

# Making Your Apps Smarter: Machine Learning & AI

Shan-Hung Wu  
*shwu@cs.nthu.edu.tw*

Department of Computer Science,  
National Tsing Hua University, Taiwan

Software Design & Studio

# Outline

1 Web, App, and Business Intelligence

2 What's Machine Learning & AI?

3 What's Deep Learning?

4 What's Generative AI?

5 Making Smart Apps

# Outline

① Web, App, and Business Intelligence

② What's Machine Learning & AI?

③ What's Deep Learning?

④ What's Generative AI?

⑤ Making Smart Apps

# Your Term Project

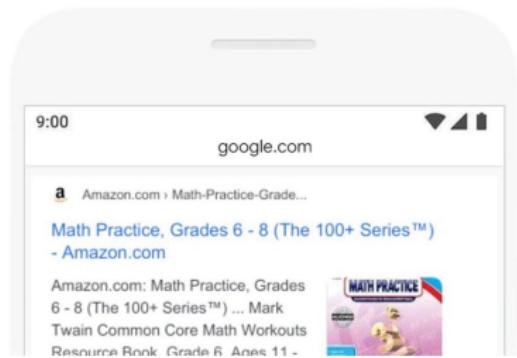
To *design & implement* an  
*intellectual app* that solves *real  
problems*.

# Example: Intention Identification

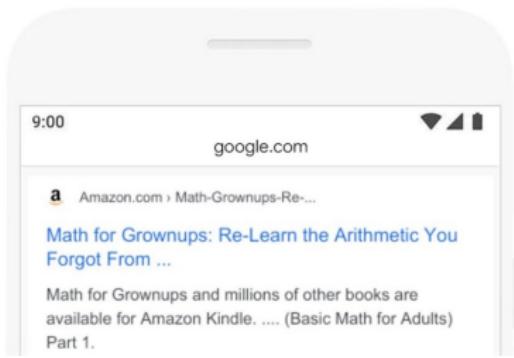


math practice books for adults

BEFORE



AFTER



# Example: Spam Detection



Donald J. Trump  @realDonaldTrump · 12h  
There is NO WAY (ZERO!) that Mail-In Ballots will be anything less than substantially fraudulent. Mail boxes will be robbed, ballots will be forged & even illegally printed out & fraudulently signed. The Governor of California is sending Ballots to millions of people, anyone.....

 Get the facts about mail-in ballots

31.2K 29.2K 100.8K 

Donald J. Trump  @realDonaldTrump

....living in the state, no matter who they are or how they got there, will get one. That will be followed up with professionals telling all of these people, many of whom have never even thought of voting before, how, and for whom, to vote. This will be a Rigged Election. No way!

 Get the facts about mail-in ballots

# Example: Product Recommendation

## Frequently Bought Together



Price For All Three: \$258.02

Add all three to Cart

- This item: [The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition \(Springer Series in Statistics\)](#) by Trevor Hastie
- [Pattern Recognition and Machine Learning \(Information Science and Statistics\)](#) by Christopher M. Bishop
- [Pattern Classification \(2nd Edition\)](#) by Richard O. Duda

## Customers Who Bought This Item Also Bought



[All of Statistics: A Concise Course in Statist... by Larry Wasserman](#)  
★★★★★ (8) \$60.00



[Pattern Classification \(2nd Edition\)](#) by Richard O. Duda  
★★★★☆ (27) \$117.25



[Data Mining: Practical Machine Learning Tools an... by Ian H. Witten](#)  
★★★★☆ (29) \$41.55



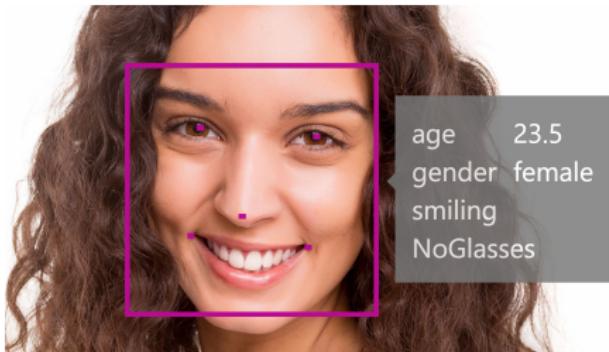
[Bayesian Data Analysis, Second Edition \(Texts in... by Andrew Gelman](#)  
★★★★☆ (10) \$56.20



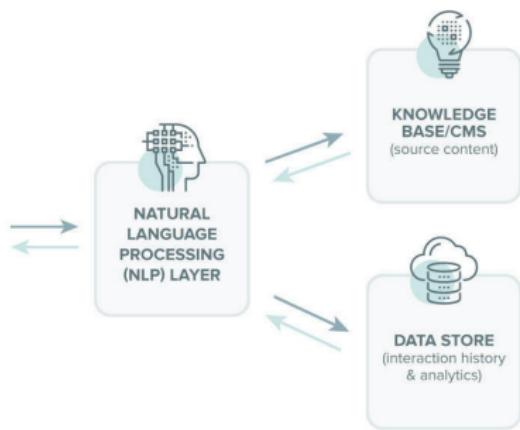
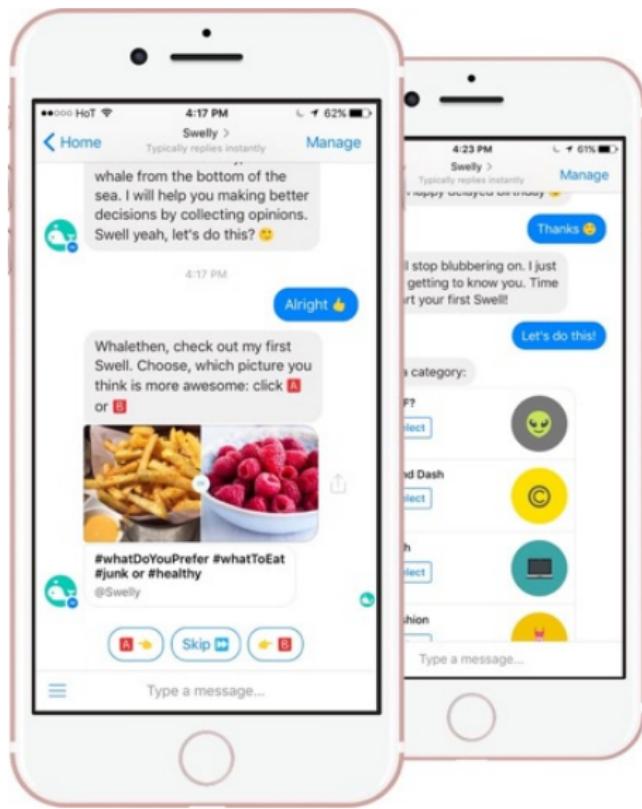
[Data Analysis Using Regression and Multilevel /... by Andrew Gelman](#)  
★★★★☆ (13) \$39.59

The screenshot shows a Spotify interface for a 'Discover Weekly' playlist. On the left, a sidebar lists various sections like 'Browse', 'Radio', 'YOUR MUSIC', and 'PLAYLISTS'. The main area features a large image of a person's face with the text 'MADE FOR MOORISSA' and 'Discover Weekly'. Below it, the text reads: 'Your weekly mixtape of fresh music. Enjoy new discoveries and deep cuts chosen just for you. Updated every Monday, so save your favourites!' It also says 'Made for Moorissa Tjokro by Spotify • 30 songs, 2 hr 6 min'. At the bottom, there's a track list with titles like 'Home (feat. Jeremy Camp)', 'Adam Cappa, Jeremy C...', and '2 days ago'. To the right, there's a 'FIND FRIENDS' button and a section titled 'See what your friends are playing' showing friends like Julia Eger and Ben Khan.

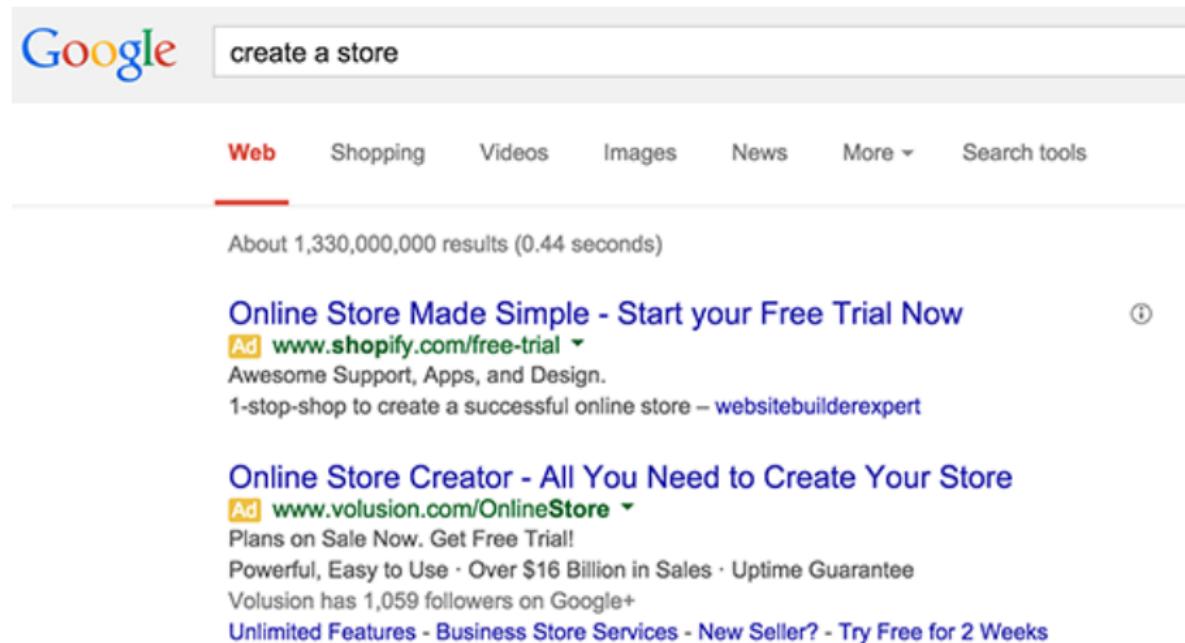
# Example: Image Understanding



# Example: Chat Bot & Service Automation



# Example: Marketing & Advertisement



A screenshot of a Google search results page. The search query "create a store" is entered in the search bar. The results are filtered by "Web". There are approximately 1,330,000,000 results found in 0.44 seconds.

The first result is an advertisement for Shopify:

**Online Store Made Simple - Start your Free Trial Now**  
**Ad** [www.shopify.com/free-trial](http://www.shopify.com/free-trial) ▾  
Awesome Support, Apps, and Design.  
1-stop-shop to create a successful online store – [websitebuilderexpert](#)

The second result is another advertisement for Volusion:

**Online Store Creator - All You Need to Create Your Store**  
**Ad** [www.volusion.com/OnlineStore](http://www.volusion.com/OnlineStore) ▾  
Plans on Sale Now. Get Free Trial!  
Powerful, Easy to Use · Over \$16 Billion in Sales · Uptime Guarantee  
Volusion has 1,059 followers on Google+  
Unlimited Features - Business Store Services - New Seller? - Try Free for 2 Weeks

- All based on **Machine Learning** & **AI** technologies

# Outline

① Web, App, and Business Intelligence

② What's Machine Learning & AI?

③ What's Deep Learning?

④ What's Generative AI?

⑤ Making Smart Apps

# AI vs. Machine Learning

- Artificial Intelligence (AI): the goal
  - Creating systems that can function intelligently and independently
  - Mirroring or surpassing human capabilities
- Machine Learning (ML): a means of achieving AI
  - Enabling machines to *learn from data*

# Prior vs. Posteriori Knowledge

- To solve a problem, we need an algorithm
  - E.g., sorting

# Prior vs. Posteriori Knowledge

- To solve a problem, we need an algorithm
  - E.g., sorting
  - ***A priori knowledge*** is enough

# Prior vs. Posteriori Knowledge

- To solve a problem, we need an algorithm
  - E.g., sorting
  - ***A priori knowledge*** is enough
- For some problem, however, we do not have the a priori knowledge
  - E.g., to tell if a post is toxic or not
  - The correct answer varies in time and from site to site

# Prior vs. Posteriori Knowledge

- To solve a problem, we need an algorithm
  - E.g., sorting
  - *A priori knowledge* is enough
- For some problem, however, we do not have the a priori knowledge
  - E.g., to tell if a post is toxic or not
  - The correct answer varies in time and from site to site
- Machine learning algorithms use the *a posteriori knowledge* to solve problems

# Prior vs. Posteriori Knowledge

- To solve a problem, we need an algorithm
  - E.g., sorting
  - ***A priori knowledge*** is enough
- For some problem, however, we do not have the ***a priori knowledge***
  - E.g., to tell if a post is toxic or not
  - The correct answer varies in time and from site to site
- Machine learning algorithms use the ***a posteriori knowledge*** to solve problems
  - Takes ***examples*** as additional input to algorithm

# General ML Step 1: Data Preparation

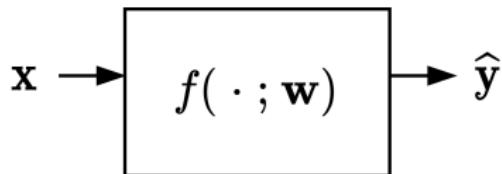
- Pre-process data (e.g., integration, cleaning, etc.)
- Define vector **features** to have a dataset:

$$\mathbb{X} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^N, \text{ where } \mathbf{x}^{(i)} \in \mathbb{R}^D \text{ and } \mathbf{y}^{(i)} \in \mathbb{R}^K,$$

- E.g., in toxic post detection:
  - $\mathbf{x}^{(i)}$  represents counts of different tokens
  - $\mathbf{y}^{(i)} \in \{0, 1\}$  indicates if the post  $\mathbf{x}^{(i)}$  is toxic or not

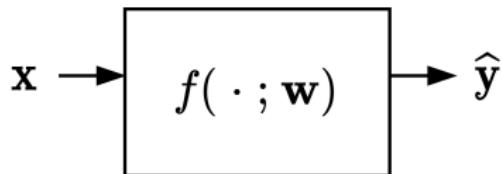


## General ML Step 2: Model Development



- ① Assume a **model**  $\{f(\cdot; \mathbf{w})\}_{\mathbf{w}}$  that is a collection of candidate functions  $f$ 's
  - Each  $f$  predicts label  $\hat{y}$  given an input  $x$
  - $f$  is assumed to be parametrized by  $\mathbf{w}$

## General ML Step 2: Model Development

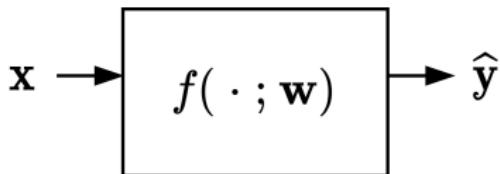


- ① Assume a **model**  $\{f(\cdot; \mathbf{w})\}_{\mathbf{w}}$  that is a collection of candidate functions  $f$ 's
  - Each  $f$  predicts label  $\hat{y}$  given an input  $x$
  - $f$  is assumed to be parametrized by  $\mathbf{w}$
- ② Define a **cost function**

$$C(\mathbf{w}; \mathbb{X})$$

that measures “how good a particular  $f(\cdot; \mathbf{w})$  can explain the training data  $\mathbb{X}$ ” (posteriori knowledge)

## General ML Step 2: Model Development



- ① Assume a **model**  $\{f(\cdot; \mathbf{w})\}_{\mathbf{w}}$  that is a collection of candidate functions  $f$ 's
  - Each  $f$  predicts label  $\hat{\mathbf{y}}$  given an input  $\mathbf{x}$
  - $f$  is assumed to be parametrized by  $\mathbf{w}$
- ② Define a **cost function**

$$C(\mathbf{w}; \mathbb{X})$$

that measures “how good a particular  $f(\cdot; \mathbf{w})$  can explain the training data  $\mathbb{X}$ ” (posteriori knowledge)

- ③ **Training**: employ an algorithm that solves

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} C(\mathbf{w}; \mathbb{X})$$

- Where “learning” happens

# General ML Step 3: Testing & Deployment

- ① **Testing**: evaluate the performance of the learned  $f(\cdot; \mathbf{w}^*)$  using another, *unseen* test dataset  $\mathbb{X}'$ 
  - Examples in  $\mathbb{X}'$  should have the same distribution with those in  $\mathbb{X}$
  - A model minimizing  $C(\mathbf{w}; \mathbb{X})$  does *not* necessarily give high test performance
- ② If  $f(\cdot; \mathbf{w}^*)$  has satisfactory test performance, deploy it to solve real-world problem

# Outline

① Web, App, and Business Intelligence

② What's Machine Learning & AI?

③ What's Deep Learning?

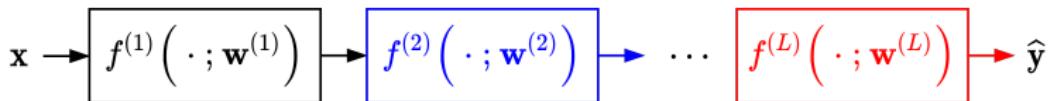
④ What's Generative AI?

⑤ Making Smart Apps

# What is Deep Learning?

- ML where an  $f(\cdot; \mathbf{w})$  has many (deep) layers

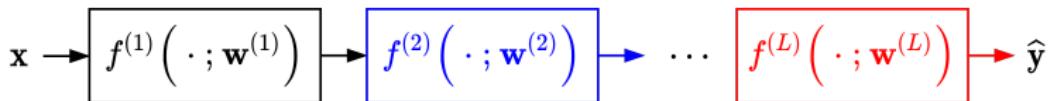
$$\hat{\mathbf{y}} = f^{(L)}(\dots f^{(2)}(f^{(1)}(\mathbf{x}; \mathbf{w}^{(1)}); \mathbf{w}^{(2)}) \dots; \mathbf{w}^{(L)})$$



# What is Deep Learning?

- ML where an  $f(\cdot; \mathbf{w})$  has many (deep) layers

$$\hat{\mathbf{y}} = f^{(L)}(\dots f^{(2)}(f^{(1)}(\mathbf{x}; \mathbf{w}^{(1)}); \mathbf{w}^{(2)}) \dots; \mathbf{w}^{(L)})$$



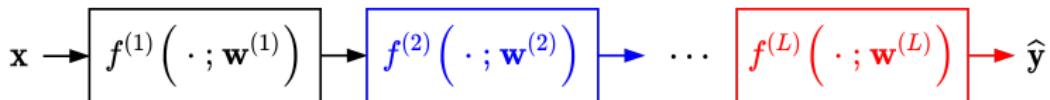
- Pros:

- Learns features from raw data automatically, called ***representation learning***
- Learns a complex function (e.g., visual objects to labels)

# What is Deep Learning?

- ML where an  $f(\cdot; \mathbf{w})$  has many (deep) layers

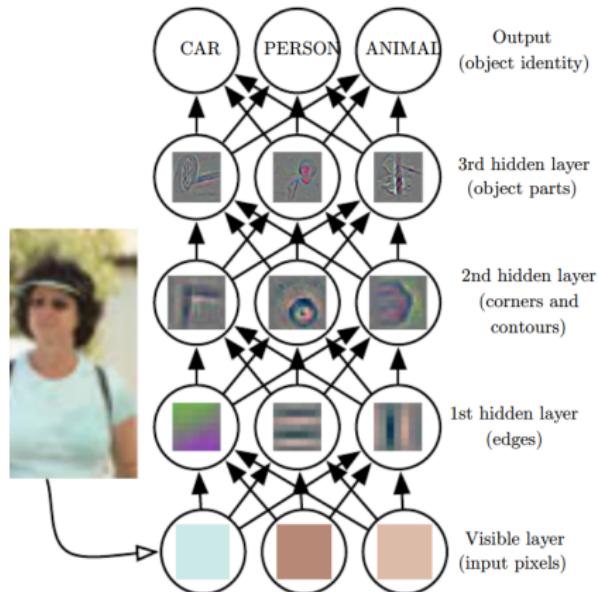
$$\hat{\mathbf{y}} = f^{(L)}(\dots f^{(2)}(f^{(1)}(\mathbf{x}; \mathbf{w}^{(1)}); \mathbf{w}^{(2)}) \dots; \mathbf{w}^{(L)})$$



- Pros:
  - Learns features from raw data automatically, called **representation learning**
  - Learns a complex function (e.g., visual objects to labels)
- Cons:
  - Usually needs large data to train a model well
  - High computation costs (at both training and test time); needs GPU acceleration

# Representation Learning

- Automatically learned features also called *embeddings*
- Helps understanding what's learned
- Also enable new ways of using deep models
  - To be discussed later



# Outline

① Web, App, and Business Intelligence

② What's Machine Learning & AI?

③ What's Deep Learning?

④ What's Generative AI?

⑤ Making Smart Apps

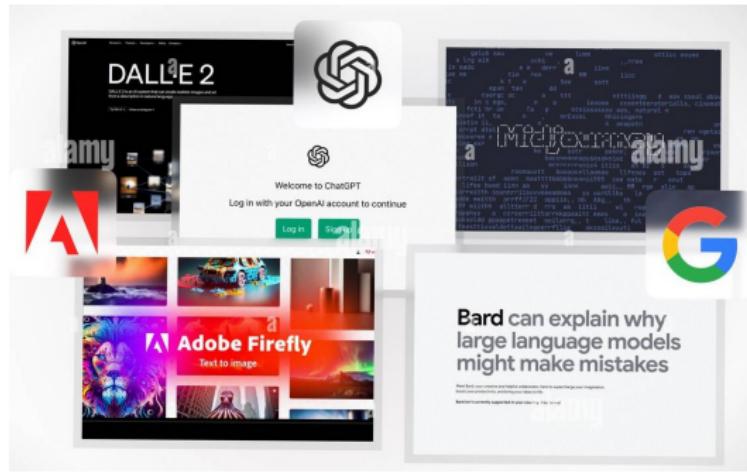
# Generative AI

- The goal
- To generate ***structural*** and ***novel*** output (such as images, text, music, etc.) that cannot be deemed fake by humans



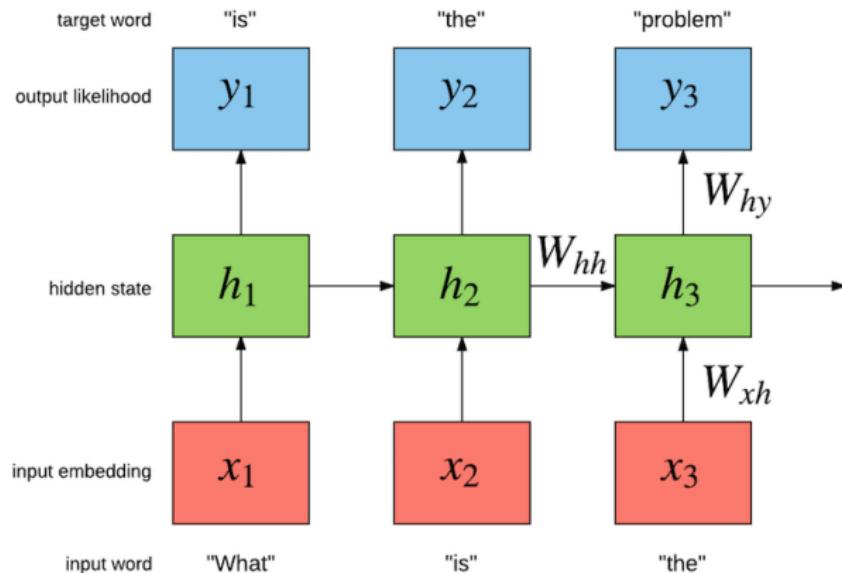
# Generative Models

- The means
- Dataset:  $\mathbb{X} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^N$ , where  $\mathbf{y}^{(i)}$  can be as complex as  $\mathbf{x}^{(i)}$  and **cannot be exhausted**
- Image generation models (e.g., diffusion models)
- Text/language generation models (e.g., GPTs)
- Cross-modal generation models (e.g., GPT4 with vision)

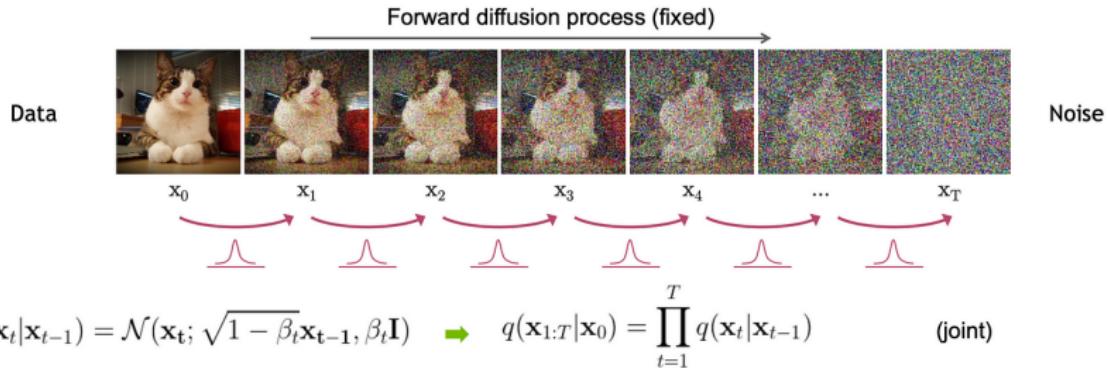


# Autoregressive Models for Text Generation

- Text can be considered as 1D time-series data
- Autoregressive models** takes its previous output as current input
  - Learns the conditional transition distributions of tokens
  - Rather than the joint distribution of all tokens

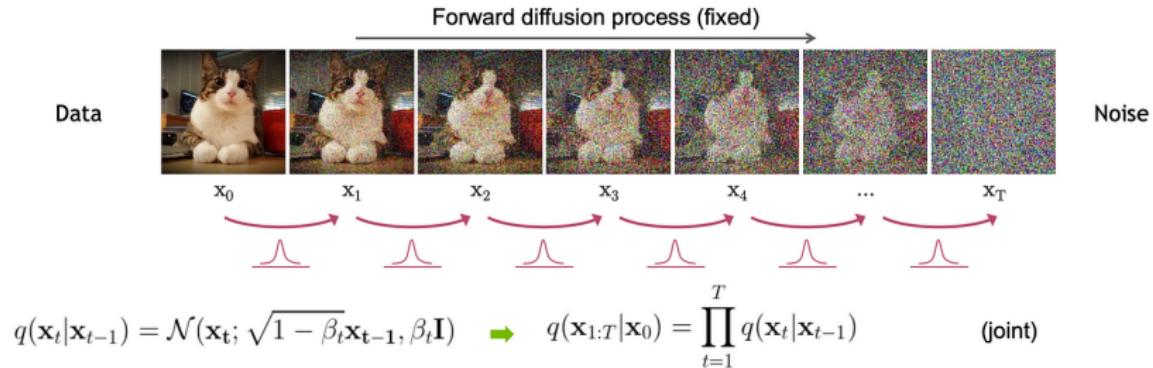


# Diffusion Models for Image Generation



- Note a time-series data naturally; need other strategy to simplify learning

# Diffusion Models for Image Generation



- Note a time-series data naturally; need other strategy to simplify learning
- Forward diffusion is stepwise and deterministic (no learning)
- Model learns to de-noise at each step to generate images
  - Input: noisy image  $\mathbf{x}_t$ , step  $t$ , and  $y$  (e.g., text prompt)
  - Output: less noisy image  $\mathbf{x}_{t-1}$

# Training Trick 1: Self-Supervised Pre-training

- Before training on  $\mathbb{X} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^N$
- Pre-train model on  $\mathbb{X}' = \{(\mathbf{x}^{(i,1)}, \mathbf{x}^{(i,2)})\}_{i=1}^M$ , where  $\mathbf{x}^{(i,1)}$  and  $\mathbf{x}^{(i,2)}$  are parts of the same structural data point
  - Applies to both text and images
- Why?



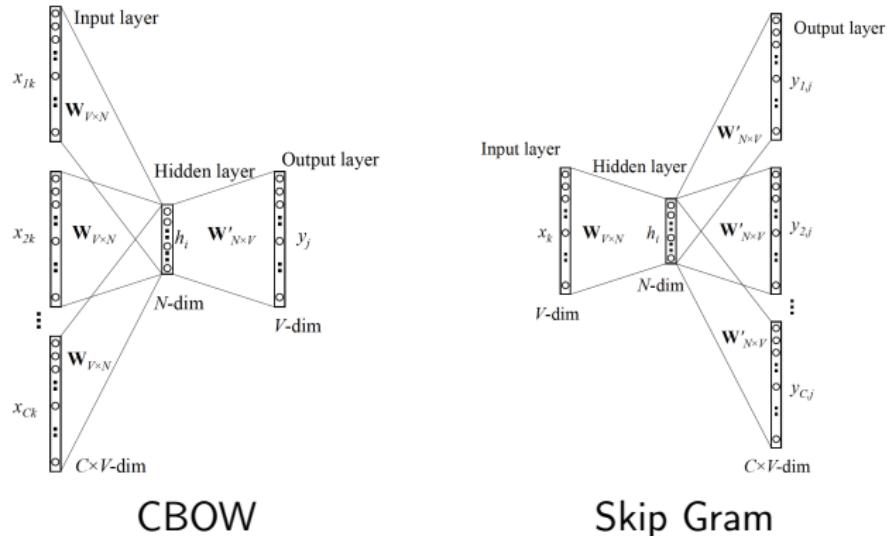
# Training Trick 1: Self-Supervised Pre-training

- Before training on  $\mathbb{X} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^N$
- Pre-train model on  $\mathbb{X}' = \{(\mathbf{x}^{(i,1)}, \mathbf{x}^{(i,2)})\}_{i=1}^M$ , where  $\mathbf{x}^{(i,1)}$  and  $\mathbf{x}^{(i,2)}$  are parts of the same structural data point
  - Applies to both text and images
- Why?  $M \gg N$ 
  - GPT-4 is trained on  $\sim 13$  trillion tokens ( $\sim 10$  trillion words)
  - LAION has 400 million 256X256 images
- Use “common sense” to learn  $\mathbf{y}^{(i)}$  of limited numbers



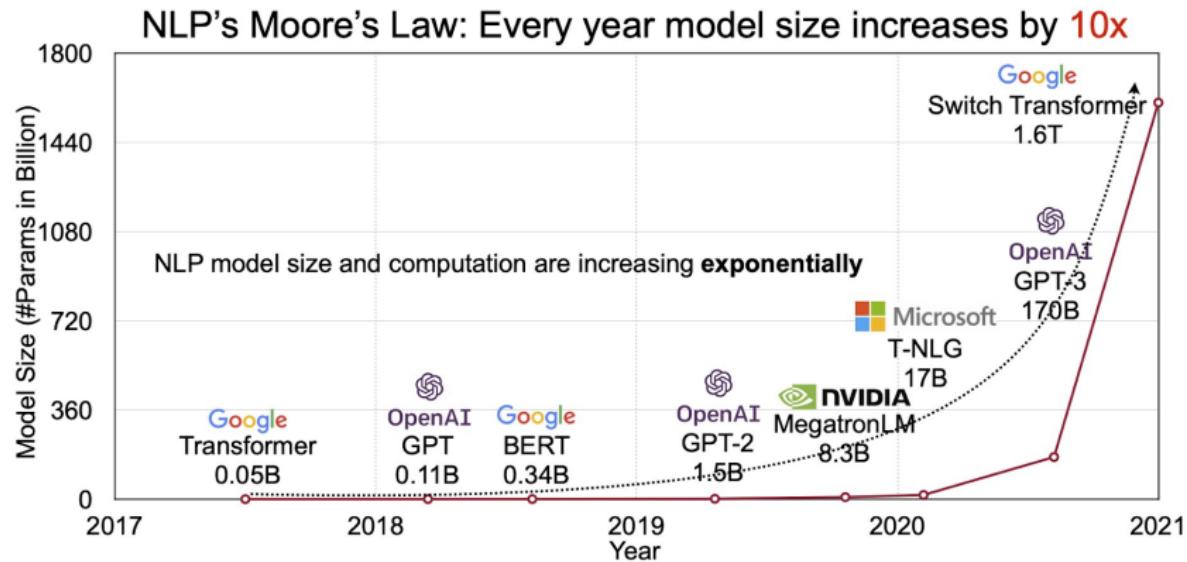
# Semantic Embeddings

- Pre-training deep models enables semantic embeddings
  - Features learned by intermediate layers
- E.g., word2vec [3, 2]: “... *the cat sat* on...”



- Powers modern search and recommendation systems
  - Google Search, Youtube/Tiktok videos, Instagram Feeds, Spotify playlists, etc.

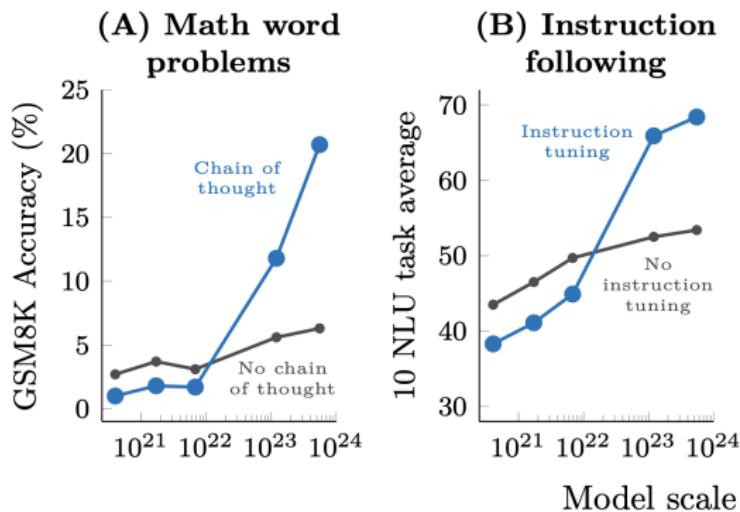
## Training Trick 2: Large Models



- Training costs [4]:
  - 110M params: \$2.5k–\$50k
  - 340M params: \$10k–\$200k
  - 1.5B param: \$80k–\$1.6m

# Size Does Matter!

- Emerging abilities of LLMs [5]



- A balance: 70B parameters + 1.4T training tokens [1]

# Outline

① Web, App, and Business Intelligence

② What's Machine Learning & AI?

③ What's Deep Learning?

④ What's Generative AI?

⑤ Making Smart Apps

# Using Existing ML Models

- Flutter warpper of ML Kit from Google
  - Designed to be run *locally* on mobile devices
- Supports image tasks
  - Barcode scanning, doc scanning, face detection, image labeling, object detection, etc.
- Supports NLP tasks:
  - Language identification, translation, entity (date/time/address/phone number) extraction, smart reply, etc.

# Using Advanced Generative Models

- [OpenAI's APIs](#)
  - Chat, image generation, embeddings, etc.
- Demo
  - Install the “http” package
  - Obtain your API

# Customizing Models (1/2)

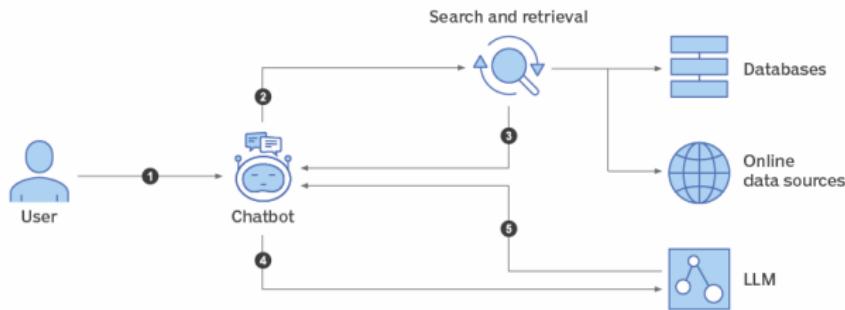
- How to customize a model for your specific tasks?

# Customizing Models (1/2)

- How to customize a model for your specific tasks?
- Fine-tuning model using your own data
  - Not possible if weights are unavailable
- Write better prompts

# Customizing Models (2/2)

- Enable *Retrieval Augmented Generation* (RAG) through Assistant API
  - Demo
  - Does not modify model's weights



- Ask model to perform “actions” defined by you via Function Calling API
  - E.g., code interpreter

## Back to Your Project

To *design & implement* an  
*intellectual app* that solves *real  
problems*.

# Homework: Find Problems Deserve to Solve

- Accounts for ***at least 20%*** of your term project score
- Our next lecture assumes you have found some already
- Tips:
  - What bothers ***you*** or people around you?
  - Make ***few*** people ***very*** happy
  - Avoid “platforms”
  - To have a good idea, you need ***many*** bad ones
  - Understand the root cause
    - The “5 Whys” approach by Sakichi Toyoda

# 5 Whys

- ① Why do some people become the *King of Periods?*

# 5 Whys

- ① Why do some people become the *King of Periods*?
  - Because they might not grasp the context or emotional subtleties
- ② Why might they not grasp the context or emotional subtleties?

# 5 Whys

- ① Why do some people become the *King of Periods*?
  - Because they might not grasp the context or emotional subtleties
- ② Why might they not grasp the context or emotional subtleties?
  - Because they are more logic-oriented or straightforward thinkers
- ③ Why are they more logic-oriented or straightforward thinkers?

# 5 Whys

- ① Why do some people become the *King of Periods*?
  - Because they might not grasp the context or emotional subtleties
- ② Why might they not grasp the context or emotional subtleties?
  - Because they are more logic-oriented or straightforward thinkers
- ③ Why are they more logic-oriented or straightforward thinkers?
  - Because they are CS (or STEM) students
- ④ Why does our education emphasize logic more to these people?

# 5 Whys

- ① Why do some people become the *King of Periods*?
  - Because they might not grasp the context or emotional subtleties
- ② Why might they not grasp the context or emotional subtleties?
  - Because they are more logic-oriented or straightforward thinkers
- ③ Why are they more logic-oriented or straightforward thinkers?
  - Because they are CS (or STEM) students
- ④ Why does our education emphasize logic more to these people?
  - Because the subtle art of emotional navigation is overshadowed
- ⑤ Why are there no educational tools for emotional subtleties?

# 5 Whys

- ① Why do some people become the *King of Periods*?
  - Because they might not grasp the context or emotional subtleties
- ② Why might they not grasp the context or emotional subtleties?
  - Because they are more logic-oriented or straightforward thinkers
- ③ Why are they more logic-oriented or straightforward thinkers?
  - Because they are CS (or STEM) students
- ④ Why does our education emphasize logic more to these people?
  - Because the subtle art of emotional navigation is overshadowed
- ⑤ Why are there no educational tools for emotional subtleties?
  - Because they are complex and situation dependent

# 5 Whys

- ① Why do some people become the *King of Periods*?
  - Because they might not grasp the context or emotional subtleties
- ② Why might they not grasp the context or emotional subtleties?
  - Because they are more logic-oriented or straightforward thinkers
- ③ Why are they more logic-oriented or straightforward thinkers?
  - Because they are CS (or STEM) students
- ④ Why does our education emphasize logic more to these people?
  - Because the subtle art of emotional navigation is overshadowed
- ⑤ Why are there no educational tools for emotional subtleties?
  - Because they are complex and situation dependent
  - It's your turn!

# Reference I

- [1] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al.  
Training compute-optimal large language models.  
*arXiv preprint arXiv:2203.15556*, 2022.
- [2] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean.  
Efficient estimation of word representations in vector space.  
*arXiv preprint arXiv:1301.3781*, 2013.
- [3] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean.  
Distributed representations of words and phrases and their compositionality.  
In *Advances in neural information processing systems*, pages 3111–3119, 2013.

## Reference II

- [4] Or Sharir, Barak Peleg, and Yoav Shoham.  
The cost of training nlp models: A concise overview.  
*arXiv preprint arXiv:2004.08900*, 2020.
- [5] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph,  
Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou,  
Donald Metzler, et al.  
Emergent abilities of large language models.  
*arXiv preprint arXiv:2206.07682*, 2022.