ECML PKDD 2020 Hands on Tutorial

Put Deep Learning to Work

Accelerate Deep Learning through AWS SageMaker and ML Services

Tim O'Brien (AWS)

Sr. Solutions Architect AI/ML

tpobrien@amazon.com

Rachel Hu (AWS)

Applied Scientist

rlhu@amazon.com

Wen-ming Ye (AWS)

Sr. Solutions Architect AI/ML

wye@amazon.com

aws

Objectives

- Understand AI/ML industry trends
- Overview on Machine Learning Tools at AWS
- Introduce to BERT deep dive
- Get Hands on with AWS
  - Elastic Container Service + NGC (In SageMaker)
  - Build, Train (multi-GPU), and Deploy (GPU host)

Agenda https://github.com/goldmermaid/ECMLPKDD2020

- Welcome and Logistics

- Amazon AI and SageMaker Overview

- BERT and Transformer

- Lab: Training/FineTuning Q&A model

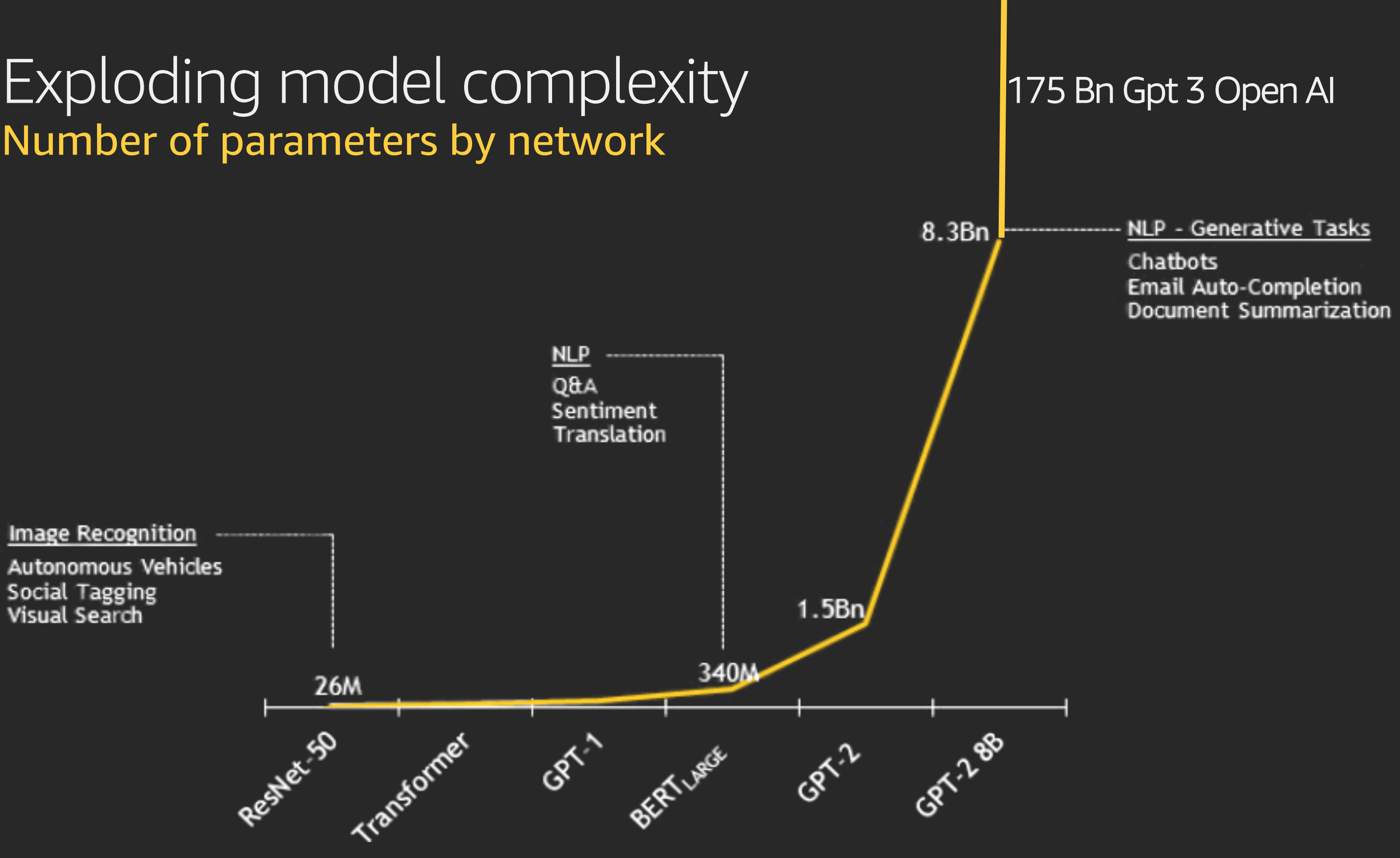
(10 min break during model training)

- Lab: Model Deployment

- Conclusion

Exploding model complexity

Number of parameters by network

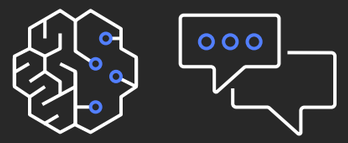


Machine learning use cases (Day One)

Applications that benefit from accelerated compute

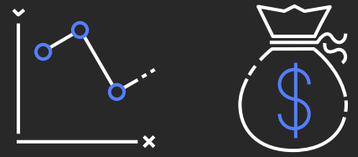
Machine learning/AI

Natural language processing



Image/ video analysis

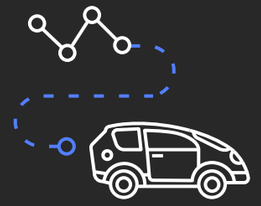
Financial services



Healthcare & life sciences

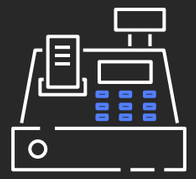
Manufacturing

Autonomous vehicle systems



Recommendation systems

Retail



Travel and hospitality

Energy

The AWS ML Stack

Broadest and most complete set of Machine Learning capabilities

AI SERVICES (Backed by our Open source Libraries)

VISION
- Amazon Rekognition

SPEECH
- Amazon Polly
- Amazon Transcribe *+Medical* New

TEXT/NLP
- Amazon Comprehend *+Medical*
- Amazon Translate
- Amazon Textract

SEARCH
- Amazon Kendra New

CHATBOTS
- Amazon Lex

PERSONALIZATION
- Amazon Personalize

FORECASTING
- Amazon Forecast

FRAUD
- Amazon Fraud Detector New

DEVELOPMENT
- Amazon CodeGuru New

CONTACT CENTERS
- Contact Lens *For Amazon Connect* New

ML SERVICES (Build your own)

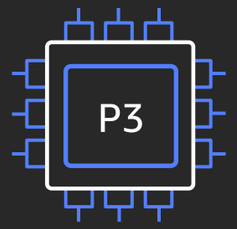
- Amazon SageMaker
- Ground Truth
- ML Marketplace
- SageMaker Studio IDE New
  - Built-in algorithms
  - Notebooks
  - Experiments New
  - Model training & tuning
  - Debugger New
  - Autopilot New
  - Model hosting
  - Model Monitor New
- Neo
- Augmented AI New

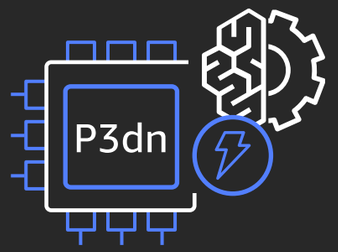
ML FRAMEWORKS & INFRASTRUCTURE (Framework support and EC2 Accelerated Instance Types)

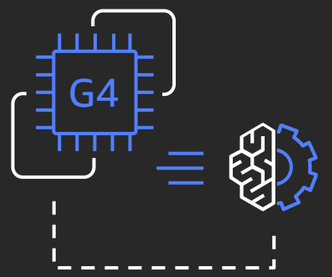
TensorFlow · mxnet · PYTORCH · GLUON · Keras · DeepGraphLibrary · scikit-learn · HOROVOD

Deep Learning AMIs & Containers | GPUs & CPUs | Elastic Inference | Inferentia New | FPGA

Accelerated compute portfolio for machine learning







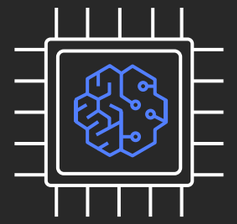
ML training

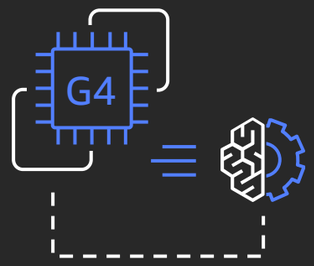
P3/P3dn GPU compute instance

- Up to 1 PetaFLOP of compute with 8x NVIDIA V100 GPUs
- Up to 256 GB of GPU memory
- Up to 100 Gbps of networking
- Designed to handle large distributed training jobs for fastest time to train

G4: GPU compute instance

- Up to 520 TeraFLOPs of compute with 8x NVIDIA T4 GPUs
- Cost-effective, small-scale training jobs





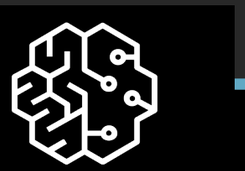
ML inference

AWS Inf1 instance

- Up to 2000 TOPs with 16x AWS-designed AWS Inferentia accelerators
- Lowest cost per inference in the cloud
- Designed for high throughput and low latency

G4: GPU compute instance

- Up to 1030 TOPs of compute with 8x NVIDIA T4 GPUs
- Increased performance, lower latency and reduced cost per inference compared to previous GPU-based instances

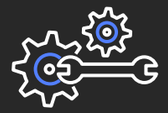


Amazon SageMaker

Machine learning for every developer & data scientist
Build, train, and deploy ML models at scale
https://aws.amazon.com/sagemaker/

Amazon SageMaker

Bringing machine learning to all developers

Pre-built notebooks for common problems	Built-in, high performance algorithms	One-click training	Optimization	One-click deployment	Fully managed with auto scaling, health checks, automatic handling of node failures, and security checks
					
Collect and prepare training data	Choose and optimize your ML algorithm	Set up and manage environments for training	Train and tune model (trial and error)	Deploy model in production	Scale and manage the production environment

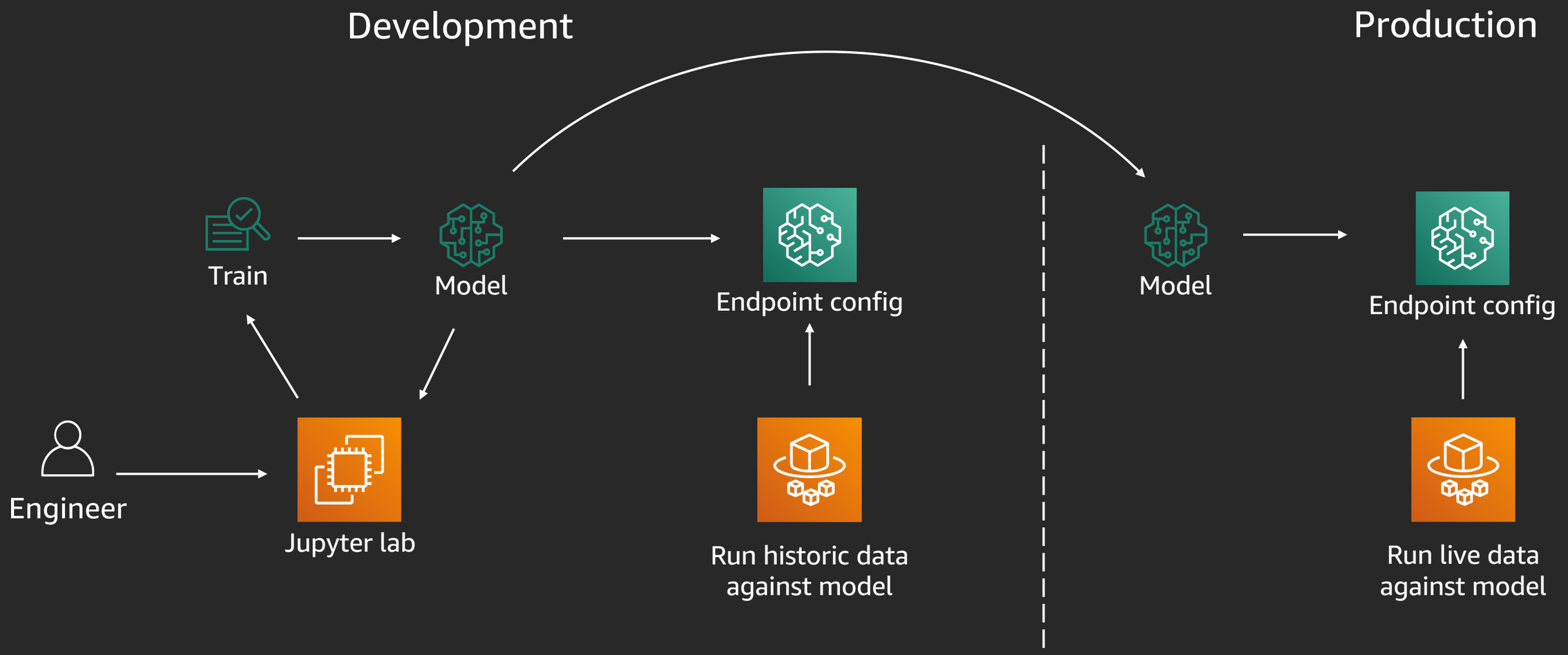
End-to-end machine learning platform

Flexible model training

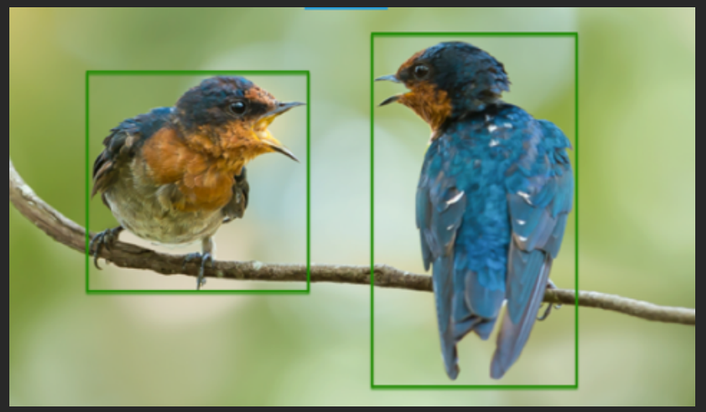
Keras · mxnet · GLUON · Chainer · PyTorch · TensorFlow · HOROVOD

Pay by the second

Build, train, and test with Amazon SageMaker



Use pre-built labeling workflows or set up your own



Bounding boxes

Image Classification

Text Classification

Semantic Segmentation

Custom and
Active Learning

Pools of Workforces

Public

An on-demand 24 x7 workforce of over 500,000 independent Contractors worldwide, powered by Amazon Mechanical Turk

Private

A team of workers that you have sourced yourself, including your own employees or contractors for handling data that needs to stay within your organization

Vendors

A curated list of third party vendors that specialize in providing data labeling services, available via the AWS Marketplace

3 ways to train models with SageMaker

Built-in Algorithms

Bring-your-own script (TensorFlow, Scikit Learn, PyTorch, …)

Bring-your-own container

Amazon SageMaker has built-in algorithms or bring your own

Classification

- Linear Learner
- XGBoost
- KNN

Working with Text

- BlazingText
- Supervised
- Unsupervised

Sequence Translation

- Seq2Seq

Computer Vision

- Image Classification
- Object Detection
- Semantic Segmentation

Recommendation

- Factorization Machines

Anomaly Detection

- Random Cut Forests
- IP Insights

Regression

- Linear Learner
- XGBoost
- KNN

Topic Modeling

- LDA
- NTM

Forecasting

- DeepAR

Clustering

- KMeans

Feature Reduction

- PCA
- Object2Vec

Amazon SageMaker APIs

Hardware →

Parameters →

Start training →

```
import boto3
import sagemaker

sess = sagemaker.Session()

pca = sagemaker.estimator.Estimator(containers[boto3.Session().region_name],
                                    role,
                                    train_instance_count=1,
                                    train_instance_type='ml.c4.xlarge',
                                    output_path=output_location,
                                    sagemaker_session=sess)
pca.set_hyperparamters(feature_dim=50000,
                       num_components=10,
                       subtract_mean=True,
                       algorithm_mode='randomized',
                       mini_batch_size=200)

pca.fit({'train': s3_train_data})
```

Host model →

```
pca_predictor = pca.deploy(initial_instance_count=1,
                           instance_type='ml.c4.xlarge')
```

Distributed training with Amazon SageMaker

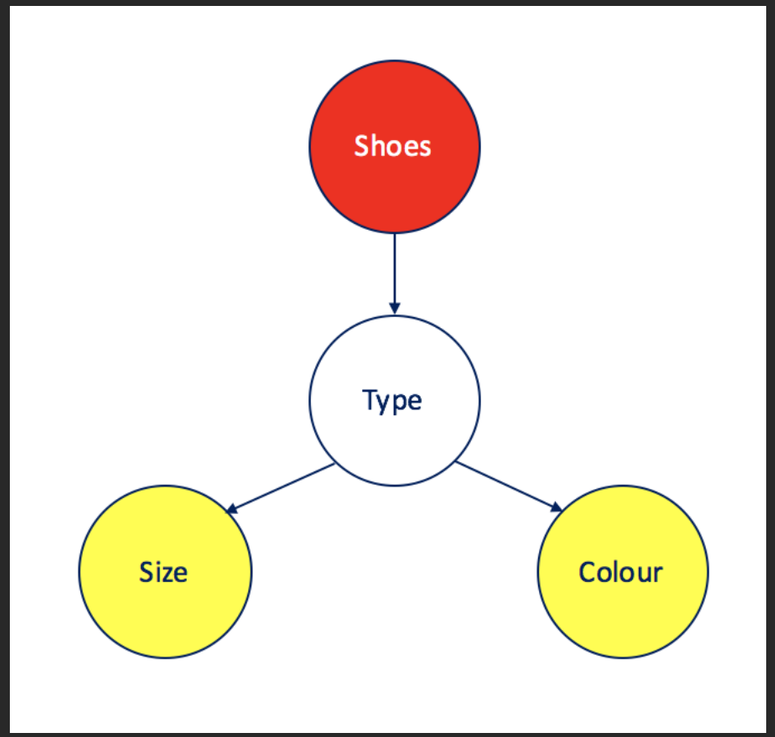
- ✓ Horovod (synchronous workers)
- ✓ Parameter servers (synchronous workers)

```
estimator = TensorFlow(entry_point='script.py', role=role,
        train_instance_type='ml.p3.2xlarge',
      train_instance_count=2,
      distributions= {'parameter_server': {'enabled': True}} )
```

Hyper-Parameter Optimization (HPO)

Hyper-parameters structure many every day processes but few people are aware of their existence. When you are looking for a pair of shoes to buy, you are solving a hyper-parameter optimization problem.

Once you have decided which type of shoes you are going to buy (hyper-parameter) you need to fit the parameters of the shoes to meet your requirements (size, color,…). Your available budget is also a hyper-parameter that will affect your choices.



Introducing Amazon SageMaker Autopilot
Automatic model creation with full visibility & control

Quick to start

Provide your data in a tabular form & specify target prediction

Automatic model creation

Get ML models with feature engineering & automatic model tuning automatically done

Visibility & control

Get notebooks for your models with source code

Recommendations & optimization

Get a leaderboard & continue to improve your model

Deploy & Manage

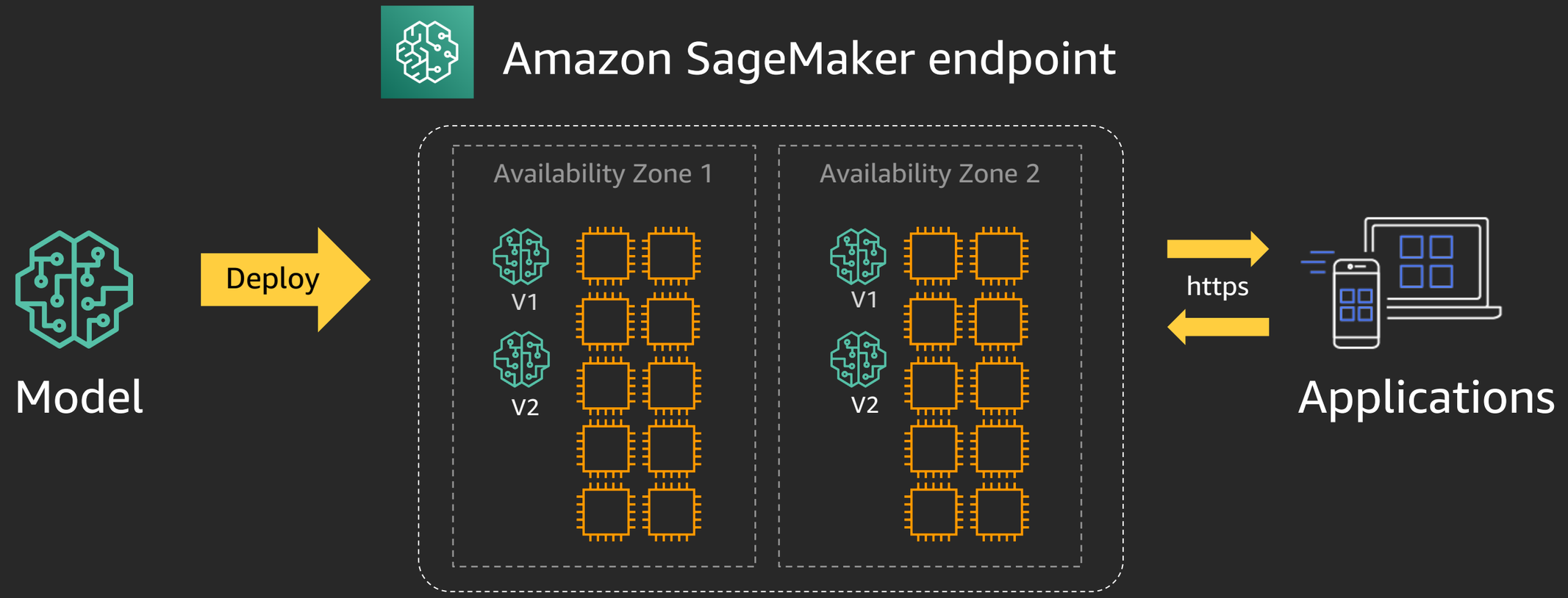
2 way to make predictions

Real-time, on-demand

Batch inference

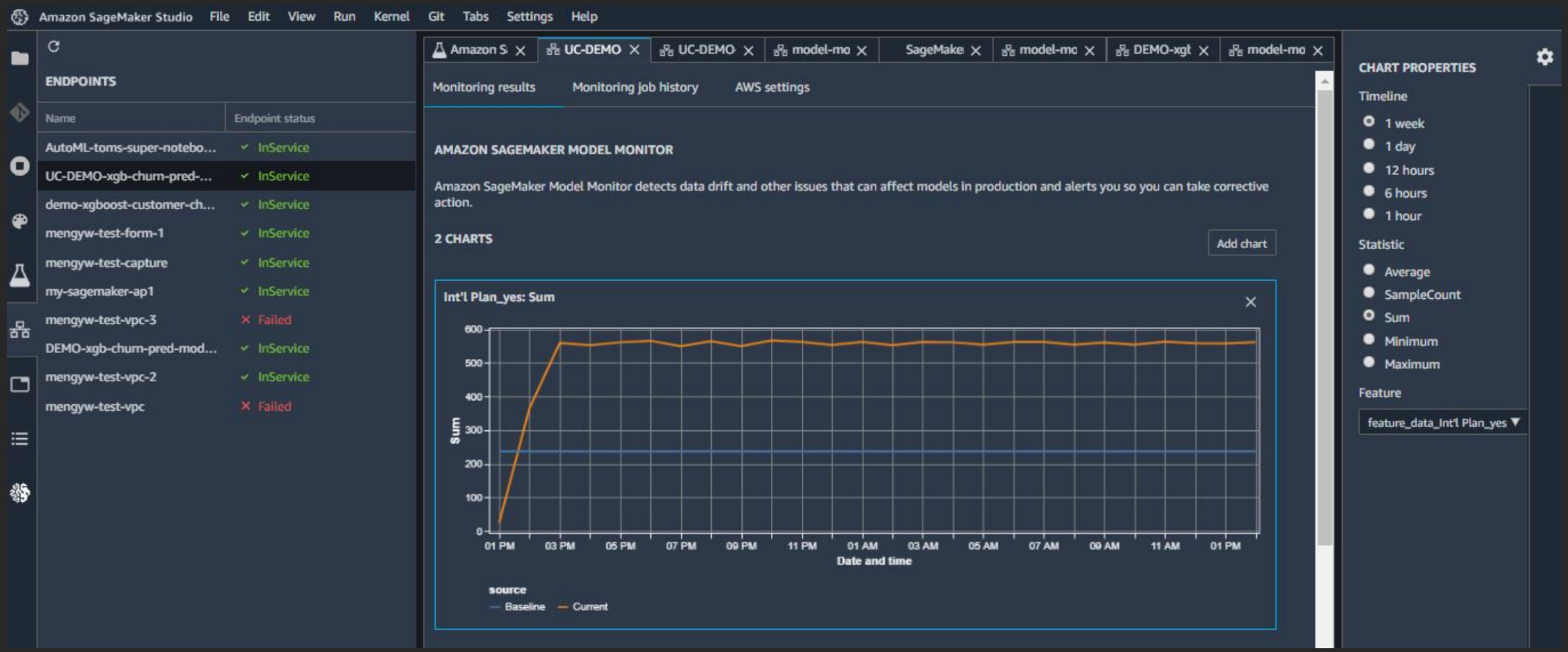
Fully managed model hosting at scale

Letting data scientists focus on models



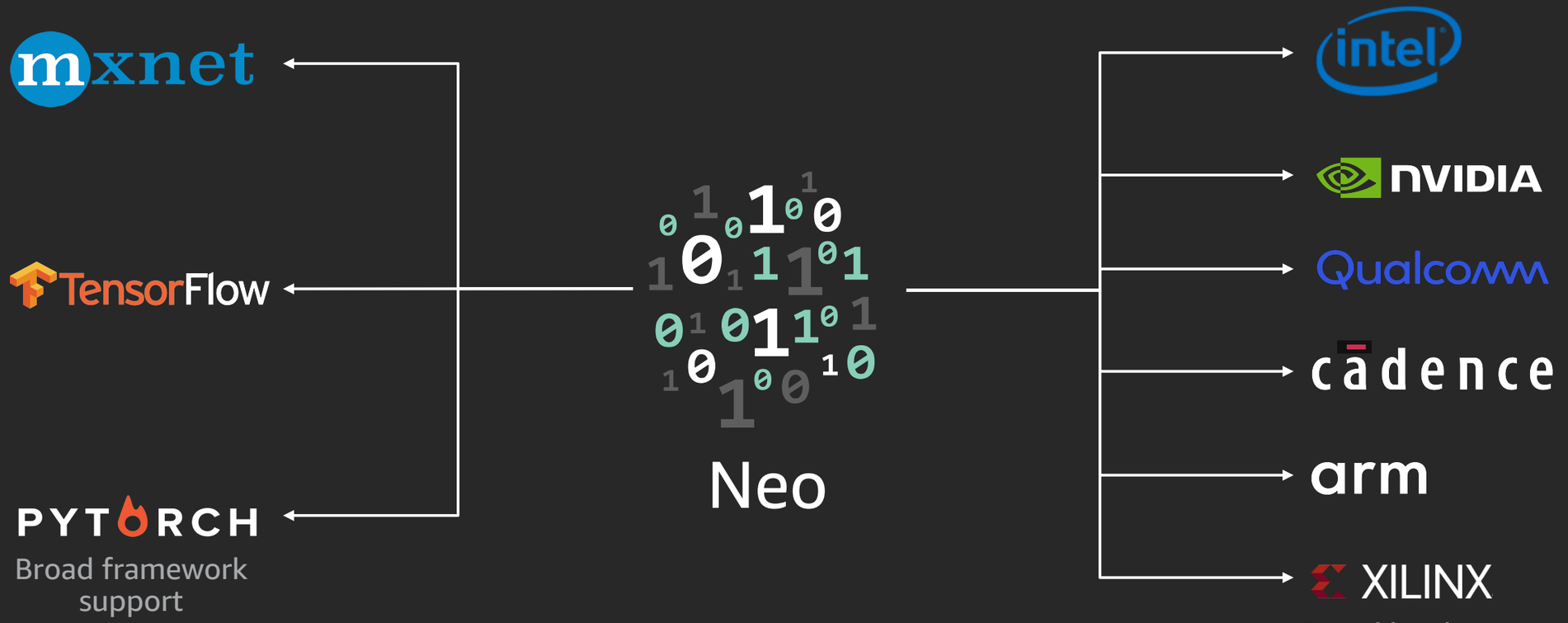
- Autoscaling across multiple AZs for variable loads
- A/B testing support

Use Amazon SageMaker Model Monitor to identify model drift and take action

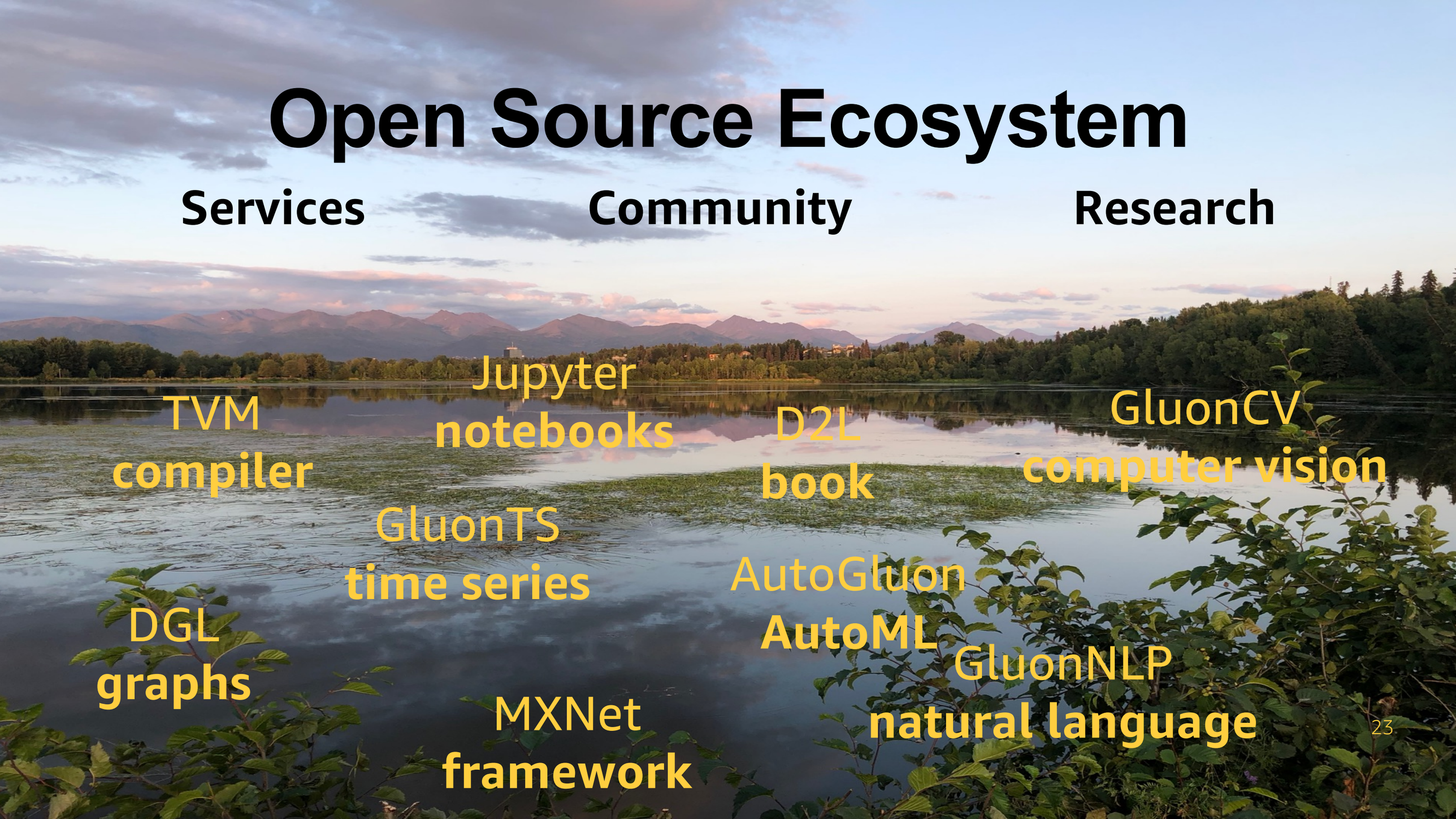


Amazon SageMaker Neo

Train once and run anywhere with 2x performance



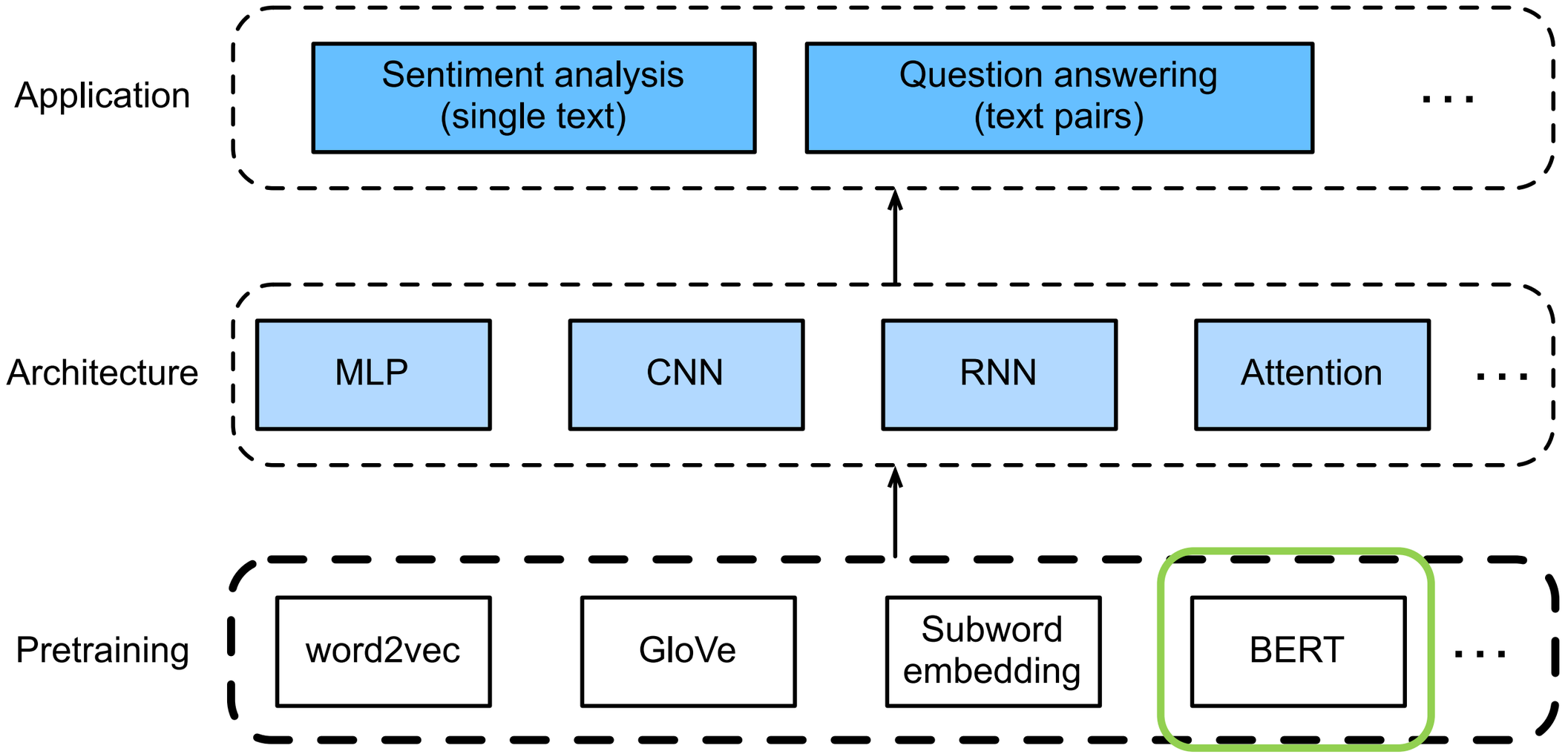
Open-source Neo-AI device runtime and compiler
1/10th the size of original frameworks


Open Source Ecosystem
Services
Community
Research
TVM
compiler
Jupyter
notebooks
D2L
book
GluonCV
computer vision
GluonTS
time series
AutoGluon
AutoML
DGL
graphs
GluonNLP
natural language
MXNet
framework

Additional Resources

- MLU on YouTube: https://bit.ly/31cBcf3
- Gluon:
  - Deep Graph Library https://www.dgl.ai/
  - NLP http://gluon-nlp.mxnet.io/
  - Computer Vision http://gluon-cv.mxnet.io/
  - Time Series https://gluon-ts.mxnet.io/
- Dive into Deep Learning: http://d2l.ai/

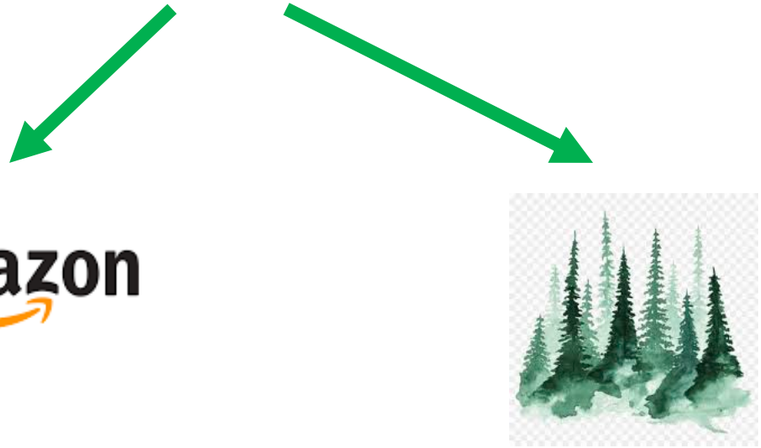
NLP Workflow



BERT

Traditional language modeling has problems …

Amazon is on fire…

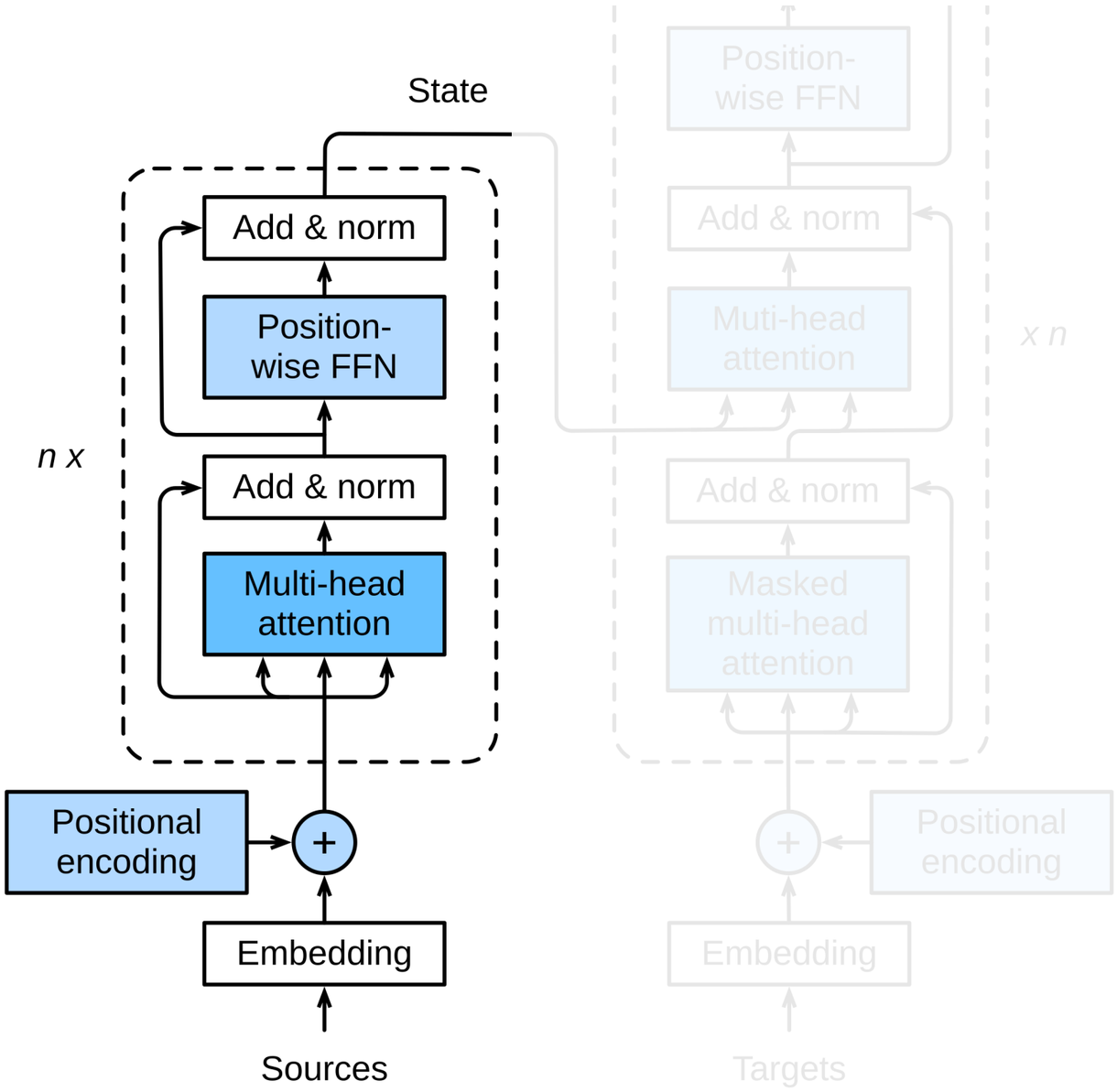


BERT

BERT –
Bidirectional Encoder Representations from Transformers

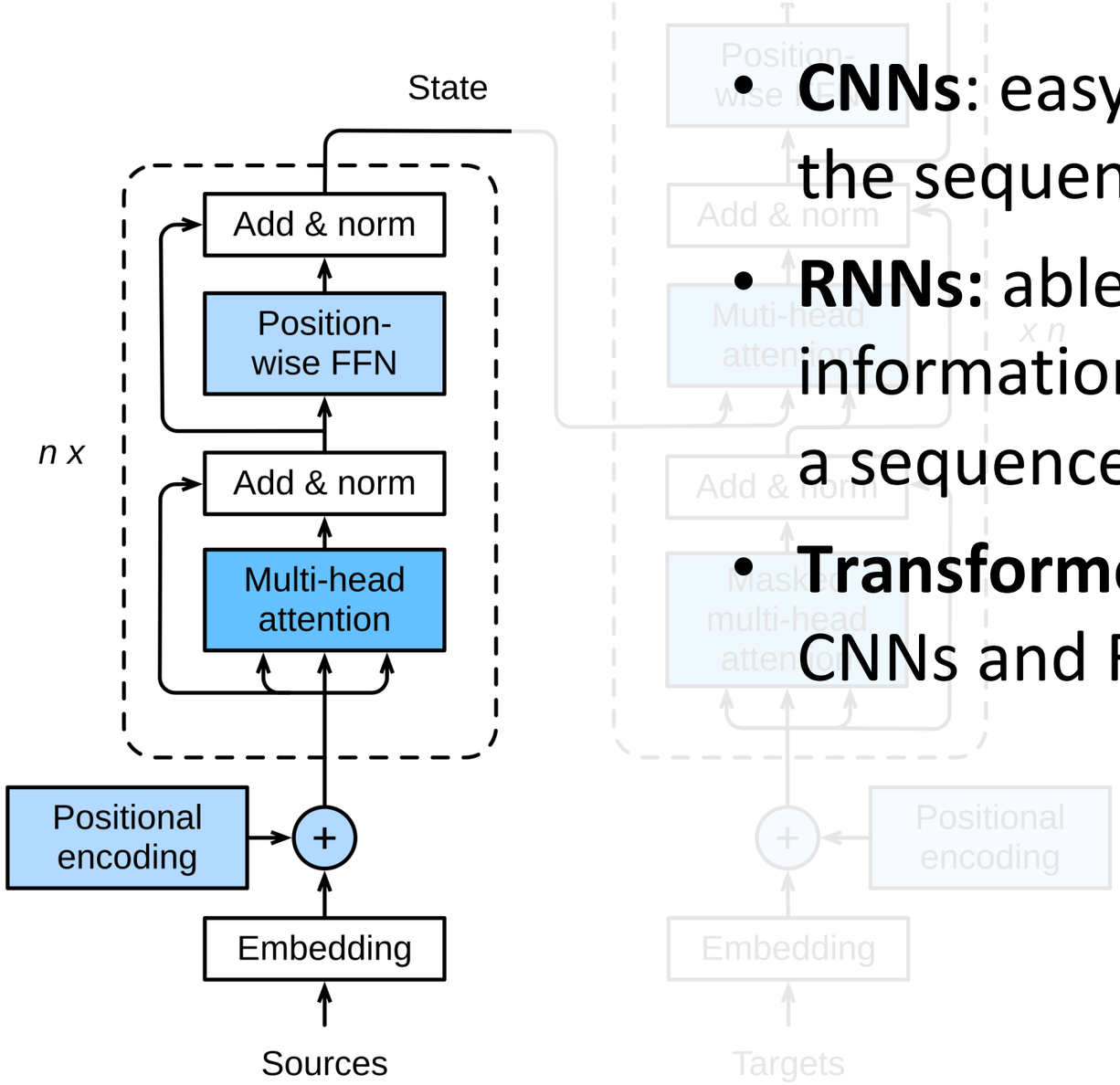
- Bidirectional Context
- Built on a Transformer encoder
- *Task-agnostic* model
- Improved 11 NLP tasks to SOTA

BERT



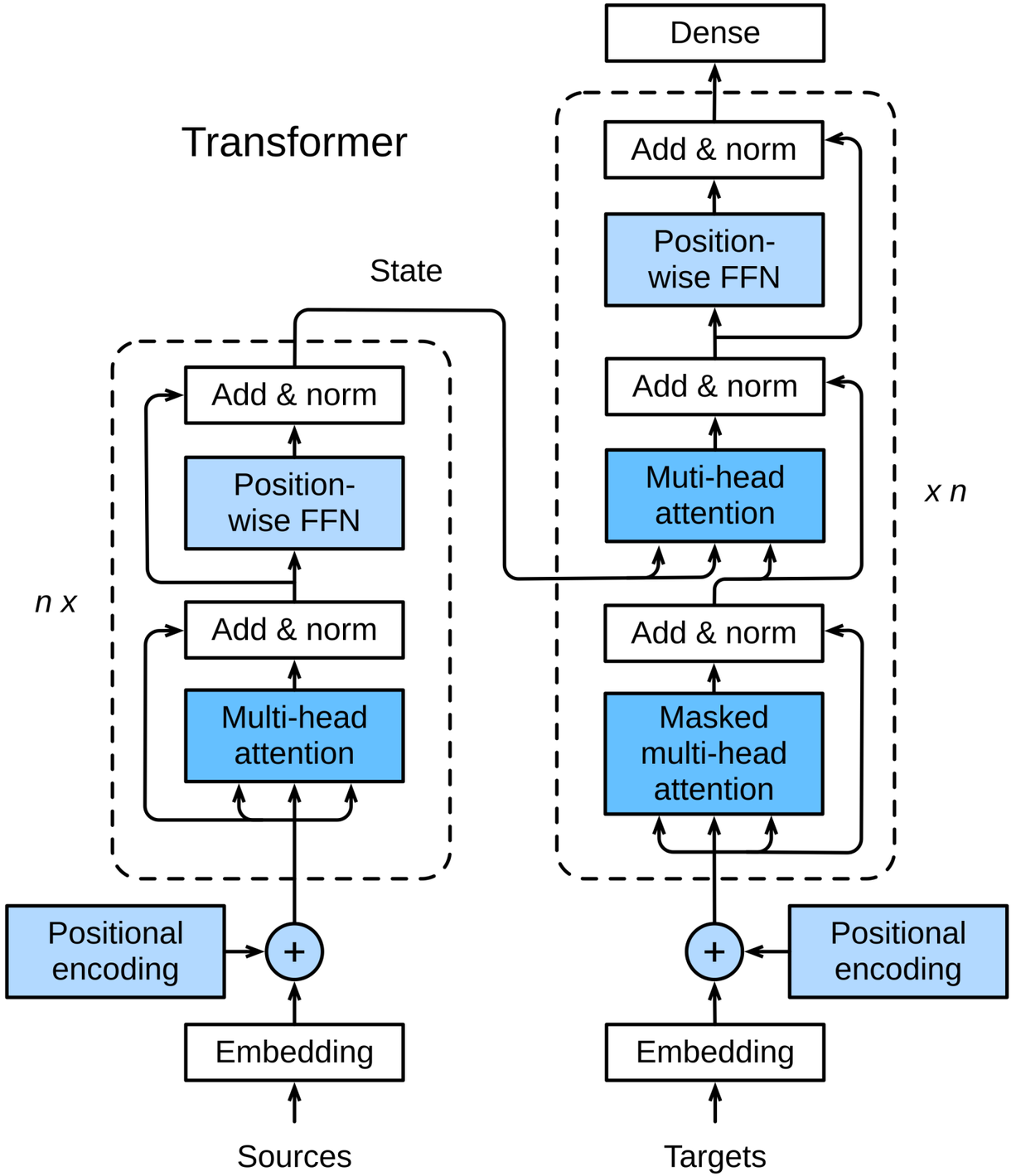
Bidirectional Encoder Representations from **Transformers**

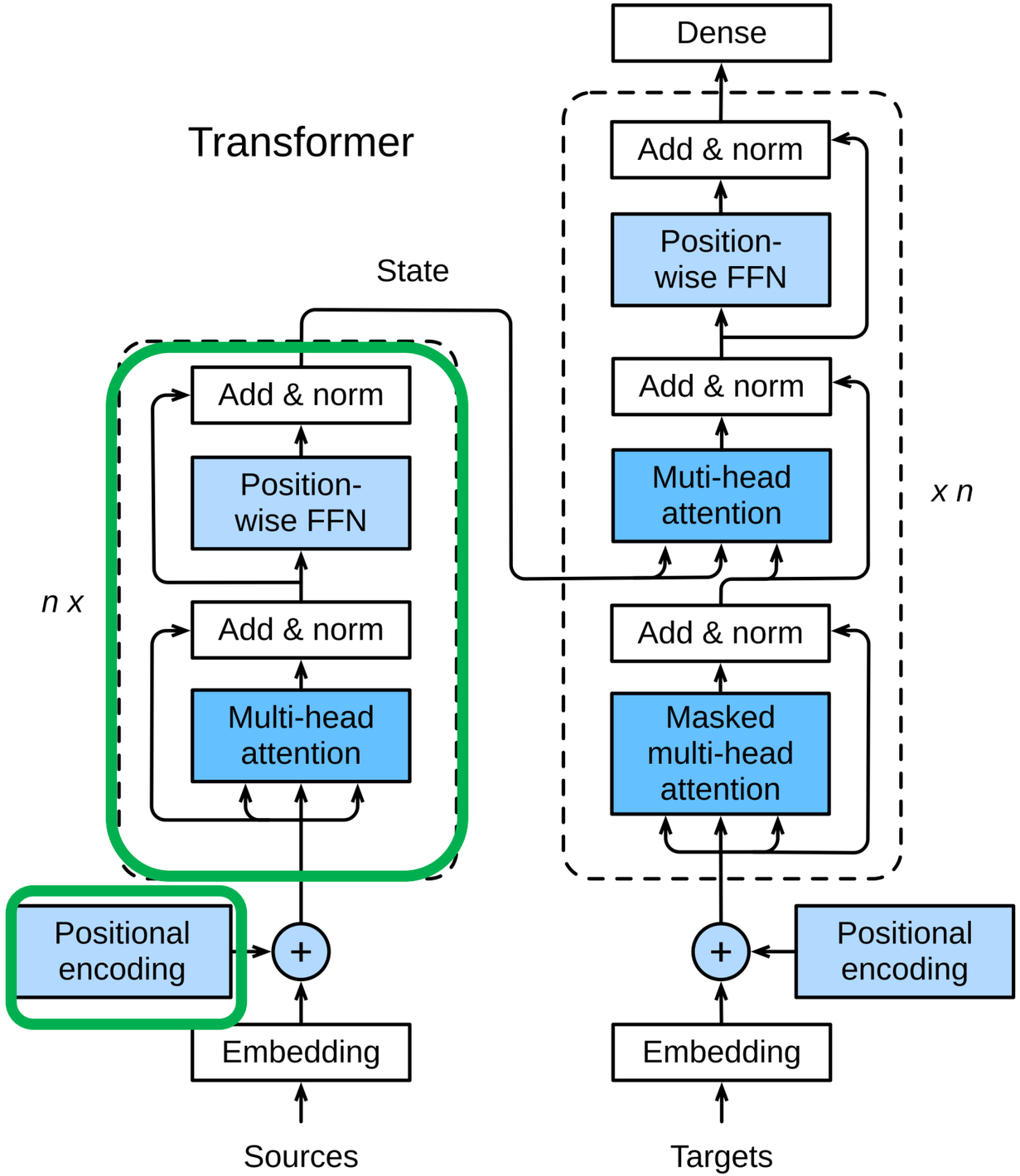
Transformer*



- **CNNs**: easy to parallelize, but cannot capture the sequential dependency
- **RNNs:** able to capture the sequential information, but unable to parallelize within a sequence
- **Transformer:** combine the advantages of CNNs and RNNs

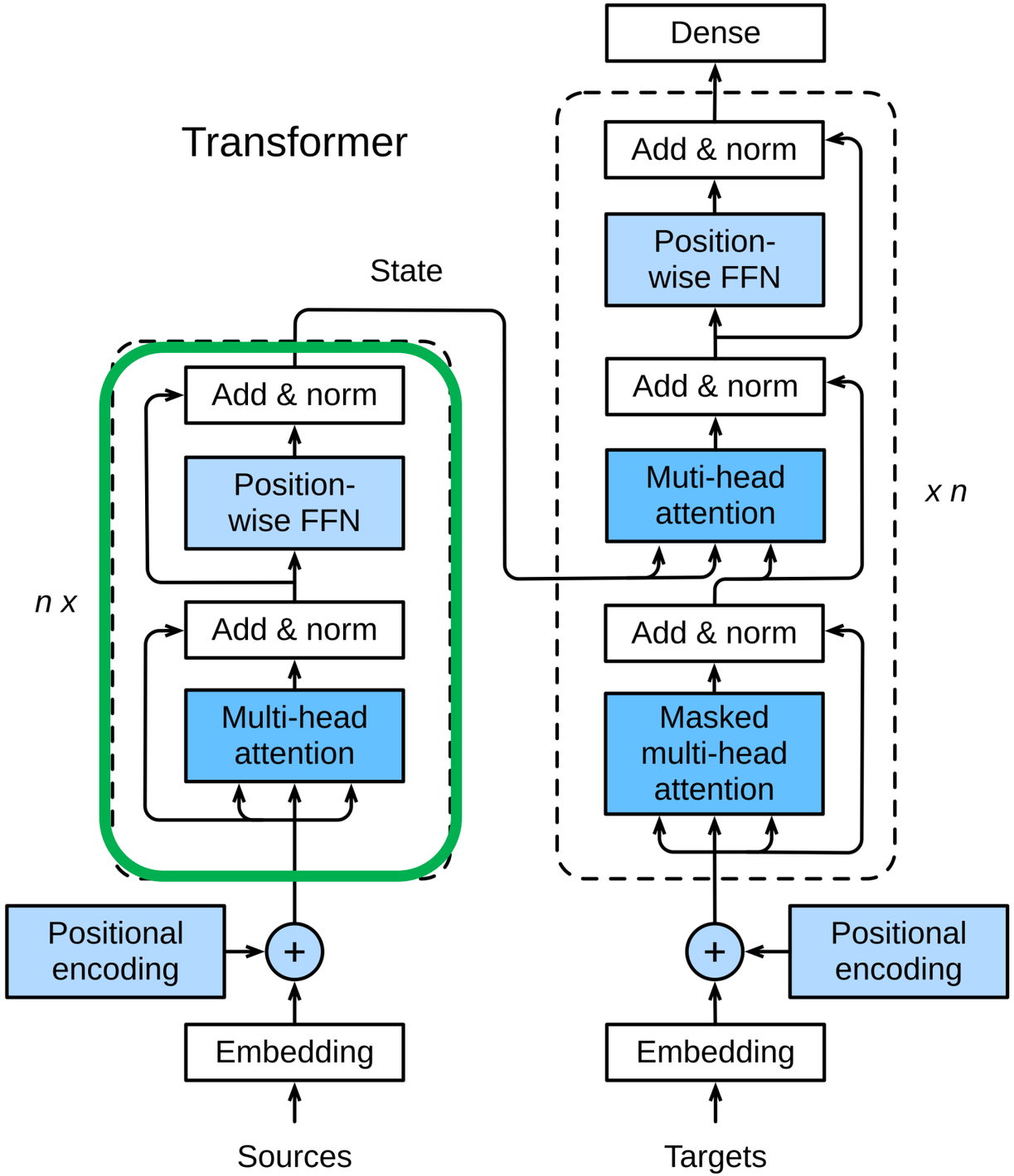
*Vaswani, et al., 2017 - “attention is all you need”


Transformer
Dense
Add & norm
Position-
wise FFN
State
Add & norm
Muti-head
attention
x n
Add & norm
Masked
multi-head
attention
Add & norm
Position-
wise FFN
Add & norm
Multi-head
attention
n x
Positional
encoding
+
Positional
encoding
+
Embedding
Embedding
Sources
Targets



Transformer Block

Positional Encoding



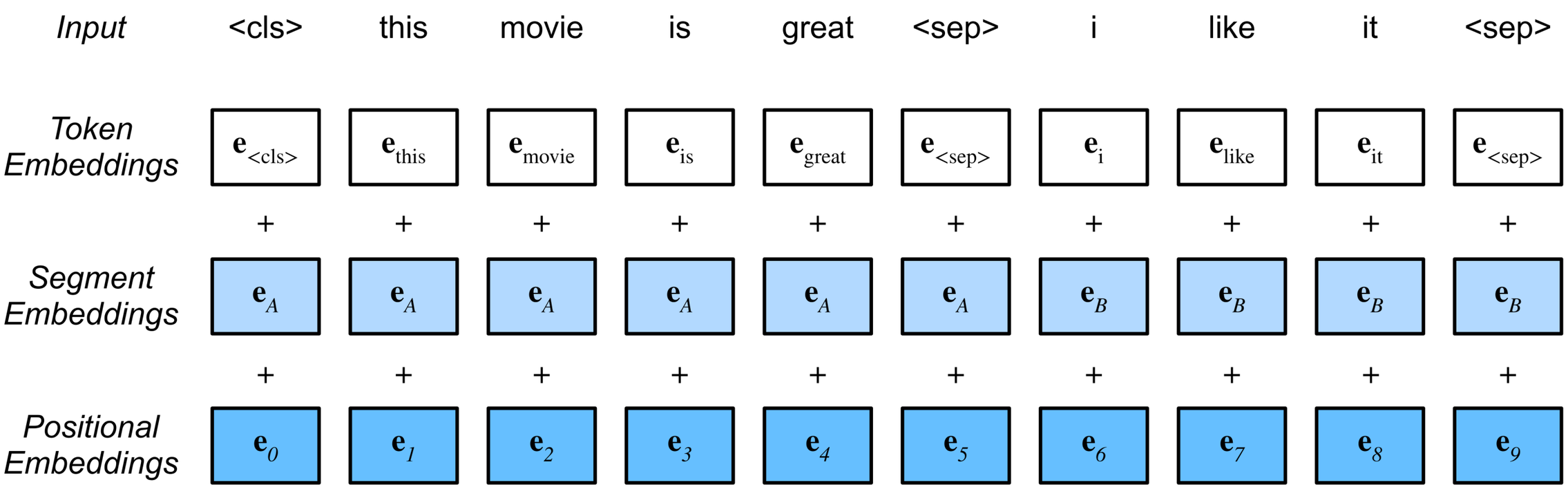
Transformer Block:

- Multi-head attention
- Position-wise feed forward network (FFN)
- Layer normalization (“Add & Norm”)

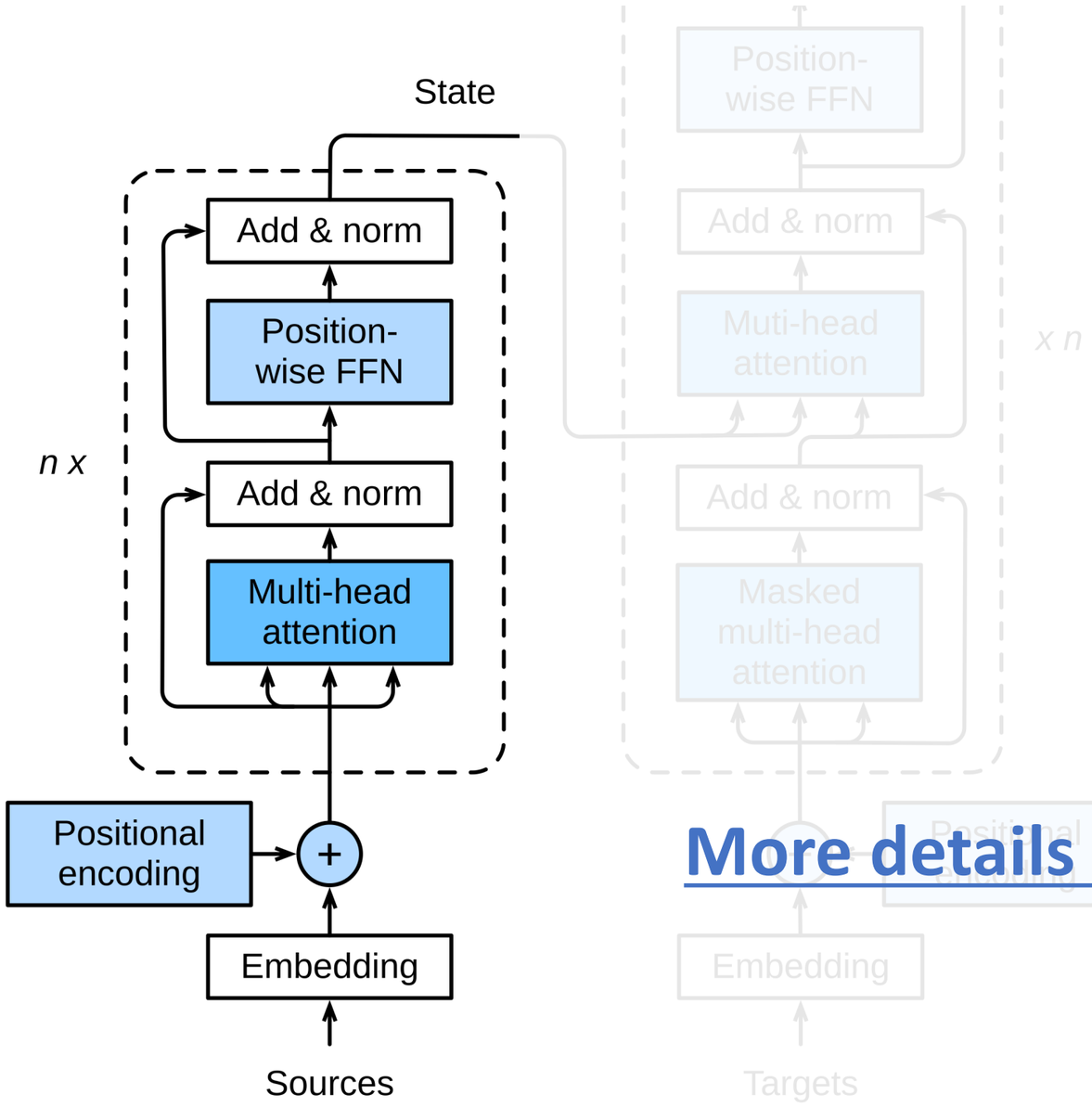
More details on D2L

BERT

- BERT input sequence embeddings



BERT



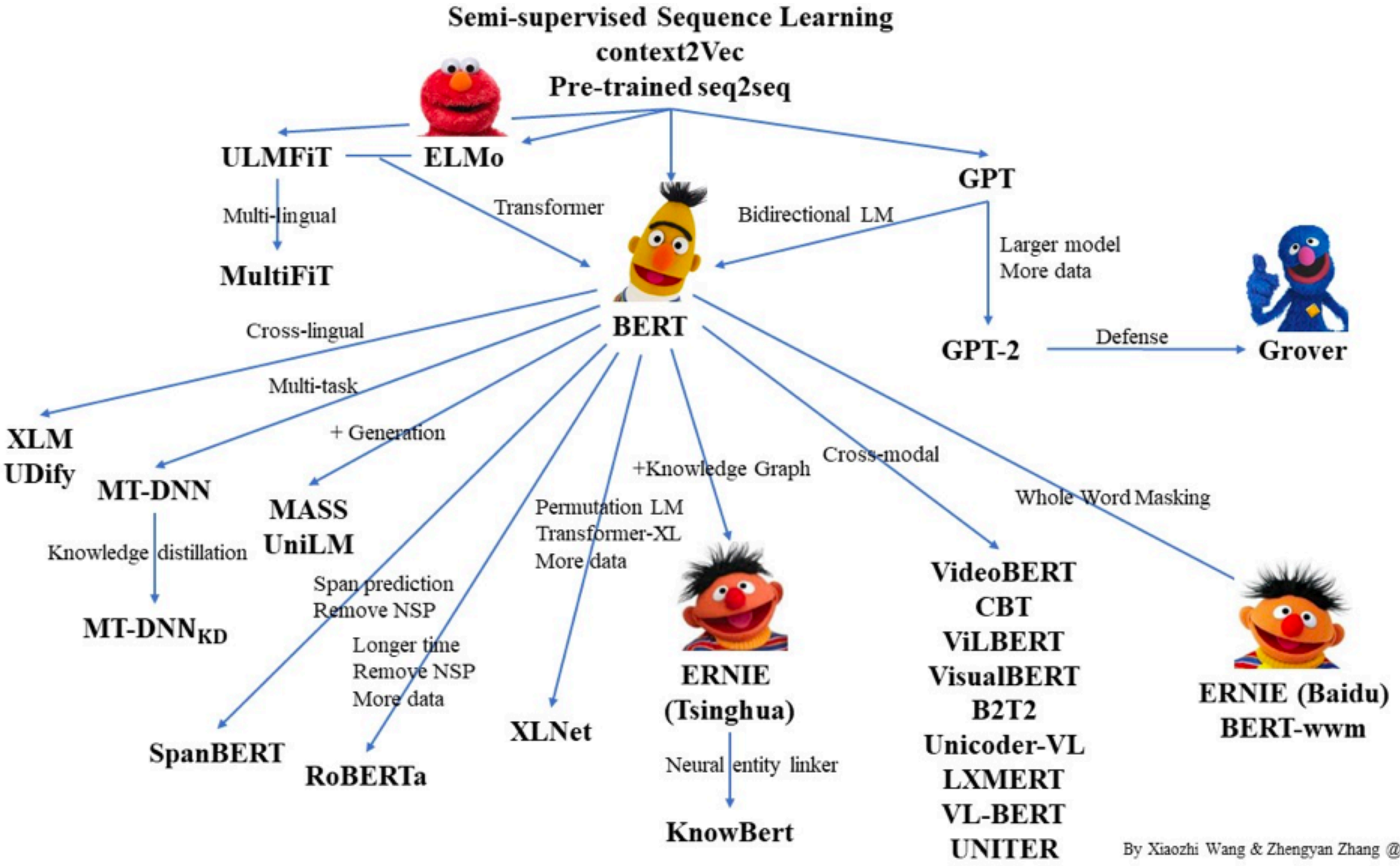
Bidirectional Encoder Representations from **Transformers**

More details on D2L

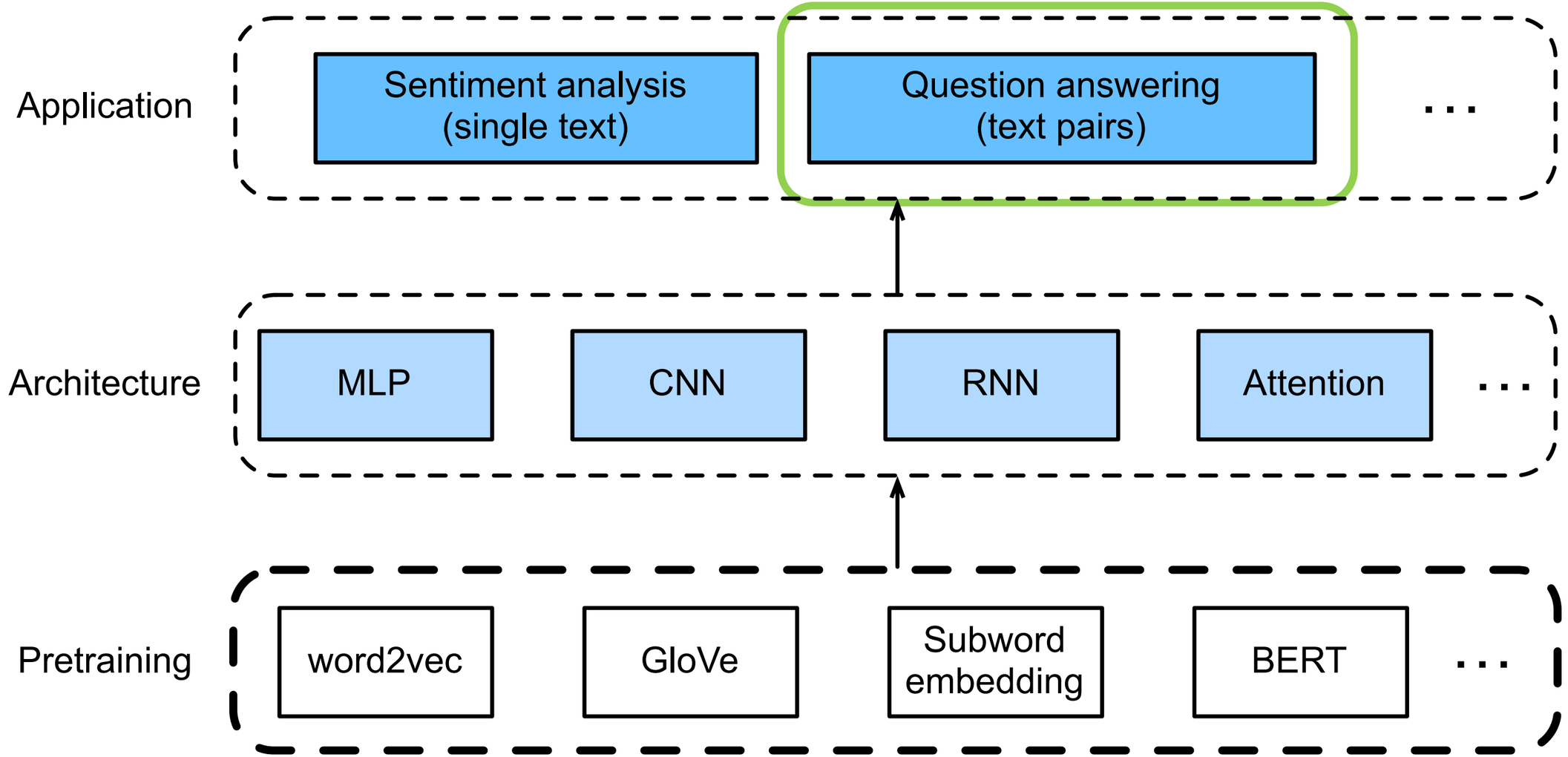
BERT

Two Methods of Usages:

- Pretrained:
  - Only train an additional output layer based on the NLP task
  - Minimal model architecture changes
- Finetuned:
  - Train an output layer
  - Tune all parameters based on a specific task and datasets


Semi-supervised Sequence Learning
context2Vec
Pre-trained seq2seq
ULMFiT
ELMo
GPT
Multi-lingual
MultiFiT
Transformer
Bidirectional LM
Larger model
More data
BERT
GPT-2
Defense
Grover
Cross-lingual
XLM
UDify
Multi-task
MT-DNN
Knowledge distillation
MT-DNN$_{KD}$
+ Generation
MASS
UniLM
Span prediction
Remove NSP
SpanBERT
Longer time
Remove NSP
More data
RoBERTa
Permutation LM
Transformer-XL
More data
XLNet
+Knowledge Graph
ERNIE
(Tsinghua)
Neural entity linker
KnowBert
Cross-modal
VideoBERT
CBT
ViLBERT
VisualBERT
B2T2
Unicoder-VL
LXMERT
VL-BERT
UNITER
Whole Word Masking
ERNIE (Baidu)
BERT-wwm
By Xiaozhi Wang & Zhengyan Zhang @THUNLP

NLP Workflow

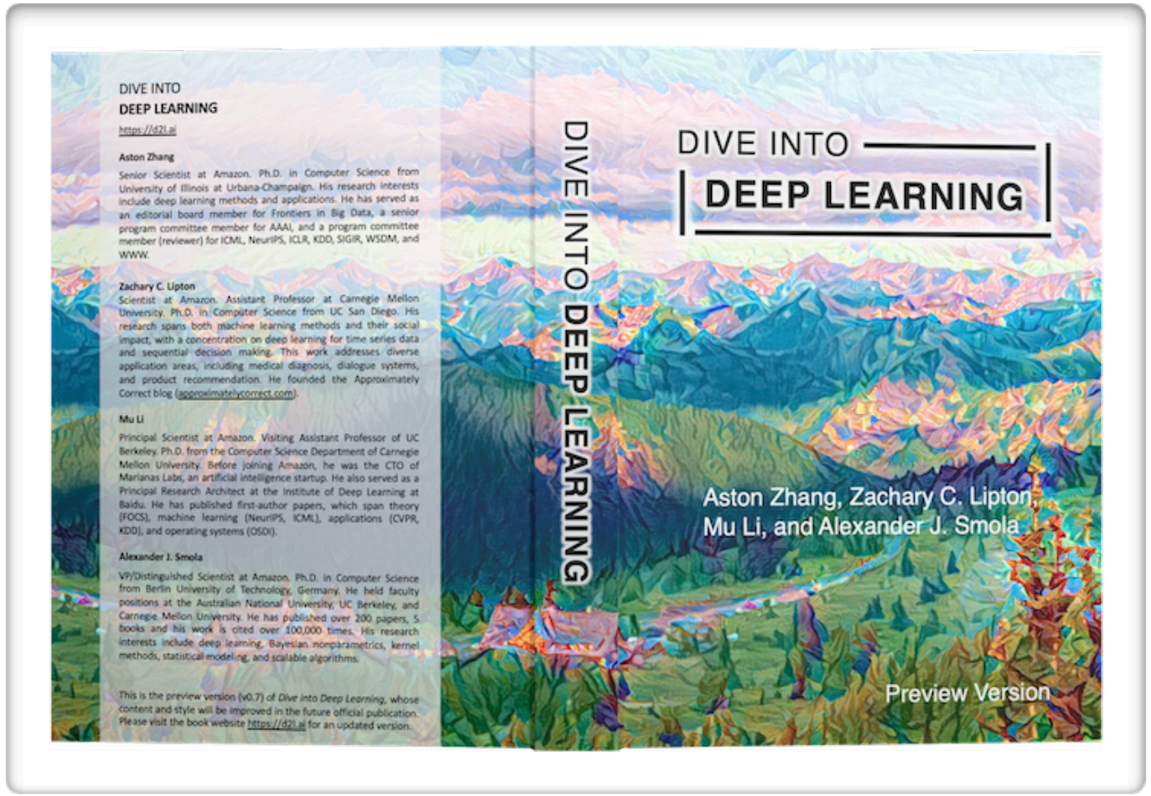


NLP Application

Question answering

- Question: Who shall use D2L?
- Passage context: *Dive into Deep Learning* (D2L, D2L.ai) is an interactive open-sourced deep learning online book with code, math, and discussions. It is designed for engineers, researchers and students to learn state of the art deep learning models with in-depth theory and fast prototype notebooks.

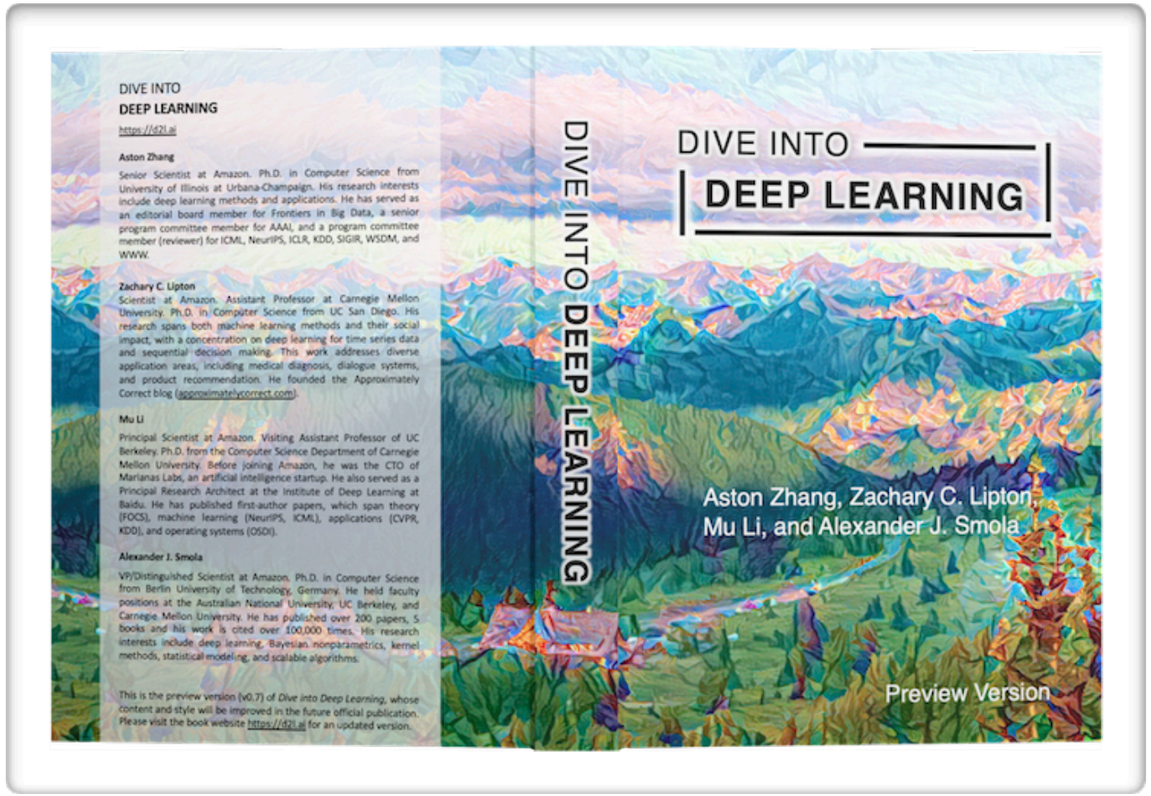
D2L



D2L.ai

- Adopted as a textbook or reference book at UC Berkeley, CMU, MIT, and 70+ universities worldwide
- Berkeley class (slides, videos at courses.d2l.ai)
- Wide theoretical coverage: statistics, optimization, machine learning basics, GPU parallel training, etc.
- 150+ runnable Jupyter Notebooks from model architectures to applications (CV, NLP, etc.)
- Multilingual content EN, ZH (in progress: KO, JA, FR, TR)

D2L

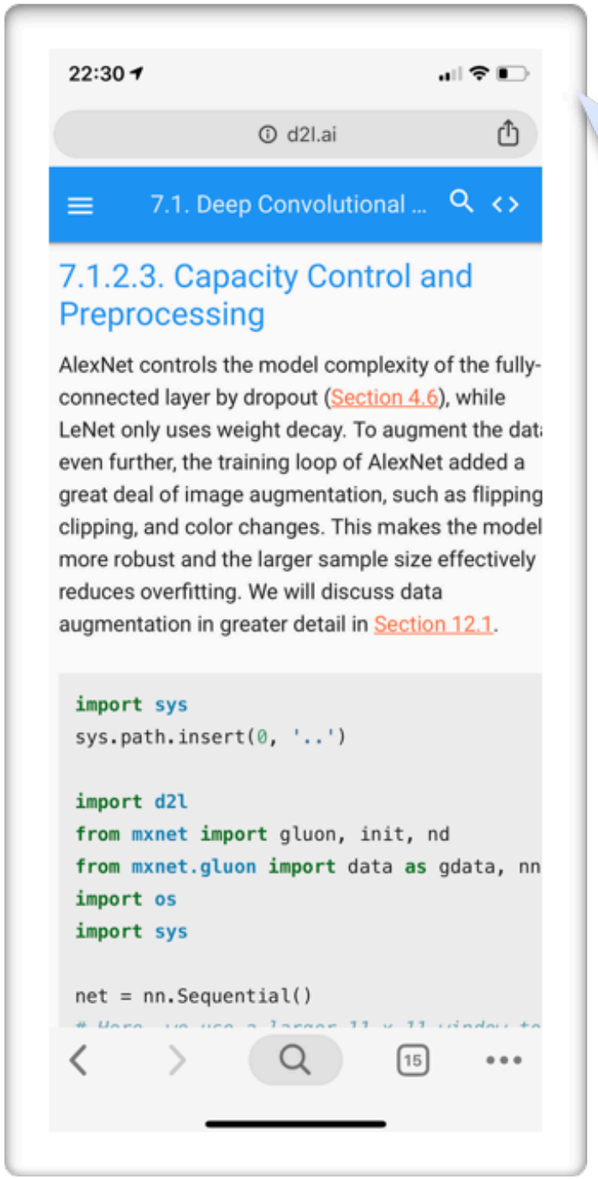


[D2L.ai](D2L.ai)

Dive Into Deep Learning is an excellent text on deep learning and deserves attention from anyone who wants to learn why deep learning has ignited the AI revolution – the most powerful technology force of our time.

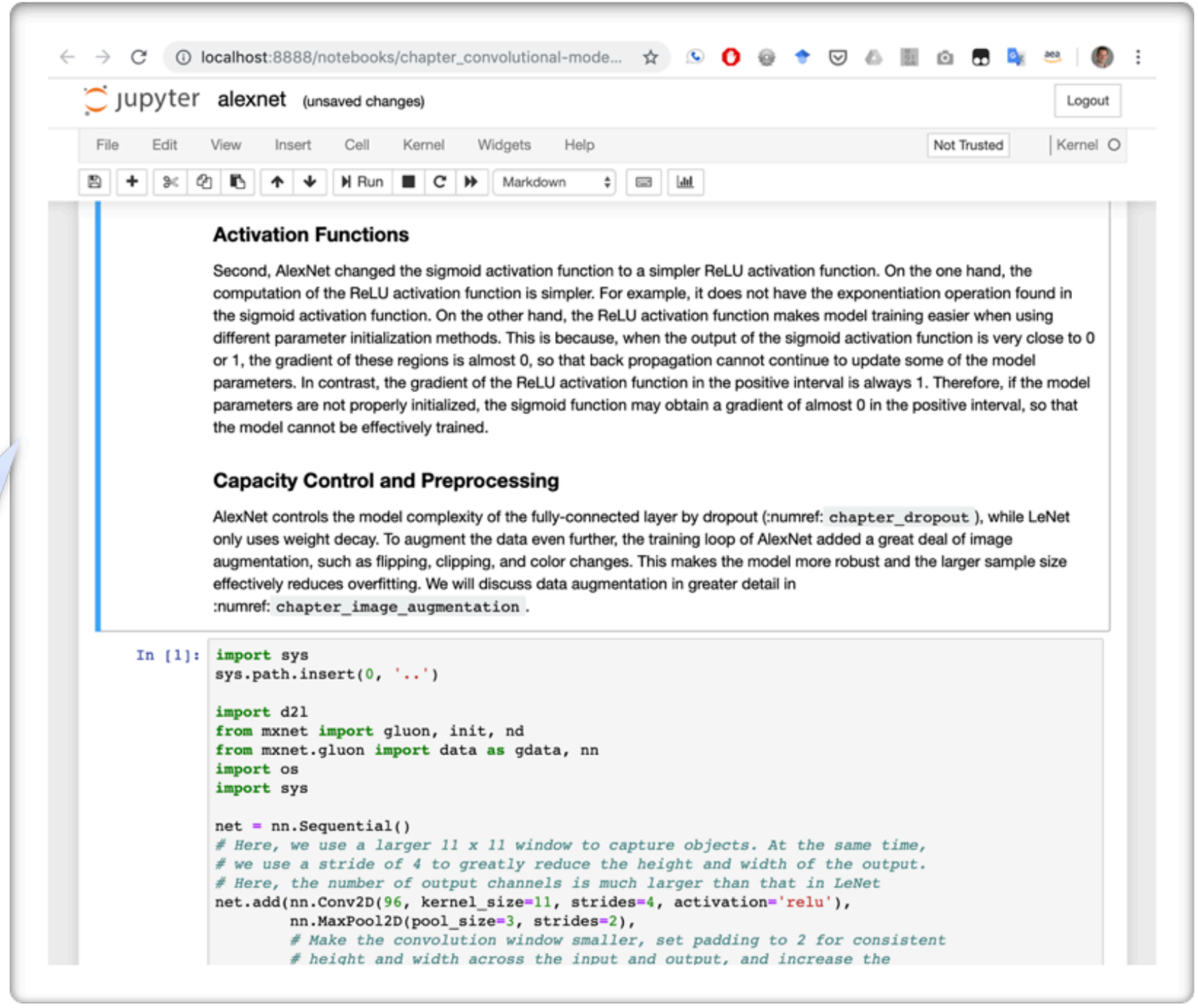
— Jensen Huang,
President and CEO of Nvidia

D2L

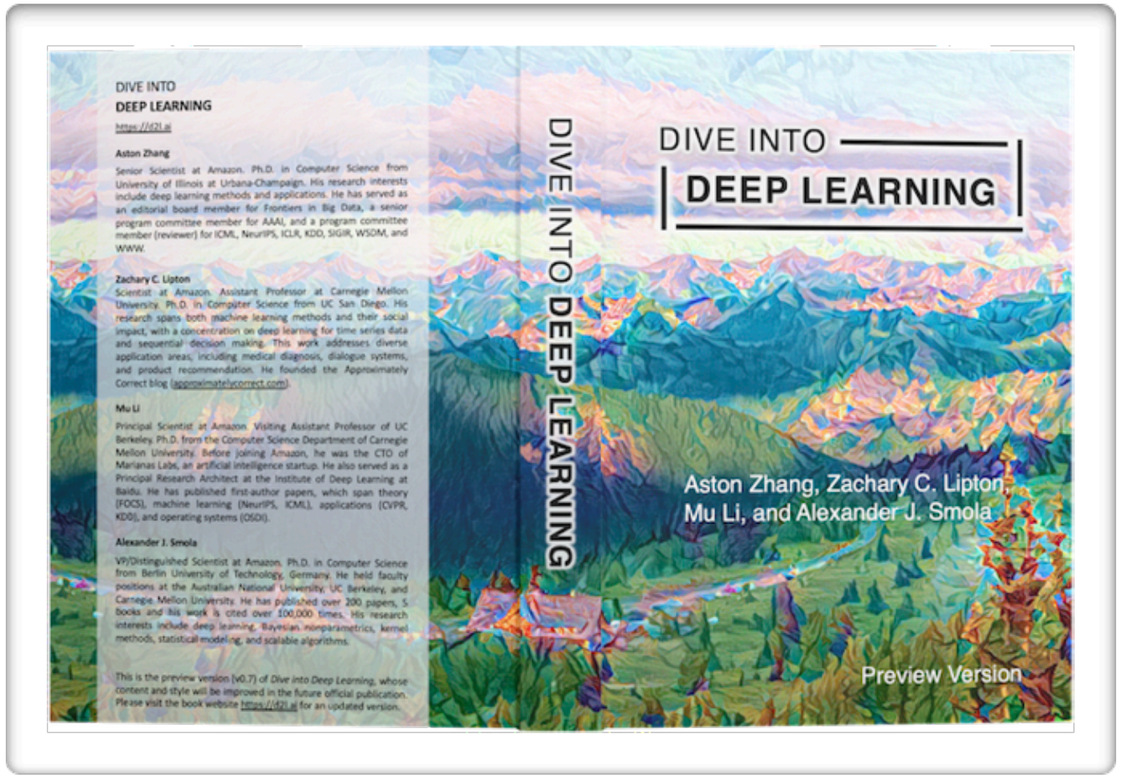


Mobile friendly

Jupyter Notebook



Gluon Ecosystem



GluonCV
Computer Vision

GluonNLP
Natural Language

GluonTS
Time Series Prediction

DGL
Deep Learning on Graphs

TVM
Deep Learning Compiler

MXNet
Imperative & Symbolic

Questions ?

Tim O'Brien (AWS)
Sr. Solutions Architect AI/ML
tpobrien@amazon.com

Rachel Hu (AWS)
Applied Scientist
rlhu@amazon.com

Wen-ming Ye (AWS)
Sr. Solutions Architect AI/ML
wye@amazon.com