COMS 4776 NNDL Lecture 1: Introduction

Richard Zemel

Course information

- Second course in machine learning, with a focus on neural networks
  - This is an advanced machine learning course following Intro to ML with an in-depth focus on cutting-edge topics
  - Assumes knowledge of basic ML algorithms: linear regression, logistic regression, maximum likelihood, PCA, EM, etc.
  - Prerequisites:
    - **Machine Learning:** COMS 4771, or equivalent
    - **Multivariable Calculus**
    - **Linear Algebra**
    - **Probability & Statistics**
  - It is your responsibility to ensure that you have these prerequisites. If you don't you should take this course next year after fulfilling them.

What should I know?

- Probability
  - Starting from the definition of independence, show that the independence of X and Y implies that their covariance is 0.
  - Write the transformation that takes $x \sim \mathcal{N}(0., 1.)$ to $z \sim \mathcal{N}(\mu, \sigma^2)$.
  - Write a code implementation to produce n independent samples from $\mathcal{N}(\mu, \sigma^2)$ by transforming n samples from $\mathcal{N}(0., 1.)$.
- Calculus
  - Let $x, y \in \Re^m$, $A \in \Re^{m \times n}$, and square matrix $B \in \Re^{m \times m}$. And where x' is the transpose of x. Answer the following questions in vector notation.
    - What is the gradient of $x'y$ with respect to x?
    - What is the gradient of $x'x$ with respect to x?
    - What is the Jacobian of A with respect to x?

What else should I know?

- Machine Learning
  - What is a validation set? Describe the trade-offs involved in assigning examples to the validation set versus the training set.
  - Suppose that you are training a decision tree but you would like to try an ensemble method. By random resampling, you create 100 copies of your data and train a separate decision tree based on each one of them, and predict outputs based on the majority vote of the trees. What is the effect of this procedure? How would your test error compare to a single decision tree predictor?
  - What are the advantages and disadvantages of k-nearest neighbors versus logistic regression? How do their decision boundaries compare?

Course information

- Expectations and marking
  - Written homeworks (40% of total mark)
    - Assignments will be a mix of written and programming problems
    - You will have 10 days to do each assignment
    - The written part will consist of 1-3 short conceptual questions
    - They may also involve some mathematical derivations
    - The programming questions must be done in Python, PyTorch
    - They will involve 10-15 lines of code, and give you a chance to experiment with the algorithms
  - Exams
    - Two tests (each worth 20%)
    - No Final
  - Project: 20%

How to get free GPUs

- **Colab (Mandatory)** Programming assignments are to be completed in Google Colab, which is a web-based iPython Notebook service that has access to a free Nvidia K80 GPU per Google account.
- **GCE (Recommended for course projects)** Google Compute Engine delivers virtual machines running in Google's data center. You get $300 free credit when you sign up.

Course information

- Textbooks
  - None, but we link to lots of free online resources (see syllabus).
  - Readings are available for some of the lectures.
- Tutorials
  - Given periodically throughout the course
  - During class
  - Programming/math background; worked-through examples
- Office Hours
  - Professor: Tuesdays 4-5, CEPSR 619
  - TAs: Various times, to help with assignments and quiz preparation

Course information

Course web page:
`http://www.cs.columbia.edu/~zemel/Class/nndl-2025/index.html`

- Username: nndl4776
- Password: nndlClass

EdStem: `https://edstem.org/us/courses/72674/discussion`

What is machine learning?

- For many problems, it's difficult to program the correct behavior by hand
  - recognizing people and objects
  - understanding human speech from audio files

What is machine learning?

- For many problems, it's difficult to program the correct behavior by hand
  - recognizing people and objects
  - understanding human speech from audio files
- Machine learning approach: program an algorithm to automatically learn from data, or from experience

What is machine learning?

- For many problems, it's difficult to program the correct behavior by hand
  - recognizing people and objects
  - understanding human speech from audio files
- Machine learning approach: program an algorithm to automatically learn from data, or from experience
- Some reasons you might want to use a learning algorithm:
  - hard to code up a solution by hand (e.g. vision, natural language processing)
  - system needs to adapt to a changing environment (e.g. spam detection)
  - want the system to perform *better* than the human programmers
  - privacy/fairness (e.g. ranking search results)

Relations to AI

- Nowadays, "machine learning" is often brought up with "artificial intelligence" (AI)

Relations to AI

- Nowadays, "machine learning" is often brought up with "artificial intelligence" (AI)
- AI often does not imply a learning based system
  - Symbolic reasoning
  - Rule based system
  - Tree search
  - etc.

Relations to AI

- Nowadays, "machine learning" is often brought up with "artificial intelligence" (AI)
- AI often does not imply a learning based system
  - Symbolic reasoning
  - Rule based system
  - Tree search
  - etc.
- Learning based system $\rightarrow$ learned based on the data $\rightarrow$ more flexibility, good at solving pattern recognition problems.

Relations to human learning

- It is tempting to imagine machine learning as a component in AI just like human learning in ourselves.

Relations to human learning

- It is tempting to imagine machine learning as a component in AI just like human learning in ourselves.
- Human learning is:
  - Very data efficient
  - An entire multitasking system (vision, language, motor control, etc.)
  - Takes at least a few years :)
- For serving specific purposes, machine learning doesn't have to look like human learning in the end.

Relations to human learning

- It is tempting to imagine machine learning as a component in AI just like human learning in ourselves.
- Human learning is:
  - Very data efficient
  - An entire multitasking system (vision, language, motor control, etc.)
  - Takes at least a few years :)
- For serving specific purposes, machine learning doesn't have to look like human learning in the end.
- It may borrow ideas from biological systems (e.g. neural networks).

Relations to human learning

- It is tempting to imagine machine learning as a component in AI just like human learning in ourselves.
- Human learning is:
  - Very data efficient
  - An entire multitasking system (vision, language, motor control, etc.)
  - Takes at least a few years :)
- For serving specific purposes, machine learning doesn't have to look like human learning in the end.
- It may borrow ideas from biological systems (e.g. neural networks).
- There may also be biological constraints.

History of machine learning

- 1957 — Perceptron algorithm (implemented as a circuit!)
- 1959 — Arthur Samuel wrote a learning-based checkers program that could defeat him
- 1969 — Minsky and Papert's book *Perceptrons* (limitations of linear models)

History of machine learning

- 1957 — Perceptron algorithm (implemented as a circuit!)
- 1959 — Arthur Samuel wrote a learning-based checkers program that could defeat him
- 1969 — Minsky and Papert's book *Perceptrons* (limitations of linear models)
- 1980s — Some foundational ideas
  - Connectionist psychologists explored neural models of cognition
  - 1984 — Leslie Valiant formalized the problem of learning as PAC learning
  - 1988 — Backpropagation (re-)discovered by Geoffrey Hinton and colleagues
  - 1988 — Judea Pearl's book *Probabilistic Reasoning in Intelligent Systems* introduced Bayesian networks

History of machine learning

- 1990s — the "AI Winter", a time of pessimism and low funding

History of machine learning

- 1990s — the "AI Winter", a time of pessimism and low funding
- But looking back, the '90s were also sort of a golden age for ML research
  - Markov chain Monte Carlo
  - Variational inference
  - Kernels and support vector machines
  - Boosting
  - Convolutional networks

History of machine learning

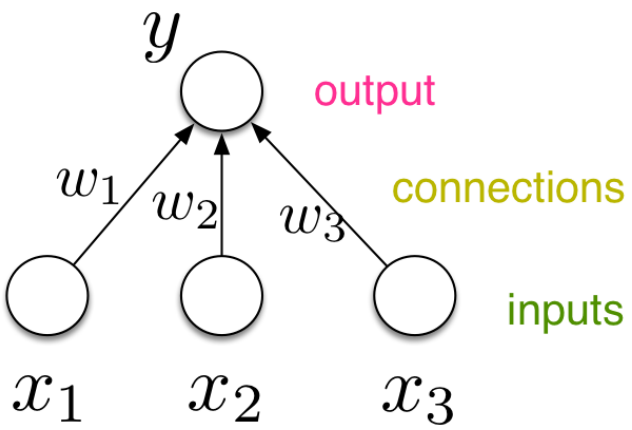
- 1990s — the "AI Winter", a time of pessimism and low funding
- But looking back, the '90s were also sort of a golden age for ML research
  - Markov chain Monte Carlo
  - Variational inference
  - Kernels and support vector machines
  - Boosting
  - Convolutional networks
- 2000s — applied AI fields (vision, NLP, etc.) adopted ML

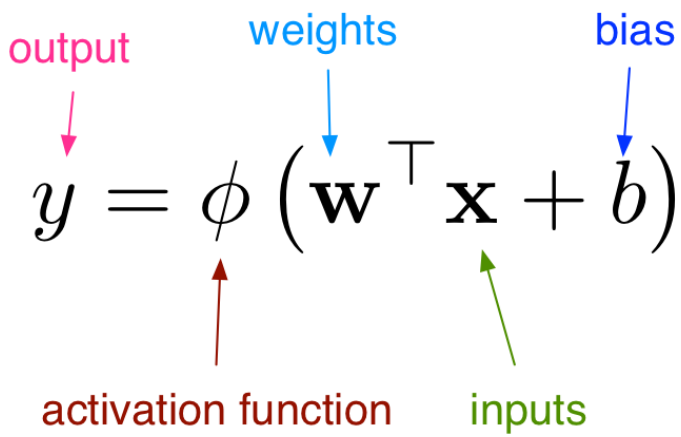
History of machine learning

- 1990s — the "AI Winter", a time of pessimism and low funding
- But looking back, the '90s were also sort of a golden age for ML research
  - Markov chain Monte Carlo
  - Variational inference
  - Kernels and support vector machines
  - Boosting
  - Convolutional networks
- 2000s — applied AI fields (vision, NLP, etc.) adopted ML
- 2010s — deep learning
  - 2010–2012 — neural nets smashed previous records in speech-to-text and object recognition
  - increasing adoption by the tech industry
  - 2016 — AlphaGo defeated the human Go champion
  - 2021–2023 — ChatGPT, AlphaFold

What are neural networks?

- Most of the biological details aren't essential, so we use vastly simplified models of neurons.
- While neural nets originally drew inspiration from the brain, nowadays we mostly think about math, statistics, etc.





- Neural networks are collections of thousands (or millions) of these simple processing units that together perform useful computations.

What are neural networks?

Why neural nets?

- inspiration from the brain
  - proof of concept that a neural architecture can see and hear!
- very effective across a range of applications (vision, text, speech, medicine, robotics, etc.)
- widely used in both academia and the tech industry
- powerful software frameworks (PyTorch, TensorFlow, etc.) let us quickly implement sophisticated algorithms

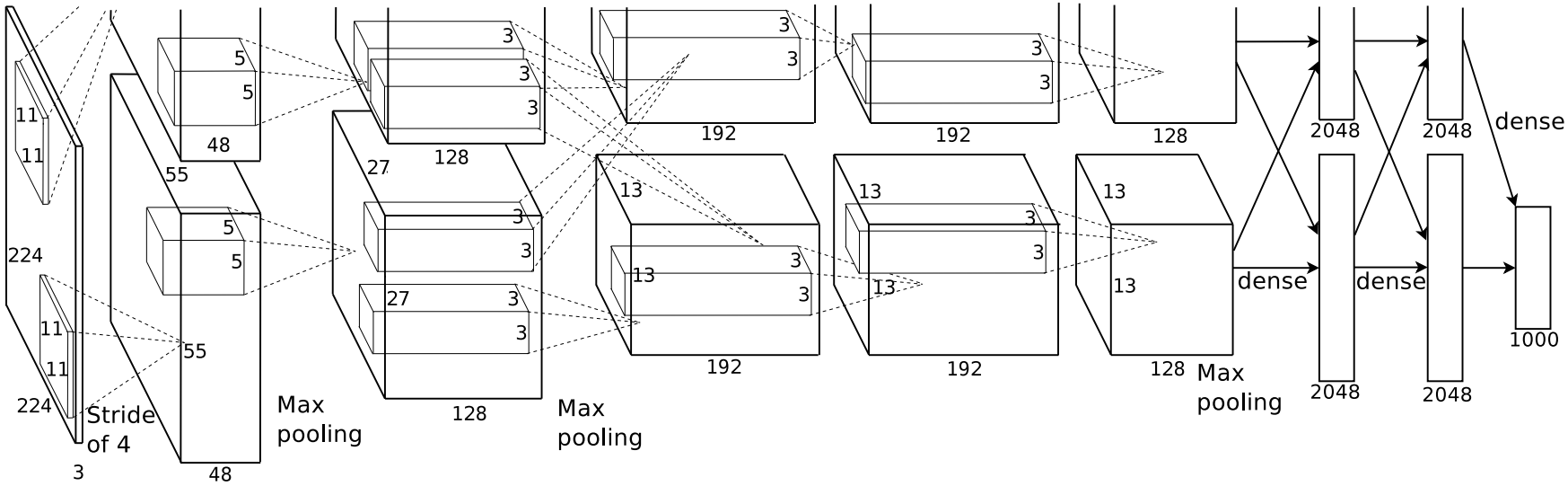
What are neural networks?

- Some near-synonyms for neural networks
  - "Deep learning"
    - Emphasizes that the algorithms often involve hierarchies with many stages of processing

"Deep learning"

Deep learning: many layers (stages) of processing

E.g. this network which recognizes objects in images:



(Krizhevsky et al., 2012)

Each of the boxes consists of many neuron-like units similar to the one on the previous slide!

"Deep learning"

- You can visualize what a learned feature is responding to by finding an image that excites it. (We'll see how to do this.)
- Higher layers in the network often learn higher-level, more interpretable representations

Edges (layer conv2d0) **Textures** (layer mixed3a) **Patterns** (layer mixed4a)

https://distill.pub/2017/feature-visualization/

"Deep learning"

- You can visualize what a learned feature is responding to by finding an image that excites it.
- Higher layers in the network often learn higher-level, more interpretable representations

Parts (layers mixed4b & mixed4c) **Objects** (layers mixed4d & mixed4e)

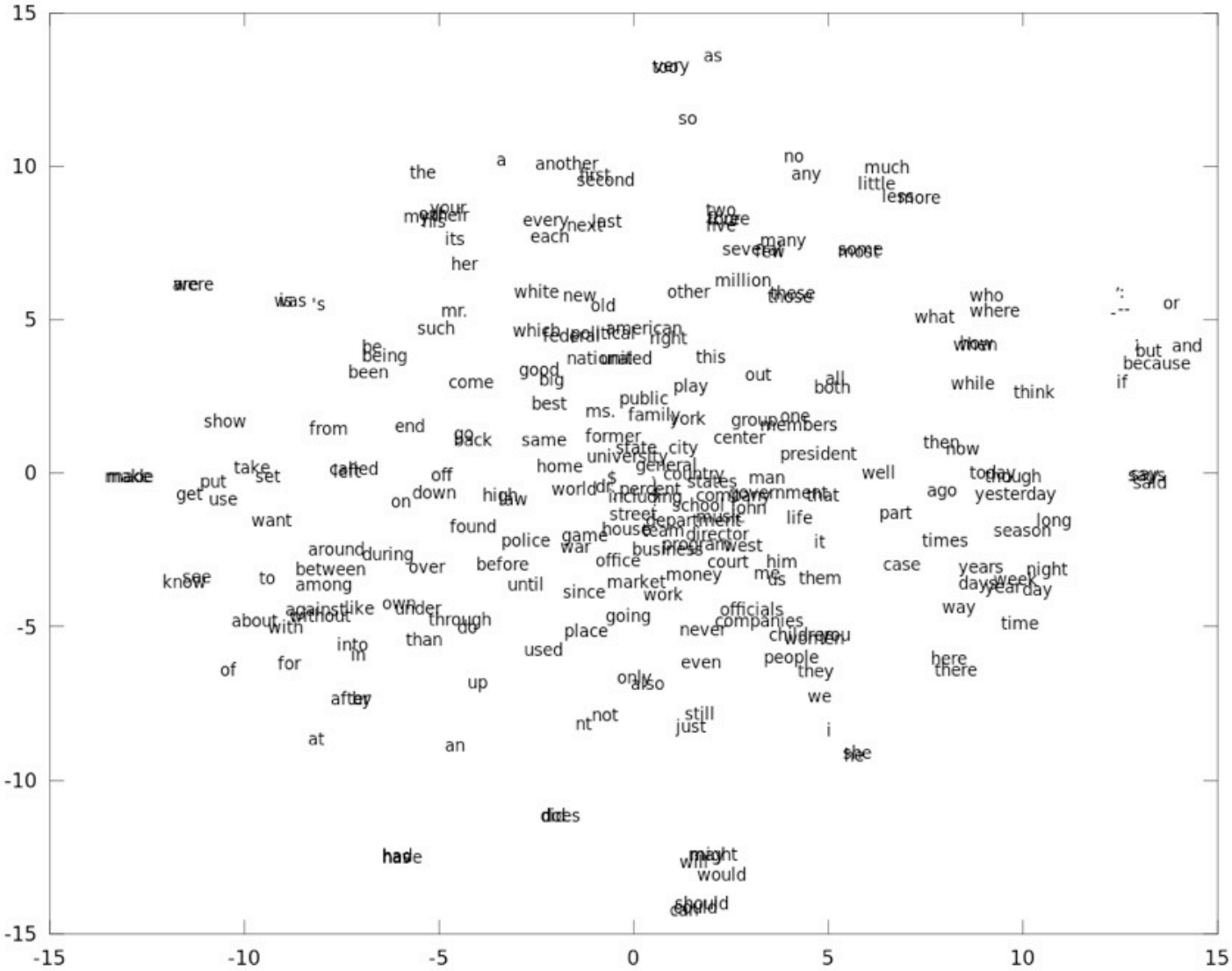
https://distill.pub/2017/feature-visualization/

What is a representation?

- How you represent your data determines what questions are easy to answer.
  - E.g. a dict of word counts is good for questions like "What is the most common word in *Hamlet*?"
  - It's not so good for semantic questions like "if Alice liked *Harry Potter*, will she like *The Hunger Games*?"

What is a representation?

Idea: represent words as vectors



What is a representation?

- Mathematical relationships between vectors encode semantic relationships between words
  - Measure semantic similarity using the dot product (or dissimilarity using Euclidean distance)
  - Represent a web page with the average of its word vectors
  - Complete analogies by doing arithmetic on word vectors
    - e.g. "Paris is to France as London is to _____"
    - France – Paris + London = _____

What is a representation?

- Mathematical relationships between vectors encode semantic relationships between words
  - Measure semantic similarity using the dot product (or dissimilarity using Euclidean distance)
  - Represent a web page with the average of its word vectors
  - Complete analogies by doing arithmetic on word vectors
    - e.g. "Paris is to France as London is to _____"
    - France – Paris + London = _____
- It's very hard to construct representations like these by hand, so we need to learn them from data
  - This is a big part of what neural nets do, whatever type of learning they are doing!

Types of machine learning

- **Supervised learning:** have labeled examples of the correct behavior, i.e. ground truth input/output response
- **Reinforcement learning:** learning system receives a reward signal, tries to learn to maximize the reward signal
- **Unsupervised learning:** no labeled examples – instead, looking for interesting patterns in the data

Supervised learning examples

Supervised learning: have labeled examples of the correct behavior

e.g. Handwritten digit classification with the MNIST dataset

- **Task:** given an image of a handwritten digit, predict the digit class
  - **Input:** the image
  - **Target:** the digit class

Supervised learning examples

Supervised learning: have labeled examples of the correct behavior

e.g. Handwritten digit classification with the MNIST dataset

- **Task:** given an image of a handwritten digit, predict the digit class
  - **Input:** the image
  - **Target:** the digit class
- **Data:** 70,000 images of handwritten digits labeled by humans
  - **Training set:** first 60,000 images, used to train the network
  - **Test set:** last 10,000 images, not available during training, used to evaluate performance

Supervised learning examples

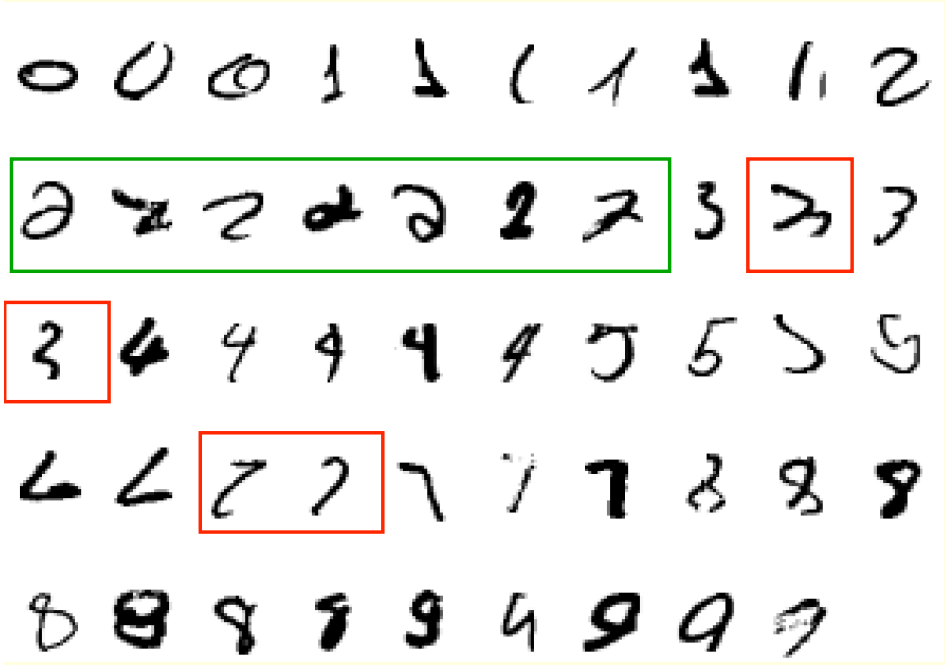
Supervised learning: have labeled examples of the correct behavior

e.g. Handwritten digit classification with the MNIST dataset

- **Task:** given an image of a handwritten digit, predict the digit class
  - **Input:** the image
  - **Target:** the digit class
- **Data:** 70,000 images of handwritten digits labeled by humans
  - **Training set:** first 60,000 images, used to train the network
  - **Test set:** last 10,000 images, not available during training, used to evaluate performance
- This dataset is the "fruit fly" of neural net research
- Neural nets already achieved $> 99\%$ accuracy in the 1990s, but we still continue to learn a lot from it

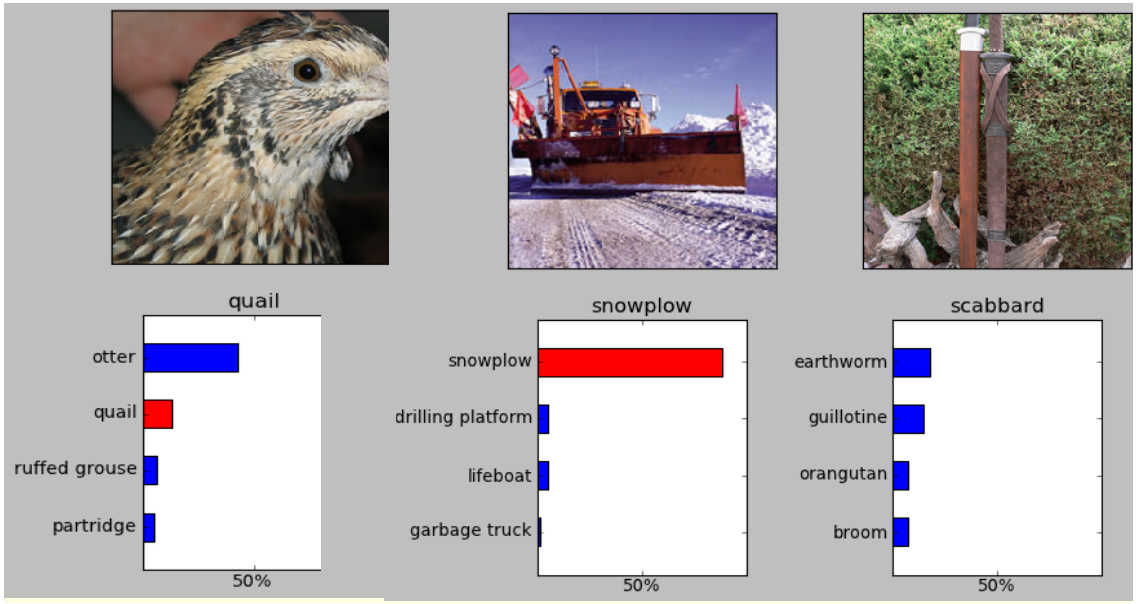
Supervised learning examples

What makes a "2"?



Supervised learning examples

Object recognition



(Krizhevsky and Hinton, 2012)

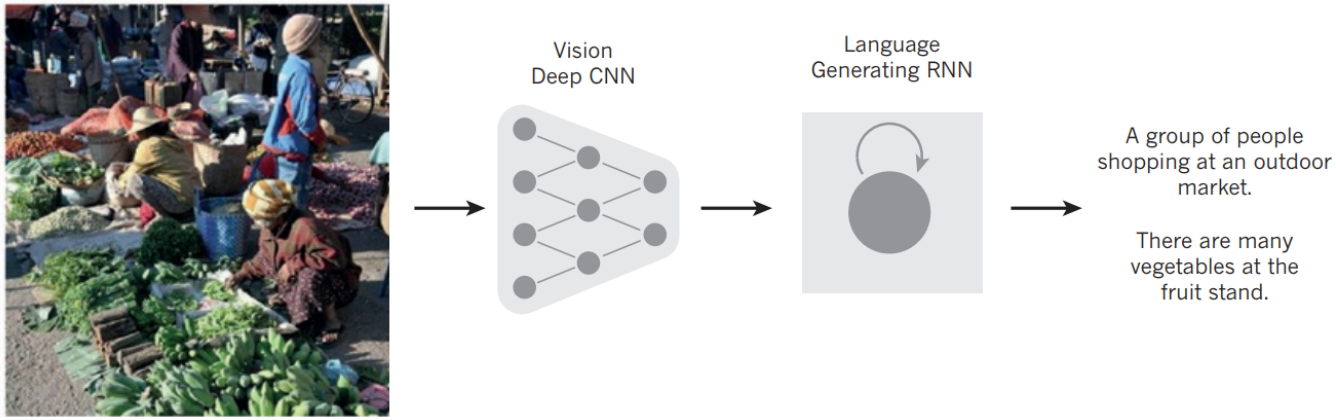
ImageNet dataset: one thousand categories, millions of labeled images

Lots of variability in viewpoint, lighting, etc.

Error rate dropped from 26% to under 4% over the course of a few years!

Supervised learning examples

Caption generation



A woman is throwing a **frisbee** in a park.

A **dog** is standing on a hardwood floor.

A **stop** sign is on a road with a mountain in the background

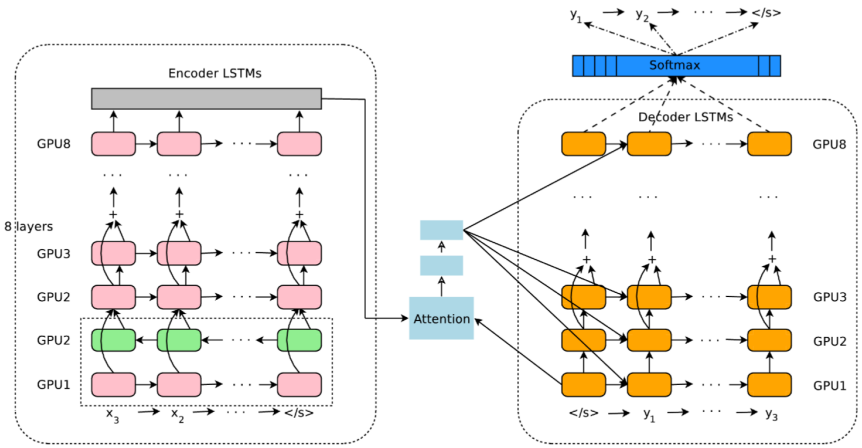
(Xu et al., 2015)

Given: dataset of Flickr images with captions

More examples at `http://deeplearning.cs.toronto.edu/i2t`

Supervised learning examples

Neural Machine Translation



(Wu et al., 2016)

Input sentence:	*Translation (PBMT):*	*Translation (GNMT):*	*Translation (human):*
李克強此行將啟動中加總理年度對話機制，與加拿大總理杜魯多舉行兩國總理首次年度對話。	Li Keqiang premier added this line to start the annual dialogue mechanism with the Canadian Prime Minister Trudeau two prime ministers held its first annual session.	Li Keqiang will start the annual dialogue mechanism with Prime Minister Trudeau of Canada and hold the first annual dialogue between the two premiers.	Li Keqiang will initiate the annual dialogue mechanism between premiers of China and Canada during this visit, and hold the first annual dialogue with Premier Trudeau of Canada.

Unsupervised learning examples

- In **generative modeling**, we want to learn a distribution over some dataset, such as natural images.
- We can evaluate a generative model by sampling from the model and seeing if it looks like the data.
- These results were considered impressive in 2014:

Denton et al., 2014, Deep generative image models using a Laplacian pyramid of adversarial networks

Unsupervised learning examples

- The progress of generative models:

Odena et al 2016

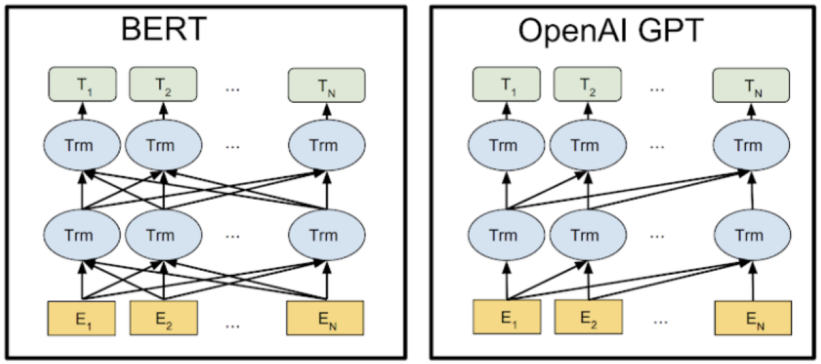
Miyato et al 2017

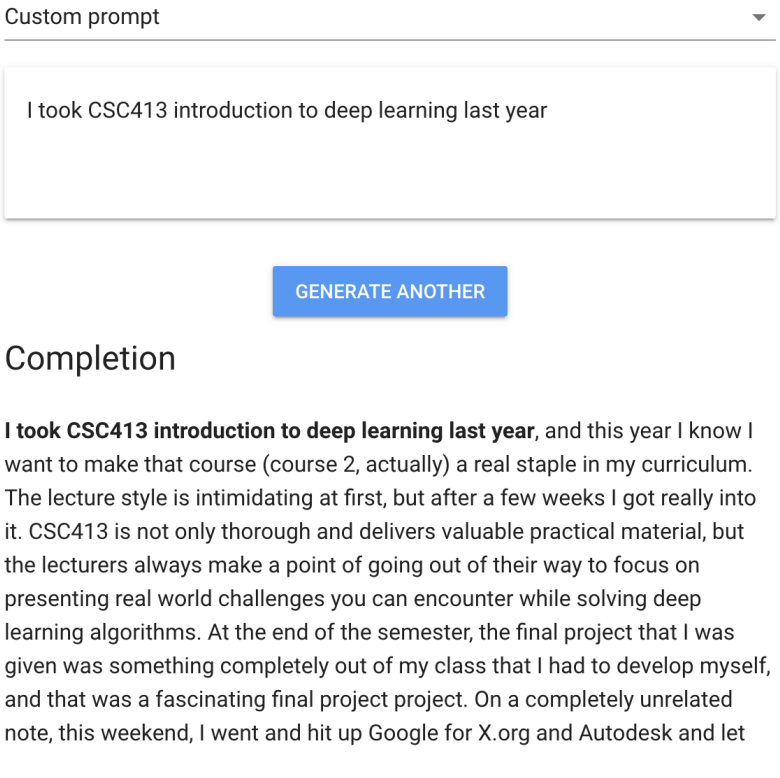
Zhang et al 2018

- Big GAN, Brock et al, 2019:

Unsupervised learning examples

- Generative models of text. Original next-gen chatbots like BERT and GPT-2 perform unsupervised learning by reconstructing words in a sentence. GPT-2 was trained from 40GB of Internet text.





https://talktotransformer.com/

Unsupervised learning examples

- Interesting result: the CycleGAN model takes lots of images of one category (e.g. horses) and lots of images of another category (e.g. zebras) and learns to translate between them.

https://github.com/junyanz/CycleGAN

Unsupervised learning examples

Automatic mouse tracking

- When biologists do behavioral genetics researchers on mice, it's very time consuming for a person to sit and label everything a mouse does
- Various groups aim to build a system for automatically tracking mouse behaviors
- Goal: show the researchers a summary of how much time different mice spend on various behaviors, so they can determine the effects of the genetic manipulations
- One of the major challenges is that we don't know the right "vocabulary" for describing the behaviors — clustering the observations into meaningful groups is an unsupervised learning task
- **video:** `http://www.sciencedirect.com/science/article/pii/S0896627315010375`

Reinforcement learning

- An **agent** interacts with an **environment** (e.g. game of Breakout)
- In each time step,
  - the agent receives **observations** (e.g. pixels) which give it information about the **state** (e.g. positions of the ball and paddle)
  - the agent picks an **action** (e.g. keystrokes) which affects the state
- The agent periodically receives a **reward** (e.g. points)
- The agent wants to learn a **policy**, or mapping from observations to actions, which maximizes its average reward over time

Reinforcement learning

DeepMind trained neural networks to play many different Atari games

- given the raw screen as input, plus the score as a reward
- single network architecture shared between all the games
- in many cases, the networks learned to play better than humans (in terms of points in the first minute)

`https://www.youtube.com/watch?v=V1eYniJ0Rnk`

Reinforcement learning for control

Learning locomotion control from scratch

- The reward is to run as far as possible over all the obstacles
- single control policy that learns to adapt to different terrains

`https://www.youtube.com/watch?v=hx_bgoTF7bs`

Software frameworks

- Scientific computing (NumPy)
  - **vectorize** computations (express them in terms of matrix/vector operations) to exploit hardware efficiency
- Neural net frameworks: PyTorch, TensorFlow, etc.
  - automatic differentiation
  - compiling computation graphs
  - libraries of algorithms and network primitives
  - support for graphics processing units (GPUs)
- For this course:
  - Python, NumPy
  - **PyTorch**, a widely used neural net framework with a built-in automatic differentiation feature

Software frameworks

Why take this class, if PyTorch does so much for you?

So you know what do to if something goes wrong!

- Debugging learning algorithms requires sophisticated detective work, which requires understanding what goes on beneath the hood.
- That's why we derive things by hand in this class!