NVIDIA

GPU ACCELERATION FOR FINANCIAL SERVICES

John Ashley, Ph.D.

General Manager, Financial Services and Technology

17 May 2021

DISCLAIMER & COPYRIGHT

Yep.

ALL opinions here are mine, not those of NVIDIA or others.

ALL errors or omissions here are also mine, not those of NVIDIA or others.

Your mileage may vary. Batteries not included.

Si
- Moore's Law
- Nanometers
- Dennard Scaling

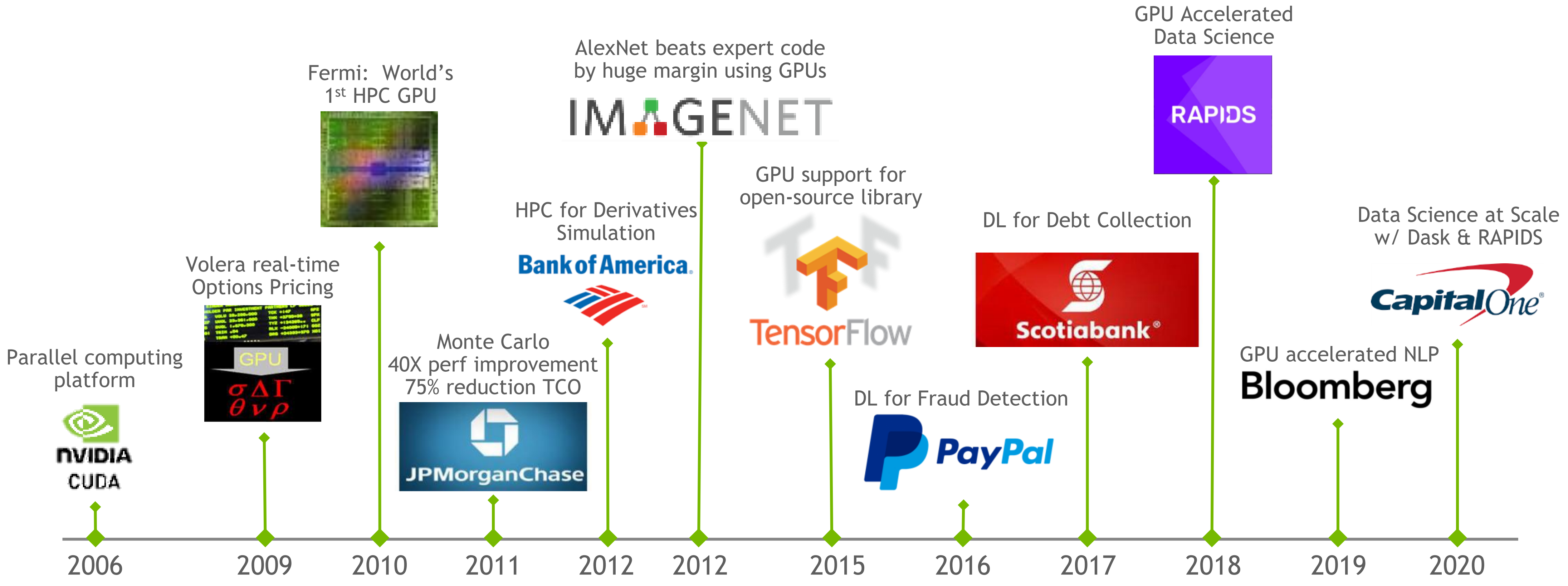
Foundations
- Chip Budgets
- CPU, GPU, FPGA, ASIC
- Discrete Math
- Parallel Math

GPU accelerated FSI
- Why do we care?
- HPC & Hybrid AI/HPC
- Deep Learning
- Building Blocks

BUT FIRST, A MESSAGE FROM OUR SPONSOR

NVIDIA GPU COMPUTING

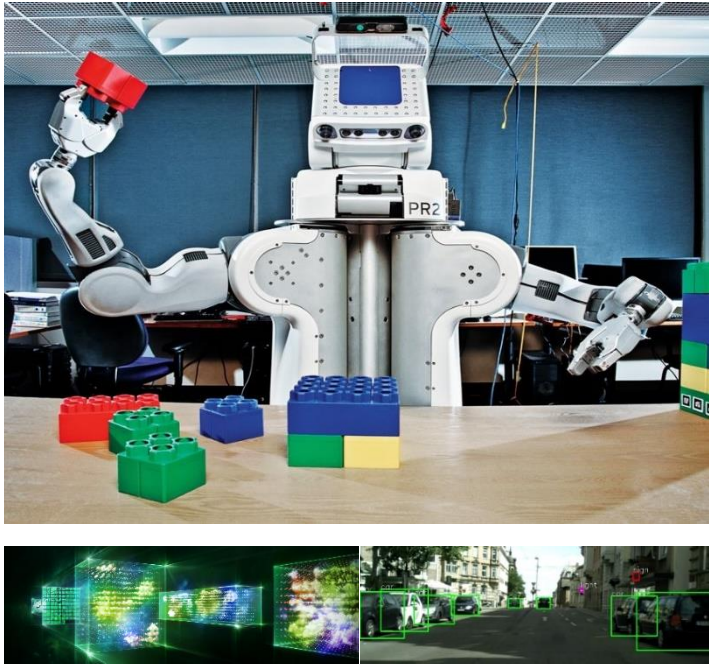
13+ Years in Financial Services



NVIDIA
“THE AI COMPUTING COMPANY”

GPU Computing

Computer Graphics



Artificial Intelligence

SILICON

2013

Presented By: John Ashley
Senior Solutions Architect, Global Finance, NVIDIA

Challenges of Accelerated Computing in Finance

Moore's Law

- Moore's Law has switched from clock rate to core count
- Parallelism is the future

"The vast majority of programmers today don't grok concurrency, just as the vast majority of programmers 15 years ago didn't yet grok objects"

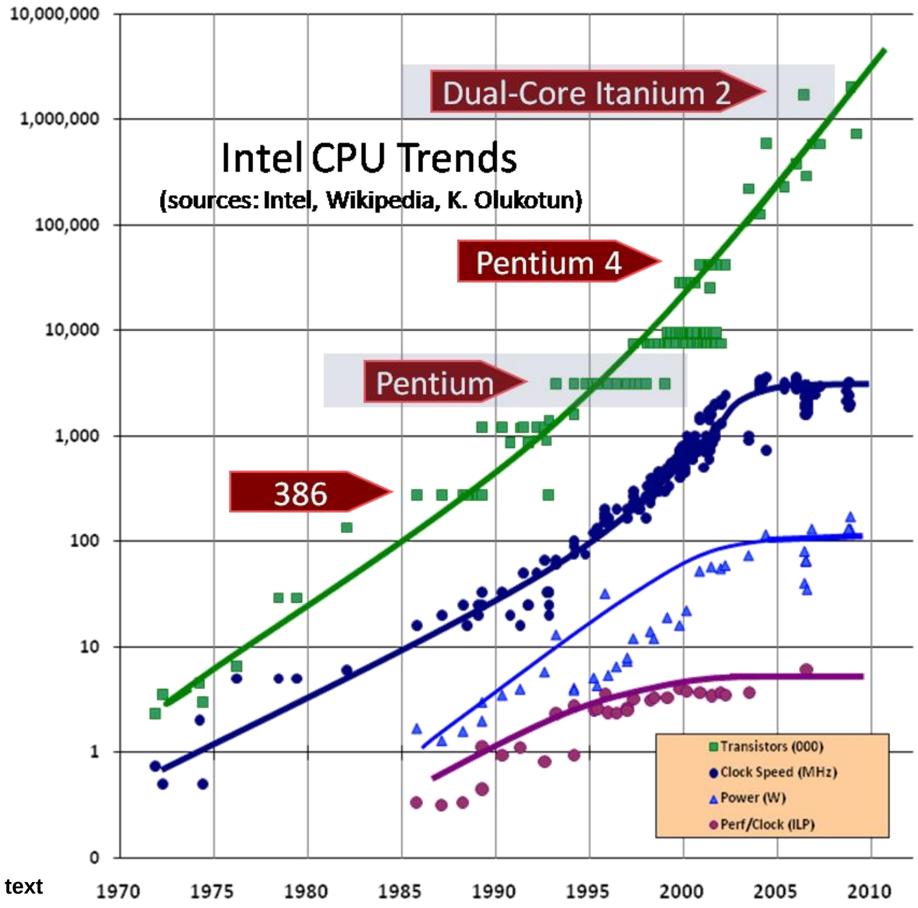
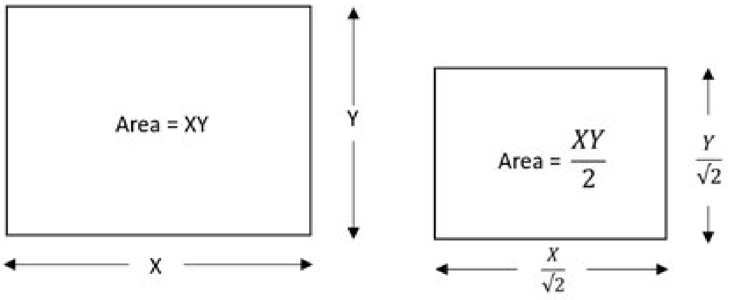


Chart & Quote from "The Free Lunch Is Over A Fundamental Turn Toward Concurrency in Software, Herb Sutter, [*Dr. Dobb's Journal*](), *30(3), March 2005*

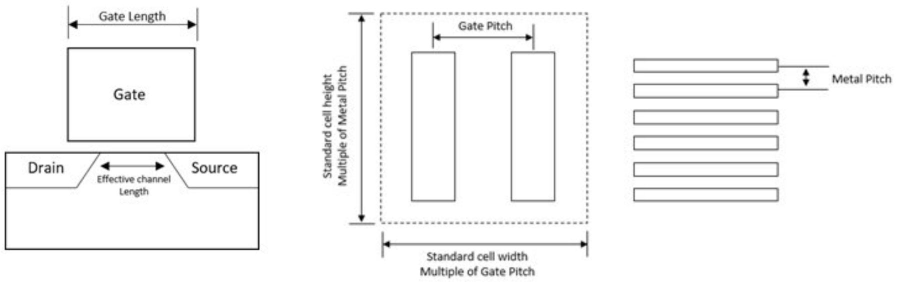
Figure 1: Intel CPU Introductions (graph updated August 2009; article text original from December 2004)

NANOMETERS (1E-9 METERS)

Size isn’t everything … but materials science and lithography are topics for another day.



Class			Wave-length λ
Ionizing radiation	γ	Gamma rays	1 pm
			10 pm
	HX	Hard X-rays	100 pm
	SX	Soft X-rays	1 nm
			10 nm
	EUV	Extreme ultraviolet	100 nm
	NUV	Near ultraviolet, visible	1 μm
	NIR	Near infrared	10 μm
	MIR	Mid infrared	100 μm
	FIR	Far infrared	



Number of Semiconductor Manufacturers with a Cutting Edge Logic Fab

SilTerra										
X-FAB										
Dongbu HiTek										
ADI	ADI									
Atmel	Atmel									
Rohm	Rohm									
Sanyo	Sanyo									
Mitsubishi	Mitsubishi									
ON	ON									
Hitachi	Hitachi									
Cypress	Cypress	Cypress								
Sony	Sony	Sony								
Infineon	Infineon	Infineon								
Sharp	Sharp	Sharp								
Freescale	Freescale	Freescale								
Renesas (NEC)	Renesas	Renesas	Renesas	Renesas						
Toshiba	Toshiba	Toshiba	Toshiba	Toshiba						
Fujitsu	Fujitsu	Fujitsu	Fujitsu	Fujitsu						
TI	TI	TI	TI	TI						
Panasonic	Panasonic	Panasonic	Panasonic	Panasonic	Panasonic					
STMicroelectronics	STM	STM	STM	STM	STM					
HLMC	HLMC		HLMC	HLMC	HLMC					
UMC	UMC	UMC	UMC	UMC	UMC		UMC			
IBM	IBM	IBM	IBM	IBM	IBM	IBM				
SMIC	SMIC	SMIC	SMIC	SMIC	SMIC		SMIC	SMIC		
AMD	AMD	AMD	GlobalFoundries	GF	GF	GF	GF			
Samsung	Samsung	Samsung	Samsung	Samsung	Samsung	Samsung	Samsung	Samsung	Samsung	Samsung
TSMC	TSMC	TSMC	TSMC	TSMC	TSMC	TSMC	TSMC	TSMC	TSMC	TSMC
Intel	Intel	Intel	Intel	Intel	Intel	Intel	Intel	Intel	Intel	Intel
180 nm	130 nm	90 nm	65 nm	45 nm/40 nm	32 nm/28 nm	22 nm/20 nm	16 nm/14 nm	10 nm	7 nm	5 nm

https://www.design-reuse.com/articles/43316/a-brief-history-of-process-node-evolution.html

https://en.wikichip.org/wiki/technology_node

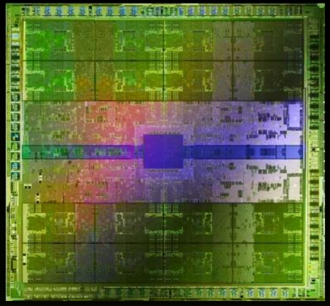
https://en.wikipedia.org/wiki/Electromagnetic_spectrum

2014

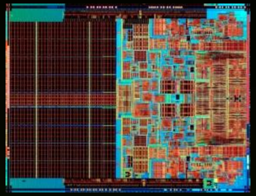
GPUs and the Future of Accelerated Computing
Napier 400

John Ashley
Solutions Architect, Financial Services
jashley@nvidia.com

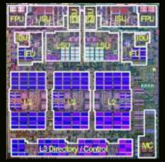
Moore’s Law is Only Part of the Story



2010: 3B transistors



2007: 580M transistors



2004: 275M transistors

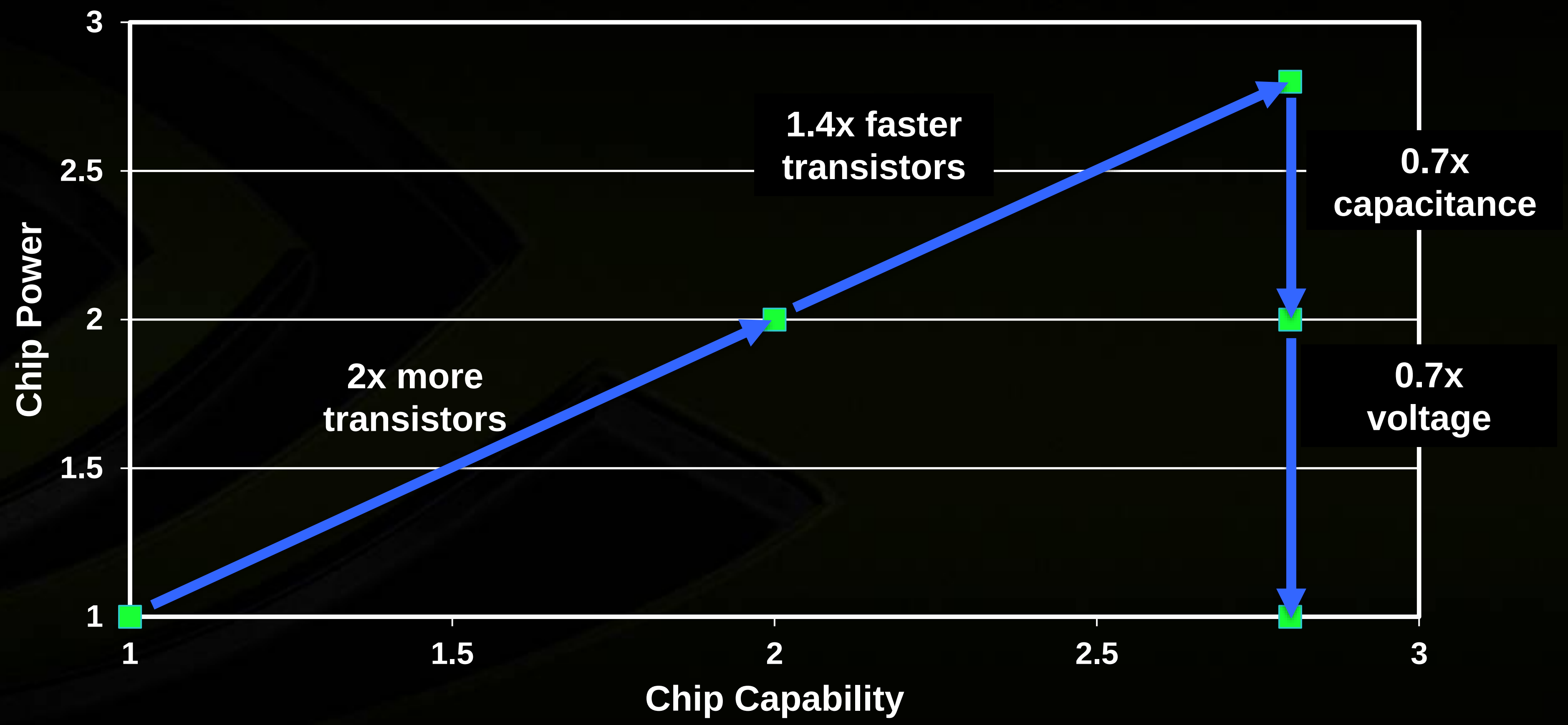
2001: 42M transistors

1997: 7.5M transistors

1993: 3M transistors

Classic Dennard Scaling

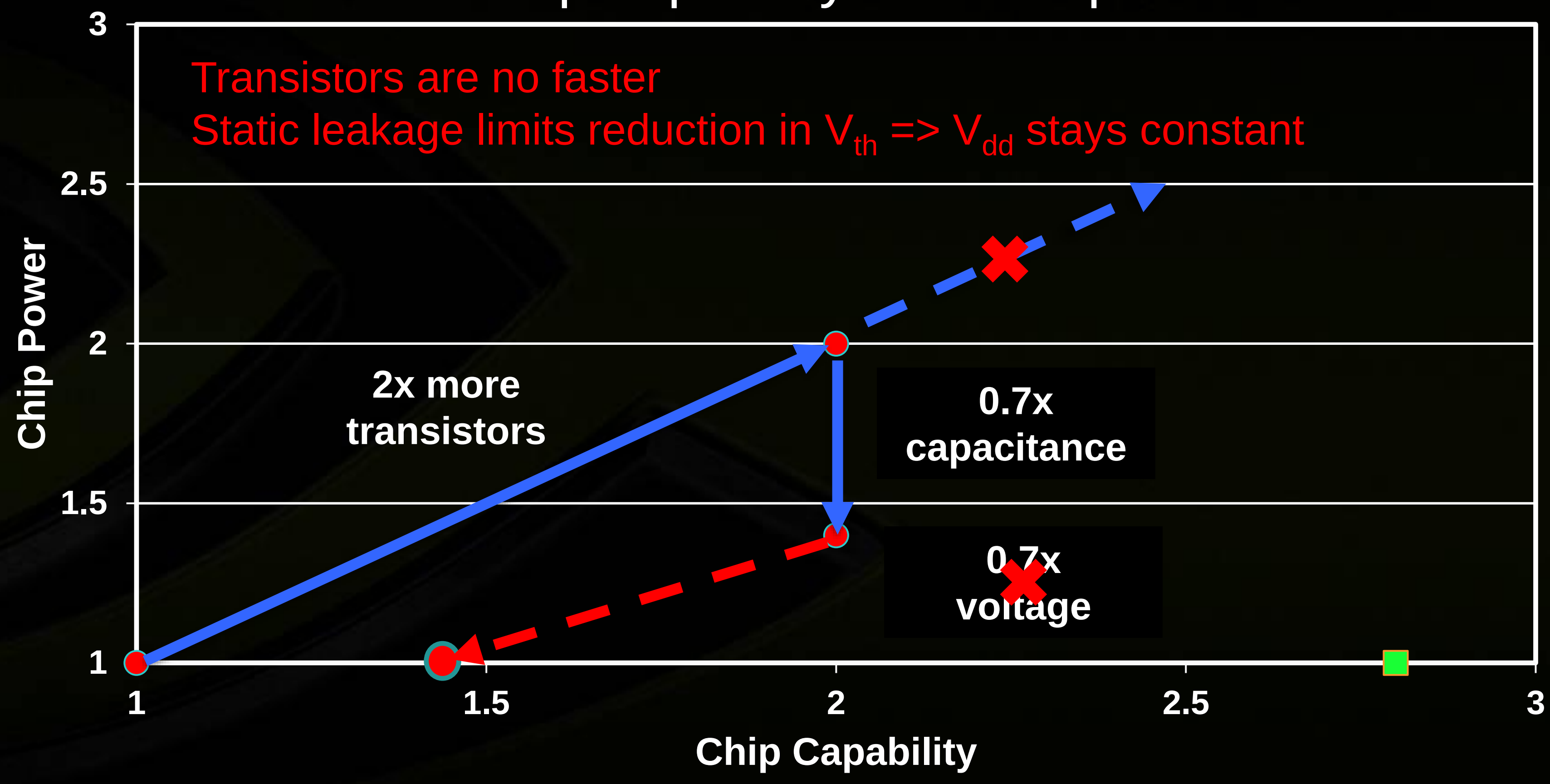
2.8x chip capability in same power



Post Dennard Scaling

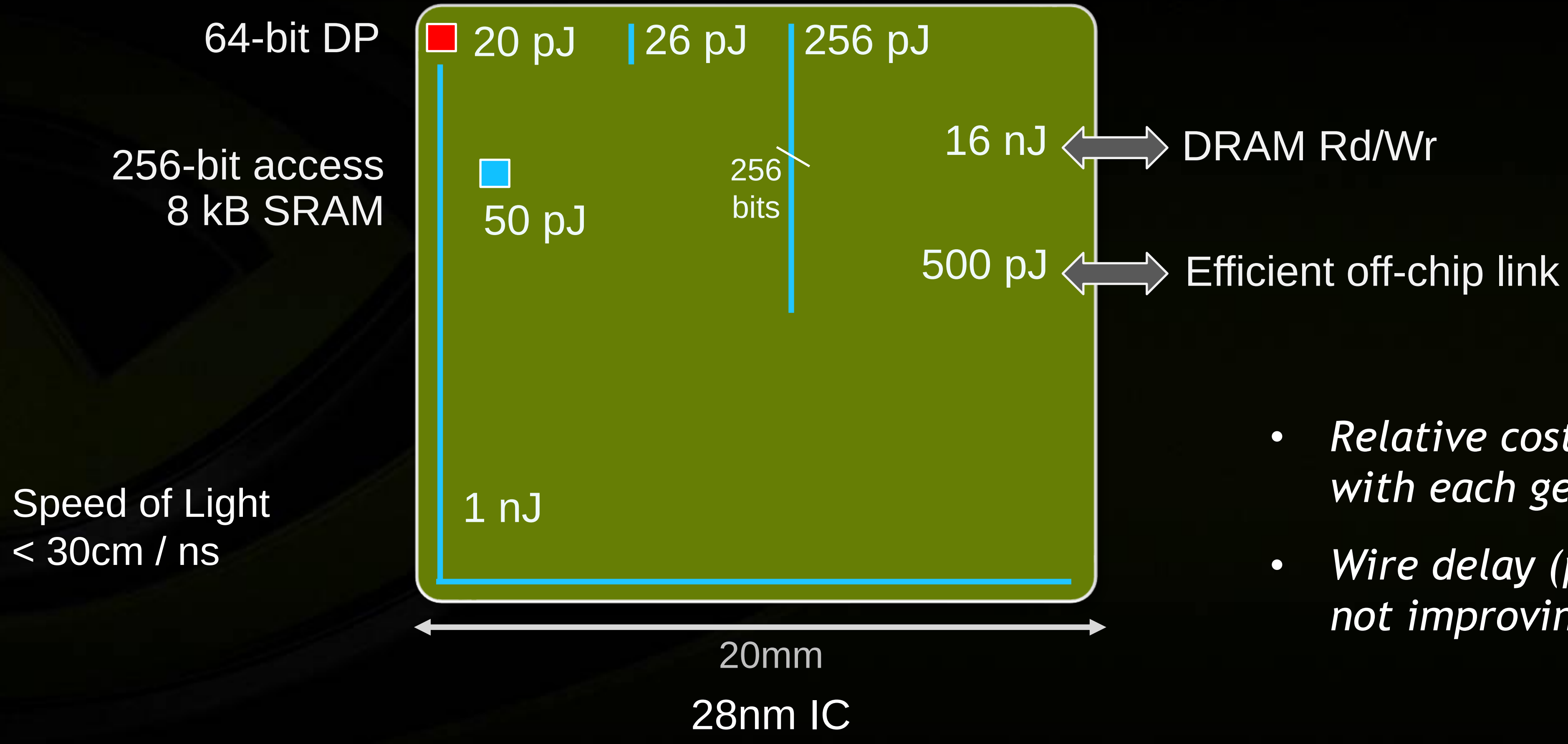
2x chip capability at 1.4x power

1.4x chip capability at same power



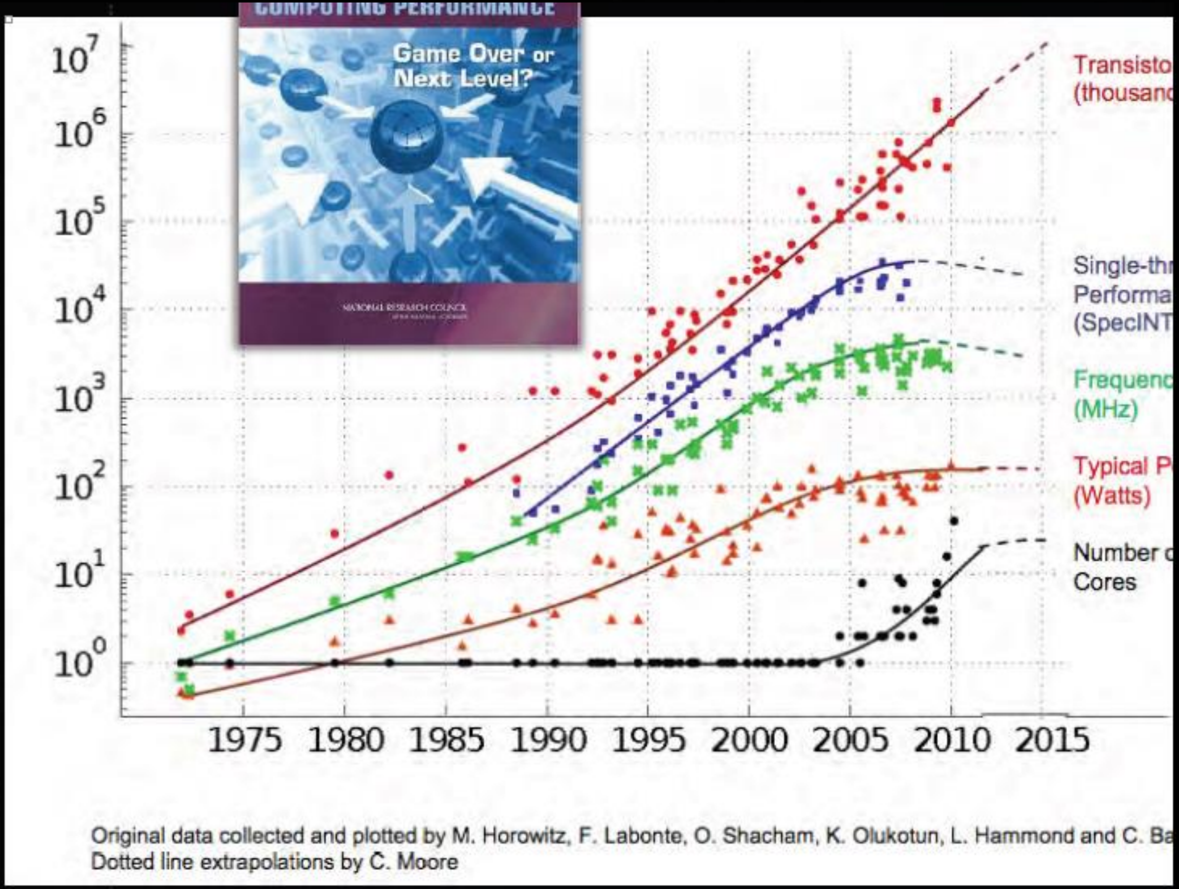
It’s not just about speed…it’s energy too.

Fetching operands costs more than computing on them



- *Relative cost grows with each generation*
- *Wire delay (ps/mm) not improving*

Moore's Law isn't what it used to be…



Moore's law is alive and well, but…

- Instruction-level parallelism (ILP) was mined out in 2001
- Voltage scaling (Dennard scaling) ended in 2005
- Most power is spent on communication

What does this mean to you?

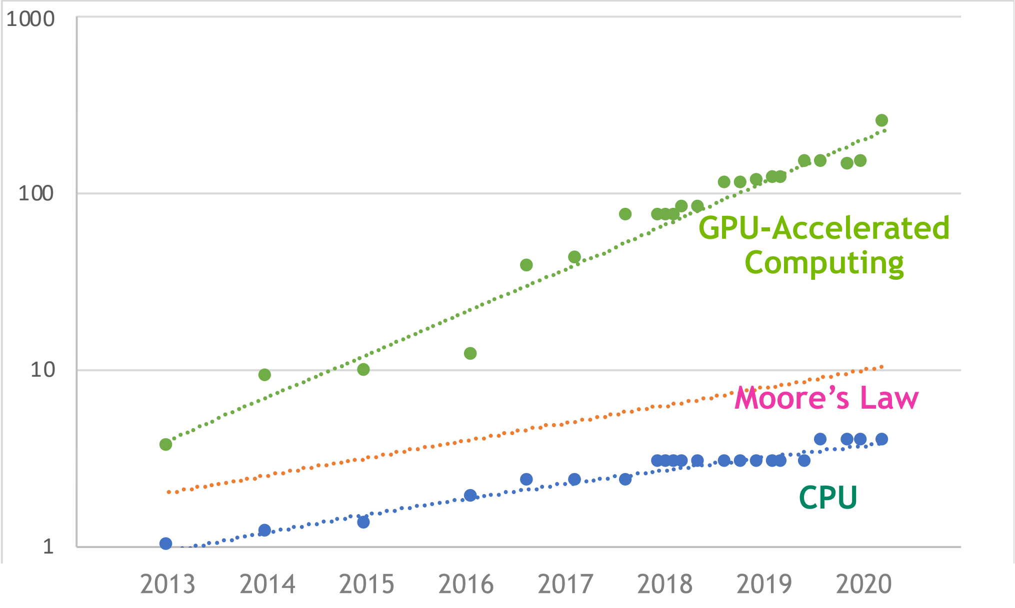
BEYOND MOORE'S LAW

Progress Of Stack In 7 Years

2013

- cuBLAS: 5.0
- cuFFT: 5.0
- cuRAND: 5.0
- cuSPARSE: 5.0
- NPP: 5.0
- Thrust: 1.5.3
- CUDA: 5.0
- Resource Mgr: r304
- Base OS: CentOS 6.2

Accelerated Server With Fermi



Measured performance of Amber, CHROMA, GTC, LAMMPS, MILC, NAMD, Quantum Espresso, SPECFEM3D

2020

- cuBLAS: 11.0
- cuFFT: 11.0
- cuRAND: 11.0
- cuSOLVER: 11.0
- cuSPARSE: 11.0
- NPP: 1`1.0
- Thrust: 1.9.0
- CUDA: 11.0
- Resource Mgr: r384
- Base OS: Ubuntu 16.04

Accelerated Server with Ampere

FOUNDATIONS

CHIP BUDGETS – AREA AND POWER

How do we spend it?

Manufacturing process improvements -> bigger chips

Lithography improvements -> more transistors / chip

Power Consumption is going up

Chip designers can add more parallelism, deeper pipelines, more special function units…

…if you can optimize for a class of workloads you can still get good scaling for those workloads.

Economies of Scale -> Cutting edge nodes in demand, expensive; need volume to control costs.

Relative volume of relevant chips: CPU > GPU > FPGA > ASIC

CPU, GPU usually current nodes, FPGA & ASIC trail

WHO'S IN THE ZOO?

These all endured in the market because they have a sweet spot

Factor	CPU	GPU	FPGA	ASIC
Top Level	General purpose, does everything; most common platform	Parallel graphics, HPC, and AI accelerator	Configurable collection of logic and functional units	Fully custom hardware
Latency	Context switches expensive Deep pipes, speculative execution, etc. to hide latency	Context switches cheap latency tolerated via context switch	Placement defines literal length of code path	Fabrication defines literal length of code path
Throughput	Multi-core (<100); multiple vector math lanes	Multi-core (>1000); TensorCores	Placement defines # of processing flows	Fabrication defines # of processing flows
Economics	Massive scale Many developers	Excellent scale – Devs = gaming+HPC+AI	Large pockets Devs = aero, mil, mfg, HFT	Each is custom Devs = subset of FPGA + auto
Competitive?	ARM vs x86 (Intel/AMD)	NVIDIA vs AMD vs Intel	Intel Altera vs AMD Xilinx	Many at older process nodes

WHO'S IN THE ZOO?

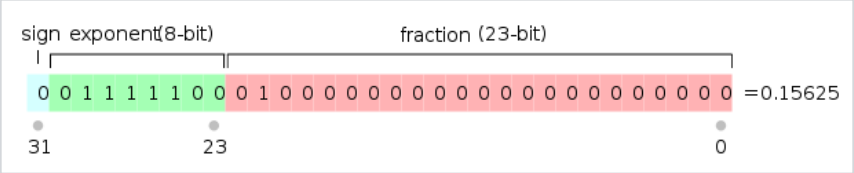
These all endured in the market because they have a sweet spot

Factor	CPU	GPU	FPGA	ASIC
Top Level	General purpose, does everything; most common platform	Parallel graphics, HPC, and AI accelerator	Configurable collection of logic and functional units	Fully custom hardware
Example: Branching	Branches would cause instruction cache stalls and so we get speculative execution to prevent context shifts. And security bugs.	Groups of threads execute in lockstep, so has to execute each TAKEN branch.	Every possible branch consumes some of the configurable resource, potentially limiting parallelism (makes code wider, so less pipes fit on card).	Every possible branch consumes some of the chip area, potentially limiting parallelism (makes code larger, so less pipelines fit on chip).
When to use	General purpose. Use this unless you need much better performance.	Throughput optimized, massively parallel. Use for AI, analytics from 10s of microseconds and up.	Latency optimized. Throughput varies. Use when latency is critical, code will be fairly stable, and there is significant time to optimize the design.	As FPGA, but the code needs to be extremely stable as it's permanent. VERY high upfront costs, unit costs can drop rapidly. Can be VERY power efficient and can be heavily optimized.

DISCRETE MATH FORMATS

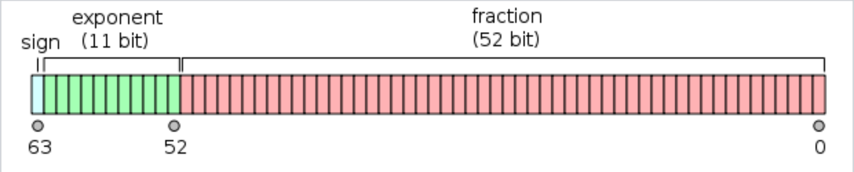
Double, single, etc…

FP numbers Sign*Precision*Base$^{(\text{Exponent-Offset})}$
Sign: positive or negative
Exponent/Range: location of decimal point
Fraction/Precision: digits



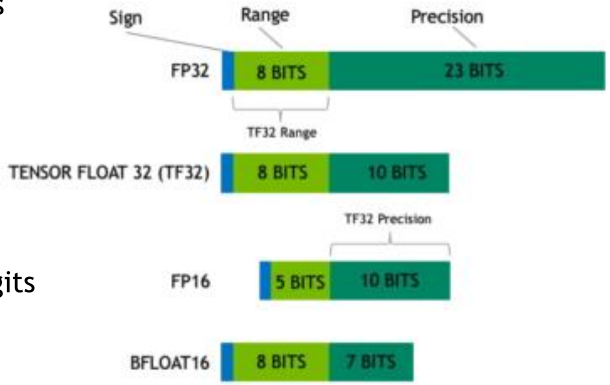
The number 0.15625 represented as a single-precision IEEE 754-1985 floating-point number. See text for explanation.

23 Bit ~ 6-7 decimal digits



The three fields in a 64bit IEEE 754 float

52 Bit ~ 15-16 decimal digits



TF32 strikes a balance that delivers performance with range and accuracy.

https://en.wikipedia.org/wiki/IEEE_754-1985

https://blogs.nvidia.com/blog/2020/05/14/tensorfloat-32-precision-format/

Exponent Bias = 127 / 1023

DISCRETE & PARALLEL MATH

Highlights

Pure mathematics: A+(B+C) = (A+B)+C.

Discrete mathematics: A+(B+C) ~= (A+B)+C

Order and relative magnitude of operands matters for discrete parallel math on computers.

Example: assume 3 digits retained at each stage, and a floating decimal. Sum (1.04, 10.1, 60.0, 22.0, 0.01) = 93.15

Sequential = (1.04+10.1,60.0, 22.0, 0.01) ->

(11.1 + 60.0, 22.0, 0.01) -> (71.1+22.0, 0.01) -> 93.1+0.01 = **93.1**

Sequential, sorted = (0.01+1.04, 10.1, 22, 60) ->

(1.05+10.1, 22.0, 60.0) -> (11.2 +22.0, 60.0) ->(33.2+60.0) = **93.2**

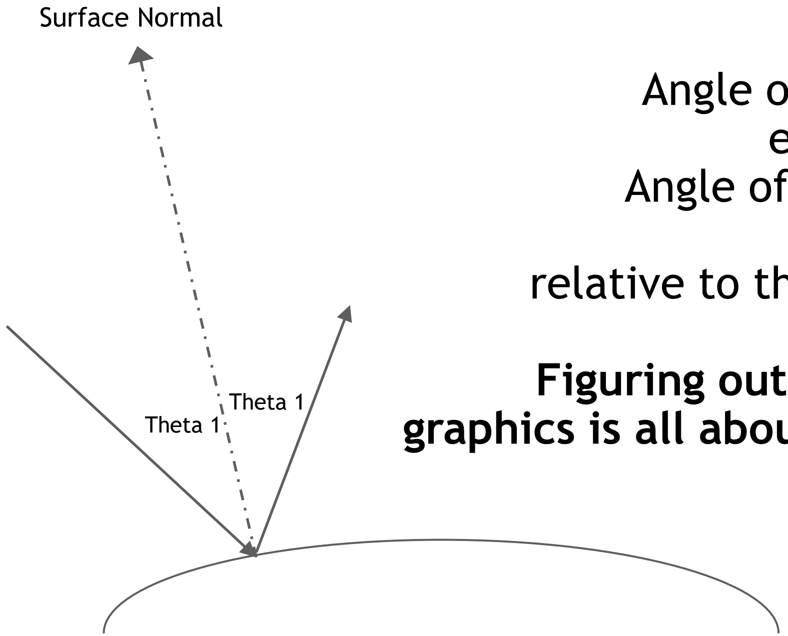
Pairwise and other parallel techniques can preserve even more significant figures over large operand sets.

For more info see https://docs.nvidia.com/cuda/floating-point/index.html

GPU ACCELERATED FSI

WHY ARE GPU’S GOOD AT ALL THIS STUFF?

Hint – nobody ever bought a single pixel screen!



Angle of Reflection
equals
Angle of Incidence …

relative to the surface normal.

Figuring out what you see in graphics is all about floating point math!

Famous
Single Pixel Screen

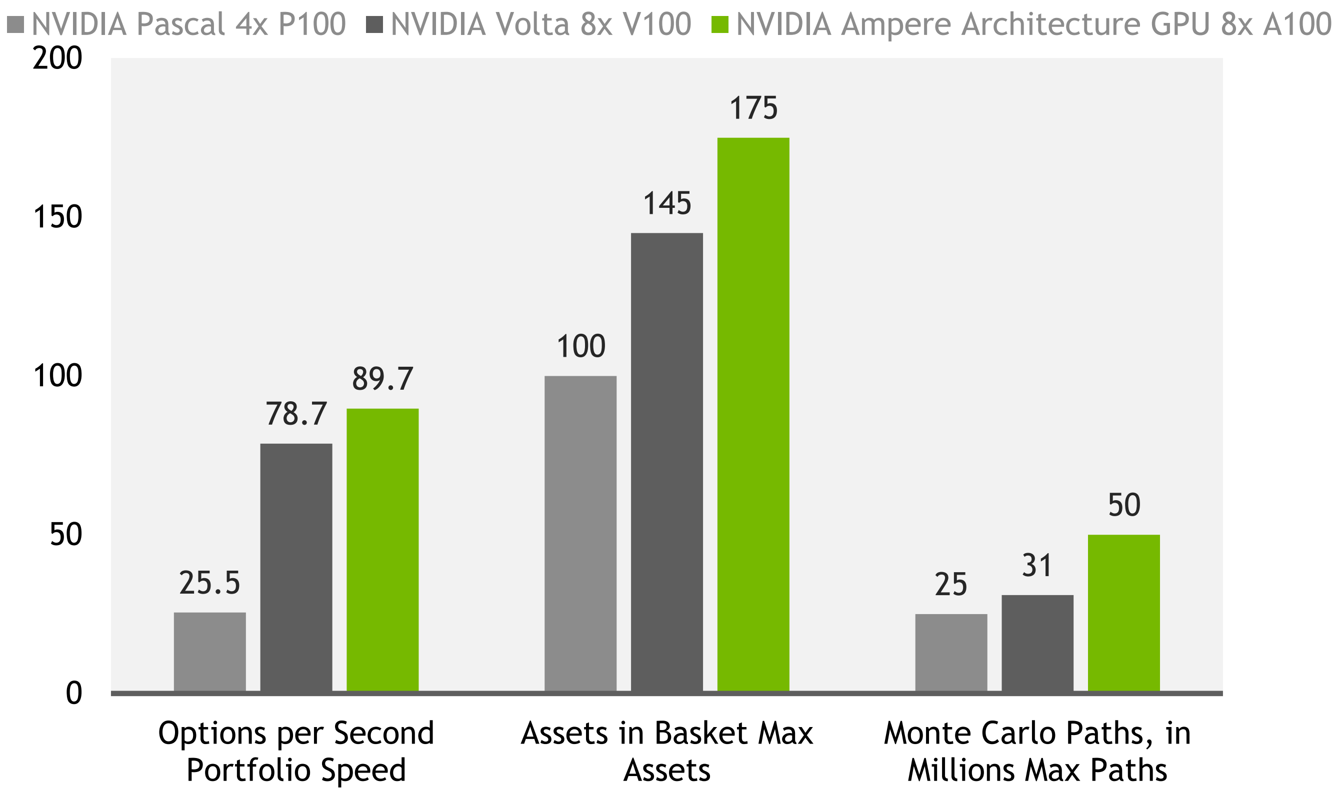
Real screens have many^2 pixels

- data parallel
- high bandwidth!

WHY DO WE CARE?

STAC A2™ BENCHMARK
MARKET RISK (PRICE & GREEKS), MONTE CARLO

- STAC A2 Benchmark
  - Developed by banks
  - Macro and micro, performance and accuracy
  - Pricing and Greeks for American exercise basket option, correlated
  - Heston dynamics, Longstaff Schwartz Monte Carlo
- Independently Audited Results
- NVIDIA DGX A100 set a new bar for these critical calculations, with impressive results in several key STAC-A2 categories
- Visit http://www.stacresearch.com/a2 for more details of the STAC Benchmark
- For more information on improvements in scalability and throughput, read [this infographic]()



- Heston dynamics – basically, mean reverting stochastic volatility
- Longstaff Schwartz Monte Carlo – LONG but quick diversion if we want to go there…
- See also https://developer.nvidia.com/blog/american-option-pricing-monte-carlo-simulation/
- https://people.maths.ox.ac.uk/gilesm/mc/module_6/american.pdf

Pricing models with early payoff

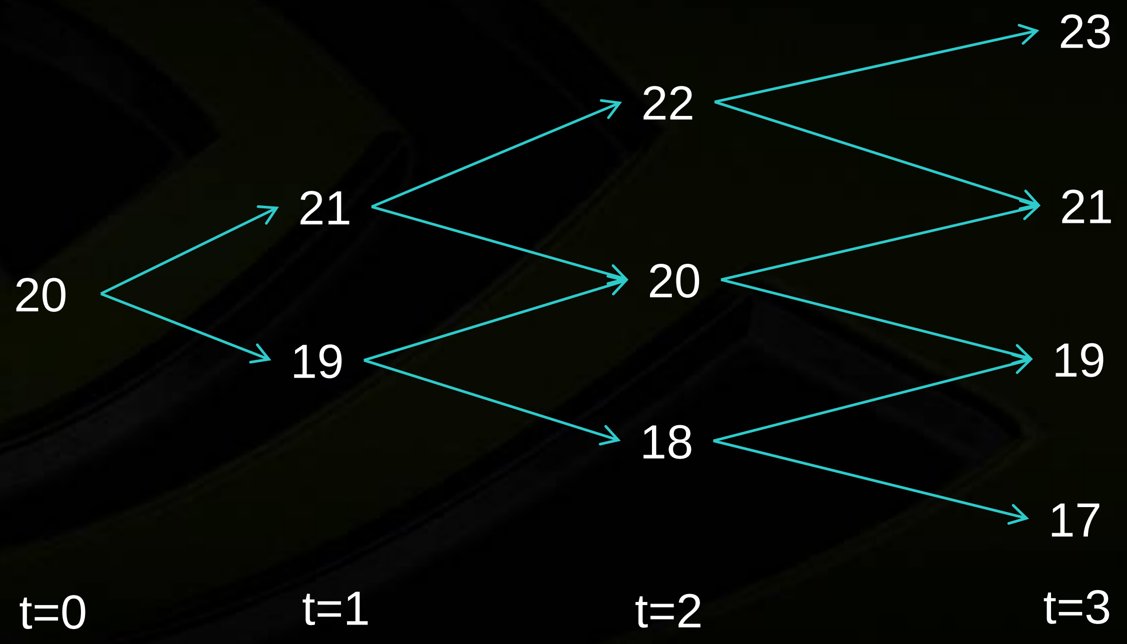
2012

- The value of the payoff function at any time t is the greater of the value of selling the instrument now or holding it for redemption later.
- We can work forwards in time to model the prices and the intrinsic values
- We can work backwards in time to evaluation the expected value of holding for later vs taking the money now.

Vanilla Bermudan Option Example

Underlying price at t=0 is 20
S_u dt = +0.5, S_d dt = -0.5 , P_u = 0.5, p_d = 0.5 , dt = 1/12, r = 0.02

Option is an ATM Put with expiry T = 1/3 and exercise at the end of every month
i.e. right to sell the stock at $20 at the end of any month in the next 3 months.



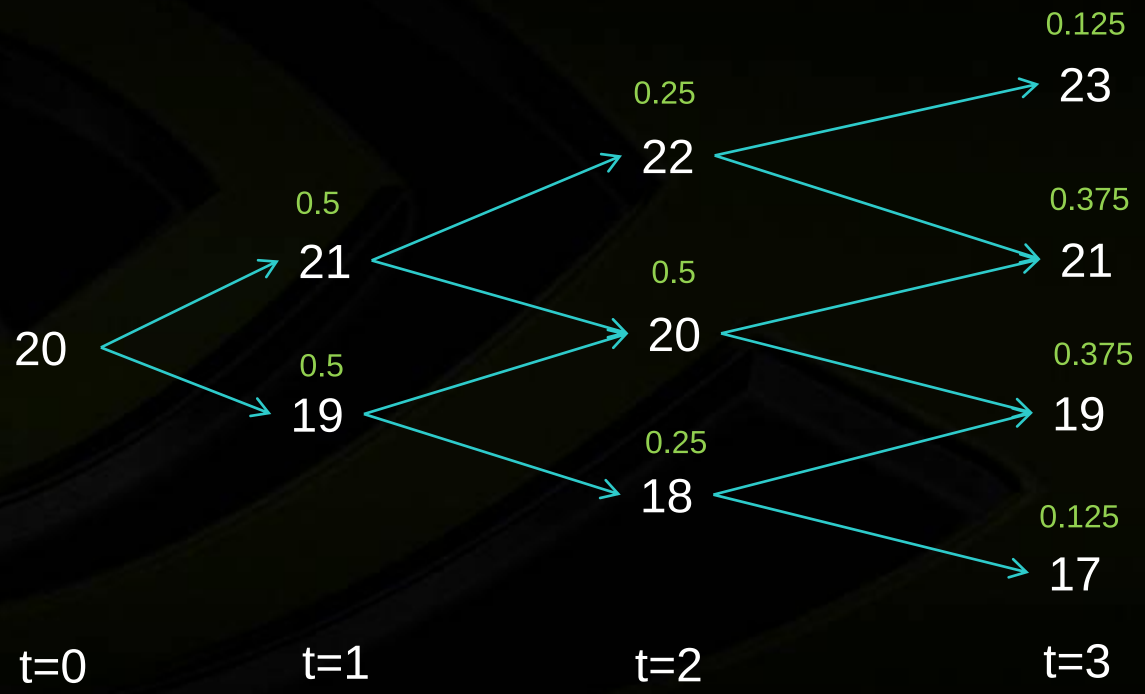
Underlying asset price   probability   intrinsic value   continuation value

Vanilla Bermudan Option Example

Underlying price at t=0 is 20

S_u dt = +0.5, S_d dt = -0.5 , P_u = 0.5, p_d = 0.5 , dt = 1/12, r = 0.02

Option is an ATM Put with expiry T = 1/3 and exercise at the end of every month i.e. right to sell the stock at $20 at the end of any month in the next 3 months.

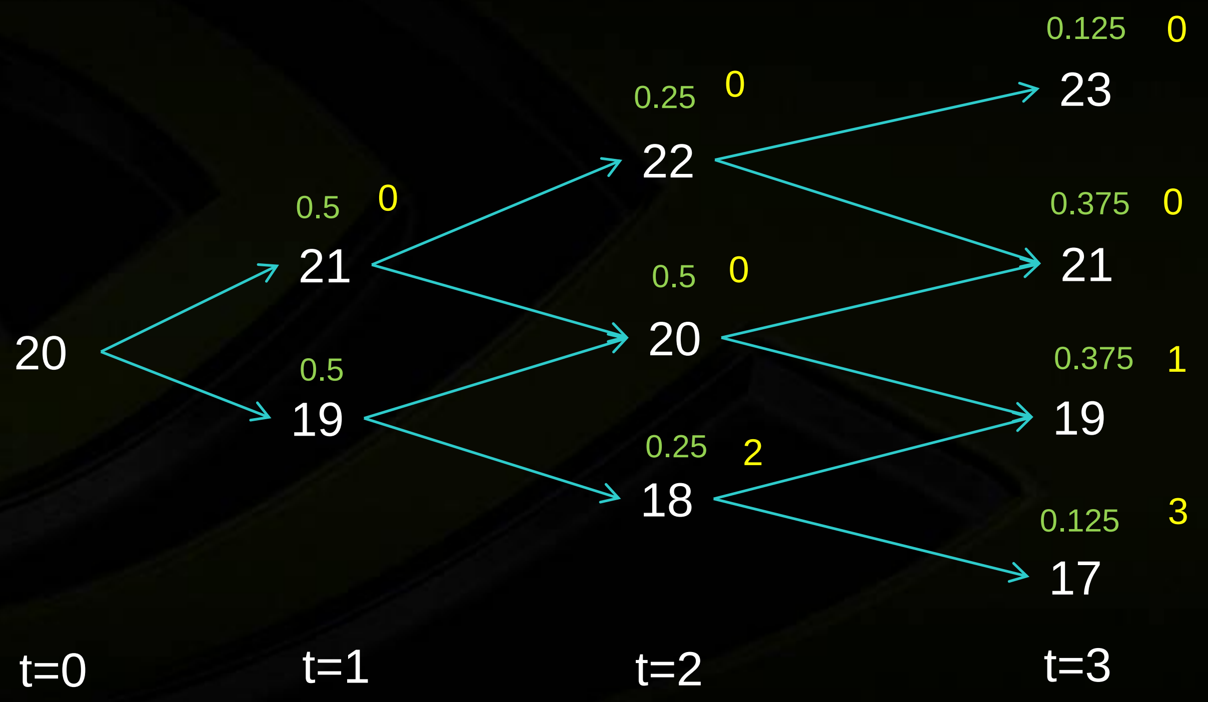


Underlying asset price probability intrinsic value continuation value

Vanilla Bermudan Option Example

Underlying price at t=0 is 20
S_u dt = +1, S_d dt = -1 , P_u = 0.5, p_d = 0.5 , dt = 1/12, r = 0.02

Option is an ATM Put with expiry T = 1/3 and exercise at the end of every month
i.e. right to sell the stock at $20 at the end of any month in the next 3 months.



Discount factors

T=1	0.9983
T=2	0.9967
T=3	0.9950

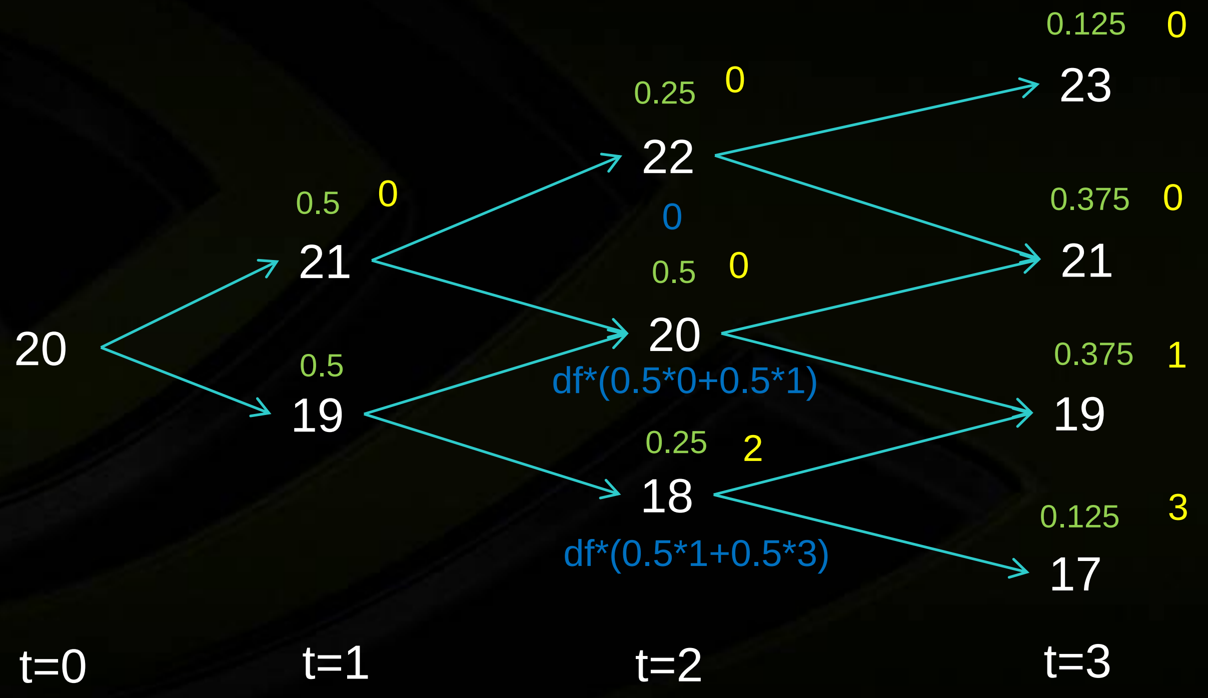
Underlying asset price probability intrinsic value continuation value

Vanilla Bermudan Option Example

Underlying price at t=0 is 20

$S\_u\, dt = +1$, $S\_d\, dt = -1$, $P\_u = 0.5$, $p\_d = 0.5$, $dt = 1/12$, $r = 0.02$

Option is an ATM Put with expiry T = 1/3 and exercise at the end of every month
i.e. right to sell the stock at $20 at the end of any month in the next 3 months.



Discount factors

T=1	0.9983
T=2	0.9967
T=3	0.9950

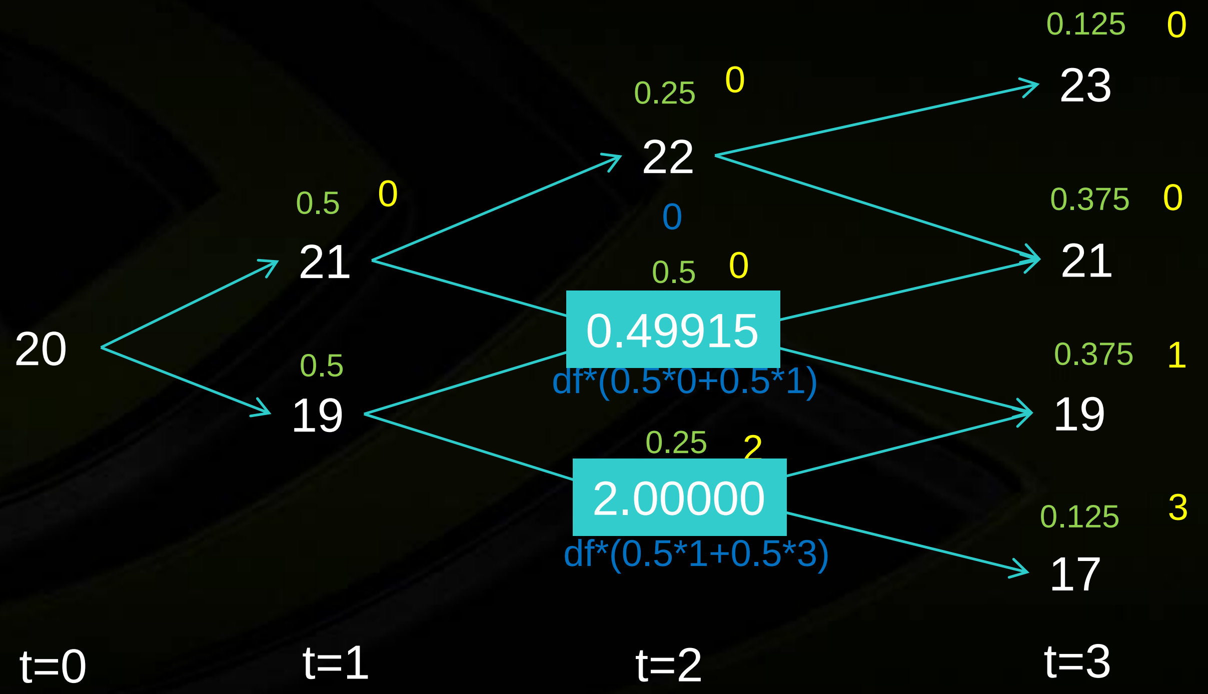
time 2, calculated expected
Continuation values

Underlying asset price probability intrinsic value continuation value

Vanilla Bermudan Option Example

Underlying price at t=0 is 20
S_u dt = +1, S_d dt = -1 , P_u = 0.5, p_d = 0.5 , dt = 1/12, r = 0.02

Option is an ATM Put with expiry T = 1/3 and exercise at the end of every month
i.e. right to sell the stock at $20 at the end of any month in the next 3 months.



Discount factors	
T=1	0.9983
T=2	0.9967
T=3	0.9950

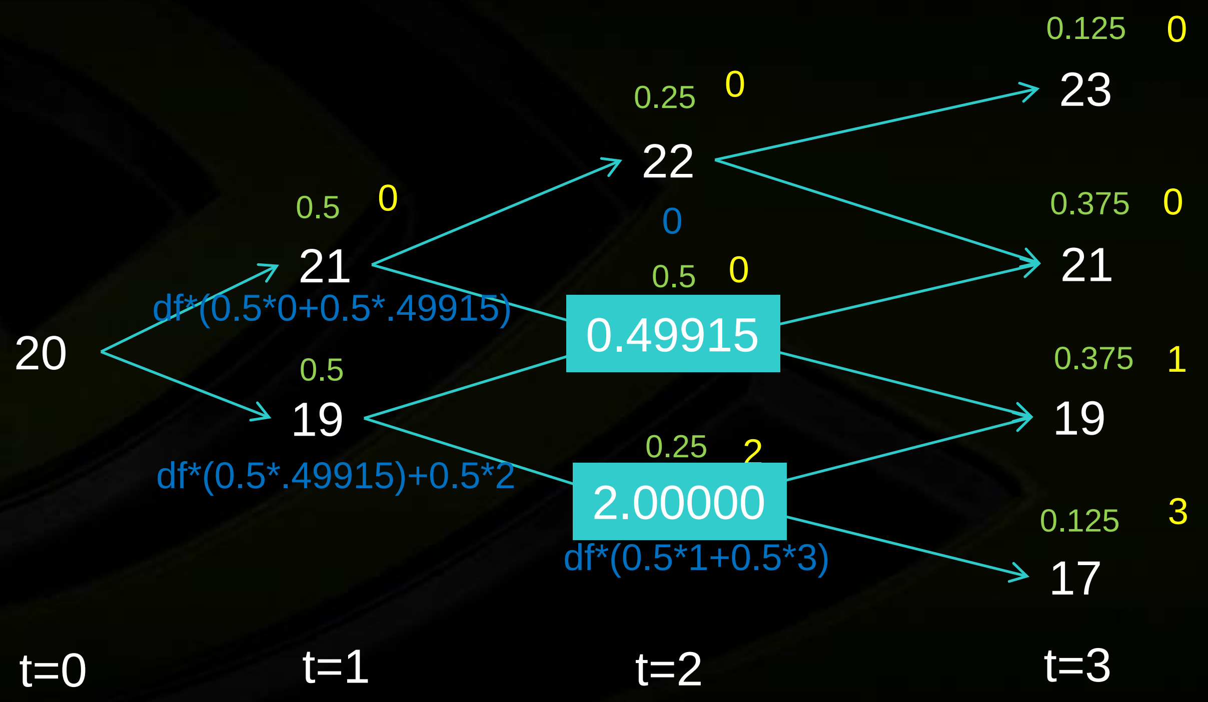
time 2, calculated expected
Continuation values

Underlying asset price probability intrinsic value continuation value

Vanilla Bermudan Option Example

Underlying price at t=0 is 20
S_u dt = +1, S_d dt = -1 , P_u = 0.5, p_d = 0.5 , dt = 1/12, r = 0.02

Option is an ATM Put with expiry T = 1/3 and exercise at the end of every month
i.e. right to sell the stock at $20 at the end of any month in the next 3 months.



Discount factors

T=1	0.9983
T=2	0.9967
T=3	0.9950

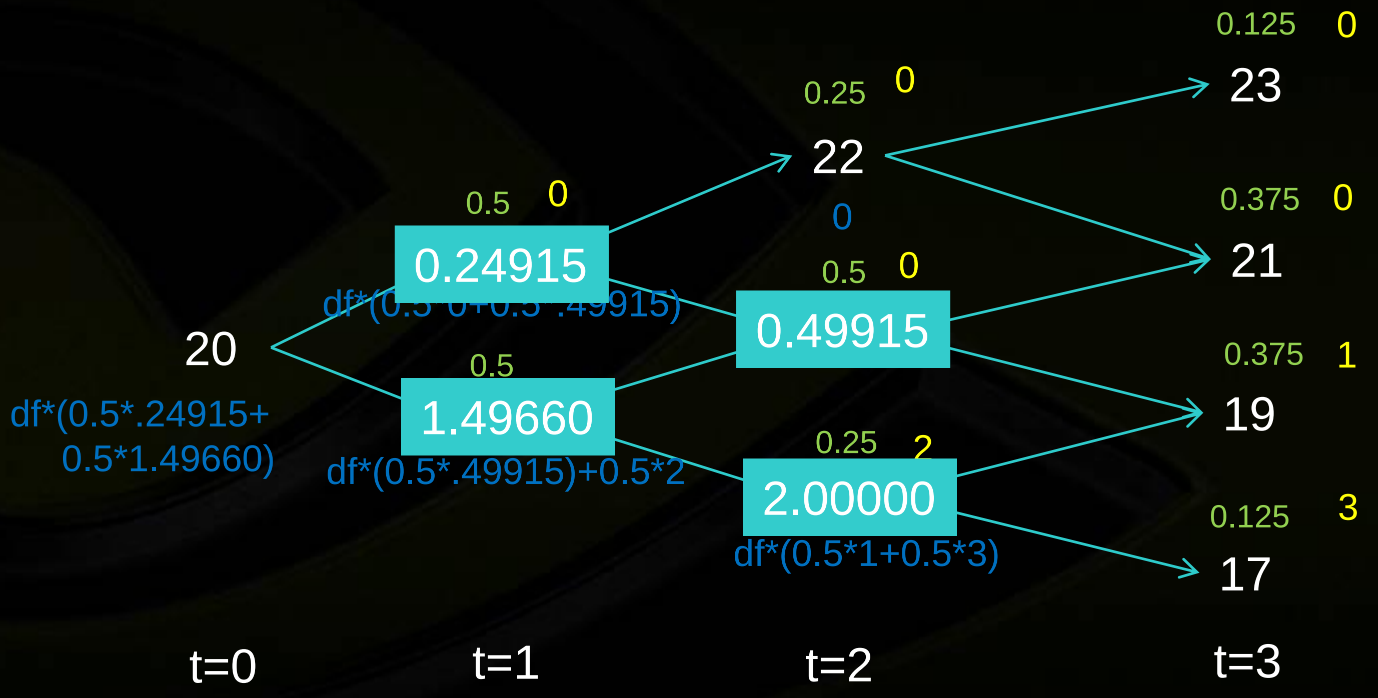
time 2, calculated expected Continuation values

Underlying asset price probability intrinsic value continuation value

Vanilla Bermudan Option Example

Underlying price at t=0 is 20
S_u dt = +1, S_d dt = -1 , P_u = 0.5, p_d = 0.5 , dt = 1/12, r = 0.02

Option is an ATM Put with expiry T = 1/3 and exercise at the end of every month
i.e. right to sell the stock at $20 at the end of any month in the next 3 months.



Discount factors
Delta t=1 0.9983

time 2, calculated expected
Continuation values

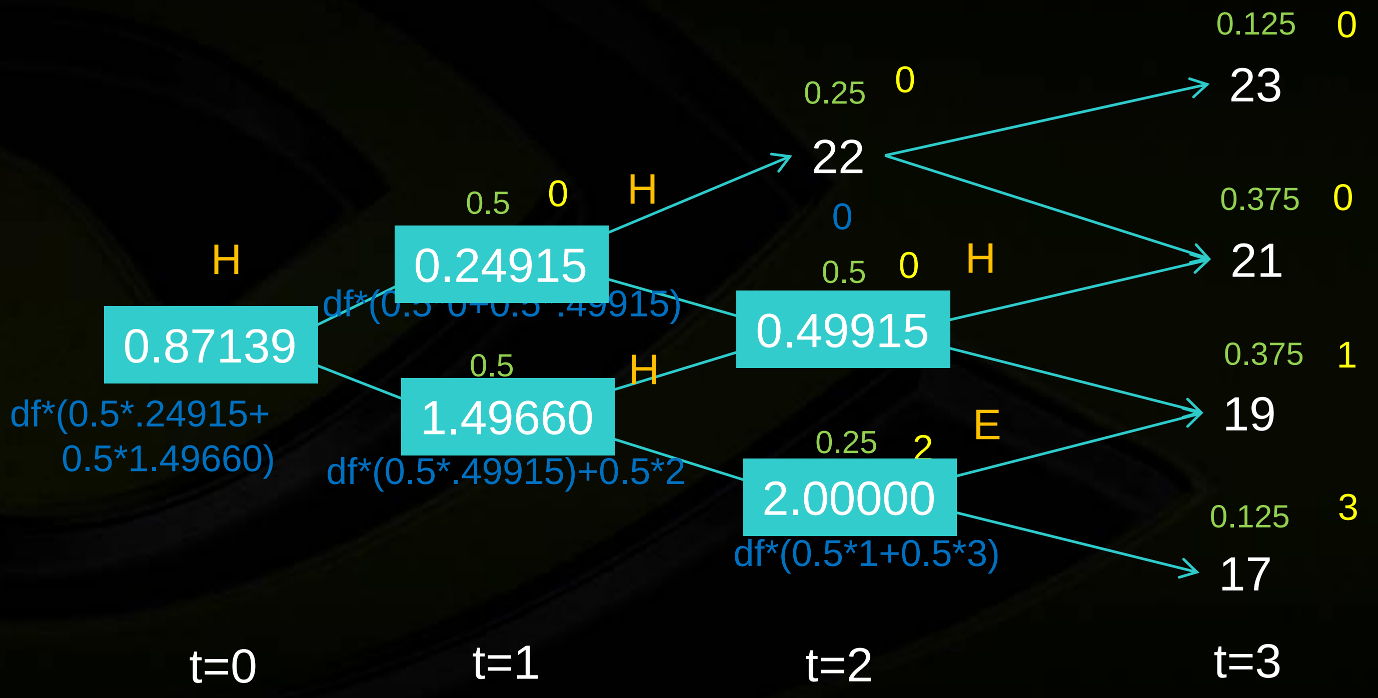
Underlying asset price   probability   intrinsic value   continuation value

Vanilla Bermudan Option Example

Underlying price at t=0 is 20

S_u dt = +1, S_d dt = -1 , P_u = 0.5, p_d = 0.5 , dt = 1/12, r = 0.02

Option is an ATM Put with expiry T = 1/3 and exercise at the end of every month
i.e. right to sell the stock at $20 at the end of any month in the next 3 months.



Discount factors
Delta t=1 0.9983

time 2, calculated expected
Continuation values

Underlying asset price   probability   intrinsic value   continuation value

Does this make sense?

- If this was a European, we would have
  - df*df*df*(0.125*0+0.375*0+0.375*1+0.125*3)
  - = 0.994908665087 * (0+0+0.375+0.375)
  - = 0.7461825
- A Bermudan or American is ALWAYS worth at least as much as a European so this is one reasonable check.
- So, under these model assumptions for this underlying, you should be willing to buy this option for $0.87139 per option
- I should be willing to sell it to you for the same amount
- Reality check – Internally, I need margin etc so I won't sell to you at the theoretical price; or I will charge you transaction fees.

Longstaff Schwartz Monte Carlo Outline

- Generate Paths
- Price the intrinsic value forward along the paths to the final timestep
- Contstruct the “exercise time vector” and set to the final timestep for all paths in the money
- For the next to last time step backwards to the first time step
  - Find all the paths that are currently “in the money”
  - Regress underlying value to value at next step
  - Use regression parameters to form expected value at next time step from underlying value
  - Hold or exercise and update exercise time accordingly
- Calculate Statistics

Longstaff Schwartz Example in Matlab

```
for step = NSteps-1:-1:1
    InMoney = find(SPaths(:,step) < K);
    XData = SPaths(InMoney,step);
    RegrMat = zeros(length(XData),NBasis);
    for k = 1:NBasis
        RegrMat(:,k) = feval(fhandles{k},XData);
    end
    YData = CashFlows (InMoney).*discountVet(ExerciseTime(InMoney)-step);
    alpha = RegrMat \ YData;
    IntrinsicValue = K - XData;
    ContinuationValue = RegrMat*alpha;
    Index = find( IntrinsicValue > ContinuationValue );
    ExercisePaths = InMoney(Index);
    CashFlows(ExercisePaths) = IntrinsicValue(Index);
    ExerciseTime(ExercisePaths) = step;
end % for step
```

ACCELERATING DIGITAL TRANSFORMATION IN FSI

AI/ML optimizes performance and outcomes

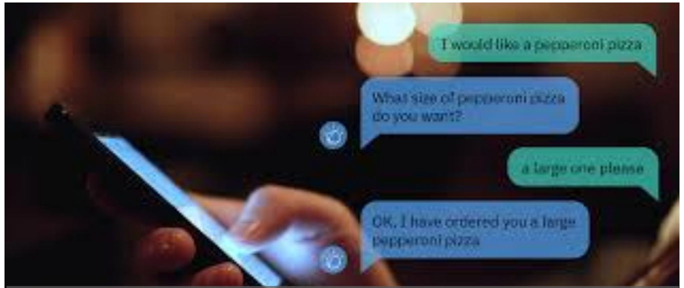

Underwriting


FRAUD DETECTION


VIRTUALIZATION (WFH)

DIGITAL PAYMENTS


RECOMMENDATIONS



CUSTOMER SERVICE

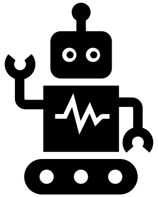

ALGORITHMIC TRADING

AI FOR TRADING

Selected Use Cases

Augmented Intelligence for Discretionary Traders

NLP
- Text Prioritization
- Text Summarization
- Named Entity Recognition & Knowledge Graphs



Artificial Intelligence for Algo Traders

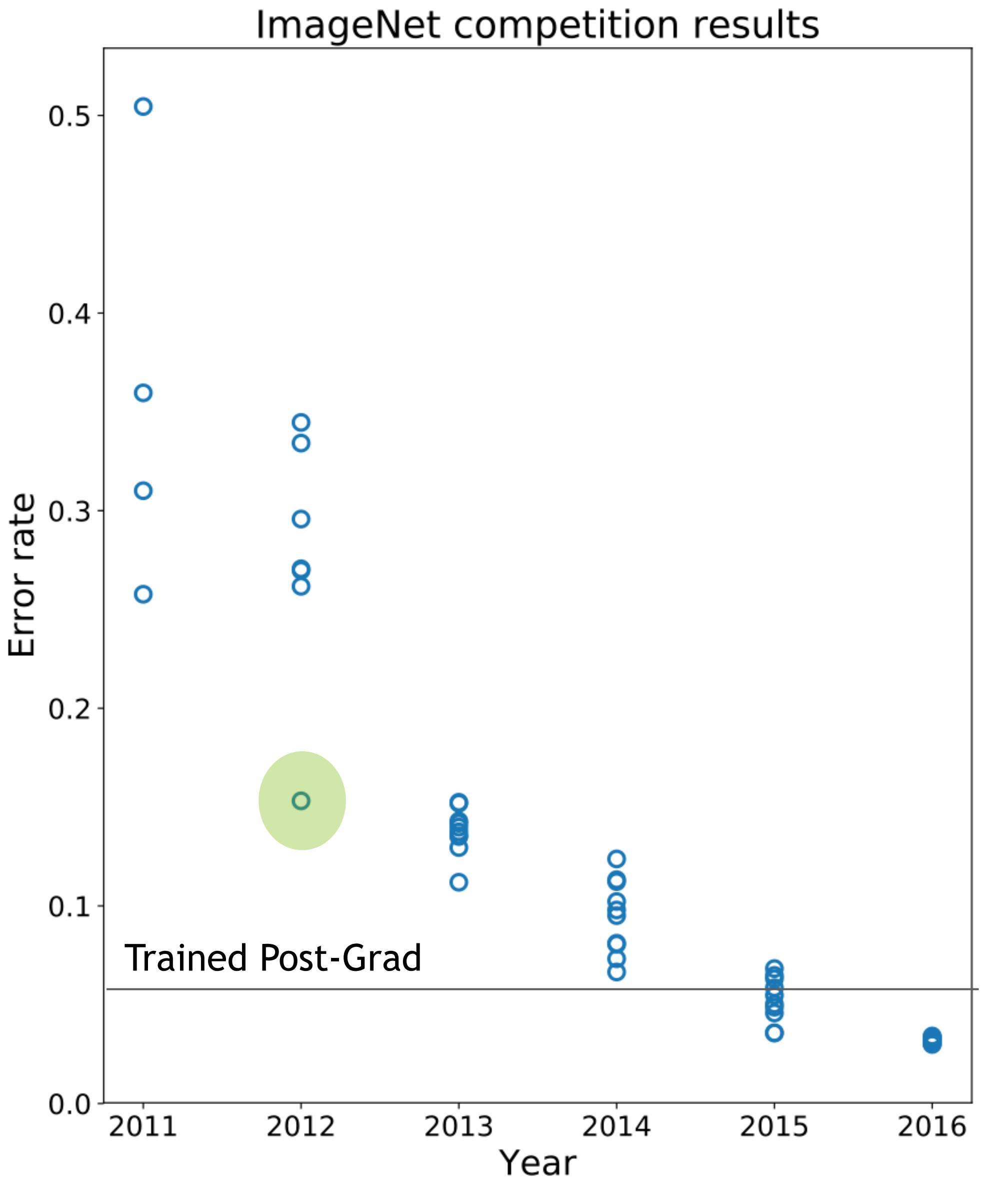
Algo Development
- Time Series via RNN / Temporal CNN
- Synthetic Data / VAE & GAN (backtesting)

Sentiment Analysis – News, Social Media, Regulatory Filings

“alt data”
Optimal execution (Reinforcement Learning)
Deep Learning for Pricing and Risk

DEEP LEARNING

IN THE BEGINNING



Highlights

2012: AlexNet, 8 layers

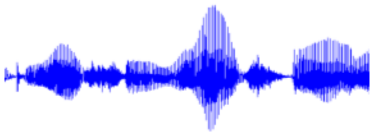
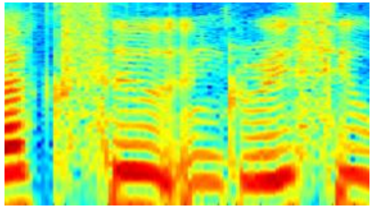
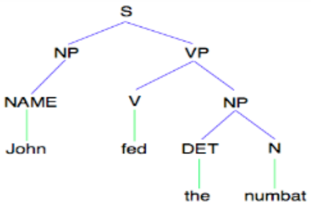
2014: Inception v1, 22 layers
2014: VGG, 19 layers

2015: ResNet, 152 layers.

Source: Wikipedia, Imagenet, arXiv

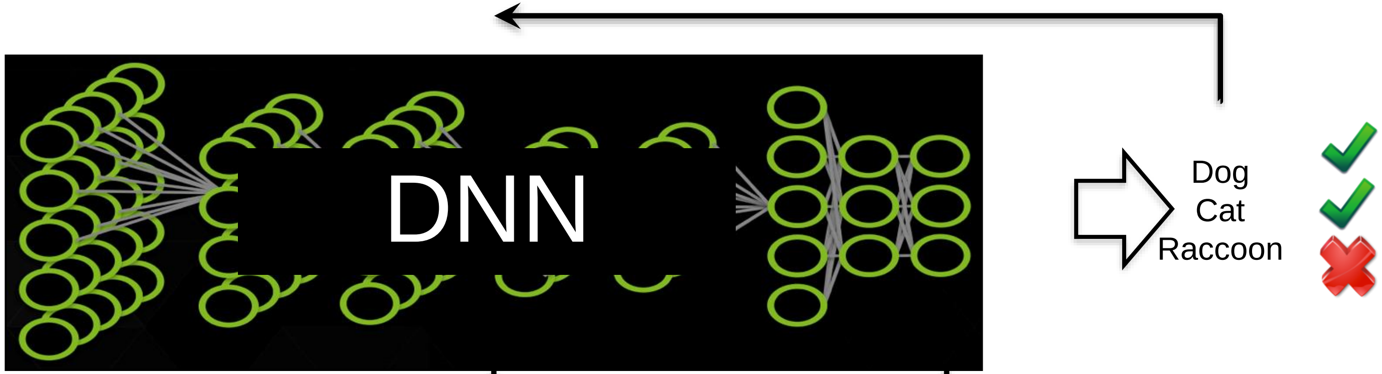
TRADITIONAL MACHINE PERCEPTION

Hand crafted feature extractors

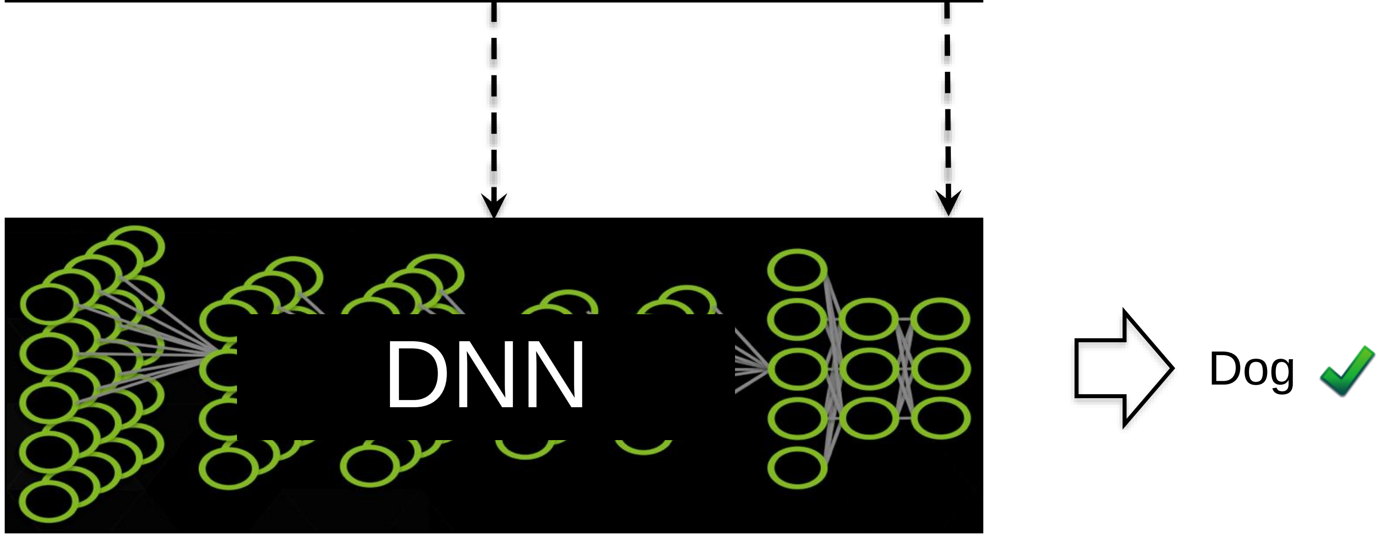
Raw data	Feature extraction	Classifier/ detector	Result
		SVM, shallow neural net, …	
		HMM, shallow neural net, …	Speaker ID, speech transcription, …
		Clustering, HMM, LDA, LSA …	Topic classification, machine translation, sentiment analysis…

DEEP LEARNING APPROACH

Training:



Inference:

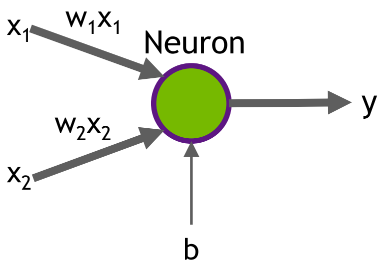


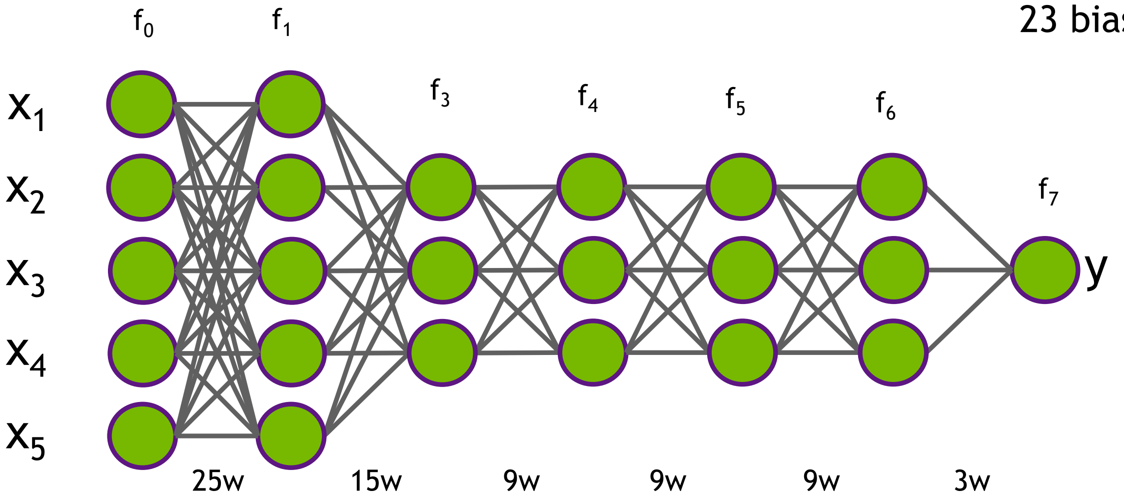
ALTERNATE VIEW: DEEP LEARNING – CONTINUOUS FUNCTION APPROXIMATION VIA SUMS OF HIERARCHICAL NON-LINEAR BASIS FUNCTIONS

A tiny example below…

<u>Free Parameters</u>
7 non-linear activations
70 weights
23 biases

$$y = f((\sum wx) + b)$$





+ the magic of backpropagation and stochastic gradient descent or other training methods

LANGUAGE UNDERSTANDING IMPROVEMENT

Reaching human level

GLUE Aggregate Score

Detect grammatical errors

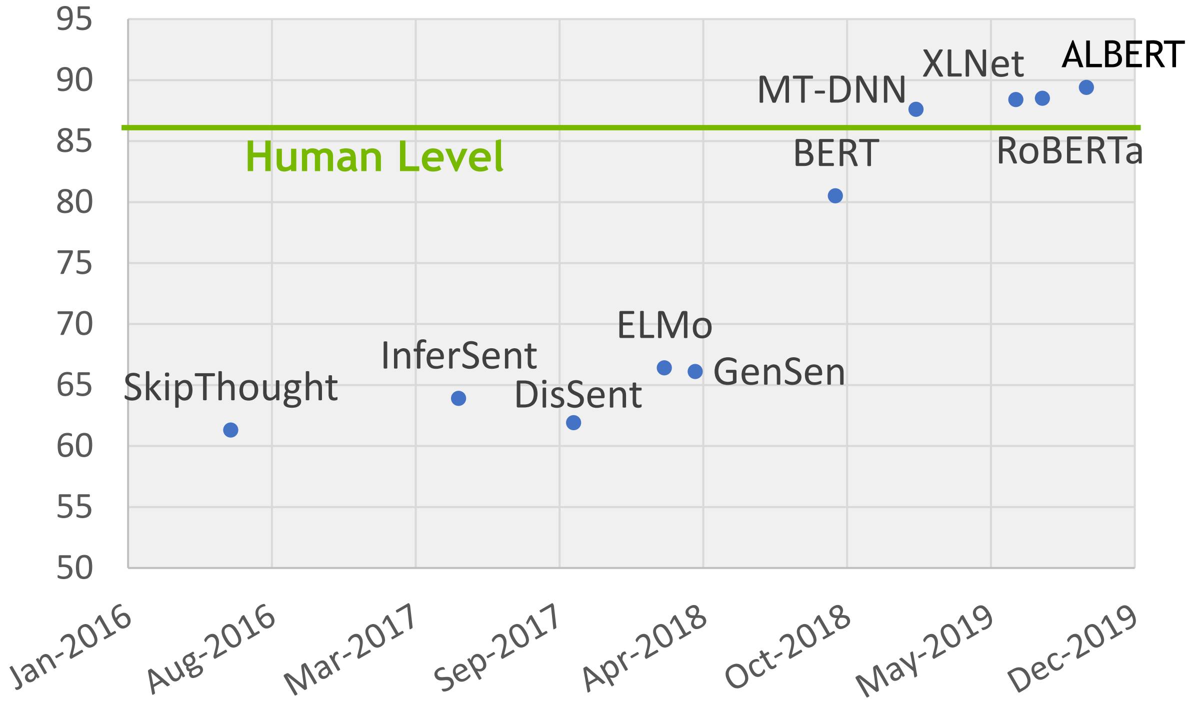
Predict if movie review is positive or negative

Decide if an abstract correctly summarizes an article

Sentence-level Semantic equivalence

Basic reading comprehension

Pronoun disambiguation



https://gluebenchmark.com/

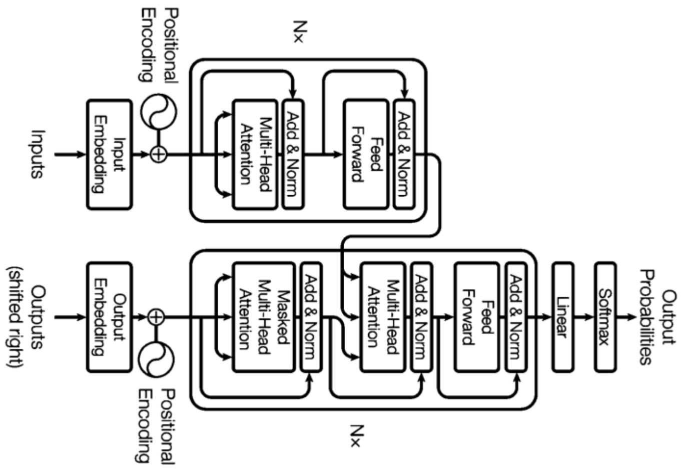
NATURAL LANGUAGE UNDERSTANDING

BERT universal language model

Input: Two sentences with 15% of words masked out

1 = “Initially he supported himself and his [redacted] by farming on a plot [redacted] family land.”

2 = “[redacted] in turn attracted the attention of [redacted] *St.* [redacted] *Post-Dispatch*, which sent a reporter to Murray to [redacted] review Stubblefield's wireless [redacted].”



Output 1: Reconstruct missing words

family, of
this, the, Louis, personally,
telephone

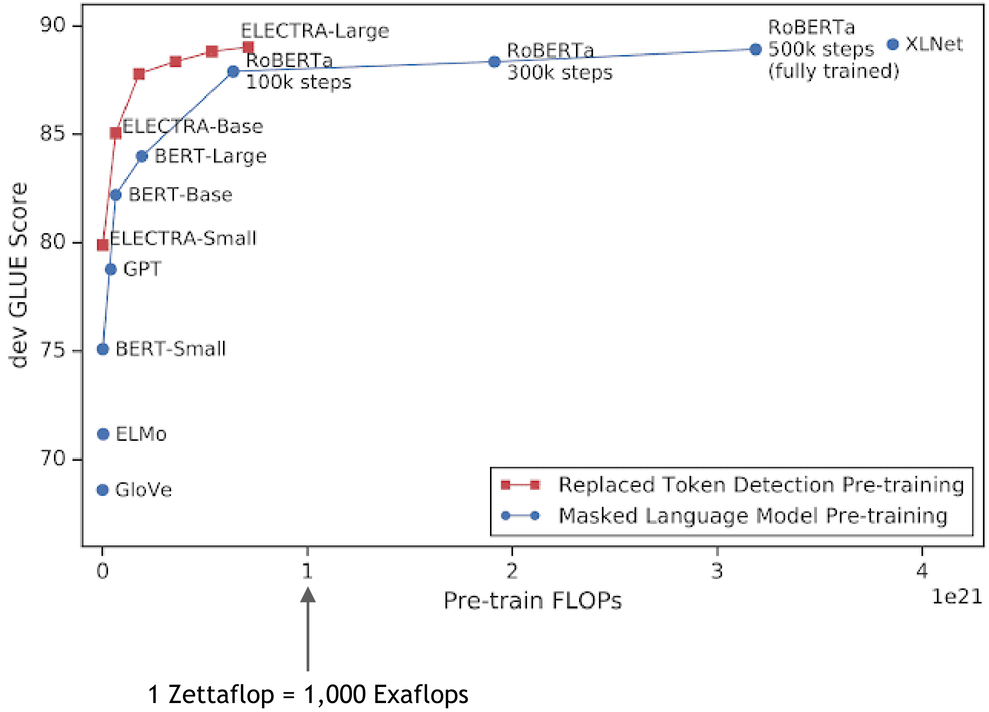
Output 2: Is two the next sentence after one?

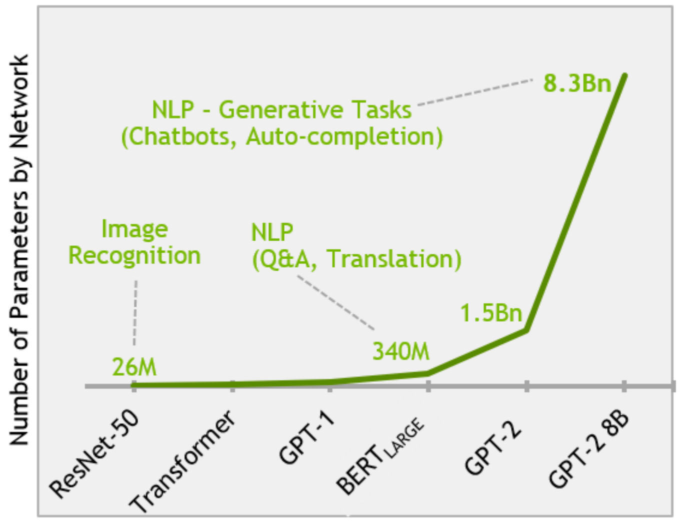
NOT_NEXT_SENTENCE

https://arxiv.org/abs/1810.04805

NLP MODELS ARE LARGE

The Training and Inference cost is high





EXPLODING MODEL SIZE

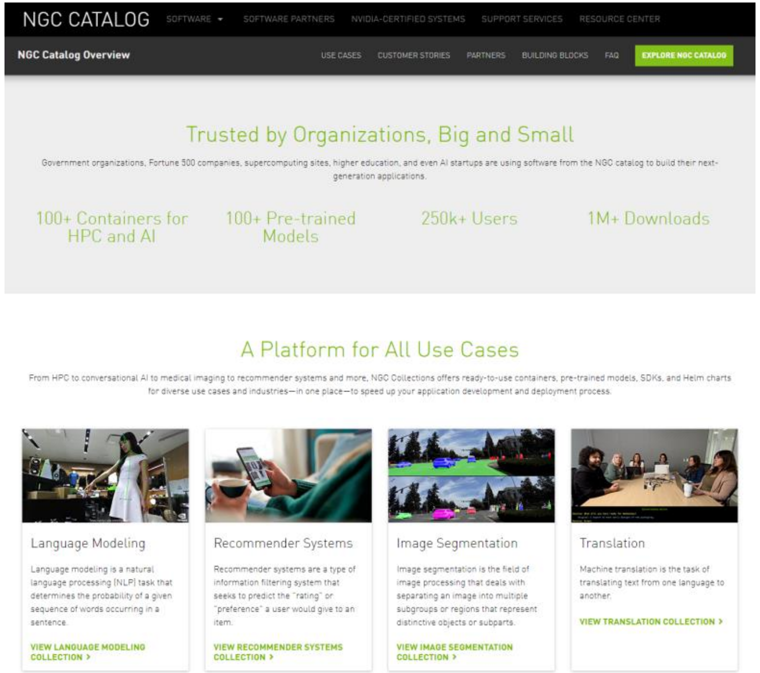
Complexity to Train

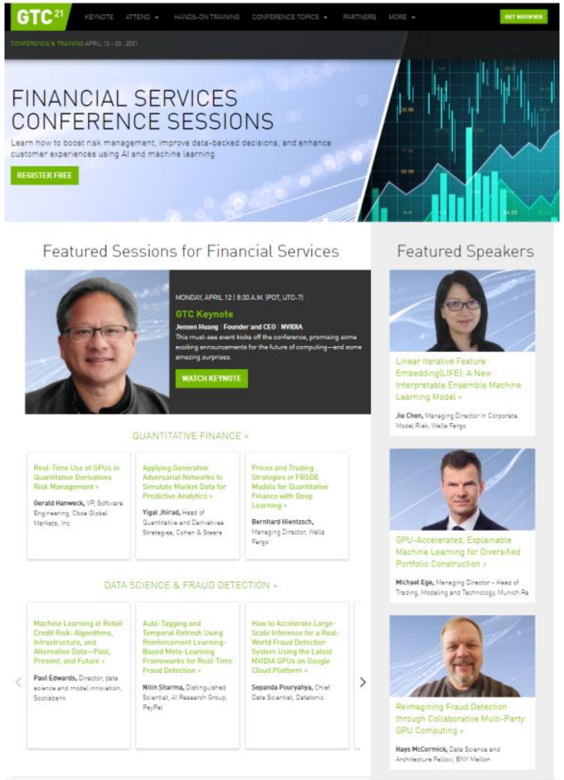
https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/

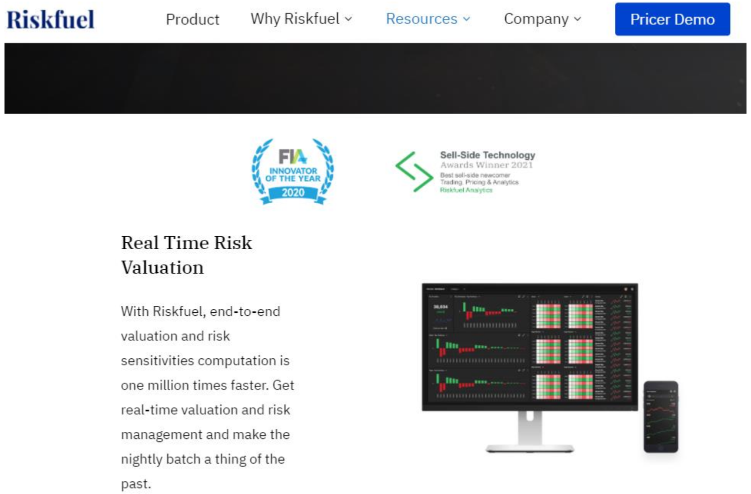
BUILDING BLOCKS

LINKS. LOTS OF LINKS

Your search engine of choice and these are a good start…







https://riskfuel.com/

https://www.nvidia.com/en-us/gpu-cloud/

https://www.nvidia.com/en-us/industries/finance/

https://www.nvidia.com/en-us/gtc/topics/financial-services/

https://info.nvidia.com/advancing-trader-tomorrow-ai-reg-page?ondemandrgt=yes#

https://www.cqfinstitute.org/content/quantspeak

QUESTIONS?