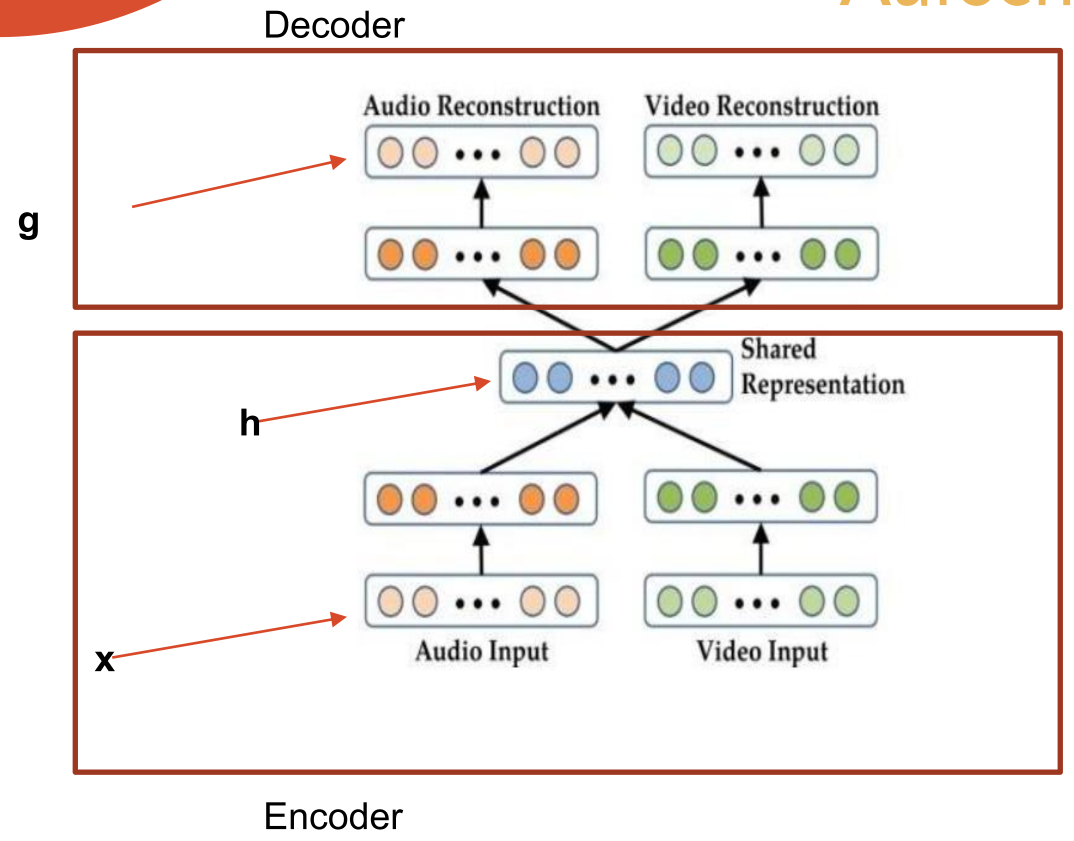
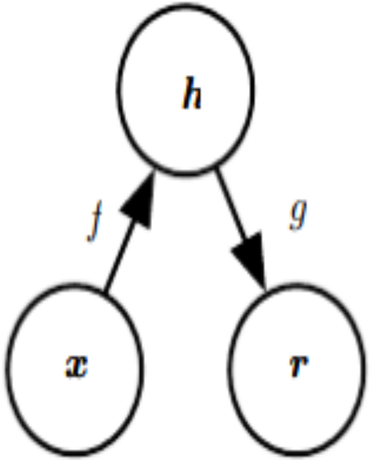
Autoencoders

- 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 $\mathbf{h}$ 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