

Aceleradores e *Deep Learning*

André V. Lopes, Carybé, Victor Mayrink

Data: 5 de Julho de 2017

IME-USP

1. O Que é Deep Learning?
2. GPGPUs
3. FPGAs
4. Hardwares Dedicados - ASICs
5. Go
6. Conclusão

O Que é Deep Learning?

O Que é Deep Learning?

Deep Learning é um ramo da área de aprendizado de máquina que envolve a utilização de redes neurais profundas (Deep Neural Networks) e com muitas camadas (>2) de processamento (*deep*).

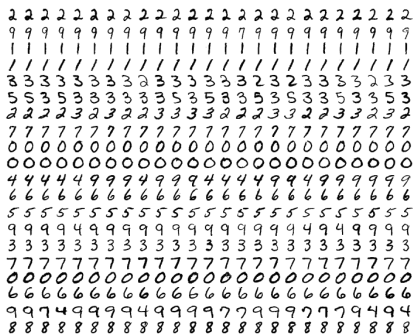
- ⊙ Redes neurais convolucionais (Convolutional Neural networks)
- ⊙ Redes neurais recorrentes (Recurrent Neural Networks)
- ⊙ Perceptron de múltiplas camadas (Multilayer perceptron)
- ⊙ Deep Belief Networks

Normalmente são usados em aplicações que envolvem processamento de sinais, imagens, vídeos e etc.

Em geral são modelos de aprendizado de máquina que exigem uma **carga computacional extrema**.

Exemplos

Mnist é um conjunto de imagens de dígitos manuscritos de 0 a 9



A 20x20 grid of handwritten digits from the MNIST dataset. The digits are arranged in a grid, with each row and column containing a variety of handwritten styles. The digits are: 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2; 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9; 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1; 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1; 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3; 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5; 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2; 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7; 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0; 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0; 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4; 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6; 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5; 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9; 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3; 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7; 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0; 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6; 9, 9, 7, 4, 9, 9, 9, 4, 9, 9, 9, 7, 7, 9, 9, 7, 7, 9, 4, 9, 4; 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8.

Outros exemplos (competições Kaggle):

- ⦿ Transforming How We Diagnose Heart Disease (2nd DSB)
- ⦿ Passenger Screening Algorithm Challenge (\$1.5M).

Objetivo: fazer com que um computador seja capaz de realizar tarefas que normalmente são difíceis de se programar seguindo a lógica tradicional:

- ⦿ Reconhecer os objetos de uma imagem
- ⦿ Identificar a voz de uma pessoa em um sinal de áudio
- ⦿ Dirigir um veículo

Redes Neurais Artificiais são modelos computacionais bio-inspirados que procuram emular a forma como o cérebro humano funciona.

Mas afinal, como nós aprendemos?

O processo de aprendizado ocorre quando o indivíduo, após um **número suficiente de experiências**, é capaz de assimilar e organizar informações para **construir e ampliar seu próprio conhecimento**

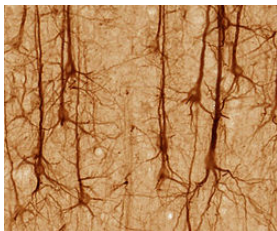
“What we want is a machine that can learn from experience.”

— Alan Turing, 1947

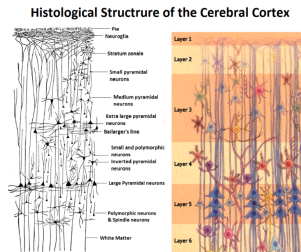
Como funciona?

O cérebro é formado por uma rede complexa de células elementares, os **neurônios**.

A rede neural é estruturada em **camadas**



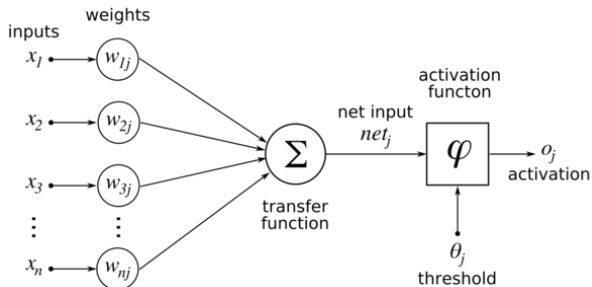
(a) Neurônios



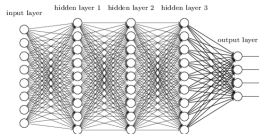
(b) Estrutura em camadas

O cérebro humano possui, em média, cerca de **86 bilhões** de neurônios. [Frederico *et. al.*, 2009]

Modelo computacional do neurônio



(a) Modelo computacional do neurônio



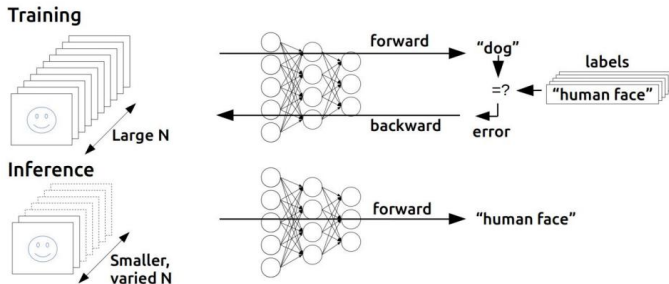
(b) Deep Neural Network

Custo computacional

Duas fases:

O maior esforço computacional ocorre na fase de **treinamento**

Uma vez que o modelo está treinado, a **predição** é relativamente simples.



Agravantes:

- ⦿ Muitas camadas de processamento
- ⦿ Muitos neurônios por camada
- ⦿ Grande volume de dados
- ⦿ Muitas iterações até a convergência
- ⦿ Cross Validation & Ensemble

Potencial de paralelização

Apesar do grande custo computacional, essas tarefas tem muito potencial de paralelização.

- ⦿ Os neurônios da rede compreendem blocos elementares de processamento
- ⦿ As observações da amostra de treinamento podem ser processadas de forma independente (em uma mesma época)

Assim, é possível obter um ganho de desempenho considerável com a utilização de **hardwares aceleradores**.

GPGPUs



Aceleração para treinamento de redes profundas

- ⊙ **Deep Learning** envolve grande quantidade de multiplicação de matrizes e convoluções, as quais podem ser paralelizadas em GPU.
- ⊙ **Deep Neural Networks** contém estruturas uniformes, tal que, em uma camada há neurônios de tipos idênticos que fazem o mesmo tipo de computação.
- ⊙ O algoritmo de treinamento envolve pontos flutuantes.

Nvidia Titan X (6600 GFLOPS) **vs.** i7-5960x (380 GFLOPS)

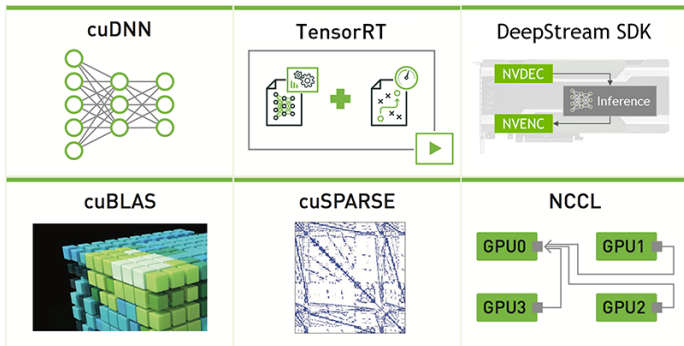
As ferramentas mais utilizadas são o **CUDA** e o **CuDNN**.

CuDNN (NVIDIA CUDA® Deep Neural Network library)

- ⦿ Biblioteca de funções para o desenvolvimento de redes neurais profundas
- ⦿ Contém implementações otimizadas de rotinas padrões tal como forward e backward convolution, pooling, normalization, and activation layers
- ⦿ Faz parte do **NVIDIA Deep Learning SDK**

NVIDIA Deep Learning SDK

NVIDIA Deep Learning SDK



<https://developer.nvidia.com/deep-learning-software>

O **NVIDIA Deep Learning SDK** é usado em diversos frameworks, tais quais, TensorFlow, Caffe, CNTK, **Theano** e Torch.

Caffe



DL4J
Deeplearning4j

K
KERAS

Microsoft
CNTK

MatConvNet

MINERVA

mxnet



theano

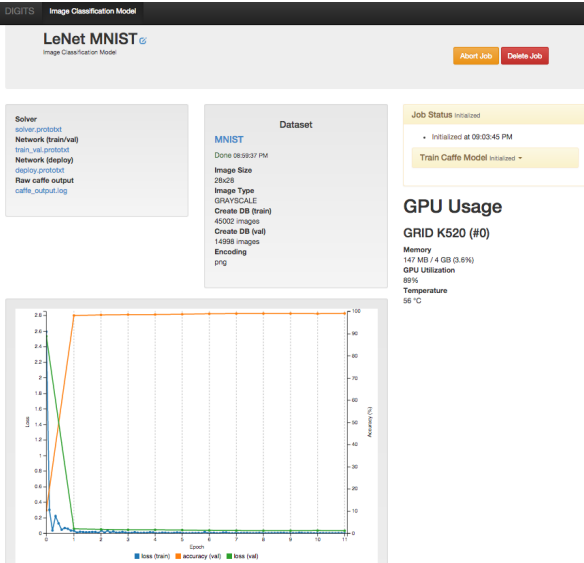


Há diversas bibliotecas que utilizam **Theano** como back-end, permitindo uma fácil implementação de redes neurais, entre elas, **Keras** e **Lasagne**.

NVIDIA Deep Learning GPU Training System **DIGITS** permite treinar e criar rapidamente redes neurais profundas (DNN's).

- ⦿ Criação, treinamento e visualização de DNN's para classificação, segmentação e detecção
- ⦿ Uso/Download de modelos pre-treinados, assim como AlexNet, GoogleNet e LeNet
- ⦿ Busca por hiper-parâmetros ótimos de taxa de treinamento e tamanho de mini-batch
- ⦿ Permite uso de Multi-GPU

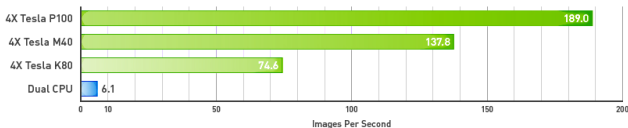
<https://developer.nvidia.com/digits>



<http://christopher5106.github.io/big/data/2015/07/16/deep-learning-install-caffe-cudnn-cuda-for-digits-python-on-ubuntu-14-04.html>

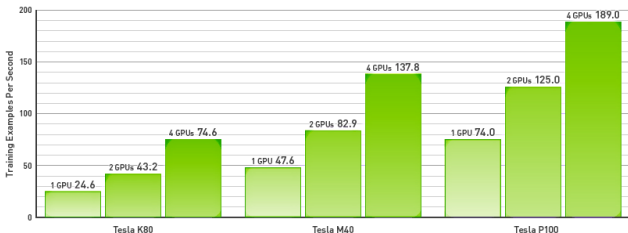
Benchmarks - GPU vs GPU vs MultiGPU

TensorFlow Image Classification Training Performance



Dual CPU System: Dual Intel E5-2699 v4 @ 3.6 GHz | GPU-Accelerated System: Single Intel E5-2699 v4 @ 3.6 GHz, NVIDIA® Tesla® K80/M40/P100 (PCIe) | Google's Inception v3 image classification network, 500 steps, 64 Batch Size, cuDNN v5.1

TensorFlow Inception v3 Training Scalable Performance on Multi-GPU Node



GPU-Accelerated System: Single Intel E5-2699 v4 @ 3.6 GHz, NVIDIA® Tesla® K80/M40/P100 (PCIe) | Google's Inception v3 image classification network, 500 steps; 64 Batch Size, cuDNN v5.1

<http://www.nvidia.com/object/gpu-accelerated-applications-tensorflow-benchmarks.html>

O próximo benchmark compara GPU's e um modelo de CPU no modelo de rede neural VGG19. O modelo VGG19 utiliza:

- ⊙ 16 camadas convolucionais
- ⊙ 5 camadas de max-pooling
- ⊙ 3 Camadas conectadas (FeedFoward)

Fonte : github.com/jcjohnson/cnn-benchmarks

Benchmarks

GPU	Memory	Architecture	CUDA Cores	FP32 TFLOPS	Release Date
Pascal Titan X	12GB GDDR5X	Pascal	3584	10.16	August 2016
GTX 1080	8GB GDDR5X	Pascal	2560	8.87	May 2016
GTX 1080 Ti	11GB GDDR5X	Pascal	3584	10.6	March 2017
Maxwell Titan X	12GB GDDR5	Maxwell	3072	6.14	March 2015

Há também a comparação com o dual Intel Xeon E5-2630 v3.
Dual Intel Xeon E5-2630 v3 (8 cores each plus hyperthreading means 32 threads) and 64GB RAM

Benchmarks

GPU	cuDNN	Forward (ms)	Backward (ms)	Total (ms)
Pascal Titan X	5.1.05	48.09	99.23	147.32
GTX 1080 Ti	5.1.10	48.15	100.04	148.19
Pascal Titan X	5.0.05	55.75	134.98	190.73
GTX 1080	5.1.05	68.95	141.44	210.39
Maxwell Titan X	5.1.05	73.66	151.48	225.14
GTX 1080	5.0.05	79.79	202.02	281.81
Maxwell Titan X	5.0.05	93.47	229.34	322.81
Maxwell Titan X	4.0.07	139.01	279.21	418.22
Pascal Titan X	None	121.69	318.39	440.08
GTX 1080	None	176.36	453.22	629.57
Maxwell Titan X	None	215.92	491.21	707.13
CPU: Dual Xeon E5-2630 v3	None	3609.78	6239.45	9849.23

Modelo VGG19 usado na competição ILSVRC-2014

- ⦿ A GPU Pascal Titan X é 1.31x a 1.43x mais rápida que a GTX 1080
- ⦿ A pascal TITAN X com cuDNN é 2.2x a 3.0x mais rápida
- ⦿ A pascal TITAN X com cuDNN é 49x a 74x mais rápida que um dual Xeon E5-2630 v3 CPUs

FPGAs



Field Programmable Gate Array

- ⦿ *GPFPAs* são inadequadas e ineficientes

Field Programmable Gate Array

- ⊙ *GPFGAs* são inadequadas e ineficientes
- ⊙ Assim como *ASICs* são particularmente adequadas para a etapa de inferência

Field Programmable Gate Array

- ⊙ *GPFGAs* são inadequadas e ineficientes
- ⊙ Assim como ASICs são particularmente adequadas para a etapa de inferência
- ⊙ Consomem menos energia, por esse motivo é perfeita para ser empregada em DNNs na robótica e automação.

Field Programmable Gate Array

- ⊙ *GPFGAs* são inadequadas e ineficientes
- ⊙ Assim como ASICs são particularmente adequadas para a etapa de inferência
- ⊙ Consomem menos energia, por esse motivo é perfeita para ser empregada em DNNs na robótica e automação.
- ⊙ Algoritmos de DNNs e suas implementações estão constantemente evoluindo

Field Programmable Gate Array

- ⊙ *GPFGAs* são inadequadas e ineficientes
- ⊙ Assim como ASICs são particularmente adequadas para a etapa de inferência
- ⊙ Consomem menos energia, por esse motivo é perfeita para ser empregada em DNNs na robótica e automação.
- ⊙ Algoritmos de DNNs e suas implementações estão constantemente evoluindo

Solução:

Field Programmable Gate Array

- ⊙ *GPFPGAs* são inadequadas e ineficientes
- ⊙ Assim como ASICs são particularmente adequadas para a etapa de inferência
- ⊙ Consomem menos energia, por esse motivo é perfeita para ser empregada em DNNs na robótica e automação.
- ⊙ Algoritmos de DNNs e suas implementações estão constantemente evoluindo

Solução:

DNN-specific FPGAs

A Deephi é uma empresa recente que produz FPGAs para DNNs e fornecem:

- ⦿ Compressores de DNNs (permite às FPGAs trabalharem com mais dados)
- ⦿ Compiladores de Deep Learning, que compilam em minutos ao invés de horas/dias

Hardwares Dedicados - ASICs

Hardwares dedicados - ASICs

Alguns fatores vêm contribuindo para o desenvolvimento de hardwares dedicados:

- ⦿ **Computação intensiva** com requisitos específicos
- ⦿ **Aplicação extensiva** dos algoritmos de IA
- ⦿ Avanços na área de **IoT** (alto desempenho e baixa potência)

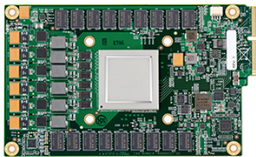
Algumas iniciativas:

- ⦿ Google TPU
- ⦿ Nervana Engine (Acquired by Intel for \$482M)
- ⦿ Graphcore IPU (Intelligence Processing Unit)

Tensor Processing Unit

Tensor Processing Unit (TPU) é um hardware desenvolvido pelo Google e otimizado para o **TensorFlow** Projetado para ser usado na fase de **inferência**

- ⦿ Google Search
- ⦿ Google Image Search
- ⦿ Google Photos
- ⦿ Google Cloud Vision API
- ⦿ Google Translate
- ⦿ Google DeepMind



TPU e Servidor Google

Tensor Processing Unit

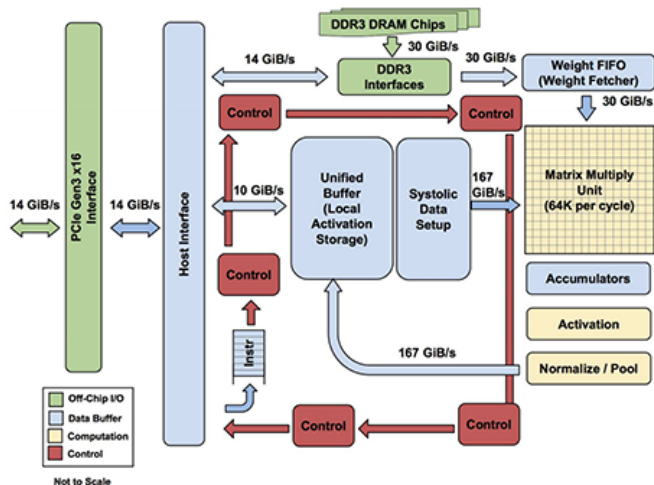


Diagrama de blocos da TPU

Conjunto de instruções

Modelo CISC otimizado para executar as instruções mais frequentes de redes neurais:

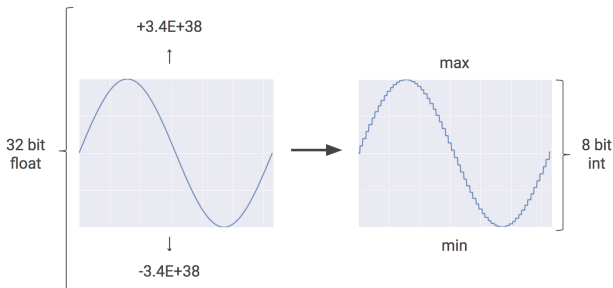
Tabela: Instruções da TPU

TPU Instruction	Function
Read_Host_Memory	Read data from memory
Read_Weights	Read weights from memory
MatrixMultiply/Convolve	Multiply or convolve with the data and weights,accumulate the results
Activate	Apply activation functions
Write_Host_Memory	Write result to memory

Quantização

O TensorFlow utiliza uma estratégia, chamada **quantização**, para reduzir a precisão dos dados.

- ⦿ Em geral, não compromete a acurácia das previsões
- ⦿ Reduz o esforço computacional
- ⦿ Reduz o consumo de memória e energia



Estratégia de quantização do TensorFlow

Comparação

Name	LOC	Layers					Nonlinear function	Weights	TPU Ops / Weight Byte	TPU Batch Size	% of Deployed TPUs in July 2016
		FC	Conv	Vector	Pool	Total					
MLP0	100	5				5	ReLU	20M	200	200	61%
MLP1	1000	4				4	ReLU	5M	168	168	
LSTM0	1000	24		34		58	sigmoid, tanh	52M	64	64	29%
LSTM1	1500	37		19		56	sigmoid, tanh	34M	96	96	
CNN0	1000		16			16	ReLU	8M	2888	8	5%
CNN1	1000	4	72		13	89	ReLU	100M	1750	32	

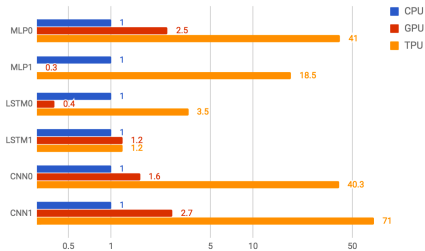
Experimentos realizados

Model	Die										Benchmarked Servers				
	mm ²	nm	MHz	TDP	Measured		TOPS/s		GB/s	On-Chip Memory	Dies	DRAM Size	TDP	Measured	
					Idle	Busy	8b	FP						Idle	Busy
Haswell E5-2699 v3	662	22	2300	145W	41W	145W	2.6	1.3	51	51 MiB	2	256 GiB	504W	159W	455W
NVIDIA K80 (2 dies/card)	561	28	560	150W	25W	98W	--	2.8	160	8 MiB	8	256 GiB (host) + 12 GiB x 8	1838W	357W	991W
TPU	NA*	28	700	75W	28W	40W	92	--	34	28 MiB	4	256 GiB (host) + 8 GiB x 4	861W	290W	384W

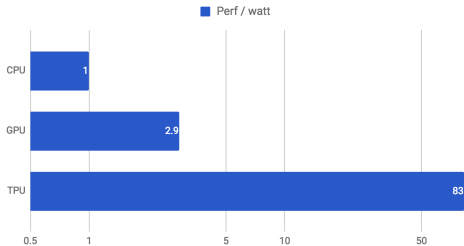
Consumo de memória

Comparação

Resultados obtidos:



Perfomance (pred/s norm.)



Eficiência energética

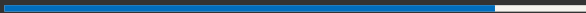
Intel Nervana

- ⦿ Startup fundada em 2014 e adquirida pela Intel em 2016
- ⦿ Nervana Engine tem previsão de ser lançado em 2017
- ⦿ 8 ASICs conectados em uma configuração torus
- ⦿ Arquitetura 16 bits
- ⦿ 32 Gb de memória
- ⦿ Velocidade de acesso à memória de até 8 terabits/s
- ⦿ Tecnologia de 28nm, com promessa de redução para 16nm

GraphCore IPU

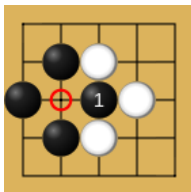
- ⦿ IPU = Intelligence Processing Unit
- ⦿ Entre os investidores estão a Samsung e Dell
- ⦿ Também com previsão de lançamento para 2017
- ⦿ Promessa de ganho de desempenho da ordem de 10x a 100x em comparação aos melhores sistemas atuais
- ⦿ Poplar é o framework de desenvolvimento com interface em C++ e Python
- ⦿ Promessa de uma linha de IPU's de baixa potência para sistemas embarcados

Go



Go é um jogo simples com apenas 2 regras:

- ⦿ Toda *pedra* precisa ter ao menos uma casa de liberdade
- ⦿ A configuração do tabuleiro **nunca** deverá retornar a uma configuração anterior



O número de configurações possíveis para o tabuleiro é da ordem de $10^{171} \gg 10^{82}$ = número de átomos do universo

Desenvolvido pela Google DeepMind, **AlphaGo** foi o primeiro programa a derrotar alguns dos campeões mundiais de Go

Em outubro de 2015 uma versão distribuída do AlphaGo (1,202 CPUs and 176 GPUs) derrotou **Fan Hui**, o campeão europeu de nível 2-dan, por 5-0

Em 15 de março de 2016 outra versão distribuída do AlphaGo (1,920 CPUs and 280 GPUs) derrotou **Lee Sedol**, um campeão mundial de nível 9-dan, por 4-1

De 29 de dezembro de 2016 a 5 de janeiro de 2017 uma versão alternativa do AlphaGo jogou online contra jogadores profissionais sob as alcunhas *Magist*, *Magister* e *Master* acumulando o recorde de 60 vitórias e 0 derrotas

Utiliza uma Árvore de Busca de Monte Carlo guiada por uma *rede de valor* e duas *redes de política* (uma rápida e imprecisa e outra devagar mas precisa), todas CDNN

Rede de Política: Determina os movimentos mais promissores

Rede de Valor: Avalia o estado atual do jogo

As redes de política foram treinadas com 30 milhões de jogadas de humanos e então foram treinadas jogando entre si através de aprendizado reforçado

Enquanto isso a rede de valor foi treinada com 30 milhões de configurações de tabuleiro

Configuration and performance

Configuration ↕	Search threads ↕	No. of CPU ↕	No. of GPU ↕	Elo rating ↕
Single ^[9] p. 10-11	40	48	1	2,181
Single	40	48	2	2,738
Single	40	48	4	2,850
Single	40	48	8	2,890
Distributed	12	428	64	2,937
Distributed	24	764	112	3,079
Distributed	40	1,202	176	3,140
Distributed	64	1,920	280	3,168

Configuration and strength^[52]

Versions ↕	Hardware ↕	Elo rating ↕	Matches ↕
AlphaGo Fan	Distributed	nearly 3,000	5:0 against Fan Hui
AlphaGo Lee	50 TPUs, distributed	about 3,750	4:1 against Lee Sedol
AlphaGo Master	single machine with TPU v2	about 4,750	60:0 against professional players; Future of Go Summit

Conclusão

Conclusão

- ⦿ O campo de aplicação têm expandido consideravelmente
- ⦿ Aplicações que exigem altíssima capacidade computacional
- ⦿ Aplicações em sistemas embarcados
- ⦿ Atualmente a melhor solução para treinamento é com a utilização de GPU's
- ⦿ Resultados surpreendentes com a utilização de hardware dedicado, tanto em performance quanto em eficiência energética
- ⦿ O único ASIC em operação (TPU) ainda não é comercializado
- ⦿ Grandes players se movimentando para lançar hardwares em um futuro próximo

Referências



Jeff Heaton

Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural Networks

Heaton Research, Inc

ISBN = 1505714346

Year = 2015



Alan V. Oppenheim

Discrete - Time Signal Processing

Prentice Hall Press, 2009



European Broadcasting Union

Specification of the Broadcast Wave Format (BWF)

2011



Norman P. Jouppi *et. al.*

In-Datacenter Performance Analysis of a Tensor Processing Unit

International Symposium on Computer Architecture

Junho, 2017