



KDD 2019 Hands-On Workshop

Put Deep Learning to Work:
A Practical Introduction on AWS



Wenming Ye,
Sr. Solution Architect

+



Miro Enev,
Sr. Solution Architect

AGENDA



- 1. Account Setup**
- 2. Intro to AWS Services** [SageMaker, Rekognition, Comprehend]
- 3. Intro to DL** [DL Architectures, Demo Links, Types of Learning]
- 4. Intro to hands-on-labs**
 1. Object Detection, Single Shot Detector (SSD)
 2. GluonNLP: BERT fine-tuning for sentiment analysis
 3. Anomaly Detection in Time-Series Data

Logistics

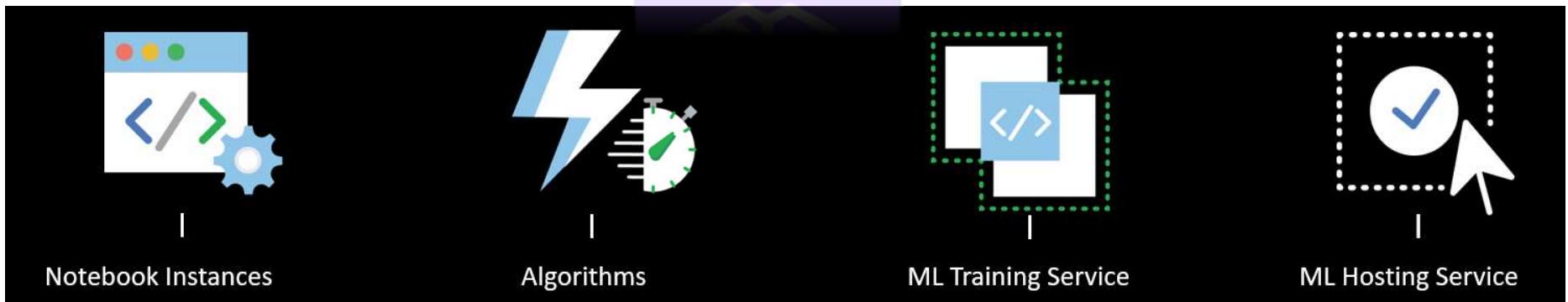
- **Get Lab Instructions:** <https://bit.ly/2YsxhuH>
- **Create** an AWS Account
 - **Add** Credit Security Code to your Account
- **Join** Chime:
 - **Paste** account number into Chime Window
 - **Wait** for confirmation for white listing
- **Ask** TA for help!
 - Leo, Haibing, Zha, Wen-ming, Ginni

AWS Services

▼ All services

- Compute
 - EC2
 - Lightsail
 - ECR
 - ECS
 - EKS
 - Lambda
 - Batch
 - Elastic Beanstalk
 - Serverless Application Repository
- Storage
 - S3
 - EFS
 - FSx
 - S3 Glacier
 - Storage Gateway
 - AWS Backup
- Database
 - RDS
 - DynamoDB
 - ElastiCache
 - Neptune
 - Amazon Redshift
 - Amazon QLDB
 - Amazon DocumentDB
- Migration & Transfer
 - AWS Migration Hub
 - Application Discovery Service
 - Database Migration Service
 - Server Migration Service
 - AWS Transfer for SFTP
 - Snowball
 - DataSync
- Networking & Content Delivery
 - VPC
 - CloudFront
 - Route 53
 - API Gateway
 - Direct Connect
 - AWS App Mesh
 - AWS Cloud Map
 - Global Accelerator
- Developer Tools
 - CodeStar
 - CodeCommit
 - CodeBuild
 - CodeDeploy
 - CodePipeline
 - Cloud9
 - X-Ray
- Robotics
 - AWS RoboMaker
- Blockchain
 - Amazon Managed Blockchain
- Satellite
 - Ground Station
- Management & Governance
 - AWS Organizations
 - CloudWatch
 - AWS Auto Scaling
 - CloudFormation
 - CloudTrail
 - Config
 - OpsWorks
 - Service Catalog
 - Systems Manager
 - Trusted Advisor
 - Managed Services
 - Control Tower
 - AWS License Manager
 - AWS Well-Architected Tool
 - Personal Health Dashboard
 - AWS Chatbot
- Analytics
 - Athena
 - EMR
 - CloudSearch
 - Elasticsearch Service
 - Kinesis
 - QuickSight
 - Data Pipeline
 - AWS Glue
 - MSK
- Security, Identity, & Compliance
 - IAM
 - Resource Access Manager
 - Cognito
 - Secrets Manager
 - GuardDuty
 - Inspector
 - Amazon Macie
 - AWS Single Sign-On
 - Certificate Manager
 - Key Management Service
 - CloudHSM
 - Directory Service
 - WAF & Shield
 - Artifact
 - Security Hub
- Media Services
 - Elastic Transcoder
 - Kinesis Video Streams
 - MediaConnect
 - MediaConvert
 - MediaLive
 - MediaPackage
 - MediaStore
 - MediaTailor
- AWS Cost Management
 - AWS Cost Explorer
 - AWS Budgets
 - AWS Marketplace Subscriptions
- Mobile
 - AWS Amplify
 - Mobile Hub
 - AWS AppSync
 - Device Farm
- AR & VR
 - Amazon Sumerian
- Application Integration
 - Step Functions
 - Amazon EventBridge
 - Amazon MQ
 - Simple Notification Service
 - Simple Queue Service
 - SWF
- Customer Engagement
 - Amazon Connect
 - Pinpoint
 - Simple Email Service
- Business Applications
 - Alexa for Business
 - Amazon Chime
 - WorkMail
- End User Computing
 - WorkSpaces
 - AppStream 2.0
 - WorkDocs
 - WorkLink
- Internet of Things
 - IoT Core
 - Amazon FreeRTOS
 - IoT 1-Click
 - IoT Analytics
 - IoT Device Defender
 - IoT Device Management
 - IoT Events
 - IoT Greengrass
 - IoT SiteWise
 - IoT Things Graph
- Game Development
 - Amazon GameLift

AWS SageMaker



AWS SageMaker Algorithms



All Categories

- Infrastructure Software (33)
- DevOps (33)
- Business Applications (2)
- Machine Learning (243)
- IoT (10)
- Industries (45)

Filters

- Perception Health (31)
- RocketML (23)
- Sensafai (18)
- Cloudwick (11)
- Figure Eight (11)
- Persistent Systems (10)
- Telventow Inc. (8)
- Modjalo (8)
- Intel® AI (8)
- Amazon Web Services (8)

Vendors

- Perception Health
- RocketML
- Sensafai
- Cloudwick
- Figure Eight
- Persistent Systems
- Telventow Inc.
- Modjalo
- Intel® AI
- Amazon Web Services

Show more

Software Pricing Plans

- Free (75)
- Hourly (168)

Software Free Trial

- Free Trial (118)

Delivery Method

- Clear
- Amazon SageMaker (243)

Resource Type

- Clear
- Algorithm (62)
- Model Package (181)

Region

- US East (N. Virginia) (229)
- US East (Ohio) (243)
- US West (Oregon) (236)
- US West (N. California) (229)
- EU (Frankfurt) (211)

Show more

TIBCO® Data Science

Text Similarity Analyzer

★★★★★ (0) | Version v1 | Sold by TIBCO Software Inc.

Engineers word/document features on a corpus with NLP methods, and uses these features to compare new text to the corpus.

Autoencoder for Anomaly Detection

★★★★★ (0) | Version v1 | Sold by TIBCO Software Inc.

Identifies potential anomalies from transaction and/or sensor data with a deep learning autoencoder.

Hospital Readmission

★★★★★ (0) | Version v1 | Sold by TIBCO Software Inc.

Predicts hospital readmission rates from DRG codes, billing and EMR data.

GluonNLP Sentence Generator

★★★★★ (1) | Version 1.0 | Sold by Amazon Web Services

Pre-trained sequence sampler for sentence generation, powered by GluonNLP.

GluonNLP English to German Translation

★★★★★ (0) | Version 1.0 | Sold by Amazon Web Services

Model for english to german translation, powered by GluonNLP.

GluonCV SSD Object Detector

★★★★★ (0) | Version 1.1 | Sold by Amazon Web Services

Single Shot MultiBox Detector (SSD) is a powerful network for fast and accurate object detection, powered by GluonCV.

GluonCV Faster-RCNN Object Detector

★★★★★ (0) | Version 1.1 | Sold by Amazon Web Services

Faster RCNN is a powerful network for accurate object detection, powered by GluonCV.

GluonCV ResNet50 Classifier

★★★★★ (0) | Version 1.0 | Sold by Amazon Web Services

Image feature extraction and ImageNet category prediction using ResNet50-v1d network, provided by GluonCV.

GluonCV DeepLab Semantic Segmentation

★★★★★ (0) | Version 1.0 | Sold by Amazon Web Services

DeepLab is a powerful model for image semantic segmentation, powered by GluonCV.

GluonCV YOLOv3 Object Detector

★★★★★ (1) | Version 1.1 | Sold by Amazon Web Services

YOLOv3 is a powerful network for fast and accurate object detection, powered by GluonCV.

Showing 1 - 10

All Categories

- Infrastructure Software (29)
- DevOps (1)
- Machine Learning (62)
- Industries (2)

Filters

Clear all filters

Vendors

- BucketML (15)
- Sensafai (11)
- Intel® AI (7)
- Outpace Systems (6)
- Peak (5)
- Imperva (4)
- Outpace (4)
- TIBCO Software Inc. (3)
- Dimensional Mechanics (2)
- BellSoft (1)

Show more

Software Pricing Plans

- Free (23)
- Hourly (39)

Software Free Trial

- Free Trial (32)

Delivery Method

- Clear
- Amazon SageMaker (62)

Resource Type

- Clear
- Algorithm (62)
- Model Package (181)

Region

- US East (N. Virginia) (61)
- US East (Ohio) (62)
- US West (Oregon) (61)
- US West (N. California) (61)
- EU (Frankfurt) (61)

Show more

TIBCO® Data Science

Text Similarity Analyzer

★★★★★ (0) | Version v1 | Sold by TIBCO Software Inc.

Engineers word/document features on a corpus with NLP methods, and uses these features to compare new text to the corpus.

Autoencoder for Anomaly Detection

★★★★★ (0) | Version v1 | Sold by TIBCO Software Inc.

Identifies potential anomalies from transaction and/or sensor data with a deep learning autoencoder.

Hospital Readmission

★★★★★ (0) | Version v1 | Sold by TIBCO Software Inc.

Predicts hospital readmission rates from DRG codes, billing and EMR data.

Image Recognition (Trainable Algorithm)

★★★★★ (0) | Version v1 | Sold by Sensafai

Automatic Image Tagging and Recognition (Trainable Algorithm)

Demand Forecasting for Intermittent Data

★★★★★ (0) | Version 0.1 | Sold by Peak

An ensemble demand forecasting model, for intermittent data

PEAK

Free Trial

TRINITI

Cognitive QnA

★★★★★ (0) | Version v1.0beta | Sold by Active.ai

Provides an FAQ data source that you can query from your bot or application

Outpace

Implicit BPR

★★★★★ (0) | Version 0.9.6 | Sold by Outpace Systems

A recommender system for implicit feedback datasets using Bayesian Personalized Ranking.

rocketML

Free Trial

RocketML Text Latent Semantic Analysis

★★★★★ (0) | Version 0.1 | Sold by RocketML

Semantic analysis and indexing on a list of documents

sensafai

Free Trial

Facial Recognition Algorithm for Images

★★★★★ (0) | Version v1 | Sold by Sensafai

Face Recognition/Identification in Images

rocketML

Free Trial

RocketML Sparse RandomForest Classification

★★★★★ (0) | Version 0.1 | Sold by RocketML

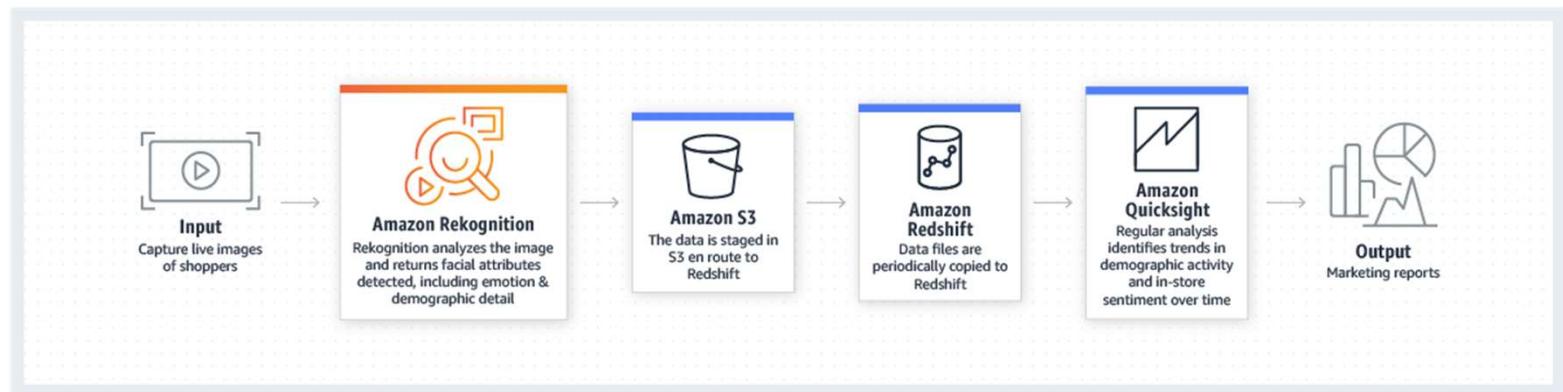
Classification on LIBSVM data type using Random Forests

Showing 1 - 10

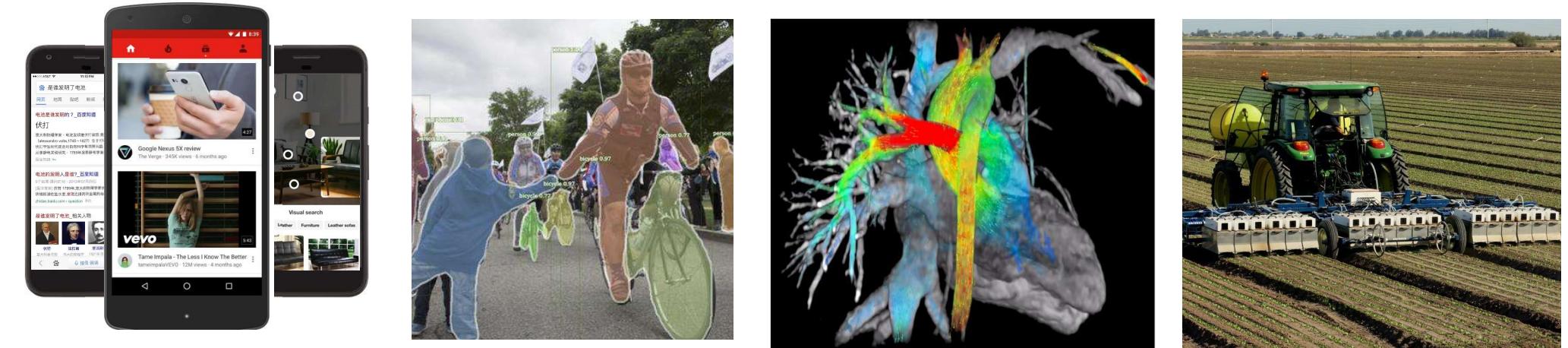
1 2 3 4 5 6 7 ►

AWS ML/AI Services Console Walkthrough

- Machine Learning**
 - Amazon SageMaker
 - Amazon Comprehend
 - AWS DeepLens
 - Amazon Lex
 - Machine Learning
 - Amazon Polly
 - Rekognition
 - Amazon Transcribe
 - Amazon Translate
 - Amazon Personalize
 - Amazon Forecast
 - Amazon Textract
 - AWS DeepRacer



Impact of ML/DL



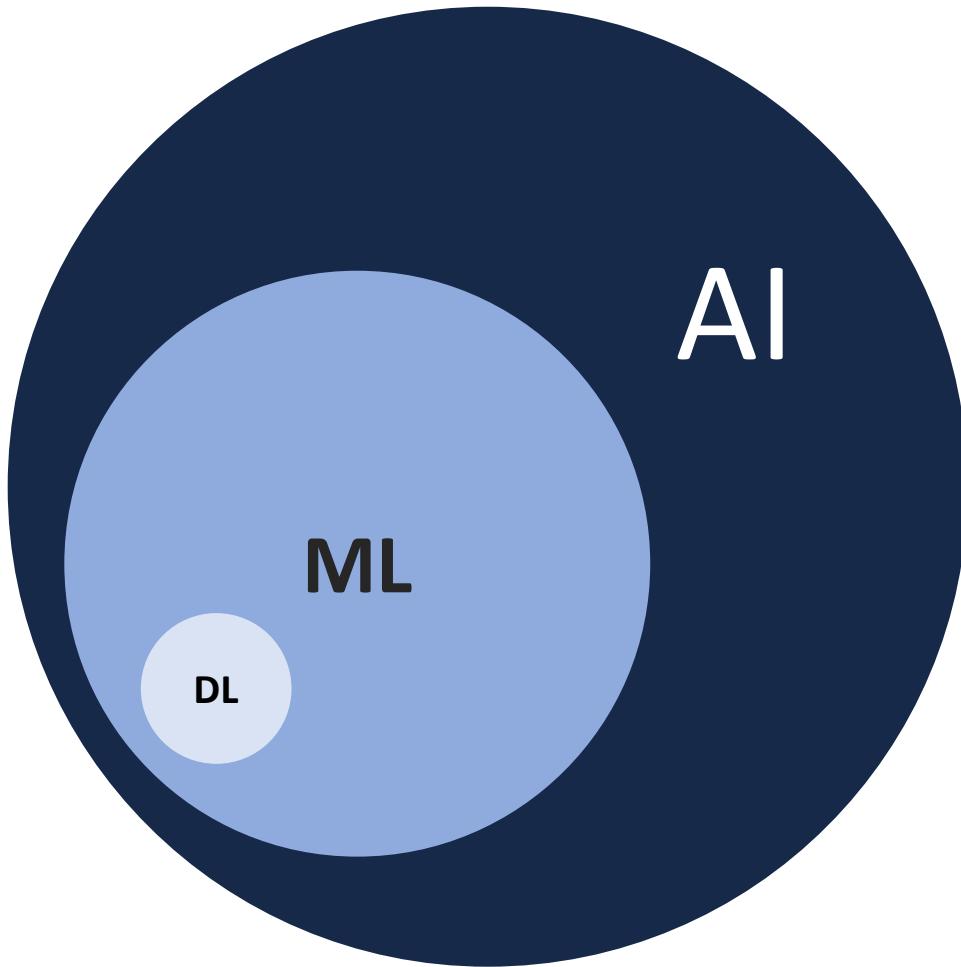
1 – Natural Language

2 – Visual Perception / Robotics

3 – Anomaly Detection

4 – Medicine [Radiology, Drug Discovery, Remote/Robo surgery]

DL in Context



A diagram titled "ML TRIBES" enclosed in a large blue circle. Inside are six smaller colored circles (yellow, blue, green, brown, purple, and dark grey) each with a small white tent icon below it. In the center, five cartoon scientist icons stand in a row. The yellow circle is labeled "SYMBOLIST", the blue "ANALOGIST", the green "BAYESIAN", the brown "EVOLUTIONIST", and the purple "CONNECTIONIST".

SYMBOLIST

ANALOGIST

BOOSTER

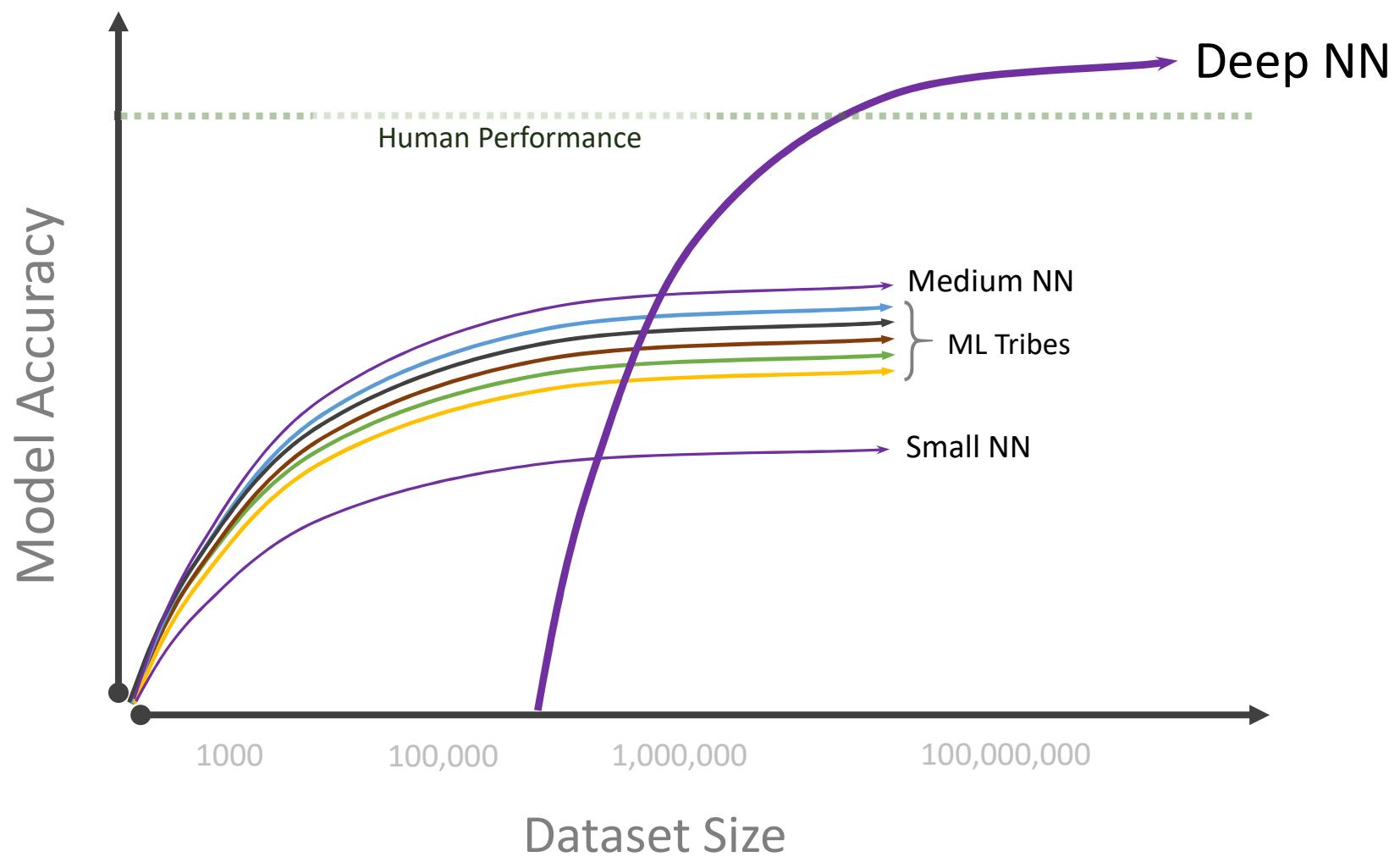
BAYESIAN

ML TRIBES

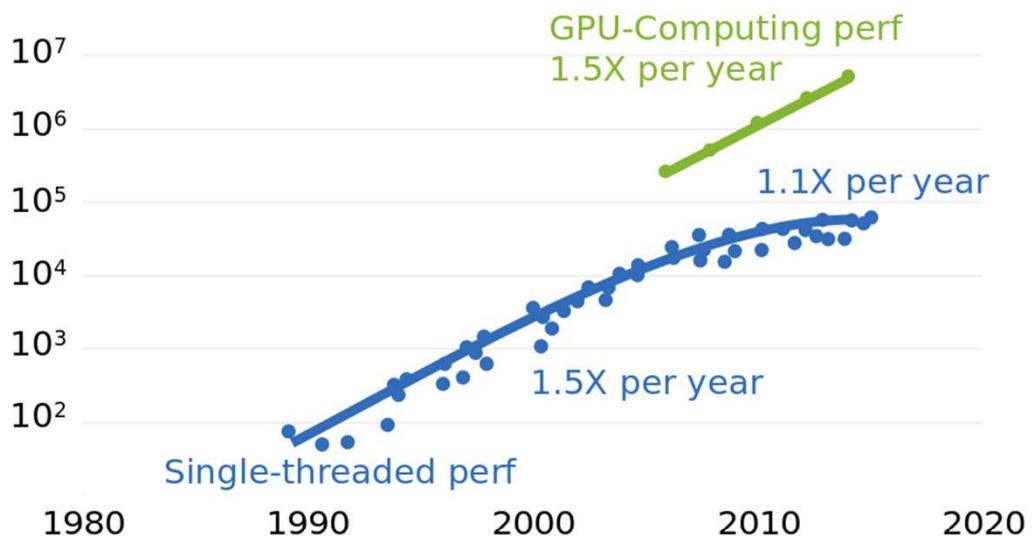
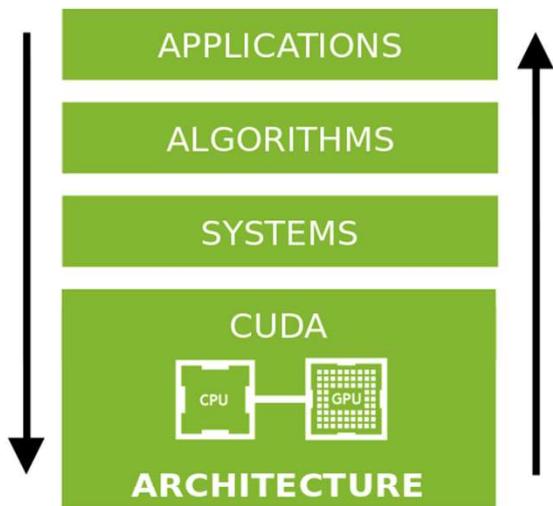
CONNECTIONIST

EVOLUTIONIST

Learning at Scale



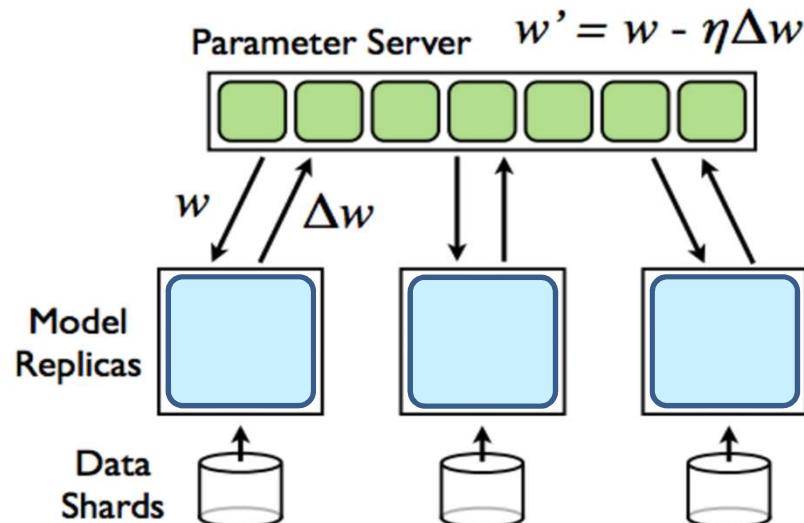
GPU Computing Model



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten New plot and data collected for 2010-2015 by K. Rupp

DL Compute = AWS + NVIDIA

Data Parallel Training [SageMaker]

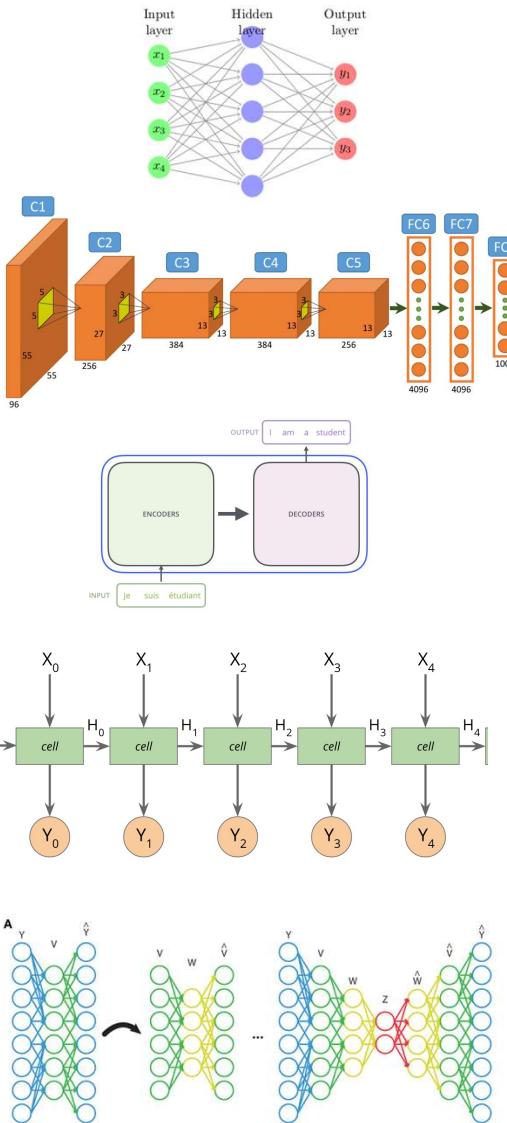
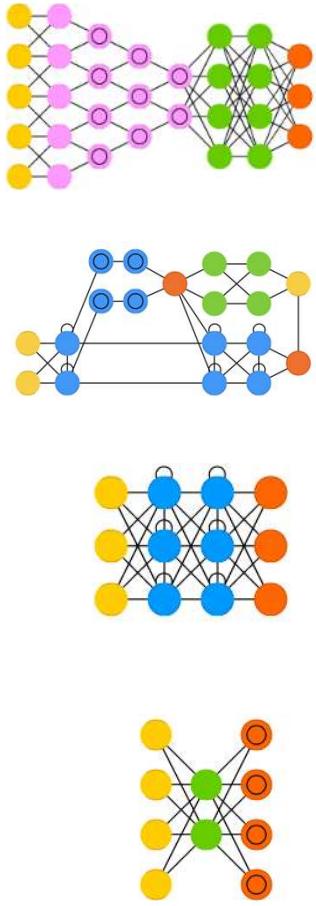


Volta100 [ec2.p3]

5,120 CUDA cores [640 Tensor cores]

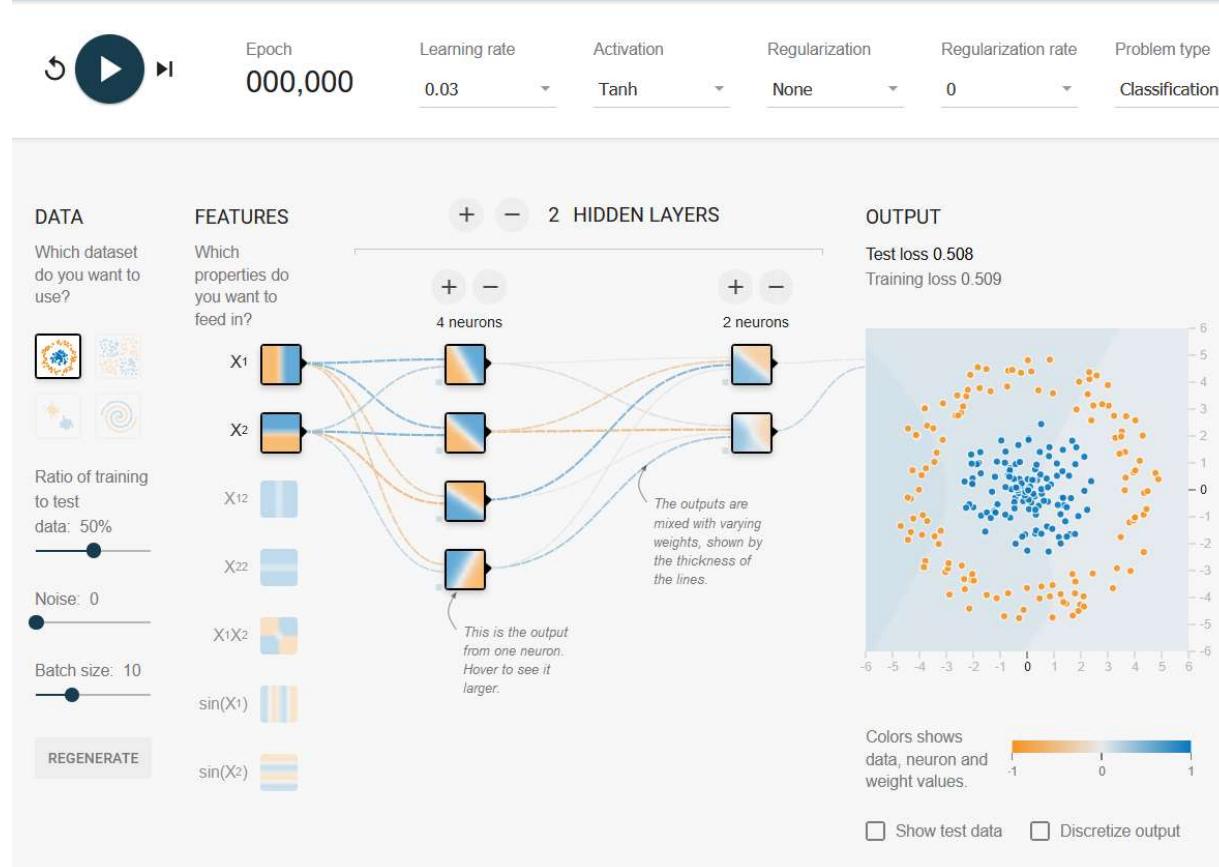


Key DL Architectures



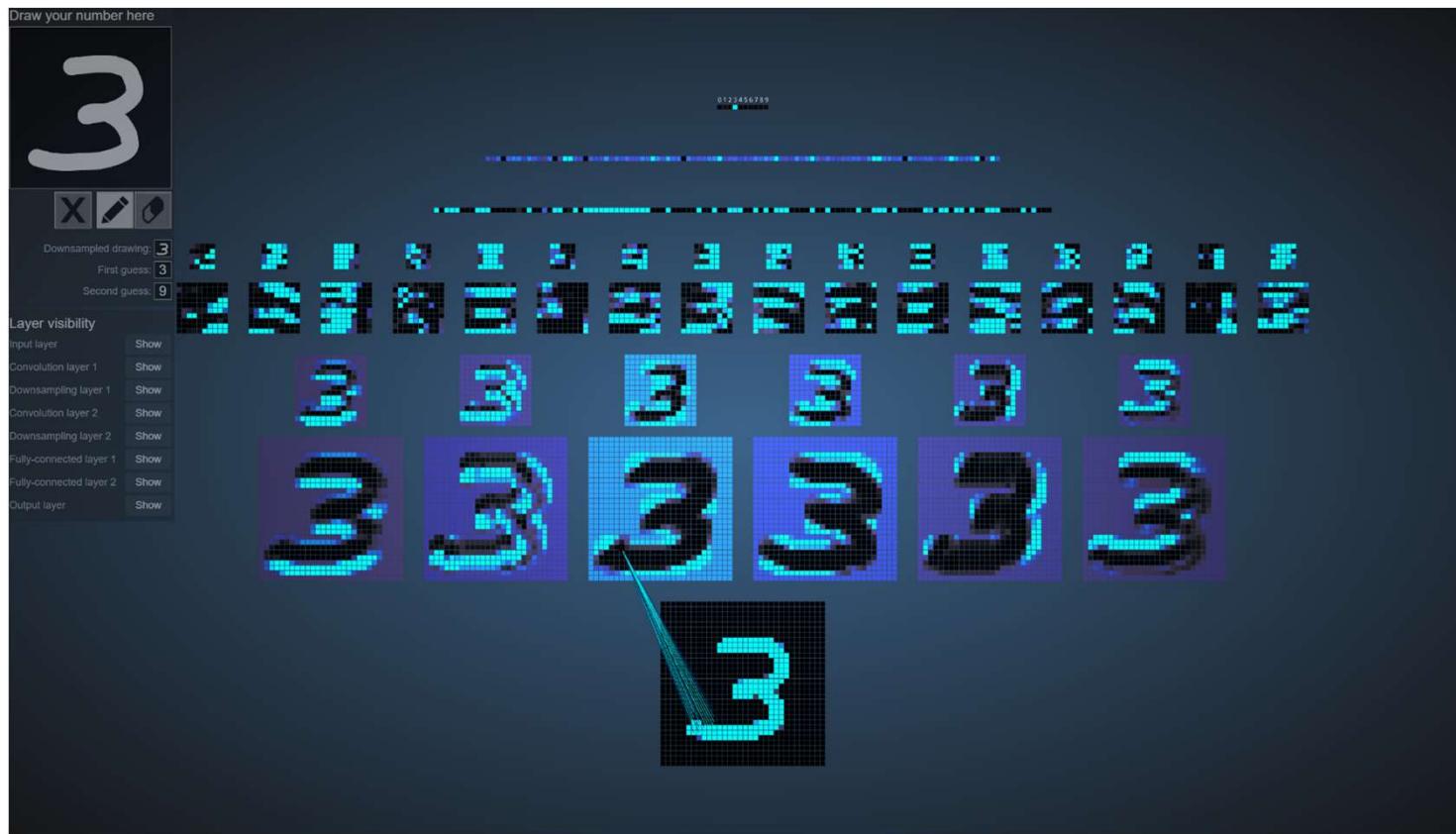
- **MLP [demo]**
- **CNN [Lab 1] [demo]**
vision & speech perception
- **Attention/Transformer [Lab 2]**
- **RNN**
- **Autoencoder [Lab 3]**
anomaly detection, model RL

MLP Demo



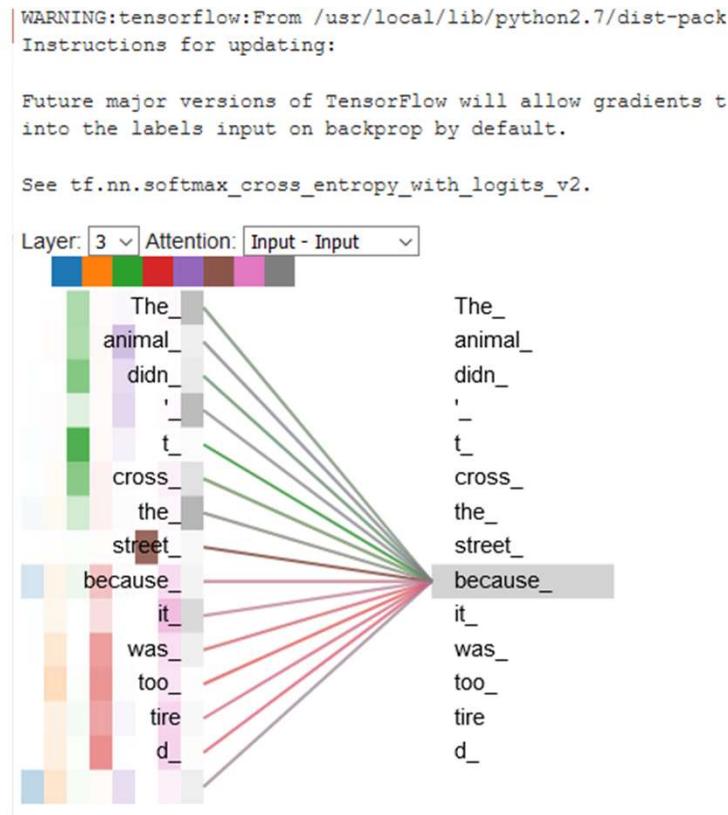
<https://playground.tensorflow.org>

CNN Demo



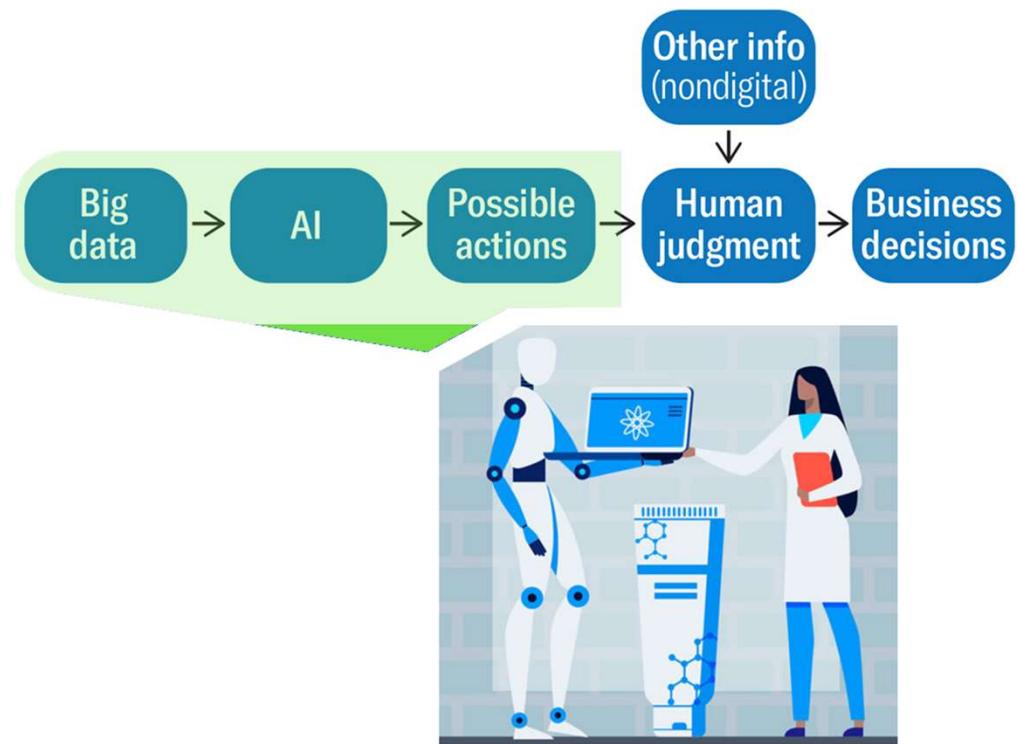
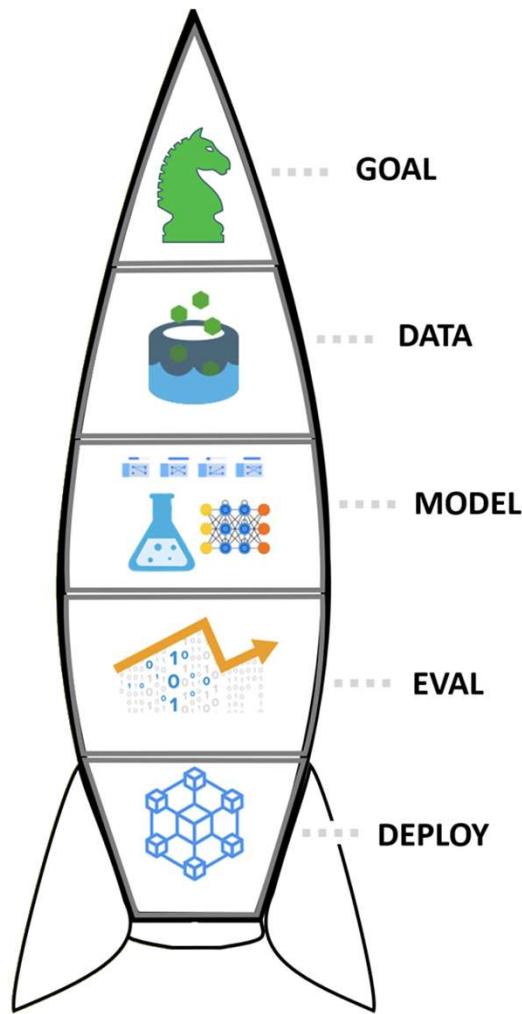
<http://scs.ryerson.ca/~aharley/vis/>

Transformer Demo

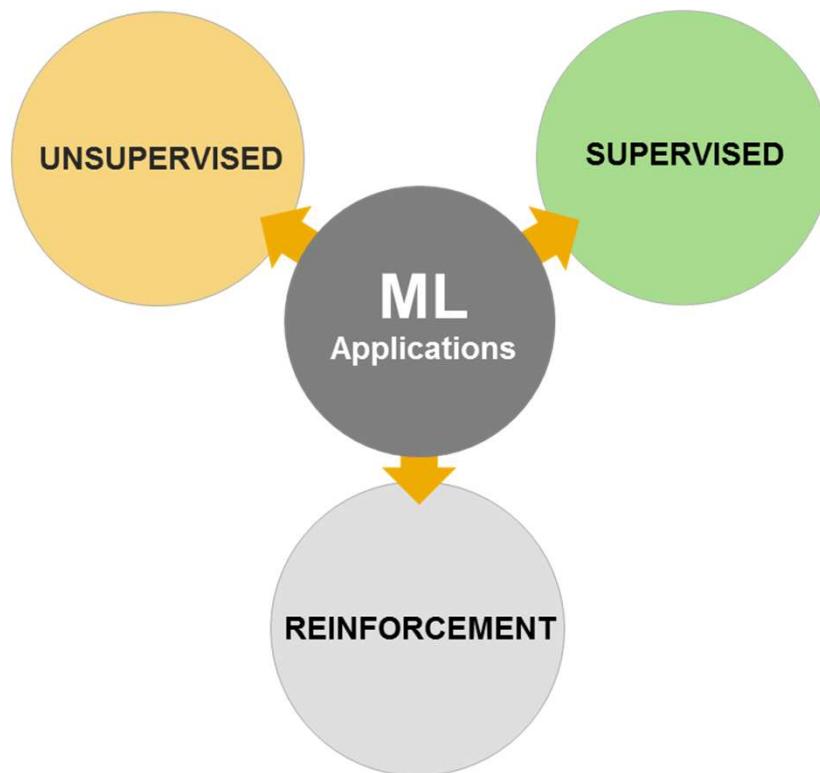


[English to German Transformer \[Tensor2Tensor Demo \]](#)

Applying ML

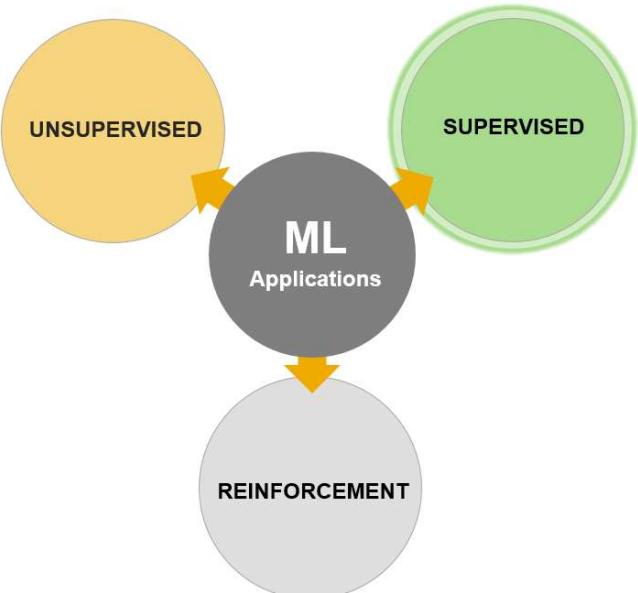


Types of Learning



Supervised Learning

- ~100,000s of examples
 - Each example consists of
 - Data [Vector]
 - Label(s)

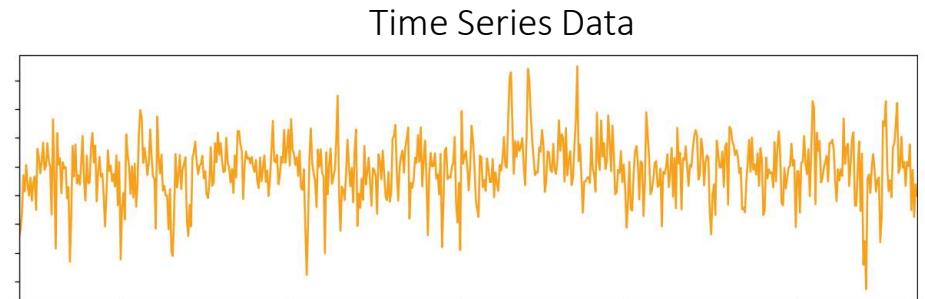
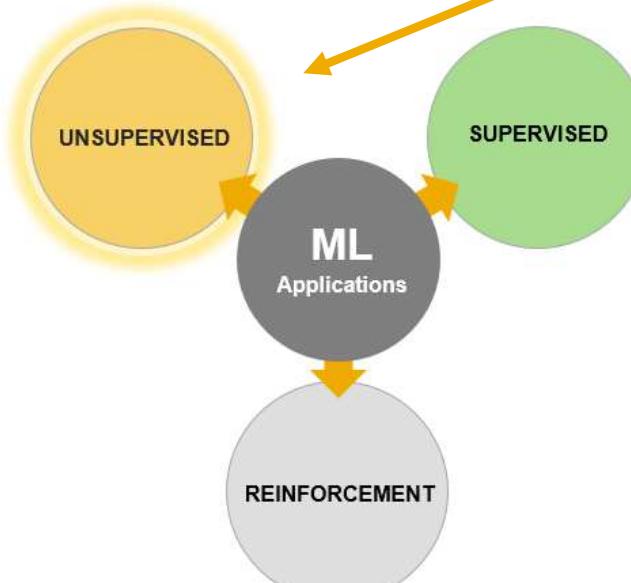


sample #1

label #1: 8, 'eight', [0, 0, 0, 0, 0, 0, 0, 0, 1, 0]

Unsupervised Learning

- In unsupervised learning, no labels are available
 - Typically larger datasets (millions)
 - Each example consists of
 - Data [Vector]
 - No Labels



A sequence of DNA bases (A, T, G, C) arranged in a grid. The sequence starts with A G A T A A G A A G T G T T G T G G A A A A G T A T G T T G and continues for several lines.

```
A G A T A A G A A G T G T T G T G G A A A A G T A T G T T G  
T A T T T A G A A G T A A G T G A T G G T T A T T T G G A G  
T A A G T A G G T A G A A T T A A G A G G T T A T A G G G A  
A A G G G A A T T G G T G G A A G G T G A T T T A A T G A A  
G A A T G T T T G T G A T T A G G A T A G T A T G T A A G  
A T A G A G A T T G G A A T G G G G G A T G A T G A T T A  
A T G A A G T T T A G G A G G T G T G A T G A A T G A G A A G  
T T A G G G A A G G A T T A A T G T G T A A G G G T A T T A  
G G A A A A T G G A A A A A T T A T G G G T T T T G T G G  
A G G G A G T G T A A A G T G G T G T G G A A A A G A T T G
```

A stack of three books with a yellow ribbon bookmark, representing text data.

WIKIPEDIA		
The Free Encyclopedia		
English	5 802 000+ articles	日本語
Deutsch	2 270 000+ Artikel	1138 000+記事
Русский	1 527 000+ статьи	Español
Italiano	1 503 000+ voci	2 080 000+ artículos
Português	1 016 000+ artigos	中文
Polski	1 319 000+ haset	한국어

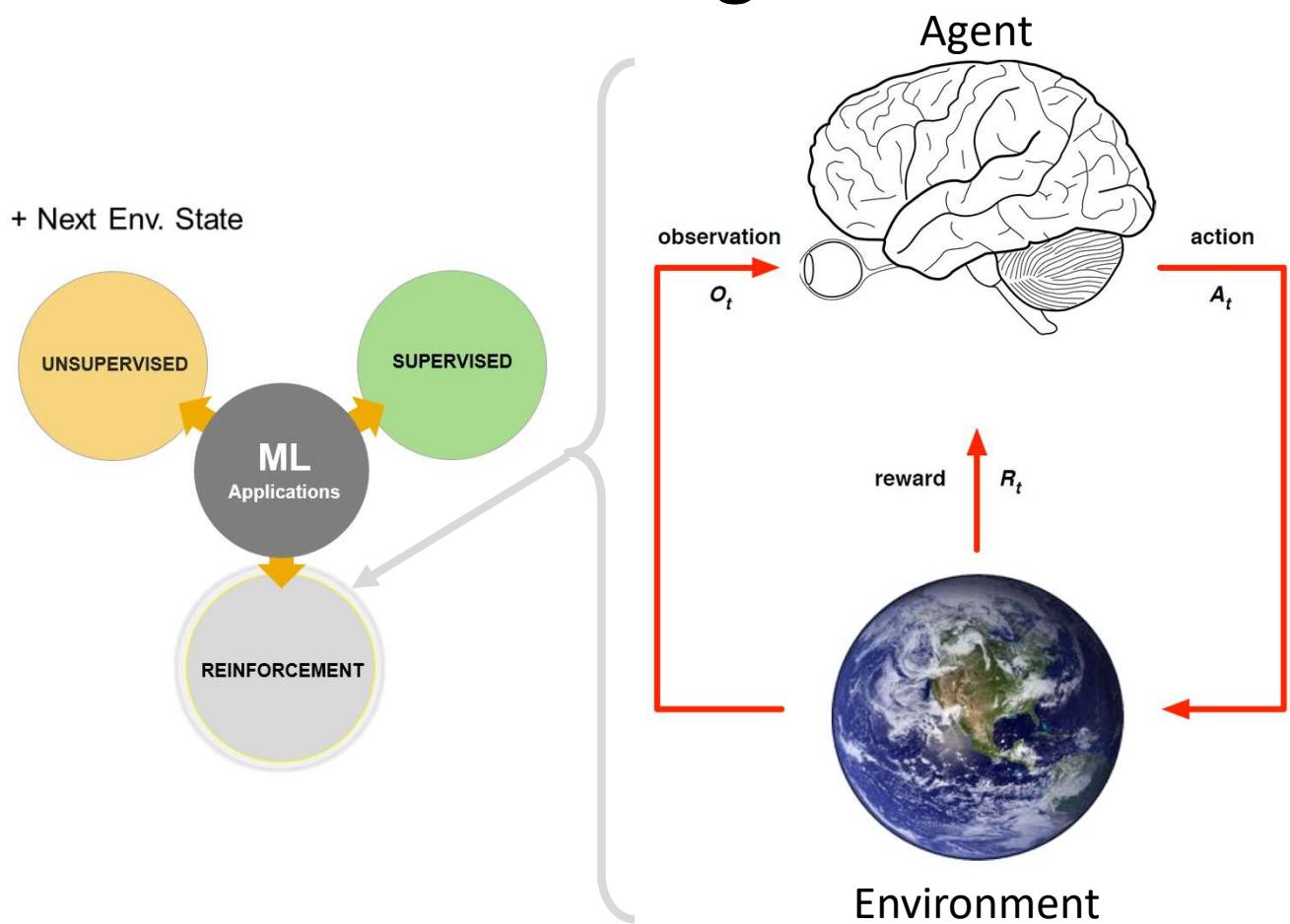
Reinforcement Learning

- **RL: Dataset built with experience**

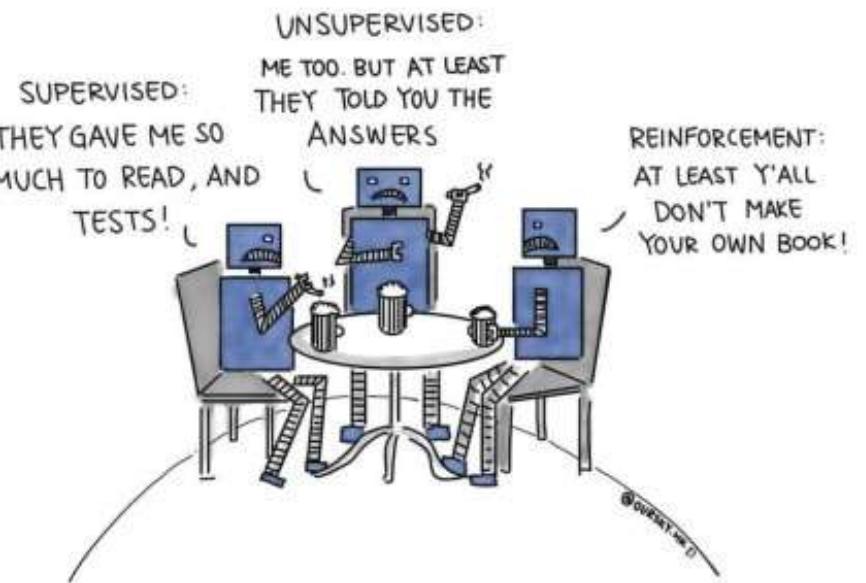
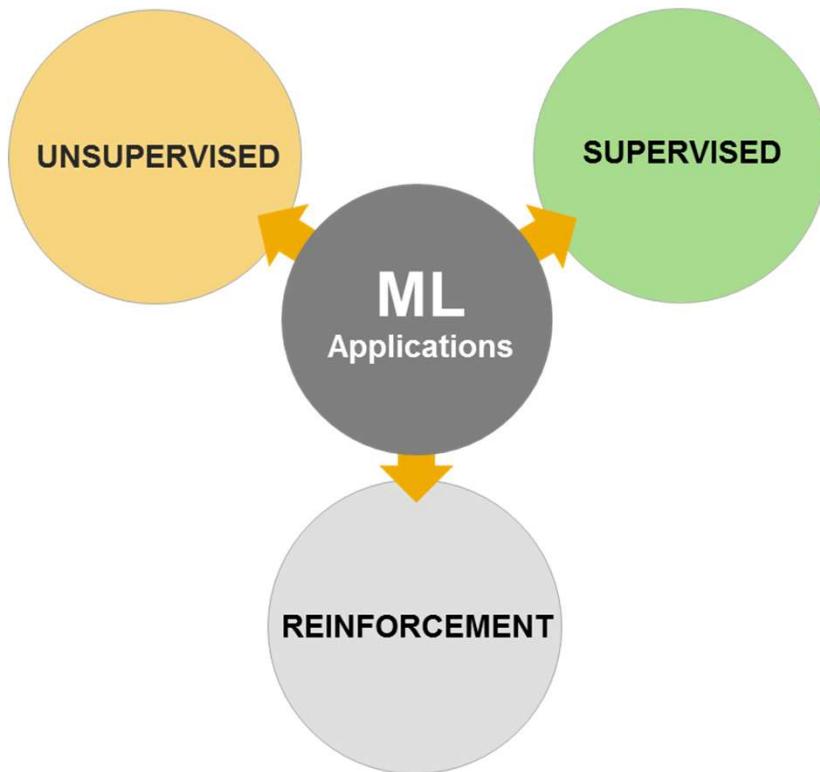
- Experience = Current Env. State + Action + Next Env. State

- **RLRL Feedback Loop**

- At each step the **agent**
 - Executes action: A_t
 - Receives observation: O_t
 - Receives reward: R_t
 - The **environment**
 - Receives action: A_t
 - Emits observation: O_{t+1}
 - Emits reward: R_{t+1}



Three Types of Learning



Lab 1

Problem/Objective: Object Detection

AWS Service: Rekognition

DL Architecture Overview: CNN Demo [[3D](#), [flat](#)]

Relevant Applications: Robotics, Retail, AV

SOTA : [paperswithcode.com/sota]

Lab 2

Problem/Objective: fine-tuning BERT for Question Answering

AWS Service: Comprehend

DL Architecture Overview: Illustrated BERT

Relevant Applications: Virtual Assistant/Chatbot

SOTA : [paperswithcode.com/sota]

Lab 3

Problem/Objective: Anomaly Detection

AWS Service: SageMaker, NEO

DL Architecture Overview: Autoencoder

Relevant Applications: Time-Series

Lab 1

Problem/Objective: Object Detection

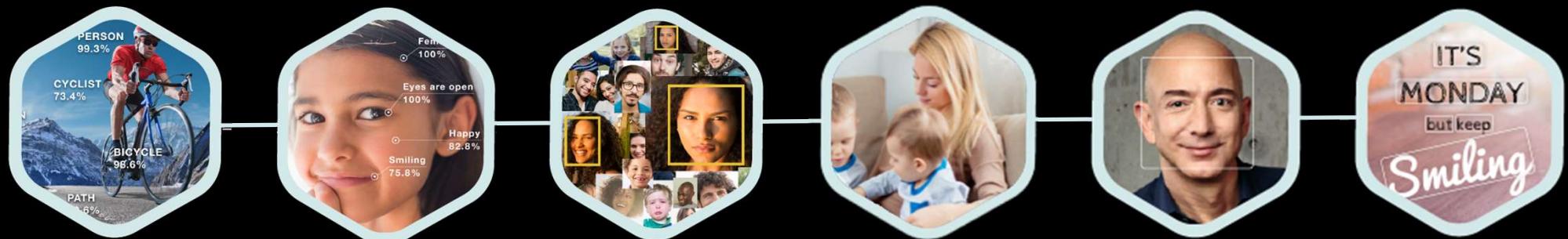
AWS Service: Rekognition

DL Architecture Overview: CNN Demo [[3D](#), [flat](#)]

Relevant Applications: Robotics, Retail, AV

SOTA : [paperswithcode.com/sota]

Amazon Rekognition Image



Object and Scene
Detection

Facial
Analysis

Face
Recognition

Unsafe Image
Detection

Celebrity
Recognition

Text in Image

Amazon Rekognition Video



Object and Activity
Detection

Person
Tracking

Face
Recognition

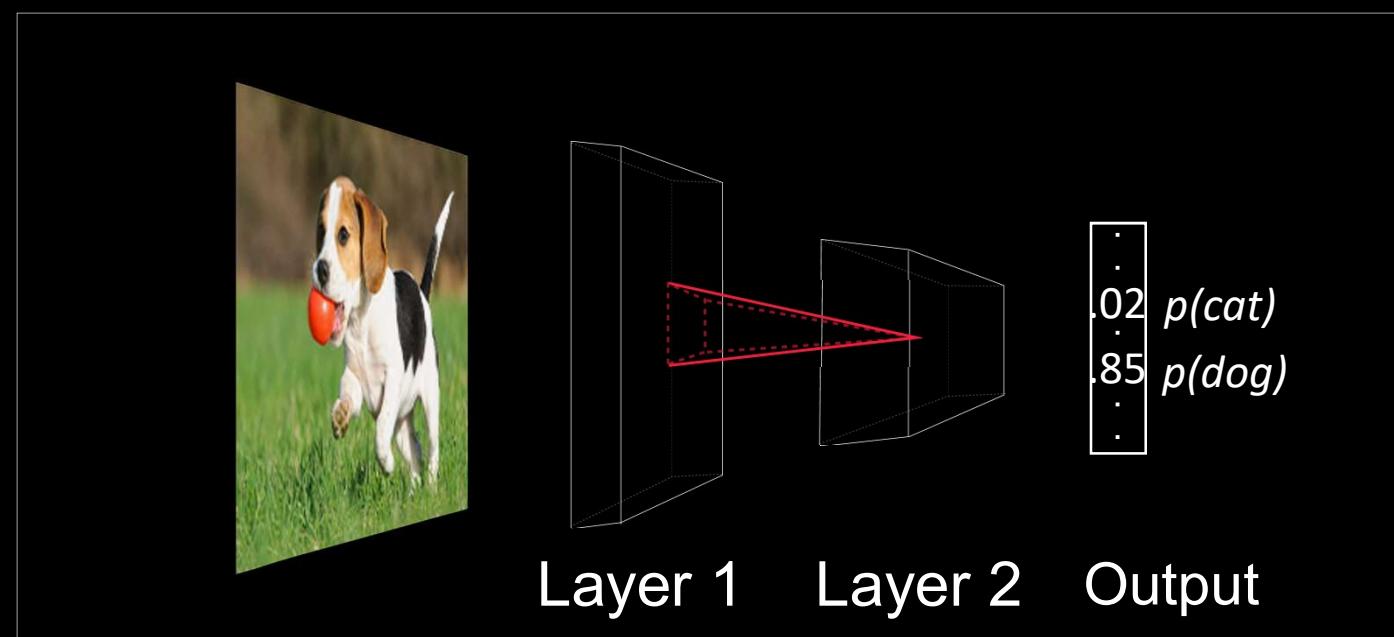
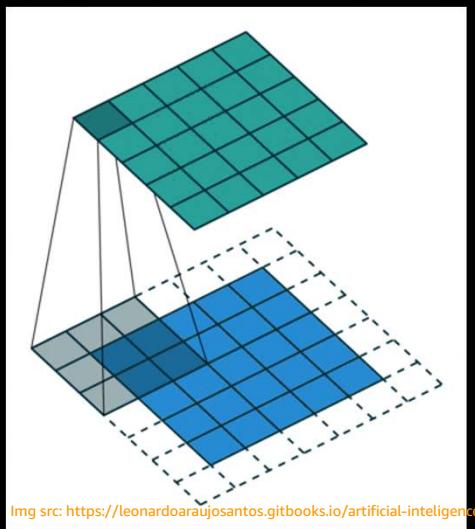
Real-time Live Stream

Content Moderation

Celebrity Recognition

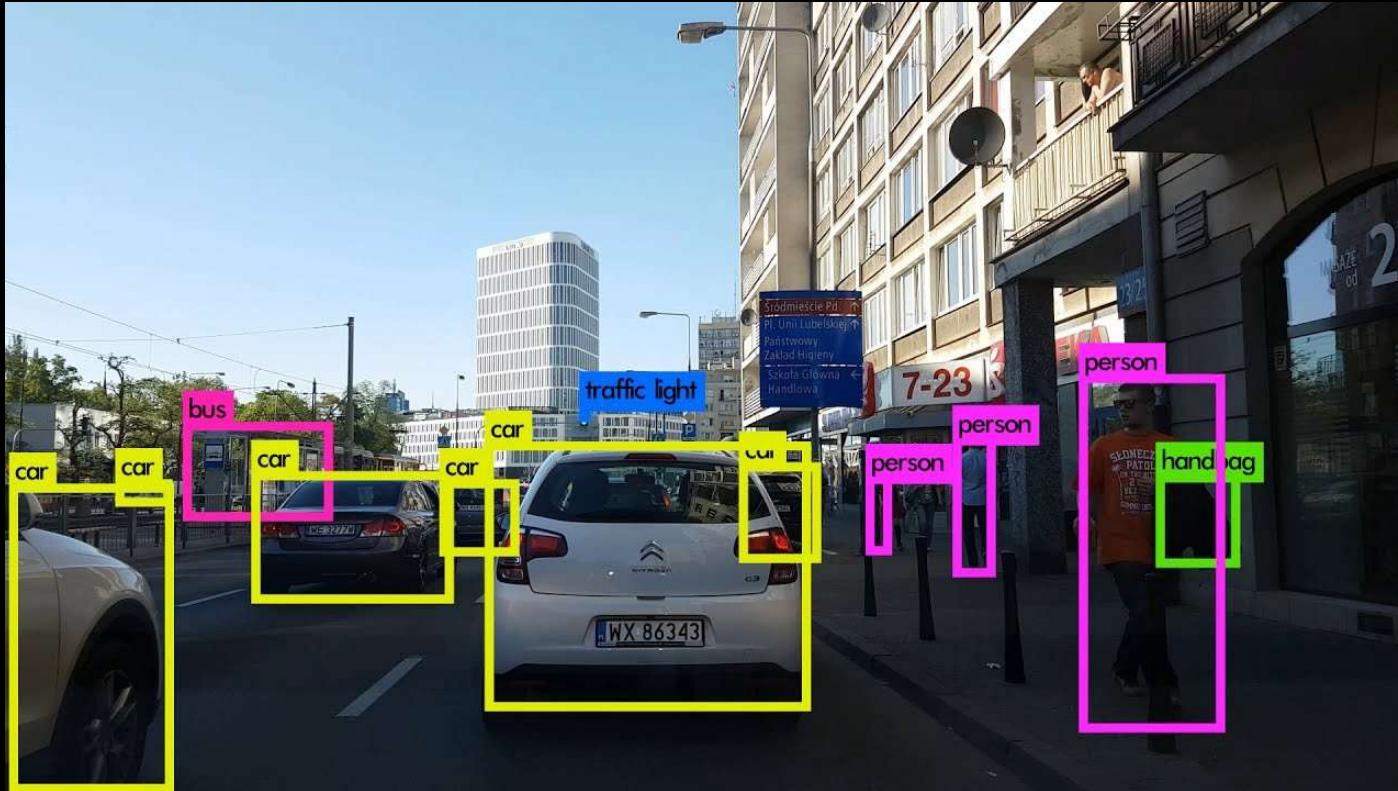
Deep Learning in Computer Vision

Convolutional neural network

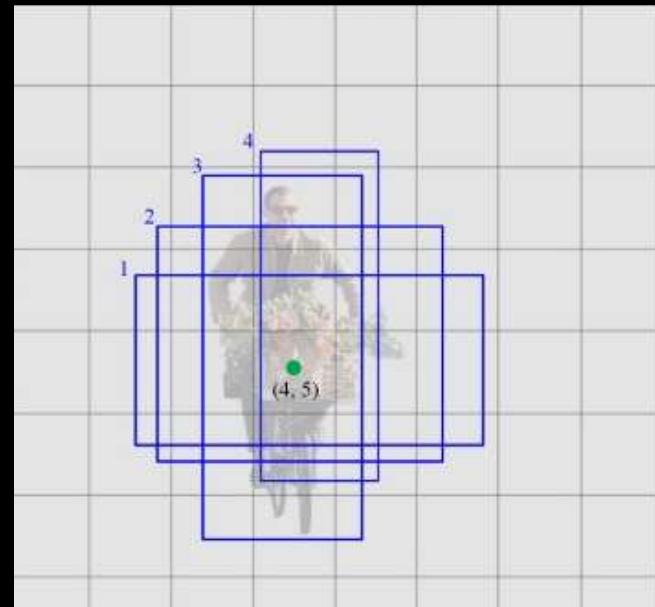


Explore spatial information with convolution layers

Object Detection



Single Shot Detector



Lab 2

Problem/Objective: fine-tuning BERT for Question Answering

AWS Service: Comprehend

DL Architecture Overview: Illustrated BERT

Relevant Applications: Virtual Assistant/Chatbot

SOTA : [[paperswithcode.com/sota](https://paperswithcode.com/sota/question-answering)]

https://github.com/astonzhang/KDD19-tutorial/blob/master/07_bert_app/bert.ipynb

2018 –NLP’s Big Bang!

Leaderboard

SQuAD2.0 tests the ability of a system to not only answer reading comprehension questions, but also abstain when presented with a question that cannot be answered based on the provided paragraph. How will your system compare to humans on this task?

Rank	Model	EM	F1
	Human Performance Stanford University (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Jul 22, 2019	XLNet + DAAF + Verifier (ensemble) PINGAN Omni-Sinicic	88.592	90.859
2 Jul 19, 2019	XLNet + SG-Net Verifier (ensemble) Shanghai Jiao Tong University & CloudWalk	88.050	90.645
3 Jul 23, 2019	XLNet + SG-Net Verifier (single model) Shanghai Jiao Tong University & CloudWalk	87.046	89.899
3 Mar 20, 2019	BERT + DAE + AoA (ensemble) Joint Laboratory of HIT and iFLYTEK Research	87.147	89.474
3 Jul 20, 2019	RoBERTa (single model) Facebook AI	86.820	89.795
4 Mar 15, 2019	BERT + ConvLSTM + MTL + Verifier (ensemble) Layer 6 AI	86.730	89.286

<https://rajpurkar.github.io/SQuAD-explorer/>

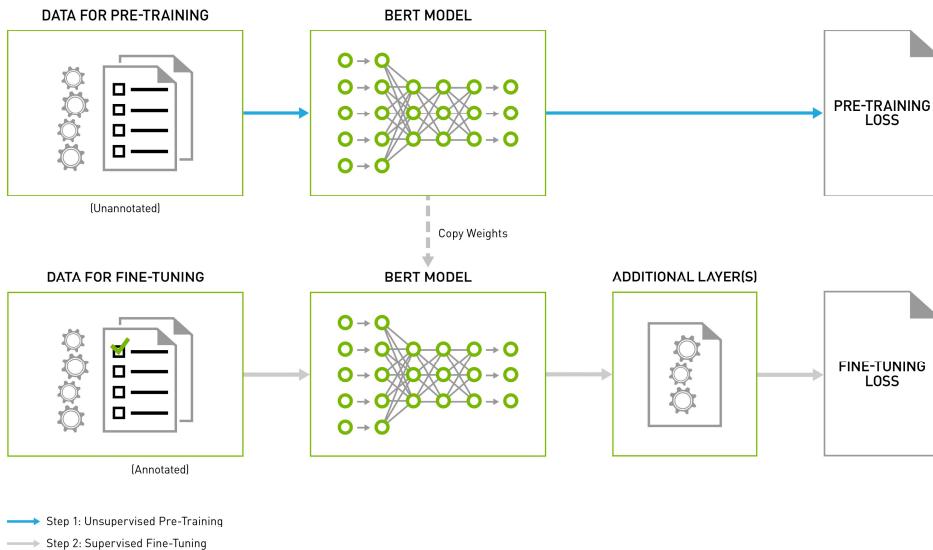
- pre-training representations on massive data
- fine tuning for specific tasks
- matching human performance [?]
- protocol for M2M, M2H

“overall, people are still much better than machines at comprehending the complexity and nuance of language.” - J. Gao, MSR Asia

BERT Pipeline

[pretraining into Q&A]

<https://github.com/NVIDIA/DeepLearningExamples>



- **BERT objective:**

$$\text{total_loss} = \text{masked_lm_loss} + \text{next_sentence_loss}$$

Input: The man went to the [MASK]₁. He bought a [MASK]₂ of milk
Labels: [MASK]₁ = store; [MASK]₂ = gallon

Sentence A = The man went to the store.

Sentence B = He bought a gallon of milk.

Label = IsNextSentence

Sentence A = The man went to the store.

Sentence B = Penguins are flightless.

Label = NotNextSentence

- **Input Question:**

Where do water droplets collide with ice crystals to form precipitation?

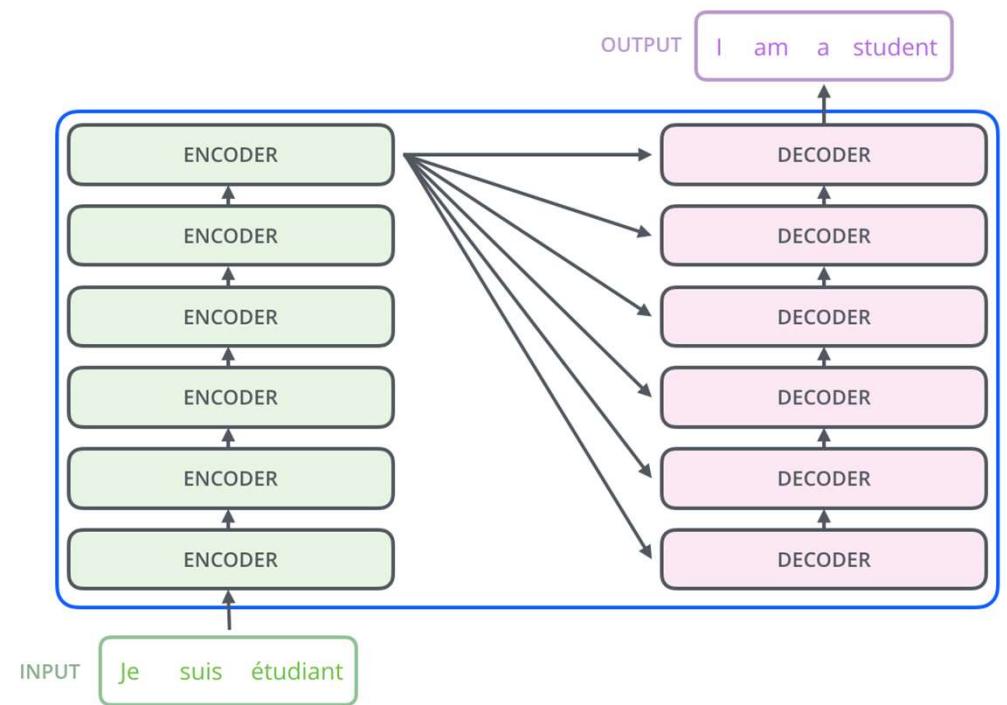
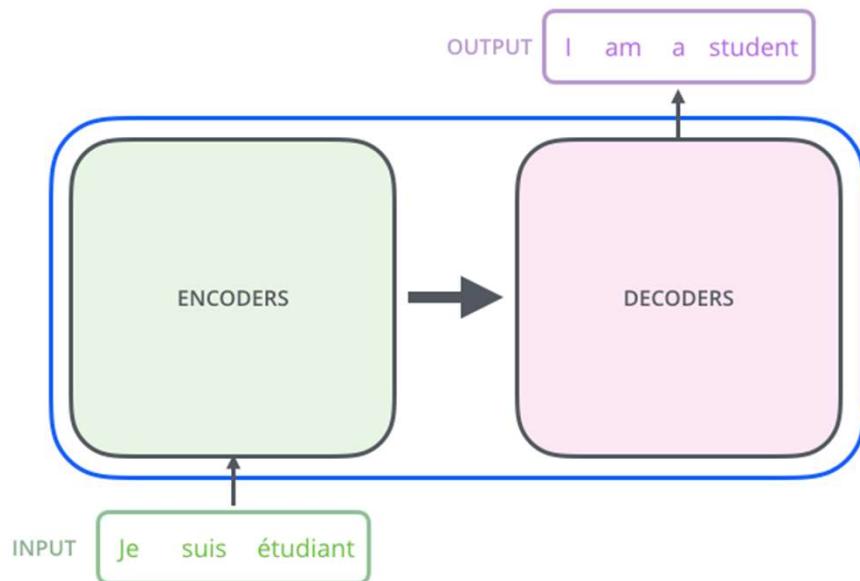
- **Input Paragraph:**

... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. ...

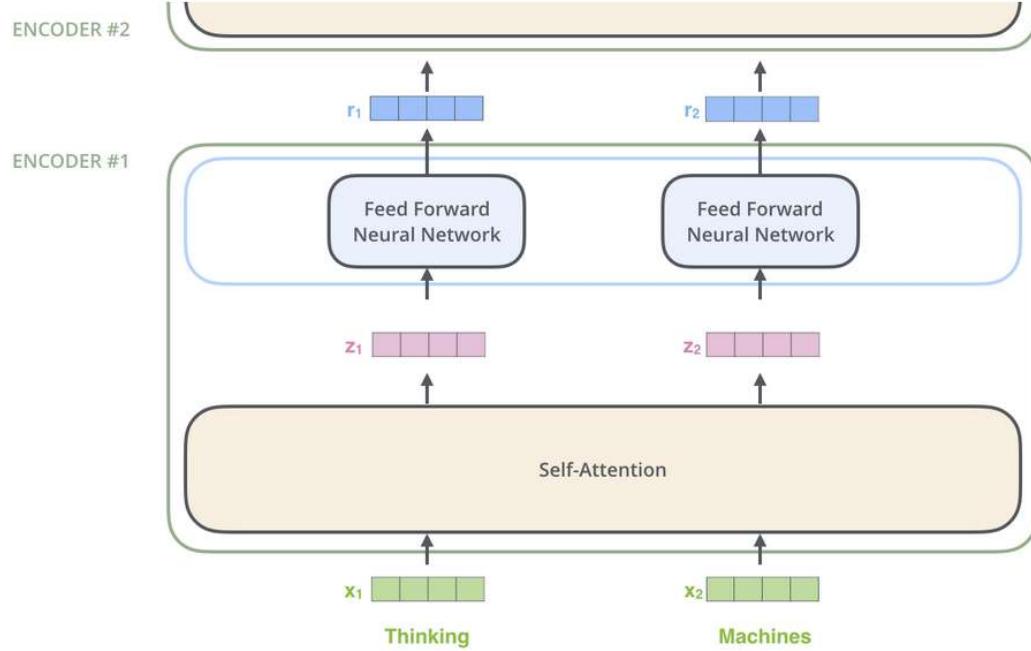
- **Output Answer:**

within a cloud

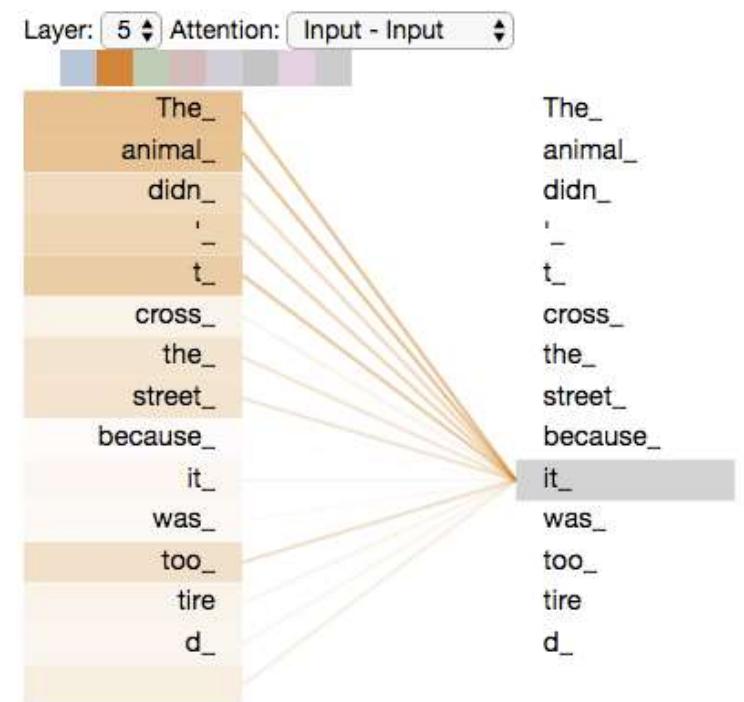
Transformer [Encoder – Decoder]



Self-Attention



The word at each position passes through a self-attention process. Then, they each pass through a feed-forward neural network -- the exact same network with each vector flowing through it separately.

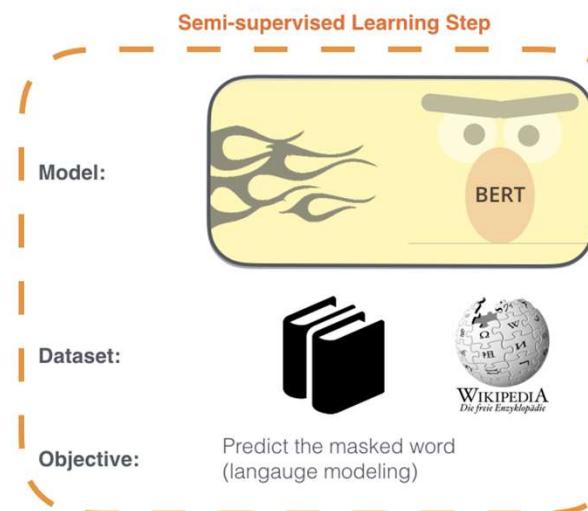


[Interactive Attention Viz](#)

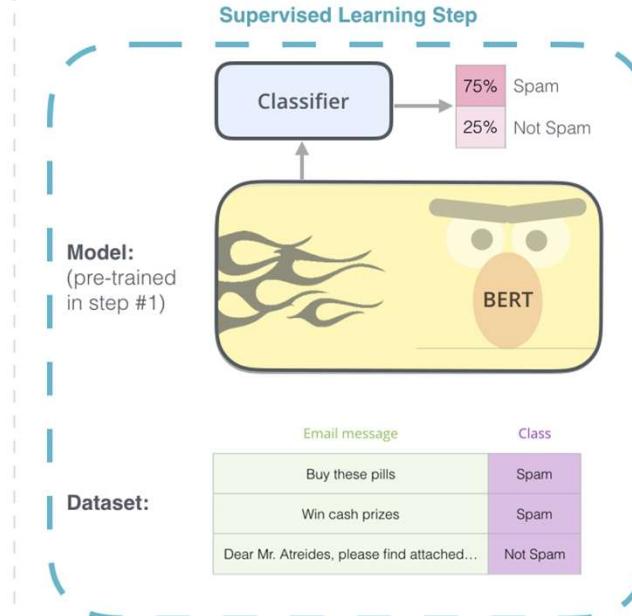
References

1 - Semi-supervised training on large amounts of text (books, wikipedia..etc).

The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.

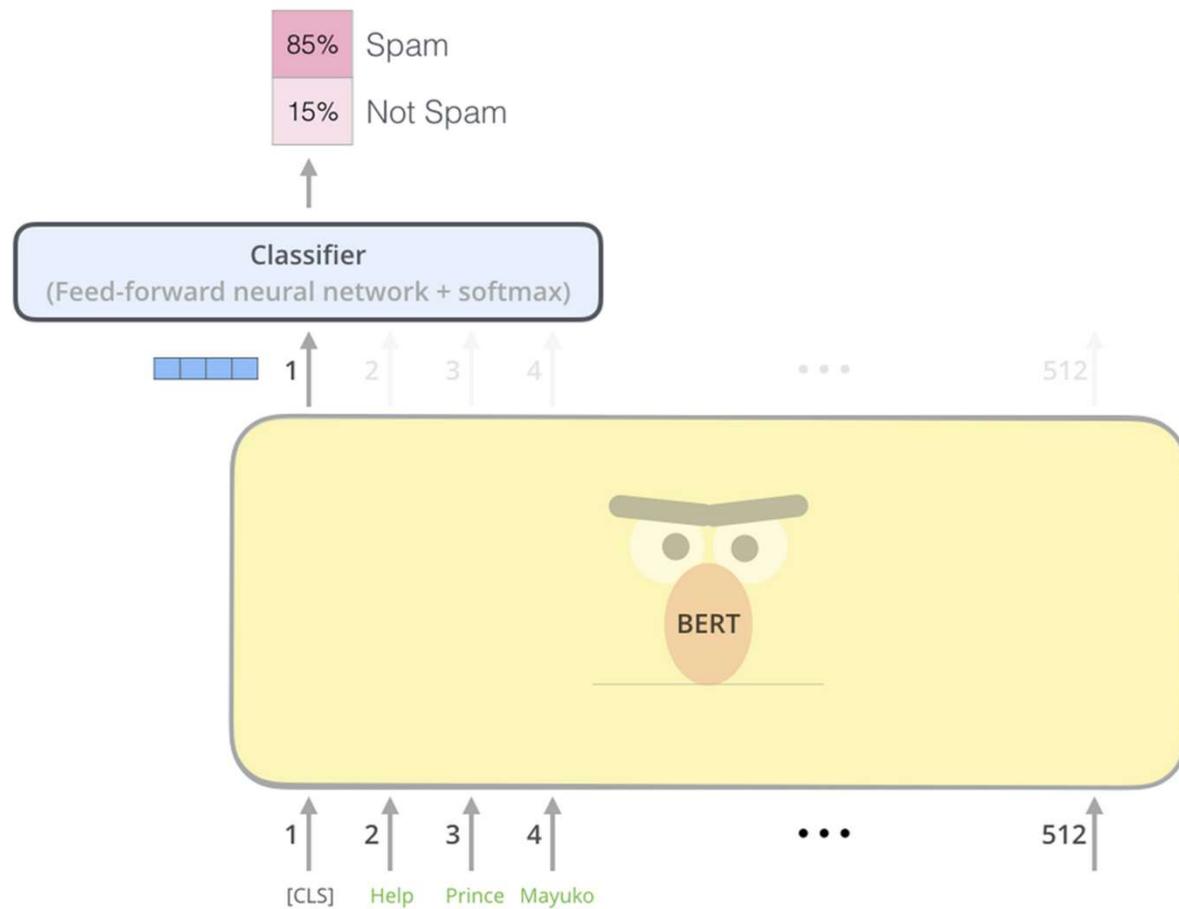


2 - Supervised training on a specific task with a labeled dataset.

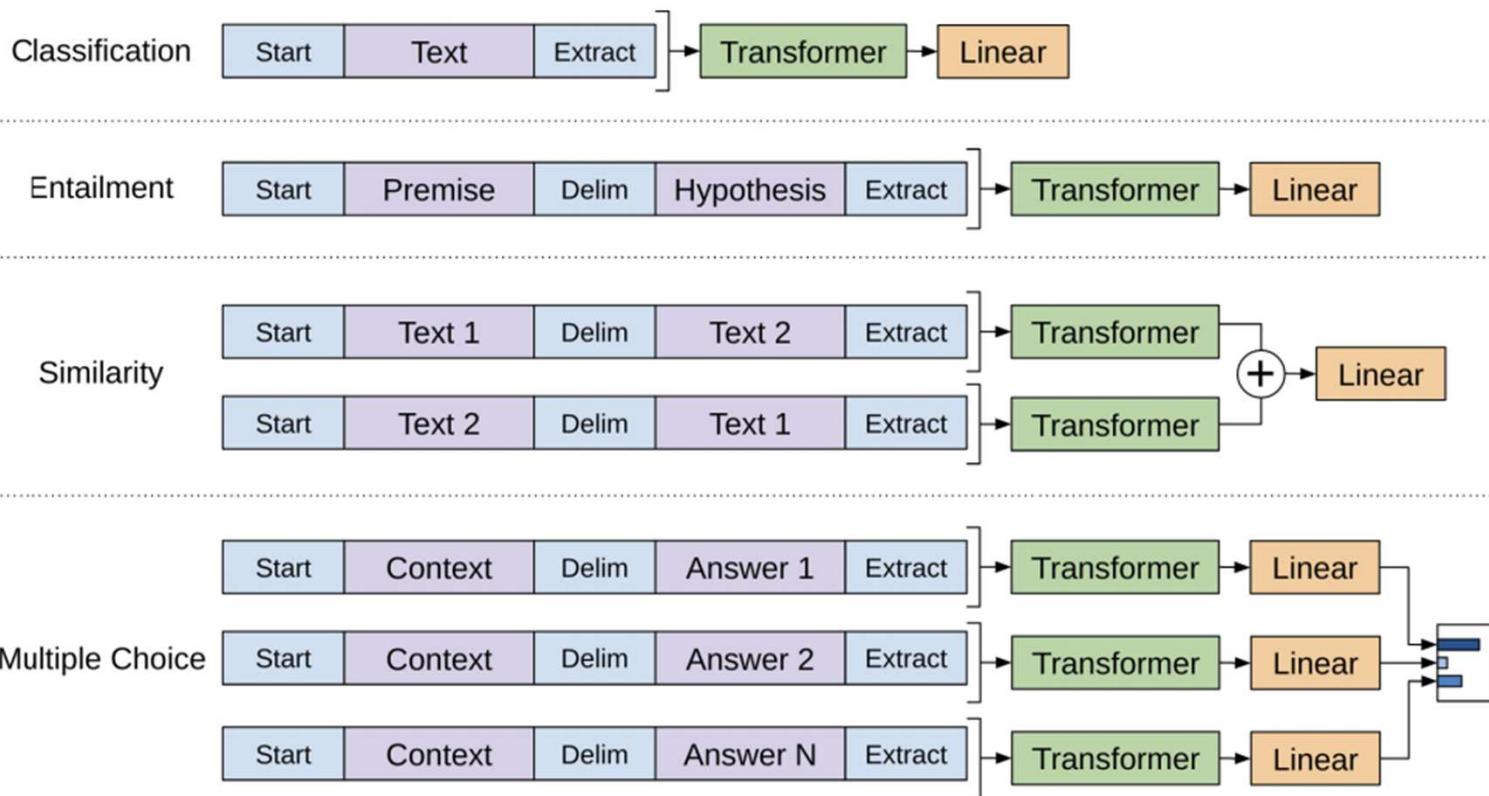


- <https://yashuseth.blog/2019/06/12/bert-explained-faqs-understand-bert-working/>
- <https://arxiv.org/abs/1810.04805v2>
- <http://jalammar.github.io/illustrated-transformer/>
- <http://jalammar.github.io/illustrated-bert/>

Transformer + Upstream Classifier



Different Upstream Tasks



Gluon NLP Lab [Sentiment Analysis]

BERT Pre-training and Fine-tuning

Preparation

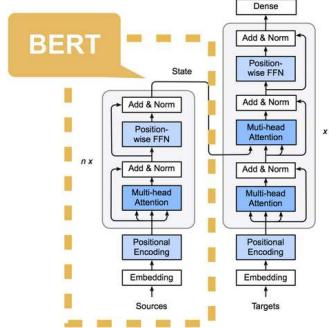
First, let's import necessary modules.

Note that `utils` includes some Blocks defined in the previous transformer notebook

```
In [1]: import random, math
import d2l
import numpy as np
import mxnet as mx
from mxnet import gluon, nd
import gluonnlp as nlp

from utils import PositionalEncoding, MultiHeadAttention
from utils import AddNorm, PositionWiseFFN, EncoderBlock
from utils import train_loop, predict_sentiment
```

Encoder



Segment Embedding

Different from the transformer encoder, the BERT encoder has an additional embedding for segment information.

Input	[CLS]	my	dog	is	cute	[SEP]	he	likes	play	# #ing	[SEP]
Token Embeddings	$E_{[CLS]}$	E_{my}	E_{dog}	E_{is}	E_{cute}	$E_{[SEP]}$	E_{he}	E_{likes}	E_{play}	$E_{# #ing}$	$E_{[SEP]}$
Segment Embeddings	E_A	E_A	E_A	E_A	E_A	E_B	E_B	E_B	E_B	E_B	E_B
Position Embeddings	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}

Similar to the Transformer encoder defined in the previous section, the BERT encoder has embeddings for words and positions. The `EncoderBlock` contains position-wise feed-forward network and self-attention blocks to encode inputs. For BERT, the newly added segment embedding captures the segment information of the input sentence pairs, used for the next sentence prediction task.

BERT Encoder

```
In [2]: class BERTEncoder(gluon.nn.Block):
    def __init__(self, vocab_size, units, hidden_size,
                 num_heads, num_layers, dropout, **kwargs):
        super(BERTEncoder, self).__init__(**kwargs)
        # segment_embed for segment information
        self.segment_embed = gluon.nn.Embedding(2, units)
        self.word_embed = gluon.nn.Embedding(vocab_size, units)
        self.pos_encoding = PositionalEncoding(units, dropout)
        self.blks = gluon.nn.Sequential()
        for i in range(num_layers):
            self.blks.add(EncoderBlock(units, hidden_size, num_heads, dropout))

    def forward(self, words, segments, mask, *args):
        X = self.word_embed(words) + self.segment_embed(segments)
        X = self.pos_encoding(X)
        for blk in self.blks:
            X = blk(X, mask)
        return X
```

Using the BERT Encoder

Now let's test the `BERTEncoder` with a data batch of 2 sentence pairs, each with 8 words. Random integers are used to represent words for demonstration purpose. For segment information, we use 0 to indicate the word comes from the first sentence, 1 to indicate the second sentence.

```
In [3]: encoder = BERTEncoder(vocab_size=30000, units=768, hidden_size=3072,
                           num_heads=12, num_layers=12, dropout=0.1)
encoder.initialize()

num_samples, num_words = 2, 8
# random words for testing
words = nd.random.randint(low=0, high=30000, shape=(num_samples, num_words))
# the corresponding segment information for each word
segments = nd.array([[0,0,0,0,1,1,1,1],[0,0,0,1,1,1,1,1]])

encodings = encoder(words, segments, None)
print(encodings.shape) # (batch_size, num_words, units)
```

(2, 8, 768)

Next Sentence Classifier

Let us take a look at the first pre-training task: next sentence prediction. For this task, the encoding of the first token (the "[CLS]" token) is passed to a feed-forward network to make prediction.

Since next sentence prediction is a binary classification problem, we can use `SigmoidBinaryCrossEntropyLoss` as the loss function. In the following code block, we pass the encoding results to the `NSClassifier` to get the next sentence prediction. We use 1 as the label for true next sentence, and 0 otherwise. The prediction result and the label are then passed to the loss function for loss evaluation.

```
In [4]: class NSClassifier(gluon.nn.Block):
    def __init__(self, units=768, **kwargs):
```

Refs

PyTorch-Transformers

 PyTorch-Transformers

 PASSED

PyTorch-Transformers (formerly known as `pytorch-pretrained-bert`) is a library of state-of-the-art pre-trained models for Natural Language Processing (NLP).

The library currently contains PyTorch implementations, pre-trained model weights, usage scripts and conversion utilities for the following models:

1. **BERT** (from Google) released with the paper [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](#) by Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova.
2. **GPT** (from OpenAI) released with the paper [Improving Language Understanding by Generative Pre-Training](#) by Alec Radford, Karthik Narasimhan, Tim Salimans and Ilya Sutskever.
3. **GPT-2** (from OpenAI) released with the paper [Language Models are Unsupervised Multitask Learners](#) by Alec Radford*, Jeffrey Wu*, Rewon Child, David Luan, Dario Amodei** and Ilya Sutskever**.
4. **Transformer-XL** (from Google/CMU) released with the paper [Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context](#) by Zihang Dai*, Zhilin Yang*, Yiming Yang, Jaime Carbonell, Quoc V. Le, Ruslan Salakhutdinov.
5. **XLNet** (from Google/CMU) released with the paper [XLNet: Generalized Autoregressive Pretraining for Language Understanding](#) by Zhilin Yang*, Zihang Dai*, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le.
6. **XLM** (from Facebook) released together with the paper [Cross-lingual Language Model Pretraining](#) by Guillaume Lample and Alexis Conneau.

These implementations have been tested on several datasets (see the example scripts) and should match the performances of the original implementations (e.g. ~93 F1 on SQuAD for BERT Whole-Word-Masking, ~88 F1 on RocStories for OpenAI GPT, ~18.3 perplexity on WikiText 103 for Transformer-XL, ~0.916 Pearson R coefficient on STS-B for XLNet). You can find more details on the performances in the Examples section of the [documentation](#).

Section	Description
Installation	How to install the package
Quick tour: Usage	Tokenizers & models usage: Bert and GPT-2
Quick tour: Fine-tuning/usage scripts	Using provided scripts: GLUE, SQuAD and Text generation
Migrating from pytorch-pretrained-bert to pytorch-transformers	Migrating your code from pytorch-pretrained-bert to pytorch-transformers
Documentation	Full API documentation and more

<https://github.com/huggingface/pytorch-transformers>

Papers With Code

 Papers With Code

<https://paperswithcode.com/area/natural-language-processing>

Browse > Natural Language Processing

Natural Language Processing

340 leaderboards • 200 tasks • 100 datasets • 2571 papers with code

Machine Translation

- Machine Translation: 40 leaderboards, 444 papers with code
- Transliteration: 15 papers with code
- Unsupervised Machine Translation: 9 leaderboards, 8 papers with code
- Multimodal Machine Translation: 7 papers with code
- Low-Resource Neural Machine Translation: 5 papers with code

See all 6 tasks

Question Answering

- Question Answering: 40 leaderboards, 343 papers with code
- Open-Domain Question Answering: 2 leaderboards, 18 papers with code
- Answer Selection: 3 leaderboards, 13 papers with code
- Community Question Answering: 11 papers with code
- Knowledge Base Question Answering: 1 leaderboard, 5 papers with code

Language Modelling

- Language Modeling: 3 leaderboards, 347 papers with code
- Sentence Pair Modeling: 2 papers with code

Sentiment Analysis

- Sentiment Analysis: 20 leaderboards, 254 papers with code
- Aspect-Based Sentiment Analysis: 2 leaderboards, 21 papers with code
- Multimodal Sentiment Analysis: 1 leaderboard, 11 papers with code
- Twitter Sentiment Analysis: 4 papers with code
- Fine-Grained Opinion Analysis: 1 leaderboard, 2 papers with code

Text Classification

- Text Classification: 32 leaderboards, 144 papers with code
- Document Classification: 9 leaderboards, 59 papers with code
- Sentence Classification: 5 leaderboards, 20 papers with code
- Textvec Text Categorization: 17 papers with code
- Emotion Classification: 1 leaderboard, 16 papers with code

See all 6 tasks

<https://paperswithcode.com/>

Lab 3

[optional]

Problem/Objective: Anomaly Detection

AWS Service: SageMaker, NEO

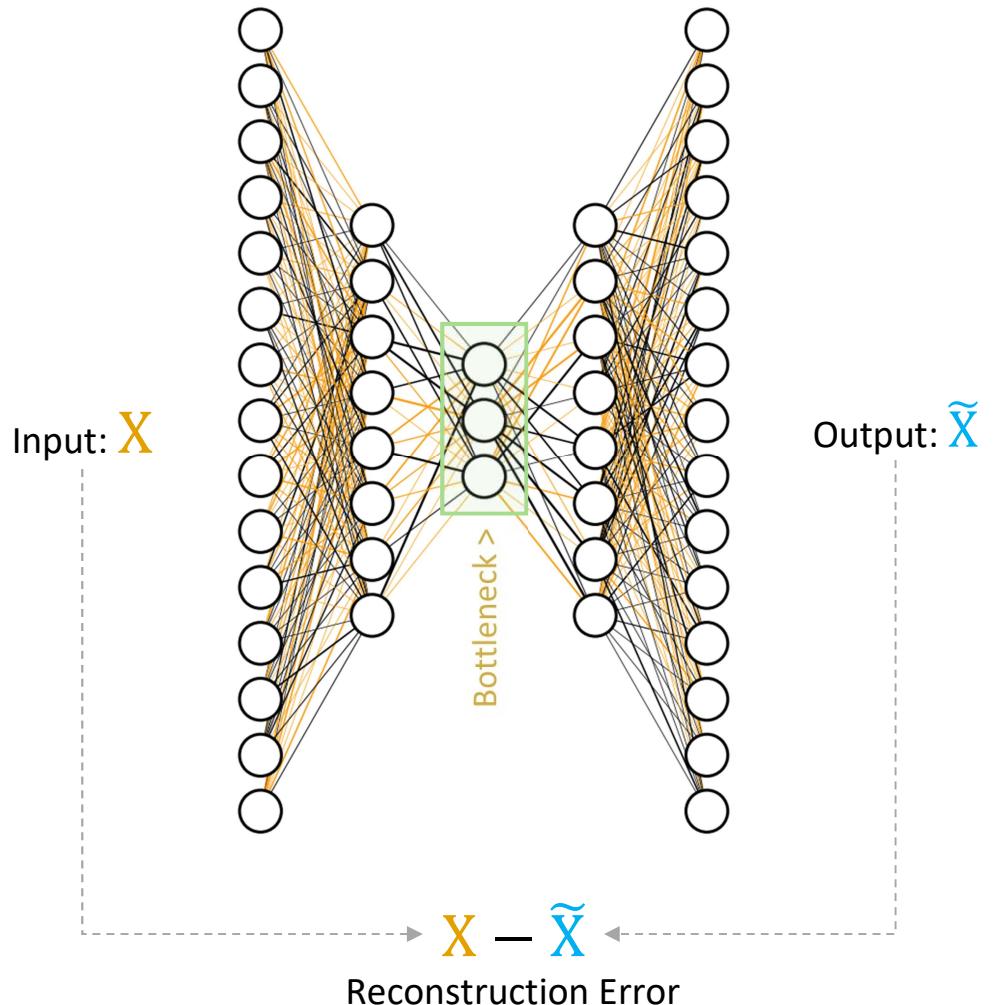
DL Architecture Overview: Autoencoder

Relevant Applications: Time-Series

https://github.com/miroenev/teach_DL/anomaly_detection

Deep Autoencoder

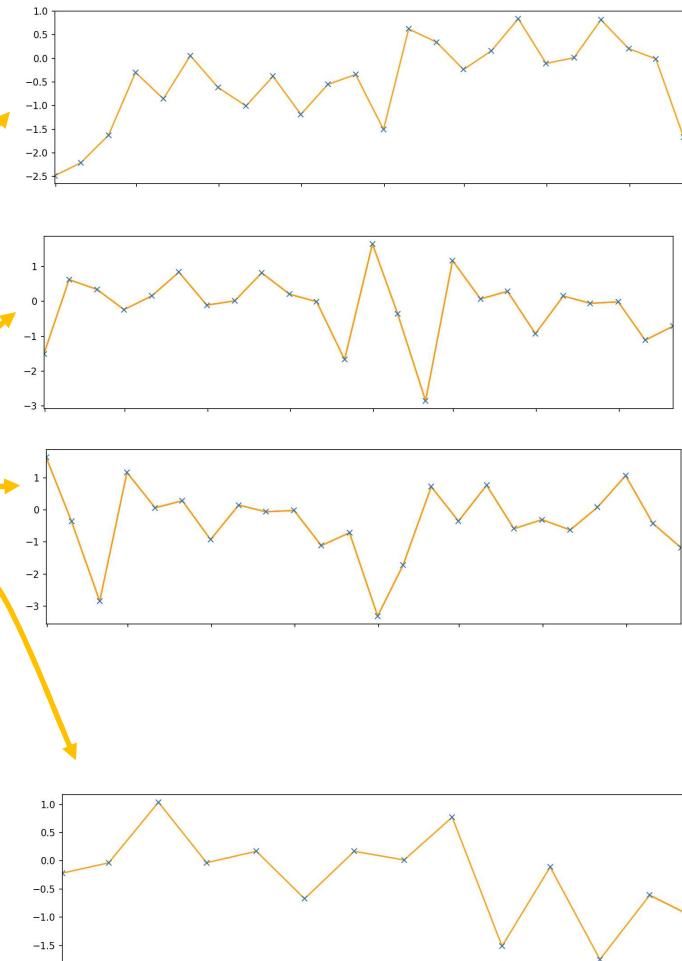
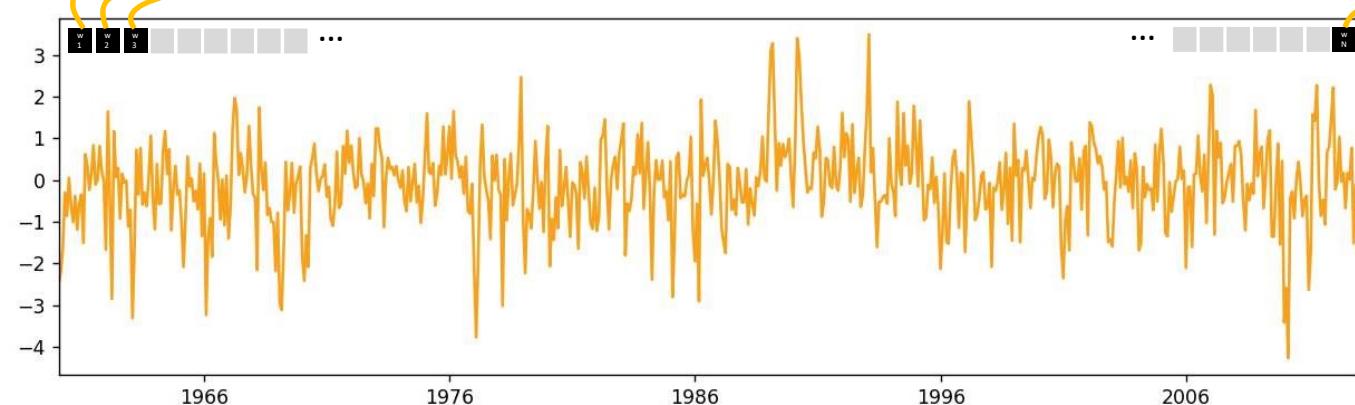
- Input layer
 - Time-Horizon
- Bottleneck layer
 - Summarized representation/'embedding'
- Output layer
 - Same dimensionality as input
- Reconstruction error
 - High errors indicate potential anomaly



Data pre-processing

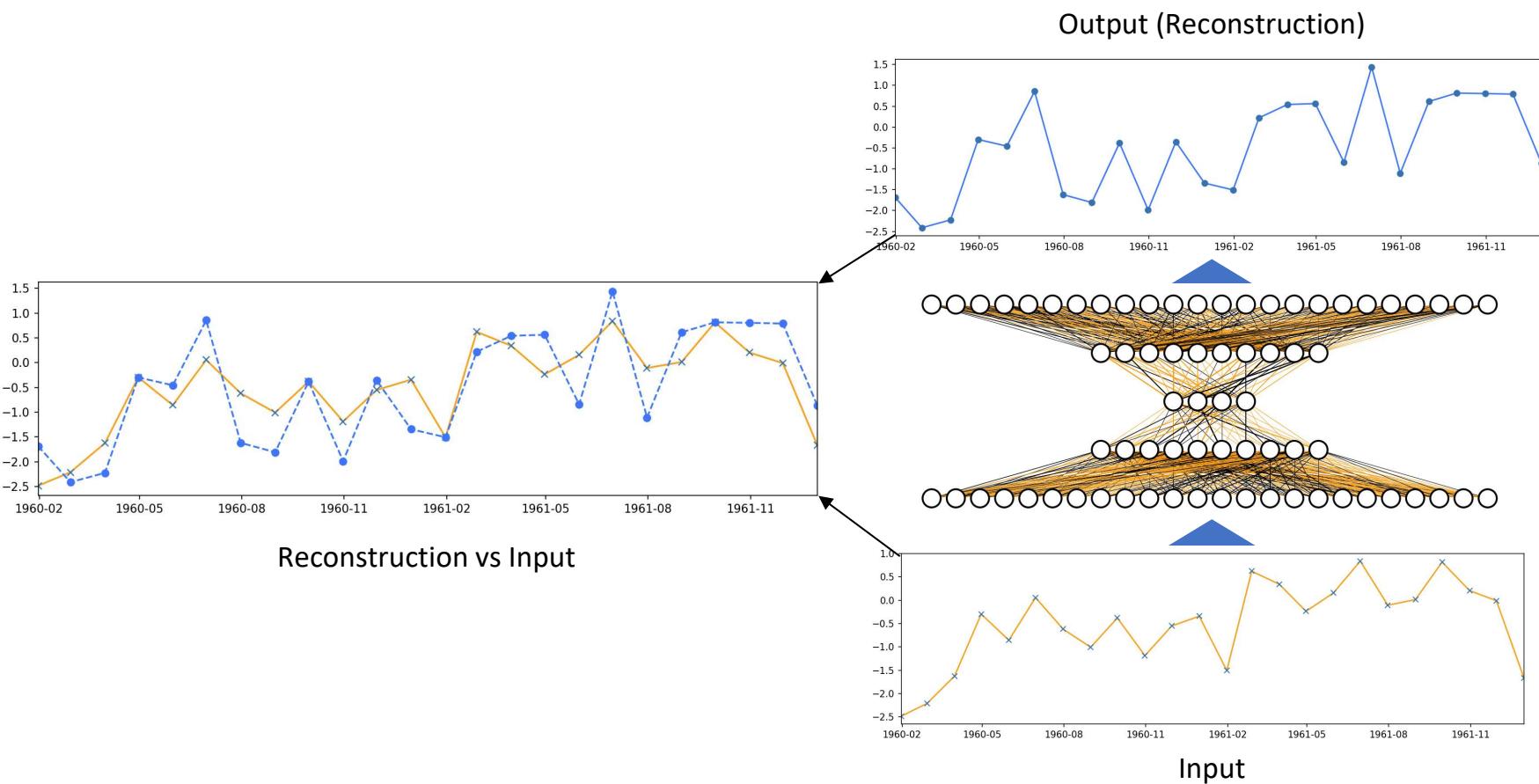
- Sliding windows

[overlap = hyperparameter]



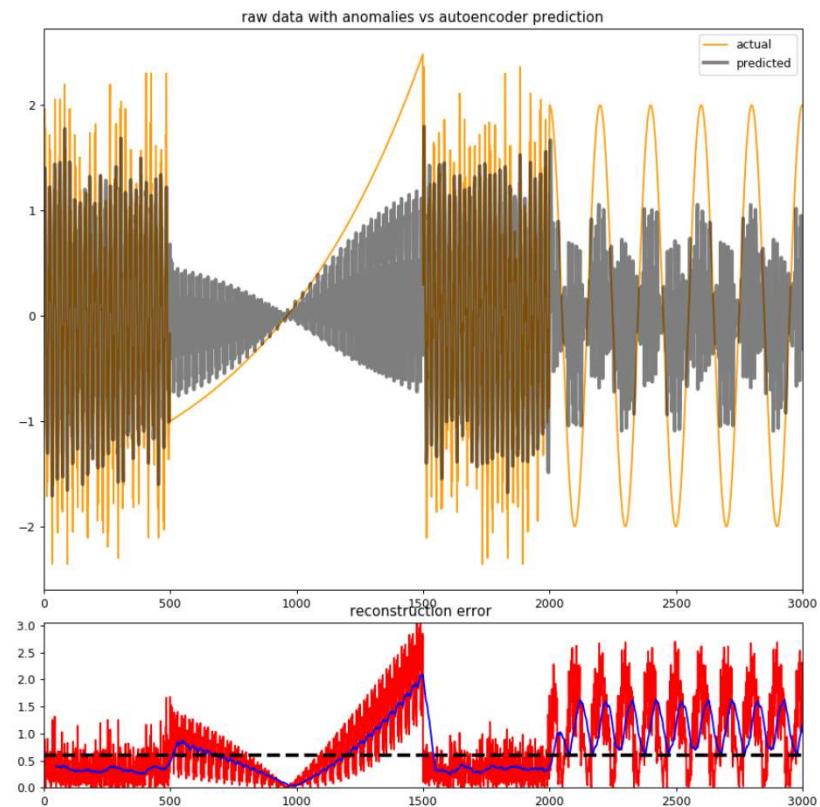
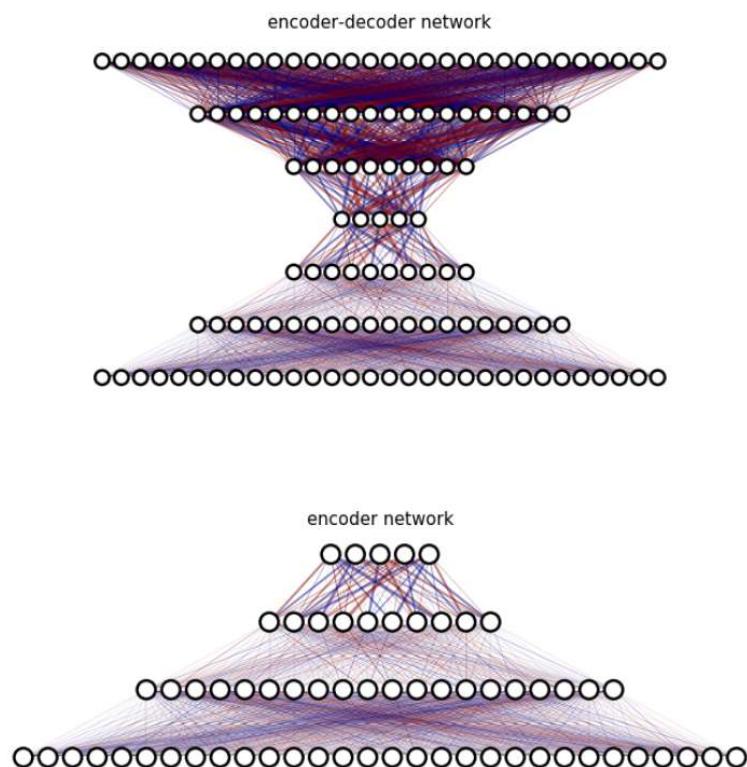
Reconstruction Error

[Trigger for Anomaly]



Anomaly Detection on Time Series Data

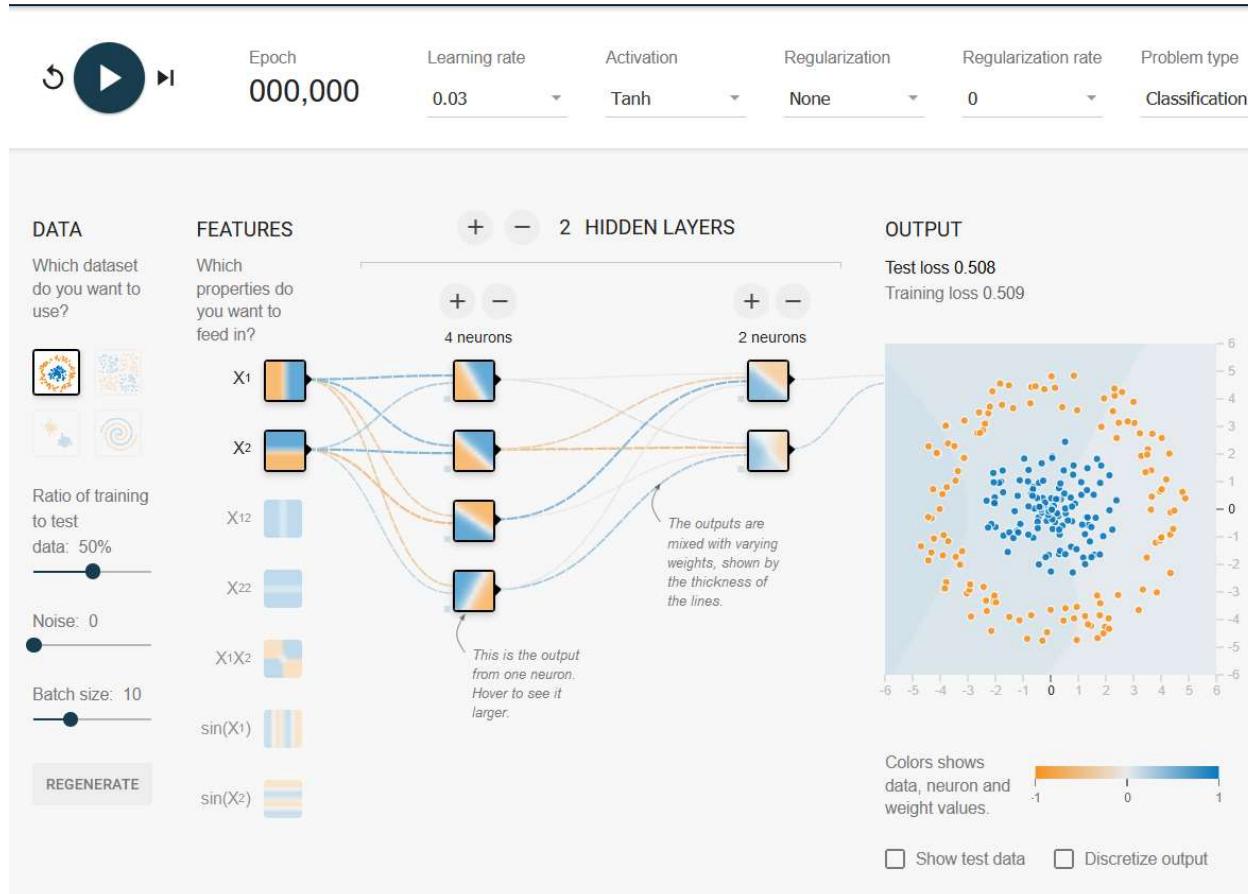
In this lab we'll generate [synthetic] time-series data and train a DL model [autoencoder] to encode [summarize/embed] and reconstruct sliding windows from this dataset. In this context, the amount of reconstruction error (the difference between an input and the trained network's output) is used as an indicator of potential anomalies. To test this we'll mix in artificial anomalies into normal data and analyze the model's reconstruction error.



Thank you KDD!

Appendix

Demo



Resources

<http://www.arxiv-sanity.com/top>

Arxiv Sanity Preserver
Built in spare time by @karpathy to accelerate research.
Serving last 79657 papers from cs [CV|CL|G]AI[NE]stat.ML

User: Pass: Login or Create

most recent top recent top hype friends discussions recommended library

Only show v1 | Last day Last 3 days Last week Last month Last year All time

Top papers based on people's libraries:

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova
5/24/2019 (v1: 10/11/2018) cs.CL

1810.04805v2 pdf show similar discuss

We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications. BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5% (7.7% point absolute improvement), MultiNLi accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (6.1 point absolute improvement).

An Introduction to Deep Reinforcement Learning
Vincent Francois-Lavet, Peter Henderson, Riaeshat Islam, Marc G. Bellemare, Joelle Pineau
12/3/2018 (v1: 11/30/2018) cs.LG | cs.AI | stat.ML

1811.12560v2 pdf show similar discuss

Deep reinforcement learning is the combination of reinforcement learning (RL) and deep learning. This field of research has been able to solve a wide range of complex decision-making tasks that were previously out of reach for a machine. Thus, deep RL opens up many new applications in domains such as healthcare, robotics, smart grids, finance, and many more. This manuscript provides an introduction to deep reinforcement learning models, algorithms and techniques. Particular focus is on the aspects related to generalization and how deep RL can be used for practical applications. We assume the reader is familiar with basic machine learning concepts.

A Style-Based Generator Architecture for Generative Adversarial Networks
Tero Karras, Samuli Laine, Timo Aila
2/29/2019 (v1: 12/12/2018) cs.NE | cs.LG | stat.ML

1812.04948v3 pdf show similar discuss

<https://www.reddit.com/r/MachineLearning/top/?t=year>

r/MachineLearning

Posts

VIEW SORT TOP PAST YEAR

1.2k We are Oriol Vinyals and David Silver from DeepMind's AlphaStar team, joined by StarCraft II pro players TLO and MaNa! Ask us anything
Posted by u/OriolVinyals 6 months ago 1.0K Comments Share Save Hide Report

1.1k Discussion [D] Has anyone noticed a lot of ML research into facial recognition of Uyghur people lately?
Posted by u/Kickuchiyo 2 months ago 234 Comments Share Save Hide Report

37 We know you see a lot of our ads. We thought we'd give you a little break from Triplebyte Mike – awesome as he is – and offer you this mostly blank space instead. Enjoy! (Or, take our coding quiz, if you feel like it.)
Posted by u/Triplebyte_official 1 month ago from triplebyte.com PROMOTED triplebyte.com LEARN MORE

1.0k Discussion [D] If all you're doing is copy/pasting someone else's blog/tutorial/stackoverflow and making minor adjustments, please do not create another frigging Medium article. It's just worthless noise.
Posted by u/halfassadmin 1 month ago 139 Comments Share Save Hide Report

949 UC Berkeley and Berkeley AI Research published all materials of CS 188: Introduction to Artificial Intelligence, Fall 2018 inst.eecs.berkeley.edu/~cs188...
Posted by u/dronecub 7 months ago 56 Comments Share Save Hide Report

906 [R] Few-Shot Unsupervised Image-to-Image Translation preview.reddit.it/q7yd1...
Posted by u/mingyuliu 2 months ago 48 Comments Share Save Hide Report

908 Discussion [D] What is the best ML paper you read in 2018 and why?
Posted by u/omniscientclown 7 months ago 122 Comments Share Save Hide Report

774 Discussion [D] An analysis on how AlphaStar's superhuman speed is a band-aid fix for the limitations of imitation learning.
Posted by u/SoulDrivenOlivies 6 months ago 291 Comments Share Save Hide Report

760 Discussion [Discussion] When ML and Data Science are the death of a good company: A cautionary tale.
Posted by u/AlexSnakeKing 3 months ago

DLI- Deep Learning Institute

<https://www.nvidia.com/en-us/deep-learning-ai/education/>

DEEP LEARNING FUNDAMENTALS

- ✓ Fundamentals of Deep Learning for Computer Vision 
- ✓ Getting Started with AI on Jetson Nano 
- ✓ Image Classification with DIGITS
- ✓ Object Detection with DIGITS
- ✓ Optimization and Deployment of TensorFlow Models with TensorRT
- ✓ Accelerating Data Science Workflows with RAPIDS
- ✓ Image Segmentation with TensorFlow
- ✓ Signal Processing with DIGITS

DEEP LEARNING FOR DIGITAL CONTENT CREATION

- ✓ Image Style Transfer with Torch
- ✓ Rendered Image Denoising Using Autoencoders
- ✓ Image Super Resolution Using Autoencoders

DEEP LEARNING FOR HEALTHCARE

- ✓ Modeling Time Series Data with Recurrent Neural Networks in Keras
- ✓ Medical Image Classification Using the MedNIST Dataset
- ✓ Data Science Workflows for Deep Learning in Medical Applications
- ✓ Medical Image Segmentation with DIGITS
- ✓ Image Classification with TensorFlow: Radiomics—1p19q Chromosome Status Classification
- ✓ Medical Image Analysis with R and MXNet
- ✓ Data Augmentation and Segmentation with Generative Networks for Medical Imaging
- ✓ Coarse-to-Fine Contextual Memory for Medical Imaging

DEEP LEARNING FOR INTELLIGENT VIDEO ANALYTICS

- ✓ AI Workflows for Intelligent Video Analytics with DeepStream

CUDA - X

<https://www.nvidia.com/en-in/technologies/cuda-x/>

- 1 Computational Finance
- 2 Climate, Weather and Ocean Modeling
- 2 Data Science and Analytics
- 4 Artificial Intelligence
 - DEEP LEARNING AND MACHINE LEARNING
- 8 Federal, Defense and Intelligence
- 10 Design for Manufacturing/Construction: CAD/CAE/CAM
 - COMPUTATIONAL FLUID DYNAMICS
 - COMPUTATIONAL STRUCTURAL MECHANICS
 - DESIGN AND VISUALIZATION
 - ELECTRONIC DESIGN AUTOMATION
 - INDUSTRIAL INSPECTION
- 19 Media & Entertainment
 - ANIMATION, MODELING AND RENDERING
 - COLOR CORRECTION AND GRAIN MANAGEMENT
 - COMPOSITING, FINISHING AND EFFECTS
 - EDITING
 - ENCODING AND DIGITAL DISTRIBUTION
 - ON-AIR GRAPHICS
 - ON-SET, REVIEW AND STEREO TOOLS
 - WEATHER GRAPHICS
- 29 Medical Imaging
- 30 Oil and Gas
- 31 Research: Higher Education and Supercomputing
 - COMPUTATIONAL CHEMISTRY AND BIOLOGY
 - NUMERICAL ANALYTICS
 - PHYSICS
 - SCIENTIFIC VISUALIZATION
- 47 Safety and Security
- 49 Tools and Management

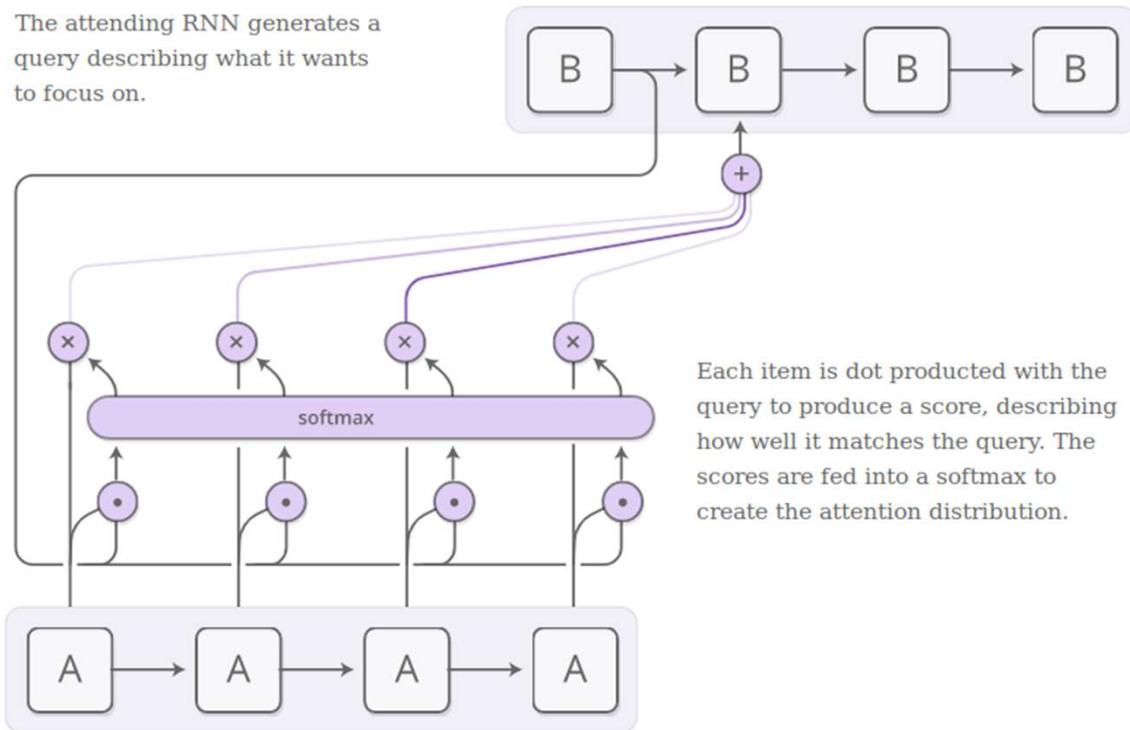
Mixed Precision

Conclusions

- **Mixed precision training benefits:**
 - Math, memory speedups
 - Larger minibatches, larger inputs
- **Automatic Loss Scaling simplifies mixed precision training**
- **Mixed precision matches FP32 training accuracy for a variety of:**
 - **Tasks:** classification, regression, generation
 - **Problem domains:** images, language translation, language modeling, speech
 - **Network architectures:** feed forward, recurrent
 - **Optimizers:** SGD, Adagrad, Adam
- **Note on inference:**
 - Can be purely FP16: storage and math (use library calls with FP16 accumulation)
- **More details:**
 - S81012: Training Neural Networks with Mixed Precision: Real Examples (Thu, 9am)
 - <http://docs.nvidia.com/deeplearning/sdk/mixed-precision-training/>

Attention

The attending RNN generates a query describing what it wants to focus on.



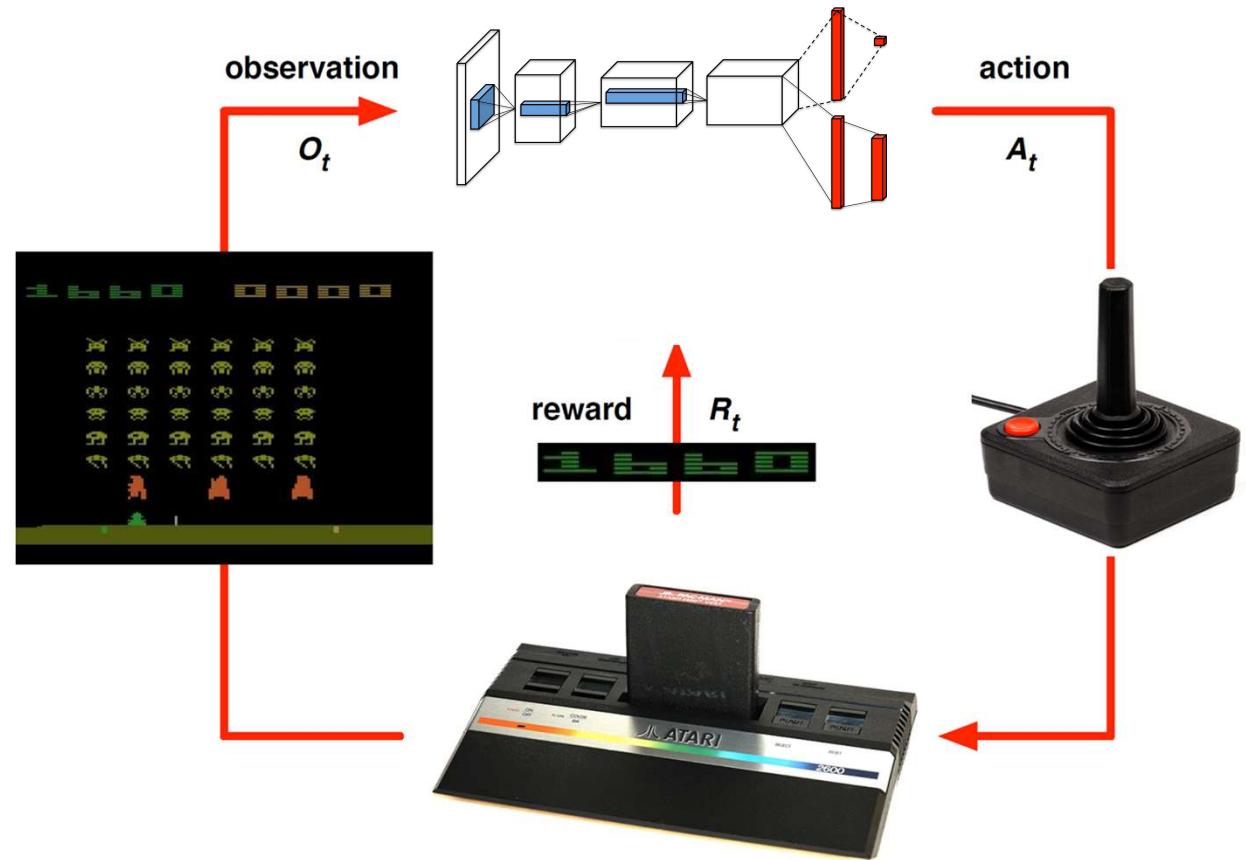
- <https://distill.pub/2016/augmented-rnns/#attentional-interfaces>

Reinforcement Learning

Applying RL to games

Atari Example

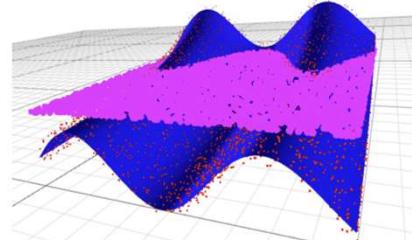
- Agent is a DL network
 - Interprets screen pixels
 - Outputs game action
 - Possible to use CNN+RNN
- Environment is Atari Emulator
 - Game AI
- Reward is game score



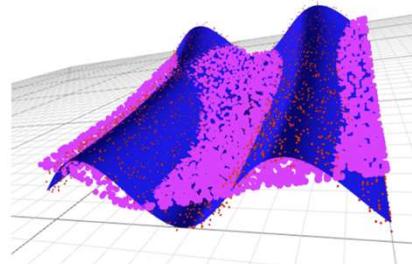
DL Intro

- Visualization & Model Training
 - Full Fidelity Visualization [vs] Graph Nodes
 - Model evolution over time

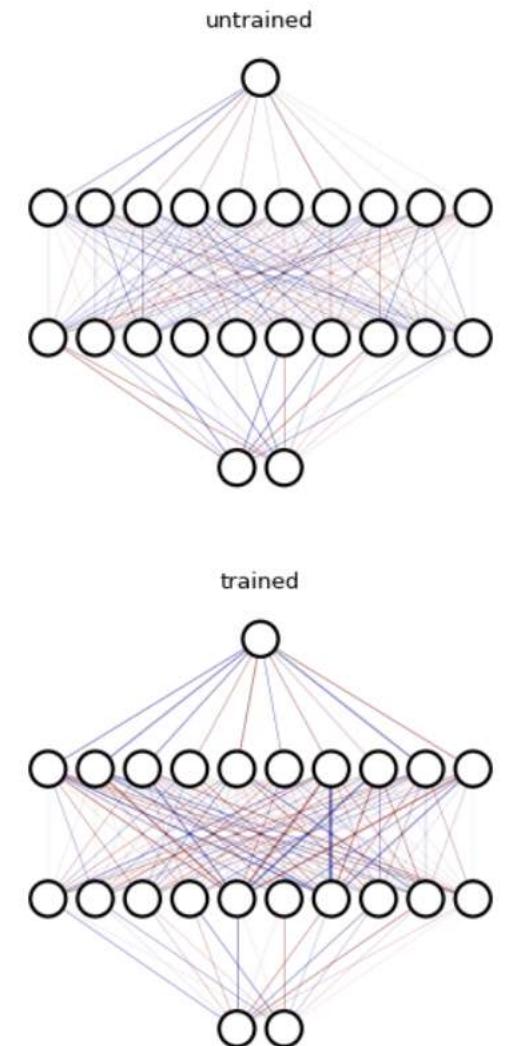
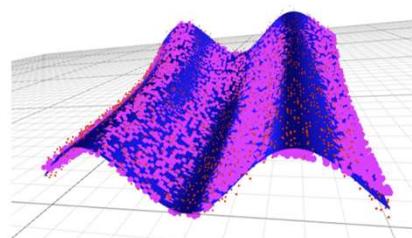
Before Training
[Random Initialization]



Midway
[150 epochs]



Done Training
[300 epochs]



GluonNLP Integration in ELIT



Word2Vec (Mikolov et al., 2013)

GloVe (Pennington et al., 2014)

FastText (Bojanowski et al., 2017)

ELMo (Peters et al., 2018)

BERT (Devlin et al., 2018)

MXNet Implementation of
Contextual String Embedding
model (Akbik et al., 2018)