UNIVERSITÄT
BERN

Representation Learning

Paolo Favaro

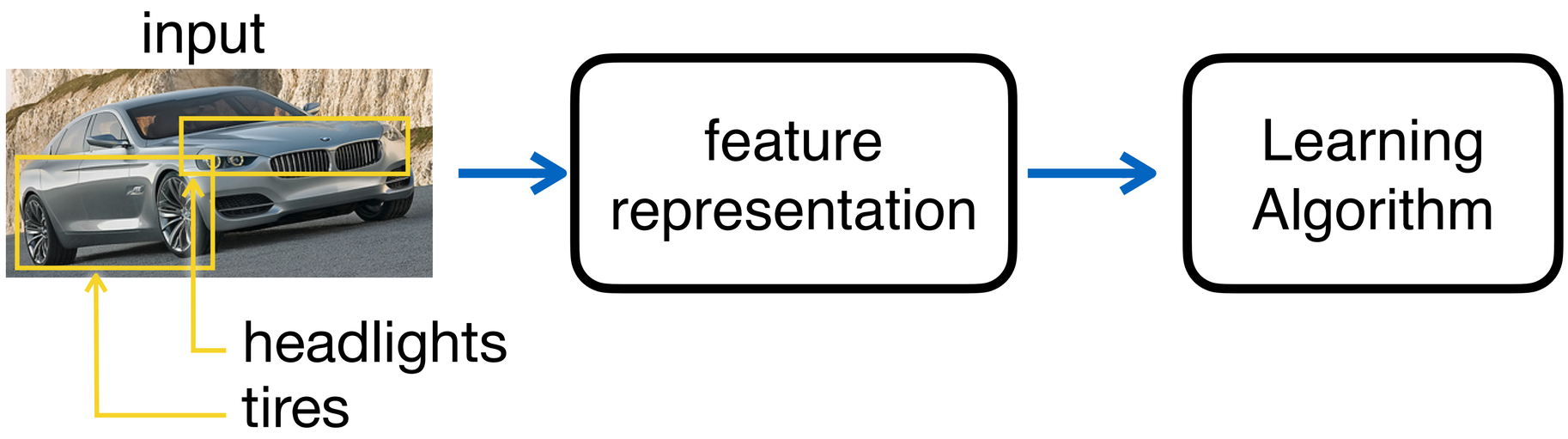
Contents

- Representation learning
  - Transfer learning, domain adaptation, zero, one and few-shot learning, disentangling of causal factors
- Based on **Chapter 15** of Deep Learning by Goodfellow, Bengio, Courville

Representation Learning

- A guideline to design neural networks
- Aims at identifying the causes of the observed data
- Allows to combine unsupervised and semi-supervised learning
- Useful in multitask learning and transfer learning

Representation Learning



Representation Learning

- **General rule**: a good representation is one that makes a subsequent learning task easier
  - Encourage the representation to have independent components
  - It is easier to extend the representation by adding new independent components
- **Example**: Learn a representation that makes probability density estimation easier

Applications

- Transfer learning
- Domain adaptation
- One/Few/Zero-shot learning
- Semi-supervised learning

Transfer Learning

- The learner performs multiple tasks
- Learn a representation for $p_1(x)$ and use it to learn a representation for $p_2(z)$
- We assume that many factors of variations in the data x and z are **shared**
- **Example**: x are images of cats and dogs, and z are images of ants and wasps
  They share low-level notions, such as edges and visual shapes, geometrical and illumination effects

Transfer Learning

(i) Tasks that share semantics of the **input** (one input for all tasks)

(ii) Tasks that share semantics of the **output** (one output for all tasks)

- **Example #1**: speech recognition. Several phonemes mapped to the same sentence.
- **Example #2**: an image, a sketch and a caption may represent the same scene (use the same feature vector for all)

Domain Adaptation

- Same task different data distributions

- **Example**: Sentiment analysis
  (a) Train sentiment predictor based on reviews about books and music
  (b) Use the predictor to analyze content about consumer electronics
  (c) The vocabulary/style might change, but there should be a common function to decide sentiment

One-Shot Learning

- Only one labeled example is given!
- Humans seem to learn with very few labeled examples
- What we have learned on other categories allows us to separate data into factors of variation
- Those factors are useful to classify new categories

Zero-Shot Learning

- No labeled examples are given!
- Needs a formal description of the categories
- **Example**: a person learns about animals from a book without ever having seen them
  If the description is accurate enough, the person should be able to recognize them in images without labeled examples

GPT-3

- In-context learning
- Large scale datasets + general purpose tasks = emergence of new capabilities

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  cheese =>                           ← prompt
```

GPT-3

- In-context learning
- Large scale datasets + general purpose tasks = emergence of new capabilities

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← example
3  cheese =>                           ← prompt
```

GPT-3

- In-context learning
- Large scale datasets + general purpose tasks = emergence of new capabilities

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1  Translate English to French:          ← task description
2  sea otter => loutre de mer            ← examples
3  peppermint => menthe poivrée          ←
4  plush girafe => girafe peluche        ←
5  cheese =>                             ← prompt
```

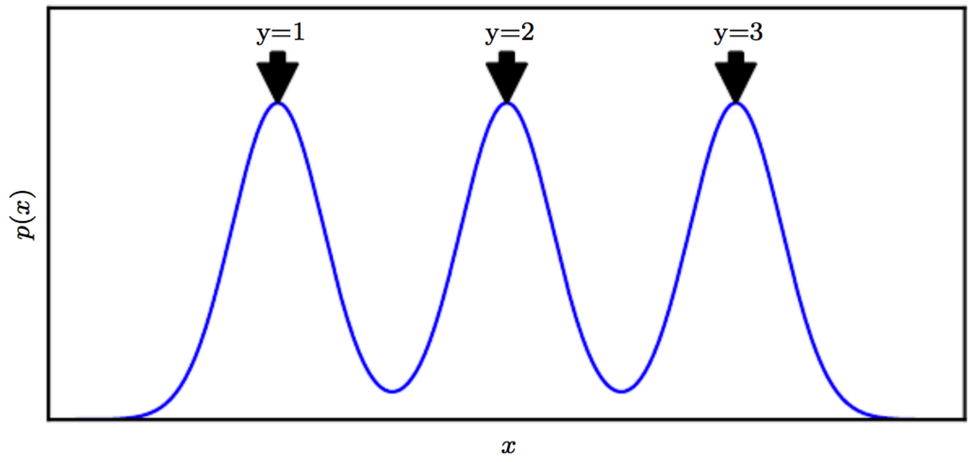
Semi-Supervised Disentangling of Causal Factors

- What makes one representation better than another?
  - Features correspond to separate causes
  - Features are easy to model (e.g., sparse, independent)

Unsupervised Learning

- **Failure example**: p(x) is uniform, find f(x) = E[y|x]
  — A training set of x does not give us information about p(y|x)

- **Success example**: p(x) a mixture with well-separated Gaussians (one for each y instance) — then p(x) will capture p(y|x) very well



Unsupervised Learning

- What makes p(y|x) and p(x) tied together?
  - y closely associated to the causal factors of x
  - How many causal factors do we need?
- Do humans encode all details in their representation?
  - Evidence shows that humans change their representation depending on the task they perform

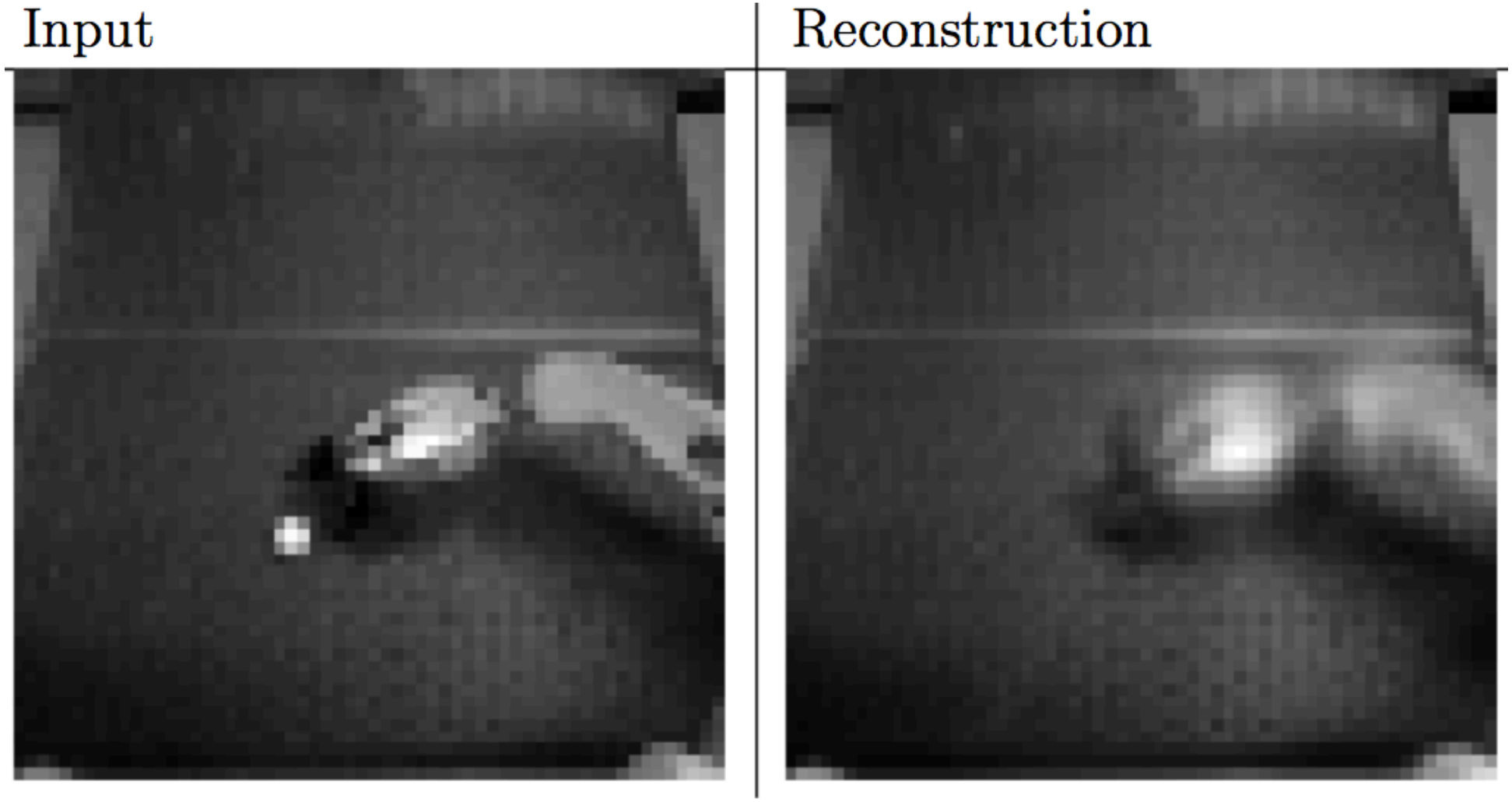
Unsupervised Learning

- **Example**: Let h be all the causal factors of x and let y be one of them
- $p(h,x) = p(x|h)\ p(h)$
- $p(x) = E_h[p(x|h)]$
- Suppose $y = h_i$, but we do not know which one
- Solution: Predict all h_j and predict y from h

Unsupervised Learning

- Another way to change the representation is to change the loss function (eg, L^2 vs entropy)
- For example, L^2 says that factors of variations are important only when they lead to a change of brightness
- This is not a good choice if we are interested in small objects

Unsupervised Learning Failure



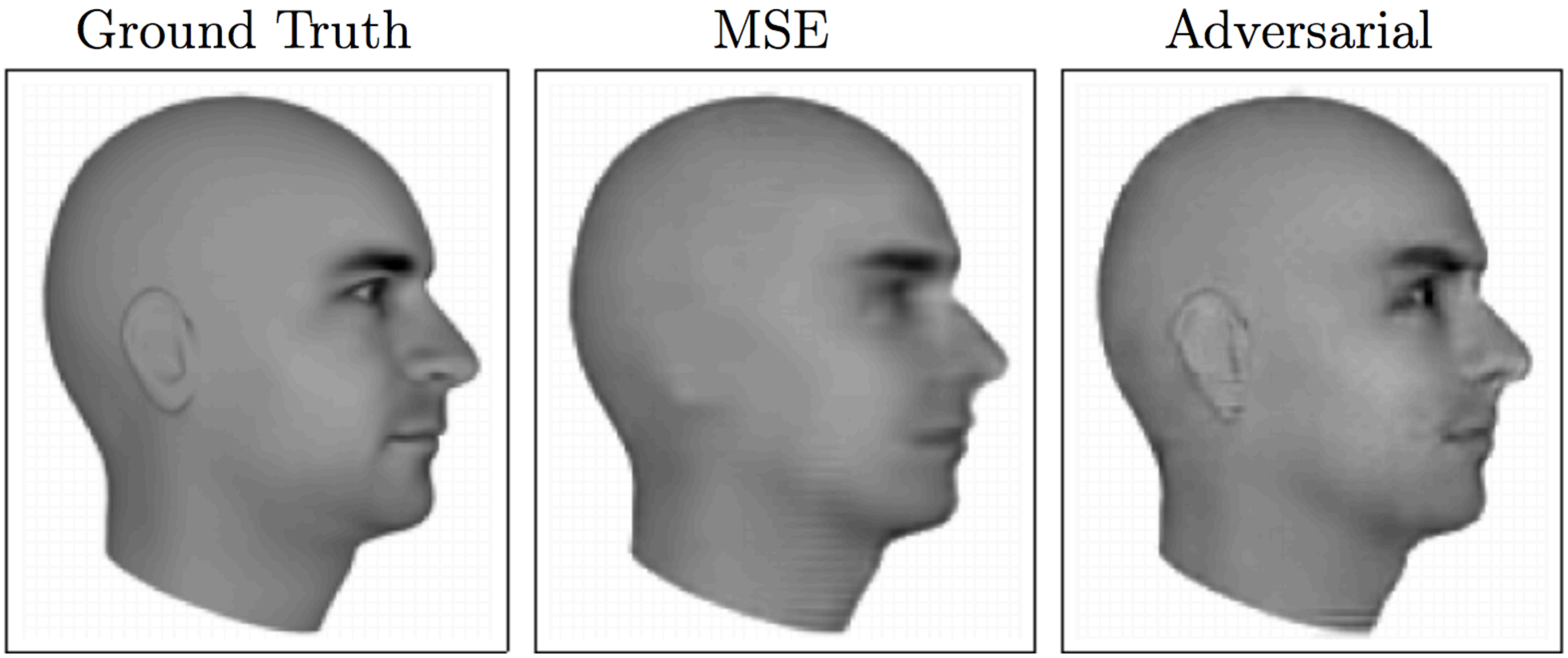
Learning the Loss

- We might not know the best loss function for a task beforehand
- One solution is then to **learn** it
- **Generative adversarial networks** (GAN) provide one such framework

Generative Adversarial Networks

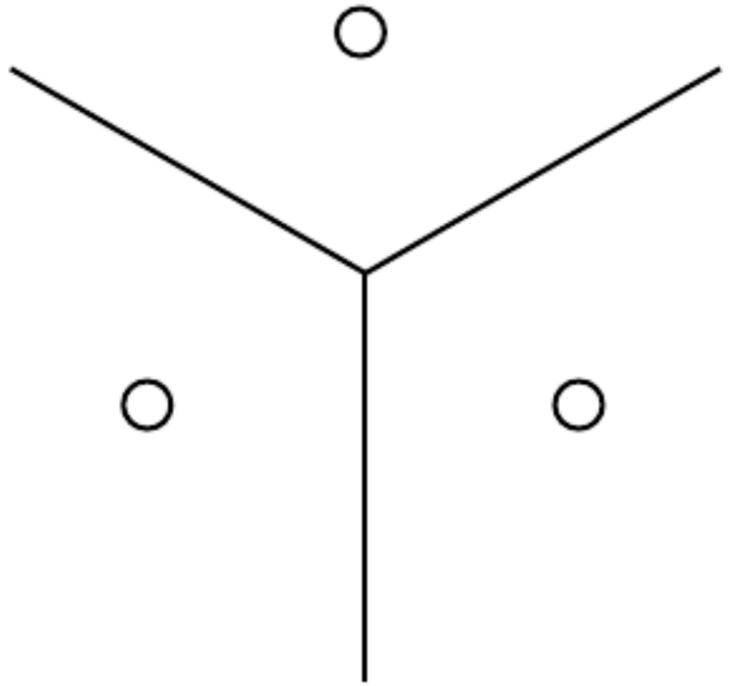
- We are given a training dataset $\mathbf{x}_1, \ldots, \mathbf{x}_m$
- Two models: a **generator G** and a **discriminator D**
- **D** tries to distinguish samples from the training set from samples from **G** (it is a binary classifier)
- **G** generates samples from random Gaussian noise and tries to fool **D**
- Equilibrium is reached when G wins

Generative Adversarial Networks



Local vs Distributed Representations

Local Representation



Local Representation

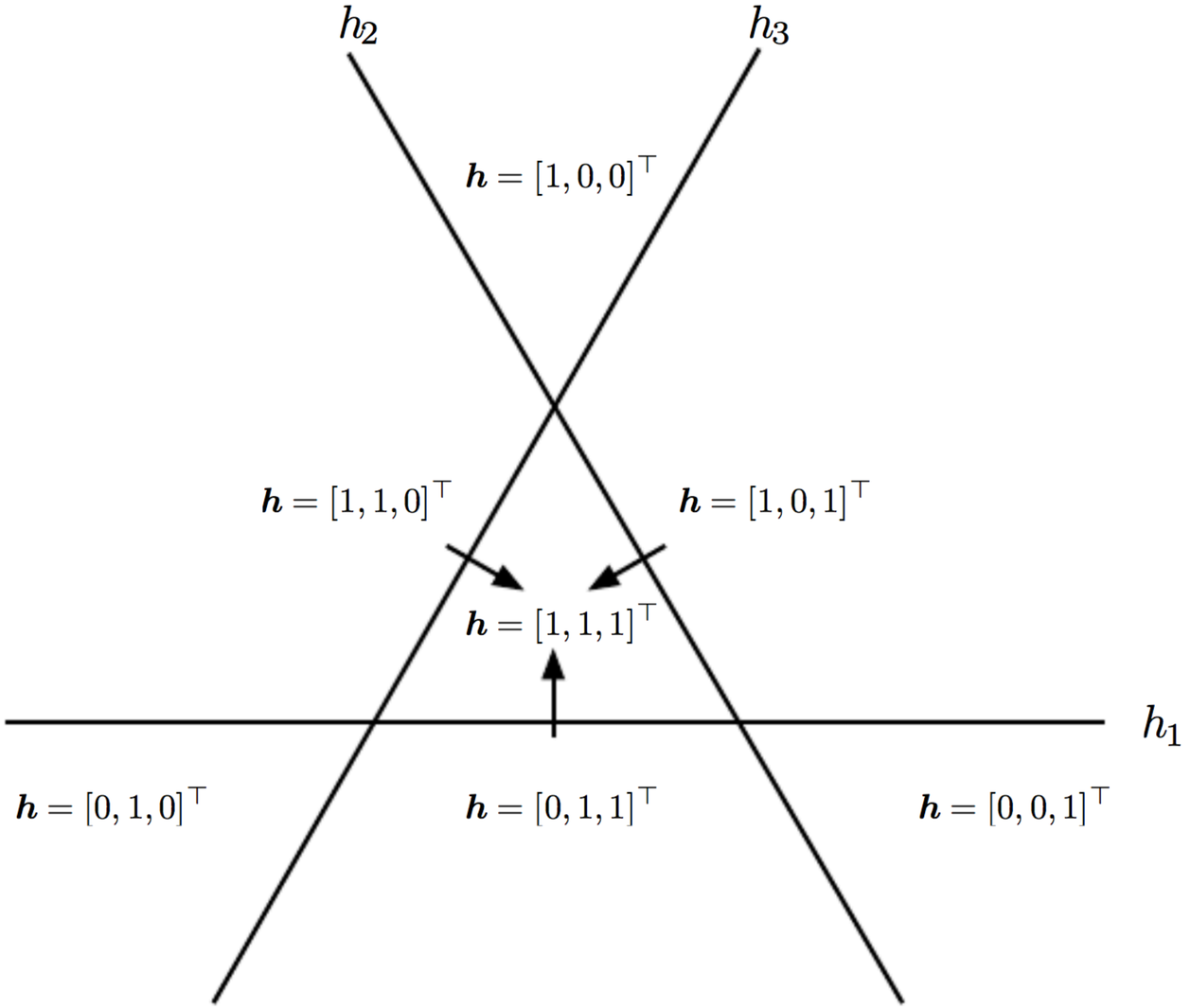
- Clustering methods such as **k-means**: an input is assigned to one cluster
- **k-nearest neighbor**: an input is assigned to one or a few data samples (from the training set)
- **Decision trees**: an input is assigned to one leaf (a path on the tree does not share information with the other paths)

Local Representation

- Relies on **smoothness constraint**: small input changes should give small output changes

- Because each input/output mapping is independent from the others, this representation **does not generalize** to new inputs

Distributed Representation



Distributed Representation

- **Example**: Classification into “cat” and “dog”
- Many concepts can describe both (e.g., “has_fur” and “number_of_legs”)
- A distributed representation based on such concepts would share features
- This helps to better generalize, because an example contributes to both categories at once

Distributed Representation

- Consists of a set of elements that can be set separately from each other
- A data sample is then represented by a pattern of activity distributed over each element

Distributed Representation

- Introduces a gain when representing complex structures with a small number of parameters
- Consider n d-dimensional features, then they can distinguish $O(n^d)$ regions through thresholding with O(nd) parameters
- A local representation would require $O(n^d)$ parameters
- Generalization favors the distributed representation

Distributed Representation

- The capacity is limited although the encodings are exponentially many
- Based on the assumption that data is not structured in any possible random way, but favors shared substructures
- This seems to be the case for images for example