- Variational Continual Learning
- Cuong V. Nguyen, E. Turner

# The Problem

We are given a sequence of datasets $D_1, \ldots, D_T$. This dataset can even be for different tasks. Our goal is, upon receiving a new dataset $D_t$, update our model from $m_{t-1}$, to $m_t$, so that it adapts to the new dataset, but **still works for all previous datasets**. This is continual learning.

# The Idea

Assume that datasets and datapoints are independent (but they can have different distributions, although they may share parameters). Then, $p(\theta|D_{1:T})$ can be decomposed as

$$p(\theta|D_{1:T}) \propto p(D_{1:T}|\theta)p(\theta) = p(D_{1:T-1}|\theta)p(D_T|\theta)p(\theta) = p(\theta|D_{1:T-1})p(D_T|\theta)$$

Here, $p(\theta|D_{1:T})$ will be our model that summarizes $D_{1:T}$. When given $D_{T+1}$, we compute $p(\theta|D_{1:T+1})$ by multiplying $p(\theta|D_{1:T})$ with $p(D_T|\theta)$, and normarlizing.

However, $p(\theta|D_{1:T})$ is intractable. So we will use an $q_t(\theta)$ to approximate this. If we define the operator $proj$ as

$$proj(p^*(\theta)) = \arg\min_{q \in Q} KL(q(\theta) \| p^*(\theta))$$

Then we can compute $q_t(\theta)$ recursively as

$$q_t(\theta) = proj(q_{t-1}(\theta)p(D_t|\theta))$$

# Episodic Memory Enhancement

$q_t(\theta)$ is an approximation. Plus, $q_t$ is computed recursively. This means error can easily accumulate. To mitigate this problem, we extend VCL to include a small representative set of data $C_t$ for $D_{1:t}$, which we will call the "coreset". **We hope the information from this coreset will be passed to the final approximation as precise as possible**. We will decompose

$$p(\theta|D_{1:T}) = p(\theta|D_{1:T} \backslash C_t)p(C_t|\theta) = \tilde{q}_t(\theta)p(C_t|\theta)$$

In this way, only $p(\theta|D_{1:T} \backslash C_t)$ will be computed recursively. In particular, it will be computed as follows:

$$p(\theta|D_{1:T} \backslash C_t) = p(\theta|D_{1:t-1} \backslash C_{t-1})p(C_{t-1} \backslash C_t|\theta)p(D_t \backslash C_t|\theta) \approx \tilde{q}_{t-1}(\boldsymbol{\theta})p\left(\mathcal{D}_t \cup C_{t-1} \backslash C_t|\boldsymbol{\theta}\right)$$

You can just draw a Venn graph to see why this is true.

**Algorithm 1** Coreset VCL

**Input:** Prior $p(\boldsymbol{\theta})$.

**Output:** Variational and predictive distributions at each step $\{q_t(\boldsymbol{\theta}), p(y^*|\boldsymbol{x}^*, \mathcal{D}_{1:t})\}_{t=1}^{T}$.

Initialize the coreset and variational approximation: $C_0 \leftarrow \emptyset$, $\tilde{q}_0 \leftarrow p$.

**for** $t = 1 \dots T$ **do**

Observe the next dataset $\mathcal{D}_t$.

$C_t \leftarrow$ update the coreset using $C_{t-1}$ and $\mathcal{D}_t$.

Update the variational distribution for non-coreset data points:

$$\tilde{q}_t(\boldsymbol{\theta}) \leftarrow \arg\min_{q \in \mathcal{Q}} \mathrm{KL}\big(q(\boldsymbol{\theta}) \,\|\, \tfrac{1}{Z} \tilde{q}_{t-1}(\boldsymbol{\theta})\, p(\mathcal{D}_t \cup C_{t-1} \setminus C_t|\boldsymbol{\theta})\big). \tag{2}$$

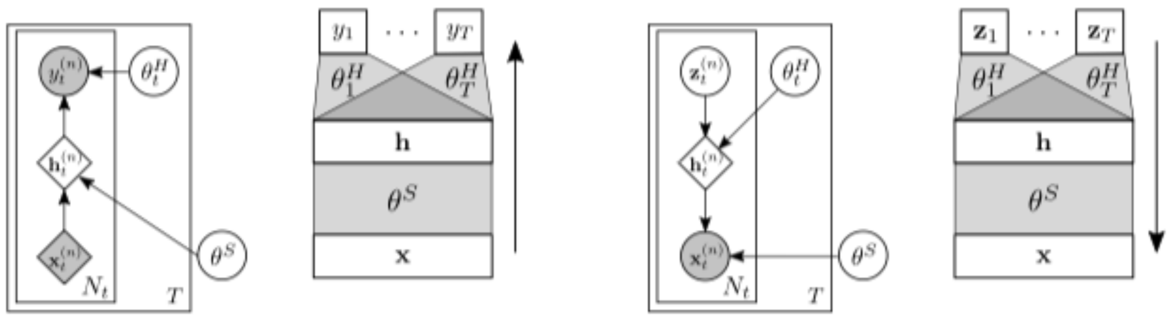
Compute the final variational distribution (only used for prediction, and not propagation):

$$q_t(\boldsymbol{\theta}) \leftarrow \arg\min_{q \in \mathcal{Q}} \mathrm{KL}\big(q(\boldsymbol{\theta}) \,\|\, \tfrac{1}{Z} \tilde{q}_t(\boldsymbol{\theta}) p(C_t|\boldsymbol{\theta})\big). \tag{3}$$

Perform prediction at test input $\boldsymbol{x}^*$: $p(y^*|\boldsymbol{x}^*, \mathcal{D}_{1:t}) = \int q_t(\boldsymbol{\theta}) p(y^*|\boldsymbol{\theta}, \boldsymbol{x}^*) \mathrm{d}\boldsymbol{\theta}$.

**end for**

# VCL in Discriminative Models




(a) multi-head discriminative network (b) multi-head generative network

There are two cases:

1. All datapoints come from the same distribution
2. Datapoints in different datasets come from different distributions, **and even the output can differ**.

For the first case, we can just use a single-head network. For the second, there will be multiple heads. **Note, here we will define different $p(y|x,\theta)$ for different tasks.** $\theta$ will include a shared part $\theta^S$, but will also include task specific part $\theta_t^H$. This weights are only used in their $p(y|x,\theta)$, which means that they will be updated only once.

# VCL in Generative Models

We assume a single $p(z)$. So $p(x|z,\theta)$ must differ for different tasks. Other things are just trivial.

# Experiments

Dicriminative

- Permuted-MINST: trivial. Single head-network
- Split-MNIST. 0/1, 2/3, 4/5. We must use multi-head network

Generative

- MNIST: 0, 1, 2, 3, ...

- Characters.