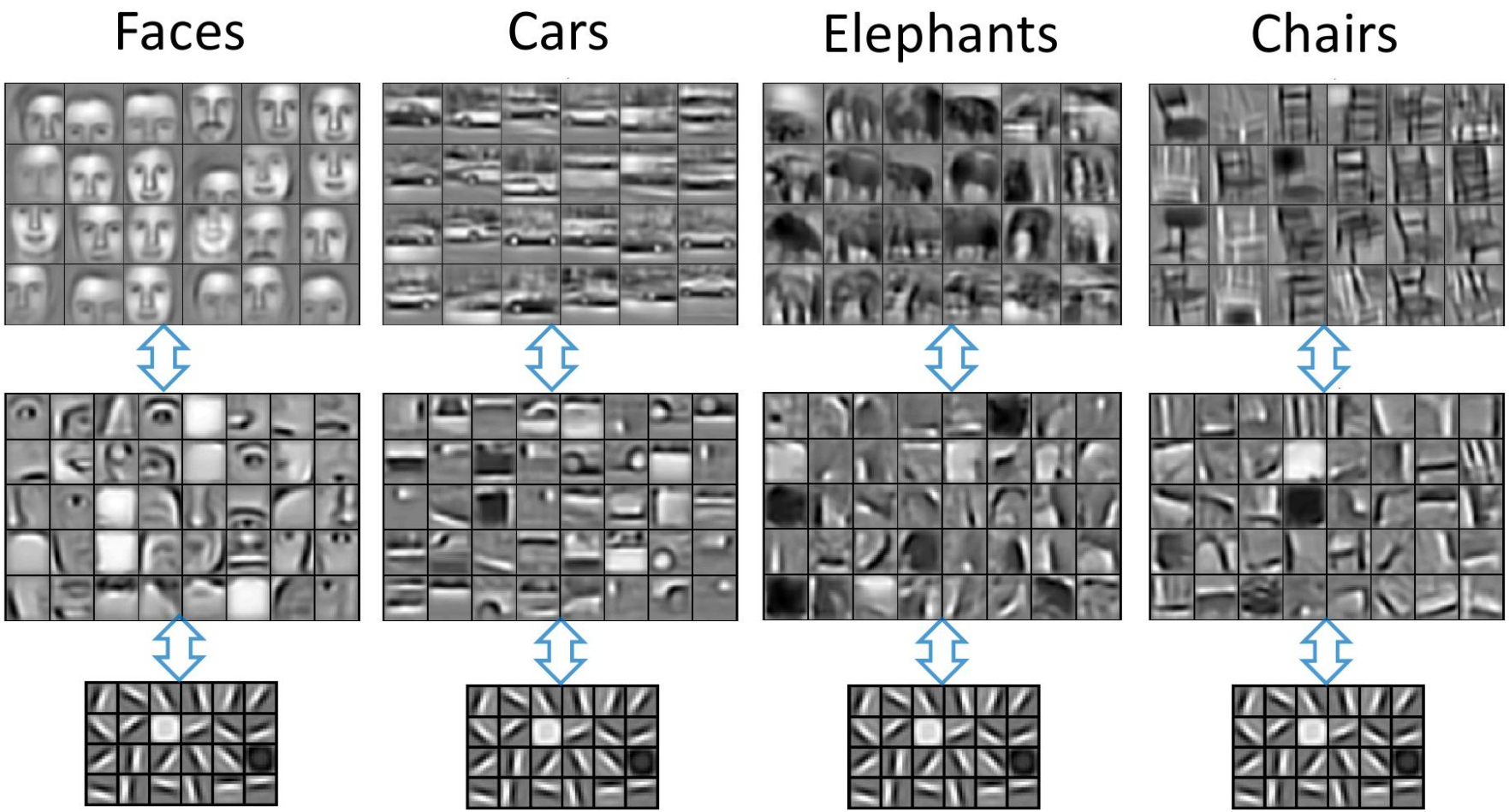
Hierarchical and Composable Feature Reps.

- Lower level layers learn generic and local feature detectors that operate as building blocks (part detectors). Ex.:
  - In the case of speech: first layers learn phonemes detectors.
  - In the case of vision: first layers learn edge (oriented gradients) and center sorround (gabor filters).
- Higher level layers encode specialized and global concept detectors that are more robust to invariances. Ex.:
  - Robust to different speakers in the case of voice recognition.
  - Robust to different views in the case of visual recognition.



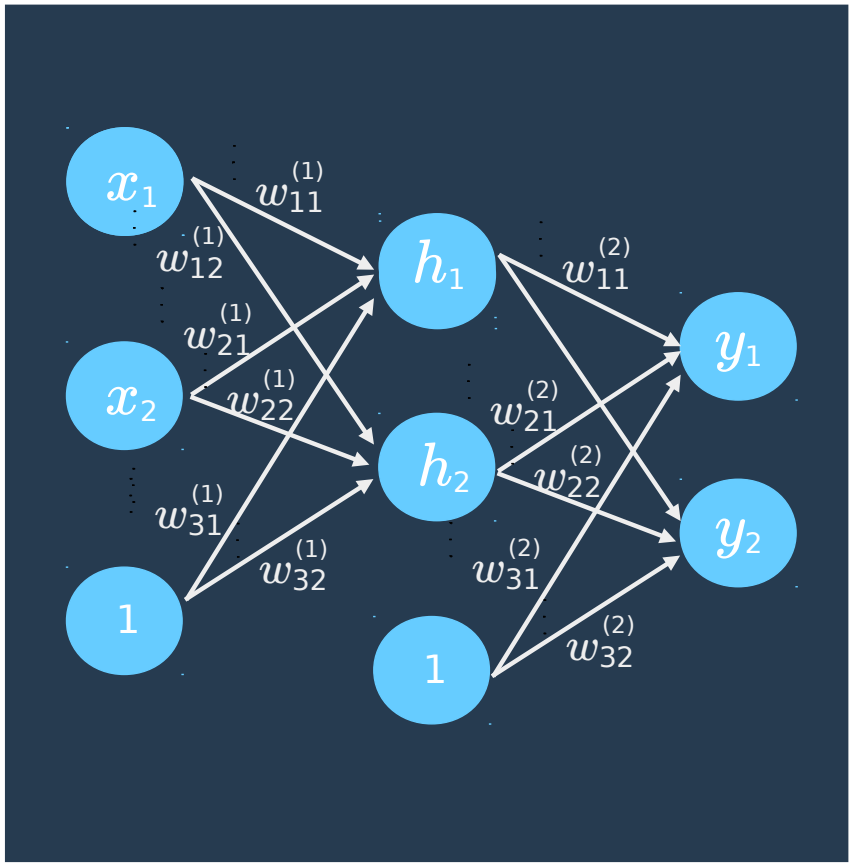
- Each module transforms its input representation into a higher-level of abstraction.
- High-level features are more global and more invariant.
- Low-level features are shared among categories.





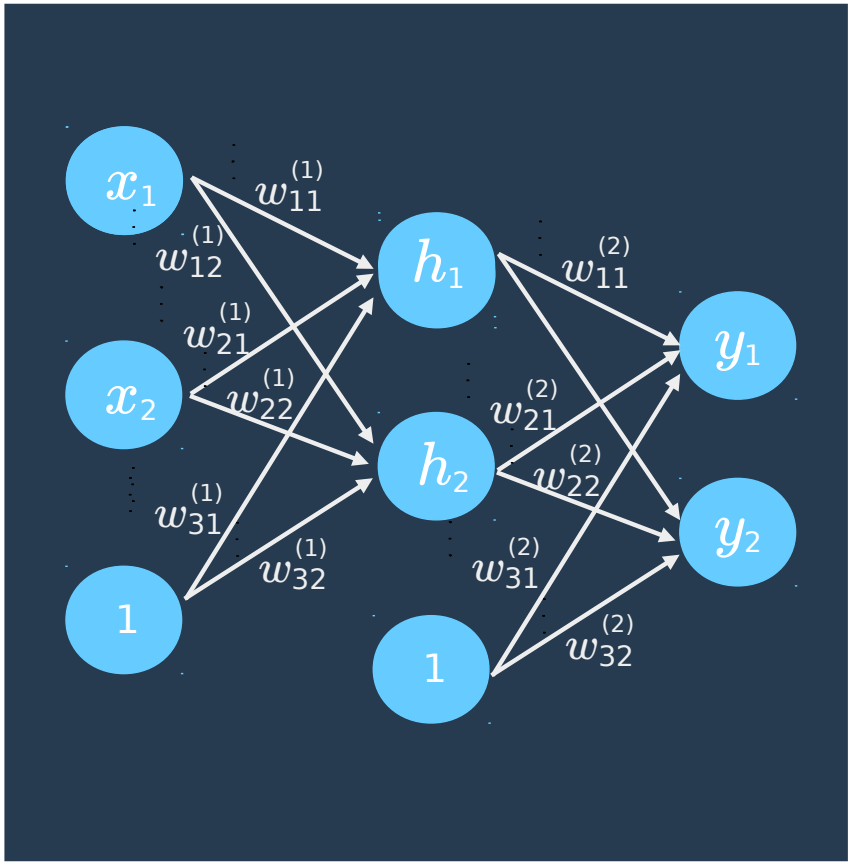
Where should we have more "features", at lower or higher level of the hierarchy?

Theoretician's dilemma: Why Deep?



Theory shows that a feedforward NW with 1 hidden layer can represent most input-output relations.

Let's introduce some notation



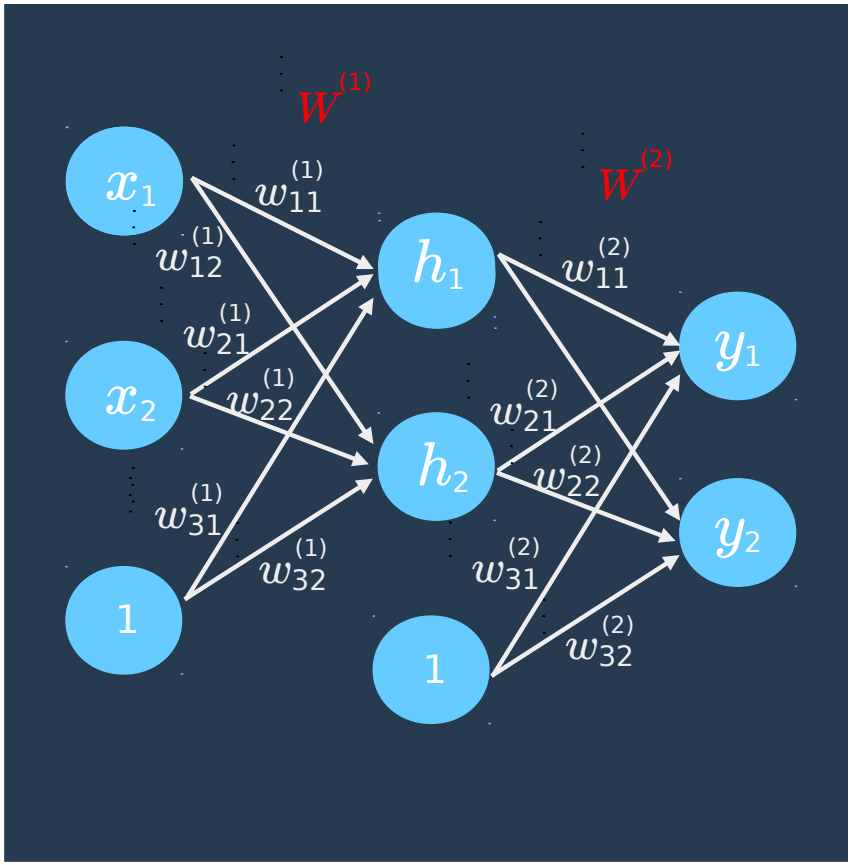
$$\begin{pmatrix} h_1 \\ h_2 \end{pmatrix} = \sigma \begin{pmatrix} w_{11}^{(1)} & w_{21}^{(1)} & w_{31}^{(1)} \\ w_{12}^{(1)} & w_{22}^{(1)} & w_{32}^{(1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} h_1 \\ h_2 \end{pmatrix} = \sigma \left(W^{(1)} X \right)$$

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \sigma \begin{pmatrix} w_{11}^{(2)} & w_{21}^{(2)} & w_{31}^{(2)} \\ w_{12}^{(2)} & w_{22}^{(2)} & w_{32}^{(2)} \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \sigma \left(W^{(2)} H \right)$$

Let's introduce some notation



$$\begin{pmatrix} h_1 \\ h_2 \end{pmatrix} = \sigma \begin{pmatrix} w_{11}^{(1)} & w_{21}^{(1)} & w_{31}^{(1)} \\ w_{12}^{(1)} & w_{22}^{(1)} & w_{32}^{(1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} h_1 \\ h_2 \end{pmatrix} = \sigma \left(W^{(1)} X \right)$$

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \sigma \begin{pmatrix} w_{11}^{(2)} & w_{21}^{(2)} & w_{31}^{(2)} \\ w_{12}^{(2)} & w_{22}^{(2)} & w_{32}^{(2)} \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \sigma \left(W^{(2)} H \right)$$

$$Y = \sigma \left(W^{(2)} \sigma \left(W^{(1)} \vec{X} \right) \right)$$

Theoretician's dilemma: Why Deep?

$$Y = \sigma\left(W^{(2)}\sigma\left(W^{(1)}\vec{X}\right)\right)$$

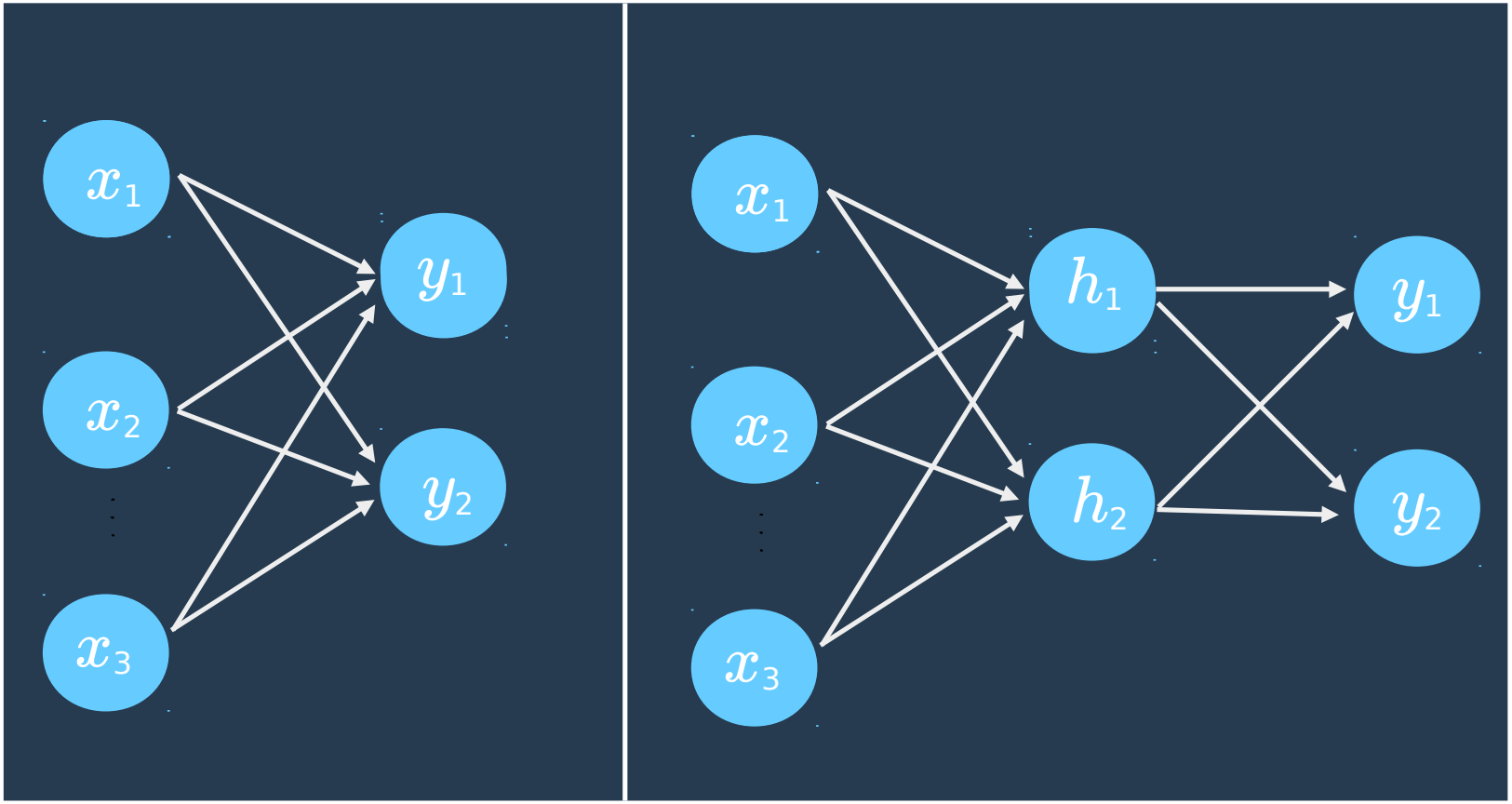
V/S

$$Y = \sigma\left(W^{(K)}\sigma\left(\ldots\ \sigma\left(W^{(2)}\sigma\left(W^{(1)}\vec{X}\right)\right)\ldots\right)\right)$$

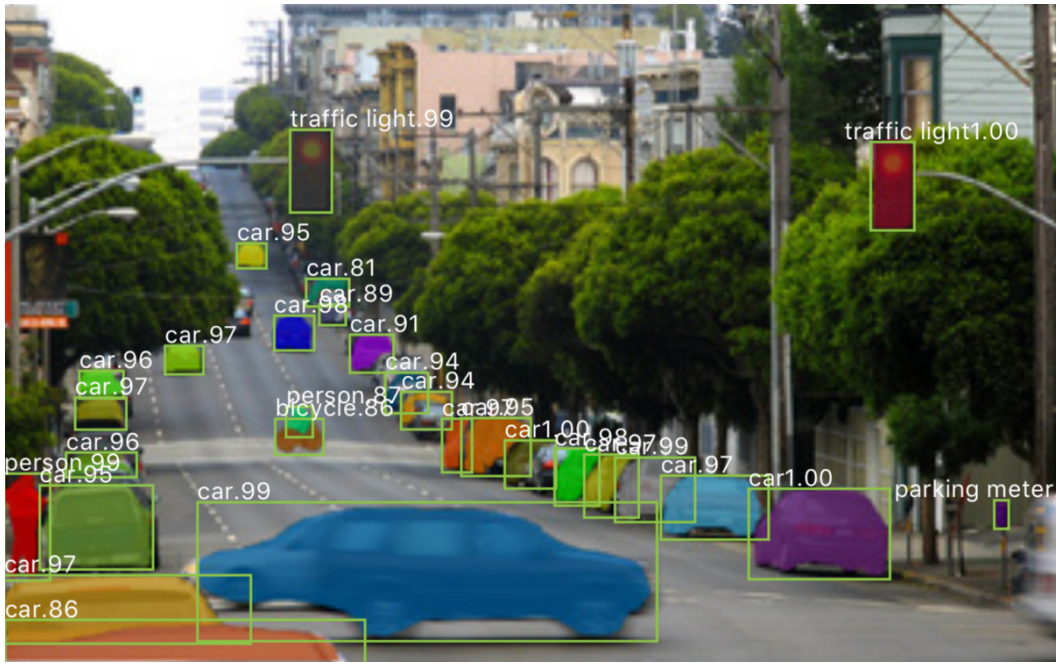
Solution: Learning efficiency

A hierachical compositional architecture provides an efficient learning framework, where lower level layers are shared by higher level layers.

A shared low level representation



DL in Action



Input sentence:	*Translation (PBMT):*	*Translation (GNMT):*	*Translation (human):*
李克強此行將啟動中加總理年度對話機制，與加拿大總理杜魯多舉行兩國總理首次年度對話。	Li Keqiang premier added this line to start the annual dialogue mechanism with the Canadian Prime Minister Trudeau two prime ministers held its first annual session.	Li Keqiang will start the annual dialogue mechanism with Prime Minister Trudeau of Canada and hold the first annual dialogue between the two premiers.	Li Keqiang will initiate the annual dialogue mechanism between premiers of China and Canada during this visit, and hold the first annual dialogue with Premier Trudeau of Canada.

DL is Awesome: Example GPT3

```
Q: What is your favorite animal?
A: My favorite animal is a dog.

Q: Why?
A: Because dogs are loyal and friendly.

Q: What are two reasons that a dog might be in a bad mood?
A: Two reasons that a dog might be in a bad mood are if it is hungry or if it is hot.

Q: How many eyes does a giraffe have?
A: A giraffe has two eyes.

Q: How many legs does a frog have?
A: A frog has four legs.

Q: Are there any animals with three legs?
A: No, there are no animals with three legs.

Q: Why don't animals have three legs?
A: Animals don't have three legs because they would fall over.
```

Challenges

Model fitting

- How to adjust millions of parameters without overfitting.
  - We require lot of training data, even worse: labeled data.
- Tons of labeled data, How?
  - Use a creative, cheap, and/or fast ways to label data.
  - Apply training tricks (add noise, synthetic data, etc.).
  - Transfer learning.
  - Self-supervision.

Challenges

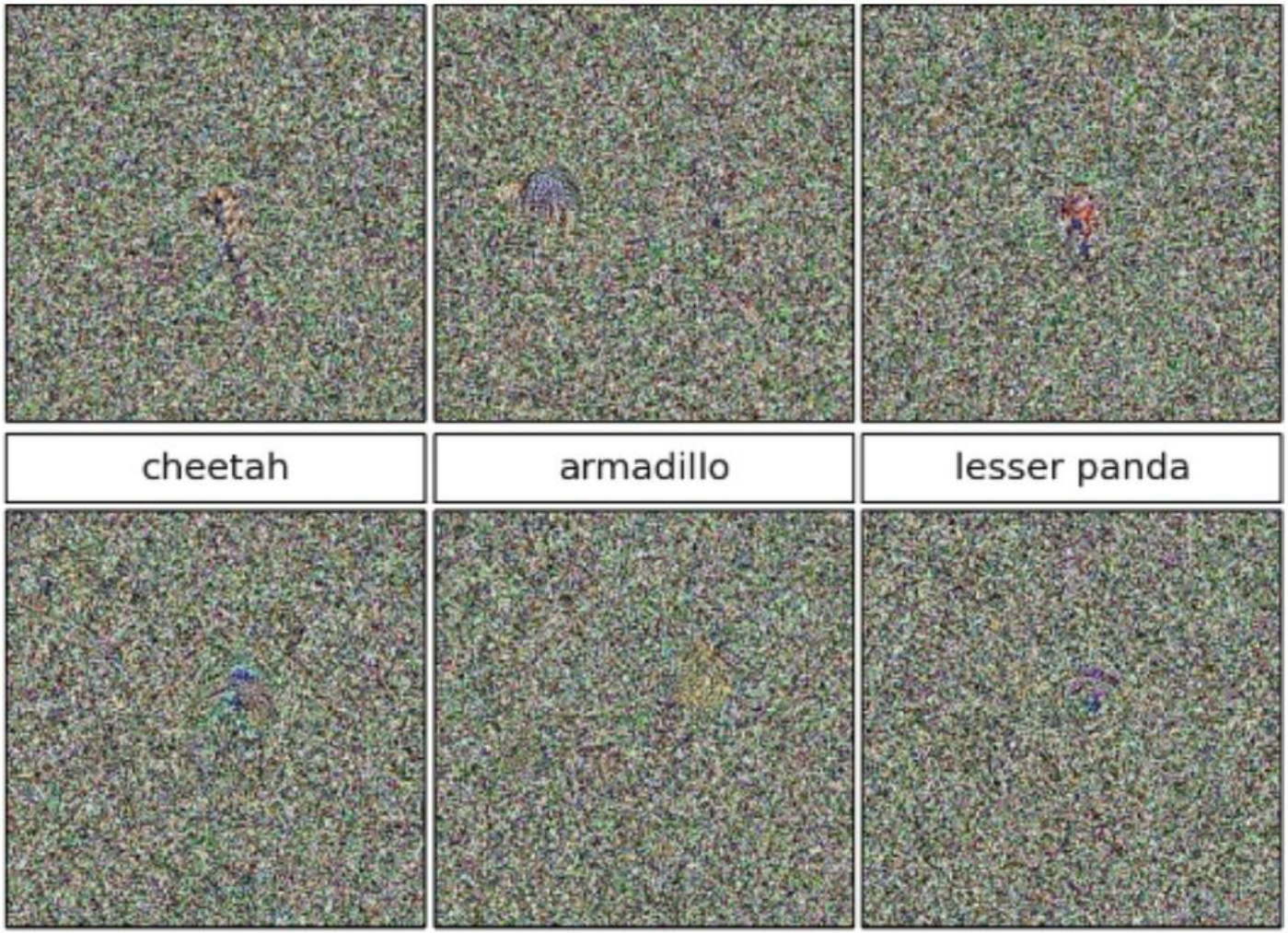
Model fitting

- How to adjust millions of parameters without overfitting.
  - We require lot of training data, even worse: labeled data.
- Tons of labeled data, How?
  - Use a creative, cheap, and/or fast ways to label data.
  - Apply training tricks (add noise, synthetic data, etc.).
  - Transfer learning.
  - Self-supervision.

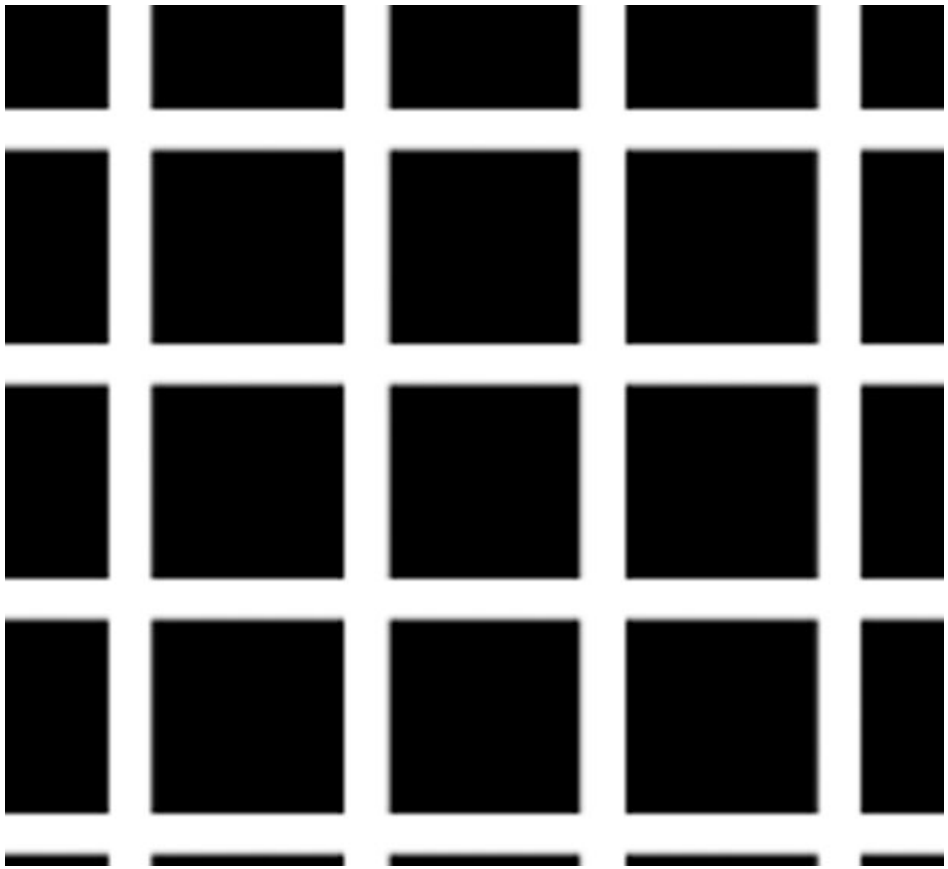
Difficult optimization and scalability

- Layer-wise training using tons of data.
- Parallel training, use of high performance computing (GPUs).

Criticism



Humans are also fooled



Memory vs Reasoning