

Báo cáo Phương pháp Nghiên cứu Khoa học

Nguyễn Xuân Vỹ - Trần Đức Huy - Nguyễn Lê Sang

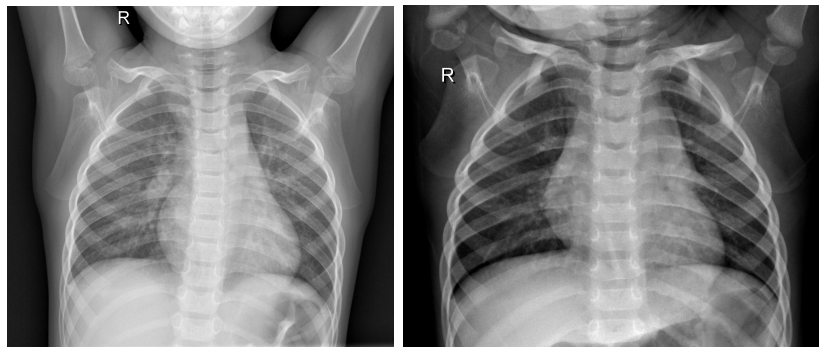
Ngày 25 tháng 7 năm 2019

Giới thiệu

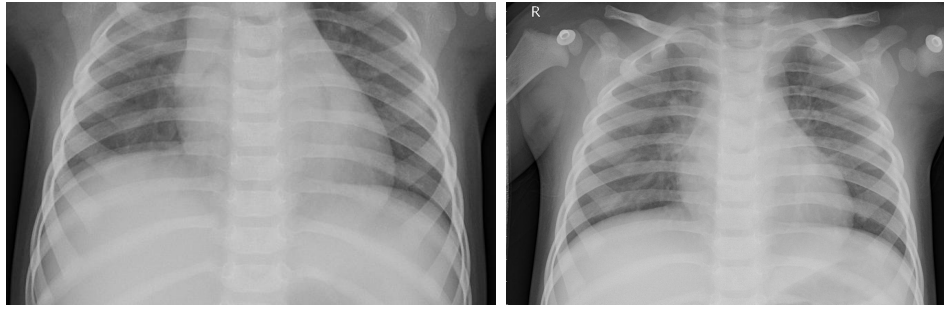
Viêm phổi là một trong những căn bệnh gây tử vong ở mọi nhóm tuổi với số ca lên đến 4 triệu người, chiếm 7% dân số thế giới mỗi năm [8]. Hiện nay, để đào tạo ra những bác sĩ chuyên ngành nhằm chuẩn đoán chính xác căn bệnh này đòi hỏi rất nhiều thời gian cho việc đào tạo cũng như xảy ra nhiều rủi ro chuẩn đoán sai với những bác sĩ chưa có nhiều năm kinh nghiệm. Mục tiêu của nhóm là học tập và thí nghiệm các phương pháp có thể hỗ trợ phán đoán dành cho các bác sĩ về căn bệnh này bằng Deep Learning. Lựa chọn một mô hình mẫu ban đầu - ResNet50 [3], thử nghiệm những phương pháp mới với mục tiêu là nhằm đề xuất cải tiến tốt hơn cho mô hình khi sử dụng tập dữ liệu CheX-Ray Images (Pneumonia) [6] trên Kaggle.

Tập dữ liệu

Tập dữ liệu được sử dụng là CheX-Ray Images (Pneumonia) [6]. Đây là tập hợp các tấm ảnh X-Quang chụp tại lồng ngực của bệnh nhân. Về mặt tổng thể, bộ dữ liệu gồm 5,863 bức ảnh được chia làm 3 phần: training, validation và testing với mỗi phần gồm 2 thư mục là Normal và Pneumonia. Trong tập training, dữ liệu nhãn Pneumonia lớn hơn dữ liệu nhãn Normal đến 3 lần, dẫn đến tình trạng dữ liệu mất cân bằng, gây khó khăn cho việc huấn luyện dữ liệu.



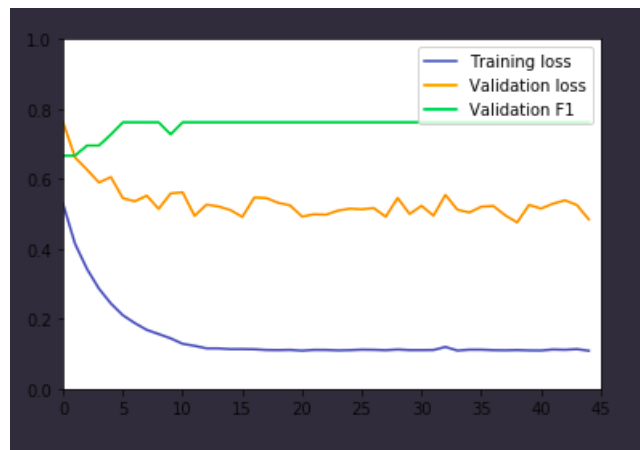
Hình 1: Ảnh về trạng thái Bình thường



Hình 2: Ảnh về trạng thái bị viêm phổi

Mô hình mẫu

Mô hình ban đầu được sử dụng để giải quyết bài toán này là ResNet50, với hàm mất mát Binary Cross Entropy, hàm tối ưu là Stochastic Gradient Descent với threshold mặc định là 0.5 (*Mã nguồn baseline của VietAI*) [7]. Hàm mất mát BCELoss hoạt động tốt với những bộ dữ liệu cân bằng (số lượng dữ liệu mỗi loại là bằng nhau) và SGD - được đánh giá là cho điểm hội tụ tốt hơn khi so sánh với những Adaptive optimization methods khác - [4]. Thế nhưng với bộ dữ liệu không cân bằng, BCELoss không thể hoạt động tốt, số lượng dữ liệu không nhiều để đạt được điểm hội tụ - giai đoạn mà SGD hoạt động không hiệu quả bằng những thuật toán mới.



Hình 3: Biểu đồ huấn luyện của baseline

Điểm chính xác của mô hình mẫu đã được là 0.771 (với threshold = 0.5). Với kết quả trên, yêu cầu phải cải thiện độ chính xác là bắt buộc.

Các thử nghiệm cải tiến

• Thử nghiệm 1

Việc threshold cố định một giá trị là không phù hợp khi ở mỗi mức khác nhau, điểm chính xác của mô hình sẽ khác nhau. Do đó nhóm lần lượt *thử nghiệm thresholds* trong khoảng từ 0 đến 1 và nhận được kết quả tốt hơn - độ chính xác là 0.869 tại threshold bằng 0.75.

Nguyên nhân của việc này là do dữ liệu mất cân đối, lệch về label 1 - bị bệnh quá nhiều khiến cho ngưỡng threshold cũng bị lệch về giá trị 1 nhiều hơn.

- **Thử nghiệm 2**

Nhóm sử dụng hàm tối ưu mới là Adam [1] nhưng không sử dụng *thử nghiệm thresholds*. Kết quả đạt được là hơn so với baseline khi độ chính xác là 0.833 với threshold mặc định 0.5. Nguyên nhân đến từ sự khác biệt giữa Adam và SGD với dataset không đủ lớn để tiến về điểm hội tụ.

- **Thử nghiệm 3**

Nhóm sử dụng transfer learning kết hợp với việc đóng băng các hyperparameters, sử dụng hàm tối ưu là Adam [1] nhưng không sử dụng *thử nghiệm thresholds*. Kết quả đạt được là tốt nhất với các thí nghiệm trước đó khi độ chính xác là 0.894 với threshold mặc định 0.5. Nguyên nhân của việc này xuất phát từ transfer learning - sử học các tham số học trước có sẵn và đóng băng chúng trừ lớp cuối, điều này đồng nghĩa với việc sử dụng thêm dữ liệu - tích hợp trong pretrained model.

- **Thử nghiệm 4**

Nhóm sử dụng transfer learning kết hợp với việc *không* đóng băng các hyperparameters, hàm tối ưu Adam [1] và không sử dụng *thử nghiệm thresholds*. Độ chính xác đạt được 0.811 với threshold mặc định 0.5. Kết quả bị giảm đi là do nguyên nhân không đóng băng hyperparameters trong khi bộ dataset không lớn, khiến việc học không tối ưu.

- **Thử nghiệm 5**

Để cải thiện tình trạng dữ liệu bị imbalance, nhóm đề sử dụng giải pháp Overstampling và BCEWithLogitsLoss thay thế cho BCELoss, kết hợp với chỉnh sửa trong thí nghiệm 3 và *thử nghiệm thresholds*. Kết quả là có được độ chính xác 0.889.

Các thử nghiệm trên các mô hình khác

- **Mô hình DenseNet121**

Sử dụng mô hình DenseNet121 [2] với transfer learning, hàm tối ưu Adam [1] và đóng băng các hyperparameters, kết hợp với overstampling và *thử nghiệm thresholds*. Kết quả là tốt hơn khi đạt được độ chính xác 0.902. Nhược điểm khi thực thi mô hình này là tốc độ học rất lâu, gấp đôi so với thời gian huấn luyện trên ResNet50.

- **Mô hình EfficientNetB2 lần 1**

Nhóm sử dụng Adam [1], tranfer learning và BCEWithLogitsLoss trên mô hình EfficientNet B2 [5]. Độ chính xác đạt được là 0.928 với thời gian huấn luyện chỉ bằng với thời gian học của ResNet50.

- **Mô hình EfficientNetB2 lần 2**

Nhóm sử dụng lại những phương pháp của Mô hình EfficientNetB2 [5] lần 1, kết hợp thêm overstampling. Độ chính xác đạt được là 0.931 với thời gian huấn luyện chỉ bằng với thời gian học của ResNet50.

Tổng kết kết quả đạt được

Nhóm đã thực hiện 12 thí nghiệm trên các mô hình nhằm tăng độ chính xác và phù hợp với tập dữ liệu. Để tăng độ chính xác, với mô hình ResNet50, nhóm đề xuất:

- Sử dụng transfer learning. Với tập dữ liệu nhỏ, việc huấn luyện các tham số khởi tạo ngẫu nhiên sẽ khiến độ chính xác không cao. Các lớp đầu của mô hình chỉ là để nhận dạng những đường - nét cơ bản nên tham số học từ tập dữ liệu nào cũng sẽ như nhau, nên việc sử dụng transfer learning sẽ cải thiện độ chính xác.
- Đóng băng các tham số trừ lớp cuối. Vì dữ liệu không đủ lớn nên việc học các tham số sẽ không hiệu quả.
- Lựa chọn threshold phù hợp. Với tập dữ liệu phức tạp, ở những mức threshold khác nhau sẽ cho ra kết quả khác nhau. Để cải thiện độ chính xác thì cần lựa chọn sau khi đã so sánh tại các điểm threshold trong miền giới hạn.
- Hàm tối ưu Adam. Adam dễ dàng vượt qua được điểm cực tiểu địa phương hơn so với SGD và tiến đến cực tiểu thấp nhất mà ta mong muốn nhanh hơn SGD.

Khi so sánh mô hình ResNet50 đã cải tiến so với DenseNet121, kết quả đạt được là gần giống nhau, chỉ chênh lệch 0.008 nhưng thời gian huấn luyện của ResNet50 lại nhanh hơn và chưa kể đến số lớp của ResNet50 cũng ít hơn DenseNet121 rất nhiều. Với EfficientNetB2, thời gian huấn luyện của 2 mô hình là gần giống nhau nhưng EfficientNetB2 lại tỏ ra hiệu quả hơn hẳn. Toàn bộ source code và thí nghiệm của nhóm được công bố trên Github: [git@github.com:xuan-vy-nguyen/PPNCKH.git](https://github.com/xuan-vy-nguyen/PPNCKH.git)

Tài liệu

- [1] Jimmy Lei Ba Diederik P. Kingma. Adam: A method for stochastic optimization. 2015.
- [2] Laurens van der Maaten Kilian Q. Weinberger Gao Huang, Zhuang Liu. Densely connected convolutional networks. 2016.
- [3] Shaoqing Ren Jian Sun Kaiming He, Xiangyu Zhang. Deep residual learning for image recognition. 2015.
- [4] Yan Liu Xu Sun Liangchen Luo, Yuanhao Xiong. Adaptive gradient methods with dynamic bound of learning rate. 2019.
- [5] Quoc V. Le Mingxing Tan. Efficientnet: Rethinking model scaling for convolutional neural networks. 2019.
- [6] Paul Mooney. Chex-ray images (pneumonia).
- [7] Nguyễn Lê Sang Nguyễn Xuân Vỹ, Trần Đức Huy. <https://github.com/xuan-vy-nguyen/ppnckh.git>.
- [8] Wikipedia. Pneumonia.