Introduction

Representation Learning

Vladimir Zaigrajew

2025-03-05

Vladimir Zaigrajew - vladimir.zaigrajew.dokt@pw.edu.pl

Tymoteusz Kwieciński - tymoteuszkwiecinski@gmail.com

You can find us in Room 316, MINI, *PW*

Remember every information you can find on our Github Repo:

Figure 1: QR code to course Github Repo

Figure 2: QR code to our Github Repo

Let's chat on slack or discord, becouse I don't like Teams :/.

Core Concept

In machine learning, we usually want to predict some value $y \in Y$ given some data $x \in X$: we want to learn a function $f : X \to Y$

Domain	Task	Example Output
Image	Segmentation, Detection, Classification	Class labels, Bounding boxes
Text	Sentiment Analysis, Next Word Prediction	Sentiment scores, Text generation
Multimodal	Image Description, Image Generation	Generated images, Text descriptions

Instead of learning a direct mapping $f : X \to Y$ from input to output, representation learning approach split the problem into two parts: learning a representation $g : X \to Z$ that transforms raw data into a meaningful feature space, followed by learning a classifier/predictor $h : Z \to Y$.

X and Y don't have to be from the same domain.

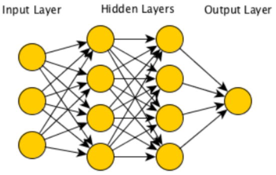
The complete model can be expressed as $f(x) = h(g(x))$

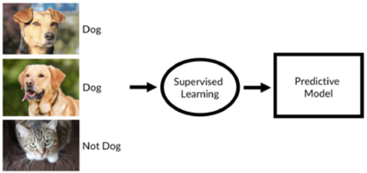
Learning Representation - Why?

This approach brings several advantages. Most importantly, we can learn representations without labels (unsupervised/self-supervised), reducing the need for manual labeling. With good representations, simpler classifiers can then be used for different tasks, making learning faster and more efficient.

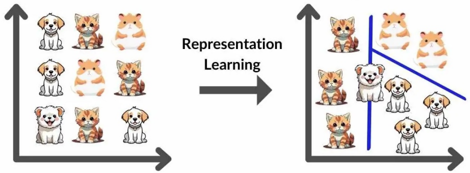
Representation learning transforms complex data into a simpler format that captures important features. Think of face recognition: instead of working with raw pixels, we learn meaningful features like pose and identity, making the recognition task easier.

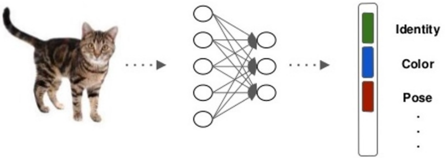
Traditional





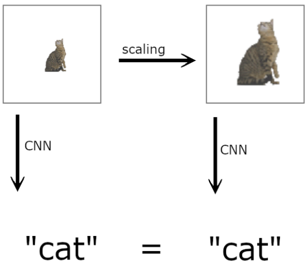
With Representation Learning





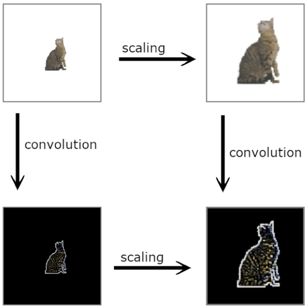
What makes a good representation? I

- **Smoothness:** Similar inputs should have similar representations. If $x_1 \approx x_2$, then $g(x_1) \approx g(x_2)$. This fundamental property ensures that our representations are stable and meaningful.
- **"Less" supervised learning:** Good representations can be learned with minimal supervision, enabling self-supervised and semi-supervised approaches.
- **Invariances/Equivariance/Coherence:** Generally, small temporal/spatial changes should result in similar representations.
  Domain specific: image representations should be invariant under transformations like rotations, color jitter etc.



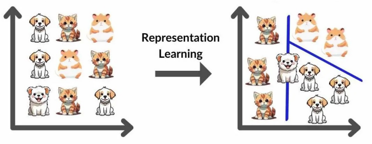
(a) **Invariance**[1]

- Example: Face recognition should be invariant to lighting changes
- $h(g(x)) = h(g(T(x)))$, where T is some not important transformation



(b) **Equivariance**[1]

- Example: If you rotate an image, the features should rotate similarly
- If T is a transformation, then $g(T(x)) = T(g(x))$



(c) **Coherence/Smoothness**

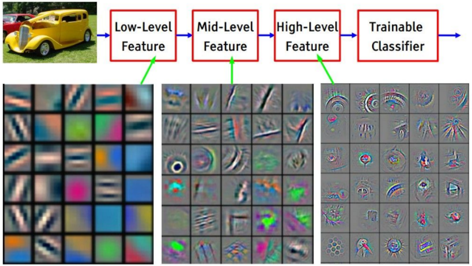
- Close inputs should map to close representations
- Important for generalization and robustness
- If $x_1 \approx x_2$, then $g(x_1) \approx g(x_2)$

[1]Source: https://towardsdatascience.com/sesn-cec766026179/

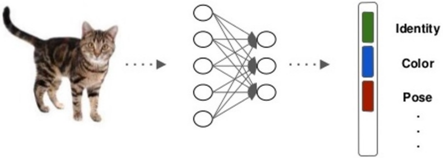
- **Smoothness:** Similar inputs should have similar representations. If $x_1 \approx x_2$, then $g(x_1) \approx g(x_2)$. This fundamental property ensures that our representations are stable and meaningful.
- **“Less” supervised learning:** Good representations can be learned with minimal supervision, enabling self-supervised and semi-supervised approaches.
- **Invariances/Equivariance/Coherence:** Generally, small temporal/spatial changes should result in similar representations.
  Domain specific: image representations should be invariant under transformations like rotations, color jitter etc.
- **Multiple explanatory factors:** Representations should capture diverse aspects of the data, so that the representation is useful for many different tasks.

What makes a good representation? II

- **Natural clustering:** Representations should reflect natural categories in data, which aligns with human-interpretable groupings.
  Example: In vehicle classification, representations should cluster vehicles by type (car, truck, motorcycle) rather than by brand.
- **Hierarchical explanatory factors:** Features organized from concrete to abstract, starting from low-level (edges, colors) to high-level (objects, scenes).
- **Disentangle underlying factors:** Each dimension represents distinct meaningful features, making it easier to understand and manipulate the representation.
- **Sparsity:** For any input x , only few factors are relevant $\Rightarrow$ most dimensions of $g(x)$ should be zero.



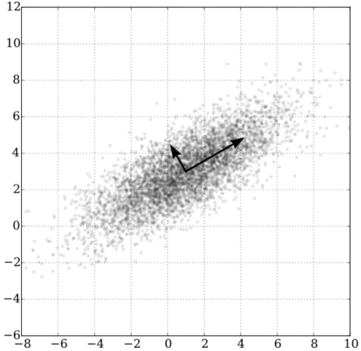
(a) **Natural clustering and Hierarchical explanatory factors**



(b) **Disentangle underlying factors and Sparsity**

Traditional representation learning algorithms

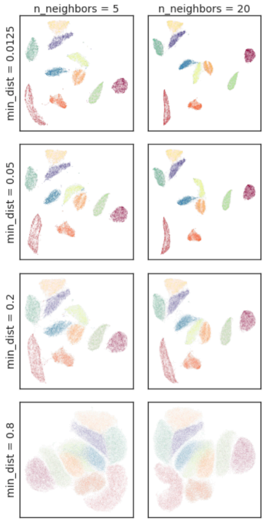
- Principal Component Analysis (PCA)
- Independent Component Analysis (ICA)
- Linear Discriminant Analysis (LDA)
- Multidimensional Scaling (MDS)
- ISOMAP



[2]

[2]"Principal component analysis." Wikipedia, Wikimedia Foundation, 10 Apr. 2023, en.wikipedia.org/wiki/Principal component analysis.

- Principal Component Analysis (PCA)
- Independent Component Analysis (ICA)
- Linear Discriminant Analysis (LDA)
- Multidimensional Scaling (MDS)
- ISOMAP
- *t-SNE* (t-Distributed Stochastic Neighbor Embedding)
- *UMAP* (Uniform Manifold Approximation and Projection)

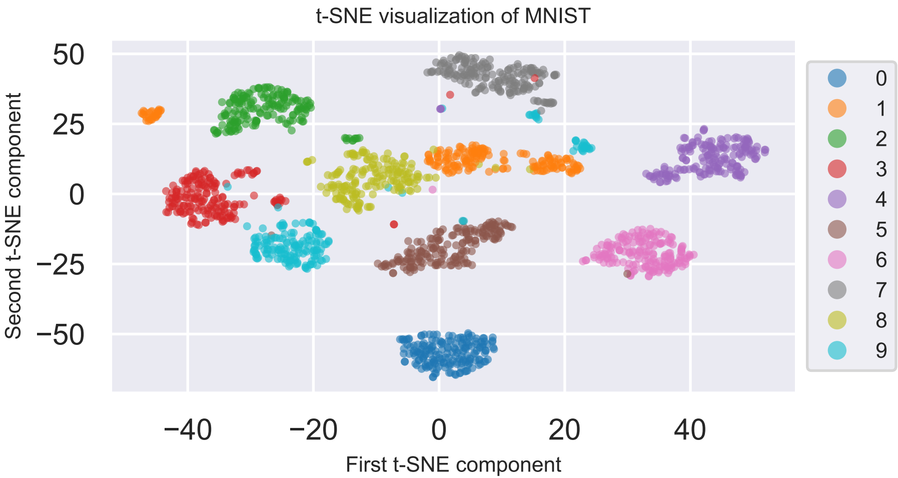


[3]

[3] McInnes, Leland, John Healy, and James Melville. "UMAP: uniform manifold approximation and projection for dimension reduction. arXiv." arXiv preprint arXiv:1802.03426 10 (2018).

Manifold Learning

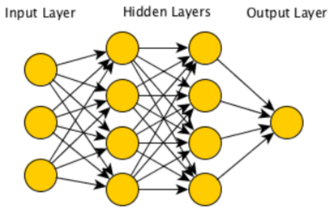
The manifold hypothesis states that real-world high-dimensional data tends to lie on or near a lower-dimensional manifold.



Neural Networks Representations

Neural networks learn representations at multiple levels:

- Before Input Layer (pixels, words, etc.)
- After Input Layer
- Hidden Layer
    - Progressively more abstract features
    - Combine and transform earlier representations
    - Different layers capture different aspects
- Final Layers (task-specific representations)

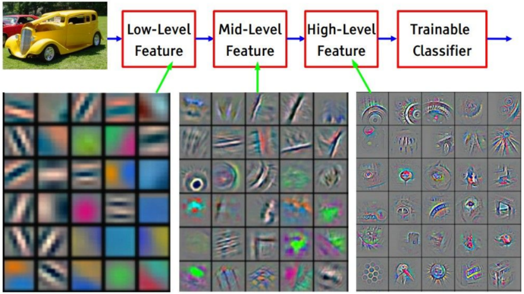


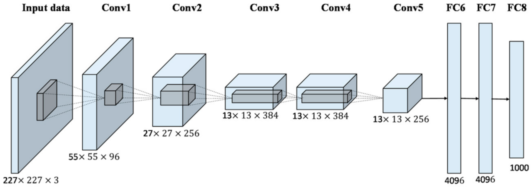
```python
import torch

# Define a simple neural network
class SimpleNN(torch.nn.Module):
    def __init__(self):
        super(SimpleNN, self).__init__()
        self.fc1 = torch.nn.Linear(10, 5)
        self.fc2 = torch.nn.Linear(5, 2)

    def forward(self, x):
        x = torch.relu(self.fc1(x))
        x = self.fc2(x)
        return x

# Create a model instance
model = SimpleNN()
model = model.to('cpu')

# Example input
input_data = torch.randn(1, 10)

# Forward pass
output = model(input_data)
```

```
Input shape: torch.Size([1, 10])
Input dtype: torch.float32
Input device: cpu

Output shape: torch.Size([1, 2])
Output dtype: torch.float32
Output device: cpu
Output: tensor([[-0.5015, -0.2340]], grad_fn=<AddmmBackward
```

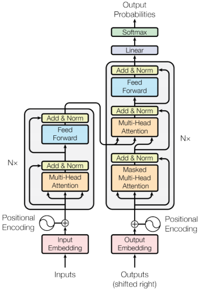
CNN Representations

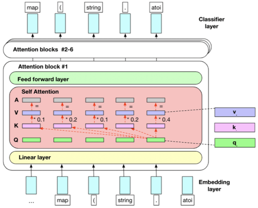
CNNs demonstrate hierarchical representation learning






```
Resnet18 layer layer1 representation shape torch.Size([1, 64, 56, 56])
Resnet18 layer layer2 representation shape torch.Size([1, 128, 28, 28])
Resnet18 layer avgpool representation shape torch.Size([1, 512, 1, 1])
```






```python
import torch
import transformer_lens

# Load a model (eg GPT-2 Small)
model = transformer_lens.HookedTransformer.from_pretrained(
  "gpt2-small")

# Run the model and get logits and activations
input_str = "Tancowala ryba z rakiem, ryba z rakiem"
with torch.no_grad():
    logits, activations = model.run_with_cache(input_str)
block_to_visualize = activations['blocks.11.ln1.hook_normalized']
Loaded pretrained model gpt2-small into HookedTransformer
```

```
Input text: Tancowala ryba z rakiem, ryba z rakiem
Logits shape: torch.Size([1, 20, 50257])
After embedder layer: torch.Size([1, 20, 768])
After Layer 11: torch.Size([1, 20, 768])
```

[4]

[4] Vaswani,Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

Closing Thoughts

Key takeaways about representation learning:

1. Data representation matters
   - Raw data (images, text) is often not optimal for ML models
   - Good representations make learning easier
2. Multiple levels of abstraction
   - From raw features to high-level concepts
   - Different representations serve different purposes
3. Future directions
   - Self-supervised learning (**Next week**)
   - Multi-modal representations
   - More interpretable representations

For more, read this lecture from the Lab of HHU Dusseldorf (clickable link).