DSCI 565: INTRODUCTION TO DEEP LEARNING FOR DATA SCIENCE

This content is protected and may not be shared, uploaded, or distributed.

Ke-Thia Yao

Lecture 1, 2025 August 25

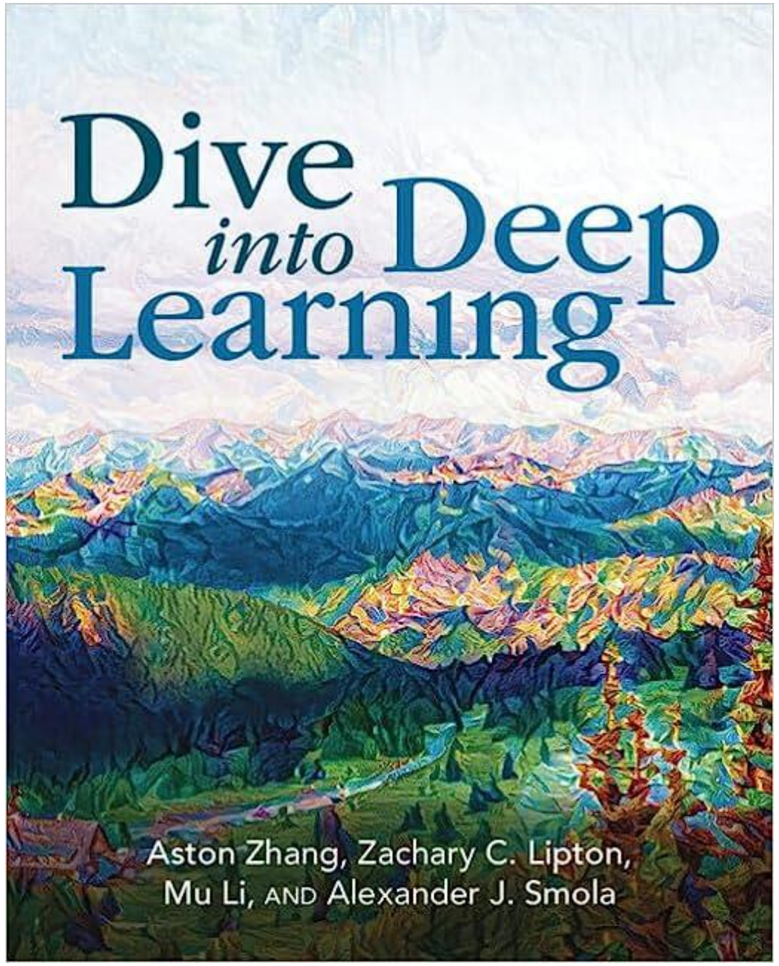
Learning Objective

- After successful completion of this course, students will be able to:
  - Understand the principles of deep learning and deep neural networks
  - Know the suitability of specific deep learning algorithms to various data domains, such as images, text, and graphs
  - Design, implement and train deep neural models
  - Perform regularization, training optimization and hyperparameter tuning on deep learning models

Learning Approach

- DSCI 565 takes a practitioner's approach with emphasis on intuition about how to use deep learning algorithms with different types of data for different domains
- Following the philosophy of the course textbook Dive into Deep Learning this course adopts a learning-by-doing method,
- Which involves first introducing deep learning concepts and then providing exercises on real datasets.

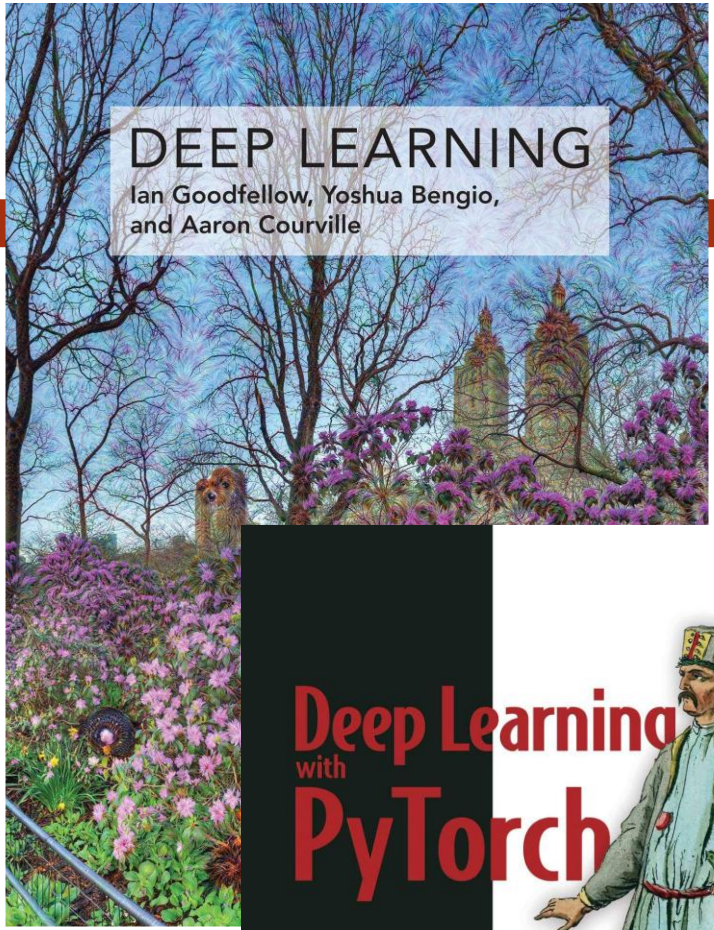
Textbook

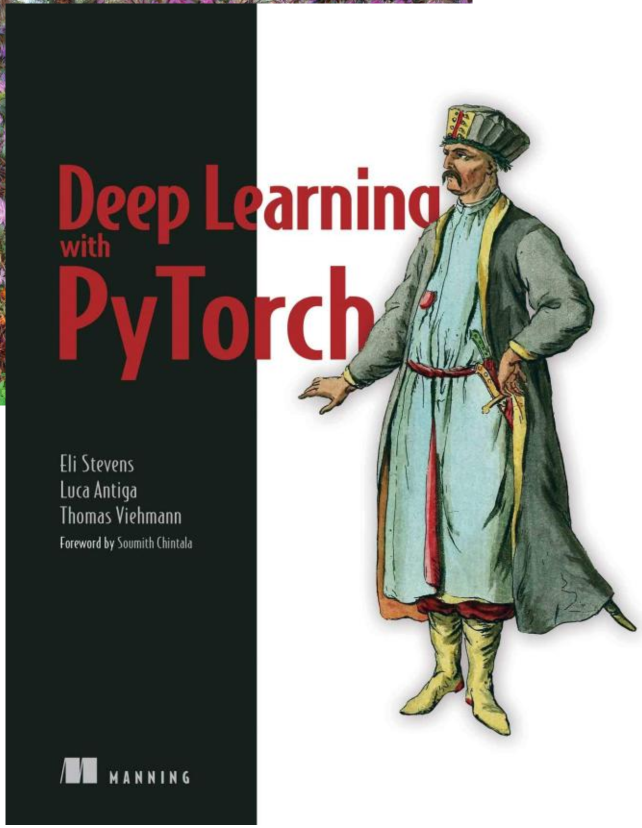


- Zhang, Aston, Zachary C. Lipton, Mu Li, and Alexander J. Smola. 2023. *Dive into Deep Learning*. 1st edition. Cambridge University Press.
- Latest version available online: https://d2l.ai/d2l-en.pdf
- Github: https://github.com/d2l-ai/d2l-en

Optional Textbooks

- Goodfellow, Ian and Bengio, Yoshua and Courville, Aaron (2016)
  - Available online: https://www.deeplearningbook.org/
- Stevens , Eli and Antiga, Luca and Viehmann, Thomas (2020)
  - Available online through USC library





Office Hours

- Ke-Thia Yao, kyao@isi.edu
  Mondays and Wednesdays, and by appointment
  3PM-4PM
  PHE 514
- TA: Dylan Rowe, dylanr@usc.edu
  Tuesdays
  2PM-3PM
  Zoom: https://usc.zoom.us/j/91081706457?pwd=eLb6NsLSJQ75MavsTwz4FgEMF8oZqa.1

Grading Breakdown

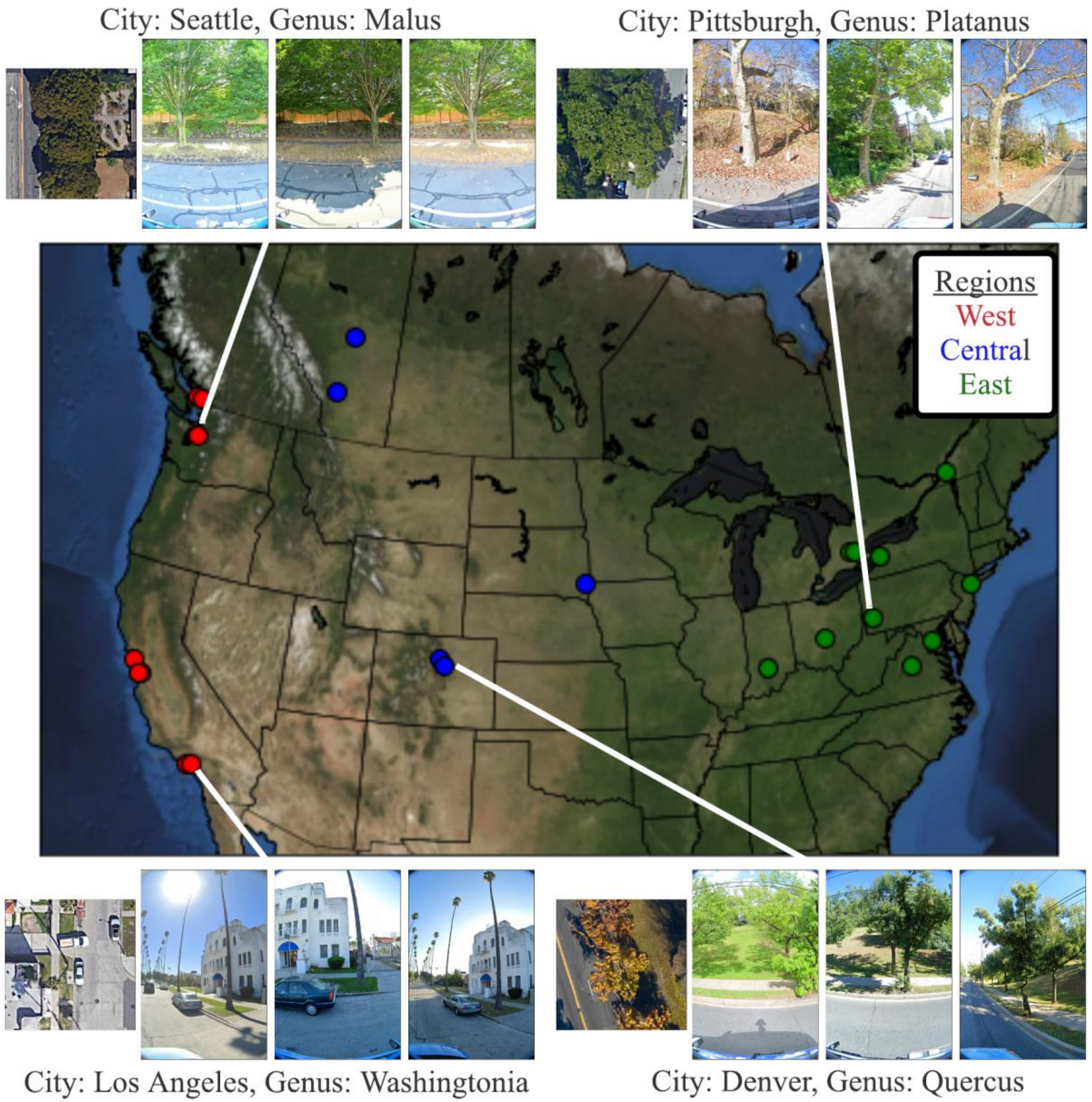
Assessment Tool (assignments)	% of Grade
Midterm Exam (Week 9)	30%
Homework/Programming Assignments	35%
Class Participation	5%
Semester Project	30%
TOTAL	100%

Semester Project

- Purpose of the project is for you to practice applying deep learning algorithms and models to challenging learning problems
- Ideas for projects
  - Look at recent papers from machine learning conferences, such as NeurIPS, ICML, CVPR. Papers usually comes Git repositories with implementations
  - Look at Hugging Face for pretrained deep models
  - Look at available dataset and benchmark datasets
    - NeurIPS 2022 Datasets and Benchmarks https://nips.cc/virtual/2022/events/datasets-benchmarks-2022
    - CPVR 2022 Dataset Contributions https://cvpr2022.thecvf.com/dataset-contributions

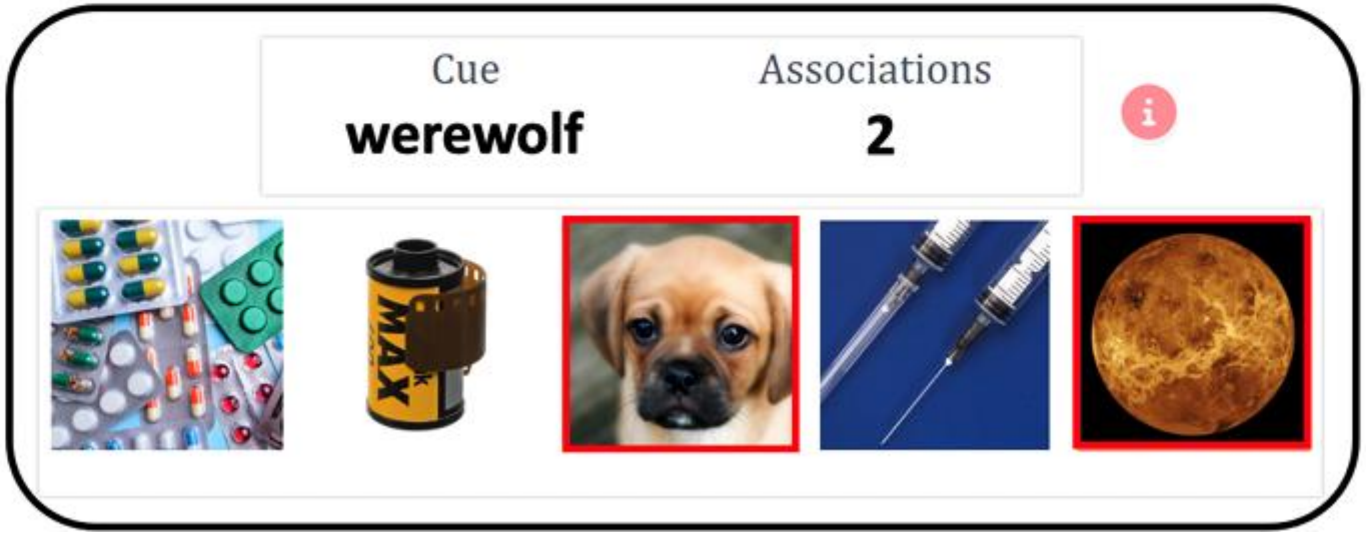
Example Dataset: The Auto Arborist Dataset

- A Large-Scale Benchmark for Multiview Urban Forest Monitoring Under Domain Shift
- We propose urban forest monitoring as an ideal testbed for working on several computer vision challenges
  - domain generalization, fine-grained categorization, long-tail learning, multiview vision
  - while working towards filling a crucial environmental and societal need.



Example Dataset: WinoGAViL

Association Instance



- Gamified Association Benchmark to Challenge Vision-and-Language Models
- Given text cue find appropriate image associations

Physics Inspired ML: Solver-in-the-Loop

Solver-in-the-Loop: Learning from Differentiable Physics to Interact with Iterative PDE-Solvers, Um et. al. 2020

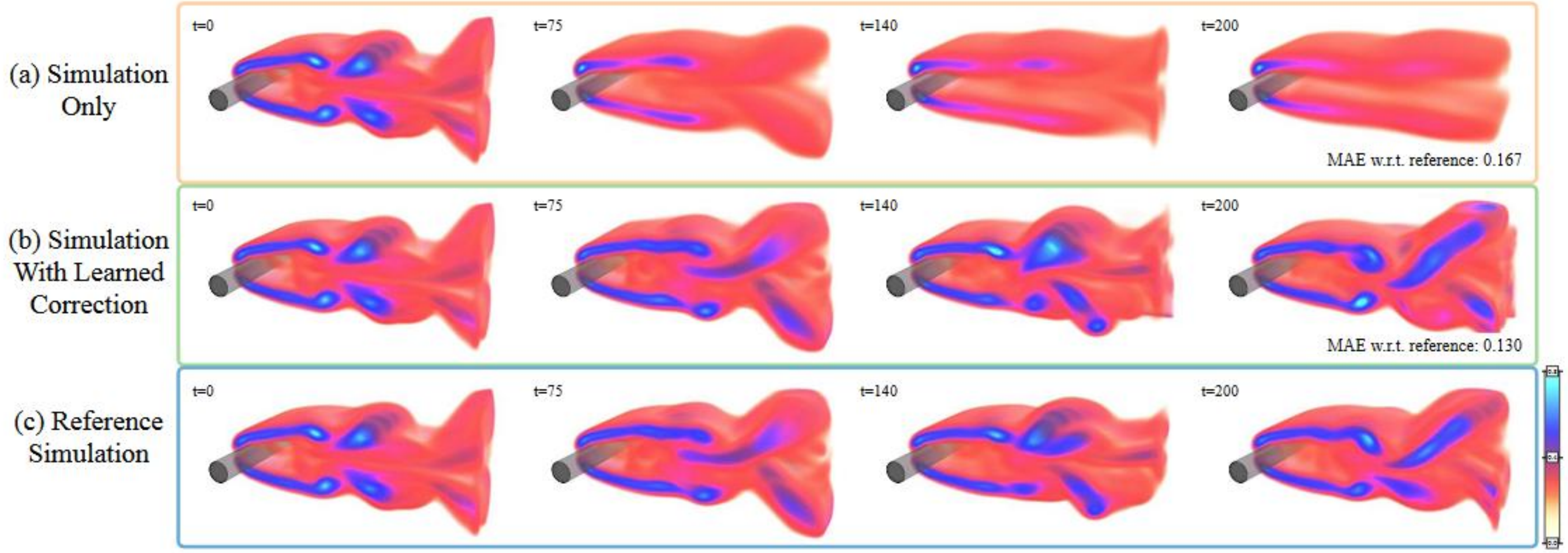


Figure 1: A 3D fluid problem (shown in terms of vorticity) for which the regular simulation introduces numerical errors that deteriorate the resolved dynamics (a). Combining the same solver with a learned corrector trained via differentiable physics (b) significantly reduces errors w.r.t. the reference (c).

Coarse mesh simulation

Coarse mesh w/ correction

Fine mesh simulation

Semester Project Timeline

- Week 3: Identifying team members and project topics
- Week 6: Proposal due (team member, topics and milestone)
- Week 8: Mid-term report due (data description, preliminary results)
- Week 15: Project presentation and Poster session (open to all faculty and students)
- Finals week: Final report due (task and model description, major discovery, lessons learned)

Semester Project Grading Breakdown

- All aspects of the project combined are 30% of the semester grade, with a breakdown:
  - Proposal: 5%
  - Mid-term report: 5%
  - Final report and presentation: 20%

Sample Project from Previous Semesters

- Multi-Modal Fusion and Lightweight Enhancement for Sequential Recommendation
  - Combines text and images for product recommendation
- Training Deep Language Models for Tweets Sentiment Analysis
  - Compares LSTM, BERT and BERT finetuned with LoRA
- Cross-Generative Model Generalization: A Study on Swin Transformer and U-Net for AI-Generated Image Classification
  - Fake image detection on unseen generative models
- Transforming Image Captioning: Integrating SwinV2, CSwin, and DeiT Architectures into the Pure Transformer (PureT) Model
  - Replicates PureT paper results, and improve by integrating with other backbones
- Natural Language Processing to Reduce Racial Bias in Diagnosis of Mental Health Disorders
  - Compares knowledge-intensive approach against BERT, MentalBERT, Bio_ClinicalBERT

Sample of Deep Learning Systems

Dimension Reduction:
Restricted Boltzmann Machine (RBM) Autoencoders

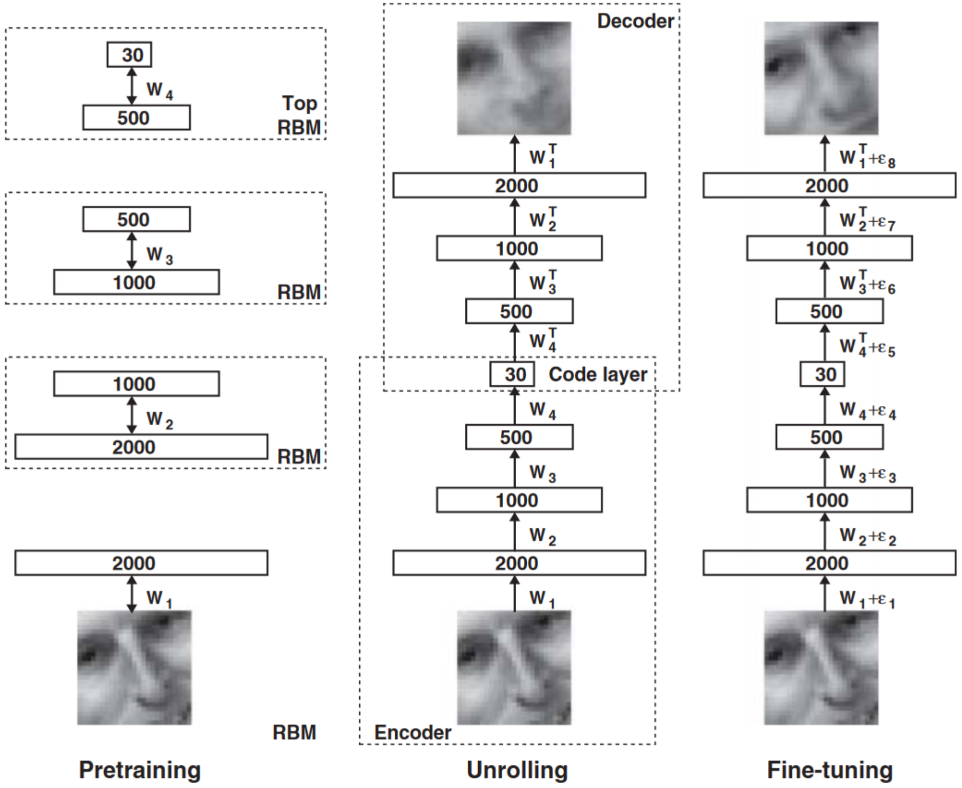
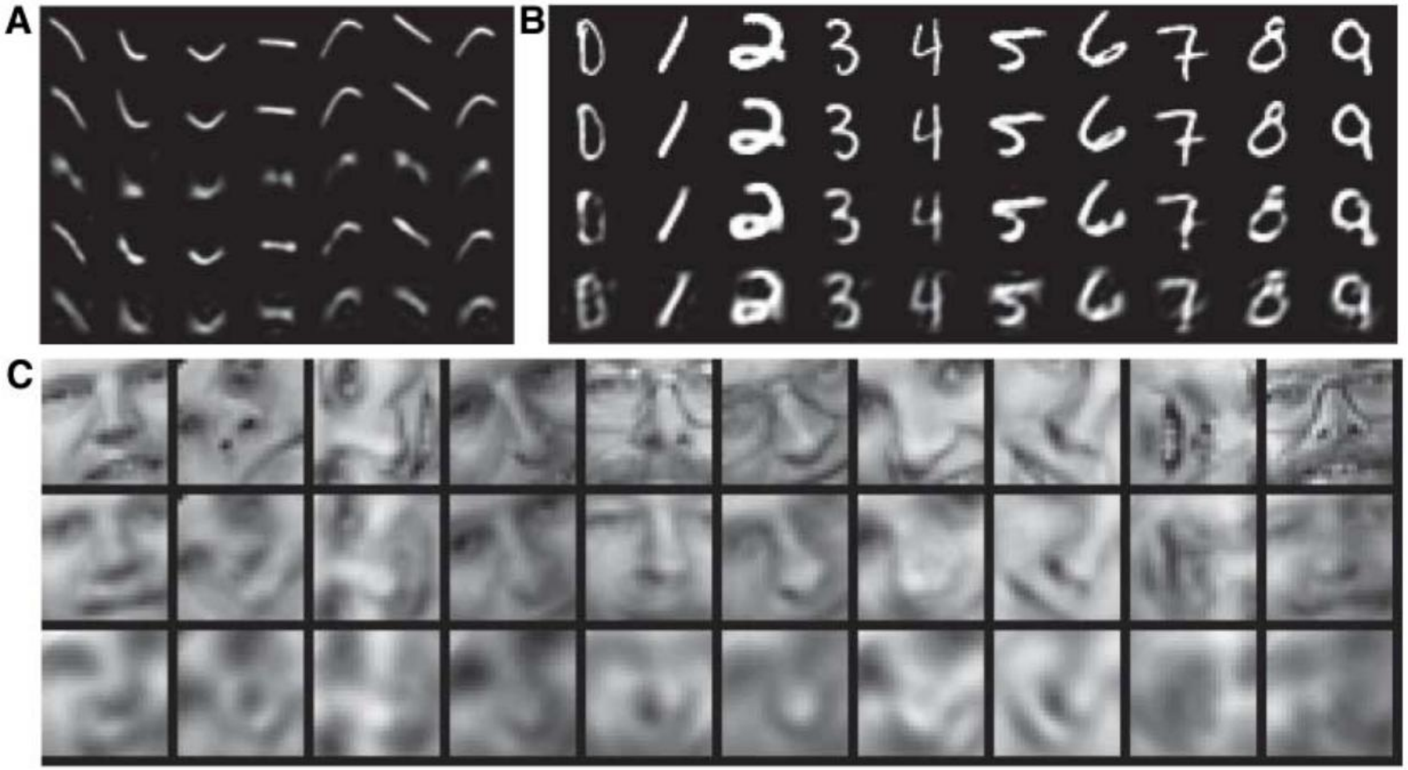


Fig. 1. Pretraining consists of learning a stack of restricted Boltzmann machines (RBMs), each having only one layer of feature detectors. The learned feature activations of one RBM are used as the "data" for training the next RBM in the stack. After the pretraining, the RBMs are "unrolled" to create a deep autoencoder, which is then fine-tuned using backpropagation of error derivatives.

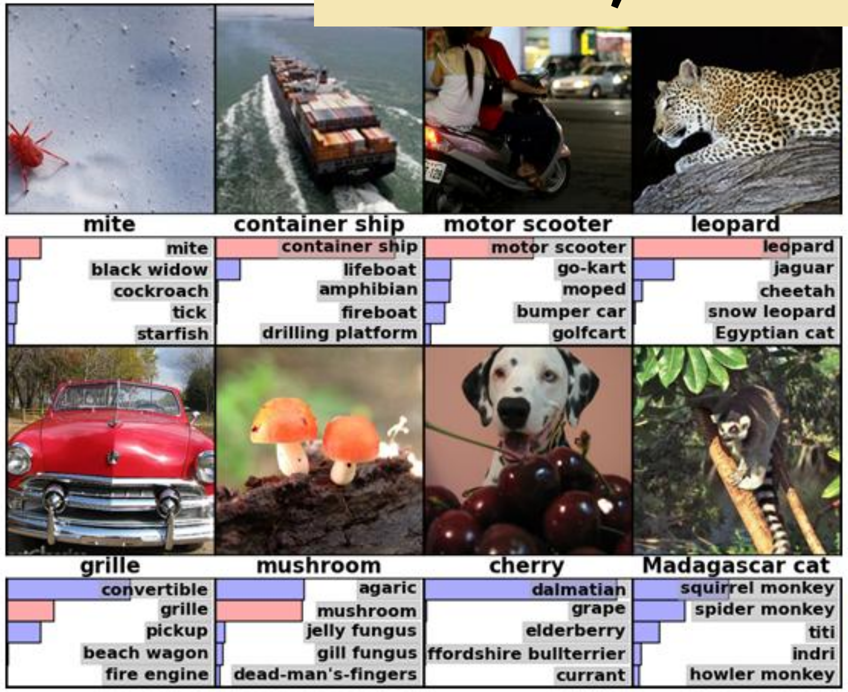


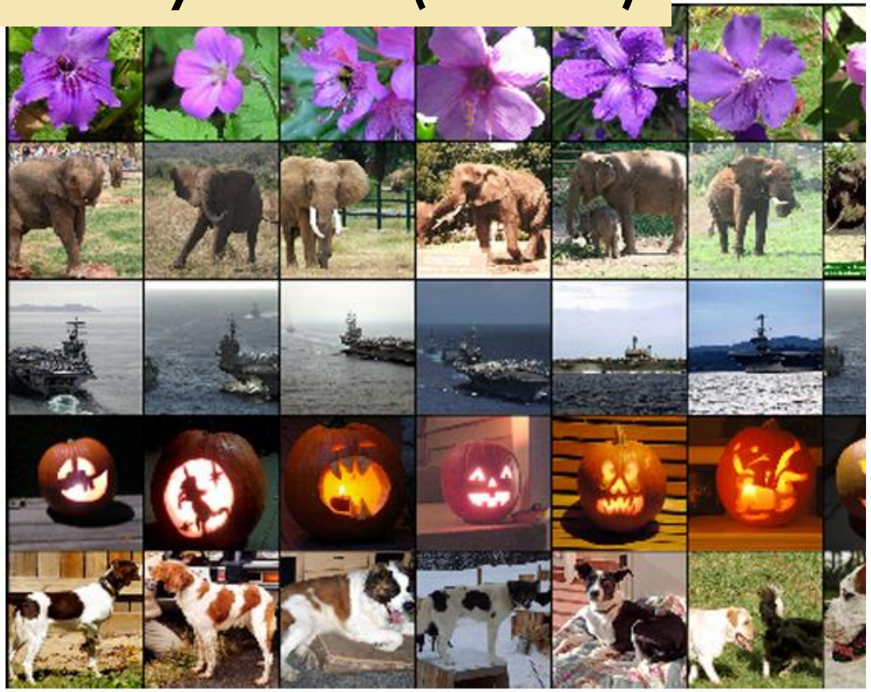
Hinton and Salakhutdinov (2008)

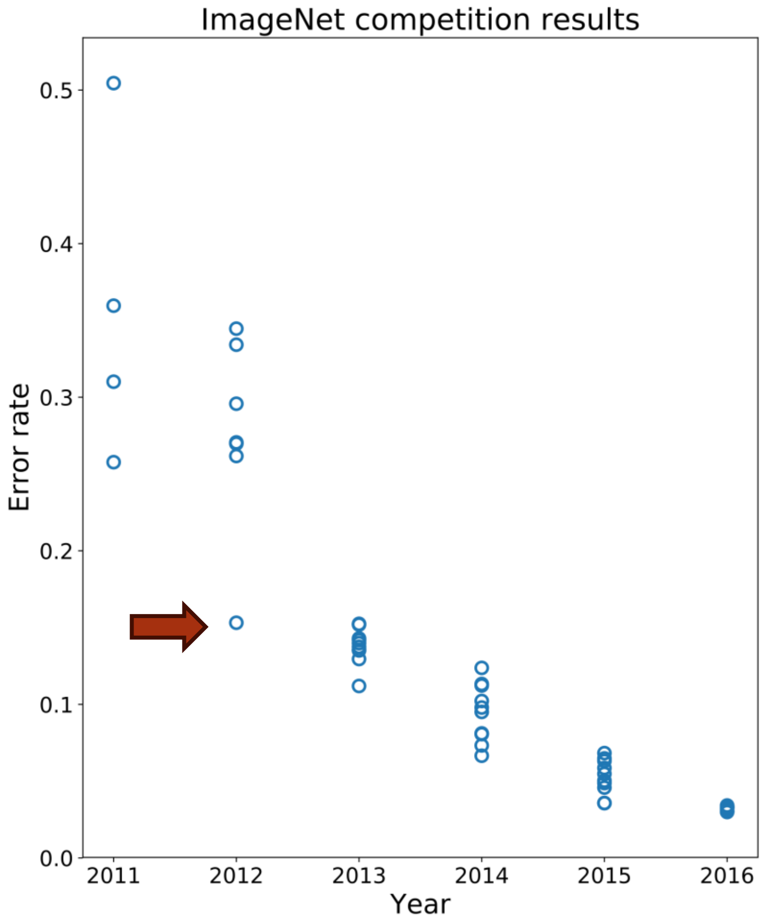
Image Classification: ImageNet Competition

- Classify image into one of 1000 classes
- >1million training images

AlexNet, Kerizhevsky et al (2012)



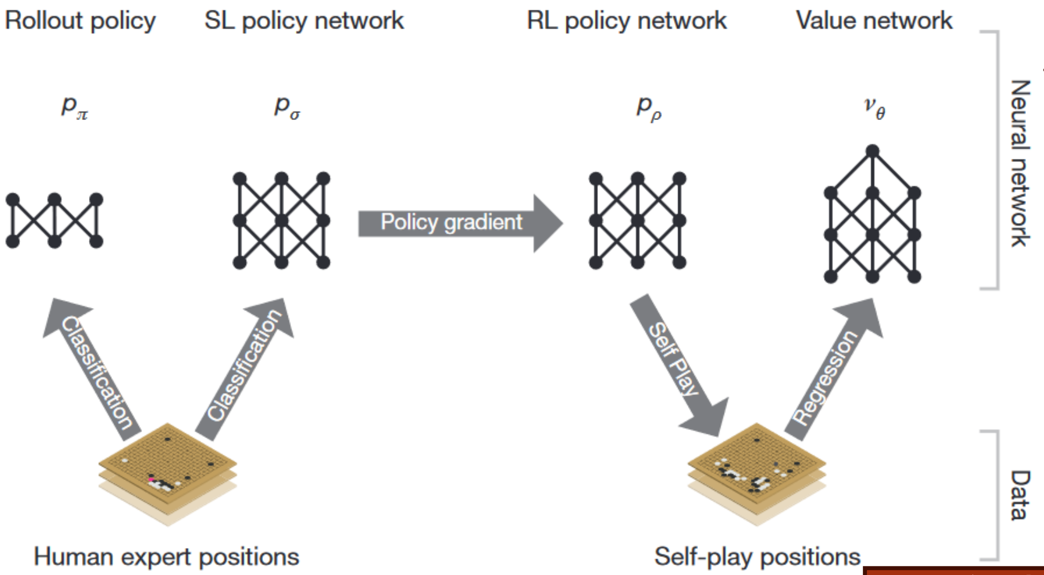




https://en.wikipedia.org/wiki/ImageNet

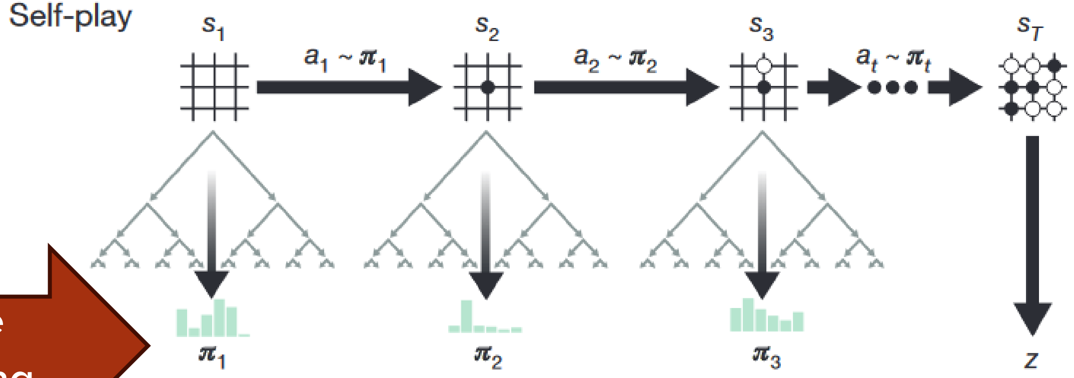
Reinforcement Learning: AlphaGo, AlphaGo Zero

- 2015: AlphaGo won Fan Hui, European champion
- 2016: AlphaGo won Lee Sedol, one of the strongest player
- 2017: AlphaGo won Ke Jie, the top-ranked player
- 2017: AlphaGo Zero beats AlphaGo 100 to 0 games



AlphaGo: Initiate with Supervised Learning

AlphaGo Zero: Pure Reinforcement Learning



Generative Adversarial Network (GAN)

Goodfellow et al (2014)

StyleGAN, Karras et al (2019)

Diffusion Models: Text-to-Image Generation

Stable Diffusion

- Prompt:
  A photograph of an astronaut riding a horse
- Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. "High-Resolution Image Synthesis with Latent Diffusion Models." arXiv. http://arxiv.org/abs/2112.10752.

Deep Mind: Learning to Simulate Complex Physics with Graph Networks

- Sanchez-Gonzalez et al 2020, ICML
- Learn to simulate how fluids, rigid solids and deformable materials interacting with one another
- Learn by looking a few steps of SPH, then try to predicate the next step
- Represented particles (nodes) and particle relationships (edges) as embeddings of
  - Positions, velocities, material properties
  - Displacements, spring constants

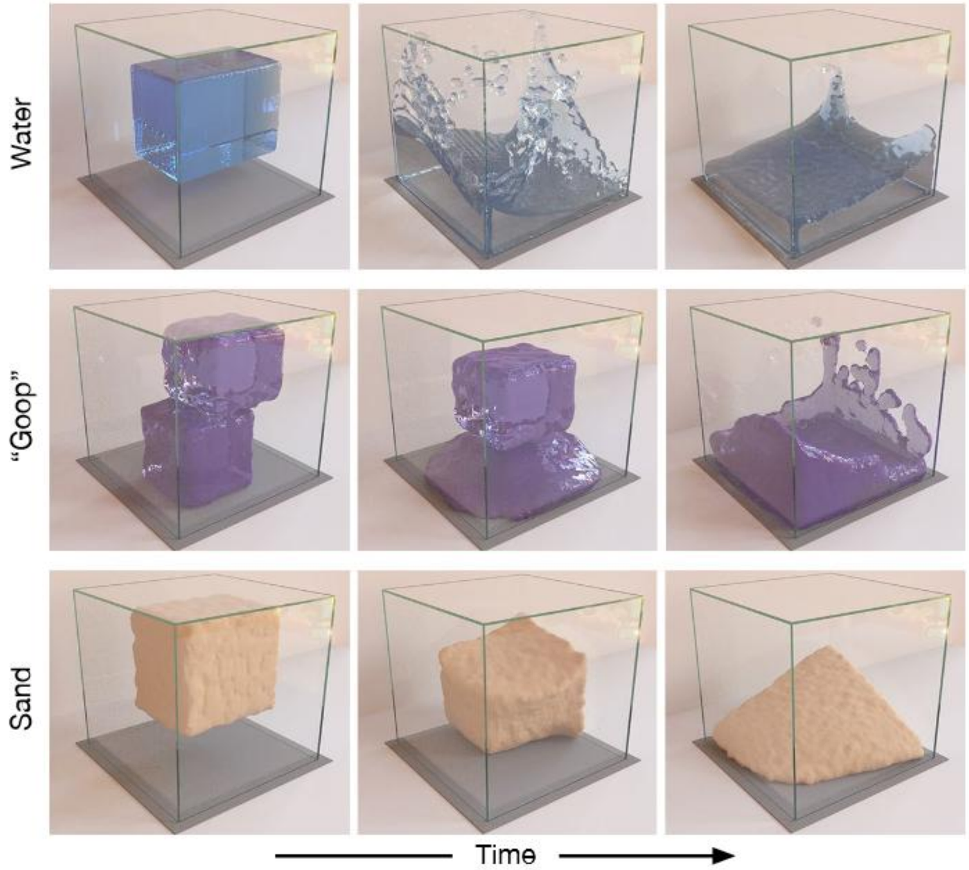
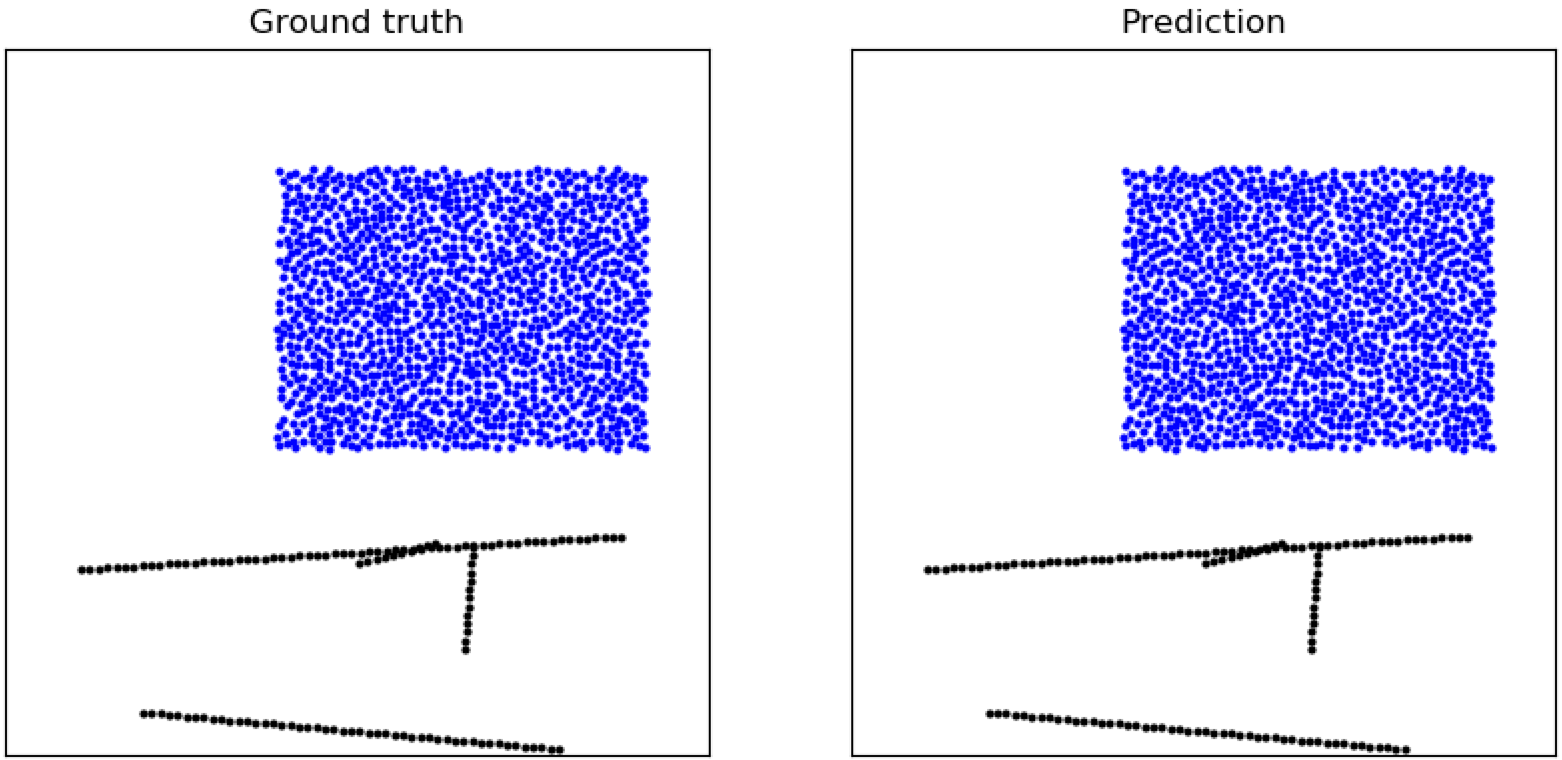


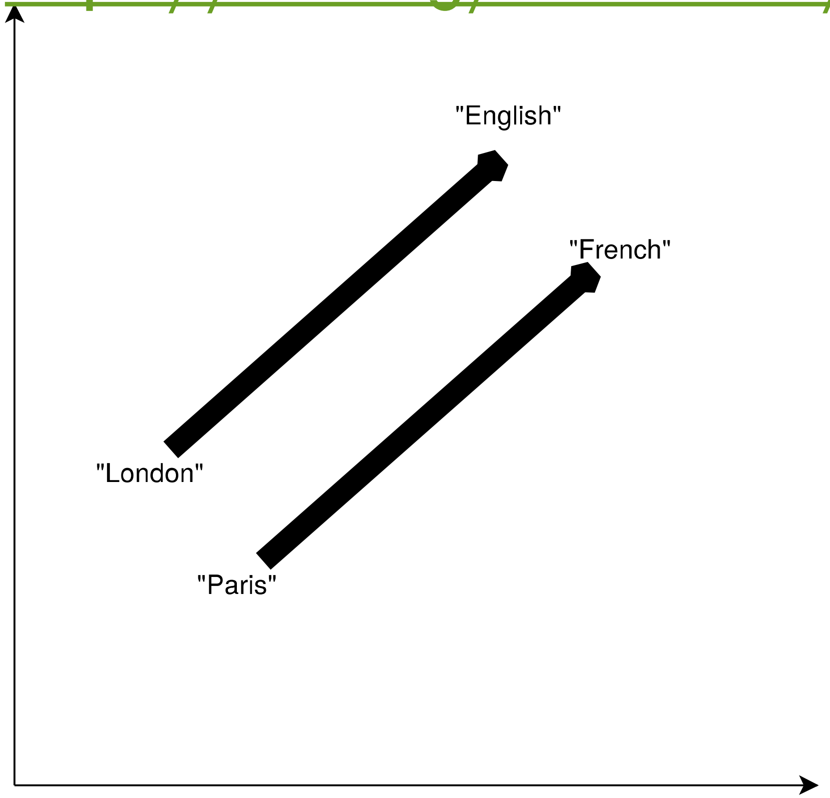
Figure 1. Rollouts of our GNS model for our WATER-3D, GOOP-3D and SAND-3D datasets. It learns to simulate rich materials at resolutions sufficient for high-quality rendering [video].

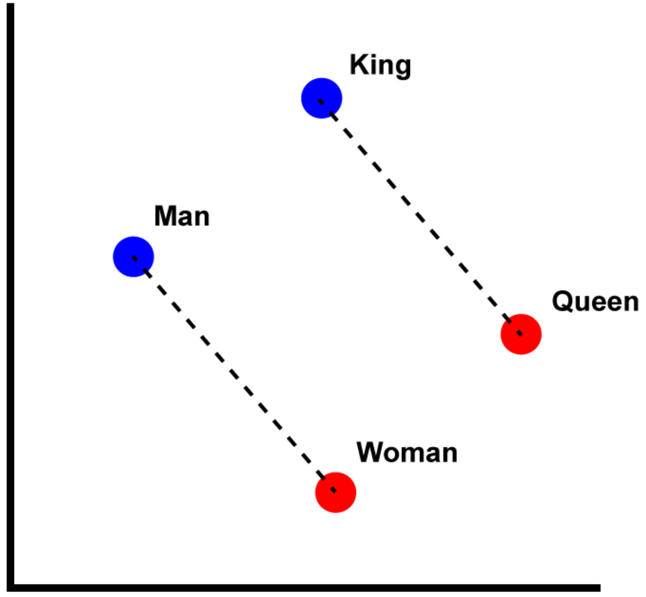
Deep Mind: Learn to Simulate



Word Embedding: Word2Vec

- Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. “Efficient Estimation of Word Representations in Vector Space.” arXiv. https://doi.org/10.48550/arXiv.1301.3781.





Large Language Models: ChatGPT (2022)

KY

describe bias variance trade off

The bias-variance tradeoff is a fundamental concept in machine learning that refers to the tradeoff between how well a model fits the training data and how well it generalizes to new, unseen data.

Bias refers to the difference between the expected prediction of the model and the true values of the target variable. If a model has high bias, it means that it is underfitting the data and is not complex enough to capture the underlying patterns in the data. This can lead to poor performance on both the training and test data.

Variance, on the other hand, refers to the variability of the model's predictions for different training sets. If a model has high variance, it means that it is overfitting the data and is too complex, capturing noise and random fluctuations in the training data. This can lead to

KY

describe bias variance trade off in the context of decision trees

In the context of decision trees, the bias-variance tradeoff refers to the tradeoff between the complexity of the tree and its ability to generalize to new data. Decision trees can be prone to overfitting, which means that they can become too complex and capture noise and random fluctuations in the training data, leading to high variance.

A decision tree with high bias will be too simple and will not capture the underlying patterns in the data. On the other hand, a decision tree with high variance will be too complex and will capture noise and random fluctuations in the training data.

To achieve a balance between bias and variance, we can use techniques such as pruning and setting the minimum number of samples required to split a node. Pruning involves removing branches

Road to Deep Learning

Growth in dataset size, computer memory and compute power

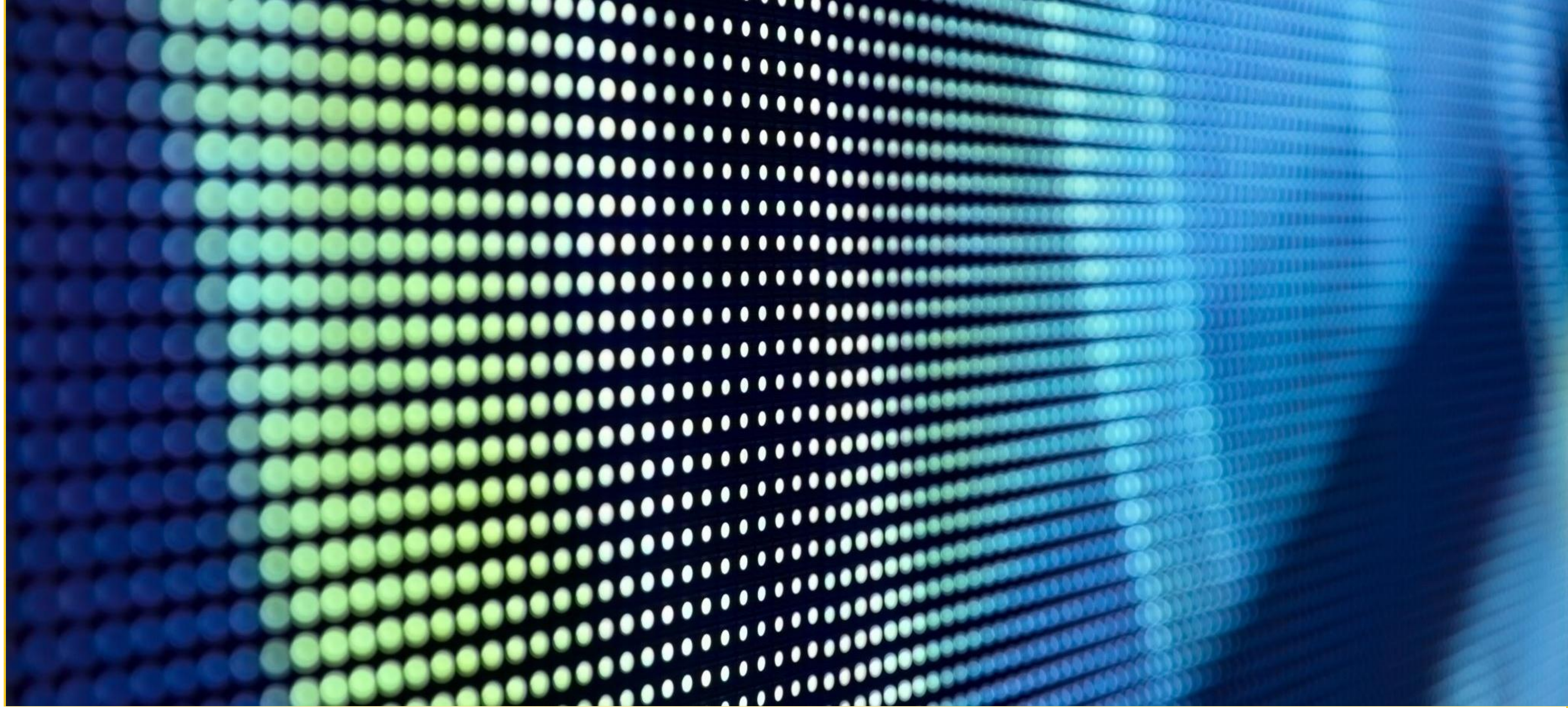
Decade	Dataset	Memory	Floating point calculations per second
1970	100 (Iris)	1 KB	100 KF (Intel 8080)
1980	1 K (house prices in Boston)	100 KB	1 MF (Intel 80186)
1990	10 K (optical character recognition)	10 MB	10 MF (Intel 80486)
2000	10 M (web pages)	100 MB	1 GF (Intel Core)
2010	10 G (advertising)	1 GB	1 TF (NVIDIA C2050)
2020	1 T (social network)	100 GB	1 PF (NVIDIA DGX-2)

Road to Deep Learning

- Innovations in neural network architecture and optimization methods
  - Dropout to avoid overfitting
  - Attention mechanism to focus on relevant input data based on current context/state of the neural network
  - Transformer architecture build on just the attention mechanism
  - Modeling probabilities of text sequences, large language models (LLM)
  - Deep Generative models, diffusion models, reinforcement learning …

Essence of Deep Learning

- Deep networks, i.e., many neural network layers
- End-to-end training, model trained from raw data (image, audio, text) to directly outputting desired results
  - No feature engineering needed
- Others
  - Transition from parametric to nonparametric models
  - Acceptance of suboptimal solutions
  - Reproducibility, requires sharing of code and data



Software

Installation

- Refer to: https://www.d2l.ai/chapter_installation/index.html
- Install Miniconda
  - Download here: https://docs.conda.io/en/latest/miniconda.html
  - Installation Instructions: https://conda.io/projects/conda/en/stable/user-guide/install/index.html
- Examples:
  - Linux or Windows WSL: bash Miniconda3-latest-Linux-x86_64.sh
  - macOS: bash Miniconda3-latest-MacOSX-x86_64.sh
  - Windows: Miniconda3-latest-Windows-x86_64.exe

Installation

- Create Conda environment for DSCI 565
- Linux example:

```
~/miniconda3/bin/conda init
conda create --name d2l python=3.9 -y
conda activate d2l
pip install torch==2.0.0 torchvision==0.15.1 scipy
```

```
pip install d2l==1.0.3
```

Install latest from d2l git repo

```
mkdir ~/git
cd git
git clone https://github.com/d2l-ai/d2l-en.git
cd d2l-en
pip install -e .
```

```
mkdir d2l-en && cd d2l-en
curl https://d2l.ai/d2l-en.zip -o d2l-en.zip
unzip d2l-en.zip && rm d2l-en.zip
cd pytorch
```

```
jupyter lab
```

Installation

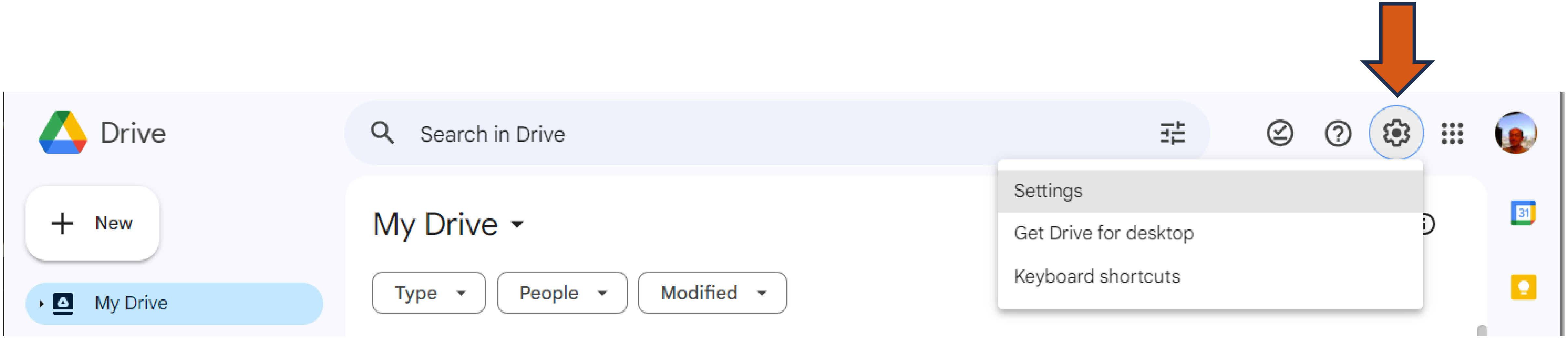
- If you have a CUDA-compatible GPU
- Install CUDA 1.8
  https://developer.nvidia.com/cuda-11-8-0-download-archive
- And install the corresponding pytorch-cuda
  https://pytorch.org/get-started/locally/

Google Colab

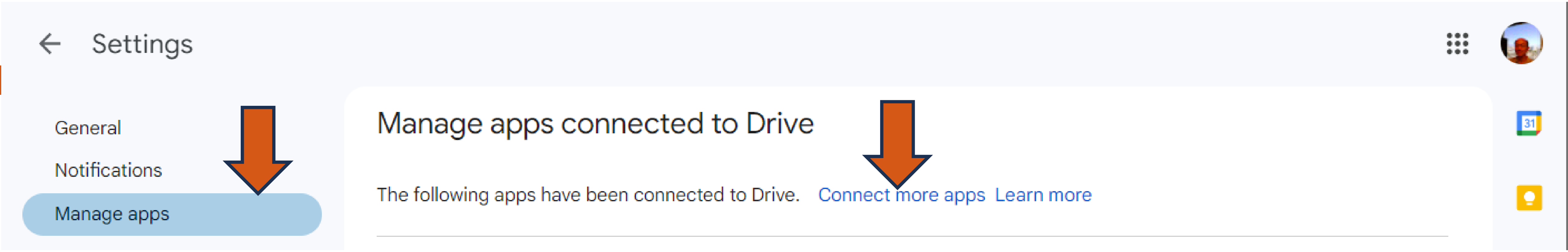
- Go to https://research.google.com/colaboratory/
  - You will need a Google account
- To use GPU go to Edit->Notebook Settings
- Also, from Google Drive you can directly create Colab Jupyter Notebooks by installing the Colab app on Google Drive

Jupyter Notebook on Google Colaboratory

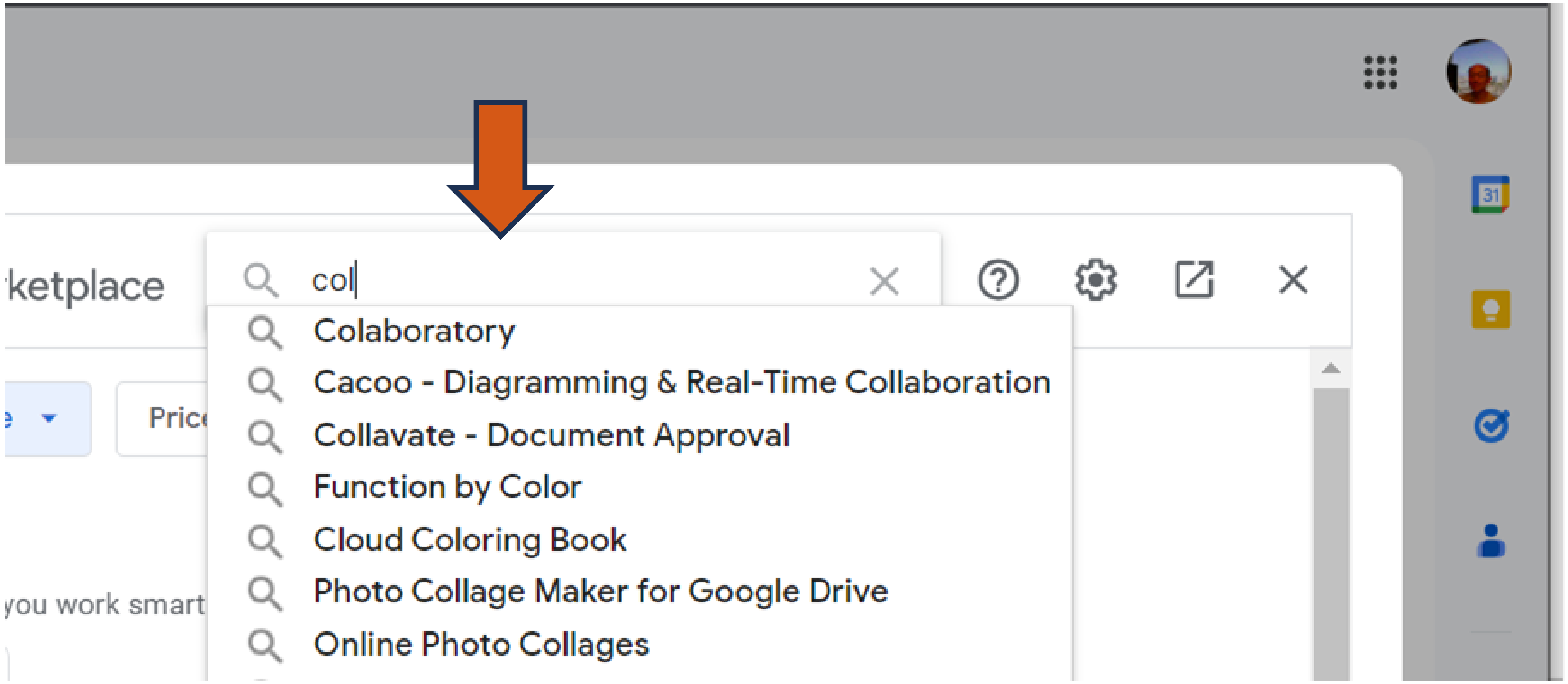
- Install Google Colaboratory app in Google drive
  - Go to the Settings by clicking on Gear icon next to the search bar

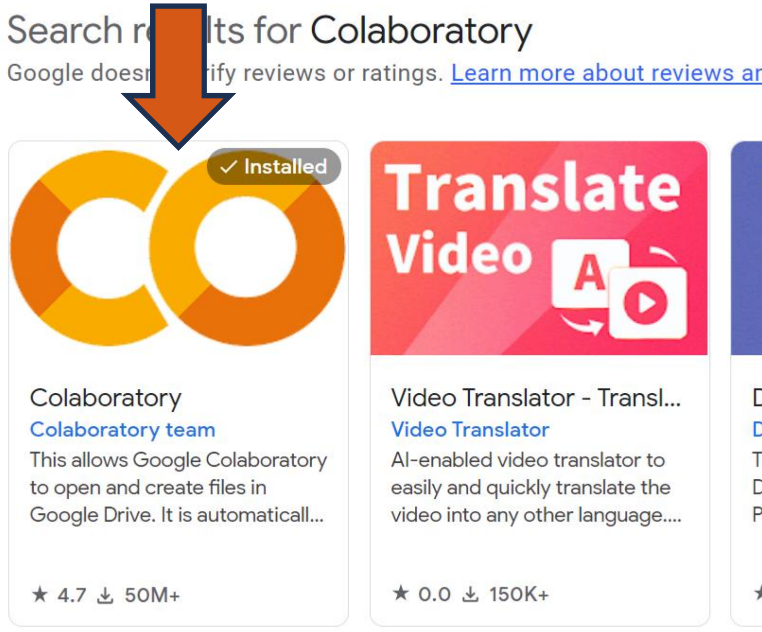


Click “Manage apps” and click “Connect more apps”



Search for “colaboratory” app





Click on “Install” and give permission