Uso de aceleradores para tarefas de Machine Learning em servidores de alto desempenho

Ricardo Mazza Zago

ricardoz@outlook.com

Faculdade de Engenharia Elétrica e de Computação - FEEC

Universidade Estadual de Campinas - Unicamp

Resumo

O uso de aprendizado de máquina na nuvem vem crescendo, com as mais diversas aplicações sendo classificação de imagens, reconhecimento de objetos, tradução e classificação de textos, reconhecimento de áudio, dentre muitas outras. Para possibilitar todas estas aplicações novos *hardwares* e abordagens são necessárias. Este trabalho visa detalhar os novos *hardwares*, já lançados ou não, e servir como um meio de guiar para a próxima geração de servidores que serão implantados. Foram discutidos conceitos básicos de redes neurais, padrões de mercado e *hardwares*.

Key words: Machine Learning; Servidores; Aceleradores.

1 Introdução

Problemas que no passado eram de difícil resolução atualmente podem ser resolvidos com um menor esforço utilizando aprendizado de máquina. Classificação de imagens, reconhecimento de objetos, tradução e classificação de textos, reconhecimento de áudio, etc. Seja no ambiente universitário, com diversas pesquisas fazendo uso de *hardware* amplamente disponível no mercado, as GPUs, unidade de processamento gráfico. Seja no ambiente empresarial, onde também são feitos grandes investimentos na tecnologia, desde desenvolvimento de *software* até de *hardware* específico para a tarefa.

Na área de *software* podemos destacar alguns *frameworks* como: Tensorflow [1], PyTorch [2], Caffe [3], e Microsoft Cognitive Toolkit [4]. Os *softwares* desenvolvidos por grandes empresas são realmente utilizados em ambiente de produção. Um claro exemplo disso é o Google Tensorflow, que é a base das ferramentas online do Google.

Na questão de *hardware* para treinamento e inferência dos modelos as GPUs são muito utilizadas, como as produzidas pela Nvidia. No entanto, outras empresas estão entrando no mercado como AMD, além de *hardwares* desenvolvidos especialmente para este fim como o Tensor Processing Unit (TPU) do Google, Intel Xeon Phi, Nervana e Movidius, mesmo, soluções utilizando FPGAs (*Field Programmable Gate Array*) da Altera e Xilinx, além de diversas pequenas *startups*, como a Cambricon. O TPU do Google foi desenvolvida "in house" e é de uso exclusivo do Google, onde as demais pessoas apenas recentemente podem alugar tempo de uso via plataforma online.

Atualmente a questão de qual *hardware* utilizar em servidores para aprendizado de máquina ainda é uma questão em aberto. Dispositivos da Nvidia são amplamente utilizados, mas isso deve-se principalmente por ser a única solução madura no mercado, mas com a entrada de concorrentes este panorama deve mudar.

2 Conceitos básicos

Antes da análise dos *hardwares* disponíveis, uma introdução aos conceitos fundamentais de aprendizado de

[1] Desenvolvido pelo Google, disponível em: `https://www.tensorflow.org/`

[2] Desenvolvido principalmente pelo Facebook, disponível em: `https://pytorch.org/`

[3] Desenvolvido pelo Berkeley Vision and Learning Center, disponível em: `http://caffe.berkeleyvision.org/`

[4] Disponível em: `https://www.microsoft.com/en-us/cognitive-toolkit/`

máquina será feita. Inicialmente este campo começou com a observação do funcionamento do cérebro humano, onde cientistas passaram a tentar imitar seu funcionamento. Até hoje, não existe nenhum sistema que consiga simular um cérebro completo, mas alguns fundamentos de seu funcionamento foram modelados e implementados em dispositivos que simulam um neurônio chamado de perceptron.

O esquema de um neurônio é mostrado na Figura 1, onde ele pode ser comunicar com outros neurônios pelos dendritos e axônios. Um perceptron assim como mostrado na Figura 2, recebe diversas entradas, seja fornecida pelo usuário ou de outros perceptrons, multiplica estas entradas por um peso, que é definido durante o treinamento. Posteriormente, o peso destas entradas é somado e o resultado passa por uma função matemática.

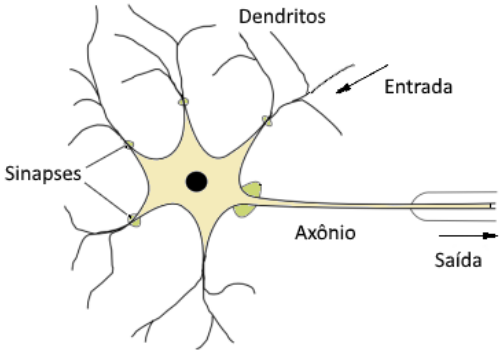

Figura 1. Elementos básicos de um neurônio [5] .

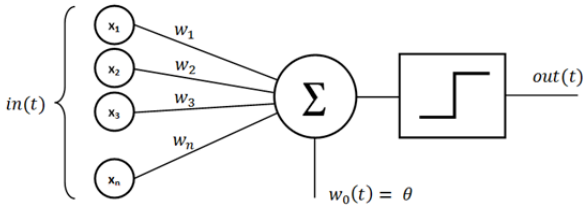

Figura 2. Perceptron utilizado em uma rede neural, repare que para cada entrada existe um peso w_n e posteriormente estas entradas são somadas e passam por uma função [6] .

Diversas funções podem ser utilizadas. No geral, os requerimentos são que ela possa ser derivada e seja estritamente crescente. O requerimento de derivação existe devido ao algoritmo de treinamento das redes, conhecido como backpropagation, que utiliza derivações sucessivas para ajustar os pesos da rede. Enquanto ser estritamente crescente é importante para quando for utilizada a derivada não exista mínimos ou máximos locais.

[5] Com adaptações, disponível em: `http://galaxy.agh.edu.pl/~vlsi/AI/intro1/networks.html`.

[6] Com adaptações, disponível em: `https://github.com/cdipaolo/goml/tree/master/perceptron`.

Na Figura 3 são mostradas as 3 principais funções utilizadas: a sigmoide, a tangente hiperbólica e a ReLU (*Rectified Linear Unit*). Detalhe que mesmo que a ReLU não deseja derivável em 0, ele possui derivadas triviais para todas as demais posições, o que a torna atraente devido a menor necessidade de operações.

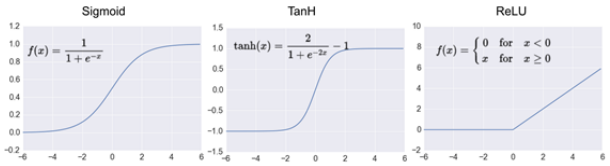

Figura 3. Funções Sigmoide, Tangente Hiperbólica e ReLU [7] .

Uma certa quantidade de dados é utilizada para o treinamento da rede. A topologia da rede é montada antes de iniciar o treinamento e mantida constante durante todo o processo.

Uma rede neural é a junção de neurônios interligados. Todos podem estar em apenas uma camada ou mesmo em várias camadas, podendo chegar até a dezenas. A Figura 4 ilustra uma rede neural com duas camadas escondidas, sendo conhecida como MLP (*Multilayer Perceptron*), onde a necessidade para esta nomenclatura é a presença de ao menos uma camada escondida.

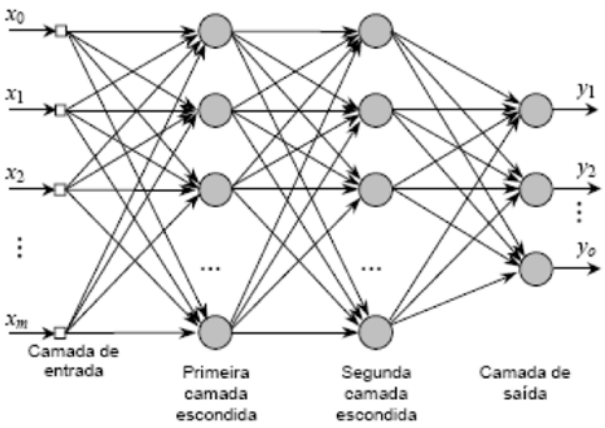

Figura 4. Rede neural MLP com duas camadas escondidas [8] .

Para ilustrar a aplicação de redes neurais vamos trabalhar com um exemplo. Um dataset [9] muito famoso é o MNIST [Lecun et al., 1998]. Ele é um conjunto de dados

[7] Com adaptações, disponível em: `https://medium.com/@shiyan/xavier-initialization-and-batch-normalization-my-understanding-b5b91268c25c`.

[8] Com adaptações, disponível em: `http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1413-70542010000200002`.

[9] Conjunto de dados de diversas características e tamanhos para uso em Data Science.

de números escritos a mão, sua resolução é 28x28 e possui tons de cinza. Na Figura 5 alguns dígitos presentes neste conjunto são mostrados.

Figura 5. Exemplos de números do dataset MNIST [10] .

Uma das maneiras de se treinar uma rede neural utilizando um dataset como este é tomar todas as 784 entradas (28x28) e utilizar como input da rede neural. Já na outra ponta, podem existir 10 saídas, onde ela é treinada para a saída referente ao número ser 1 e as demais 0. Desta maneira, na hora de avaliar um digito, tarefa conhecida como inferência, basta tomar a maior saída, que esta será o digito que a rede neural acredita ser o correto.

Porém, tomar as 784 entradas como um vetor, não como uma matriz, remove as informações de proximidade. Ou seja, em uma imagem, o valor individual, sozinho, de um pixel pode não ter informações, mas ele em conjunto com os de sua proximidade podem trazer informações relevantes, desta maneira é utilizado um método conhecido como convolução para considerar pixel próximos. Este método demonstrado na Figura 6, toma uma função, que pode ser o máximo, mínimo, média, soma dos elementos, dentre outros, de um espaço na matriz e move o resultado para outra matriz. Esta operação reduz as dimensões da matriz e pode ser feita sucessivamente ou mesmo utilizando diferentes formas para gerar diversas outras matrizes.

3 Necessidade de hardware específico

As operações em redes neurais são basicamente dividas em duas etapas: o treinamento da rede, onde é necessária uma grande capacidade de processamento, e mesmo em grandes servidores, a depender da rede, pode levar várias semanas. E a fase de inferência ou predição, onde uma

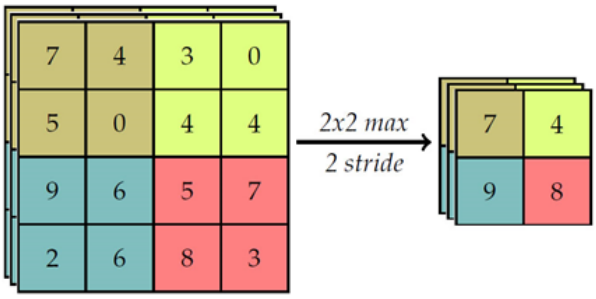

Figura 6. Exemplo de uma operação de convolução que faz uso da operação máximo [11] .

rede já treinada é utilizada para seu fim. Um exemplo é utilizá-la para classificar fotos em um *Smartphone* localmente, como a Apple faz.

Ambas as partes podem ser executadas em uma CPU comum, mas em um tempo maior e em processadores especiais. Placas de vídeo geralmente são utilizadas para o treinamento de rede, enquanto a primeira versão do TPU do Google para a fase de inferência. Os modelos desenvolvidos pelo Google são executados no lado de servidor, desta forma, mesmo que o modelo seja simples, com a quantidade de requisições é necessário muito processamento.

4 Linguagens de programação, formatos numéricos e instruções específicas

O lançamento da arquitetura Tesla pela Nvidia em 2006 permitiu a programação das GPUs com a linguagem de programação CUDA, muito parecida com a Linguagem C++. Esta linguagem, inerentemente paralela, aliada a alta capacidade de processamento das GPUs permitiu o desenvolvimento de redes neurais maiores, com mais parâmetros e a própria criação de *frameworks* que facilitaram o treinamento delas.

Desta maneira, a linguagem CUDA se tornou o padrão de desenvolvimento de *frameworks*. Porém, como ela é proprietária, criou-se um virtual monopólio nas soluções da Nvidia. Consequentemente, foi criada a linguagem OpenCL pelo Khronos Group como um padrão aberto a ser adotado entre os diversos outros fornecedores de GPUs [12] , vindo a ser sua versão aberta, que, no entanto, não ganhou suporte nos *frameworks* de *machine learning*.

A AMD recentemente começou o desenvolvimento de um sistema chamado ROCm [13] , que dentre várias coisas permite a conversão de programas em CUDA para

[10] Com adaptações, disponível em: `https://upload.wikimedia.org/wikipedia/commons/2/27/MnistExamples.png`.

[11] Com adaptações, disponível em: `http://www.computacaointeligente.com.br/artigos/redes-neurais-convolutivas-cnn/`.

[12] Considerando o mercado de dispositivos móveis, existem muitas empresas, dentre elas: ARM, Imagination, Qualcomm e Vivante.

[13] `https://rocm.github.io/index.html`

código C++ portável [14]. O uso deste sistema ainda é bem reduzido, no entanto, passou a permitir executar o framework Tensorflow em placas da AMD.

A aritmética de ponto flutuante comumente utilizada é baseada em precisão simples (32 bits) e dupla precisão (64 bits). No geral, para o treinamento de rede neurais não é necessária tanta resolução, onde 16 bits pode ser mais do que o suficiente, o que diminui os requerimentos de *hardware* da FPU (*Float Point Unit*). Em vista disso, alguns formatos numéricos como o Bfloat16, que possui 16 bits e o mesmo intervalo que um número IEEE de precisão simples, sacrificando a resolução [15]. E, também, o Intel Flexpoint [Köster et al., 2017] foram criados.

Algumas instruções SIMD (*Single Instruction, Multiple Data*) utilizadas em processadores podem acelerar a aplicação de aprendizado de máquina em CPUs da Intel. Novos processadores de servidores e a próxima linha desktop oferecem suporte as instruções conhecidas como AVX-512. Estas instruções possibilitam ao processador trabalhar com treinamento e inferência com maior capacidade que os atuais, sendo uma das novidades da Intel.

5 GPUs

As GPUs no passado eram utilizadas exclusivamente para renderização de jogos digitais. No entanto, com o passar do tempo sua arquitetura majoritariamente paralela permitiu que elas atingissem um desempenho muito superior as CPUs. É importante destacar, que este desempenho superior é em relação a tarefas que podem ser paralelizadas, enquanto tarefas cuja execução é serial são executas muito mais lentamente do que nas CPUs.

Os códigos utilizados em *Machine Learning* basicamente são multiplicações de matrizes, desta maneira podem ser paralelizadas atingindo desempenhos muito maiores. As duas maiores fabricantes de GPUs são as americanas Nvidia e AMD, onde a Nvidia, como já citado, domina o mercado.

5.1 Nvidia

A capacidade de cálculo das GPUs, cresceu muito ao longo dos últimos anos. A capacidade teórica, por exemplo, de uma GeForce GTX Titan é de 4500 GFLOPS, enquanto um processador Intel com arquitetura Sandy Bridge não chega a 250 GFLOPS, isso, claro, não considera processadores com instruções AVX-512. A evolução das placsa Nvidia até o lançamento GTX Titan é mostrada na Figura 7. No geral, devido aos diversos *overheads* não é possível atingir tal desempenho. Uma placa mais moderna como uma Titan Xp possui 12000 GFLOPS.

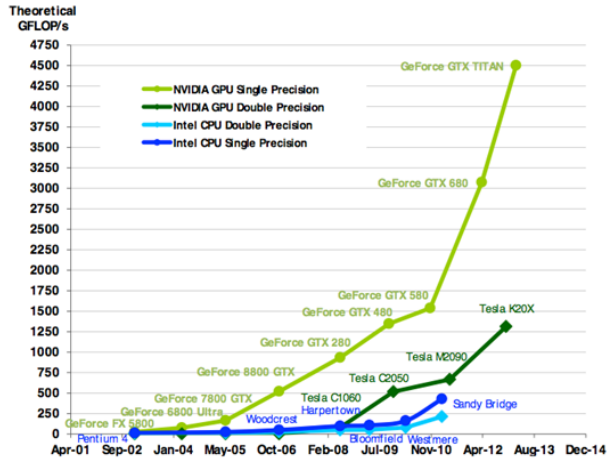

Figura 7. Linha do tempo da capacidade de processamento de GPUs e processadores da Intel [16].

A Nvidia em 2017 alterou os termos de uso proibindo o uso de placas domésticas, chamadas de Geforce, em servidores para aprendizado de máquina, liberando apenas a lista Tesla, que possuí um custo maior [17].

A placa com maior capacidade de processamento atualmente é a Tesla V100, mostrada na Figura 8, com arquitetura Volta, superior a Pascal vendida a usuários domésticos, possuindo capacidade de processamento de 14899 GFLOPS quando se considera precisão simples. A placa que faz uso deste chip possui um TDP (*Thermal Design Power*) entre 250 e 300 W, possui memória HBM2 (*High Bandwidth Memory*), podendo atingir uma banda de memória de até 900 GB/s.

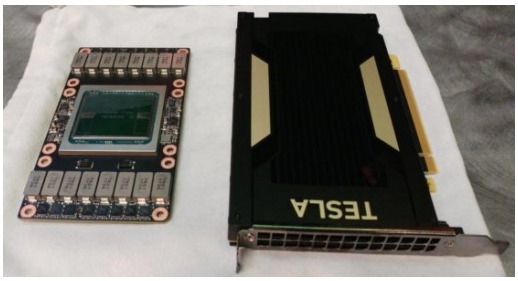

Figura 8. Placas Tesla V100 [18].

Uma vantagem desta arquitetura é a capacidade de realizar cálculos com os chamados "*tensor cores*", que traba-

[14] https://github.com/ROCm-Developer-Tools/HIP

[15] https://en.wikipedia.org/wiki/Bfloat16_floating-point_format

[16] Com adaptações, disponível em: http://homepages.math.uic.edu/~jan/mcs572/mcs572notes/lec27.html.

[17] www.theregister.co.uk/2018/01/03/nvidia_server_gpus/

[18] Com adaptações, disponível em: https://linustechtips.com/main/topic/783321-titan-volta-pics/.

lham com dados de 16 bits e o acumulador de 32 bits [19], desta maneira obtendo uma capacidade de cálculo de 125 TFLOPS, que chega a 1000 TFLOPS em sistemas como o Nvidia DGX1 com suas 8 GPUs [20]

Outras empresas também desenvolvem soluções para servidores. A Supermicro, por exemplo, vende servidores, como o 4028GR-TVRT [21], mostrado na Figura 9, que possuem capacidade para até 8 GPUs V100, que ocupam 4U de espaço. Desta maneira é possível atingir vários PFLOPS em um único rack.

Figura 9. Servidor Supermicro 4028GR-TVRT e o rack superior onde são acomodados os 8 chips Tesla V100.

Neste caso foi considerada a GPU com maior capacidade de processamento, desconsiderando o custo. Caso o custo seja considerado, placas de gerações anteriores são mais atraentes.

5.2 AMD

A AMD é uma rival histórica da Nvidia no mercado de placas de vídeo. Em 2007 ela fez a compra da ATI, tradicional empresa canadense do ramo e com o passar o tempo alterou o nome da divisão para AMD, unificando a marca. Placas da AMD possuem capacidade de processamento equivalente as da Nvidia, ficando um pouco para trás devido a recente crise que a empresa passou, que a obrigou a diminuir substancialmente seus investimentos em pesquisa e desenvolvimento.

A AMD lançou 3 placas aceleradoras com foco no mercado de aprendizado de máquina, são elas: a Radeon Instinct MI25 Accelerator, Radeon Instinct MI8 Accelerator e Radeon Instinct MI6 Accelerator [22], que possuem respectivamente 12,29, 8,19 e 5,73 TFLOPS de capacidade de processamento, sendo competitivas com as soluções da Nvidia. Elas possuem vantagem no custo, mas pecam no suporte a *frameworks*.

[19] `http://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf`
[20] `www.nvidia.com/en-us/data-center/dgx-1/`
[21] Informações disponíveis em: `https://www.supermicro.com/products/system/4U/4028/SYS-4028GR-TVRT.cfm`
[22] `https://www.amd.com/en/graphics/servers-radeon-instinct-mi`

5.3 Intel Xeon e Xeon Phi

Processadores x86, utilizados em computadores pessoais tem uma alta capacidade de processamento serial, mas pecam em capacidade de processamento paralelo. A Intel, em continuação ao seu fracassado projeto de GPU Larabee, desenvolveu processadores com muitos núcleos x86 simples, vendidos na linha Xeon Phi.

Enquanto os Xeon tradicionais possuem uma quantidade limitada de núcleos, apenas recentemente chegando até 28. Os processadores com arquitetura Skylake de servidores já possuem algumas instruções AVX-512 e são utilizados pelo Facebook em suas tarefas de inferência.

É interessante notar, que a justificativa do Facebook é que como são tarefas das mais variadas os processadores Xeon oferecem maior flexibilidade. No entanto, estes resultados foram apresentados em uma conferência da Intel, o que de certa maneira pode enviesar os resultados [Feldman, 2018].

Uma justificativa para o uso de Xeons para treinamento é a grande capacidade de memória, limitada a 32 GB no caso de GPUs como a V100, onde alguns datasets podem não caber.

Já no caso do Xeon Phi, existiam duas versões destes processadores: as que utilizam placas discretas, como GPUs se comunicando via interface PCIe com o processador principal, que recentemente foi descontinuado. E os que utilizam sockets de processador, conforme a Figura 10.

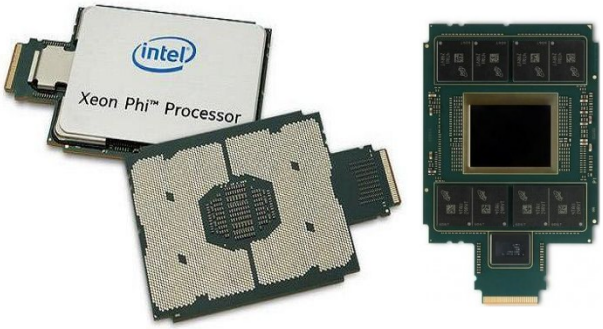

Figura 10. Processador Xeon Phi codinome Knights Landing para socket LGA 3647 e ele sem o dissipador de alumínio.

Estes processadores, ao contrário das aceleradoras PCIe, podem ser utilizados como principais, para executar sistemas operacionais completos. Além disso, alguns modelos possuem dois conectores de rede Omni-path chegando a 100 Gbps de banda [23].

[23] `https://www.golem.de/news/knights-landing-intel-veroeffentlicht-xeon-phi-mit-bis-zu-7-teraflops-1606-121642.html`.

A nova revisão, lançada em dezembro de 2017, com codinome Knights Mill, em sua versão mais poderosa possui 72 núcleos e 288 threads, 36 MB de cachê L2, 16 de memória MCDRAM com largura de banda de mais de 400 GB/s e acesso a até 384 GB de memória DDR4.

Considerando que cada núcleo possui capacidade de 128 FLOPS por ciclo [24] , os 72 núcleos a 1,6 GHz tem capacidade de 14,75 TFLOPS consumindo 320 W de potência.

5.4 Intel Nervana

A Nervana foi adquirida pela Intel em 2016, eles propunham uma solução completa, desde um *framework* até o desenvolvimento de um ASIC (*Application Specific Integrated Circuits*), mas inicialmente utilizavam uma solução de *hardware* da Nvidia.

A primeira versão, com codinome Lake Crest estava prevista para 2017, mas apenas agora chega para parceiros da Intel, cujo foco é pavimentar o desenvolvimento de *software*. A versão para consumidores será o Spring Crest, denominada NNP-L1000, previsto para o final de 2019, prometendo de 3 a 4 vezes mais capacidade de treinamento do que a versão atual [Feldman, 2018]. Ambas as versões já suportam o formato Bfloat16, que deve ser o padrão de formato numérico dos próximos sistemas de aprendizado de máquina.

A versão atual, o Lake Crest, possui uma capacidade de processamento de aproximadamente 40 TFLOPS com TDP de 210 W, enquanto a Nvidia V100 chega a 125 TFLOPS com TDP de 300 W. No entanto, a Intel argumenta que a capacidade real, utilizando um *software* de multiplicação de matrizes fica em 27 TFLOPS na V100 e 38 TFLOPS na sua solução. Como o *hardware* ainda não foi lançamento e mesmo quando for, apenas poucos parceiros da Intel terão acesso, estas informações não podem ser confirmadas.

5.5 Intel Movidius

Assim como a Nervana, a Intel fez a aquisição da empresa em 2016. O foco desta empresa é um pouco diferente e talvez não devesse estar na lista, mas como no fim as soluções lutam entre si, a inclusão se justifica.

A companhia possui um ASIC chamado Myriad 2, que possui diversos núcleos cujo foco é o funcionamento com baixo consumo de potência. Ele é vendido para consumidores finais via um dispositivo USB, formato parecido com um *flashdrive*.

Foi desenvolvido para ser utilizado em dispositivos ARM como computadores, drones, robôs e dispositivos IoT (*Internet of Things*) e em monitoramento por câmera, onde pode realizar reconhecimento de pessoas e objetivos. Possui capacidade de processamento de até 150 GFLOPS consumindo menos de 1 W de potência [25] .

A empresa também fazia parte do chamado Google Project Tango, que utilizava diversos sensores para criar smartphones para a chamada realidade aumentada. Repare que aqui temos uma disputa entre dois conceitos de aprendizado de máquina, enquanto as soluções vistas anteriormente se focam na nuvem esta fica no próprio *smartphone*.

5.6 Google TPU

O Google faz um uso imenso de aprendizado de máquina em suas aplicações, nas mais diversas áreas. Talvez a mais transparente para o usuário final seja o aplicativo Google Photos, onde uma busca como "fotos de praia", "piscina", dentre outras, na maioria das vezes, retorna o resultado esperado dentro das fotos do usuário. Não é necessário o usuário previamente classificar as imagens, é tudo feito automaticamente.

Com esses grandes requerimentos de processamento o Google passou a desenvolver internamente chips para aplicação, que resultaram até hoje em 3 versões. A primeira é bem documentada [Jouppi et al., 2017], existem *benchmarks* em comparação com uma GPU Tesla K80 da Nvidia e um processador de arquitetura Haswell da Intel.

Para facilitar implantação, a placa é conectada em um slot padrão de SSDs, que possui interface PCIe 3.0. Uma placa com a conexão é mostrada na Figura 11. Uma diferença entre as demais soluções é que ela faz uso de números inteiros de 8 bits, portanto não é possível calcular sua performance em FLOPS, já que é uma medida de ponto flutuante, mas sim em operações, que chegam em pico a até 92 TOps.

Nos anos seguintes o Google lançou duas novas versões, não tão bem documentadas. A segunda versão, já suportando aritmética de ponto flutuante, tem uma capacidade de processamento de 45 TFLOPS por TPU, totalizando 180 TFLOPS com 4 chips por módulo e um total de 11,5 PFLOPS por rack [Bright, 2017], que é mostrado na Figura 12.

Já a terceira versão dobra a capacidade de processamento por chip e novamente aumenta a densidade nos racks em 4 vezes, desta maneira atingindo uma melhora de 8 vezes em relação a segunda geração.

[24] `https://fuse.wikichip.org/news/653/intel-silently-launches-knights-mill/`

[25] Disponível em: `https://www.tomshardware.com/news/movidius-fathom-neural-compute-stick,31694.html`

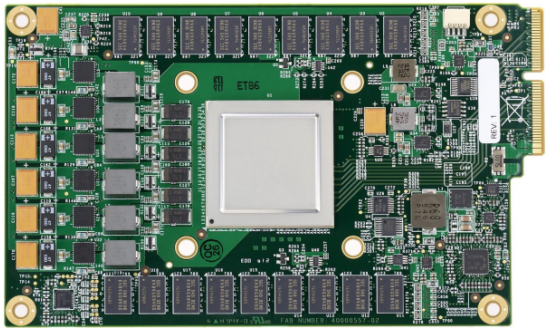

Figura 11. Placa do TPU de primeira geração [Bright, 2017].

Figura 12. TPUs do Google de segunda geração [Bright, 2017].

5.7 Cambricon

Como já citado no caso da Intel Movidius, existe uma demanda para o uso de aprendizado de máquina no próprio dispositivo, sem enviar informações para serviços em nuvem. Esta é a abordagem utilizada pela Apple, que adicionou esta capacidade em seus últimos SoCs (System of Chip) com tecnologia da CEVA [Eassa, 2017].

A Huawei fabricante chinesa de equipamentos de redes, telecomunicações e *smartphones*, utiliza SoCs da HiSilicon, uma subsidiária. A última iteração destes chips, o Kirin 970, conta com um processador especializado para aprendizado de máquina, chamado de *Neural Processing Unit* (NPU), cuja arquitetura foi desenvolvida por uma startup chinesa chamada Cambricon [Liu et al., 2016].

Esta empresa placa desenvolveu aceleradoras discretas PCIe 3.0 para uso em servidores e desktops. A empresa promete um desempenho de 64 TFLOPS (considerando aritmética de ponto flutuante de 16 bits) com clock de 1,0 GHz e consumo de 80 W, enquanto a placa sendo executada no modo de alto-desempenho a 1,3 GHz tem desempenho de 83,2 TFLOPS e consumo de 110 W. Uma foto do chip MLU100 fabricado pela TSMC em 16 nm é mostrada na Figura 13 [26] .

[26] Disponível em: `http://www.cambricon.com/products/MLU/`.

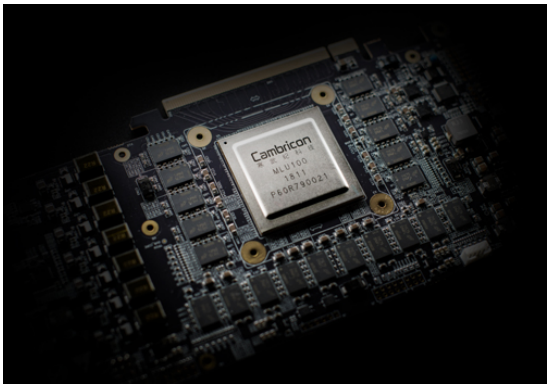

Figura 13. Placa com chip Cambricon MLU100 fabricado pela TSMC em 16 nm.

5.8 FPGAs

Por fim, outra solução para aprendizado de máquina é o uso de FPGA (Field Programmable Gate Array), que são chips programáveis, cuja lógica pode ser alterada pelo desenvolvedor. A Microsoft, através do *Project Brainwave*, vem utilizado esta solução, adicionando a FPGA Intel Stratix 10 280, mostrada na Figura 14, nos servidores e atingindo uma capacidade efetiva de 39,5 TFLOPS [Chung et al., 2018]. Um dos motivos para o uso é a baixa latência, que não pode ser atingida com CPUs ou GPUs.

Figura 14. Placa utilizada em servidores da Microsoft com a FPGA Intel Stratix 10 [Freund, 2017].

5.9 Comparações

Cada uma das opções apresenta *hardwares* com capacidades distintas. Por exemplo, a primeira versão do TPU do Google apenas suportava cálculo utilizando inteiros de 8 bits, enquanto a Cambricon também suporta ponto flutuante de 16 bits, mas sem oferecer suporte a outras precisões.

O próprio sistema de instruções e cálculo de matrizes no TPU do Google e na Cambricon é especialmente desenvolvido para este tipo de cálculo, sem muita flexibilidade, é de supor que na Nervana sejaparecido. Enquanto soluções não dedicadas como as CPUs, GPUs e FPGAs possuem uma flexibilidade total no desenvolvimento, qualquer novo algoritmo pode ser implementado, já que é um processador de propósito geral. Novos algoritmos estão em desenvolvimento, pois o campos de aprendizado de máquina ainda está em estágio inicial.

Tabela 1
Principais concorrentes em aprendizado de máquina

	Cambricon	Radeon Instinct MI25	Tesla V100	Xeon Phi	Nervana	TPU 3
Clock (GHz)	1,3	1,5	1,455	1,6	-	-
Banda de memória (GB/s)	102,4	484	900	Mais de 400	-	-
Quantidade de memória (GB)	32	16	32	16 MCDRAM + 384 DDR4	-	-
Half Precision (TFLOPS)	83,2	24,6	30	29,5	-	-
Single Precision (TFLOPS)	-	12,3	15	14,75	-	-
Double Precision (TFLOPS)	-	0,768	7,5	7,38	-	-
Formato de dados especial	166,4 TOPS	-	125 TFLOPS	-	40 TFLOPS	90 TFLOPS
TDP (W)	110	300	300	320	210	-
Form Factor	PCIe	PCIe	PCIe/SXM2	Socket LGA3647	PCIe	PCIe

Na Tabela 1 são mostradas as diversas opções citadas durante o texto. Deve-se atentar que não é possível uma comparação direta entre as diversas plataformas. As capacidades de processamento são de pico, pode ser muito difícil ou mesmo impossível implementar certos algoritmos e fazer uso dessa grande capacidade.

6 Conclusão

Foram apresentadas diversas soluções que podem ser utilizadas para aprendizado de máquina. Cada solução tem os seus pontos fortes e fracos. Mesmo as grandes empresas de tecnologia ainda divergem sobre qual caminho seguir. De um lado temos o Google desenvolvendo soluções próprias, a seguir o Facebook utilizando processadores x86 e, por fim, a Microsoft com soluções FPGA.

A solução que atualmente se destaca nas demais empresas é a Nvidia, que possui *hardwares* com os mais variados custos. Isto permite seu uso desde ambiente acadêmico nos notebooks dos alunos até em servidores de alto desempenho, tudo com o mesmo *framework*.

A AMD está voltando a investir, depois de anos estagnada devido à crise na empresa. Na realidade, em capacidade de processamento, as GPUs da AMD não ficam nem um pouco atrás das demais soluções. O problema surge no suporte a *frameworks*, que é muito limitando, o que deve mudar nos próximos anos com o projeto ROCm.

A Intel tem uma linha de produtos bem confusa. De um lado possui soluções clássicas baseadas em x86, com núcleos de alto desempenho e com grande quantidade de núcleos, uma solução em desenvolvimento para concorrer com a Nvidia (Nervana), uma solução de baixo consumo de energia (Movidius) e as próprias FPGAs. Estas soluções deveriam ser mais integradas, a Nervana e a Movidius terem a mesma arquitetura, para poderem executar o mesmo *software*.

O TPU do Google é proprietária, então não é uma opção de mercado e, provavelmente, será aposentada quando uma empresa apresentar uma solução adotada como padrão de mercado. Querendo ou não, a Nvidia desenvolve placas de vídeo, parte do chip ainda é destinada a este mercado, não exclusivo para aprendizado de máquina, desta maneira, não são totalmente otimizadas para a tarefa.

Enquanto não existir um padrão, seja de instruções como x86 e ARM, mas para aprendizado de máquina ou mesmo um padrão de armazenamento de números, não é possível definir qual sistema é o melhor. Talvez a melhor escolha seja se pautar pela flexibilidade proporcionada pelas GPUs, que podem executar qualquer novo algoritmo em um custo relativamente baixo nos modelos de entrada.

Referências

[Bright, 2017] Bright, P. (2017). Google brings 45 teraflops tensor flow processors to its compute cloud. Website.

[Chung et al., 2018] Chung, E., Fowers, J., Ovtcharov, K., Papamichael, M., Caulfield, A., Massengill, T., Liu, M., Lo, D., Alkalay, S., Haselman, M., Abeydeera, M., Adams, L., Angepat, H., Boehn, C., Chiou, D., Firestein, O., Forin, A., Gatlin, K. S., Ghandi, M., Heil, S., Holohan, K., Husseini, A. E., Juhasz, T., Kagi, K., Kovvuri, R., Lanka, S., van Megen, F., Mukhortov, D., Patel, P., Perez, B., Rapsang, A., Reinhardt, S., Rouhani, B., Sapek, A., Seera, R., Shekar, S., Sridharan, B., Weisz, G., Woods, L., Xiao, P. Y., Zhang, D., Zhao, R., and Burger, D. (2018). Serving DNNs in real time at datacenter scale with project brainwave. *IEEE Micro*, 38(2):8–20.

[Eassa, 2017] Eassa, A. (2017). Is this apple inc.'s secret new chip supplier? Website.

[Feldman, 2018] Feldman, M. (2018). Intel lays out new roadmap for ai portfolio. Website.

[Freund, 2017] Freund, K. (2017). Microsoft: Fpga wins versus google tpus for ai.

[Jouppi et al., 2017] Jouppi, N. P., Young, C., Patil, N., Patterson, D. A., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., Boyle, R., Cantin, P., Chao, C., Clark, C., Coriell, J., Daley, M., Dau, M., Dean, J., Gelb, B., Ghaemmaghami, T. V., Gottipati, R., Gulland, W., Hagmann, R., Ho, R. C., Hogberg, D., Hu, J., Hundt, R., Hurt, D., Ibarz, J., Jaffey, A., Jaworski, A., Kaplan, A., Khaitan, H., Koch, A., Kumar, N., Lacy, S., Laudon, J., Law, J., Le, D., Leary, C., Liu, Z., Lucke, K., Lundin, A., MacKean, G., Maggiore, A.,

Mahony, M., Miller, K., Nagarajan, R., Narayanaswami, R., Ni, R., Nix, K., Norrie, T., Omernick, M., Penukonda, N., Phelps, A., Ross, J., Salek, A., Samadiani, E., Severn, C., Sizikov, G., Snelham, M., Souter, J., Steinberg, D., Swing, A., Tan, M., Thorson, G., Tian, B., Toma, H., Tuttle, E., Vasudevan, V., Walter, R., Wang, W., Wilcox, E., and Yoon, D. H. (2017). In-datacenter performance analysis of a tensor processing unit. *CoRR*, abs/1704.04760.

[Köster et al., 2017] Köster, U., Webb, T., Wang, X., Nassar, M., Bansal, A. K., Constable, W., Elibol, O., Hall, S., Hornof, L., Khosrowshahi, A., Kloss, C., Pai, R. J., and Rao, N. (2017). Flexpoint: An adaptive numerical format for efficient training of deep neural networks. *CoRR*, abs/1711.02213.

[Lecun et al., 1998] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[Liu et al., 2016] Liu, S., Du, Z., Tao, J., Han, D., Luo, T., Xie, Y., Chen, Y., and Chen, T. (2016). Cambricon: An instruction set architecture for neural networks. In *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*. IEEE.