Continual learning with VAEs

Timo Flesch, David Nagy
Outline

● Part 1: Quick Tutorial (Timo)
  ○ Variational Inference
  ○ Variational Autonencoders

● Part 2: Discussion of Paper (David)
Variational Inference
Life-Long Disentangled Representation Learning with Cross-Domain Latent Homologies

Alessandro Achille, Tom Eccles, Loic Matthey, Christopher P Burgess, Nick Watters, Alexander Lerchner, Irina Higgins
UCLA, DeepMind
achille@cs.ucla.edu,
{eccles, lmatthey, cpburgess, nwatters, lerchner, irinah}@google.com
Goals

- continual learning of new tasks without catastrophic forgetting
  - unlabeled (but “piecewise stationary”)
  - unsupervised

- reuse previous representations on new tasks (positive transfer)
  - improve sample efficiency
  - enables scaling with number of tasks
  - how? learn structure of current environment, reasonable expectation of natural tasks is that they tend to adhere to structure of the real world

- extend VAEs to piecewise stationary data instead of i.i.d.
Related work

continual learning approaches:

- **data-based**, augment current data with older training data
- **architecture-based**, dynamically augment network with task-specific modules, often with shared intermediate representations for positive transfer
- **weights-based**, slow down learning in important weights

- 1 & 2 are inefficient in terms of memory use
- 3 typically uses task labels to update loss function after switches
- most of CL literature deals with task-based settings, which reduces positive transfer, because implicitly learned representations overfit by discarding information that is irrelevant to current task but may be required for future tasks
Problem formalisation

- set $S$ of $K$ environments
  
  $S = \{s_1, s_2, ..., s_K\}$

- set $Z$ of $N$ independent data generative factors
  
  $Z = \{z_1, z_2, ..., z_N\}$

- the aim of life-long representation learning can be seen as estimating the full set of generative factors
Generative model

- Generative factors $\sim \mathcal{N}(0, I)$
- Environment $\sim \text{Categorical}(\pi_1, \ldots, \pi_K)$
- Env. mask $\sim \text{Bernoulli}(\omega_1, \ldots, \omega_K)$
- Observation $\sim p(\cdot | z^s, s)$
Inferring the generative factors $z$

$$q(z^s|x^s) = a^s \odot \mathcal{N}(\mu(x), \sigma(x)) + (1-a^s) \odot \mathcal{N}(0, I)$$

- unused factors set to prior
- note that $\mu$ and $\sigma$ depend only on the data $x$ and not on the environment to ensure that the semantic meaning of each latent dimension remains consistent for different environments.
Inferring the generative factors $z$

- For the factors that are used in a given environment
  - MDL loss, almost the same as $\beta$-VAE loss but instead of $\beta$, a target rate is used
  - progressively increasing rate helps with disentangling (Burgess et al., 2017)

$$L_{MDL}(\phi, \theta) = \mathbb{E}_{z^s \sim q_{\phi}(\cdot|x^s)}[-\log p_{\theta}(x|z^s, s)] + \gamma \text{KL}(q_{\phi}(z^s|x^s)||p(z)) - C$$

- Reconstruction error
- Representation capacity
- Target
Inferring the generative factors $\mathbf{z}$

- For the factors that are used in a given environment
  - MDL loss, almost the same as $\beta$-VAE loss but instead of $\beta$, a target rate is used
  - progressively increasing rate helps with disentangling (Burgess et al., 2017)

$$
\mathcal{L}_{\text{MDL}}(\phi, \theta) = \mathbb{E}_{\mathbf{z}^s \sim q_{\phi}(\cdot | \mathbf{x}^s)} \left[ -\log p_{\theta}(\mathbf{x} | \mathbf{z}^s, s) \right] + \gamma \left[ \text{KL}(q_{\phi}(\mathbf{z}^s | \mathbf{x}^s) \parallel p(\mathbf{z})) - C \right]^2
$$

- Reconstruction error
- Representation capacity
- Target

Single latent traversals
Inferring the latent mask $a$

- beta-VAE / MDL indirectly optimises

$$\mathbb{E}[q(z^s | x^s)] \approx p(z)$$
Inferring the latent mask $a$

- beta-VAE / MDL indirectly optimises

$$\mathbb{E}[q(z^s|x^s)] \approx p(z)$$

- meaning that the marginal posterior will be similar to the prior for known factors. This is used to define an atypicality score, which, if larger than some threshold $\lambda$, turns off the factor

$$\alpha_n = \text{KL} \left( \mathbb{E}_{x_{\text{batch}}} [q_\phi(z^s_n|x_{\text{batch}}^s)] \parallel p(z_n) \right)$$

$$a^s_n = \begin{cases} 
1, & \text{if } \alpha_n < \lambda \\
0, & \text{otherwise}
\end{cases}$$

- note that latent dimensions $z_n$ that have not yet learnt to represent any data generative factors, are automatically unmasked
Inferring the environment $s$

- may seem tempting to learn through amortised VI but
  - parametric learning is slow, whereas new data clusters have to be inferred very fast
- They use a heuristic instead
  - decide whether new dataset $s_{r+1}$ or one of the $r$ previous datasets
    - is current data likely under any of the known environments?
Inferring the environment $s$

- is current data likely under any of the known environments?
  - even if yes, could just be using a subset of the same factors
    - so we have to also check if generative factors are the same
  - moving triangle and moving square are different environment, even though they share the same generative factor (object position)
    - so we also have to check reconstruction accuracy

$$s = \begin{cases} \hat{s}, & \text{if } \mathbb{E}_{z^\hat{s}} [p_\theta(x^s_{\text{batch}}|z^\hat{s}, \hat{s})] \leq \kappa L_{\hat{s}} \land a^s = a^{\hat{s}} \\ s_{r+1}, & \text{otherwise} \end{cases}$$

- auxiliary classifier to infer most likely previous experiment

$$\hat{s} = \arg \max q(s|x^s_{\text{batch}})$$
Preventing catastrophic forgetting

- deep generative replay
- dreaming data from using a snapshot of VASE
- Wasserstein distance for encoder and KL divergence for decoder

\[ \mathbf{x}_{\text{old}} \sim q_{\theta_{\text{old}}} (\cdot | \mathbf{z}, s_{\text{old}}) \]

\[ s_{\text{old}} \sim p(S) \]

\[ \mathbf{z} \sim p(\mathbf{z}) \]

\[ \tilde{\mathbf{x}} \]

\[ \tilde{\mathbf{z}} \]

\[ \mathcal{L}_{\text{past}} (\phi, \theta) = \mathbb{E}_{\mathbf{z}, s', \mathbf{x}'} \left[ D[q_{\phi}(\mathbf{z} | \mathbf{x'}), q_{\phi'}(\mathbf{z'} | \mathbf{x'})] + D[q_{\theta}(\mathbf{x} | \mathbf{z}, s'), q_{\theta'}(\mathbf{x'} | \mathbf{z}, s')] \right] \]

- snapshot parameters are set to current parameters every \( \tau \) training steps
Model summary

\[
\mathcal{L}(\phi, \theta) = \mathbb{E}_{z^s \sim \phi(\cdot | x^s)} \left[ -\log p_\theta(x | z^s, s) \right] + \gamma \mathbb{KL}(q_\phi(z^s | x^s) || p(z)) - C^2 + \\
\mathbb{E}_{z, s', x'} \left[ D[q_\phi(z | x'), q_{\phi'}(z' | x')] + D[q_\theta(x | z, s'), q_{\theta'}(x' | z, s')] \right].
\]

MDL on current data

“Dreaming” feedback on past data
Results

VASE

• moving fashion → MNIST → moving MNIST
• latent traversals at the end of training seeded with samples from the three datasets
• rows correspond to latent dimensions $z_n$, columns correspond to the traversal values
• latent use progression throughout training is demonstrated in colour

Baseline
Semantic transfer

- the two domains share many semantically related factors $z$, but these are rendered into very different visuals $x$ (EDE, NatLab)
- the disentangled VASE finds semantic homologies between the two datasets (e.g. cacti - red objects, fog - wall). The entangled VASE only maps lower level statistics.
Imagination-driven exploration

- You have just learned about translation of fashion items along the x and y axis
- Now you see a number
- If your learning algorithm is compositional, you should be able to imagine how you translate that number along x and y, too!
- This is precisely what imagination-driven exploration achieves
Imagination-driven exploration

Idea:

- Train network on dataset X1, learn latents
- Obtain a sample $z^*$ from your prior $p(z)$
- Now feed in an exemplar from a new dataset X2
  - store the latent $z^*$ activated by that sample
  - Minimise the mismatch between new latent $z$ and old latent sample $z^*$, by choosing an action $g(x,z)$ that moves the new image from X2 appropriately

$$\mathcal{L}_{agent} = \mathbb{E}_{x \sim p(x)} \mathbb{E}_{z^* \sim p(z)} \mathbb{E}_{z \sim q(z|g(z^*,x),x)} \| z^* - z \|^2.$$

- Generate an output by passing the new $z$ through the decoder, let’s call this $x_2^*$
- Store this new output (translated object from X2) in your training data
Imagination-driven exploration