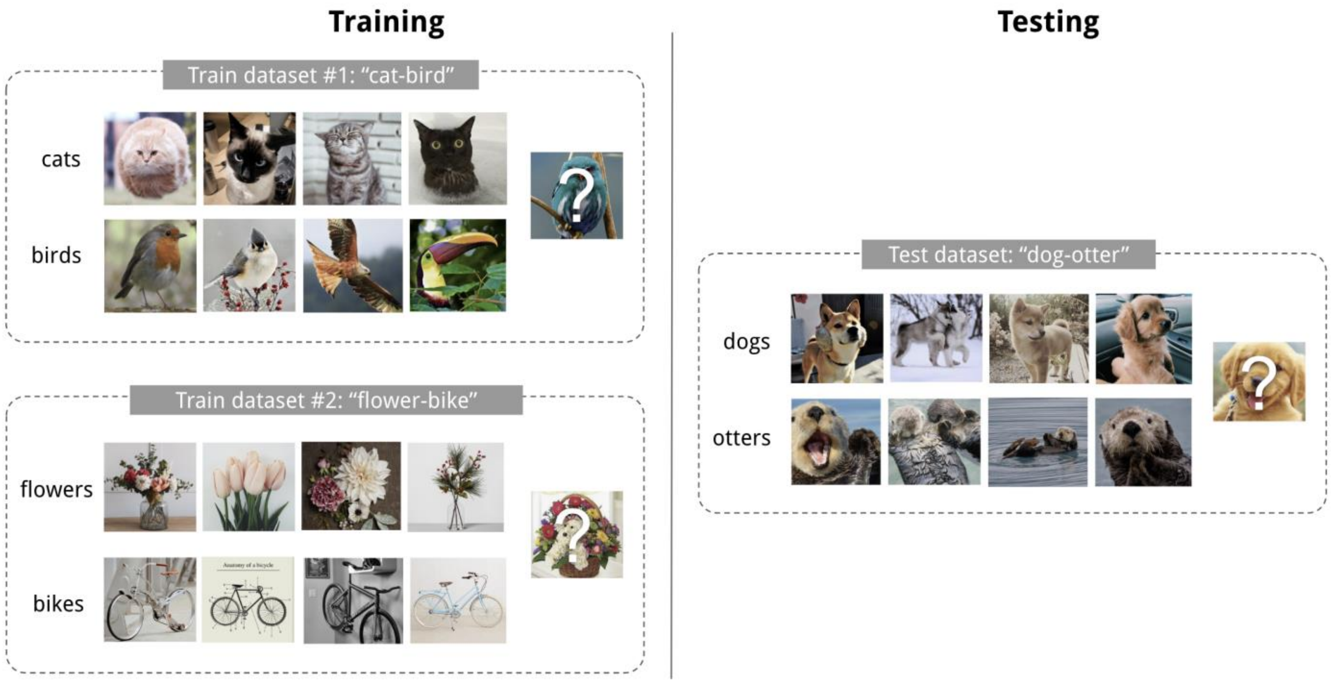
Prototypical Network for Few-Shot Learning

Jake Snell, Kevin Swersky, Richard S. Zemel

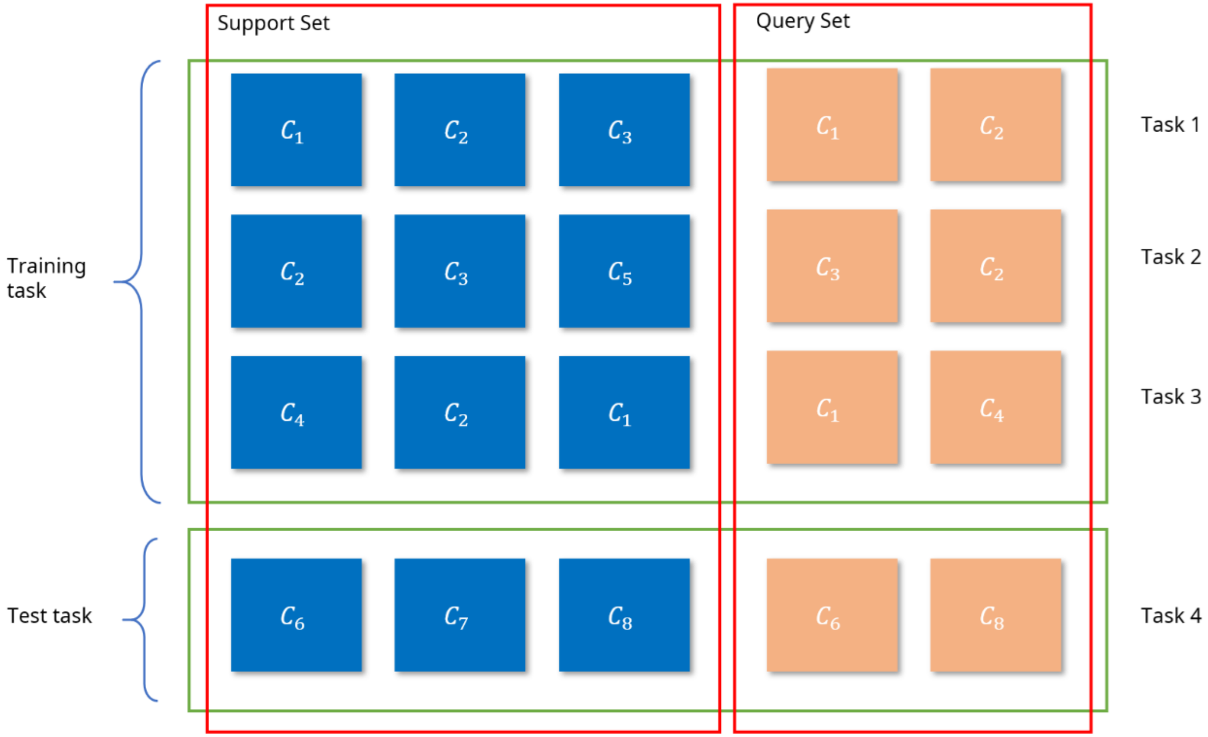
小样本学习

- 小样本学习Few-shot classification——在小样本中学习需要的知识



元学习

- 元学习Meta Learning——学习如何学习leaning to learn



给定的新样本被称之为 Support set

相应的测试集合被称之为 Query set

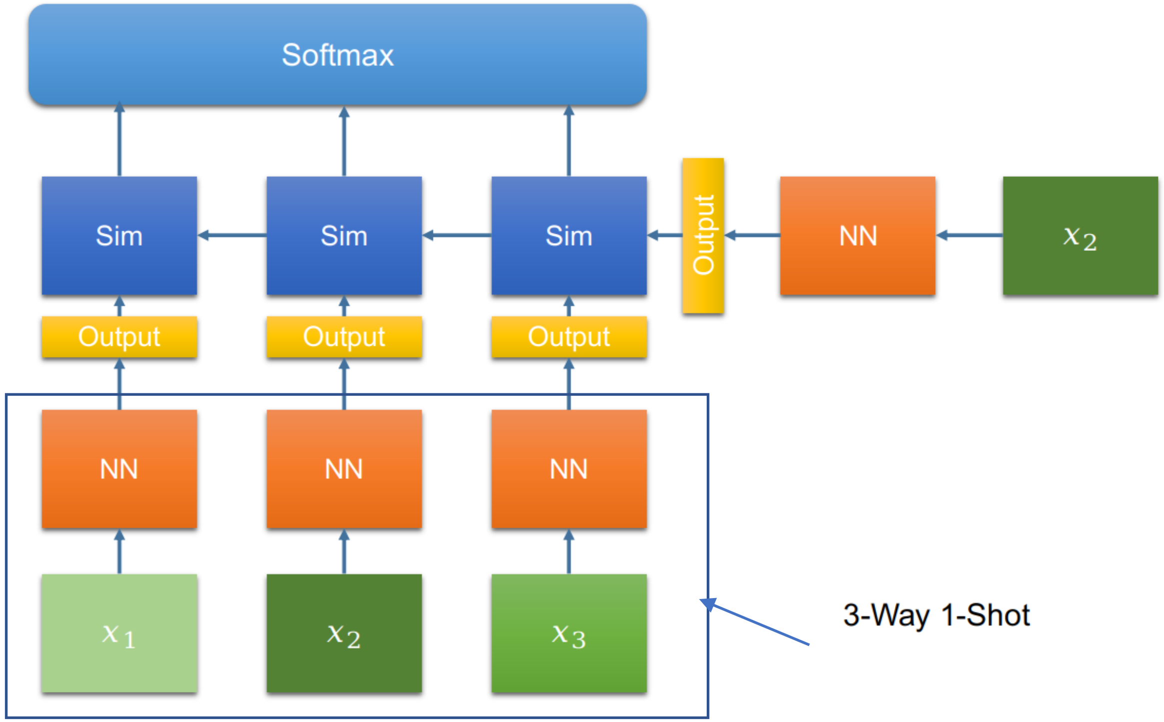
每训练一次被称为一个 episode

用来训练 F 的 Task 被称之为 Training task,

验证F有效性的被称为 Testing task

N-Way-M-Shot

- 元学习基于一个 Task 的支持集中给出的 N 个新类(Way) M 个实例（shot）中学到一个模型 F



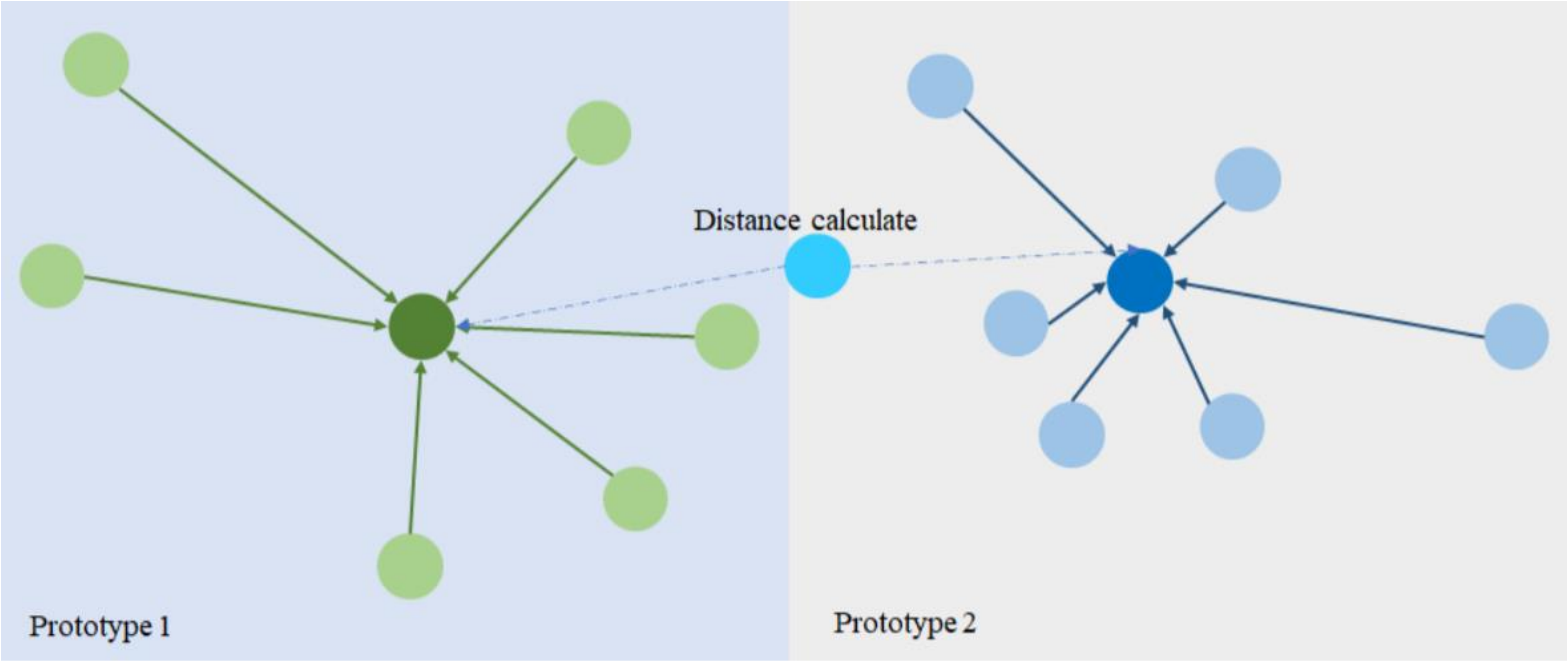
- 3-Way 1-Shot示意图，x1, x2，x3表示三个类别的数据，每个类别只有一个数据，右侧x2表示测试数据

元学习分类

- **基于度量的方法：Matching Network、Prototypical Network**
- 基于模型的方法：MANN、Meta Networks
- 基于优化的方法：MAML、Reptile

1. Chang J R, Chen Y S. Pyramid stereo matching network[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 5410-5418.
2. Snell J, Swersky K, Zemel R S. Prototypical networks for few-shot learning[J]. arXiv preprint arXiv:1703.05175, 2017.
3. Santoro A, Bartunov S, Botvinick M, et al. Meta-learning with memory-augmented neural networks[C]//International conference on machine learning. PMLR, 2016: 1842-1850.
4. Munkhdalai T, Yu H. Meta networks[C]//International Conference on Machine Learning. PMLR, 2017: 2554-2563.
5. Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks[C]//International Conference on Machine Learning. PMLR, 2017: 1126-1135.
6. Nichol A, Achiam J, Schulman J. On first-order meta-learning algorithms[J]. arXiv preprint arXiv:1803.02999, 2018.

原型网络



对于一个类别k而言，所有属于k类别的数据的集合记作 S_k

⚠ 公式细节

原型中心点：

$$c_k = \frac{1}{|S_k|} \sum_{(x_i,y_i)\in S_k} f_\phi\left(x_i\right)$$

X属于原型k的概率：

$$p_\phi\left(y = k|x\right) = \frac{\exp(-d(f_\phi(x), c_k))}{\sum_{k'} \exp\left(-d(f_\phi\left(x\right), c_{k'})\right)}$$

https://blog.waynehfut.com/2020/11/02/prototypical_network_for_few_shot_learning/

算法细节

Algorithm 1 Training episode loss computation for Prototypical Networks. N is the number of examples in the training set, K is the number of classes in the training set, $N_C \leq K$ is the number of classes per episode, N_S is the number of support examples per class, N_Q is the number of query examples per class. RANDOMSAMPLE(S, N) denotes a set of N elements chosen uniformly at random from set S, without replacement.

Input: Training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where each $y_i \in \{1, \ldots, K\}$. $\mathcal{D}_k$ denotes the subset of $\mathcal{D}$ containing all elements $(\mathbf{x}_i, y_i)$ such that $y_i = k$.

Output: The loss J for a randomly generated training episode.

$V \leftarrow \text{RANDOMSAMPLE}(\{1, \ldots, K\}, N_C)$ ▷ Select class indices for episode

for k in $\{1, \ldots, N_C\}$ **do**

$S_k \leftarrow \text{RANDOMSAMPLE}(\mathcal{D}_{V_k}, N_S)$ ▷ Select support examples

$Q_k \leftarrow \text{RANDOMSAMPLE}(\mathcal{D}_{V_k} \setminus S_k, N_Q)$ ▷ Select query examples

$$\mathbf{c}_k \leftarrow \frac{1}{N_C} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_\phi(\mathbf{x}_i)$$ ▷ Compute prototype from support examples

end for

$J \leftarrow 0$ ▷ Initialize loss

for k in $\{1, \ldots, N_C\}$ **do**

for $(\mathbf{x}, y)$ in Q_k **do**

$$J \leftarrow J + \frac{1}{N_C N_Q} \left[d(f_\phi(\mathbf{x}), \mathbf{c}_k)) + \log \sum_{k'} \exp(-d(f_\phi(\mathbf{x}), \mathbf{c}_{k'})) \right]$$ ▷ Update loss

end for

end for

算法细节

当前支持集所对应的原型：

$$\mathbf{c}_k \leftarrow \frac{1}{N_S} \sum_{(x_i, y_i) \in S_k} f_\phi(x_i)$$

该类所有位于Q_k数据的损失：

$$J \leftarrow J + \frac{1}{N_C N_Q} \left[d(f_\phi(x), \mathbf{c}_k) + \log \sum_{k'} \exp(-d(f_{\phi(x)}, c_{k'})) \right]$$

推导：

$$J_i^k = -\log\left(\frac{\exp(-d(f_\phi(x_i), c_k))}{\sum_{j=1}^{NQ} \exp(-d f_\phi(x_i), c_j)} \right) = d(f_{\phi(x_i)}, c_k) + \log \sum_{j=1}^{NQ} \exp(-d(f_\phi(x_i), c_j))$$

原理

- 原型网络视作混合概率密度估计
- 原型实际上是度量两个不同分布中数据的距离
- Bergman 散度-距离计算概率

$$d_{\varphi}(z, z') = \varphi(z) - \varphi(z') - (z - z')\nabla\varphi(z')$$

$\nabla\varphi(z')$表示函数$\varphi(x)$在z'处的梯度

$$p_{\psi}(z|\theta) = \exp\{z^{T}\theta - \psi(\theta) - g_{\psi}(z)\} = \exp\{-d_{\varphi}(z, \mu(\theta)) - g_{\varphi}(z)\}$$

$$p(z \mid \Gamma) = \sum^{K} \pi_{k} p_{\psi}\left(z \mid \theta_{k}\right) = \sum^{K} \pi_{k} \exp\left(-d_{\varphi}\left(z, \mu\left(\theta_{k}\right)\right) - g_{\varphi}(z)\right)$$

$$p(y = k \mid z) = \frac{\pi_{k} \exp\left(-d_{\varphi}\left(z, \mu\left(\theta_{k}\right)\right)\right)}{\sum_{k'} \pi_{k'} \exp\left(-d_{\varphi}\left(z, \mu\left(\theta_{k}\right)\right)\right)}$$

⚠ 原理

- 计算的线性原理
- 使用欧氏距离时:

$$d(z, z') = \|z - z'\|$$

$$-d\left(f_\phi(x), c_k\right) \longrightarrow -\left\|f_\phi(x) - c_k\right\|^2 = \boxed{-f_\phi(x)^\top f_\phi(x)} + 2c_k^\top f_\phi(x) - c_k^\top c_k$$

$$2c_k^\top f_\phi(x) - c_k^\top c_k = w_k^\top f_\phi(x) + b_k \quad , w_k = 2c_k, \ b_k = -c_k^\top c_k$$

- 学习的就是一个图嵌入

Reference

- Lilian Weng. Meta-Learning: Learning to Learn Fast. https://lilianweng.github.io/lil-log/2018/11/30/meta-learning.html. 2018-11-30
- Cyprien NIELLY. Few-shot Learning with Prototypical Networks. https://towardsdatascience.com/few-shot-learning-with-prototypical-networks-87949de03ccd
- Daisukelab. Prototypical Networks as a Fine Grained Classifier. https://www.kaggle.com/c/humpback-whale-identification/discussion/81085
- Hung-yi Lee. Meta Learning https://www.youtube.com/watch?v=EkAqYbpCYAc

Q&A

Hao Wang

https://blog.waynehfut.com/2020/11/02/prototypical_network_for_f
ew_shot_learning/