ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

Khoa Khoa học Máy tính

BÁO CÁO ĐỒ ÁN CUỐI KỲ

FEW-SHOT LEARNING FOR CLASSIFYING HANDWRITTEN TEXT IMAGES

Môn học: Nhập môn thị giác máy tính

Lớp: CS231.N21.KHTN

Giảng viên: TS. Mai Tiến Dũng

Thành viên nhóm:

20520835 – Phạm Nguyễn Xuân Trường

TP. Hồ Chí Minh, tháng 6 năm 2023

Mục lục

1. Đặt vấn đề

Việc thu thập và gán nhãn dữ liệu chữ viết tay có thể tốn kém và công phu. Điều này đặc biệt đúng trong các ứng dụng như nhận dạng chữ ký hoặc OCR (Optical Character Recognition) trong việc chuyển đổi hình ảnh chữ viết tay thành văn bản điện tử.

Đặc biệt, chữ viết tay có tính đa dạng cao. Mỗi người có phong cách viết riêng và có thể thay đổi trong các điều kiện khác nhau. Hơn nữa, các chữ cái và từ ngữ mới có thể xuất hiện mà không có sẵn trong dữ liệu huấn luyện ban đầu.

Vì những lý do trên, bài toán few-shot learning cho phân loại chữ viết tay là cần thiết và hứa hẹn mang lại nhiều lợi ích trong lĩnh vực nhận dạng chữ viết tay và OCR.

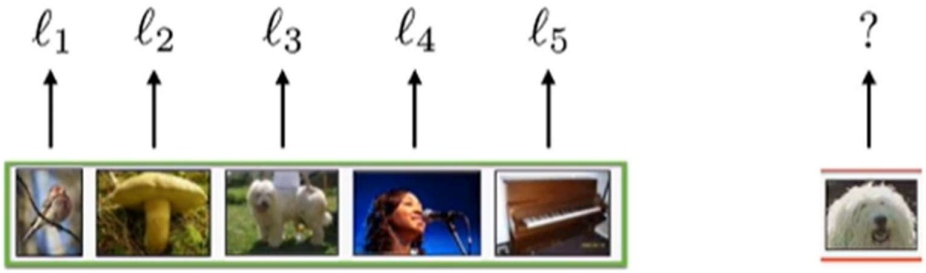
2. Giới thiệu bài toán

Few-shot images classification là một bài toán phân loại các ảnh vào các lớp khác nhau khi chỉ có một số lượng hạn chế các ví dụ huấn luyện cho mỗi lớp.

Input: Đầu vào sẽ bao gồm nhiều tác vụ, mỗi tác vụ gồm 3 phần:

- Tập dữ liệu huấn luyện (support set): Đây là tập hợp các hình ảnh huấn luyện được sử dụng để huấn luyện mô hình few-shot. Tập dữ liệu huấn luyện thường bao gồm một số lượng nhỏ các ví dụ từ mỗi lớp hoặc phân loại. Ví dụ, trong bài toán phân loại chữ cái, mỗi chữ cái có thể có chỉ vài hình ảnh mẫu.
- Tập dữ liệu kiểm tra (query set): Đây là tập hợp các hình ảnh mà chúng ta muốn phân loại dựa trên mô hình few-shot đã được huấn luyện. Tập dữ liệu kiểm tra thường chứa các hình ảnh mới mà mô hình chưa từng thấy trong quá trình huấn luyện.
- Nhãn lớp: Đối với mỗi hình ảnh trong tập dữ liệu huấn luyện, cần có nhãn lớp tương ứng để biết chúng thuộc vào lớp nào. Nhãn lớp giúp mô hình học cách phân loại chính xác các hình ảnh vào các lớp tương ứng.

Output: Phân loại các hình ảnh trong tập dữ liệu kiểm tra (query set) vào các lớp tương ứng trong từng tác vụ.



3. Lý do chọn bài toán

Few-shot images classification cho phép mô hình học máy nhận biết và phân loại các đối tượng mới mà nó chưa từng thấy trước đó. Điều này rất hữu ích trong các tình huống thực tế, khi các công ty do dự khi phải chi tiêu nhiều thời gian và tiền bạc cho dữ liệu được chú thích cho một giải pháp có thể mang lại lợi nhuận.

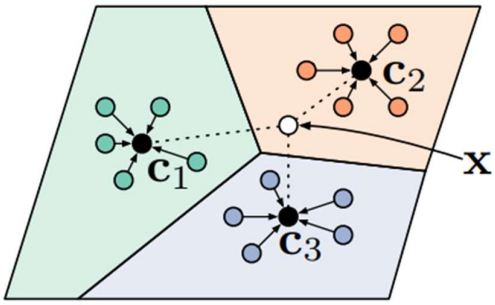
Học sâu (Deep learning) cần phải linh hoạt vì các đối tượng liên quan có thể được thay thế liên tục bằng những đối tượng mới.

Bài toán liên quan đến một số công nghệ tiên tiến như: Meta learning, GANs, Graph neural networks, …

4. Phương pháp tiếp cận

4.1 Prototypical Networks (ProtoNet)

Prototypical network là một kiến trúc mô hình trong lĩnh vực few-shot learning, được giới thiệu bởi Jake Snell, Kevin Swersky và Richard S. Zemel vào năm 2017. [1]



(a) Few-shot

Mô hình Prototypical network bao gồm các bước sau:

1. Encoding: Các ảnh trong tập dữ liệu huấn luyện (support set) được mã hóa bằng một mạng nơ-ron (thường là mạng nơ-ron tích chập) để tạo ra các biểu diễn đặc trưng.
2. Prototypical vectors (c_k): Đối với mỗi lớp trong tập dữ liệu huấn luyện, mô hình tính toán một nguyên mẫu (prototypical vector) bằng cách lấy trung bình của các biểu diễn đặc trưng của các ảnh trong lớp đó. Nguyên mẫu được coi là một biểu diễn trung tâm của lớp.

$$\mathbf{c}_k = \frac{1}{|S_k|} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_\phi(\mathbf{x}_i)$$

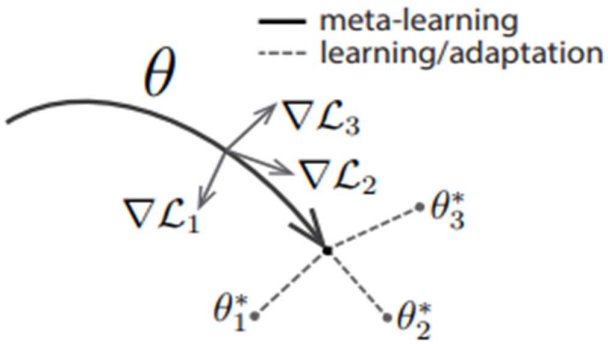
3. Phân loại: Đối với mỗi ảnh trong tập dữ liệu kiểm tra (query set), mô hình tính toán khoảng cách *d* (Euclidean, Cosine) giữa điểm dữ liệu và các nguyên mẫu. Sau đó, dự đoán được thực hiện bằng cách chọn lớp có nguyên mẫu gần nhất với ảnh dựa trên khoảng cách (có xác xuất *p* lớn nhất).

$$p_{\phi}(y = k \mid \mathbf{x}) = \frac{\exp(-d(f_{\phi}(\mathbf{x}), \mathbf{c}_k))}{\sum_{k'} \exp(-d(f_{\phi}(\mathbf{x}), \mathbf{c}_{k'}))}$$

Hàm softmax dựa trên khoảng cách d để tính xác xuất ảnh thuộc lớp k

4.2 Model-Agnostic Meta-Learning (MAML)

Model-Agnostic Meta-Learning (MAML) là một phương pháp trong lĩnh vực meta-learning, được giới thiệu bởi Chelsea Finn, Pieter Abbeel và Sergey Levine vào năm 2017. [2]



Ý tưởng chính của MAML là tạo ra một mô hình cơ bản (base model) có khả năng học cách thích ứng nhanh với các tác vụ mới.

Quá trình huấn luyện MAML gồm hai giai đoạn chính: inner loop và outer loop.

1. Inner Loop:

– Trong giai đoạn này, mô hình cơ bản được huấn luyện trên tập support set từ mỗi tác vụ.
– Mục tiêu của giai đoạn này là cập nhật các tham số của mô hình để nhanh chóng thích ứng với các tác vụ cụ thể.
– Mô hình cơ bản sử dụng gradient descent để điều chỉnh các tham số và tối ưu hóa hàm mất mát trên tập support set.

$$\min_{\theta} \mathcal{L}_T(\theta), \text{ learned via update } \left| \theta' = \theta - \alpha \cdot \nabla_{\theta} \mathcal{L}_T(\theta) \right.$$

Tối ưu hóa hàm mất mát trên tập support set ở tác vụ T

2. Outer Loop:
- Trong giai đoạn này, mô hình cơ bản được đánh giá và điều chỉnh thông qua việc tính gradient của hàm mất mát trên các tập dữ liệu từ nhiều tác vụ.
- Mục tiêu của giai đoạn này là cập nhật các tham số của mô hình cơ bản để nâng cao khả năng thích ứng với các tác vụ mới.
- Mô hình cơ bản sử dụng gradient descent trên các gradient đã tính để điều chỉnh các tham số.

$$\min_{\theta} \sum_{T \sim p(\mathfrak{I})} \mathcal{L}_T(\theta'), \text{ learned via an update}$$
$$\theta \leftarrow \theta - \beta \nabla_{\theta} \left(\sum_{T \sim p(\mathfrak{I})} \mathcal{L}_T(\theta') \right)$$

Tối ưu hóa mô hình cơ bản trên các gradient của tất cả các tác vụ

Quá trình này tạo ra một mô hình cơ bản có khả năng học cách thích ứng nhanh chóng với các tác vụ mới. Khi gặp một tác vụ mới, mô hình cơ bản có thể được điều chỉnh một cách nhanh chóng thông qua một vài bước gradient descent.

5. Thực nghiệm

5.1 Dataset

OmniGlot dataset là một tập dữ liệu phổ biến được sử dụng trong lĩnh vực học máy, đặc biệt là trong bài toán few-shot learning và meta-learning. Nó được giới thiệu bởi Lake và đồng nghiệp vào năm 2015. [3]

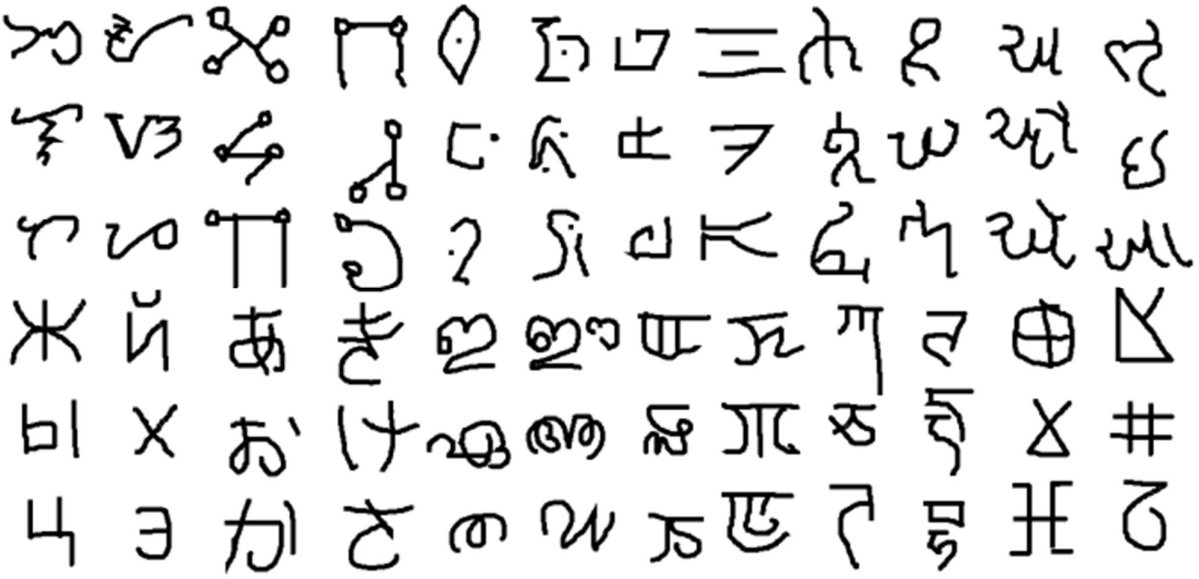
OmniGlot dataset có các đặc điểm sau:

- Số lượng ngôn ngữ: Tập dữ liệu bao gồm các ký tự từ hơn 50 ngôn ngữ khác nhau trên thế giới, bao gồm cả ngôn ngữ châu Á, ngôn ngữ Latin, và nhiều hệ thống chữ viết khác nhau như tiếng Hy Lạp, tiếng Do Thái, tiếng Hàn Quốc, v.v.

- Số lượng chữ cái và ký tự: Mỗi ngôn ngữ có một tập hợp các chữ cái hoặc ký tự riêng biệt. Tập dữ liệu OmniGlot chứa khoảng 1623 chữ cái khác nhau. Mỗi chữ cái, có 20 ví dụ có sẵn. Mỗi hình ảnh chứa 1 ký tự và có kích thước 105x105 pixel.

- Các tập Train/Validation/Test gồm 1028/172/423 lớp (ký tự)

- Độ khó và đa dạng: Tập dữ liệu OmniGlot chứa các ký tự và chữ cái có hình dạng phức tạp và đa dạng, đòi hỏi mô hình phải thích ứng và phân loại chính xác các ký tự mới.



5.2 Độ đo

Trong few-shot images classification, một trong những độ đo phổ biến để đánh giá kết quả là độ chính xác (accuracy).

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

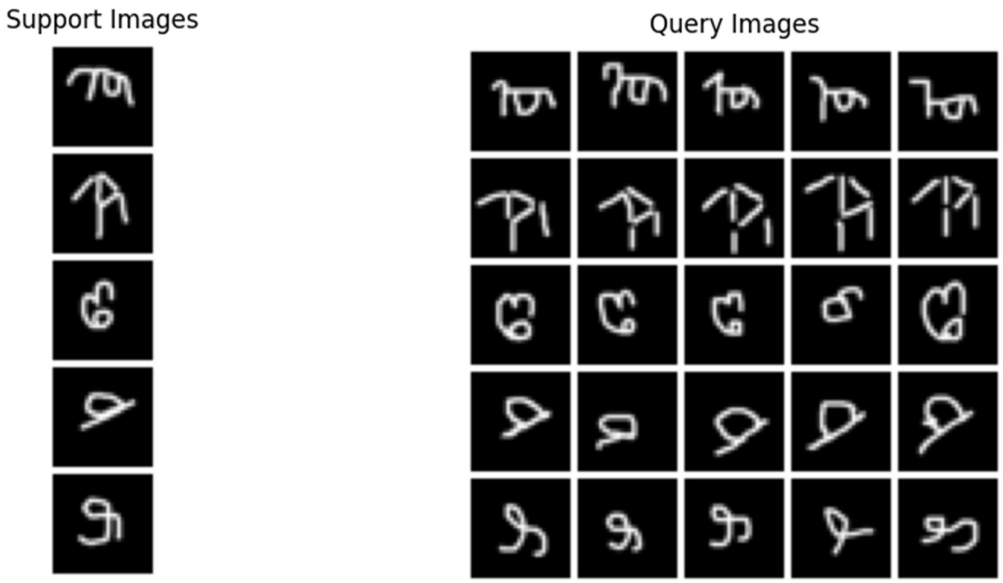
Lý do chính để sử dụng độ chính xác là bởi đây là một phép đo đơn giản và trực quan. Nó cho biết tỷ lệ các điểm dữ liệu được phân loại đúng, là một phản chiếu đáng tin cậy về hiệu suất của mô hình. Độ chính xác càng cao thì mô hình càng tốt.

5.3 Train

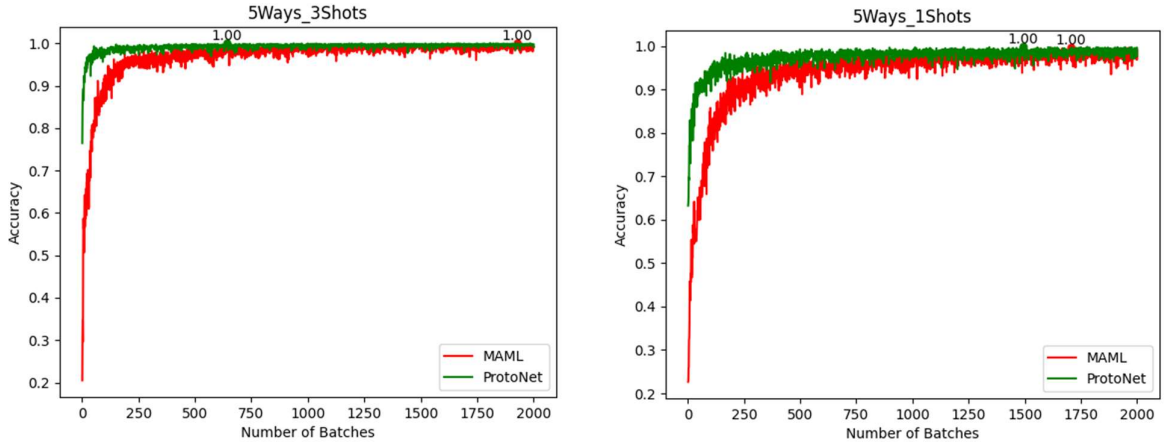
Cả hai mô hình ProtoNet và MAML đều được huấn luyện 2000 iterations, mỗi iterations có batch size là 16. Bộ dữ liệu sẽ được random trong quá trình huyến luyện.

Mục đích chọn 2000 iterations là bởi vì phần cứng máy tính có hạn chế, so với bài báo trong MAML là 60000 iterations là con số nhỏ nhưng kết quả mang lại khả quan.

Kích thước của tập support set là 5N3K (5 lớp, mỗi lớp 3 ví dụ) và 5N1K (5 lớp, mỗi lớp 1 ví dụ), tập query set sẽ bao gồm 15 ảnh (test shot) cho mỗi lớp



Ví dụ minh họa 5N1K và 5 test shot



5.4 Test

Sau khi được huấn luyện, cả hai mô hình sẽ được đánh giá trên 500 iterations đầu tiên không chọn ngẫu nhiên để đảm bảo công bằng khi đánh giá, mỗi iteration có batch size là 16.

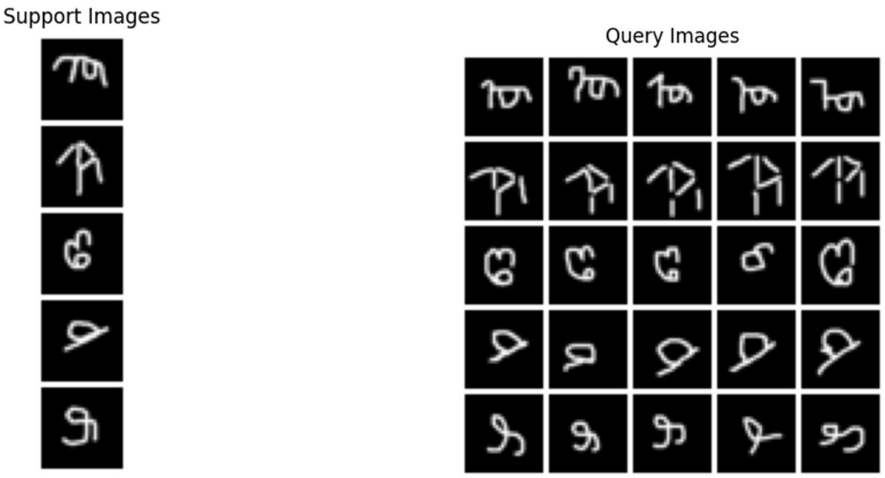
Lý do chọn 500 iterations là bởi khi chọn theo thứ tự thì có thể chọn hết được cái tất cả các ký tự trong tập test (423 ký tự).

Bảng kết quả trung bình accuracy mỗi iteration

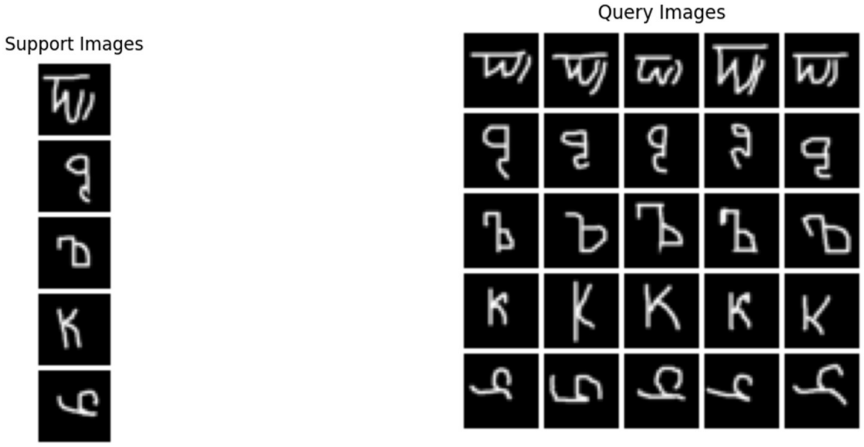
	5 Ways 1 Shot		5 Ways 3 Shots	
	ProtoNet	MAML	ProtoNet	MAML
Accuracy	0.8476	**0.9333**	**0.9629**	0.9518

5.5 Demo

Sau đây là một số minh họa cho mô hình MAML.



Support Labels: [2 1 0 4 3]
Query Labels: [2 2 2 2 2 1 1 1 1 1 0 0 0 0 0 4 4 4 4 4 3 3 3 3 3]
Predict Labels: [2 2 2 2 2 2 2 2 2 2 0 0 0 0 0 4 3 4 4 4 3 3 3 4 3]
Accuracy: 0.7200



Support Labels: [0 2 3 1 4]

Query Labels: [0 0 0 0 0 2 2 2 2 2 3 3 3 3 3 1 1 1 1 1 4 4 4 4 4]

Predict Labels: [0 0 0 0 0 2 2 2 2 2 3 3 3 3 3 1 1 1 1 1 4 4 4 4 4]

Accuracy: 1.0000

Code cho phần thực nghiệm: **https://github.com/pnxuantruong/Fewshot_Omniglot**

6. Kết luận

- Prototypical Networks đơn giản và dễ triển khai, đồng thời đạt được hiệu suất tốt trong các bài toán few-shot learning.
- MAML cho phép học cách thích ứng nhanh chóng với các tác vụ mới và không bị ràng buộc bởi một mô hình cụ thể. Nó có khả năng linh hoạt và mạnh mẽ trong việc thích ứng với nhiều bài toán và mô hình. Tuy nhiên, việc huấn luyện MAML có thể yêu cầu nhiều dữ liệu và phức tạp hơn so với Prototypical Networks vì cần hai lần tính gradient.
- Cả hai mô hình hoạt động khá tốt với bộ dữ liệu Omniglot, tuy nhiên vẫn còn có trường hợp ký tự quá phức tạp hoặc mất nét, thì vẫn chưa dự đoán tốt. Hướng phát triển tiếp theo là cải thiện hai mô hình này thêm bằng cách huấn luyện nhiều hơn.

Tài liệu

[1] Prototypical Networks for Few-shot Learning https://arxiv.org/abs/1703.05175
[2] Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks https://arxiv.org/abs/1703.03400
[3] Omniglot dataset https://github.com/brendenlake/omniglot