UNrolled Generalized EM for Transductive Few-Shot Learning

Application to Bowel Obstruction diagnosis.

Abdelbaki GUIR, Kaiyuan XU

M2 ATSI
Paris-Saclay University

Supervisors : Pr.Emilie Chouzenoux, Dr.Quentin VANDERBECQ.

Table of Contents

Table of Contents

Introduction

- Deep learning has revolutionized medical imaging classification, enabling automated diagnosis with high accuracy.
- What if we have a limited labeled medical images,

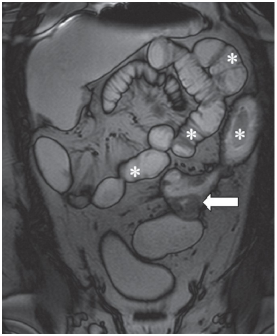

Figure: Small Bowel Obstruction [1]

[1] Lienemann, A., Kirchhoff, S. (2010). MRI of Adhesions and Small Bowel Obstruction.

Table of Contents

2.1 Few-shot learning (FSL)

Few-shot learning (FSL) focuses on training models to generalize to new tasks with very **few labeled** examples.

- **Support set:** Few labeled examples for training. ($\mathbb{S} \subset \{1, \ldots, N\}$)
- **Query set:** Unlabeled examples to classify. ($\mathbb{Q} \subset \{1, \ldots, N\} \setminus \mathbb{S}$)
- **Inductive FSL:** Predicts each query sample independently of others.
- **Transductive FSL:** Uses the statistics of all unlabeled query samples collectively for improved accuracy.

2.2 Hyperparameters for Regularization in Transductive FSL

Class Balance λ : Shannon entropy of class distribution (Class Balance) $\Psi(u)$

Temperature Scaling T : Entropic barrier (Prediction entropy) $\Phi(u)$

2.3 Unrolling

Manually tuning these hyperparameters is computationally expensive and dataset-specific. $\rightarrow$ learning to optimize

- Parameters of the algorithm become trainable variables in the network.
- Maps each iteration of an **algorithm** to a neural network layer.

3.4 Maximum likelihood - Expectation Maximization (EM) Algorithm

E-step

$$\text{Expectation } Q(\theta|\theta^{(p)}) = \mathbb{E}\left[L(\theta, n)\middle| m, \theta^{(p)}\right], \text{with } L(\theta, n) := \log p(n|\theta)$$

M-step

$$\text{Maximization } \theta^{(p+1)} = \arg\max_{\theta \in \Theta} Q(\theta|\theta^{(p)})$$

2.5 Generalized

Gaussian distribution : image (vision)

$$p(\boldsymbol{z}_n \mid \boldsymbol{\theta}_k) \propto \exp\left(-\frac{1}{2}\|\boldsymbol{z}_n - \boldsymbol{\theta}_k\|^2\right).$$

Dirichlet distribution : image + text (vision-language)

$$p(\boldsymbol{z}_n \mid \boldsymbol{\theta}_k) = \frac{1}{\mathcal{B}(\boldsymbol{\theta}_k)} \prod_{i=1}^{K} z_{n,i}^{\theta_{k,i}-1} \mathbb{I}_{\boldsymbol{z}_n \in \Delta_K},$$

with the normalization factor $\mathcal{B}(\boldsymbol{\theta}_k)$

$$\mathcal{B}(\boldsymbol{\theta}_k) = \frac{\prod_{i=1}^{K} \Gamma(\theta_{k,i})}{\Gamma\left(\sum_{i=1}^{K} \theta_{k,i}\right)}.$$

Notations

$\mathbb{S} \subset \{1, \ldots, N\}$: Indices of samples within Support (labeled)
$\mathbb{Q} \subset \{1, \ldots, N\} \setminus \mathbb{S}$: Indices of samples within Query (unlabeled)
y_n : One-hot-encoding of the label
z_n : Feature vectors
$u_{n,k}$: Soft-Assignment vectors (Probability that the n-th sample belongs to k-th class)
$p(\boldsymbol{z}_n \mid \boldsymbol{\theta}_k)$: Pdf of z_n given that it belongs to class k , with θ_k the Feature distribution parameter

i

Problem formulation

$$\underset{\mathbf{u},\boldsymbol{\theta}}{\text{minimize}} \quad \mathcal{L}(\mathbf{u},\boldsymbol{\theta}) + \lambda\Psi(\mathbf{u}) + T\Phi(\mathbf{u}),$$

$$\text{subject to} \quad \mathbf{u}_n \in \Delta_K \quad \forall n \in \mathcal{Q},$$

$$u_{n,k} = y_{n,k} \quad \forall n \in \mathcal{S}, \forall k \in \{1, \ldots, K\}.$$

Terms:

1. **Negative Log-Likelihood ($L(u, \theta)$):** Fits query data (z_n) to class distributions (θ_k):

$$L(u, \theta) = -\sum_{n=1}^{N} \sum_{k=1}^{K} u_{n,k} \ln p(z_n | \theta_k)$$

2. **Class Balance Regularization ($\Psi(u)$):** Promotes balanced clustering:

$$\Psi(u) = -\sum_{k=1}^{K} \pi_k \ln \pi_k, \quad \pi_k = \frac{1}{|Q|} \sum_{n \in Q} u_{n,k}$$

3. **Entropy Regularization ($\Phi(u)$):** Encourages soft, non-confident assignments:

$$\Phi(u) = \sum_{n=1}^{N} \sum_{k=1}^{K} u_{n,k} \ln u_{n,k}$$

Remarks

1. $T = 1$ and $p(\boldsymbol{z}_n \mid \boldsymbol{\theta}_k) \sim$ Dirichlet distribution : Transductive FSL
2. $T = 1$ and $\lambda = \frac{1}{|\mathbb{Q}|}$: EM algorithm

Algorithm 1 GEM based few-shot classification algorithm

Input: Compute $\boldsymbol{z}_n$ for the dataset samples, initialize $\boldsymbol{u}_n^{(0)}$ and $\boldsymbol{\theta}_k^{(0)}$, and fix the number of iterations L,
for $\ell = 0, 1, \ldots, L-1$ **do**
// Update Distribution parameters for each class using a given an estimation algorithm (denoted here by "DP_est")

$$\boldsymbol{\theta}_k^{(\ell+1)} = \text{DP\_est}(\boldsymbol{u}_{\cdot,k}^{(\ell)}, \boldsymbol{\theta}_k^{(\ell)}), \quad \forall k \in \{1, \ldots, K\},$$

// Update class proportions

$$\pi_k^{(\ell+1)} = \frac{1}{|\mathbb{Q}|} \sum_{n \in \mathbb{Q}} u_{n,k}^{(\ell)}, \quad \forall k \in \{1, \ldots, K\},$$

// Update assignment vectors for all query samples

$$\boldsymbol{u}_n^{(\ell+1)} = \text{softmax}\left(\frac{1}{T}\left(\ln \mathrm{p}\left(\boldsymbol{z}_n \mid \boldsymbol{\theta}_k^{(\ell+1)}\right) + \frac{\lambda}{|\mathbb{Q}|}\ln(\pi_k^{(\ell+1)})\right)_k\right), \quad \forall n \in \mathbb{Q}.$$

end for

Figure: GEM based few-shot classification algorithm. cf. p.5 (`https://arxiv.org/abs/2412.16739v1`)

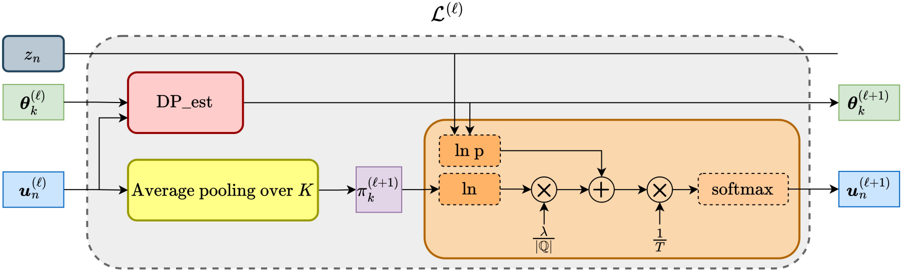

Figure: An overview of the unrolled GEM algorithm for a given iteration. cf. p.5 (https://arxiv.org/abs/2412.16739v1)

Each EM iteration **becomes** a neural network layer :

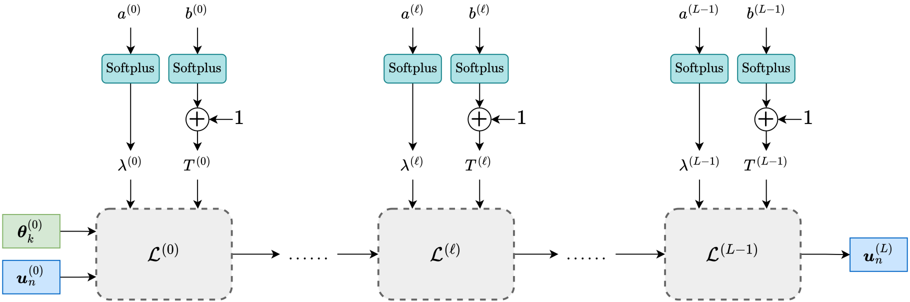

Figure: Overall architecture of the designed UNEM. cf. p.6 (`https://arxiv.org/abs/2412.16739v1`)

$a^{(\ell)}$: a learnable parameter for the unrolled architecture

$$\lambda^{(\ell)} = \text{Softplus}(a^{(\ell)}) = \log(1 + \exp(a^{(\ell)}))$$

$b^{(\ell)}$: a parameter that needs to be learned during the training of the unrolled model

$$T^{(\ell)} = 1 + \text{Softplus}(b^{(\ell)}) = 1 + \log(1 + \exp(b^{(\ell)}))$$

Table of Contents

Dataset creation

- The provided dataset was a result of passing Resnet feature extractor bunch of abdominal CT scans, to get a tabular dataset for the classification,

Dataset creation

- The provided dataset was a result of passing Resnet feature extractor bunch of abdominal CT scans, to get a tabular dataset for the classification,

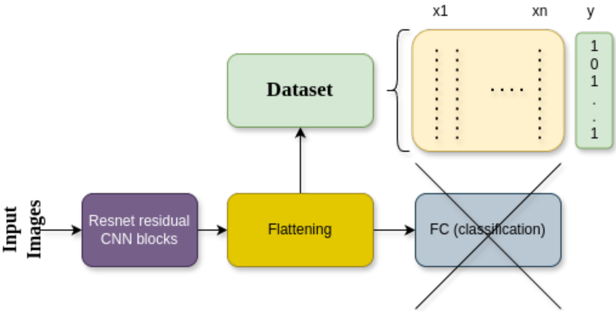

Figure: Dataset creation diagram.

Task generation

For our tasks we will be creating 3 differents tasks :

- Support Set: A small number of labeled examples for each class (5, 10, 20 shots).

Task generation

For our tasks we will be creating 3 differents tasks :

- Support Set: A small number of labeled examples for each class (5, 10, 20 shots).
- Query Set: Unlabeled examples that will be fixed at **2** exampels.

Task generation

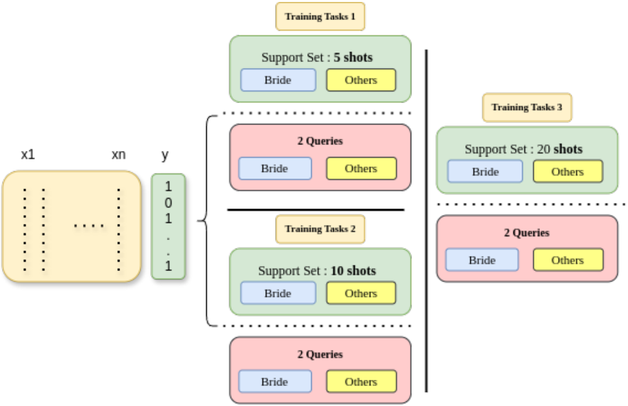

Figure: Tasks generations from the dataset.

Task generation

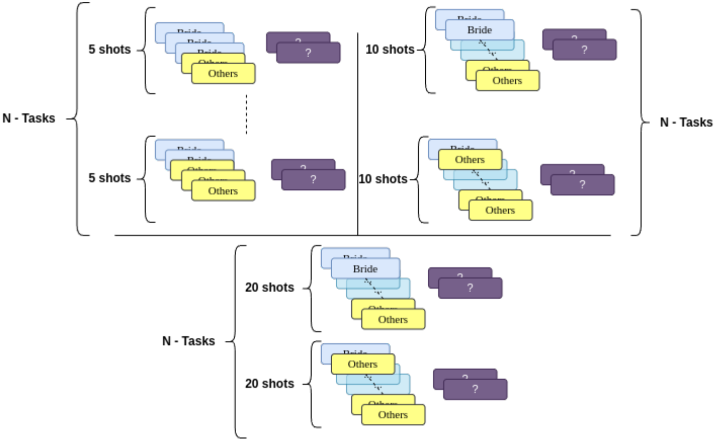

Figure: Tasksfor each shot.

Training parameters

Below is training parameters for the UNEM model :

Training parameters

Below is training parameters for the UNEM model :

- n_shots = [5,10,20] $\rightarrow$ number of shots in the support sets,

Training parameters

Below is training parameters for the UNEM model :

- n_shots = [5,10,20] $\rightarrow$ number of shots in the support sets,
- k_eff = 2 $\rightarrow$ number of classes exisiting in the query set,

Training parameters

Below is training parameters for the UNEM model :

- n_shots = [5,10,20] $\rightarrow$ number of shots in the support sets,
- k_eff = 2 $\rightarrow$ number of classes exisiting in the query set,
- k_ways = 2 $\rightarrow$ number of classes exisiting in the support set,

Training parameters

Below is training parameters for the UNEM model :

- n_shots = [5,10,20] $\rightarrow$ number of shots in the support sets,
- k_eff = 2 $\rightarrow$ number of classes exisiting in the query set,
- k_ways = 2 $\rightarrow$ number of classes exisiting in the support set,
- queries = [2] $\rightarrow$ number of query samples in query sets.

Training parameters

Below is training parameters for the UNEM model :

- n_shots $=$ [5,10,20] $\rightarrow$ number of shots in the support sets,
- k_eff $= 2 \rightarrow$ number of classes exisiting in the query set,
- k_ways $= 2 \rightarrow$ number of classes exisiting in the support set,
- queries $=$ [2] $\rightarrow$ number of query samples in query sets.
- device $=$ 'cuda'

Training parameters

Below is training parameters for the UNEM model :

- n_shots = [5,10,20] → number of shots in the support sets,
- k_eff = 2 → number of classes exisiting in the query set,
- k_ways = 2 → number of classes exisiting in the support set,
- queries = [2] → number of query samples in query sets.
- device = 'cuda'
- sampling = 'balanced' → Balanced classes in the query and support sets.

Table of Contents

- The model was trained in **17.44** seconds

Shots	Accuracy Score	Confusion Matrix	AUC
5	0.42	$\begin{bmatrix} 25 & 37 \\ 50 & 38 \end{bmatrix}$	0.42
10	0.466	$\begin{bmatrix} 34 & 39 \\ 41 & 36 \end{bmatrix}$	0.466
20	0.86	$\begin{bmatrix} 59 & 5 \\ 16 & 70 \end{bmatrix}$	0.868

Table: Performance Metrics for Different tasks on the testing set.

Discussion

-

Thanks for your attention.