CIS 6930 Topics in Computing for Data Science
Week 12: More Deep Learning Topics (2)

Tue 11/18/2021

Yoshihiko (Yoshi) Suhara

2pm-3:20pm & 3:30pm-4:50pm

Course Schedule

- (Today) Thu 11/18 **(2x 80min)** (* Finalize the term project title & plan!)
- Tue 11/23 Project presentation (1)
- Thu 11/30 Project presentation (2) (* Final session! 🥲)

Term Project
Project Presentation & Project Report

Term Project

- ~~Fri 11/12: Feedback & Pre-finalize the project plan~~
- **Fri 11/18:** Finalize the project title and plan
- Tue 11/23: Project presentations (1)
- Tue 11/30: Project presentations (2)
- **Tue 12/7:** Project report deadline

Each student must work on their own term project (**No group work**)

Did you confirm the plan with me? Please reply to the e-mail
Alternatively, use the office hours after the session to discuss it

Project Presentation (5min + 3min QA)
Tue 11/23 & Tue 11/30

The presentation should contain

- 1) Motivation (30 seconds)
- 2) Clear problem definition (1 minute)
- 3) Methodology (1-2 minutes)
- 4) Evaluation: Results and Discussion (1-2 minutes)
- 5) Key takeaway (30 seconds)

Notes

- Presentations will NOT be evaluated by the performance of your solution
- Students are encouraged to share in-progress work to get feedback from the instructor
- Please use this opportunity to improve the quality of your final project report

Project Report (due Tue 12/7)

The project report must contain

- 1) Motivation
- 2) Clear problem definition + related background
- 3) Methodology: Description of the solution/baselines and the dataset
- 4) Evaluation: Results and Discussion
- 5) Personal reflection: Challenges encountered, lessons learned

The ideal project report should also answer the following questions

- How is your solution different from other solutions (baselines)?
- How better is your solution than the baselines?
- Why is your solution better than the baselines?

Any Questions?

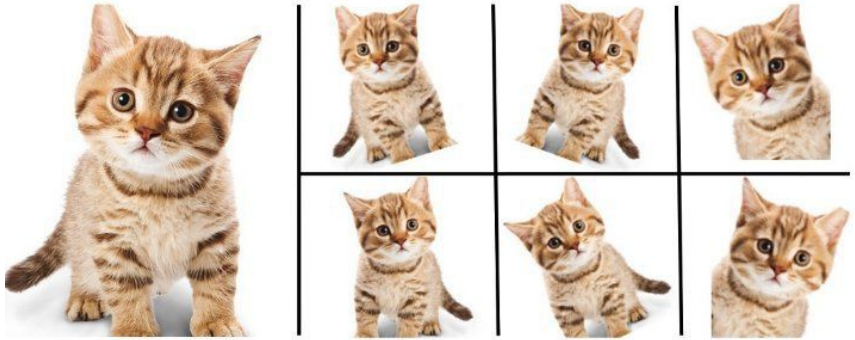
Agenda: A Series of Short Talks!

- Advanced topics & Future directions
  - Data Augmentation
  - Adversarial Attacks; Interpretability & Explainability
  - Multilingual Models
  - Vision & Language
  - Pre-trained Language Models for Data Integration
  - Open AI GPT-3
  - Prompt Learning
- Topic that are not covered by the course
  - Reinforcement Learning
  - Graph Neural Networks
  - Time Series Data
  - MLOps

Data Augmentation

What is Data Augmentation and Why?

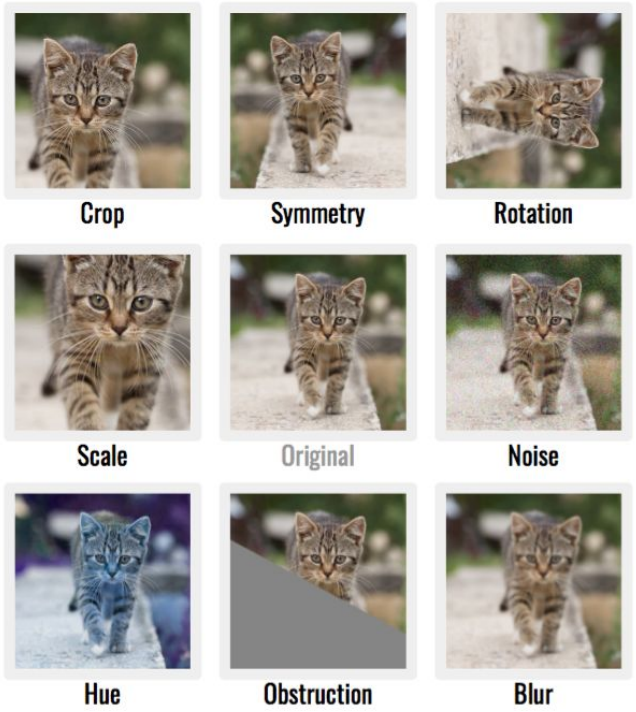
- A technique that **synthetically generate additional training examples** based on the original data to help the model more robust and generalized better



Rotom: A Meta-Learned Data Augmentation Framework for Entity Matching, Data Cleaning, Text Classification, and Beyond

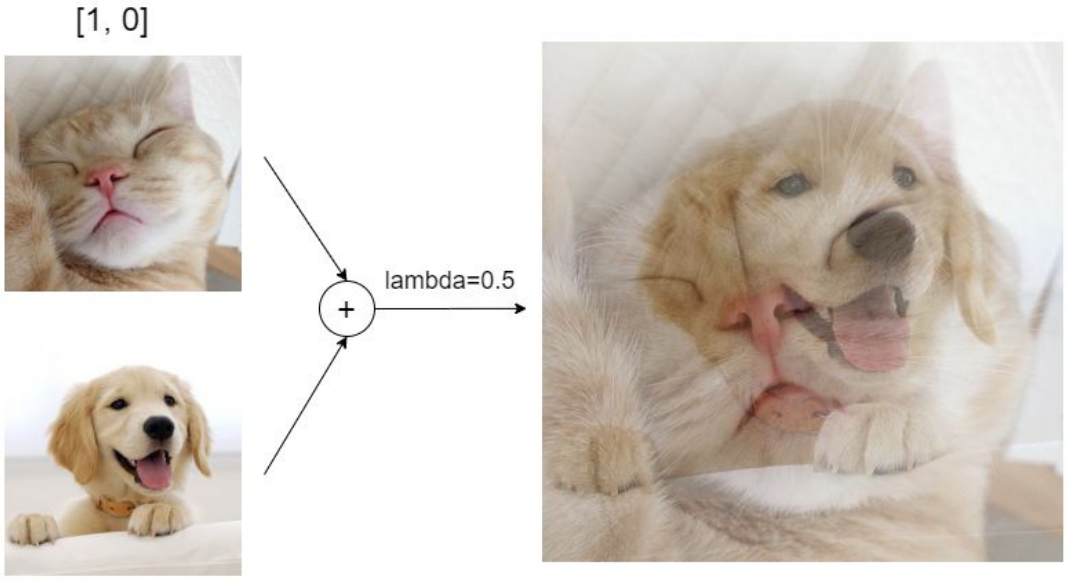
Data Augmentation for Image Data: Basic Techniques

- Invariance property



Data Augmentation for Image Data: MixUp

- Interpolate two images and their labels



[1710.09412] mixup: Beyond Empirical Risk Minimization

Cutout & CutMix

[1905.04899] CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features

Data Augmentation for Text Data

- It’s not that straightforward compared to image data. Why?

Data Augmentation for Text Data

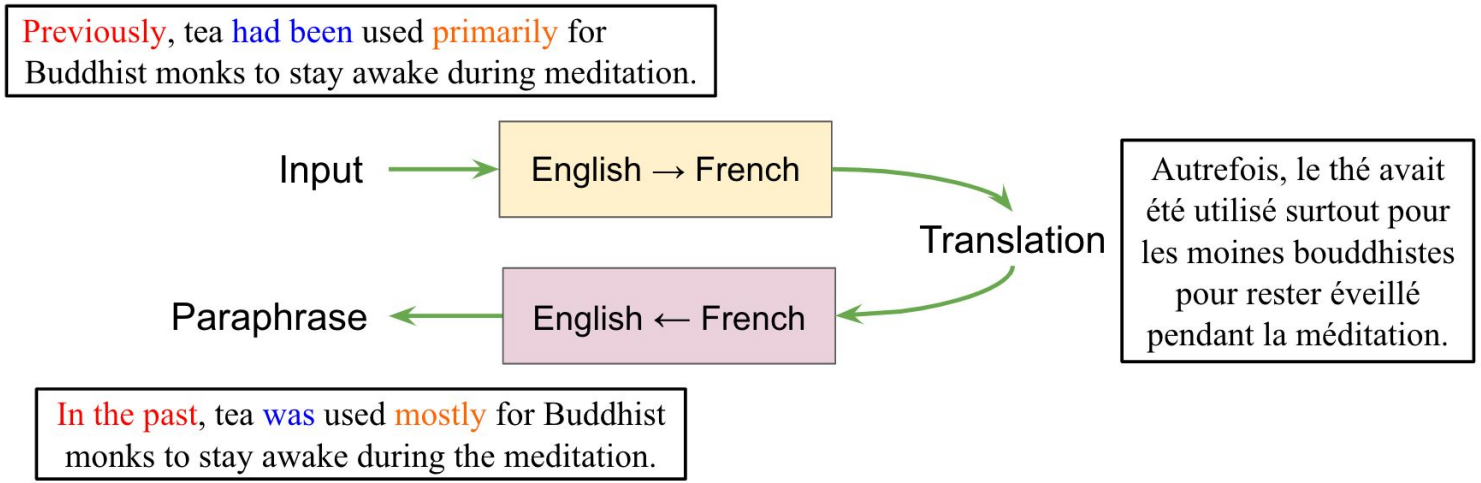
- It’s not that straightforward compared to image data. Why?
- → A little perturbation can completely change the meaning
  - e.g., It’s a cat → It’s not a cat

Data Augmentation for Text: Basic Techniques

- Character/word-level random insertion/deletion/swapping
- Replacing randomly chosen words with synonyms
- etc.

Model-based Data Augmentation

- Paraphrase generation
- Back-translation



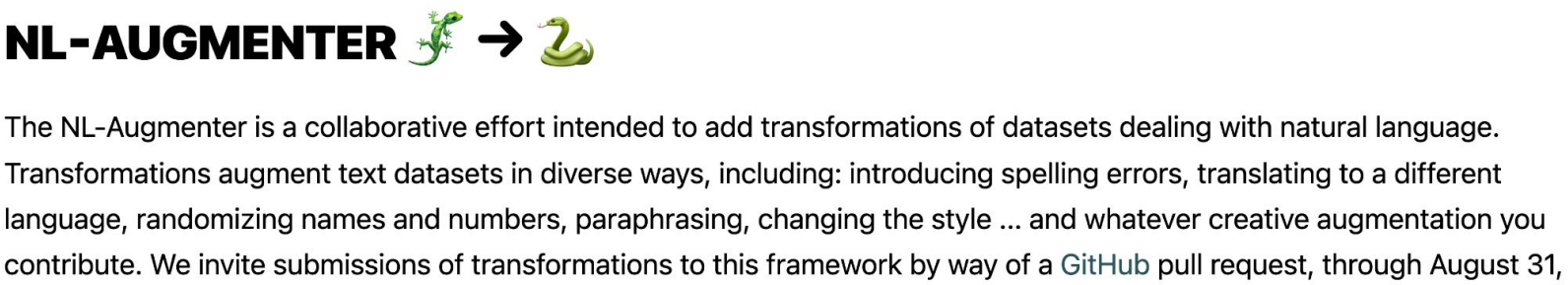
Data Augmentation Toolbox

- Image
  - https://github.com/facebookresearch/AugLy
  - etc.
- Text
  - https://github.com/styfeng/DataAug4NLP
    - [2105.03075] A Survey of Data Augmentation Approaches for NLP
  - etc.

NL-Augmenter
Collaborative Data Augmentation

https://gem-benchmark.com/nl_augmenter

Feature Engineering in the DL era?



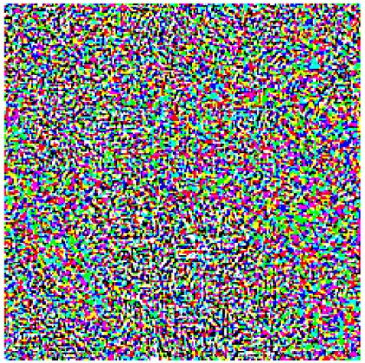
Adversarial Attacks
Interpretability & Explainability

Adversarial Examples: A Famous Example

$\boldsymbol{x}$

"panda"
57.7% confidence

$+ .007 \times$



$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"nematode"
8.2% confidence

$=$

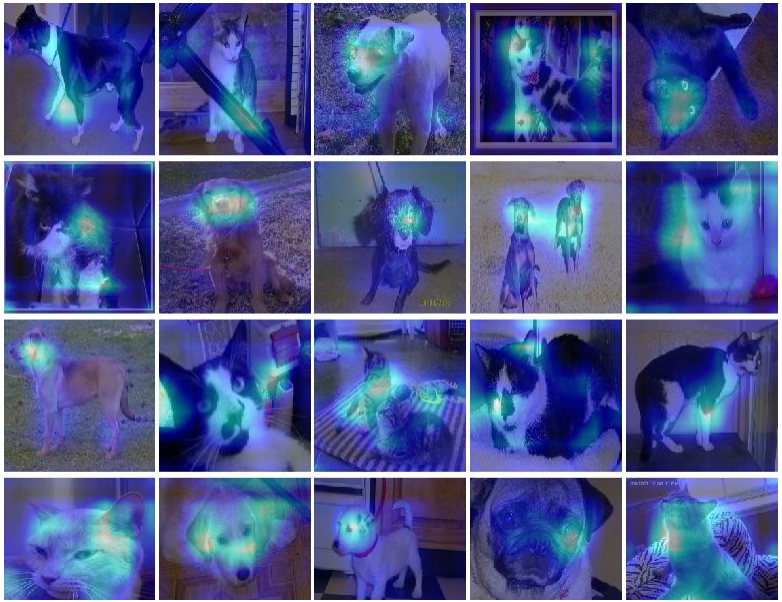
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"gibbon"
99.3 % confidence

[1412.6572] Explaining and Harnessing Adversarial Examples

Interpretability & Explainability

- No clear definition!
- A series of efforts that aim to provide “plausible” evidence



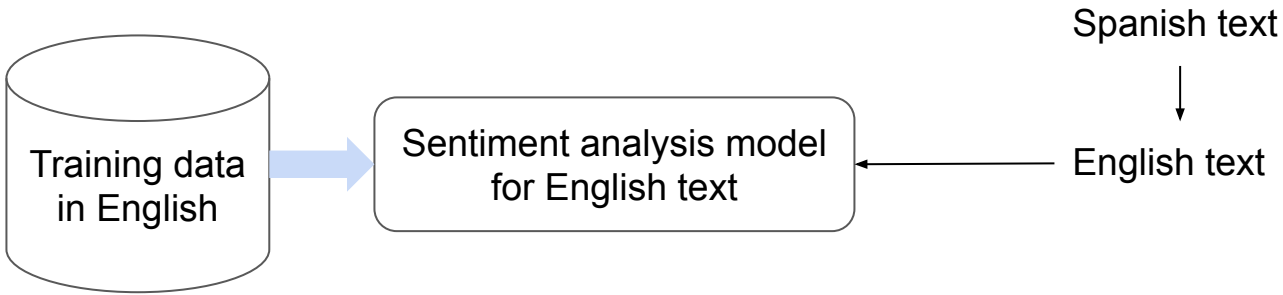
Interpretability in Deep Learning with W&B — CAM and GradCAM

Multilingual Models

Traditional NLP: Monolingual Models

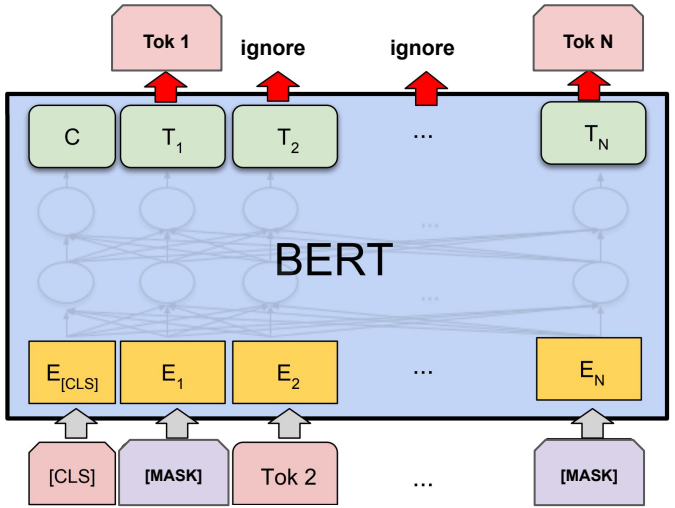
- Models are language-specific
  - An English model can be only used for English text

- One solution to apply an English model to another language is translating the input text into English (by Machine-Translation)
  - Translation quality is bottleneck :(



Multilingual BERT

- A **single** BERT model **pre-trained** on Wikipedia articles **in 102 languages**



I have a cat .

I [MASK] a cat .

Tengo un gato .

[MASK] un gato .

Multilingual pre-trained model: Example

https://huggingface.co/nlptown/bert-base-multilingual-uncased-sentiment

Research Trend

- Task-specific models → Task-independent pre-trained models + Fine-tuning
- Monolingual models → Multilingual models
- what else?

Research Trend

- Task-specific models → Task-independent pre-trained models + Fine-tuning
- Monolingual models → Multilingual models
- Unimodal (i.e., text) models → Multi-modal (e.g., text + image) models

Vision & Language

Visual Question Answering

- Input: Question + Image
- Output: Answer

Who is wearing glasses?

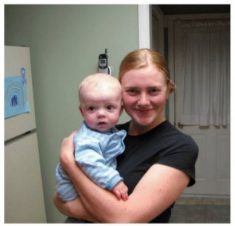
man

woman

Where is the child sitting?

fridge

arms



Is the umbrella upside down?

yes

no

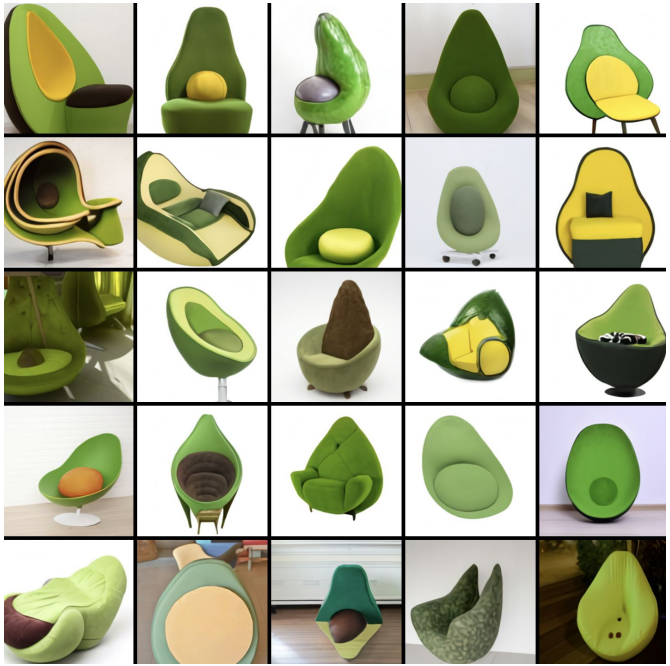
How many children are in the bed?

2

1

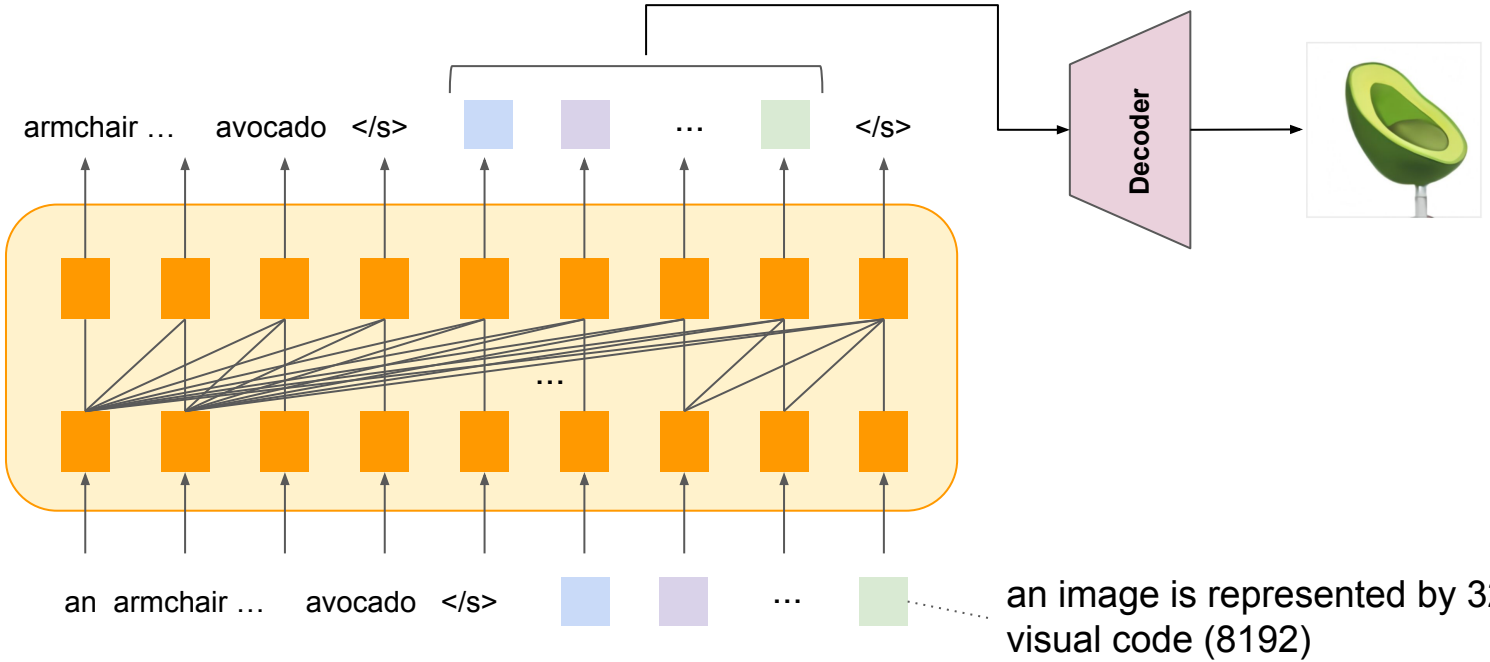
Text-to-Image Generation

- “An armchair in the shape of an avocado”



DALL·E

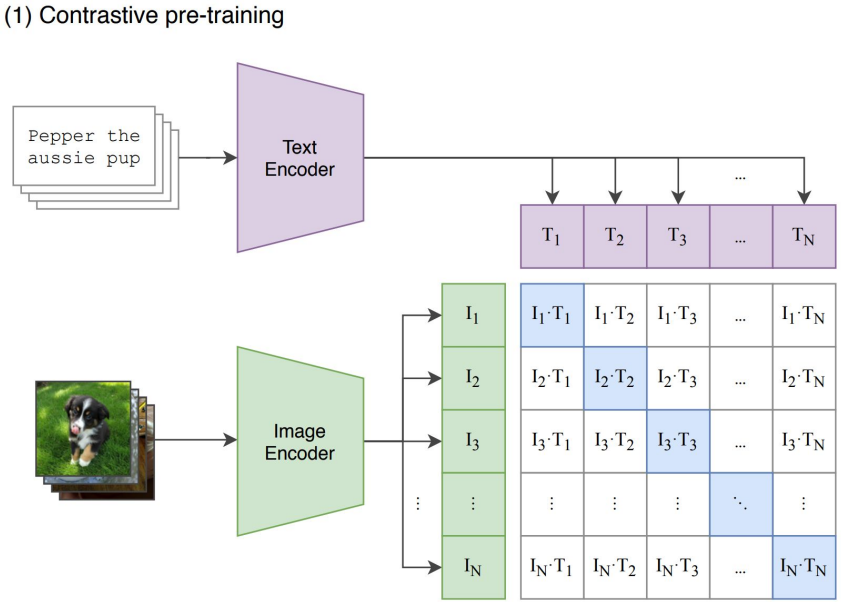
- A small GPT-3 model pre-trained with a huge number of **caption-image pairs**



[2102.12092] Zero-Shot Text-to-Image Generation

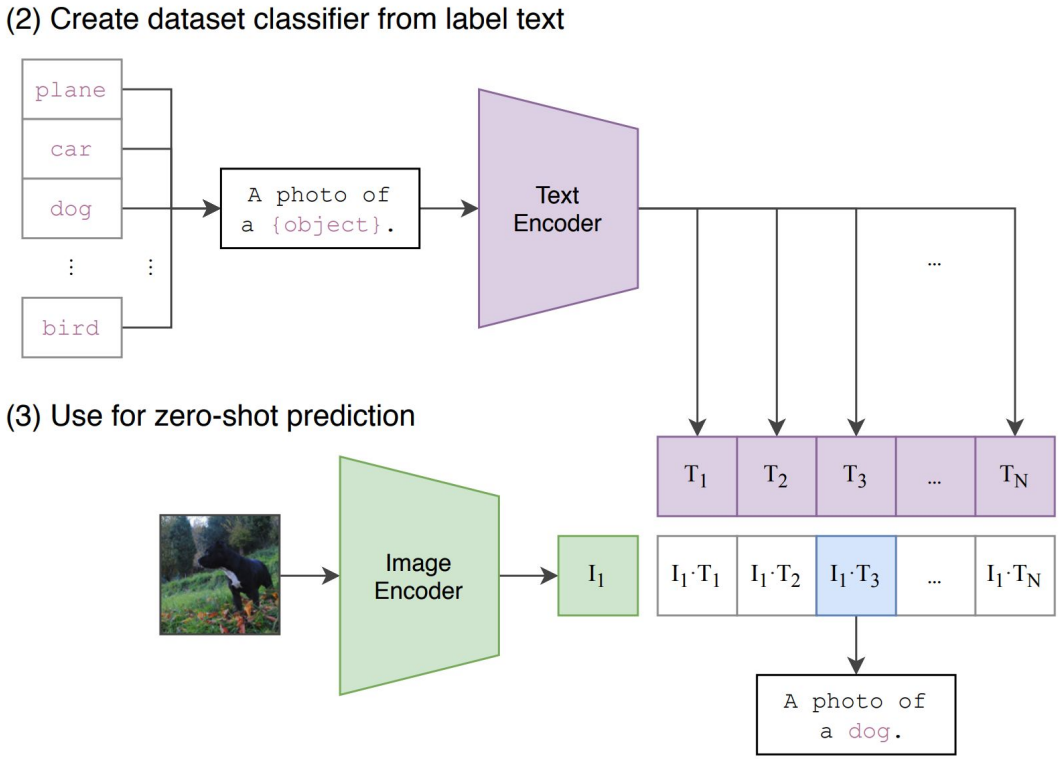
CLIP (Contrastive Language–Image Pre-training)

- Text Encoder (e.g., Transformer Encoder) + image encoder (e.g., Vision Transformer)
- Pre-training with a huge number of **caption-image pairs**



[2103.00020] Learning Transferable Visual Models From Natural Language Supervision

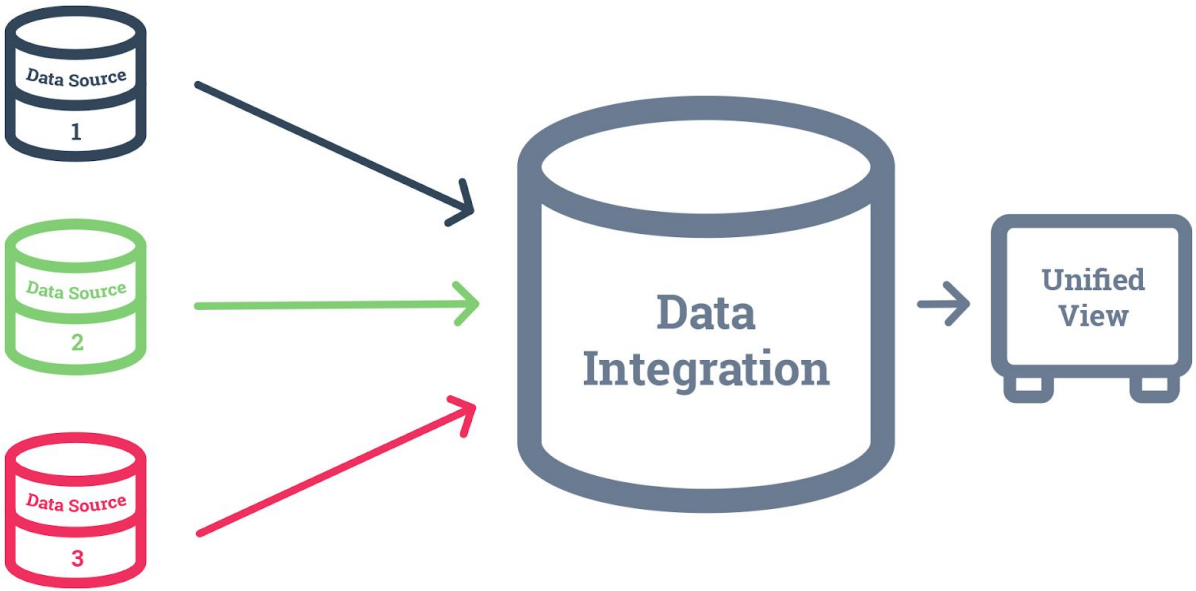
CLIP for Zero-shot Image Classification



Pre-trained LMs for Data Integration

Data Integration

- One of the ultimate goal of data management research



Entity Matching

- Input: Two collections of data entries (tables, JSON files, text, ...)
- Output: all entry pairs that refer to the same entity (products, businesses, ...)

Table A:

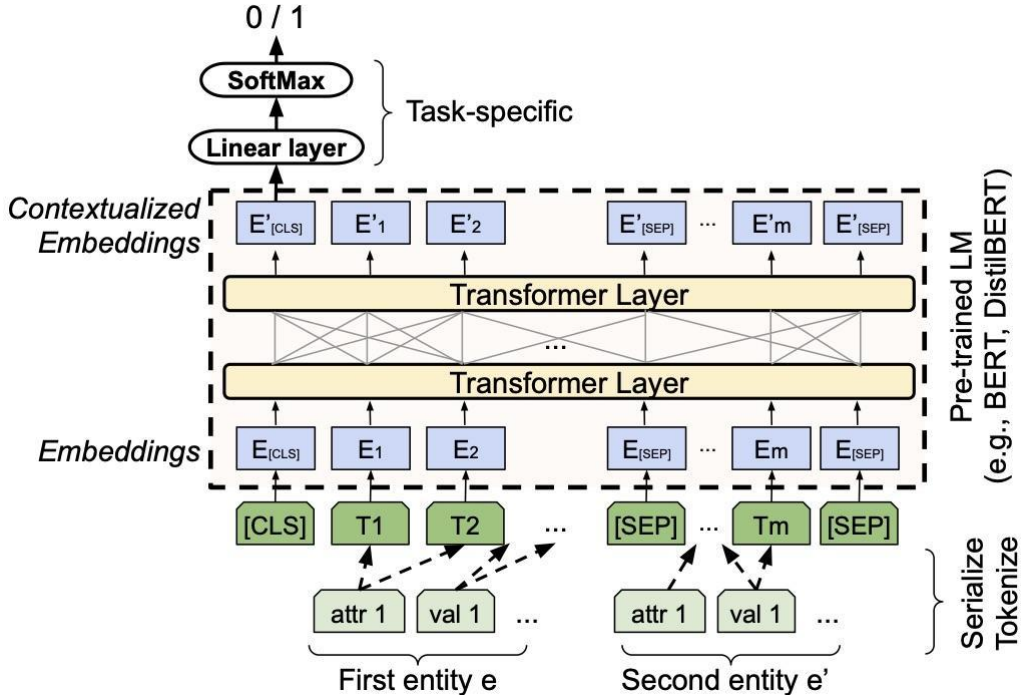
title	manf./modelno	price
instant immersion spanish deluxe 2.0	topics entertainment	49.99
adventure workshop 4th-6th grade 7th edition	encore software	19.99
sharp printing calculator	sharp el1192bl	37.63

Table B:

title	price
instant immers spanish dlux 2	36.11
encore inc adventure workshop 4th-6th grade 8th edition	17.1
new-sharp shr-el1192bl two-color printing calculator 12-digit lcd black red	56.0

[2004.00584] Deep Entity Matching with Pre-Trained Language Models

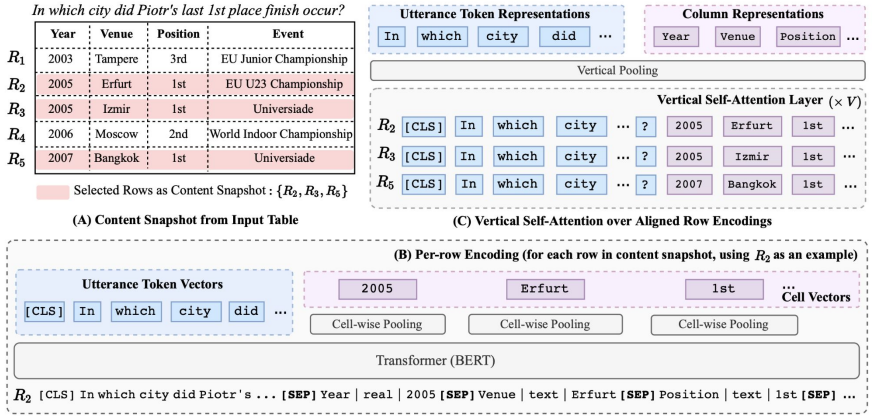
Pre-trained LM for Entity Matching

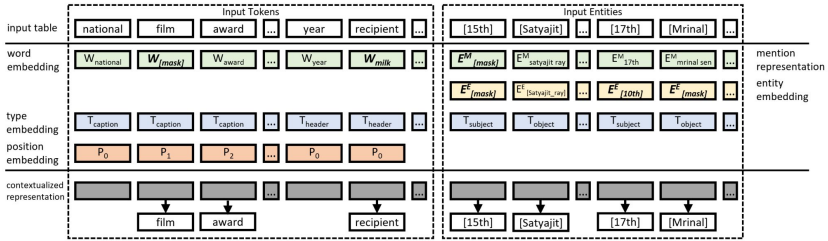


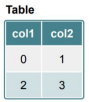
$\text{serialize}(e) ::= [\text{COL}]\ \text{attr}_1\ [\text{VAL}]\ \text{val}_1 \ldots [\text{COL}]\ \text{attr}_k\ [\text{VAL}]\ \text{val}_k$

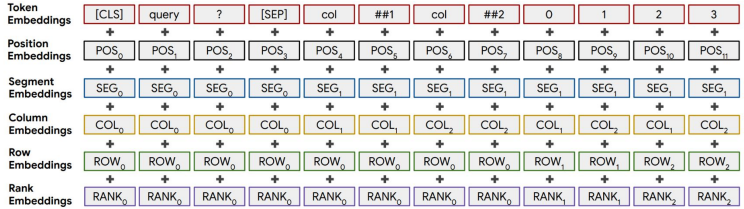
Pre-trained LMs for Table Data

- TaPas [ACL ‘20]
- TaBERT [ACL ‘20]
- TURL [VLDB ‘21]
- TABBIE [NAACL ‘21]
- …







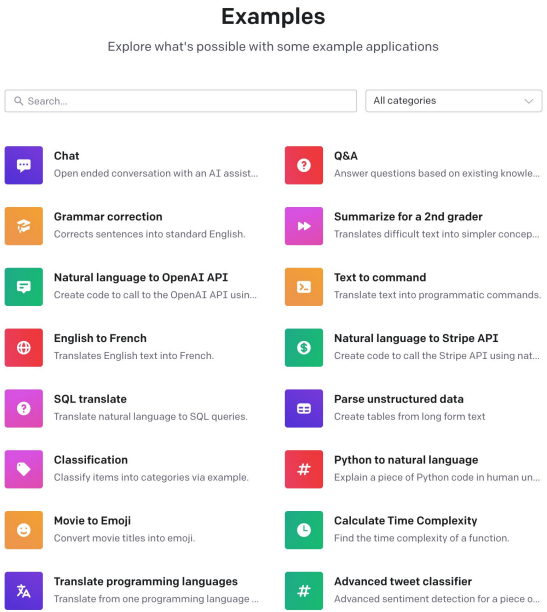


Break?

OpenAI GPT-3
Very Large Language Models

OpenAI GPT-3

- Yet another huge GPT model (175B param), which can be used for a wide variety of downstream tasks **without fine-tuning**
- Examples
  - https://beta.openai.com/examples

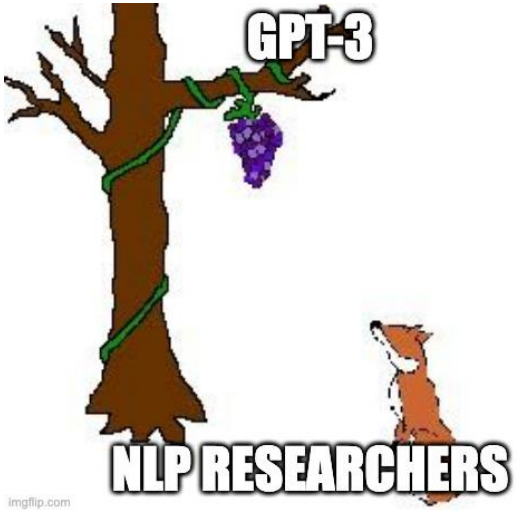


GPT-3 Hype

- OpenAI released GPT-3 API (almost only) to developers who are interested in creating applications using GPT-3
- Twitter timeline became full of GPT-3 advertisements
  - For example, https://twitter.com/sharifshameem/status/1282676454690451457?s=20

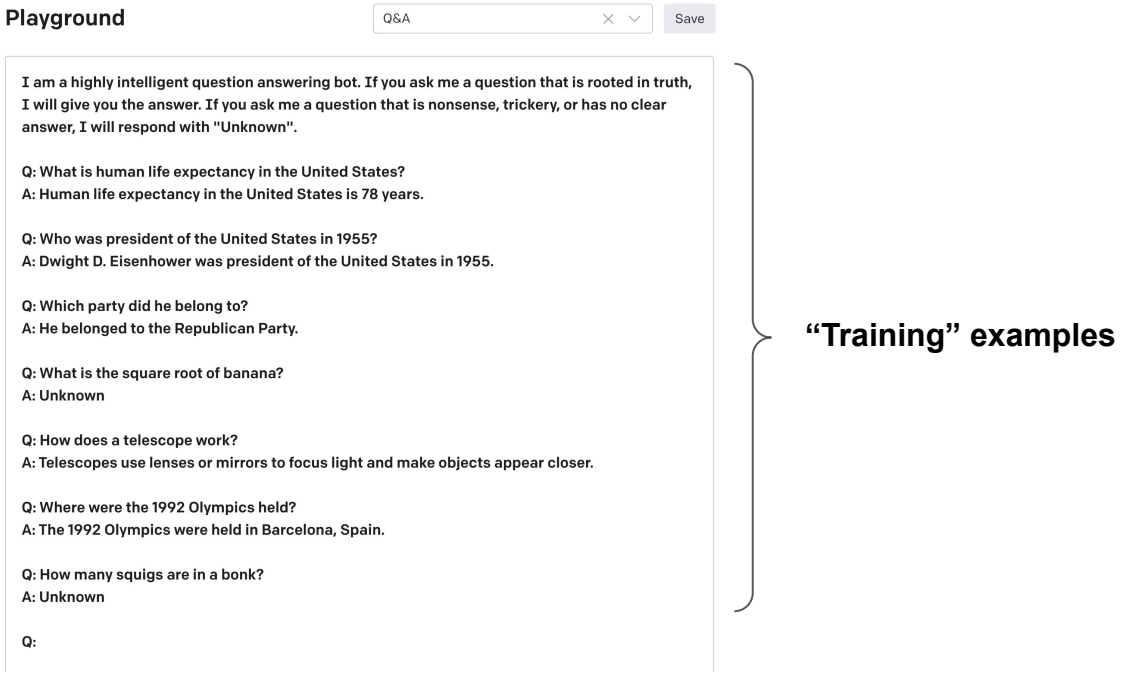
GPT-3 Hype

- OpenAI released GPT-3 API (almost only) to developers who are interested in creating applications using GPT-3
- Twitter timeline became full of GPT-3 advertisements
  - For example, https://twitter.com/sharifshameem/status/1282676454690451457?s=20



GPT-3 Takes “Training” Examples as Input

- Instead of updating the model parameters, GPT-3 **takes few training examples as input to condition** the model



GPT-3 Demos

https://gpt3demo.com/

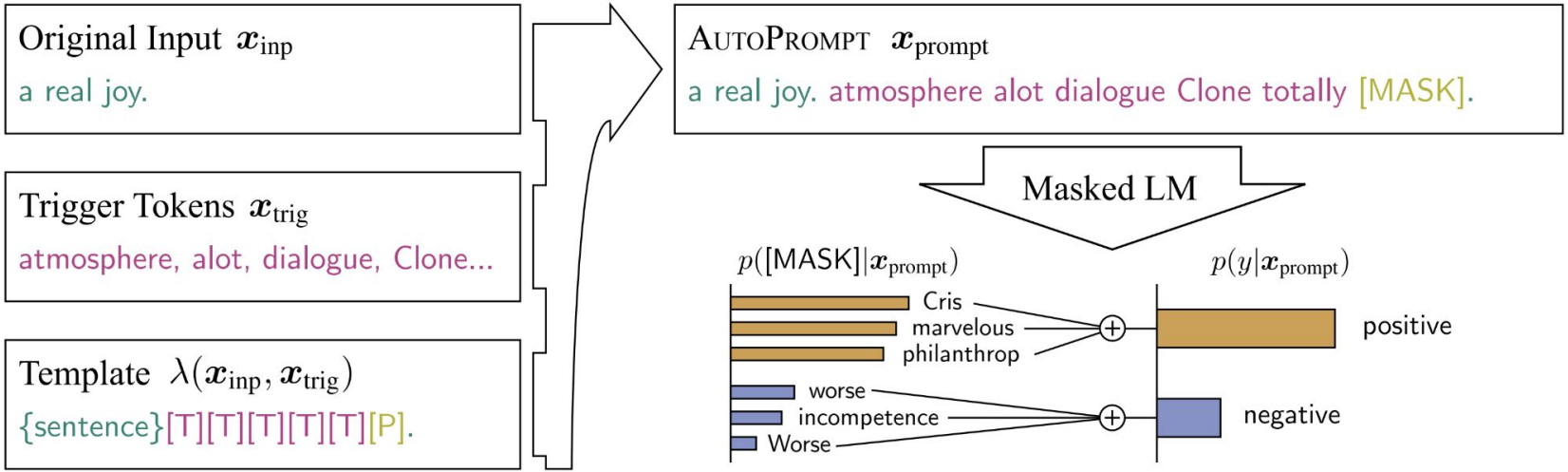
Future Direction & My Thoughts
Language Model as a Service

- Language models will keep getting larger and larger
  - So does “GPU resource” disparity :(
- Like GPT-3, those very large language models will be accessed via Web API
- Perhaps, we will stop training ML models sometime soon (?)

Prompt Learning

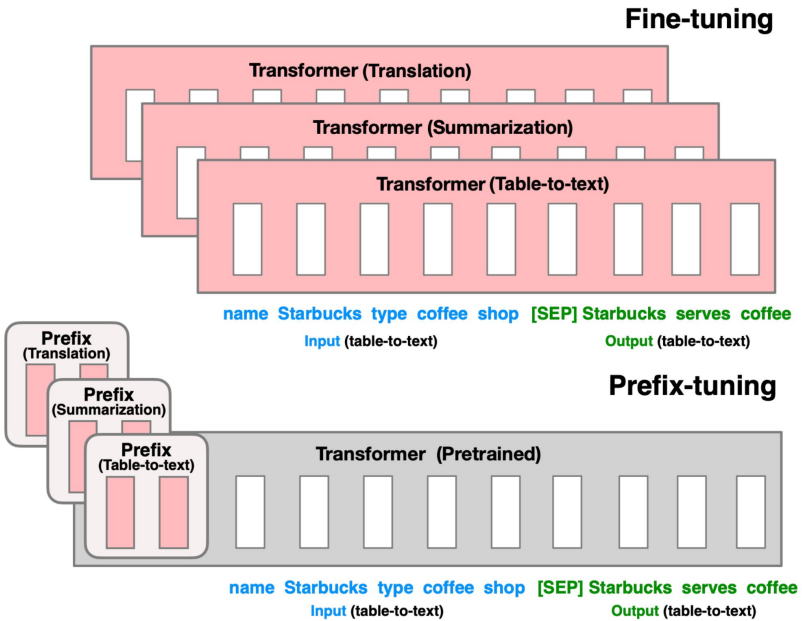
AutoPrompt

- Inserting a few **trigger tokens** that are helpful to improve the performance



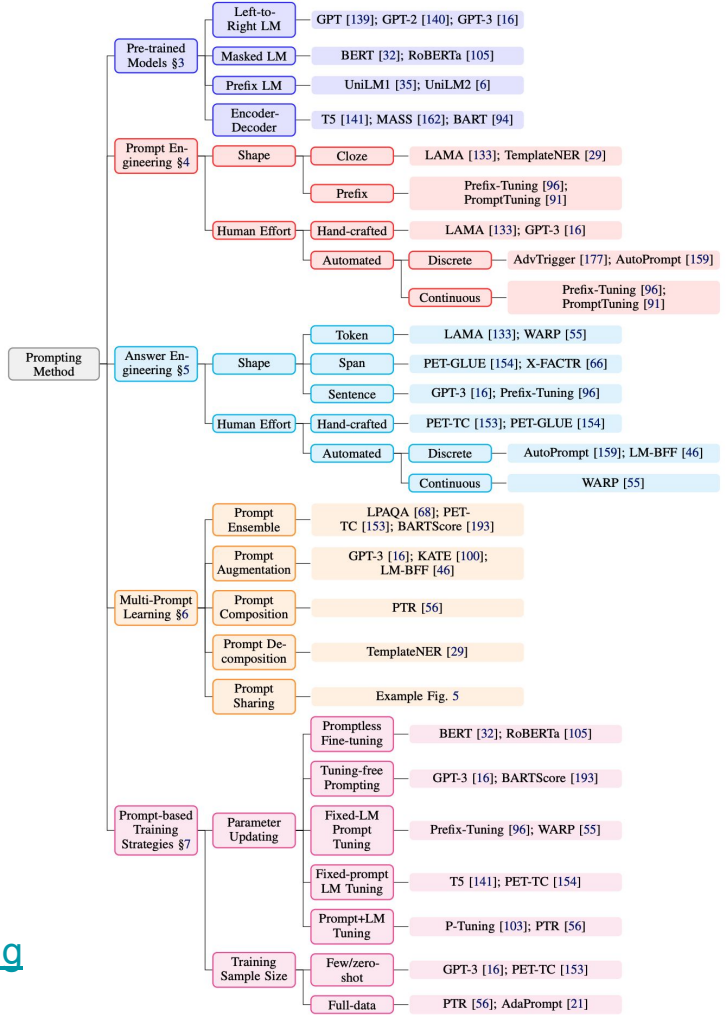
Prefix Tuning

- Only training a small number of **continuous vectors for each task** instead of fine-tuning the pre-trained language model



Very Hot Topic in 2021

Fine-tuning → Prompt learning



[2107.13586] Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing

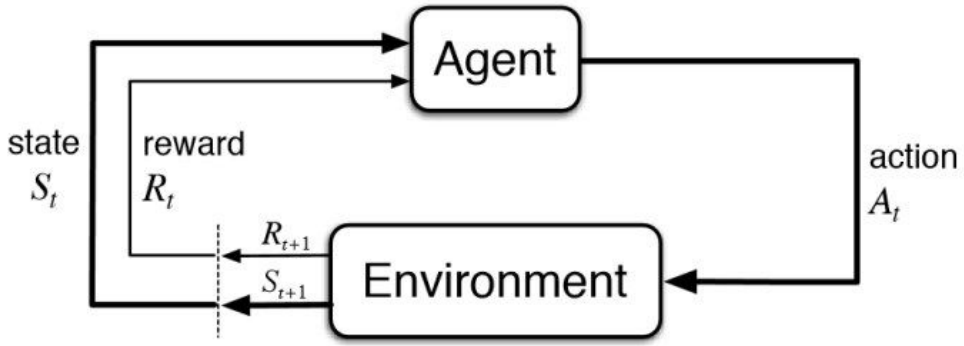
Other Topics Not Covered By This Course

Reinforcement Learning

- Powerful solution for robotics & agent development

What is Reinforcement Learning?

- Update the policy of the agent based on rewards from the environment
  - cf. optimizing the objective function with/without labels



This approach is useful especially when it's **difficult to define the success/failure** (i.e., true labels) in each step

Q-Learning: Example

- Move the agent (car) following the Q-table
- Update the Q-table once the agent receives a reward

Game Board:

Current state (*s*): 0 0 0 / 0 1 0

Q Table: γ = 0.95

	0 0 0 / 1 0 0	0 0 0 / 0 1 0	0 0 0 / 0 0 1	1 0 0 / 0 0 0	0 1 0 / 0 0 0	0 0 1 / 0 0 0
↑	0.2	0.3	1.0	-0.22	-0.3	0.0
↓	-0.5	-0.4	-0.2	-0.04	-0.02	0.0
→	0.21	0.4	-0.3	0.5	1.0	0.0
←	-0.6	-0.1	-0.1	-0.31	-0.01	0.0

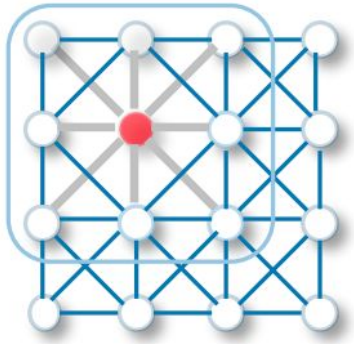
Is Reinforcement Learning Good for NLP tasks?

- Basically,
  - 1) we can define objective functions for NLP tasks
  - 2) we can tell if each of the system output is correct/incorrect

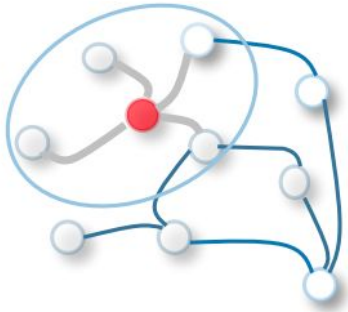
- What about chatbots (dialog management)?

Graph Neural Networks

- Essentially, neural networks with **graph convolution**



2d convolution

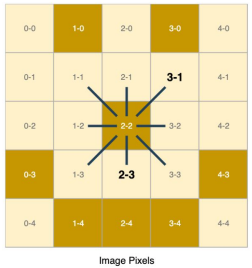


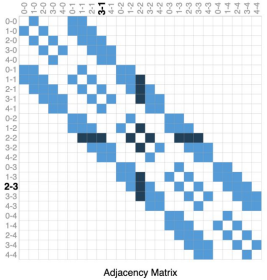
graph convolution

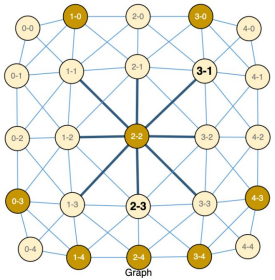
[Understanding Graph Convolutional Networks for Node Classification | by Inneke Mayachita]().

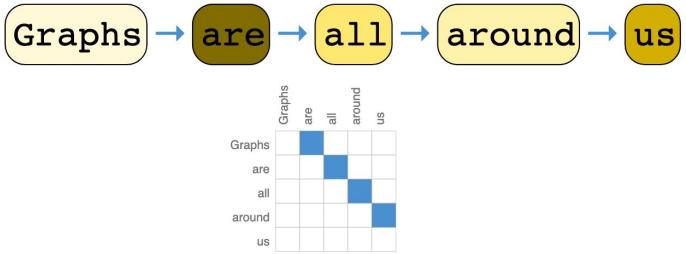
Graph Neural Networks for Image/Text Data

- What do you think?





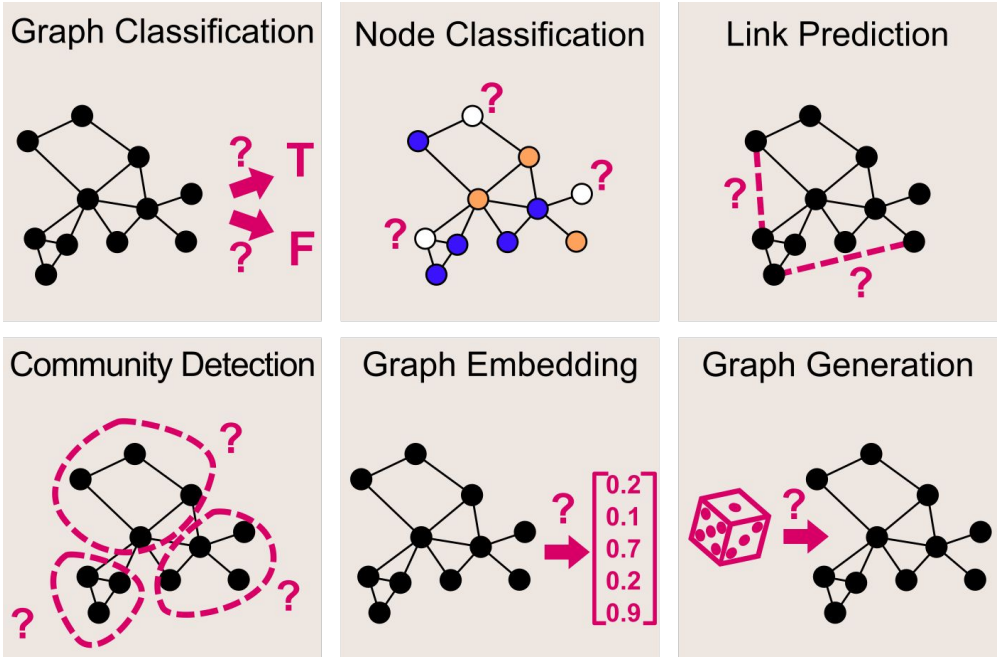




[A Gentle Introduction to Graph Neural Networks]

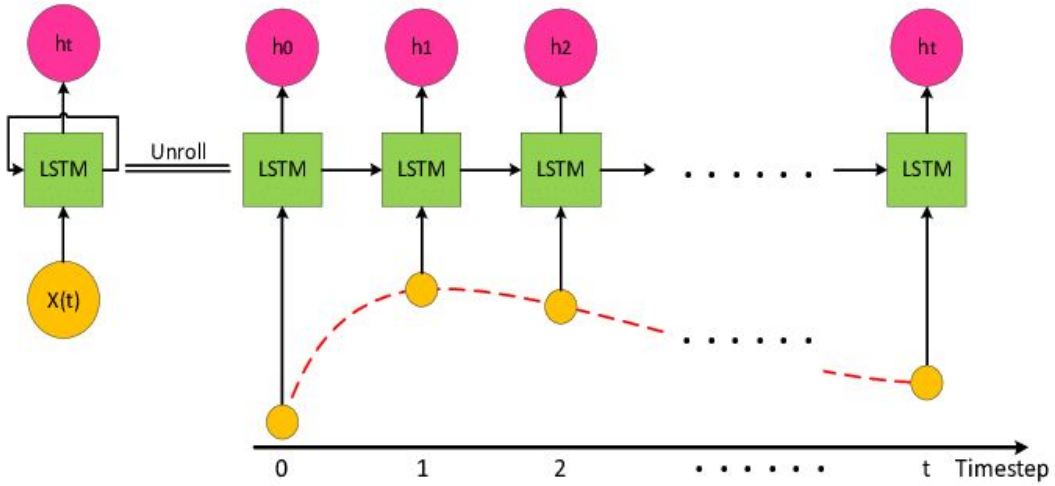
Graph Neural Networks

- A good solution for network analysis but **NOT for NLP/CV** in my opinion



Time Series Data

- Most time series problems can be formulated as **“many-to-one” or “many-to-many”-type** problems
- → RNNs or Transformers

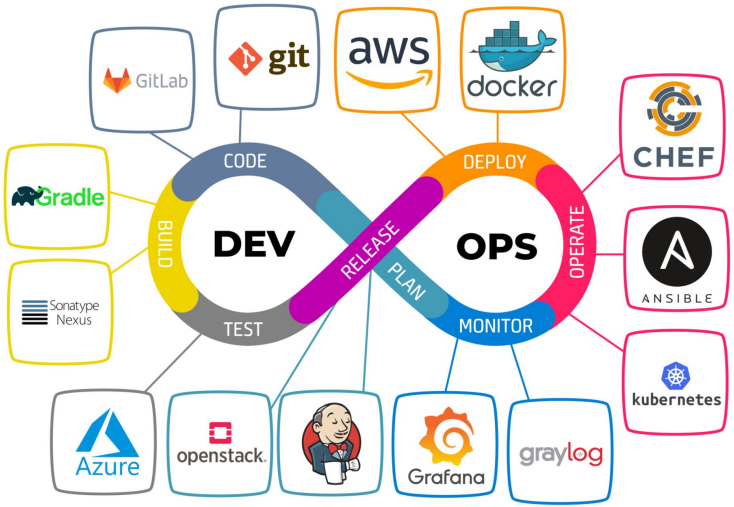


Figure

MLOps:
Real-world Machine Learning Systems

DevOps

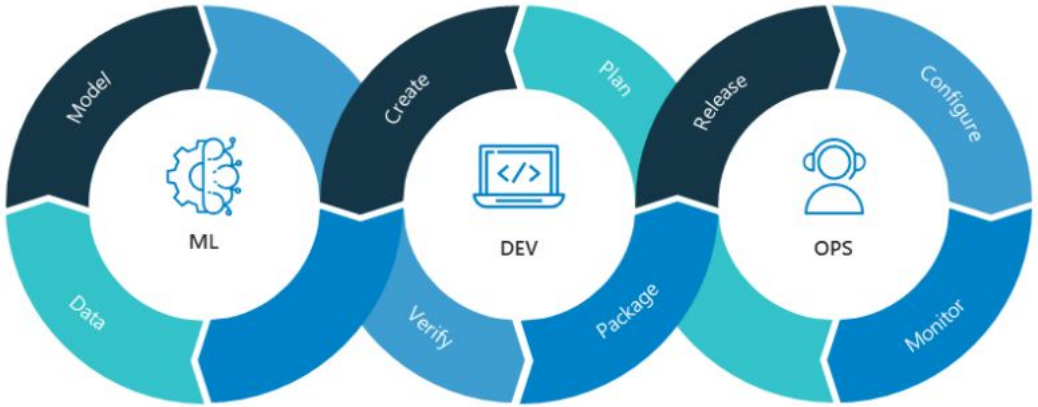
- A set of practices that combines software development (Dev) & system operations (Ops)
- To have a better software development cycle



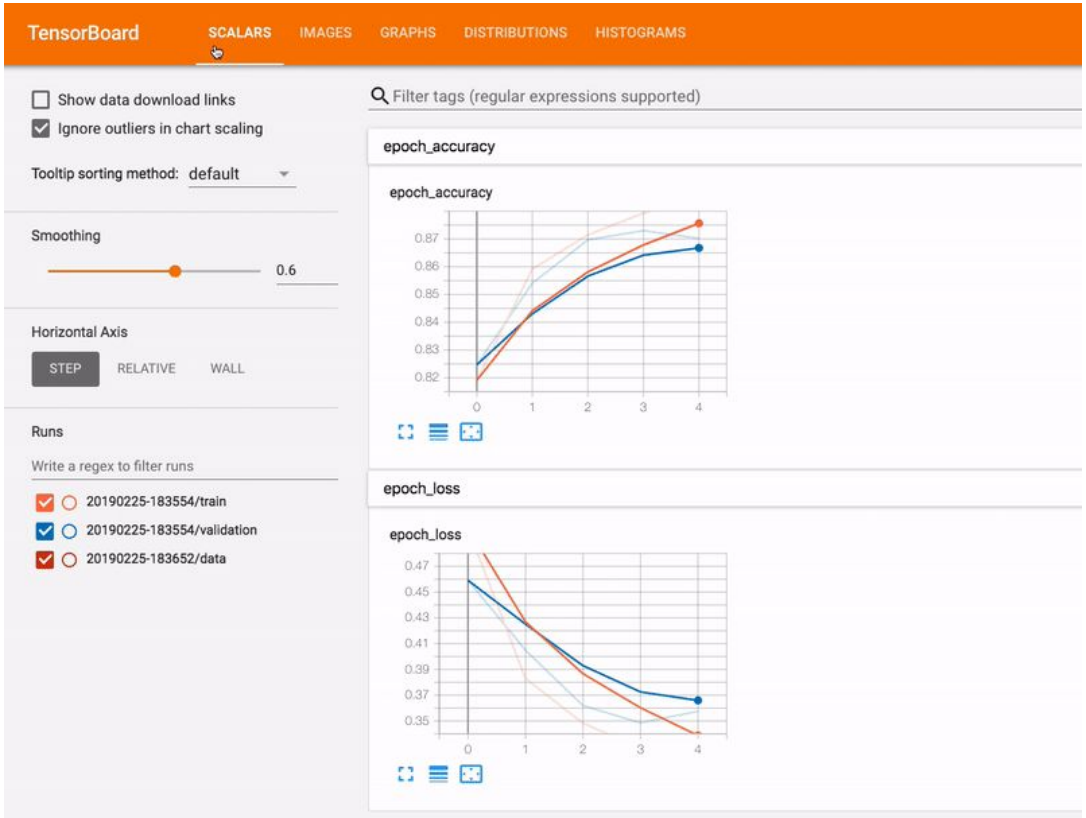
What is DevOps and where is it applied?

MLOps

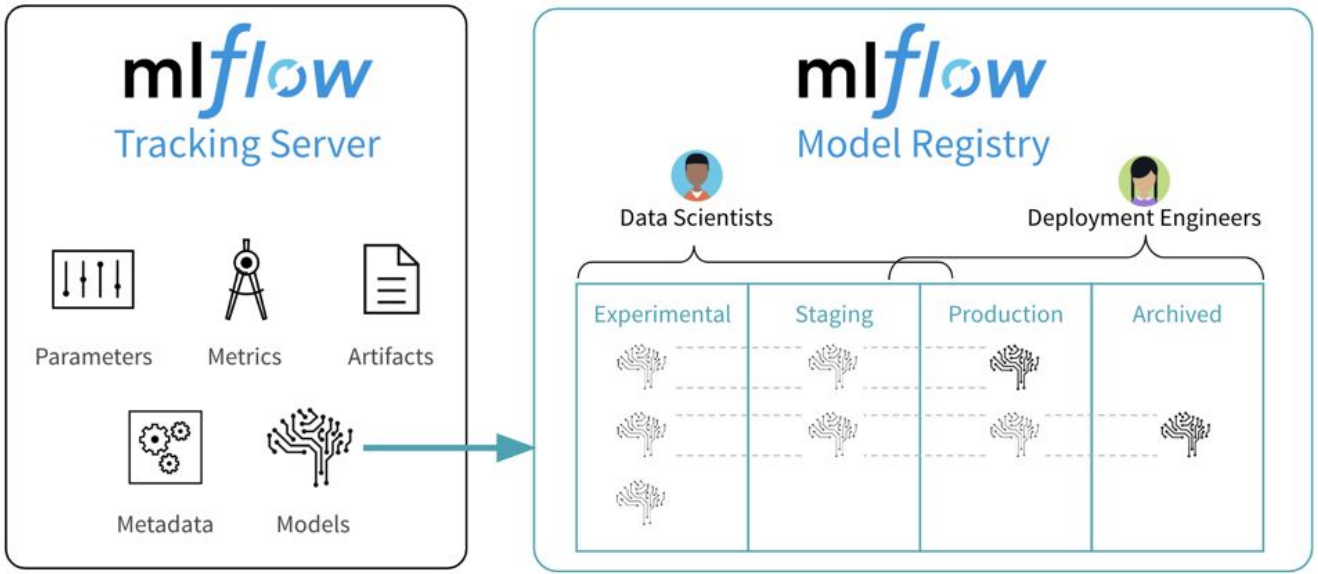
- DevOps for Machine Learning
- Key: Data & Model Management + Performance Monitor



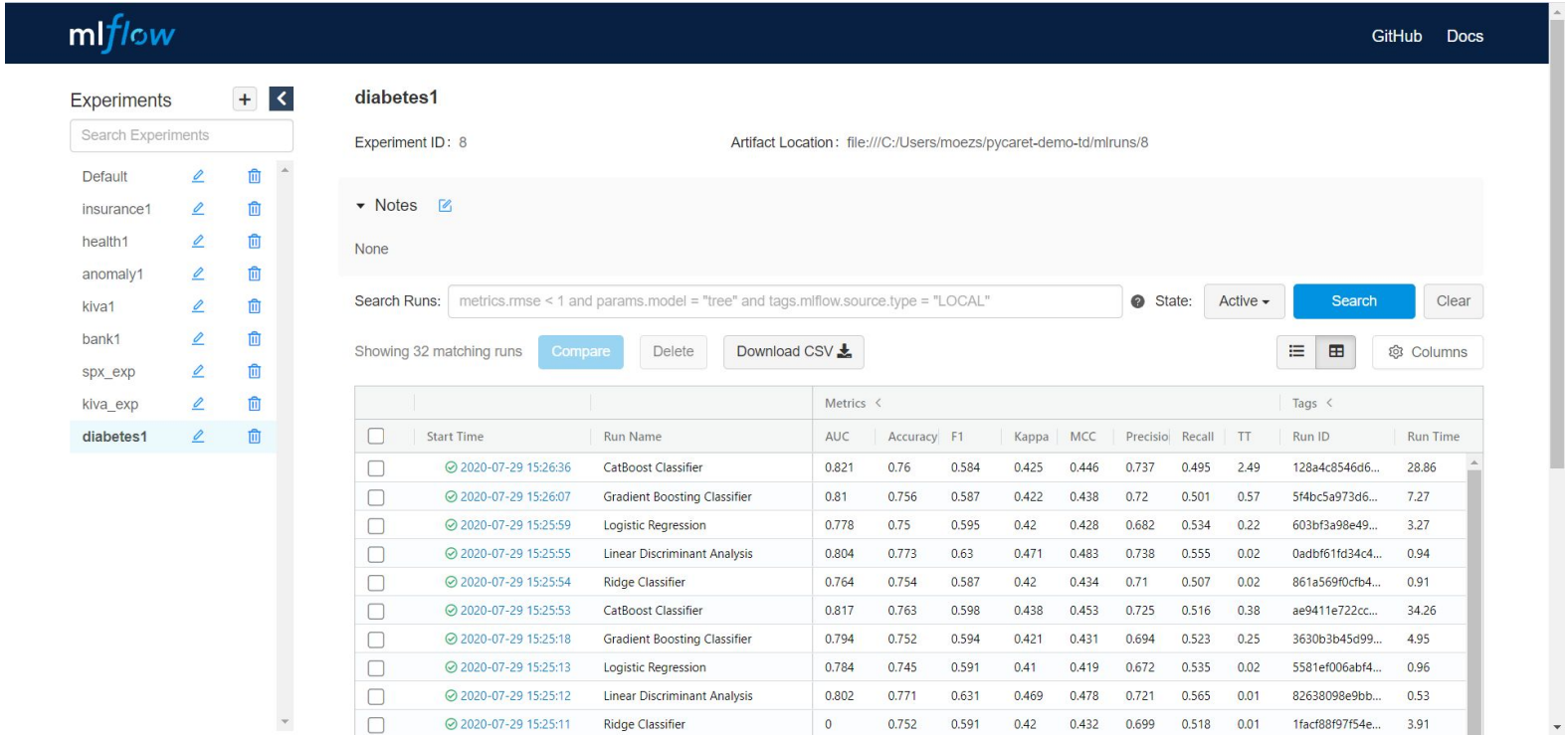
TensorBoard



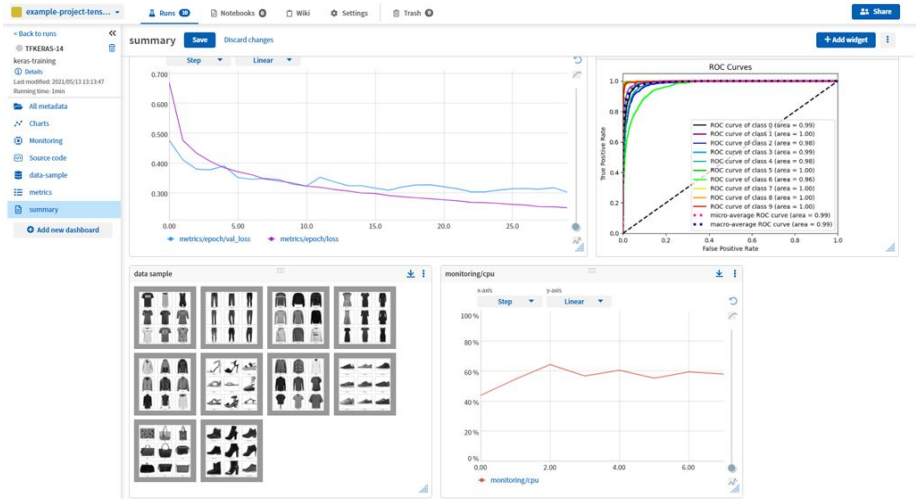
MLflow


mlflow
Tracking Server
Parameters
Metrics
Artifacts
Metadata
Models
mlflow
Model Registry
Data Scientists
Deployment Engineers
Experimental
Staging
Production
Archived

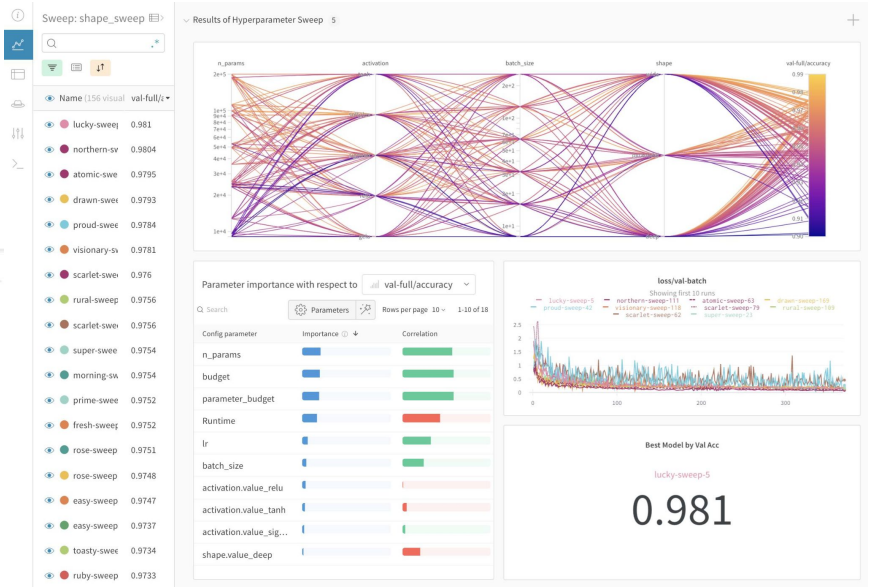
MLflow: Tracking Server



Online MLOps Platforms



Neptune.ai



Weight and Biases

Topics Not Covered But Important

- Include but not limited to…
  - Virtualization frameworks (e.g., Docker, Kubernetes)
  - Cloud services (e.g., AWS, Google Cloud, Azure)
  - ML management platforms (e.g., MLflow, WandB, Neptune)
  - Model configuration management tools (e.g., Hydra)
  - ...

- Create your own template!
  - For example, https://github.com/ashleve/lightning-hydra-template

~~A journey of a thousand miles begins with a single step~~

A system of a thousand models beings with a single line of code

PyTorch Lightning

PyTorch Lightning

PYTORCH

```
# models
encoder = nn.Sequential(nn.Linear(28 * 28, 64), nn.ReLU(), nn.Linear(64, 3))
decoder = nn.Sequential(nn.Linear(3, 64), nn.ReLU(), nn.Linear(64, 28 * 28))

encoder.cuda(0)
decoder.cuda(0)

# download on rank 0 only
if global_rank == 0:
    mnist_train = MNIST(os.getcwd(), train=True, download=True)

# split dataset
transform=transforms.Compose([transforms.ToTensor(),
                              transforms.Normalize(0.5, 0.5)])
mnist_train = MNIST(os.getcwd(), train=True, download=True, transform=transform)

# train (55,000 images), val split (5,000 images)
mnist_train, mnist_val = random_split(mnist_train, [55000, 5000])

# The dataloaders handle shuffling, batching, etc...
mnist_train = DataLoader(mnist_train, batch_size=64)
mnist_val = DataLoader(mnist_val, batch_size=64)

# optimizer
params = [encoder.parameters(), decoder.parameters()]
optimizer = torch.optim.Adam(params, lr=1e-3)

# TRAIN LOOP
model.train()
num_epochs = 1
for epoch in range(num_epochs):
  for train_batch in mnist_train:
    x, y = train_batch
    x = x.cuda(0)
    x = x.view(x.size(0), -1)
    z = encoder(x)
    x_hat = decoder(z)
    loss = F.mse_loss(x_hat, x)
    print('train loss: ', loss.item())

    loss.backward()
    optimizer.step()
    optimizer.zero_grad()

# EVAL LOOP
model.eval()
with torch.no_grad():
  val_loss = []
  for val_batch in mnist_val:
    x, y = val_batch
    x = x.cuda(0)
    x = x.view(x.size(0), -1)
    z = encoder(x)
    x_hat = decoder(z)
    loss = F.mse_loss(x_hat, x)
    val_loss.append(loss)
  val_loss = torch.mean(torch.tensor(val_loss))
  model.train()
```

PYTORCH LIGHTNING

Turn PyTorch into Lightning

Lightning is just plain PyTorch

Hands-on Session: PyTorch Lightning

- Google Colab

Summary

- Advanced topics & Future directions
  - Data Augmentation
  - Adversarial Attacks; Interpretability & Explainability
  - Multilingual Models
  - Vision & Language
  - Pre-trained Language Models for Data Integration
  - Open AI GPT-3
  - Prompt Learning
- Topic that are not covered by the course
  - Reinforcement Learning
  - Graph Neural Networks
  - Time Series Data
  - MLOps

Project Presentation (5min + 3min QA)
Tue 11/23 & Tue 11/30

- All students must attend both of the sessions
- Join the zoom link on time (I'll start the zoom 5 minutes before the session)
- The first presentation will **start at 2pm sharp**

- Presenters will share their screens to make presentations (Camara On)
- Presentations will be recorded