

# ỨNG DỤNG FISHER INFORMATION MATRIX TRONG CẮT TỈA TRỌNG SỐ MẠNG NƠ-RON ĐA NHIỆM.

CS2205.CH2023-01

Nguyễn Khánh Toàn - 240101027

# Tóm tắt

- Lớp: CS2205.CH2023-01
- Link Github của nhóm: [toannk38/CS2205.CH2023-01](https://github.com/toannk38/CS2205.CH2023-01)
- Link YouTube video: <https://youtu.be/TR5rKRufCIs>
- Ảnh + Họ và Tên của các thành viên:



Nguyễn Khánh Toàn

# Giới thiệu

- **Multi-task learning** hướng tới huấn luyện một mô hình duy nhất cho nhiều Task, nhưng việc học tuần tự thường dẫn đến hiện tượng **catastrophic forgetting**, làm giảm hiệu suất trên các Task cũ.
- **Weight pruning** (cắt tỉa trọng số) là một phương pháp giúp điều chỉnh mạng nơ-ron cho các Task mới bằng cách loại bỏ các trọng số kém quan trọng, đồng thời tối ưu hóa tài nguyên.
- **Đề xuất:** Sử dụng **Fisher Information Matrix** để định hướng quá trình cắt tỉa trọng số, nhằm bảo vệ các trọng số quan trọng cho từng tác vụ trong quá trình học tuần tự, giảm thiểu “quên” kiến thức cũ.

# Mục tiêu

- Phát triển một phương pháp dựa trên cắt tỉa trọng số để học nhiều Task tuần tự, giảm thiểu hiện tượng catastrophic forgetting.
- Sử dụng **Fisher Information Matrix** để xác định và bảo vệ các trọng số quan trọng cho từng tác vụ trong quá trình cắt tỉa.
- Đạt được hiệu quả duy trì hiệu suất tốt hơn trên các tác vụ cũ so với các phương pháp cắt tỉa truyền thống (ví dụ: dựa trên độ lớn trọng số).

# Nội dung và Phương pháp

## Hiện trạng:

**Multi-task learning:** Mục tiêu là huấn luyện một mạng nơ-ron xử lý hiệu quả nhiều Task cùng lúc hoặc tuần tự.

**Catastrophic forgetting:** Khi học tác vụ mới, mạng có thể quên kiến thức từ các Task cũ, làm giảm hiệu suất tổng thể.

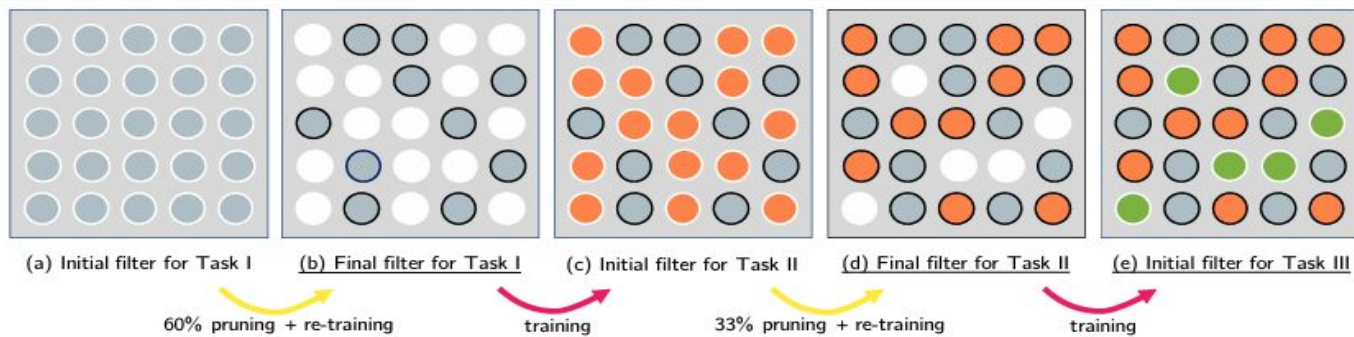
Các giải pháp hiện có bao gồm:

- Regularization (ví dụ: Elastic Weight Consolidation - EWC).
- Rehearsal (sử dụng lại dữ liệu cũ).
- Architectural methods (phân bổ tài nguyên riêng cho từng tác vụ).

# Nội dung và Phương pháp

## Packnet - weight pruning

- Sử dụng cắt tỉa lặp (iterative pruning) để phân bổ tài nguyên mạng cho nhiều Task.
- Quy trình: Huấn luyện trên một tác vụ, cắt tỉa các trọng số có độ lớn nhỏ, cố định các trọng số còn lại cho Task đó, sau đó tiếp tục huấn luyện Task mới với các trọng số chưa sử dụng.
- Hạn chế: Dựa vào độ lớn trọng số có thể không phản ánh chính xác tầm quan trọng thực sự của trọng số.



Hình 1: Weight Pruning trong multi-task learning

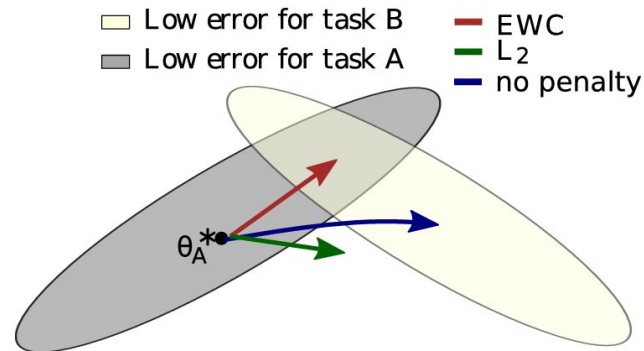
# Nội dung và Phương pháp

**Fisher Information Matrix:** Một thước đo thống kê đánh giá độ nhạy của đầu ra mô hình đối với sự thay đổi của các trọng số, phản ánh tầm quan trọng của chúng.

**Ứng dụng trong EWC [2]:**

- Được sử dụng để xác định các trọng số quan trọng cho các Task cũ.
- Áp dụng regularization để hạn chế thay đổi các trọng số này khi học Task mới.
- Sử dụng **xấp xỉ đường chéo** (diagonal approximation) của Fisher Information Matrix để giảm độ phức tạp tính toán

$$\mathcal{L}(\theta) = \mathcal{L}_B(\theta) + \sum_i \frac{\lambda}{2} F_i (\theta_i - \theta_{A,i}^*)^2$$



# Nội dung và Phương pháp

## Phương pháp đề xuất:

Lấy cảm hứng từ **PackNet**, nhưng thay việc cắt tỉa dựa trên độ lớn trọng số bằng **Fisher Information Matrix**.

### Quy trình cho từng tác vụ:

1. Huấn luyện mạng trên Task hiện tại, không cập nhật tham số của những Task trước..
2. Tính **Fisher Information Matrix** cho Task hiện tại.
3. Chọn **top-k trọng số** có giá trị Fisher Information cao nhất sử dụng cho Task hiện tại. (Bước pruning)
4. Tiếp tục huấn luyện những Task tiếp theo trên những weights còn lại.



# Kết quả dự kiến

- Dự kiến giảm thiểu hiện tượng catastrophic forgetting, duy trì độ chính xác cao trên tất cả các tác vụ đã học.
- **Fisher Information Matrix** có thể xác định tầm quan trọng của trọng số chính xác hơn so với độ lớn, dẫn đến hiệu suất tốt hơn so với phương pháp cắt tỉa truyền thống như **PackNet**.
- Phương pháp khả thi về mặt tính toán nhờ sử dụng xấp xỉ đường chéo của Fisher Information Matrix.
- Đánh giá tiềm năng trên các tập dữ liệu chuẩn (ví dụ: MNIST permutations) để so sánh hiệu quả.

# Tài liệu tham khảo

[1] Arun Mallya, Svetlana Lazebnik: PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning. CVPR 2018: 7765-7773

[2] James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, Raia Hadsell: Overcoming catastrophic forgetting in neural networks. PNAS 2017: 3521-3526