Neural Decompositional Memory

February 10, 2020

1 Motivations

- Recent trends in learning to decompose, disentangle, explainable AI
- Memory role in representing concept hierarchy or decomposition is not clear
- This paper aims to give another way to explain the role of memory in this problem

2 Latent Encoder Decoder Framework with External Memory

- VMED learns a MoG of the prior distribution for latent space, which is a form of decomposition.
  - However, its writing/reading mechanisms are DNC-like and not supportive for decompositional functionalities.
  - The KL divergence between MoG and Gaussian is only estimation and hard to optimize
  - There is no constraint on the component of the priors, which reduces decompositional functionalities
- The idea in VMED that each memory slot stores a component of the latent space is simple yet fundamental. If we assume the idea is correct, we can view there is an inner latent encoding-decoding process: $z \rightarrow \mathbf{M} \rightarrow z'$, where the memory $\mathbf{M}$ stores the components (basises) of the latent space
  - Latent Encoding: Decompose (project) the latent vector into memory slots
  - Latent Decoding: Compose (reconstruct) the latent vector from the basises stored in $\mathbf{M}$

3 Latent Encoding - Memory Wring: Recurent projections with Stored-Program Memory

- At each step of latent encoding, we need to generate a network W_t to project $z \rightarrow m_t : m_t = W_t z$. The memory is composed by $\{m_t\}_{t=1}^{N}$ row-wise.
- We can use any weight generator, including NSM: $W_t = NSM([z, h_t])$ where h_t is the hidden state of some RNN to make the system recurrent.
- Using NSM means we assume there are few main projectors that can be interpolated to decompose the latent data to its basises

4 Latent Decoding - Memory Reading: Least-square Attention

- Given $\mathbf{M}$, we want to reconstruct the latent vector by reading from the memory. The reading weight is the solution to the least square problem:

$$w_t = arg\min_{w} \left\| z - \mathbf{M}^{\top} w \right\|$$

- Basically, we want to attend to memory slots that compose the best linear estimator of z. The solution has a known form: $w_t = \left(\mathbf{M}\mathbf{M}^{\top}\right)^{-1} \mathbf{M} z$
- Again, we may want to make it recurrent, we read multiple times before determining the final reconstructed. We use $z_t' = RNN\left(\mathbf{M}^{\top} w_t, h_t\right)$ to update the memory using normal memory access mechanisms, the redo the least-square attention.