Fundamentos de Redes Neurais Artificiais

Victor São Paulo Ruela
Programa de Pós-Graduação em Engenharia Elétrica
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
Email: victorspruela@ufmg.br

***Resumo*— Este trabalho tem como objetivo apresentar uma revisão da literatura de redes neurais aritificiais, com enfoque nas evoluções desenvolvidas a partir dos principais trabalhos clássicos, os que estabeleceram os fundamentos desta grande área de pesquisa.**

I. Introdução

A Rede Neural Artificial (RNA) é uma classe de modelos muito popular em problemas de classificação, reconhecimento de padrões, regressão e predição [1]. Inspirado pelas características do cérebro humano, elas possuem como elementos básicos neurônios artificiais capazes de executar operações matemáticas, representando desta forma modelos de neurônios biológicos. Através de sua organização em diferentes estruturas de rede, tais modelos são capazes de se adaptar e representar funções matemáticas bastante complexas.

Diferentes representações estão presentes na literatura, as quais são classificadas de acordo com o seu nível de complexidade e requisitos coputacionais de implementação. Hipóteses básicas para regras de aprendizado de associações entre neurônios podem ser encontradas em trabalhos bastante antigos, como abordado no livro de William James em 1982 [2]. Entretanto, um grande marco desta área de pesquisa ocorreu na década de 40 após a introdução do modelo de McCulloch and Pitts (MCP) [3], o qual é a adotado atualmente nos principais modelos de RNAs.

O modelo MCP tem como saída a soma das ativações dos neurônios anteriores ponderados pelos pesos das conexões entre eles. Originalmente, uma função de ativação do tipo degrau é aplicada sobre sua saída, configurando modelo de soma-e-limiar originalmente descrito pelos autores. Este trabalho apresentou a configuração de diversas redes de neurônios MCP, com enfoque na implementação de funções lógicas. Vale a pena notar que os primeiros computadores digitais estavam surgindo nesta época, motivando esta aplicação. Entretanto, as estruturas apresentadas eram estáticas e não houve a sugestão de algum método de aprendizado para adaptá-las.

O aprendizado surgiu de forma mais concreta com o postulado de Hebb [4], originalmente publicado em 1949. De acordo com o autor, a eficiência de uma determinada sinapse que conecta dois neurônios é proporcional à co-ocorrência de ativação entre eles. Portanto, o princípio de aprendizado Hebbiano visa reforçar as conexões relevantes para as diferentes saída da rede, guiado pela correlação entre os neurônios. Considerando o neurônimo MPC, suas primeiras estruturas de rede e algoritmos de treinamento descritos na literatura são o *Adaline* [5], em 1960, e o Perceptron simples, em 1957 [6].

Após um período de euforia com a introdução destes últimos dois modelos, a área de pesquisa de RNAs sofreu um descrédito e frustação até o início dos anos 80. Isso decorreu do trabalho de Minsky e Papert [7], o qual generalizou as limitações destes modelos para problemas considerados fundamentais, como o do OU-exclusivo (XOR). O interesse só foi reativado após o re-descobrimento do algoritmo *back-propagation* para treinamento de redes de múltiplas camadas [8], as quais são capazes de superar as limitações até então existentes das redes de camada única. Destacam-se também a introdução dos mapas de Kohonen [9], redes recorrentes de Hopfield [10] e o modelo ART para aprendizado não-supervisionado [11]. Além disso, nesta época surgiram as primeiras conferências e periódicos dedicados à area de RNAs [12].

A partir destes princípios elementares, a área de RNAs evoluiu bastante nas últimas décadas. Após a introdução das primeiras regras de aprendizado, este tipo de modelo ganhou maior visibilidade e aplicabilidade para problemas reais, sendo possível encontrar uma enorme quantidade de aplicações publicadas [13]. Além disso, o aumento dos recursos computacionais disponíveis fomentou o desenvolvimento de novas técnicas para aprendizado e o aprimoramento das existentes, além de propostas de novas estruturas redes complexas capazes de lidar com problemas de grande dificuldade.

Portanto, o objetivo deste trabalho é apresentar a uma revisão da literatura contendo os principais trabalhos e entender a evolução dos diferentes modelos de redes neurais utilizadas atualmente. Partindo das referências clássicas, diferentes abordagens propostas serão analisadas de forma cronológica com o intuito de se entender a evolução desta área de pesquisa até o tempo presente. Este trabalho será divido da seguinte forma: a Seção II apresenta uma revisão da literatura com uma análise crítica dos principais trabalhos, os quais serão organizados usando como referência o livro de Simon Haykin [14] e as notas de aula. Na seção III serão apresentadas algumas das principais aplicações de RNAs publicadas. Finalmente, é feita uma conclusão deste trabalho.

II. Revisão da Literatura

A. Aprendizado Hebbiano

B. Perceptron

Proposto inicialmente por Rosenblatt [6], este é um modelo geralmente utilizado para a solução de problemas de classificação. No seu trabalho original, o autor descreve formas de adaptação dos parâmetros, ou pesos, da rede com o objetivo de reduzir a discrepância entre as saídas esperadas e estimadas e aprender associações entre os neurônios, o que é a base da indução para diversos algoritmos atuais. Este trabalho é considerado um marco na literatura por diversos autores. Embora descrito como uma rede de duas camadas, originalmente seu treinamento só considerava uma camada. Por esse motivo, o Perceptron simples é comumente descrito na forma de somente um neurônio MCP. Sua regra de aprendizado é bem direta e consiste em alterar iterativamente os pesos da rede adicionado o erro total entre as saídas medidas estimadas ponderada pelo vetor de entradas [14].

Se considerarmos uma função de ativação contínua e diferenciável, os pesos da rede poderão ser inferidos de forma explícita, através do cálculo da pseudo-inversa, ou pelo algoritmo do gradiente descendente [15]. Exemplos de funções de ativação com esta característica frequentemente empregadas na literatura são a função logística, tangente hiperbólica e linear [1]. Vale a pena ressaltar que a covergência destas abordagens está condicionada aos dados utilizados para treinamento serem linearmente independentes [15].

Rosenblatt provou a convergência da regra de aprendizado original, porém a mesma só é garantida para problemas linearmente separáveis, o que constitui a principal limitação deste modelo. O trabalho de Minsky e Papert [7] evidenciou bastante esta limitação e através da aplicação do Perceptron a diversos problemas considerados fundamentais, levou ao descrédito deste modelo pela comunidade científica. Após este trabalho, Rosenblatt avaliou diferentes arquiteturas de rede tentando superar esta limitação, mas não conseguiu chegar ao desenvolvimento do aprendizado para múltiplas camadas. Por conta disso, o Perceptron foi pouco estudado pelos próximos de 20 anos [15].

O interesse pelo Perceptron retornou na década de 80 com a descrição do método de aprendizado conhecido como *back-propagation*, o qual é capaz de aprender os pesos de redes de múltiplas camadas de forma eficiente [8]. Aliado a isso, o Perceptron de múltiplas camadas é capaz de descrever superfícies de separação não-lineares, superando a pricipal limitação do trabalho de Rosenblatt. Uma descrição mais completa desta família de modelos é feita na próxima seção.

C. Adaline

O Adaline foi inicialmente desenvolvido por Widrow em 1960 [5], sendo principamente aplicado em problemas de regressão lineares. Assim como o Perceptron, originalmente este modelo considera somente um neurônio MCP em sua formulação, entretando sua função de ativação é a identidade. Seu treinamento é formulado como um problema de otimização com custo quadrático, onde originalmente foi utilizado o algoritmo do gradiente descendente para sua solução.

Para este agoritmo, em cada iteração é dado um passo na direção oposta ao gradiente da função objetivo, resultanto em uma convergência gradual para o mínimo do problema. Este gradiente pode ser calculado de forma análitica para a estrutura de rede do Adaline, o qual é no fim proporcional à diferença entre os valores estimados e reais [5], bastante similar ao Perceptron simples. É fácil notar que o treinamento também pode ser realizado de forma direta através do cálculo da pseudo-inversa dos dados de entrada, já que este é um problema de mínimos quadrados [14].

Uma extensão proposta deste modelo é conhecida como Madaline, o qual é caracterizada por uma rede composta por vários Adalines. Existem duas principais regras para seu treinamento, conhecidas por MRI [16] and MRII [17]. É interessante notar que a MRI surgiu bem antes do algoritmo *back-propagation*, podendo ser considerada uma estrutura primitiva de uma rede de múltiplas camadas. Os leitores são referidos à [18] para uma descrição mais completa destas regras e suas aplicações.

D. Redes RBF

discutir teorema de Clover

E. Máquinas de aprendizado extremo

F. Perceptron de múltiplas camadas

O Perceptron de múltiplas camadas (MLP) é uma rede neural com uma ou mais camadas escondidas, ou seja, localizadas entre as entradas e saídas do modelo. Além disso, são caracterizadas por um alto grau de conectividade e por aplicar funções de ativação não-lineares e diferenciáveis ao modelo dos neurônios [14]. Estas camadas adicionais funcionam como detectores de características, aplicando transformações não-lineares sequenciais aos dados de forma que estes sejam mais facilmente separados nesse novo espaço. Portanto, a introdução das camadas escondidas permite modelar superfícies de decisão não-lineares, superando a limitação do Perceptron simples e Adaline.

O treinamento de redes de camada única vistas nas seções anteriores é bem direto pois podemos facilmente derivá-las analiticamente. Entretando, ao incluir camadas escondidas e alta conectividade, analisar teoricamente o comportamento do Perceptron torna-se mais difícil. Aliado a isso, o seu treinamento se torna mais complexo justamente por haver uma maior quantidade de estruturas de rede possíveis para representar os dados de entrada. O primeiro algoitmo eficiente de treinamento de tais foi formalizada por Rumelhart em 1985 [8], conhecido como *back-propagation*.

O *back-propagation* é uma técnica de aprendizado online (ou estocástica), na qual os pesos da rede são ajustados amostra-a-amostra. Ou seja, em cada época de treinamento, os dados de entrada são apresentados individualmente para a rede objetivando a minimização do erro das saídas estimadas e desejadas. O algoritmo pode ser dividido em duas etapas: na

primeira os dados são apresentados à rede mantendo os pesos fixos, e calculada a sua respectiva saída; na segunda, o sinal de erro em relação à saída esperada é calculado e propagado no sentido inverso da rede, onde ajustes sucessivos são realizados. A atualização dos pesos da rede é feita com base na técnica do gradiente descendente, cuja derivação completa será omitida deste trabalho por brevidade. O leitor pode encontrá-la em [8], [14].

Embora bastante eficiente, sua convergência pode ser lenta se o algoritmo não for usado corretamente [19]. É recomendado realizar a normalização das entradas para equalizar a taxa de atualização dos pesos entre as camadas, além de remover variávies altamente correlacionadas [20]. Ainda de acordo com [20], é sugerido o uso de sigmoides simétricas, como a tangente hiperbólica, as quais geralmente possuem maior velocidade de convergência. Além disso, é importante que os valores desejados estejam dentro dos limites da sigmoide escolhida. Outro fator importante é a taxa de treinamento: [21] apresenta o estudo de algumas técnicas para adaptação da taxa de treinamento presentes na literatura, mostrando que o seu uso é bastante benéfico.

É interessante notar que os problemas descritos anteriormente são similiares aos encontrados para a otimização de funções não-lineares e não-convexas. Ou seja, é possível analisar o treinamento do MLP como um problema de otimização e aplicar diferentes algoritmos e heurísticas disponíveis da literatura. De fato, isso é explorado por diversos autores, os quais aplicam o método de Newton [22], gradiente conjugado [23], Gauss-Newton [24], Levenberg-Marquardt [25] e Quasi-Newton [26].

Métodos de segunda ordem possuem o atrativo da convergência acelarada, mas em contra-partida é necessária a estimativa da Hessiana, o que exige mais recursos computacionais e está sujeito a problemas numéricos adicionais, sendo limitado a redes pequenas e a usar usando aprendizado por batelada. Portanto, este são fatores que devem ser levados em conta na hora de escolher o otimizador para realizar o treinamento de uma rede. Conforme argumentado por [20], o uso de informações de segunda ordem nem sempre é necessário em alguns problemas, para os quais a técnica do gradiente estocástico bem ajustada é dificilmente superado para problemas de larga escala.

Mais recentementes, alguns autores começaram a propor o uso de algoritmos evolucionários (EAs) para o treinamento do MLP. Eles são atrativos pelo fato de convergirem para o ótimo global se um tempo de treinamento suficiente estiver disponível. Entretanto, o custo computacional será bastate alto, além do ajuste dos hiper-parâmetros ser bastante trabalhoso. Em [27], os autores propõe o uso do algotimo Differential Evolution, o qual concluem que ele não apresenta desempenho superior ao *back-propagation*. Já [28] propôs uma técnica híbrida de algoritmos genéticos e backpropagation, a qual se mostrou menos suscetível a ficar presa em mínimos locais durante o treinamento. O leitor pode ser referir a [29] para um estudo mais completo de EAs e RNAs. Outro área interessante são as redes neurais evolutivas [30], para as quais além do treinamento dos pesos evolui-se também outras características da rede.

G. Generalização

H. Máquinas de vetores suporte

I. Aprendizado multiobjetivo

J. SOM

III. Aplicações

IV. Conclusões

A. Máquinas de Vetores Suporte

B. Aprendizado Multiobjetivo

Referências

[1] Anil K Jain, Jianchang Mao, and K Moidin Mohiuddin. Artificial neural networks: A tutorial. *Computer*, 29(3):31–44, 1996.

[2] William James. *Psychology, briefer course*, volume 14. Harvard University Press, 1984.

[3] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[4] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.

[5] Bernard Widrow and Marcian E Hoff. Adaptive switching circuits. Technical report, Stanford Univ Ca Stanford Electronics Labs, 1960.

[6] Frank Rosenblatt. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.

[7] Marvin Minsky and Seymour Papert. An introduction to computational geometry. *Cambridge tiass., HIT*, 1969.

[8] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[9] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69, 1982.

[10] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[11] Gail A Carpenter and Stephen Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer vision, graphics, and image processing*, 37(1):54–115, 1987.

[12] Bohdan Macukow. Neural networks–state of art, brief history, basic models and architecture. In *IFIP international conference on computer information systems and industrial management*, pages 3–14. Springer, 2016.

[13] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938, 2018.

[14] Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice-Hall, Inc., 2007.

[15] John Hertz, Anders Krogh, Richard G Palmer, and Heinz Horner. Introduction to the theory of neural computation. *PhT*, 44(12):70, 1991.

[16] Bernard Widrow. Generalization and information storage in network of adaline'neurons'. *Self-organizing systems-1962*, pages 435–462, 1962.

[17] Capt Rodney Winter and B Widrow. Madaline rule ii: A training algorithm for neural networks. In *Secondf Annual International Conference on Neural Networks*, pages 1–401, 1988.

[18] Bernard Widrow and Michael A Lehr. 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, 78(9):1415–1442, 1990.

[19] Yann LeCun. Efficient learning and second-order methods. *A tutorial at NIPS*, 93:61, 1993.

[20] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[21] George D. Magoulas, Michael N. Vrahatis, and George S Androulakis. Improving the convergence of the backpropagation algorithm using learning rate adaptation methods. *Neural Computation*, 11(7):1769–1796, 1999.

[22] Sue Becker, Yann Le Cun, et al. Improving the convergence of back-propagation learning with second order methods. In *Proceedings of the 1988 connectionist models summer school*, pages 29–37, 1988.

[23] Erik M Johansson, Farid U Dowla, and Dennis M Goodman. Backpropagation learning for multilayer feed-forward neural networks using the conjugate gradient method. *International Journal of Neural Systems*, 2(04):291–301, 1991.

[24] Roberto Battiti. First-and second-order methods for learning: between steepest descent and newton's method. *Neural computation*, 4(2):141–166, 1992.

[25] Martin T Hagan and Mohammad B Menhaj. Training feedforward networks with the marquardt algorithm. *IEEE transactions on Neural Networks*, 5(6):989–993, 1994.

[26] B Robitaille, B Marcos, M Veillette, and G Payre. Modified quasi-newton methods for training neural networks. *Computers & chemical engineering*, 20(9):1133–1140, 1996.

[27] Jarmo Ilonen, Joni-Kristian Kamarainen, and Jouni Lampinen. Differential evolution training algorithm for feed-forward neural networks. *Neural Processing Letters*, 17(1):93–105, 2003.

[28] Shifei Ding, Chunyang Su, and Junzhao Yu. An optimizing bp neural network algorithm based on genetic algorithm. *Artificial intelligence review*, 36(2):153–162, 2011.

[29] Seyedali Mirjalili. Evolutionary algorithms and neural networks. *Studies in Computational Intelligence*, 2019.

[30] Xin Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447, 1999.