# Representation Learning and Auto-Encoder

By Jiachen and Abrham

# What is a representation?

(From one of the most Voted Answers in Google)

“Representation is basically the space of allowed models (the hypothesis space), but also takes into account the fact that we are expressing models in some formal language that may encode some models more easily than others (even within that possible set)”

(From Wiki)

“In machine learning, feature learning or **representation learning** is a set of techniques that allows a system to **automatically discover the representations** needed for feature detection or classification from raw data.” (Wiki)

# What is a representation?

(Bengio, Yoshua, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives." IEEE transactions on pattern analysis and machine intelligence 35.8 (2013): 1798-1828.)

**Representation is a feature** of data that can entangle and hide more or less the different explanatory factors or variation behind the data.

# What is a representation?

(Bengio, Yoshua, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives." IEEE transactions on pattern analysis and machine intelligence 35.8 (2013): 1798-1828.)

**Representation is a feature** of data that can entangle and hide more or less the different explanatory factors or variation behind the data.

# What is a feature?

# What is a representation?

(Bengio, Yoshua, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives." IEEE transactions on pattern analysis and machine intelligence 35.8 (2013): 1798-1828.)

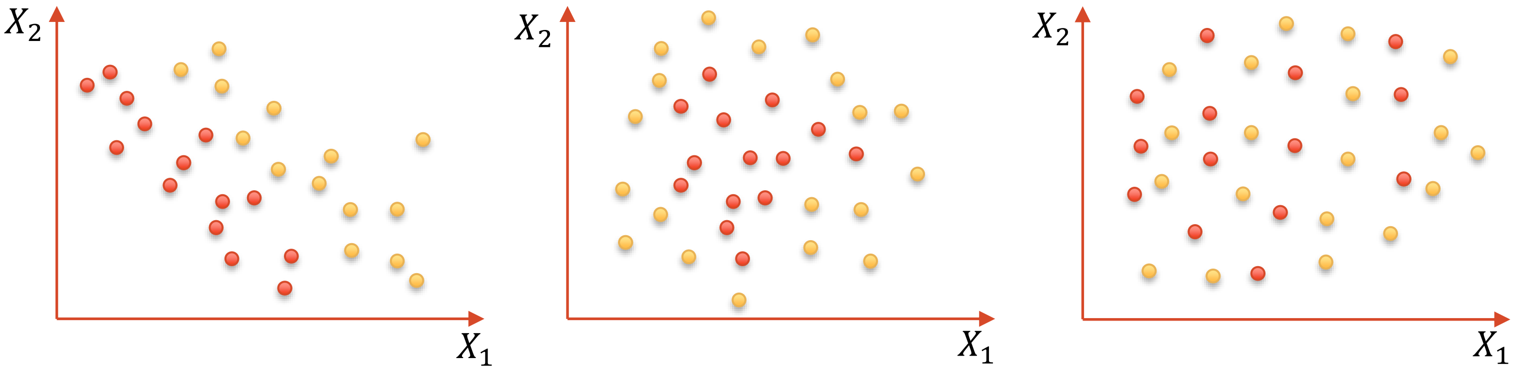
**Representation is a feature** of data that can entangle and hide more or less the different explanatory factors or variation behind the data.

# What is a feature?

1. The **input vector** to a machine learning model (1%)
2. Any Interpretable **vector** in a machine learning model (*)

# What is a representation?

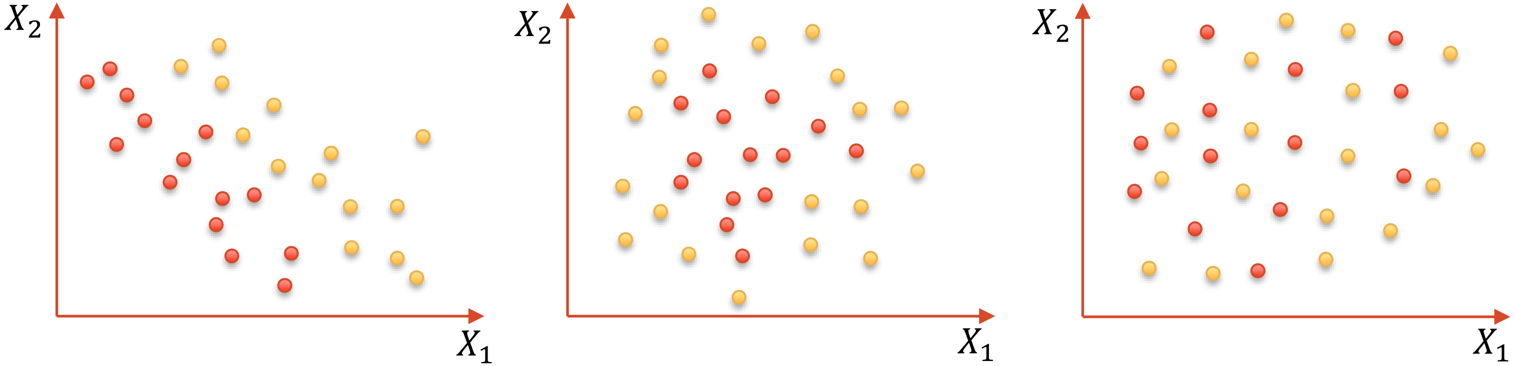
Binary Logistic Regression




Input Feature: $(X_1, X_2, 1)$ $(X_1^2, X_2^2, X_1X_2, X_1, X_2, 1)$ ?

# What is a representation?

Binary Logistic Regression




Input Feature: $(X_1, X_2, 1)$ $(X_1^2, X_2^2, X_1X_2, X_1, X_2, 1)$ ?

Representation

# What is a representation?

(Bengio, Yoshua, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives." IEEE transactions on pattern analysis and machine intelligence 35.8 (2013): 1798-1828.)

**Representation is a feature** of data that can entangle and hide more or less the different explanatory factors or variation behind the data.

# What is a feature?

1. The **input vector** to a machine learning model (1%)
2. Any Interpretable **vector** in a machine learning model (*)

# What is a representation?

(Bengio, Yoshua, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives." IEEE transactions on pattern analysis and machine intelligence 35.8 (2013): 1798-1828.)

**Representation is a feature** of data that can entangle and hide more or less the different explanatory factors or variation behind the data.

# What is a feature?

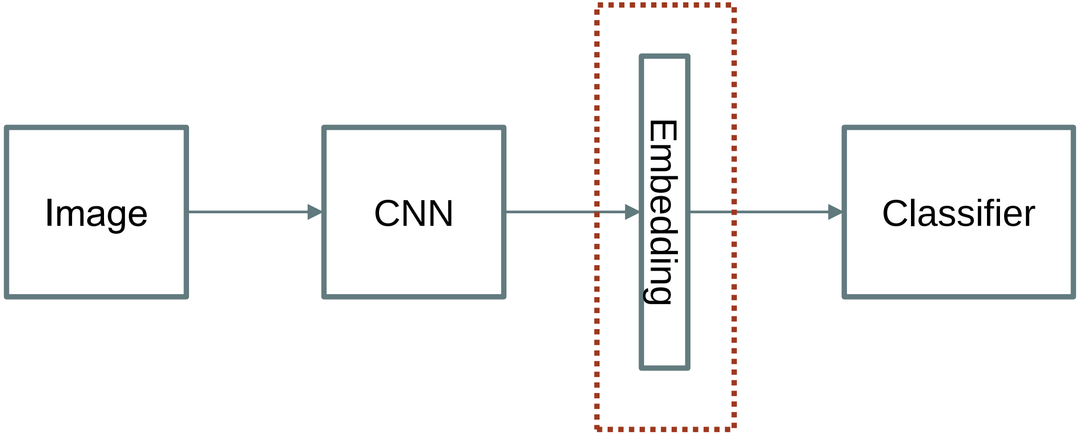
1. The **input vector** to a machine learning model (1%)
2. Any Interpretable **vector** in a machine learning model (*)

# What is a representation?

- Image Embeddings in Recognition
- Token/Sentence Embedding in NLP
- Audio Embedding
- Latent Factor in AE(PCA), VAE
- Multi-modal Representation
- ….
- Almost

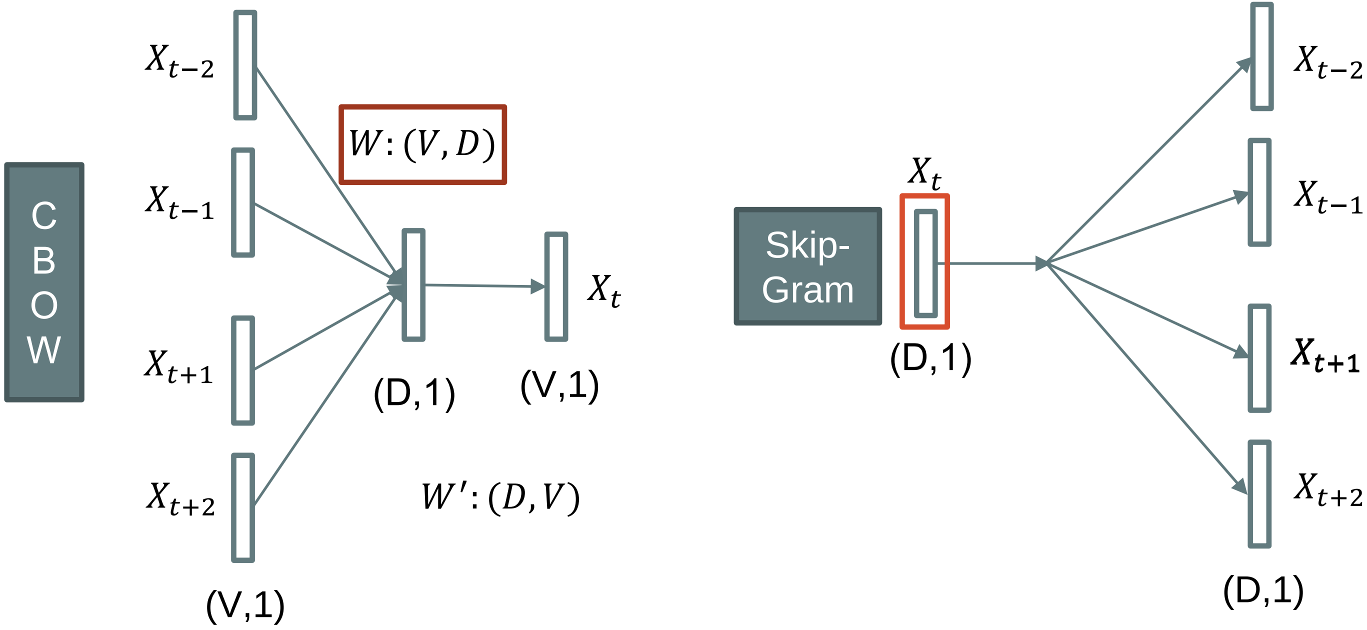
# What is a representation?

- Image Embeddings in Object Recognition (hw2p2)



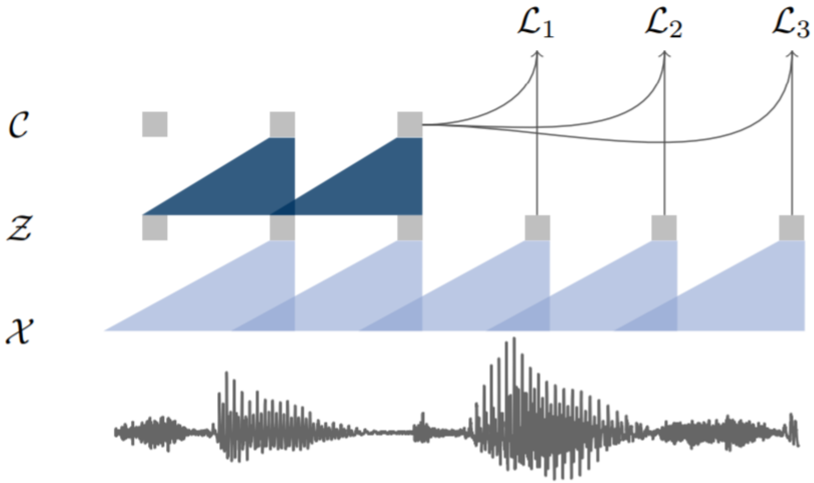
# What is a representation?

- Token Embedding (Word2Vec)



# What is a representation?

- Audio Embedding (Wav2Vec)



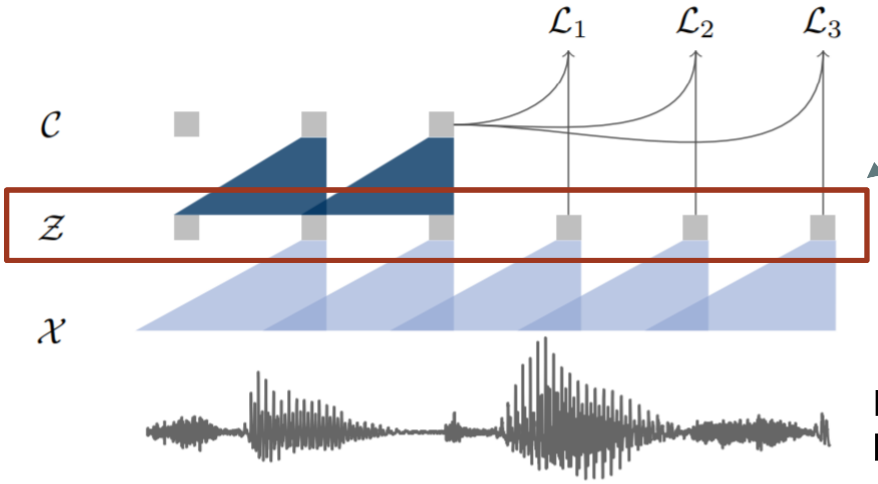

Casual CNN

$$\mathcal{L}_k = -\sum_{i=1}^{T-k} \left( \log \sigma(\mathbf{z}_{i+k}^\top h_k(\mathbf{c}_i)) + \lambda \mathop{\mathbb{E}}_{\tilde{\mathbf{z}}\sim p_n} [\log \sigma(-\tilde{\mathbf{z}}^\top h_k(\mathbf{c}_i))] \right)$$

Schneider, Steffen, et al. "wav2vec: Unsupervised pre-training for speech recognition." *arXiv preprint arXiv:1904.05862* (2019).

# What is a representation?

- Audio Embedding (Wav2Vec)




Casual CNN

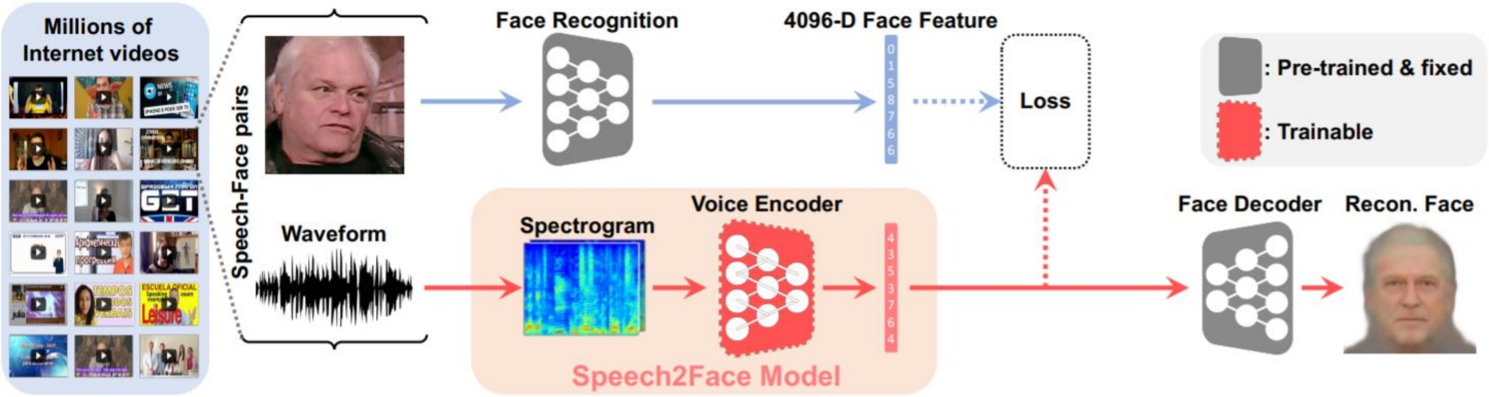
$$\mathcal{L}_k = -\sum_{i=1}^{T-k}\left(\log\sigma(\mathbf{z}_{i+k}^\top h_k(\mathbf{c}_i)) + \lambda \underset{\tilde{\mathbf{z}}\sim p_n}{\mathbb{E}}\left[\log\sigma(-\tilde{\mathbf{z}}^\top h_k(\mathbf{c}_i))\right]\right)$$

Learn a general representation that can be input to a speech recognition system

Schneider, Steffen, et al. "wav2vec: Unsupervised pre-training for speech recognition." *arXiv preprint arXiv:1904.05862* (2019).

# What is a representation?

- Multi-modal Representation(Speech2Face)




Oh, Tae-Hyun, et al. "Speech2face: Learning the face behind a voice." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

# What is a representation?

**Representation is a feature** of data that can entangle and hide more or less the different explanatory factors or variation behind the data.

# What is a feature?

Any Interpretable **vector** in a machine learning model (*)

# What is a representation?

**Representation is a feature** of data that can entangle and hide more or less the different explanatory factors or variation behind the data.

# What is a feature?

Any Interpretable **vector** in a machine learning model (*)

# How to Learn a good Representation?

# Representation Learning is a mindset

The most intriguing question for deep learning practitioners: **What architecture must be used?**

**Consider two things:**

- Optimization : can my model learn by SGD and generalize well  ( depth, skip-connections, dropout, batchnorm,etc)
- Representation : can my model capture the “inner structure”  of the data ?

These two are partly complementary, partly supplementary.

# Representation Learning is a mindset

Why are CNNs good for images ?

- Features you need in image tasks tend to be translation-invariant

Why are RNNs good for sequences ?

- Features you need in language tasks have long-term dependencies

Why is attention useful in Seq2Seq ?

- A decoded word's representation should be correlated to some specific input words
- Rather than thinking in "tricks" or academic recipes, think of what design makes sense w.r.t the properties you seek in your representation.
    - Ex : for sentiment analysis, CNNs are often better than RNNs

# Representation Learning is a mindset

- End-to-end (what you usually do)
- In an unsupervised fashion (autoencoders)
- On an alternate task
- Use a pretrained model (Ex: pretrained word embeddings)

If you use a representation learned one way and move on to the task you're really interested in, you can :

- Fine-tune the representation
- Fix it, and add deeper layers

Think about what makes most sense based on the data you have and the task you're interested in.

# Representation Learning is a mindset

**Transfer learning**

Train a neural network on an easy-to-train task where you have a lot of data.

Then, change only the final layer fine-tune it on a harder task, or one where you have less data.

Ex : **HW2p2**, if you train a network on classification then fine-tune it with centre loss for verification, the first task's representation was useful for the second.

# Representation Learning is a mindset

**Semi-supervised learning**

Train both an unsupervised model and a supervised one, with or without shared parameters.

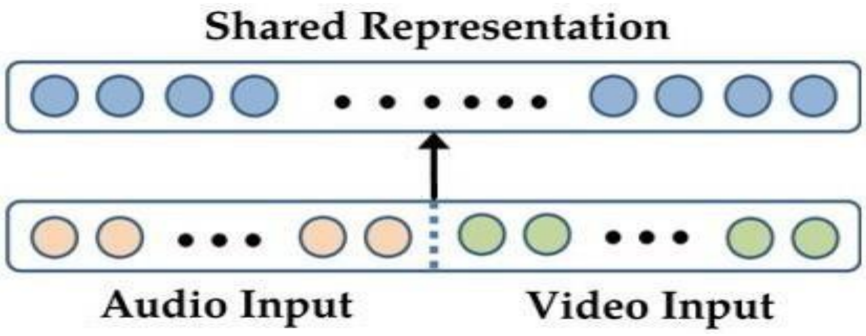
Example : you want to learn to translate from English to French. You have a lot of text in French, a lot of text in English, **but** a limited amount of aligned French-English data.

A seq2seq model with attention could only learns on the aligned data.

What do you do ?

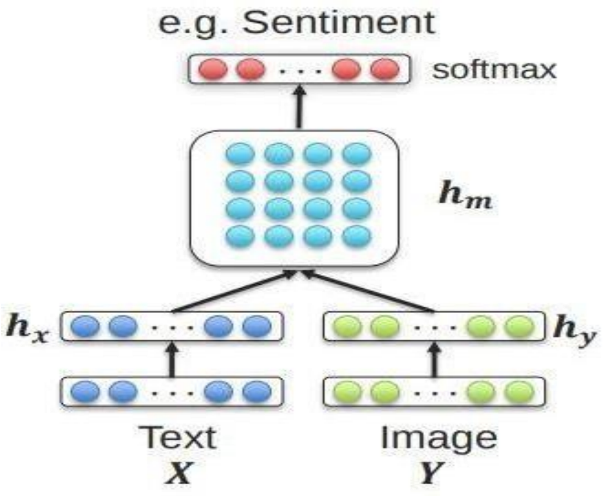
# Multimodal Representation Learning

- How do we deal with tasks involving 2 or more modalities? For instance, VQA, given an image and a question about it, find the answer.
- Approach 1 (shallow representations): Simply concatenate representations and plug that in your end-to-end network.
  - Pro : Easy to Implement
  - Con: Doesn't capture interactions between modalities



# Multimodal Representation Learning

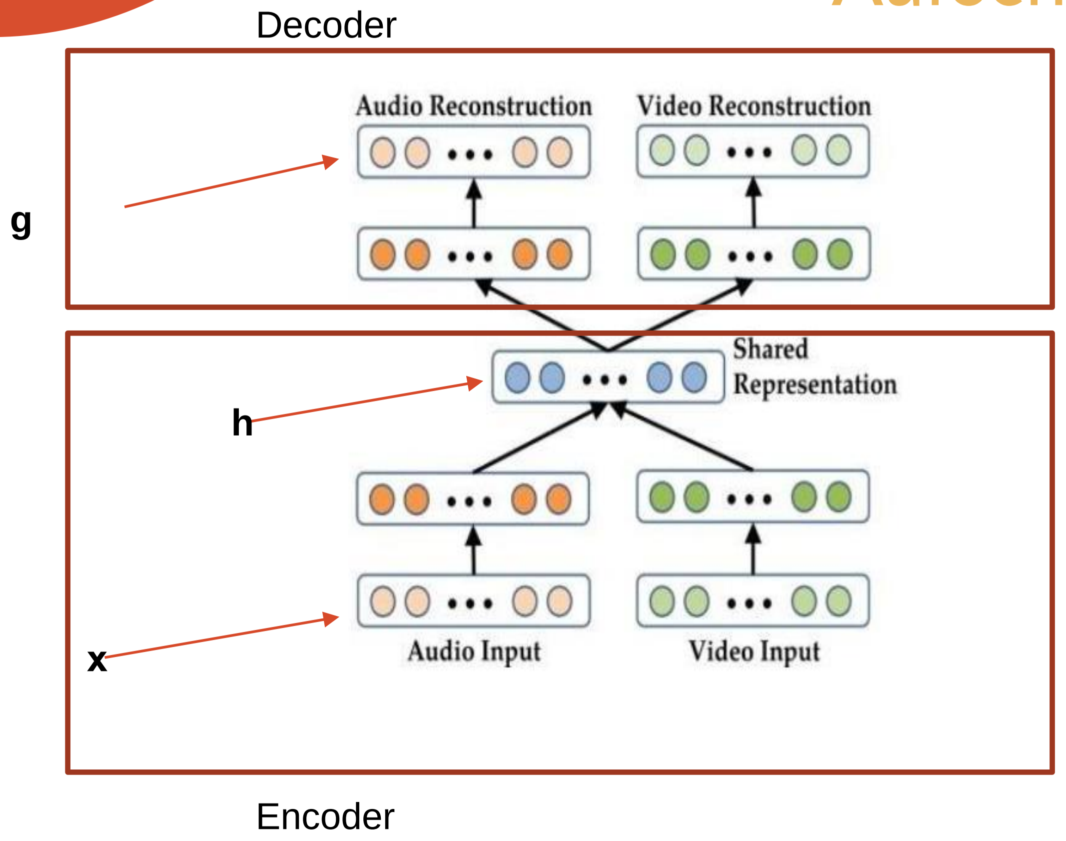
- Approach 2: (Bilinear Pooling) : Use an outer product on unimodal representation vectors to get a (large) multimodal vector.
    - Pros:
        - Captures every pattern
        - Great for **complementary** modalities, i.e. when they contain different information
    - Cons:
        - Can be more rich than needed, especially when modalities contain redundant (supplementary) information
        - Untrackable. But there are improvements like Compact Bilinear pooling or Tensor Fusion. Look it up !

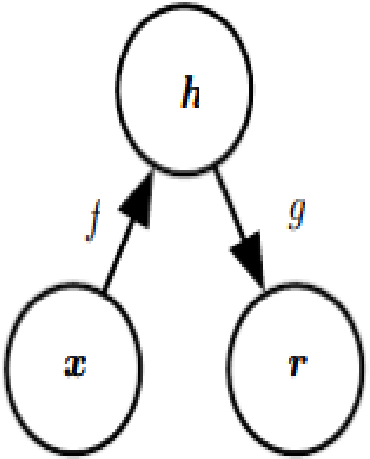


# Autoencoders

- Encode modalities in a shared space
- Train and then when training the downstream task keep only the encoder part
- Pros : Extremely robust, can reconstruct missing modalities if trained well
- Cons : Needs separate training, and often not state-of-the-art compared to pooled or coordinated representations

# Autoencoders



# Autoencoders

- Autoencoders are trained to reconstruct the input
- However, simply reconstructing the input is useless
- Usually, the output of the decoder is not what is needed

# Types of Autoencoders

- Undercomplete autoencoders
- Denoising Autoencoders
- Sparse Autoencoders
- Contractive Autoencoders
- …

# Undercomplete Autoencoders

- Autoencoders with code dimension smaller than the input dimension
- $\dim(\mathbf{h}) < \dim(\mathbf{x})$
- Minimize the loss function in the form of
    - $L(\mathbf{x}, g(f(\mathbf{x})))$
- Usually L() is the mean squared error loss
- Usually, an overcomplete autoencoder ($\dim(\mathbf{h}) > \dim(\mathbf{x})$) does not learn meaningful features i.e it only learns to reconstruct its input

# Sparse Autoencoders

- Are autoencoders that force the hidden representation **h** to have as many zeros as possible
- Loss function of the form
  - $L(\mathbf{x}, g(f(\mathbf{x}))) + \Omega(\mathbf{h})$
- The loss function L() usually something like a mean squared loss
- The regularization penalty Ω() enforces sparsity of the hidden representation
- Can learn meaningful features even if it is **overcomplete**!

# Denoising Autoencoders

- Denoising autoencoders operate on corrupted versions of the input
    - $L(\mathbf{x}, g(f(\tilde{x})))$
- Where $\tilde{x}$ is $\mathbf{x}$ + noise
- Must learn how to remove noise from input
- In the process of noise removal it ends up learning useful features about the input distribution