

SO SÁNH MÔ HÌNH HỌC SÂU VỚI CÁC PHƯƠNG PHÁP HỌC TỰ ĐỘNG KHÁC TRONG PHÂN LỚP DỮ LIỆU BIỂU HIỆN GEN MICROARRAY

Huỳnh Phước Hải¹, Nguyễn Văn Hòa¹, Đỗ Thanh Nghị²

¹ Khoa Kỹ thuật Công nghệ Môi trường, Trường Đại học An Giang

² Khoa Công nghệ thông tin & Truyền Thông, Đại học Cần Thơ

hphai@agu.edu.vn, nvhoa@agu.edu.vn, dtngghi@cit.ctu.edu.vn

TÓM TẮT— Hiện nay các mô hình học sâu (Deep Learning) tiêu biểu như mô hình mạng nơ-ron tích chập (Convolutional Neural Networks - CNNs) được ứng dụng thành công trong bài toán phân lớp ảnh, văn bản, nhận dạng tiếng nói. Ưu điểm của các mô hình học sâu là tự động học các đặc trưng của dữ liệu để thiết lập các đặc trưng mới và phân lớp dữ liệu. Trong bài báo này chúng tôi đề xuất xây dựng mô hình kiến trúc mạng nơ-ron tích chập để phân lớp dữ liệu biểu hiện gen microarray có số chiều lớn. Kết quả thực nghiệm trên 10 tập dữ liệu biểu hiện gen microarray được lấy từ ngân hàng dữ liệu y sinh (Kent Ridge) và cơ sở dữ liệu Gene Expression Omnibus (GEO) của NCBI cho thấy rằng mô hình mạng nơ-ron tích chập có độ chính xác cao hơn các mô hình đơn giản như k láng giềng (k Nearest Neighbors - k NN), cây quyết định (Decision Tree - DT). Mạng nơ-ron tích chập đạt được độ chính xác tương đương với mô hình máy học véc-tơ hỗ trợ (Support Vector Machines - SVM), rừng ngẫu nhiên (Random Forest - RF) và tốt hơn so với Adaboost và Bagging của cây quyết định.

Từ khóa— Mô hình học sâu, mạng nơ-ron tích chập, phân lớp dữ liệu biểu hiện gen microarray, k láng giềng, cây quyết định, máy học véc-tơ hỗ trợ, rừng ngẫu nhiên, Bagging, Adaboost.

I. GIỚI THIỆU

Trong những thập kỷ gần đây, mặc dù có nhiều nghiên cứu trong điều trị bệnh ung thư tuy nhiên chưa mang lại hiệu quả thực sự hiệu quả cho bệnh nhân do sự liên quan phức tạp giữa các yếu tố di truyền, mô học, đột biến gen ở các khối u (Sørli, 2001). Việc phân lớp chính xác các khối u rất quan trọng và cần thiết để chẩn đoán điều trị bệnh ung thư thành công (Pawitan, 2005). Bằng cách cho phép theo dõi mức độ biểu hiện của một số lượng lớn các gen trong một thí nghiệm có thể khám phá ra vai trò các gen cụ thể trong quá trình điều trị bệnh (Schena, 1995). Sự phát triển của các công nghệ chip dữ liệu biểu hiện gen đã tạo ra các bộ dữ liệu có số mẫu nhỏ và số chiều rất lớn được phân tích từ cường độ biểu hiện gen của các tế bào ung thư thu được qua các thí nghiệm microarray gọi là dữ liệu biểu hiện gen microarray (microarray gene expression data - MGE).

Phân lớp dữ liệu có số chiều lớn thường được biết là một trong 10 vấn đề khó của cộng đồng khai mô dữ liệu (Yang, 2006). Mô hình phân lớp cho kết quả tốt trên tập huấn luyện nhưng có kết quả thấp trên tập kiểm tra. Vấn đề khó khăn thường gặp chính là dữ liệu có số chiều quá lớn lên đến hàng nghìn chiều và dữ liệu tách rời nhau trong không gian có số chiều lớn nên việc tìm mô hình phân lớp tốt là khó khăn do có quá nhiều khả năng lựa chọn mô hình. Để tìm kiếm một mô hình phân lớp hiệu quả (phân lớp dữ liệu tốt trên tập kiểm thử) trong không gian giả thuyết lớn là vấn đề khó. Đã có nhiều giải thuật học tự động được nghiên cứu phân lớp dữ liệu có số chiều lớn như k láng giềng k NN (Fix, 1952), cây quyết định CART (Breiman, 1984) và C4.5 (Quinlan, 1983), Bagging (Breiman, 1996), Adaboost (Freund, 1995), rừng ngẫu nhiên RF (Breiman, 2001) và máy học véc-tơ hỗ trợ SVM (Vapnik, 1995).

Những năm qua, mô hình học sâu đặc biệt là mạng nơ-ron tích chập CNNs là mô hình được sử dụng phổ biến trong cộng đồng máy học cho hiệu quả trong các bài toán phân loại hình ảnh (Krizhevsky, 2012), phân loại văn bản (Kim, 2014) và gần đây đã có nhiều nghiên cứu sử dụng mạng nơ-ron tích chập trong lĩnh vực tin sinh học (Min, 2016) như phân tích Protein (Zacharaki, 2017), phân tích ảnh y khoa (Li, 2014). Ưu điểm của CNNs là tận dụng được tính năng trích chọn đặc trưng của lớp tích chập và bộ phân lớp được huấn luyện đồng thời. Ý tưởng học cùng lúc đặc trưng và bộ phân lớp có thể hỗ trợ với nhau trong quá trình huấn luyện và quá trình phân lớp tìm ra các tham số phù hợp với các véc-tơ đặc trưng tìm được từ lớp tích chập và ngược lại lớp tích chập điều chỉnh các tham số của lớp tích chập để cho các véc-tơ đặc trưng thu được là tuyến tính phù hợp với bộ phân lớp của lớp cuối cùng. Đến thời điểm hiện nay, chưa có nghiên cứu sử dụng CNNs trong phân lớp dữ liệu biểu hiện gen MGE. Trong bài viết này, chúng tôi đề xuất huấn luyện mô hình mạng tích chập CNNs để phân lớp dữ liệu MEG và so sánh hiệu quả của CNNs với các giải thuật khác như k láng giềng (k NN), cây quyết định (DT), máy học véc-tơ hỗ trợ (SVM), rừng ngẫu nhiên (RF), Bagging, Boosting. Kết quả thực nghiệm trên 10 tập dữ liệu biểu hiện gen MEG từ kho Kent Ridge (Jinyan & Huiqing, 2002), GEO (Edgar et al., 2002) cho thấy rằng mạng nơ-ron tích chập CNNs đạt được độ chính xác tương đương với mô hình máy học SVM, rừng ngẫu nhiên RF và tốt hơn so với k láng giềng k NN, cây quyết định DT, Adaboost và Bagging.

Phần còn lại của bài viết được tổ chức như sau. Các nghiên cứu liên quan về mô hình phân lớp dữ liệu biểu hiện gen MGE được trình bày trong phần II. Mô hình mạng nơ-ron tích chập phân lớp dữ liệu biểu hiện gen MGE được trình bày trong phần III. Kết quả thực nghiệm sẽ được trình bày trong phần IV trước khi kết luận và hướng phát triển trong phần V.

II. CÁC MÔ HÌNH PHÂN LỚP DỮ LIỆU BIỂU HIỆN GEN MICROARRAY

Mô hình k láng giềng gần (k Nearest Neighbors – k NN)

Giải thuật k NN (Fix & Hodges, 1952) là một giải thuật học đơn giản nhưng cho hiệu quả cao trong khai phá dữ liệu (Zhou, 2009). Giải thuật k NN không có quá trình học. Khi dự đoán lớp (nhân) của một phần tử x mới đến, giải thuật

kNN tìm trong m phần tử của tập huấn luyện k láng giềng của x , sau đó thực hiện gán lớp cho x dựa vào luật bình chọn số đông từ các lớp của k láng giềng.

Nhiều nghiên cứu sử dụng kNN để phân lớp dữ liệu MGE bởi tính đơn giản nhưng cho kết quả tốt. Li và các cộng sự sử dụng giải thuật di truyền để giảm chiều và phân lớp bệnh Lymphoma bằng kNN (Li, 2001). Nghiên cứu của Doudoit và cs (Dudoit, 2002) đã so sánh và chứng minh kNN đạt kết quả phân lớp tốt. Tập dữ liệu MEG về bệnh Prostate Cancer dùng kNN phân lớp đạt kết quả chính xác 90% (Singh, 2002). Ngoài ra kNN được sử dụng trong nhiều nghiên cứu phân lớp dữ liệu MGE trong các nghiên cứu đã được công bố trong những năm gần đây (Liu, 2002), (Pan, 2004), (Yao, 2006), (Deegalla, 2007), (Parry, 2010), (Su, 2011), (Halder, 2014), (Halder, 2015).

Mô hình cây quyết định (Decision Trees – DT)

Mô hình máy học cây quyết định DT (Breiman et al., 1984), (Quinlan, 1993) là mô hình máy học tự động được sử dụng rất nhiều trong khai mỏ dữ liệu (Zhou, 2009) với tính đơn giản và hiệu quả.

Giả sử tập dữ liệu có m phần tử $x_1, x_2, \dots, x_m$ trong không gian n chiều có lớp tương ứng là $y_1, y_2, \dots, y_m$. Giải thuật cây quyết định xây dựng cây bắt đầu từ nút gốc đến nút lá. Đây là giải thuật đệ quy phân hoạch tập dữ liệu theo các biến độc lập thành các siêu chữ nhật rời nhau mà ở đó các phần tử dữ liệu $x_i, x_j, \dots, x_k$ của phân vùng (nút lá) có các $y_i, y_j, \dots, y_k$ là thuần khiết (giống nhau trong vấn đề phân lớp). Giải thuật cây quyết định thực hiện 2 bước xây dựng cây và cắt nhánh để tránh học vẹt.

Mô hình cây quyết định cũng được sử dụng để phân lớp dữ liệu MEG như trong nghiên cứu của Netto (Netto, 2010) để phân lớp dữ liệu bệnh Leukemia, nghiên cứu của Snousy (Snousy, 2011) để phân lớp dữ liệu ung thư trên toàn bộ các gen không giảm chiều. Ngoài ra, mô hình cây quyết định được sử dụng trong một số nghiên cứu khác như (Ulfborg, 2013), (Ludwig, 2015).

Mô hình Bagging và Boosting

Mô hình Bagging (Bootstrap AGGREGatING) được (Breiman, 1996) đề xuất nhằm giảm lỗi variance nhưng không làm tăng lỗi bias quá nhiều. Giải thuật Bagging xây dựng mô hình gồm ba các bước: đầu tiên từ tập dữ liệu học LS có m phần tử, xây dựng T mô hình cơ sở độc lập nhau, mô hình thứ t được xây dựng trên tập mẫu bootstrap thứ t (lấy mẫu m phần tử có hoàn lại từ tập học LS). Dự đoán lớp của phần tử mới đến với chiến lược bình chọn số đông thu được từ dự đoán lớp của T mô hình cơ sở.

Mô hình Boosting với giải thuật tiêu biểu Adaboost của (Freund & Schapire, 1995) là một phương pháp tổng quát để cải tiến độ chính xác của các bộ phân lớp yếu. Mô hình Adaboost được xây dựng bằng cách lặp lại việc huấn luyện tuần tự các bộ phân lớp yếu T lần, mỗi lần lặp huấn luyện bộ phân lớp yếu từ dựa trên tập dữ liệu được lấy mẫu từ tập huấn luyện tập trung vào các phần tử bị phân lớp sai trong các lần lặp trước đó. Phân loại phần tử mới sử dụng kết quả bình chọn trọng số từ T bộ phân lớp yếu.

Mô hình Bagging và Boosting được (Tan, 2003) sử dụng để phân loại 7 tập dữ liệu MGE đã cho thấy hiệu quả của Bagging và Boosting tốt hơn so với cây quyết định. Bagging được kết hợp với Boosting qua nghiên cứu của (Dettling, 2004) để phân lớp bệnh ung thư. Nghiên cứu của (Osareh, 2013) đã so sánh hiệu quả của giải thuật Adaboost, Bagging với các phương pháp khác trên 8 tập dữ liệu MGE, giải thuật Adaboost và Bagging đạt hiệu quả cao trên đa số các tập dữ liệu đã chọn.

Mô hình Rừng ngẫu nhiên (Random Forest – RF)

Rừng ngẫu nhiên (Breiman, 2001) là một trong những phương pháp tập hợp mô hình thành công nhất. Giải thuật rừng ngẫu nhiên xây dựng cây không cắt nhánh nhằm giữ cho bias thấp và dùng tính ngẫu nhiên để điều khiển tính tương quan thấp giữa các cây trong rừng. Rừng ngẫu nhiên RF tạo ra một tập hợp các cây quyết định không cắt nhánh, mỗi cây được xây dựng trên tập mẫu bootstrap, tại mỗi nút phân hoạch tốt nhất được thực hiện từ việc chọn ngẫu nhiên một tập con các thuộc tính. Lỗi tổng quát của rừng ngẫu nhiên phụ thuộc vào độ chính xác của từng cây thành viên trong rừng và sự phụ thuộc lẫn nhau giữa các cây thành viên. Giải thuật rừng ngẫu nhiên cho độ chính xác cao khi so sánh với các thuật toán học có giám sát hiện nay, chịu đựng nhiễu tốt.

Nghiên cứu tiêu biểu cho thấy hiệu quả sử dụng rừng ngẫu nhiên để phân lớp 9 tập dữ liệu MGE của (Díaz-Urriarte, 2006) so với các mô hình kNN và SVM. Rừng ngẫu nhiên cho thấy hiệu quả tốt hơn khi thực hiện phân lớp các tập dữ liệu y sinh với số chiều cực lớn khi được so sánh với các giải thuật Bagging và kNN qua các nghiên cứu (Statniko, 2008) và (Pirooznia, 2008). Một số nghiên cứu cải tiến giải thuật rừng ngẫu nhiên như (Đỗ Thanh Nghị, 2010), (Đỗ Thanh Nghị, 2016) cải thiện hiệu quả của rừng ngẫu nhiên cao hơn so với tiếp cận truyền thống và tương đương với SVM. Với hiệu quả cao giải thuật rừng ngẫu nhiên được sử dụng nhiều trong các nghiên cứu phân lớp dữ liệu MGE (Moorthy, 2011), (Chen, 2012), (Aytekin, 2016), (Nishiwaki, 2017).

Mô hình máy học véc-tơ hỗ trợ (Support Vector Machines – SVM)

Mô hình máy học SVM dựa trên lý thuyết học thống kê của Vapnik (Vapnik, 1978). Trong vấn đề phân lớp nhị phân (2 lớp dương và âm), giải thuật SVM (Vapnik, 1995) tìm siêu phẳng tối ưu để phân hoạch các điểm dữ liệu

thành 2 phần sao cho các điểm cùng một lớp nằm về một phía của siêu phẳng. Siêu phẳng tối ưu là siêu phẳng tách 2 lớp ra xa nhất có thể dựa trên 2 siêu phẳng hỗ trợ song song của 2 lớp, siêu phẳng phân lớp tối ưu là siêu phẳng nằm giữa 2 siêu phẳng hỗ trợ. Khoảng cách giữa 2 siêu phẳng hỗ trợ được gọi là lề. Có thể thấy rằng lề phân hoạch càng lớn thì mô hình phân lớp càng an toàn. Những phần tử nằm ngược phía với siêu phẳng hỗ trợ được coi như lỗi. Giải thuật SVM tìm siêu phẳng tối ưu cần phải cực đại hóa lề và cực tiểu hóa lỗi. Trong phân lớp đa lớp (nhiều hơn 2 lớp), SVM có thể xây dựng trực tiếp mô hình tối ưu cho k lớp ($k > 2$) như đề xuất của (Weston, 1999). Trong thực tế có ba phương pháp được sử dụng là một chống lại tất cả 1-vs-all của (Vanik, 1995), một – một 1-vs-1 của (Krebel, 1999) và phân tách hai nhóm của (Vural, 2004).

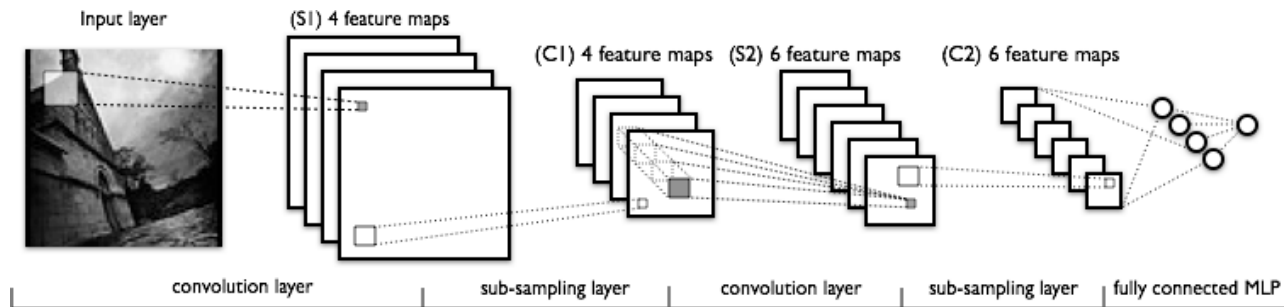
Máy học véc-tơ hỗ trợ được sử dụng nhiều để phân lớp dữ liệu MGE như nghiên cứu của Mukherjee (Mukherjee, 1999) phân loại bệnh Leukemia thu được độ chính xác đạt 100% trên tập dữ liệu có 7129 gen. Nghiên cứu của Furey (Furey, 2000) sử dụng SVM để phân lớp bệnh Ovarian bằng kỹ thuật xếp hạng gen để giảm chiều và dùng SVM để phân lớp. Nổi bật nhất và được trích dẫn nhiều nhất là nghiên cứu của nhóm tác giả Isabelle Guyon (Guyon, 2002) đã sử dụng SVM để giảm chiều và xây dựng mô hình phân lớp với độ chính xác cao khi phân lớp dữ liệu MGE thu từ bệnh ung thư đại tràng. Kết quả của nghiên cứu này cho thấy SVM có hiệu quả tốt với độ chính xác 98%. Ngoài ra nghiên cứu của Guyon đã chứng minh được các gen được lựa chọn bằng kỹ thuật SVM sẽ xây dựng được bộ phân lớp tốt hơn trong việc phân loại bệnh ung thư. SVM còn được sử dụng trong rất nhiều nghiên cứu khác như trong những năm gần đây như (T. Li, 2004), (Chuang, 2005), (Phan, 2005), (Huerta, 2006), (Alba, 2007), (Hsieh, 2010), (Vanneschi, 2011), (Nagi, 2012), (Statnikov, 2013), (Vanitha, 2015), (Cogill, 2016).

III. MÔ HÌNH MẠNG NƠ-RON TÍCH CHẬP PHÂN LỚP DỮ LIỆU BIỂU HIỆN GEN MICROARRAY

Mạng Nơ-ron tích chập

Mạng nơ-ron tích chập (CNNs) (Lecun, 1998) là một mô hình học sâu có thể xây dựng được các hệ thống phân loại với độ chính xác cao. Ý tưởng của CNNs được lấy cảm hứng từ khả năng nhận biết thị giác của bộ não người. Để có thể nhận biết được các hình ảnh trong vỏ não người có hai loại tế bào là tế bào đơn giản và tế bào phức tạp (Hubel, 1968). Các tế bào đơn giản phản ứng với các mẫu hình dạng cơ bản ở các vùng kích thích thị giác và các tế bào phức tạp tổng hợp thông tin từ các tế bào đơn giản để xác định các mẫu hình dạng phức tạp hơn. Khả năng nhận biết các hình ảnh của não người là một hệ thống xử lý hình ảnh tự nhiên mạnh mẽ và tự nhiên nên CNNs được phát triển dựa trên ba ý tưởng chính: tính kết nối cục bộ (Local connectivity hay compositionality), tính bất biến (Location invariance) và tính bất biến đối với quá trình chuyển đổi cục bộ (Invariance to local transition) (LeCun, 2015). CNNs là một dạng mạng nơ-ron chuyên dụng để xử lý các dữ liệu dạng lưới 1 chiều như dữ liệu âm thanh, dữ liệu MGE hoặc nhiều chiều như dữ liệu hình ảnh, video.

Cấu trúc cơ bản của CNNs gồm các lớp tích chập (Convolution layer), lớp phi tuyến (Nonlinear layer) và lớp lọc (Pooling layer) như hình 1. Các lớp tích chập kết hợp với các lớp phi tuyến sử dụng các hàm phi tuyến như ReLU hay tanh để tạo ra thông tin trừu tượng hơn (Abstract/higher-level) cho các lớp tiếp theo.



Hình 1: Cấu trúc cơ bản của mạng Nơ-ron Tích chập (Lecun, 1989)

Các lớp liên kết trong CNNs được với nhau thông qua cơ chế tích chập. Lớp tiếp theo là kết quả tích chập từ lớp trước đó vì vậy CNNs có được các kết nối cục bộ vì mỗi nơ-ron ở lớp tiếp theo sinh ra từ một bộ lọc được áp đặt lên một vùng cục bộ của lớp trước đó. Nguyên tắc này được gọi là kết nối cục bộ (Local connectivity). Mỗi lớp như vậy được áp đặt các bộ lọc khác nhau. Một số lớp khác như lớp pooling/subsampling dùng để lọc lại các thông tin hữu ích hơn bằng cách loại bỏ các thông tin nhiễu. Trong suốt quá trình huấn luyện, CNNs sẽ tự động học các tham số cho các lớp. Lớp cuối cùng được gọi là lớp kết nối đầy đủ (Fully connect layer) dùng để phân lớp.

Mạng nơ-ron tích chập dùng để phân lớp dữ liệu MGE có kiến trúc mạng gồm nhiều tầng. Lớp đầu vào của mạng (Input layer) là một ma trận cường độ biểu hiện gen của dữ liệu MGE. Lớp đầu ra của mạng (Output layer) là số lớp của dữ liệu. Giữa lớp đầu vào và đầu ra gồm nhiều tầng tích chập và một lớp kết nối đầy đủ. Mỗi tầng gồm một lớp tích chập, một lớp phi tuyến và một lớp lọc. Dữ liệu khi được huấn luyện qua các tầng tích chập sẽ được phân lớp ở lớp kết nối đầy đủ sử dụng bộ phân lớp Softmax.

Lớp tích chập (Convolution)

Tích chập được sử dụng đầu tiên trong xử lý tín hiệu số. Nhờ vào nguyên lý biến đổi thông tin có thể áp dụng kĩ thuật này vào xử lý ảnh và video số. Trong lớp tích chập sử dụng một bộ các bộ lọc có kích thước nhỏ hơn với ma trận đầu vào và áp lên một vùng của ma trận đầu vào và tiến hành tính tích chập giữa bộ filter và giá trị của ma trận trong vùng cục bộ đó. Các filter sẽ dịch chuyển một bước trượt (Stride) chạy dọc theo ma trận đầu vào và quét toàn bộ ma trận. Trọng số của filter ban đầu sẽ được khởi tạo ngẫu nhiên và sẽ được học dần trong quá trình huấn luyện mô hình.

Lớp phi tuyến Relu (Rectified linear unit)

Giả sử mạng CNNs có L lớp có lớp input là lớp thứ 0 . Mạng sẽ có L ma trận trọng số w được ký hiệu là $W^l \in R^{a^{(l-1)} \times a^l}$, $l = 1, 2, \dots$. Trong đó W^l là các kết nối từ lớp thứ $l-1$ đến lớp thứ l , phần tử w_{ij}^l thể hiện kết nối của nơ-ron thứ i của lớp $(l-1)$ đến nơ-ron thứ j của lớp l . Các bias thứ (l) được ký hiệu là $b^l \in R^{a^{(l)}}$. Để thực hiện phân lớp có kết quả tối ưu là quá trình đi tìm bộ tham số w và b . Mỗi nơ-ron không phải input được tính bằng công thức:

$$a_i^{(l)} = f(w_i^{(l)T} a^{(l-1)} + b_i^l)$$

(1)

Trong công thức (1) f là một hàm kích hoạt phi tuyến được áp dụng cho một ma trận. Trong kiến trúc mạng CNNs thường sử dụng hàm kích hoạt $f(x) = \max(0, x)$ chuyển toàn bộ giá trị âm trong kết quả lấy từ lớp tích chập thành giá trị 0 để tạo tính phi tuyến cho mô hình gọi là Relu. Ngoài ra còn có nhiều hàm kích hoạt khác như signmod, tang nhưng hàm relu dễ cài đặt tính toán nhanh và hiệu quả (Krizhevsky, 2012)

Lớp Pooling

Lớp Pooling sử dụng một cửa sổ trượt quét qua toàn bộ ma trận dữ liệu theo một bước trượt cho trước để tiến hành lấy mẫu. Các phương thức lấy phổ biến trong lớp Pooling là MaxPooling (lấy giá trị lớn nhất), MinPooling (lấy giá trị nhỏ nhất) và AveragePooling (lấy giá trị trung bình). Công dụng của lớp Pooling dùng để giảm kích thước dữ liệu, các tầng trong CNNs chồng lên nhau có lớp Pooling ở cuối mỗi tầng giúp cho kích thước dữ liệu được co lại nhưng vẫn giữ được các đặc trưng để lấy mẫu. Ngoài ra giảm kích thước dữ liệu sẽ giảm số lượng tham số của mạng tăng tính hiệu quả và kiểm soát hiện tượng học vẹt (Overfitting)

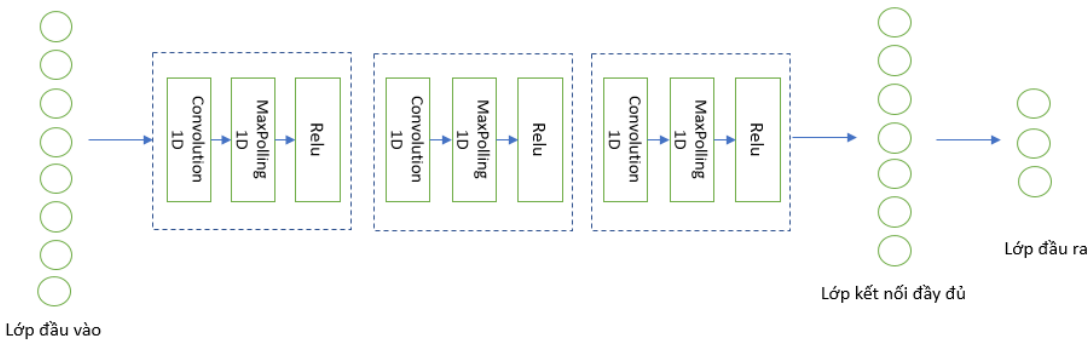
Lớp kết nối đầy đủ (Fully connect layer)

Lớp kết nối đầy đủ là một lớp giống như mạng nơ-ron truyền thẳng các giá trị được tính toán từ các lớp trước sẽ được liên kết đầy đủ vào trong nơ-ron của lớp tiếp theo. Tại lớp kết nối đầy đủ sẽ tiến hành phân lớp dữ liệu bằng cách kích hoạt hàm softmax để tính xác suất ở lớp đầu ra.

Mô hình mạng Nơ-ron tích chập phân lớp dữ liệu MGE

Dữ liệu MGE đã được chuẩn hoá là một ma trận có các dòng tương ứng các mẫu bệnh còn các cột tương ứng với các gen, giá trị tại một điểm trong ma trận là cường độ biểu hiện gen của mẫu thử dữ liệu tương ứng với gen tại cột đó. Như vậy dữ liệu MGE có đặc điểm dạng lưới 1 chiều có thể sử dụng CNNs để phân lớp.

Mô hình CNNs dùng để phân lớp dữ liệu MGE có cấu trúc như hình 2. Lớp đầu vào nhận đầu vào là dữ liệu biểu hiện gen. Các tầng tích chập lớp tích chập (Convolution 1D), lớp lọc (Maxpooling 1D), lớp phi tuyến Relu chồng lên nhau gọi là 1 tầng. Mô hình gồm nhiều tầng của 3 lớp trên xếp chồng lên nhau. Sau khi dữ liệu được trích suất qua các tầng tích chập được đưa vào một lớp kết nối đầy đủ và được phân lớp tại lớp đầu ra.



Hình 2. Mô hình CNNs phân lớp dữ liệu MGE

Mạng CNNs sẽ được huấn luyện trên các tập dữ liệu MGE, bản chất quá trình huấn luyện là sự thay đổi các trọng số liên kết của mạng. Trong quá trình này, các trọng số của mạng sẽ hội tụ dần tới các giá trị sao cho với mỗi vector đầu vào từ tập huấn luyện, mạng sẽ cho ra vector đầu ra như mong muốn. Để có được bộ tham số cho hiệu quả phân lớp tốt cần điều chỉnh các tham số của mạng như độ dài các bộ lọc trong các lớp tích chập và pooling. Đối với các tập dữ liệu có số mẫu nhỏ và số chiều rất lớn khi huấn luyện dễ xảy ra hiện tượng overfitting. Khi đó mạng học có kết quả rất tốt trên tập huấn luyện nhưng có kết quả thấp trên tập kiểm tra. Hiện tượng này xảy ra do khi mạng được huấn

luyện, mạng chuyển từ các hàm ánh xạ đơn giản đến các hàm ánh xạ phức tạp. Mạng sẽ đạt được một cấu hình tổng quát hóa tốt nhất tại một điểm nào đó trong quá trình học. Sau thời điểm đó mạng bị mô hình hóa nhiều, những gì mạng học được sẽ trở thành overfitting. Trong trường hợp này chúng tôi sử dụng kỹ thuật huấn luyện dừng sớm (Early stop training) để chọn được mô hình có kết quả tốt nhất nhưng không bị overfitting.

IV. KẾT QUẢ THỰC NGHIỆM

Chúng tôi cài đặt mô hình CNNs bằng ngôn ngữ Python, sử dụng thư viện Tensorflow (Martín, 2015) và Keras (Chollet, 2015). Các giải thuật kNN, DT-C4.5, RF-C4.5, Bagging-C4.5, Adaboost cũng được cài đặt bằng ngôn ngữ lập trình Python có kế thừa từ mã nguồn C4.5 được cung cấp bởi (Quinlan, 1993) và thư viện Sklearn (Pedregosa et al. 2011). Đối với giải thuật SVM, chúng tôi sử dụng thư viện LibSVM của (Chang, 2011). Chúng tôi chạy thực nghiệm trên môi trường với CPU Intel Xeon 3.07GHz, bộ nhớ RAM 8GB, hệ điều hành Fedora.

Dữ liệu thực nghiệm

Để so sánh CNNs với các mô hình khác chúng tôi sử dụng 10 tập dữ liệu từ các nghiên cứu biểu hiện gen của bệnh ung thư được đã được công bố công khai tại Kent Ridge (Jinyan, 2002) và kho cơ sở dữ liệu biểu hiện gen GEO của NCBI (Edgar, 2002). Dữ liệu thực nghiệm có tính đa dạng bao gồm các tập dữ liệu 2 lớp, 3 lớp; 7 tập không có tập dữ liệu kiểm tra, 3 tập dữ liệu có tập kiểm tra. Thông tin chi tiết của các tập dữ liệu được trình bày trong bảng 1.

Bảng 1: Mô tả các tập dữ liệu biểu hiện gen microarray

TT tập dữ liệu	Tên tập dữ liệu	Số mẫu	Số chiều	Số lớp	Nghi thức	Nguồn
1	Lung Cancer	181	12533	3	loo	Gordon, 2002
2	MLL	30	12582	3	trn-tst	Armstrong, 2002
3	Breast Cancer	72	7129	2	trn-tst	Veer, 2002
4	Brain Tumor	118	22215	5	loo	Pomeroy, 2005
5	Breast Cancer	96	7129	2	loo	Chin, 2006
6	Lung Cancer	77	7129	2	loo	Beer, 2002
7	Cron Cancer	30	4522	3	loo	Burczynski , 2006
8	Colon Cancer	62	2000	2	loo	Alon, 1999
9	Breast Cancer	104	22283	2	loo	Chowdaly, 2006
10	Prostate Cancer	102	12600	2	trn-tst	Singh, 2002

Thực nghiệm

Đối với các tập dữ liệu không sẵn có tập kiểm tra, chúng tôi dùng giao thức kiểm tra chéo leave-one-out (loo) vì các tập dữ liệu này có phần tử nhỏ hơn 300. Chúng tôi tiến hành đánh giá hiệu quả của các mô hình phân lớp dựa trên các tiêu chí đánh giá như Precision, Recall, F1-measure và Accuracy (Rijsbergen, 1979).

Mô hình CNNs được huấn luyện với tốc độ học là 0.01, sử dụng hàm kích hoạt softmax trong lớp đầy đủ. Hàm kích hoạt Relu được sử dụng toàn bộ trong các tầng ẩn của lớp tích chập, số lần học nhiều nhất của mạng là 100 lần. Kiến trúc mạng cơ bản gồm: Lớp đầu vào → Các tầng [Lớp Convolution → Lớp Maxpooling → Lớp Activation] → Lớp đầu ra.. Khi huấn luyện mạng chúng tôi chọn tham số Batch size=32 để chọn kích thước mẫu được học và điều chỉnh các tham số trong các lớp của mạng để có độ chính xác tốt nhất. Số lần huấn luyện mạng tối đa là 100 lần. Mô hình có kết quả phân lớp tốt nhất trên tập huấn luyện sẽ được chọn sẽ được kiểm tra trên các tập kiểm thử để đánh giá.

Đối với mô hình kNN chúng tôi thử nghiệm với nhiều tham số k (k=1,3,5,7) để chọn kết quả tốt nhất. Đối với các phương pháp tập hợp mô hình chúng tôi chạy thử nghiệm với nhiều số lượng cây quyết định khác nhau để chọn kết quả tốt nhất. Số cây quyết định được thử nghiệm từ 10 cây đến 350 cây. Qua thực nghiệm chúng tôi thấy được giải thuật RF xây dựng 200 cây quyết định sẽ đảm bảo được độ chính xác cao còn các giải thuật Bagging, Boosting xây dựng số cây nhỏ hơn 50 cây quyết định để đảm bảo được độ chính xác tốt nhất. Giải thuật SVM sử dụng hàm nhân tuyến tính là phân lớp tốt nhất các tập dữ liệu MGE. Kết quả thu được từ các mô hình phân loại (với các tham số tối ưu) như trình bày trong bảng 3, 4, 5 và 6.

Bảng 3: Kết quả phân lớp của 7 giải thuật trên 10 tập dữ liệu với tiêu chí đánh giá Accuracy (%)

TT tập dữ liệu	DT	kNN	RF	BagDT	Adaboost	SVM	CNNs
1	94.63	93.96	100	99.33	94.63	98.66	100
2	100	100	100	100	100	100	100
3	52.63	63.16	89.47	68.42	63.16	68.42	78.95
4	74.7	91.57	100	97.59	68.67	100	100
5	65.56	87.78	82.22	83.33	65.56	86.67	85.56
6	77.97	88.14	86.44	90.68	86.44	87.29	88.14
7	100	98.96	98.96	98.96	98.96	98.96	95.3
8	60.63	77.17	80.31	77.95	67.72	94.49	88.98
9	89.42	91.35	96.15	95.19	95.19	97.12	97.12

10	55.88	73.53	88.24	85.29	47.06	94.12	97
----	-------	-------	-------	-------	-------	-------	----

Bảng 4: Kết quả phân lớp của 7 giải thuật trên 10 tập dữ liệu với tiêu chí đánh giá Precision (%)

TT tập dữ liệu	DT	kNN	RF	BagDT	Adaboost	SVM	CNNs
1	95.94	96.22	100	99.33	95.94	98.82	100
2	100	100	100	100	100	100	100
3	54.01	67.54	89.47	68.42	60.53	67.22	79.1
4	76.3	91.62	100	96.53	65.59	100	100
5	66.16	82.03	77.74	80.36	66.16	82.82	85.56
6	84.64	89.33	88.11	90.26	86.35	87.26	88.14
7	97.92	99.05	98.97	100	97.92	99.05	95.3
8	57.92	77.19	84.7	75.32	70.03	93.9	80.65
9	90.38	91.39	96.2	95.3	95.3	97.14	97.12
10	95.94	96.22	100	99.33	95.94	98.82	100

Bảng 5: Kết quả phân lớp của 7 giải thuật trên 10 tập dữ liệu với tiêu chí đánh giá Recall (%)

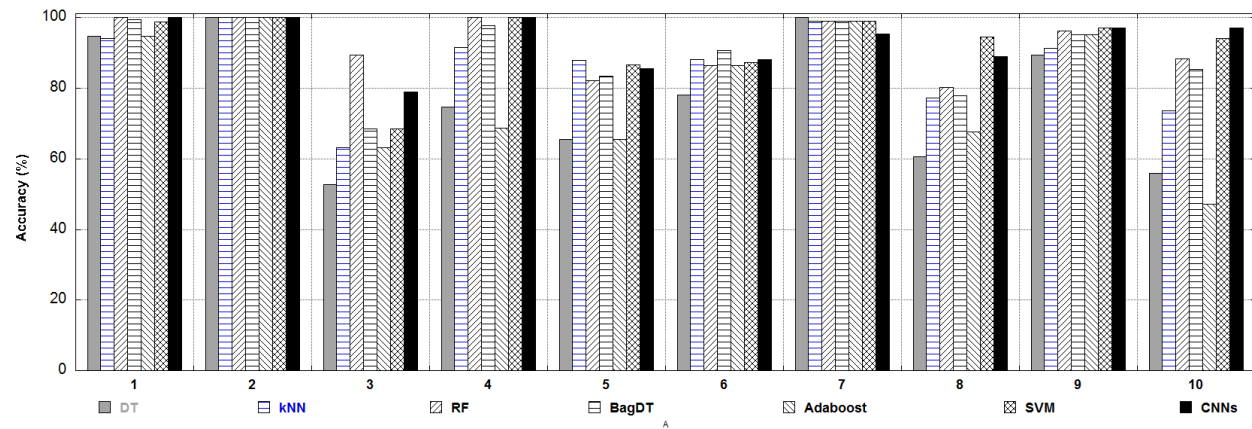
TT tập dữ liệu	DT	kNN	RF	BagDT	Adaboost	SVM	CNN
1	94.63	93.96	100	99.33	94.63	98.66	100
2	100	100	100	100	100	100	100
3	52.63	63.16	89.47	68.42	63.16	68.42	78.95
4	75.9	91.57	100	96.39	66.27	100	100
5	60	86.67	82.22	85.55	65.55	86.67	85.56
6	84.75	88.14	87.29	89.83	86.44	87.29	88.14
7	97.92	98.95	98.96	100	97.92	98.95	95.3
8	58.26	77.165	80.31	76.37	82.26	93.7	80.65
9	90.38	91.35	96.15	95.19	95.19	97.16	97.12
10	55.88	73.53	92.59	85.29	47.06	94.12	97

Bảng 6: Kết quả phân lớp của 7 giải thuật trên 10 tập dữ liệu với tiêu chí đánh giá F1 (%)

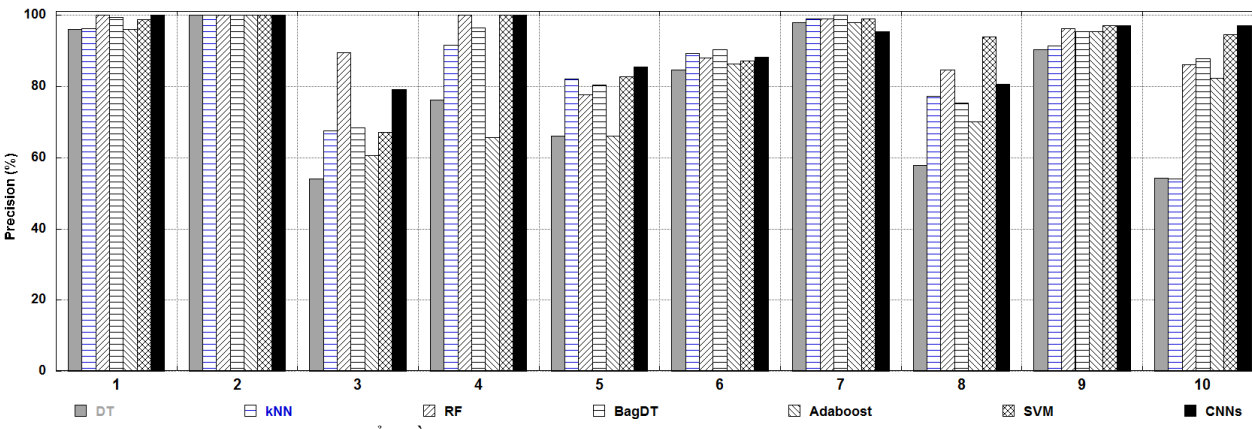
TT tập dữ liệu	DT	kNN	RF	BagDT	Adaboost	SVM	CNN
1	95.02	94.55	100	99.32	95.02	98.69	100
2	100	100	100	100	100	100	100
3	53.18	63.78	89.47	68.42	60.18	67	78
4	75.52	91.52	100	96.38	62.19	100	100
5	62.78	84.56	77.96	82.11	64.55	84.57	85.56
6	84.66	87.61	86.79	89.56	86.28	87.1	88.14
7	97.92	98.98	98.93	100	97.92	98.98	95.3
8	57.9	76.34	76.76	75.28	68.92	94.46	80.65
9	90.38	91.29	96.14	95.16	95.16	97.12	97.12
10	55.02	62.31	86.21	83.13	45.4	93.86	97

Kết quả phân lớp của 10 tập dữ liệu MEG với các tiêu chí đánh giá Precison, Recall, F1 và Accuracy của 7 giải thuật DT, kNN, RF, BagDT, Boosting, SVM và CNNs từ các bảng 3, 4, 5, 6 cho thấy CNNs có độ chính xác (Accuracy) tốt nhất với kết quả tốt nhất 5/10 tập dữ liệu, kế tiếp là SVMs và RF có kết quả tốt 4/10 tập dữ liệu. Adaboost có kết quả thấp nhưng trội hơn một ít là DT, kNN và BagDT. Đối với tiêu chí Precision, CNNs, RF và SVM tương đương nhau có kết quả tốt nhất 4/10 tập dữ liệu, Bagging có kết quả tốt nhất trên 3/10 tập dữ liệu cao hơn DT, Adaboost và KNN. Đánh giá tiêu chí Recall, CNNs có kết quả tương đương RF khi tốt nhất 4/10 tập dữ liệu và chỉ thấp hơn SVM trên 1 tập dữ liệu, Bagging có kết quả tốt nhất 3/10 tập dữ liệu cao hơn các giải thuật DT, Adaboost và KNN. Với tiêu chí F1, CNNs có kết quả tốt nhất trên 6/10 tập dữ liệu trội hơn SVM và RF có kết quả tốt trên 4/10 tập dữ liệu và tốt hơn tất cả các giải thuật khác. Đặc biệt CNNs có kết quả chính xác tổng hợp cao hơn các giải thuật khác trên các tập dữ liệu có số chiều lớn hơn 10000 chiều như các tập Lung Cancer (12533 gen), MLL (12582 gen), Brain Tumor (22215 gen), Breast Cancer (22283 gen) và Protates Cancer (12600 gen).

Hình 3 và 4 thể hiện biểu đồ trực quan so sánh kết quả của tiêu chí đánh giá Accuracy và Precision của 7 giải thuật trên 10 tập dữ liệu. Hai biểu đồ này chỉ ra rằng được giải thuật CNNs có hiệu quả tốt tương đương với SVM và giải thuật RF.



Hình 3 : Biểu đồ so sánh tiêu chí Accuracy của 7 giải thuật trên 10 tập dữ liệu



Hình 4: Biểu đồ so sánh tiêu chí Precision của 7 giải thuật trên 10 tập dữ liệu

V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Chúng tôi vừa trình bày đề xuất huấn luyện mô hình mạng tích chập CNNs để phân loại dữ liệu biểu hiện gen MGE và so sánh hiệu quả của mô hình đề xuất với các mô hình kNN, cây quyết định, Bagging, Adaboost, rừng ngẫu nhiên và máy học véc-tơ hỗ trợ. Kết quả thực nghiệm trên 10 tập dữ liệu biểu hiện gen cho thấy mô hình CNNs có thể phân lớp với độ chính xác cao hơn các mô hình học như kNN, cây quyết định và Bagging, Adaboost. Mạng tích chập CNNs cho kết quả phân lớp tương đương với máy học SVM và rừng ngẫu nhiên.

Trong tương lai chúng tôi tiếp tục nghiên cứu các mô hình học sâu để phân loại dữ liệu MGE bằng cách tinh chỉnh các tham số của mạng và xây dựng các bộ phân lớp khác sau khi dữ liệu được kết xuất qua các lớp tích chập. Ngoài nghiên cứu cải thiện chất lượng mô hình phân lớp, chúng tôi sẽ tập trung cải tiến tốc độ huấn luyện mô hình phân lớp và quá trình phân lớp của các giải thuật.

VI. TÀI LIỆU THAM KHẢO

[1] A. Krizhevsky, et al., “Imagenet classification with deep convolutional neural networks” in Advances in neural information processing systems, 2012.

[2] Y. Kim, “Convolutional neural networks for sentence classification”, 2014.

[3] S. Min, B. Lee, and S. Yoon, “Deep learning in bioinformatics” Brief. Bioinform, 2016.

[4] E. I. Zacharaki, “Prediction of protein function using a deep CNN ensemble” PeerJ Prepr, 2017.

[5] Q. Li, W. Cai, et al., “Medical image classification with convolutional neural network” in Control Automation Robotics & Vision (ICARCV) 2014.

[6] E. Fix and J. L. Hodges Jr, “Discriminatory analysis-nonparametric discrimination: Small sample performance” DTIC Document, 1952.

[7] Z. Zhou, Y. Yang, X. Wu, et al., “The Top Ten Algorithms in Data Mining” Fla. Chapman Hall CRC, 2009.

[8] L. Li, C. R. Weinberg, T. A. Darden, et al., “Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/kNN method” Bioinformatics, 2001.

[9] S. Dudoit, J. Fridlyand, and T. P. Speed, “Comparison of discrimination methods for the classification of tumors using gene expression data” J. Am. Stat. Assoc., vol. 97, no. 457, pp. 77–87, 2002.

[10] D. Singh et al., “Gene expression correlates of clinical prostate cancer behavior” Cancer Cell, 2002.

[11] H. Liu, J. Li, and L. Wong, “A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns” Genome Inform., vol. 13, pp. 51–60, 2002.

- [12] F. Pan, B. Wang, X. Hu, and W. Perrizo, "Comprehensive vertical sample-based KNN/LSVM classification for gene expression analysis" *J. Biomed. Inform.*, vol. 37, no. 4, pp. 240–248, 2004.
- [13] Z. Yao and W. L. Ruzzo, "A regression-based K nearest neighbor algorithm for gene function prediction from heterogeneous data" *BMC Bioinformatics*, vol. 7, no. 1, p. S11, 2006.
- [14] S. Deegalla and H. Boström, "Classification of microarrays with knn: Comparison of dimensionality reduction methods" in *International Conference on Intelligent Data Engineering and Automated Learning*, 2007
- [15] R. Parry et al., "k-Nearest neighbor models for microarray gene expression analysis and clinical outcome prediction" *Pharmacogenomics J.*, vol. 10, no. 4, pp. 292–309, 2010.
- [16] Y. Su, R. Wang, C. Li, and P. Chen, "A dynamic subspace learning method for tumor classification using microarray gene expression data" in *Natural Computation (ICNC), 2011 Seventh International Conference on*, 2011.
- [17] A. Halder and S. Misra, "Semi-supervised fuzzy K-NN for cancer classification from microarray gene expression data" in *Automation, Control, Energy and Systems (ACES)*, 2014
- [18] A. Halder, S. Dey, and A. Kumar, "Active Learning Using Fuzzy k-NN for Cancer Classification from Microarray Gene Expression Data" in *Advances in Communication and Computing*, Springer, 2015, pp. 103–113.
- [19] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.
- [20] J. R. Quinlan, "C4. 5: Programming for machine learning" *Morgan Kauffmann*, vol. 38, 1993.
- [21] O. P. Netto, S. R. Nozawa, et al., "Applying decision trees to gene expression data from dna microarrays: Congress of the Brazilian Computer Society, X Workshop on Medical Informatics, 2010.
- [22] M. B. Al Snousy, et al. "Suite of decision tree-based classification algorithms on cancer gene expression data" *Egypt. Inform. J.*, 2011.
- [23] B. Ulfenborg, K. Klinga-Levan, and B. Olsson, "Classification of tumor samples from expression data using decision trunks" *Cancer Inform.*, vol. 12, p. 53, 2013.
- [24] S. A. Ludwig, D. Jakobovic, and S. Picek, "Analyzing gene expression data: Fuzzy decision tree algorithm applied to the classification of cancer data" in *Fuzzy Systems (FUZZ-IEEE)*, 2015.
- [25] A. C. Tan et al., "Ensemble machine learning on gene expression data for cancer classification" 2003.
- [26] M. Dettling, "BagBoosting for tumor classification with gene expression data" *Bioinformatics*, 2004.
- [27] A. Osareh et al., "An efficient ensemble learning method for microarray classification" *BioMed Res* 2013,
- [28] R. Díaz-Urriarte et al., "Gene selection and classification of microarray data using random forest" *BMC Bioinformatics*, 2006.
- [29] A. Statnikov, L. Wang, and C. F. Aliferis, "A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification" *BMC Bioinformatics*, vol. 9, no. 1, p. 319, 2008.
- [30] M. Pirooznia, J. Y. Yang, M. Q. Yang, and Y. Deng, "A comparative study of different machine learning methods on microarray gene expression data" *BMC Genomics*, vol. 9, no. 1, p. S13, 2008.
- [31] T.-N. Do, P. Lenca, S. Lallich, and N.-K. Pham, "Classifying very-high-dimensional data with random forests of oblique decision trees" in *Advances in knowledge discovery and management*, Springer Berlin Heidelberg, 2010.
- [32] T.-N. Do, P. N. Khang, N. H. Hoà, and N. M. Trung, "Giải thuật rừng ngẫu nhiên với luật gán nhãn cục bộ cho phân lớp" *Fair '9*, 2016.
- [33] K. Moorthy et al., "RF for gene selection and microarray data classification" *Bioinformation*, 2011.
- [34] X. Chen and H. Ishwaran, "Random forests for genomic data analysis" *Genomics*, 2012.
- [35] S. Aytekin, et al., "Microarray Gene Expression Data Classification with RF" *Int. J. Eng. Sci.*, 2016.
- [36] K. Nishiwaki, K. Kanamori, and H. Ohwada, "Gene Selection from Microarray Data for Alzheimer's Disease Using Random Forest" *Int. J. Softw. Sci. Comput. Intell. IJSSCI*, vol. 9, no. 2, pp. 14–30, 2017.
- [37] S. Mukherjee et al., "Support vector machine classification of microarray data" *AI Memo 1677*, Massachusetts Institute of Technology, 1999.
- [38] T. S. Furey, et al., "Support vector machine classification and validation of cancer tissue samples using microarray expression data" *Bioinformatics*, vol. 16, no. 10, pp. 906–914, 2000.
- [39] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines" *Mach. Learn.*, vol. 46, no. 1, pp. 389–422, 2002.
- [40] T. Li, C. Zhang, and M. Ogihara, "A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression" *Bioinformatics*, vol. 20, no. 15, pp. 2429–2437, 2004.
- [41] L.-Y. Chuang, C.-H. Yang, and L.-C. Jin, "Classification of multiple cancer types using fuzzy support vector machines and outlier detection methods" *Biomed. Eng. Appl. Basis Commun.*, vol. 17, no. 06, pp. 300–308, 2005.
- [42] J. Phan, R. Moffitt, et al., "Improvement of SVM algorithm for microarray analysis using intelligent parameter selection" in *Engineering in Medicine and Biology Society*, 2005.
- [43] E. B. Huerta, B. Duval, and J.-K. Hao, "A hybrid GA/SVM approach for gene selection and classification of microarray data" in *Workshops on Applications of Evolutionary Computation*, 2006.
- [44] E. Alba, J. Garcia-Nieto, L. Jourdan, and E.-G. Talbi, "Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms" in *Evolutionary Computation*, 2007, 2007.
- [45] S.-H. Hsieh, Z. Wang, et al., "Leukemia cancer classification based on Support Vector Machine" in *Industrial Informatics (INDIN), 2010 8th IEEE International Conference on*, 2010, pp. 819–824.

- [46] L. Vanneschi, A. Farinaccio, G. Mauri, M. Antoniotti, P. Provero, and M. Giacobini, “A comparison of machine learning techniques for survival prediction in breast cancer” *BioData Min.*, vol. 4, no. 1, p. 12, 2011.
- [47] J. Nagi, G. A. Di Caro, et al., “Convolutional neural support vector machines: hybrid visual pattern classifiers for multi-robot systems” in *Machine Learning and Applications (ICMLA)*, 2012
- [48] A. Statnikov et al., “A comprehensive evaluation of multicategory classification methods for microbiomic data” *Microbiome*, vol. 1, no. 1, p. 11, 2013.
- [49] C. D. A. Vanitha, D. Devaraj, and M. Venkatesulu, “Gene expression data classification using support vector machine and mutual information-based gene selection” *Procedia Comput. Sci.*, vol. 47, pp. 13–21, 2015.
- [50] S. Cogill and L. Wang, “Support vector machine model of developmental brain gene expression data for prioritization of Autism risk gene candidates” *Bioinformatics*, vol. 32, no. 23, pp. 3611–3618, 2016.
- [51] D. H. Hubel and T. N. Wiesel, “Receptive fields and functional architecture of monkey striate cortex” *J. Physiol.*, vol. 195, no. 1, pp. 215–243, 1968.
- [52] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [53] C.-C. Chang and C.-J. Lin, “LIBSVM: a library for support vector machines” *ACM Trans. Intell. Syst. Technol. TIST*, vol. 2, no. 3, p. 27, 2011.
- [54] Martín Abadi et al., *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015.
- [55] F. Chollet and others, *Keras*. GitHub, 2015.
- [56] L. Jinyan and L. Huiqing, *Kent ridge bio-medical dataset repository*. 2002.
- [57] R. Edgar, M. Domrachev, and A. E. Lash, “Gene Expression Omnibus: NCBI gene expression and hybridization array data repository” *Nucleic Acids Res.*, vol. 30, no. 1, pp. 207–210, 2002.

A COMPARISON OF DEEP LEARNING MODEL AND OTHER METHODS IN THE CLASSIFICATION OF GENE EXPRESSION MICROARRAY DATA

Huỳnh Phước Hải, Nguyễn Văn Hòa, Đỗ Thanh Nghi

ABSTRACT— *In recent years, deep learning models, such as Convolution Neural Networks (CNNs) have been use image classification, natural language processing and speech recognition. The advantages of CNNs is that they automatically create new features from raw data and classify them. In this paper we propose to use a convolution neural networks to classify microarray gene expression data with hight dimensions. Using ten microarray datasets from the Medical Database (Kent Ridge) and NCBI Gene Expression Omnibus (GEO) database we conclude that CNNs is better higher accuracy than simple classification methods including k nearest neighbors, decision tree. CNNs are outperformed by bagging and adaboost and have performance same to random forest and SVM.*