Deep Learning

Taiane Ramos - 6426955

São Paulo, 28 de junho de 2015

Sumário

Introdução

Estamos vivendo a era da informação, onde bilhões de dados são gerados todos os dias por usuários da internet. Estes dados são chamados de Big Data e muitas empresas tentam usa-los para retirar informações úteis para o mercado. Publicidade direcionada, resultados de busca que se relacionam com os interesses do usuário e sugestões de página relacionada podem utilizar informações retiradas dessas grandes massas de dados para melhorar seus resultados. Porém, retirar informações úteis desses dados tem sido um desafio e muitas vezes são coletados muito mais dados do que se consegue processar.

Apesar de deep learning já ter sido proposto desde a década de 80, apenas atualmente temos poder computacional o suficiente para que este método de aprendizado de máquina seja viável. Atualmente, o deep learning tem se mostrado uma forma eficiente de retirar informações do big data e por este motivo tem sido muito explorado por grandes empresas como a Google, Facebook e Baidu.

O algoritmo de deep learning exige bastante poder computacional para obter um resultado satisfatório. Quanto mais completo o modelo da rede, melhores os resultados e para isso, precisa-se de muitos núcleos de processamento. As técnicas de paralelismo como clusterização e uso de placas gráficas (GPUs), tem se mostrado necessárias para o funcionamento de algoritmos de deep learning em tempo viável.

O que é Deep Learning?

Aprendizado de máquina, ou Machine learning, não é um tema novo, porém é bastante atual. O primeiro perceptron data dos anos 50 e a proposta deste algoritmo era tentar criar uma inteligência para classificação independente de programação específica. Muitos outros algoritmos de classificaçao foram propostos desde então, mas o que promete ser um divisor de águas da inteligência artificial é o deep learning.

O modelo tradicional se baseia em aprendizado supervisionado. No aprendizado supervisionado, m algoritmo aprende a partir de entradas previamente classificadas e depois tenta aplicar o que aprendeu em novos dados diferentes dos que foram usados no treinamento. Deep learning utiliza algoritmos de aprendizado não-supervisionado. Isto é, dada uma entrada de características do que se deseja classificar, sem nenhuma informação prévia de dados similares já classificados, e o algoritmo define uma classificação para a entrada.

A ideia do deep learning é que o aprendizado seja feito como o cérebro humano faria. Quando uma criança começa a aprender, não existe sempre alguém presente contando o que ela está entendendo corretamente ou não. As crianças retiram padrões das informações e

começam a classificar sem necessidade de serem ensinadas. O deep learning propõe esta abordagem.

A principal diferença entre machine learning tradicional que utiliza algoritmos não-supervisionados e deep learning é o método utilizado para extração de características relevantes para a classificação. Na abordagem tradicional, um programador faz a engenharia de forma manual, determinando quais são as características relevantes para que um certo tipo de dado seja diferenciados das outras possíveis classes. Em deep learning, o método de extração de características é feito de forma automática. O algoritmo é exposto à uma grande quantidade de dados e dinamicamente determina quais são as características relevantes para separar as entradas em classes e quais classes serão consideradas.

Esta extração de características é feita em camadas, de forma similar ao funcionamento do cérebro para a visão. Pesquisas já mostraram que o processo de visão é feito em camadas, de forma que cada camada retira uma informaçao diferente da imagem, como detecção de bordas, por exemplo. Ao final, temos a classificação da imagem em algum dos grupos já conhecidos.

Deep learning está intimamente ligado à big data. Muito se falou de big data nos últimos anos, pois o volume de dados gerado é cada vez maior, porém, não muito se sabe sobre como aproveitar essa informaçao eficientemente. A técnica de deep learning exige uma grande quantidade de dados de entrada para o treinamento da rede. A maior parte das redes que utilizam deep learning fazem seu treinamento com dados disponíveis na internet, como vídeos do Youtube ou imagens retornadas na busca do Google.

Em contrapartida, o deep learning tem se mostrado promissor para a melhor utilização de todos estes dados disponíveis. Sem uma forma de extrair significado relevante dos dados, não se pode aproveitar toda a informações contida no chamado big data. Uma das aplicações possíveis, por exemplo, seria melhorar a publicidade direcionada, que hoje se baseia apenas em tags.

Por que usar HPC para Deep Learning?

Antes de mais nada é importante avaliar se é realmente necessário usar um método de paralelismo para a resoluçao do problema. Métodos de paralelismo não são simples de serem implementados e podem trazer complicações adicionais para a implementação. O custo de comunicação entre computadores em uma abordagem distribuída pode nao compensar o ganho de processamento, ou o gargalo de transmissão de dados para GPUs pode limitar mais o tempo do que o desempenho de processamento da CPU. Além disso, abordagens já conhecidas como boas para algoritmos em modo single-thread não necessariamente serão boas abordagens para uma arquitetura distribuída ou concorrente.

Para ilustrar que paralelizar o código nem sempre implica em ganho de velocidade, temos na figura 1 uma reprodução de um gráfico apresentado por Andrew Ng et al. (2012). Neste trabalho, a equipe da Google apresenta algumas comparações de desempenho de deep

learning usando diferentes algoritmos, diferentes tipos de hardware (GPUs e CPUs) e com granularidades de paralelismo diferentes. Neste gráfico podemos observar que o pico de desempenho para o reconhecimento de fala é atingido com 8 cores e a partir disso o desempenho cai, pois o custo de comunicação entre as máquinas se torna superior ao ganho, como descrito no trabalho citado.

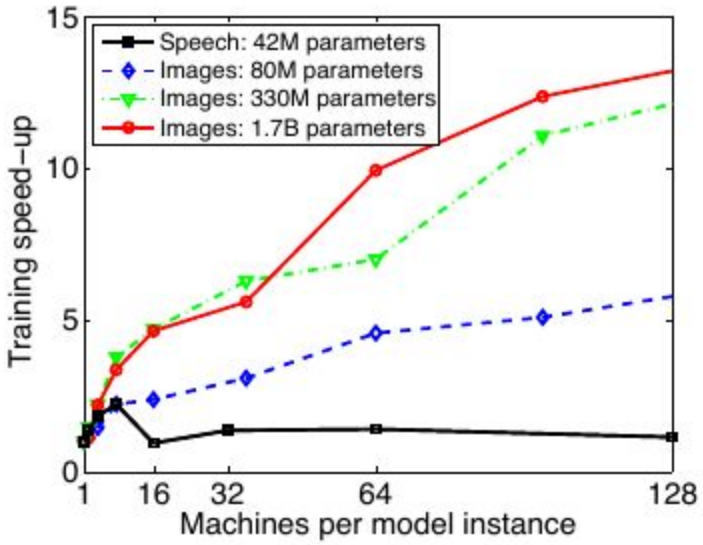

Figura 1: Reprodução do gráfico de Andrew Ng et al. (2012) que apresenta o desempenho por número de máquinas utilizadas para o processamento do algoritmo.

Outro ponto a se considerar é que não existem ferramentas de alto-nível para este tipo de análise e mesmo as bibliotecas atualmente disponiveis não são tão otimizadas quanto as bibliotecas atualmente disponíveis para programação single-thread. Um exemplo apresentado por C. J. Lin da Universidade de Taiwan envolve multiplicação de matrizes. Em sua apresentação “Big-Data Analytics: Chalenge and Opportunities” de 2014, Lin apresenta uma implementaçao simples e intuitiva de Multiplicação de Matrizes e roda este exemplo com uma matriz quadrada de lado 3000. O tempo de execução deste algoritmo é 3m24.843s. A mesma multiplicação de matrizes foi executada no MatLab e o tempo de execução foi 4.095059s.

O MatLab consegue ter um desempenho tão melhor com seu algoritmo, pois utiliza a biblioteca BLAS, que considera a arquitetura do hardware e faz a multiplicação das matrizes em blocos de forma que haja uma baixa taxa de cache miss. Já o algoritmo desenvolvido de forma intuitiva não leva em conta a localização dos dados na memória e tem uma taxa de cache miss muito maior. C. J. Lin diz que nossa condiçao atual de desenvolvimento em HPC é similar a este exemplo. Estamos tentando desenvolver algoritmos matemáticos (analise de dados e clusterização) em uma arquitetura complexa (clusters e GPUs) antes que tenhamos algo como o BLAS para nos ajudar.

Apesar dos pontos contra o desenvolvimento utilizando técnicas de HPC, algumas vezes o uso de computação distribuída ou GPUs em um mesmo servidor se faz necessário. No

trabalho de Michelle Bank e Eric Brill de 2001 foram analisados alguns dos principais métodos de aprendizado de máquina supervisionado e foi medido o desempenho de cada algoritmo conforme o tamanho da entrada. A figura 2 é uma reprodução do gráfico apresentado no trabalho e podemos ver que todos os métodos estudados tem a mesma tendência de crescimento conforme os dados de entrada. Com base nos dados de desempenho dos algoritmos, Bank e Brill concluiram que não é o melhor algoritmo que tem melhor desempenho, mas o que tem mais dados de entrada para o treinamento.

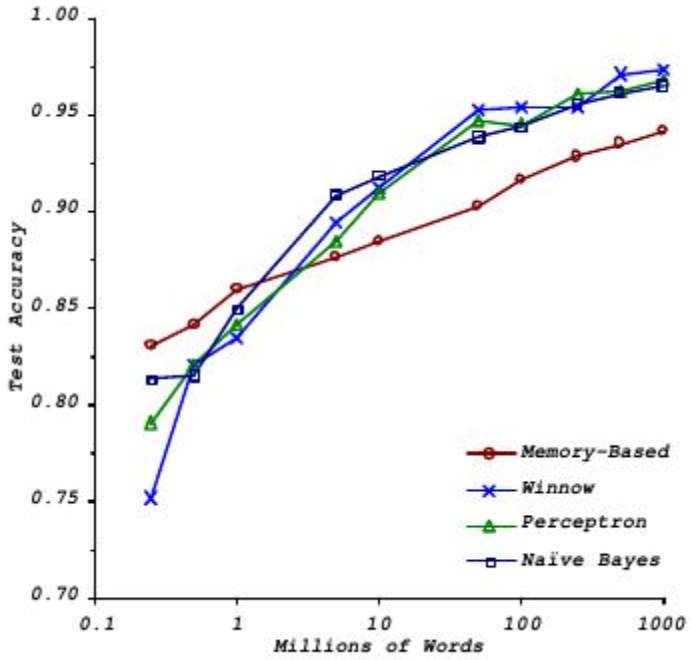

Figura 2: Reproduçao do gráfico apresentado por Michelle Bank e Eric Brill de 2001.

No trabalho de Adam Coates et al. (2011) foi feita uma analise similar para algoritmos de aprendizado não supervisionado. Neste trabalho, todos os algoritmos possuiam base de dados ilimitadas, pois os dados eram obtidos da internet conforme se fazia necessário. Em analises não supervisionadas, não se tem uma base de dados com labels para o treinamento. As características que serão usadas para classificar são obtidas conforme o algoritmo. Desta forma, quanto mais robusto o algoritmo, mais caracteristicas dos dados analisados podem ser retiradas, e consequentemente melhor é a analise. A figura 3 é uma reproduçao do gráfico apresentado no trabalho de Coates et al. e pode-se observar que todos os algoritmos estudados exibem um padrão de crescimento similar conforme a quantidade de caracteristicas estraídas dos dados. Novamente, como disse Andrew Ng em sua palestra apresentada no NIPS 2012 ao apresentar este mesmo gráfico como exemplo, não é o melhor algoritmo que tem um melhor desempenho, mas o algoritmo que consegue extrair mais características para a analise dos dados.

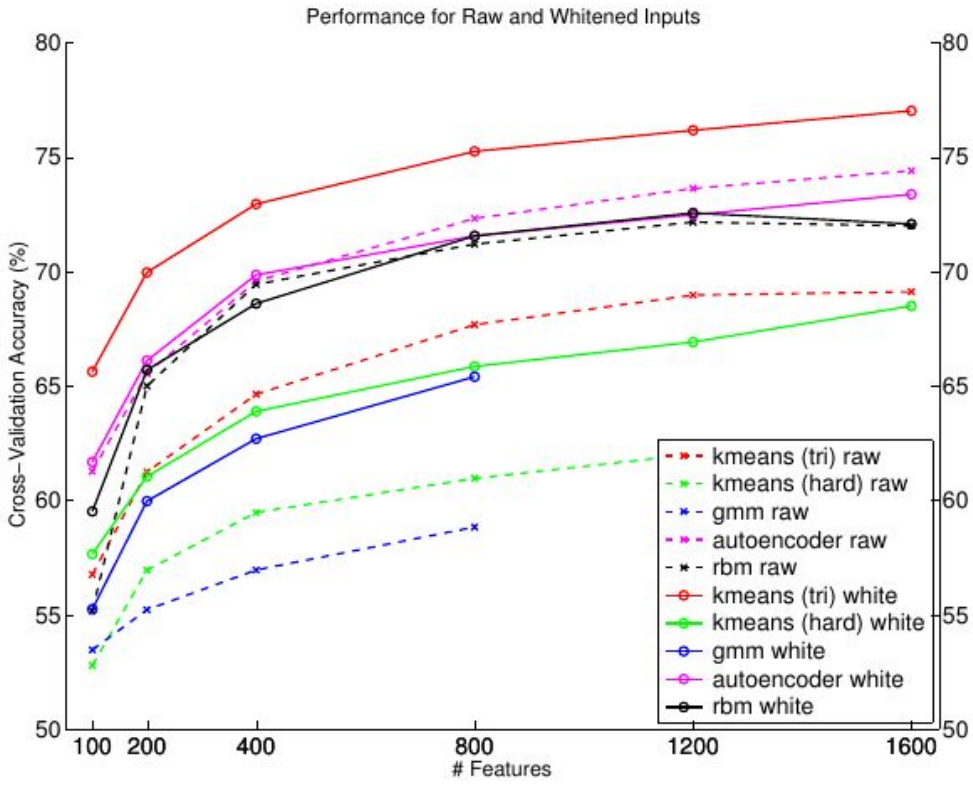

Figura 3: Reprodução do gráfico apresentado por Coates et al. (2011).

Com base nas análises de algoritmos de aprendizado de máquina apresentados, podemos ver que se faz necessário o uso de muitos dados e algoritmos robustos que exigem muito poder computacional. Para o aprendizado supervisionado, os dados precisam estar pré-classificados, o que não torna possível obter dados diretamente da internet. Se faz necessário uma base de dados. Para poder analisar um volume de dados grande o suficiente que o erro dos algoritmos se torne pequeno, pode ser necessário o uso de mais de uma máquina, dependendo da memória disponível em cada máquina. Uma dificuldade adicional neste caso é saber reunir a informação obtida separadamente em cada máquina.

Para o caso de algoritmos não supervisionados, os dados analisados não são pré-classificados, então pode-se obter a base de dados da internet de forma que a quantidade de dados para cada algoritmo e ilimitada. A quantidade de características retirada das imagens para a classificação é que torna um algoritmo melhor ou pior. Para retirar uma grande quantidade de características da imagem, é necessário um algoritmo robusto, o que exige bastante em processamento. Muitas vezes as pesquisas limitam a robustez do algoritmo pelo tempo de processamento necessário, pois uma máquina não consegue processar algoritmos muito pesados em um tempo viável para a pesquisa. Uma solução possível é aumentar o poder de processamento da máquina fazendo uso de GPUs ou distribuir o processamento em um cluster de máquinas, conforme a disponibilidade de hardware e flexibilidade do algoritmo. Discutiremos na próxima sessão com mais detalhes como escolher a abordagem de HPC para melhorar o aprendizado de máquina.

Qual abordagem de HPC utilizar?

Como já foi comentado, só é recomendado utilizar uma abordagem de HPC quando não se pode resolver o problema de forma satisfatória com programação single-thread. Hoje em dia memória e processamento são bem mais baratos e não é incomum que alguns servidores possuam memória ram e processamento suficiente para resolver muitos dos problemas de analise de dados sem a necessidade de utilizar mais de uma máquina. Se não for o caso e realmente for necessário o uso de HPC, discutiremos algumas alternativas.

Se só existe uma opção de hardware, tente fazer o melhor com o que se tem. Se o que está disponível são algumas máquinas com pouco poder computacional, pode-se tentar usar uma abordagem distribuída com Hadoop ou MPI para aproveitar várias máquinas simples e tentar fazer um processamento um pouco mais poderoso, lidando com o problema da comunicação entre as máquinas.

Se todos os seus dados podem ser copiados facilmente para uma única máquina a qual tem capacidade de memória o bastante para conter os dados, porém e necessário mais poder de processamento que a CPU do computador pode fazer, uma opção é usar GPUs. As GPUs não são simples de serem programadas, por terem uma arquitetura bastante diferente da arquitetura de CPUs com a qual estamos habituados a lidar, porém, existe bastante material na internet para auxiliar quem queira aprender. Também já existem diversos frameworks para esse tipo de desenvolvimento, o que melhora também o desempenho das aplicações.

Se seus dados são de uma escala multi-máquina, isto é, não cabem na memória de apenas uma máquina, será necessário usar um cluster. No caso de clusters, novamente podemos escolher entre usar apenas CPUs ou combinar com GPUs. A mesma analise é válida. Se o processamento for um limitante, mesmo para a abordagem distribuída, pode-se fazer uso de GPUs no cluster para ganhar desempenho, se o problema e apenas o volume de dados, talvez seja melhor fazer o processamento apenas em CPUs.

Aplicações de Deep Learning

Merck Drug Discovery Competition

“Merck drug discovery competition” foi um torneio divulgado na internet pela fabricante de remédios Merck na tentativa de conseguir um método de simulação de substâncias mais eficiente do que eles possuiam até o momento. A simulação de substâncias químicas é fundamental para a industria farmaceutica, pois no processo de desenvolvimento de uma nova

droga é preciso saber se a substancia desenvolvida realmente terá o resultado esperado e se ocorrem reações adversas pela interação da molécula do medicamento com outras moléculas.

O algoritmo vencedor desta competição é uma rede neural de deep learning desenvolvida por estudantes da universidade de Toronto. Eles desenvolveram a rede de forma que a camada mais baixa foi programada pela Merck com a informação sobre como as moléculas interagem entre si. Este algorítmo foi executado em GPUs para acelerar o processamento e conseguiu o melhor resultado competindo com abordagens de aprendizado de máquina tradicional.

Google Brain

O projeto de inteligência artificial usando deep learning da Google, o Google Brain, já tem mostrado resultados desde 2012. Em um dos primeiros trabalhos publicados pela equipe da Google liderada por Andrew Ng em 2012, eles relataram os primeiros resultados que o projeto obteve utilizando deep learning. Este trabalho foi uma prova de conceito, pois eles queriam determinar se era possível criar um algoritmo para reconhecimento de faces e outros objetos com base em imagens não catalogadas e sem prover nenhum tipo de informação para o algoritmo se a imagem continha ou não uma face.

O algoritmo de deep learning proposto utilizou 37 mil imagens de 200 x 200 pixels obtidas na internet, sem nenhum tipo de informação associada a elas. Dentre as imagens, 13026 continham faces e as restantes eram imagens para atrapalhar o treinamento. Este algoritmo foi executado em mil máquinas, cada uma com 16 cores de processamento, por 3 dias. O resultado obtido foi bastante satisfatório para a equipe, pois eles verificaram que o algoritmo conseguiu desenvolver um filtro para identificar faces, como a proposta inicial queria provar ser possivel e também outros elementos contidos nas imagens, como gatos e corpos humanos. A imagem 4 ilustra o filtro gerado por um dos neurônios da rede para identificação de faces.

Após estas pesquisas iniciais a Google já aplicou a técnica em muitos outros componentes da empresa. O reconhecimento de voz utilizado no Android recebeu melhorias do algoritmo, passando a apresentar 25% menos erros que a versão de reconhecimento de voz usada anteriormente. Outra aplicação foi para melhorar a busca por imagens da Google. Anteriormente esta busca era feita apenas por tags relacionadas à imagem e agora já começa a ter um filtro adicional que utiliza deep learning para melhorar os resultados retornados.

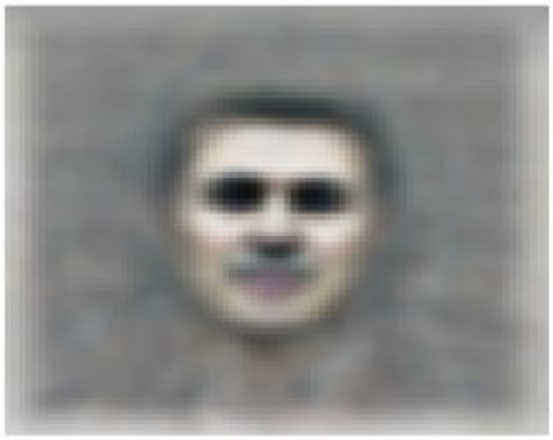

Figura 4: Filtro gerado por um dos neurônios da rede para reconhecer faces.

Identificação de Placas de Trânsito

Uma aplicaçao bastante comum de machine learning é a identificação de placas de trânsito. No trabalho de J.Stallkamp et al. de 2012 foi utilizada a técnica de deep learning para a tarefa tendo um resultado melhor que outros métodos de aprendizado de máquina. O desempenho da rede deep learning foi de 99,46% de acertos, enquanto a média do desempenho de humanos classificando as mesmas clacas foi de 98,81%, o que mostra que a classificalção da rede é melhor até que a classificação de humanos. Mesmo o melhor humano teve um desempenho menor que a rede, sendo de 99,22% de acertos.

A identificação de placas de trânsito é uma preocupaçao constante das pesquisas de carros autonomos e já vem sendo pesquisada à muitos anos. Atualmente os melhores resultados obtidos são conseguidos com redes deep learning, ficando a frente de várias outras técnicas de machine learning.

Possíveis Aplicações Futuras

O deep learning está prorcionando a extração de informações dos dados que antes nenhum algoritmo de inteligência artificial conseguia extrair. Pelo fato da técnica tentar simular o aprendizado de um cérebro, os pesquisadores acreditam que será possivel extrair um significado associado ao dado além do representado. Por exemplo, será possível associar frases e dar um significado a um texto e até formar um conhecimento ou uma espécie de pensamento.

Uma das aplicações propostas é para tradução. Atualmente os métodos de tradução utilizam frases em várias linguas pré-relacionadas. Os próprios usuarios de sistemas como o Google Tradutor entram frases que acham mais apropriadas para suas traduções e fazem pequenas correções de forma que o algoritmo aprende com a supervisão dos usuários.

Utilizando deep learning seria possível entrar uma frase em inglês em uma rede que reconhece inglês e tratar o padrão de ativação dos neurônios dentro da rede como se fosse o “pensamento” formato por aquela frase. Espera-se que o mesmo “pensamento” submetido à uma rede que entende francês resulte em uma frase em francês que é a tradução para a frase originalmente submetida em inglês. Desta forma, teríamos uma tradução baseada em significados em vez de uma tradução por palavras.

Outra aplicação possível seria melhorar as buscas por textos na internet. Atualmente as buscas são feita por palavras chave, mas não podemos saber se um texto é contra ou a favor de uma determinada opinião, por exemplo. Coisas como ironia não podem ser tratadas por este tipo de abordagem. Utilizando deep learning seria possível construir um significado a partir do relacionamento entre as frases do texto e avaliar se aquele documento é ou não relevante para a busca baseado no conhecimento ou ideia que ele apresenta, além das palavras.

Conclusão

Deep learning tem se mostrado uma técnica promissora para extrair informações de dados da internet. Muitos trabalhos tem mostrado a capacidade de aprendizado deste algoritmo ao ser exposto a um grande volume de dados. Esta técnica tem a vantagem de separar categorias não previamente estabelecidas, o que é vantajoso para aplicações com muitas categorias, como o caso de imagens e textos.

Técnicas de paralelismo estão sendo usadas por muitas pesquisas da área para que seja possível processar um algoritmo robusto em tempo viável. Quanto mais robusto o algoritmo, mais características são extraidas dos dados para serem usadas na classificação, o que torna fundamental um alto poder computacional para se conseguir um bom resultado de classificação.

Em muitos problemas onde usualmente se aplica aprendizado de máquina, como classificação de placas de transito, a tecnica de deep learning tem apresentado resultados melhores que outras técnicas tradicionais. Muitas vezes até consegue resultados melhores do que um humano conseguiria fazendo a classifiação manualmente. Por este motivo, e esperado que este algoritmo seja aplicado para muitos tipos de problemas da computação. Também é possível que novas aplicações para os dados surjam em função da maior facilidade de extrair conhecimento destes dados.

Bibliografía

Large Scale Distributed Deep Networks, Jeffrey Dean, Greg S. Corrado, Rajat Monga, Kai Chen,
Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc'Aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, Andrew Y. Ng, 2012

Building High-level Features Using Large Scale Unsupervised Learning - Quoc V. Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S. Corrado, Jeff Dean, Andrew Y. Ng, 2012

Coates, A.; Lee, H. & Ng, A. (2011), An analysis of single-layer networks in unsupervised feature learning, *in* Geoffrey Gordon; David Dunson & Miroslav Dudík, ed., 'Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics' , JMLR W&CP, , pp. 215--223 .

Scaling to Very Very Large Corpora for Natural Language Disambiguation - Michele Banko and Eric Brill, 2001
An Analysis of Single-Layer Networks in Unsupervised Feature Learning - Adam Coat, Honglak Lee, Andrew Y. Ng, 2011

Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition - J. Stallkamp ,M. Schlipsing, J. Salmen, C. Ige, 2012

Big-data Analytics: Challenges and Opportunities, 2014 - C. J. Lin - Universidade de Taiwan

Deep Learning: Intelligence from Big Data - MIT Enterprise Forum Bay Area, 2014

Machine Learning and AI via Brain simulations - Andrew Ng, 2014

http://hunch.net/?p=151364, acessado em 22 de junho de 2014

https://developer.nvidia.com/deep-learning, acessado em 22 de junho de 2014

http://blogs.nvidia.com/blog/2014/03/25/machine-learning/, acessado em 22 de junho de 2014

http://fastml.com/the-emperors-new-clothes-distributed-machine-learning/, acessado em 22 de junho de 2014