TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH

TRẢ LỜI CÂU HỎI

VanillaNet - Sức mạnh của Tối giản trong Học Sâu

GV hướng dẫn: Nguyễn Vinh Tiệp

Nhóm 7:

Họ và tên	**MSSV**
Huỳnh Anh Dũng	22520278
Hà Huy Hoàng	22520460
Nguyễn Duy Hoàng	22520467
Lê Phước Trung	20522069

TP.HCM, ngày 9 tháng 1 năm 2025

Nhóm 8

Câu hỏi. Bạn bảo mạng Vanilla net được phát triển dựa vào base là CNN, vậy sự đột phá lớn nhất so với CNN là gì

Mạng VanillaNet lấy cảm hứng từ dạng mô hình CNN truyền thống rất nhiều, gần như là toàn bộ phần cài đặt các lớp tính toán. Điểm đột phá của nghiên cứu này nằm ở các hàm kích hoạt, không phải các lớp Convolution hay cấu trúc của nó. Việc tạo ra hàm kích hoạt có dạng chuỗi kết hợp các hàm kích hoạt con, mô hình đạt được hiệu suất tính toán cao mà vẫn đảm bảo được tính phức tạp khi biểu diễn bài toán tương đương với các mô hình học sâu có các lớp hàm kích hoạt xếp chồng.

Nhóm 9

Câu hỏi. Tại sao hàm kích hoạt lại biến đổii thành hàm tuyến tính khi số epoch tăng? Việc này có tác dụng gì trong việc giải bài toán phi tuyến?

Hàm kích hoạt trong câu hỏi và hàm kích hoạt được gọi là đột phá trong giải bài toán phi tuyến không là một. Hàm kích hoạt trong câu hỏi thuộc về biện pháp huấn luyện sâu khi cài đặt mô hình. Nhóm nghiên cứu thấy rằng việc tách các lớp Convolution trong mô hình thành hai lớp Convolution con có kích thước phù hợp và cho vào giữa hai lớp con này một hàm kích hoạt phi tuyến sẽ tăng hiệu suất huấn luyện mô hình hơn. Dĩ nhiên vì hai lớp Convolution con phải là một phép biến đổi tương đương từ lớp Convolution ban đầu, hàm kích hoạt trong câu hỏi phải có dạng $A(x) = \lambda A'(x) + (1 - \lambda)x$ với $A'(x)$ là một hàm kích hoạt phi tuyến và λ giảm liên tục từ 1 về 0 để khi kết thúc quá trình huấn luyện, hai lớp Convolution con cùng hàm kích hoạt $A(x) = x$, dễ thấy là tuyến tính, là một biến đổi tuyến tính của lớp Convolution ban đầu.

Nhóm 10

Câu hỏi. Liệu có trade-off giữa performance và kiến trúc gọn nhẹ của Vanilla net hay ko?

Vẫn tồn tại trade-off trong kiến trúc của VanillaNet, kiến trúc của VanillaNet chỉ gọn chứ không nhẹ, việc "phình" to theo cấp số nhân khối đặc trưng qua từng lớp Convolution khiến VanillaNet có số lượng tham số khá lớn; tuy vậy nhờ cài đặt đặc biệt, VanillaNet có tốc độ tính toán vượt trội, biểu hiện qua số FLOPs cực khủng khi so với các mô hình cùng hiệu quả dự đoán. Nói ngắn gọn hơn, VanillaNet gọn và mạnh chứ không nhẹ, đó là trade-off lớn nhất của mô hình.

Nhóm 11

Câu hỏi. Tại sao minimalism lại quan trọng trong học sâu? Nó mang lại lợi ích gì so với các mạng phức tạp?

Tưởng tượng trong một thời đại mà các nhà nghiên cứu đều đổ xô đi phức tạp hóa mô hình của mình lên để giải quyết các bài toán dường như không tưởng, dễ thấy được nhanh rằng chóng thôi, trước cả khi mà những bài toán đó có thể được giải quyết, người ta sẽ vấp ngay vào vũng lầy của sự giới hạn tài nguyên. Các nhà phát triển hiện nay vẫn đang không ngừng cải tiến những thiết bị phần cứng để kịp bắt theo xu hướng phát triển của các nghiên cứu tính toán vực tầm vĩ mô. Nhưng khi mà cải tiến đạt đến cả giới hạn cỡ nguyên tử, điều mà hiện nay con người đã tiệm cận hoặc thậm chí đã làm được, người ta khó hình dung được mình

sẽ làm gì tiếp để bứt phá. Hoặc liệu con người sẽ có tài nguyên vô tận để mở rộng mãi cho các công trình nghiên cứu cũng như ứng dụng hay không? Câu trả lời là không; và vì vậy, chủ nghĩa tối giản mãi luôn giữ một chỗ đứng quan trọng trong không chỉ mỗi lĩnh vực học sâu nói riêng mà cả cộng đồng nghiên cứu khoa học chung lại với nhau.

Thầy Danh

Câu hỏi. "VanillaNet có sử dụng kernel size 4x4 => có gì hay hơn so với 3x3 của các kiến trúc truyền thống như VGG? Activation Function mới có gì mới để là key contribution của VanillaNet? Có thực nghiệm trên tập Ship in satellite imaginary, kết quả tương đồng với kết quả của MobileNet"

Trong kiến trúc truyền thống của VGG, mạng này sử dụng kernel 3x3 cho một lớp Convolution nhưng lại cài đặt cả hai lớp đồng thời liên tiếp nhau để phục vụ cho mục đích trích xuất đặc trưng một cách hiệu quả hơn, điều này góp phần làm giảm tổng lượng tham số cần thiết cho mô hình hoạt động nhưng vô tình đi ngược lại với tiêu chí của VanillaNet, đó là càng đơn giản càng tốt, đơn giản theo cách giảm thiểu số lượng lớp tính toán trong phần cài đặt. Bù lại, VanillaNet phải sử dụng lớp Convolution kích thước nhân 4x4 để đối trọng cho mục tiêu tinh giản của mình khi đặt lên bàn cân với khả năng trích xuất đặc trưng của lớp. Lý do đội ngũ nghiên cứu không chọn con số lớn hơn như 5 hoặc 7 theo phỏng đoán của nhóm, dễ thấy được với cấu trúc mạng, là vì đội ngũ không muốn mô hình trở nên quá lớn về mặt tham số vì vốn với cài đặt được công bố, kiến trúc mạng đã có thể được xem là đồ sộ. Đáng chú ý hơn, chỉ có lớp Convolution trong mạng đầu tiên mang kích thước nhân 4x4, số còn lại đều được cài đặt với tham số 1x1, củng cố thêm cho suy đoán của nhóm về quyết định cài đặt của đội ngũ nghiên cứu.

Hàm kích hoạt được đội nghiên cứu đề xuất mang tính đột phá ở cách đổi mới này mang đến tính phức tạp lớn hơn so với các hàm kích hoạt phi tuyến thông thường mà không làm chậm quá trình tính toán như việc xếp chồng loạt lớp kích hoạt phi tuyến truyền thống. Hàm kích hoạt được đội ngũ đưa ra có dạng biểu thức chuỗi của các hàm kích hoạt đơn giản, là những hàm kích hoạt phi tuyến thông thường, dùng trong việc cài đặt các mạng truyền thống. Phương pháp này được đội ngũ nghiên cứu nhận định là “concurrent”, tức “đồng thời”, nghĩa là hàm kích hoạt đề xuất có thể cho ra tính phi tuyến tương đương với việc chồng hàm kích hoạt truyền thống với chỉ một phép tính.

Khi nhóm giải quyết thực nghiệm trên tập dữ liệu, VanillaNet cùng MobileNet có những biểu hiện tương đồng với nhau, thế nhưng VanillaNet cho thấy khả năng học và cả dự đoán mạnh mẽ hơn nhiều so với kiến trúc còn lại. Cả hai mô hình có cùng dạng đường đồ thị giá trị loss khi thực hiện huấn luyện nhưng VanillaNet nằm quanh trong khoảng giá trị nhỏ hơn đáng kể so với MobileNet, cho thấy được khả năng mô phỏng bài toán tốt hơn so với MobileNet. Còn xét về độ chính xác dự đoán của cả hai, MobileNet ở một mức threshold nhất định, cũng có thể dự đoán đầy đủ được toàn bộ số thuyền trong ảnh đầu vào nhưng lại đi kèm với rất nhiều dự đoán sai lệch trong những vùng khác; tuy vậy, hiện tượng này không tồn tại với VanillaNet, hoặc có, nhưng lượng sai phạm chỉ chiếm tỉ suất rất nhỏ trên tổng dự đoán thuộc phân lớp “ship” (khoảng 1-2 ảnh trên toàn cảnh).