

Background for Machine Learning (Extended Deck)



AnHai Doan

Why AI? Why Now?

- **The world generates a lot of data, and a lot more is coming**

- your smart phone alone generates too much data!
- companies have tons of customer data, and can capture a lot more
- hence the popular saying “the world is increasingly data-driven”

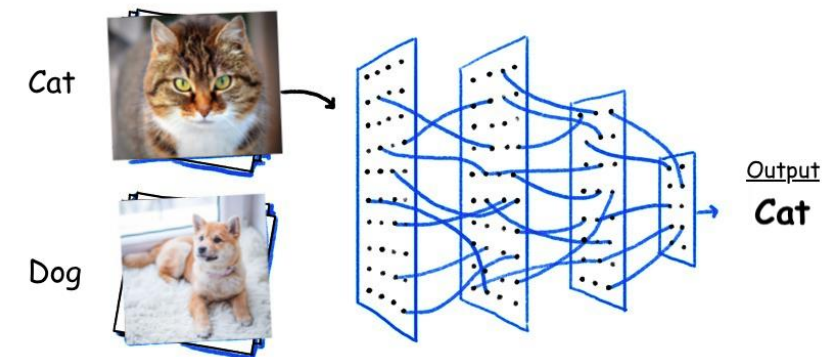
- **People recognize that this data can be super valuable**

- if can mine the data, can extract valuable insights
 - e.g., mining customer data to make recommendation, personalize marketing, etc.
 - e.g., mining user interaction on social media to make recommendation on the next story to read
- hence buzzwords “Big Data” (first appeared around 2008), “data science” (2012)



Why AI? Why Now?

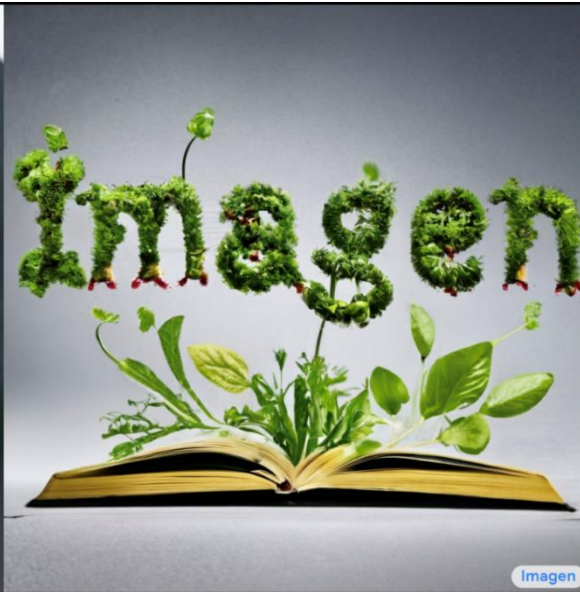
- **Need AI to mine this mountain of data**
 - can't manually examine data to find interesting patterns, there is too much data
- **So in the past decade, AI has becoming critical and ubiquitous**
 - you are now surrounded by AI, from the moment you wake up to when you go to bed
- **Also in the past decade, AI has achieved several major breakthroughs**
 - the biggest ones are in deep learning (DL)
 - DL-powered AI consumes tons of data, can do amazing things, which capture the world's imagination
 - DALL-E, ChatGPT



Examples from DALL-E



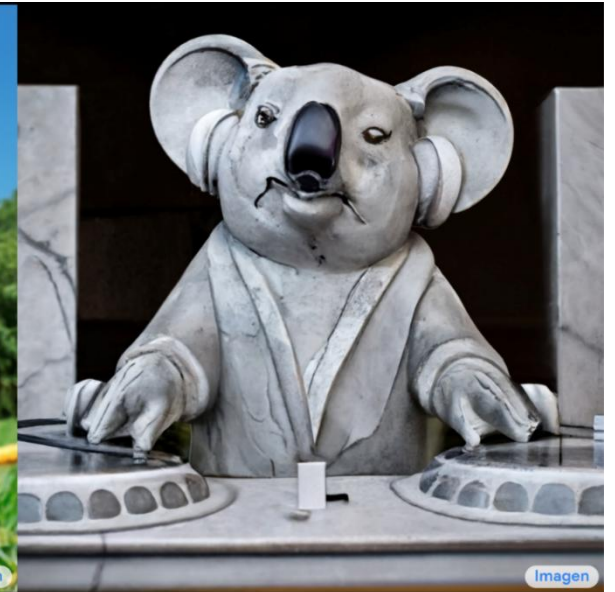
A dog looking curiously in the mirror, seeing a cat.



Sprouts in the shape of text 'Imagen' coming out of a fairytale book.

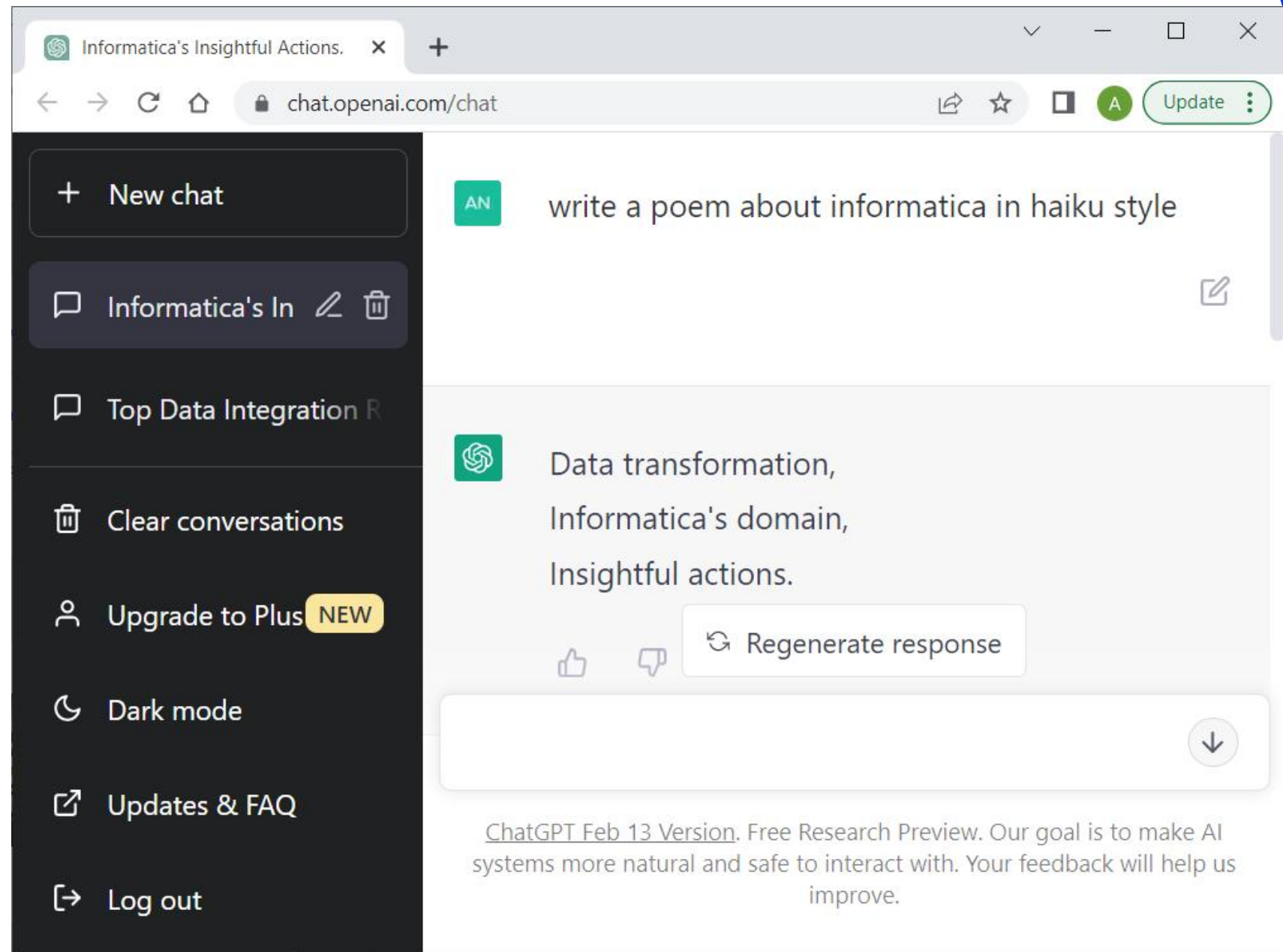


A giant cobra snake on a farm. The snake is made out of corn.

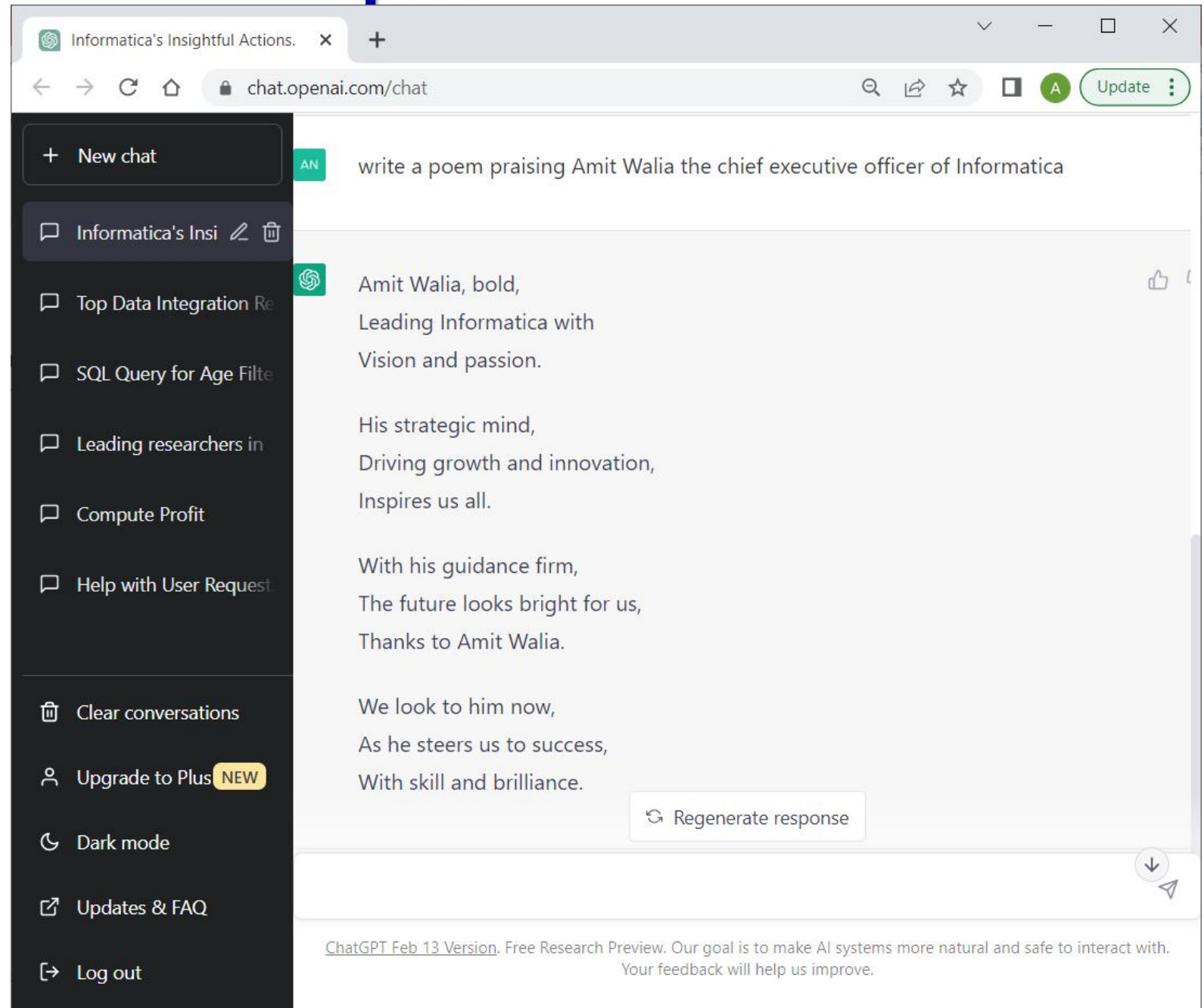


A marble statue of a Koala DJ in front of a marble statue of a turntable. The Koala has wearing large marble headphones.

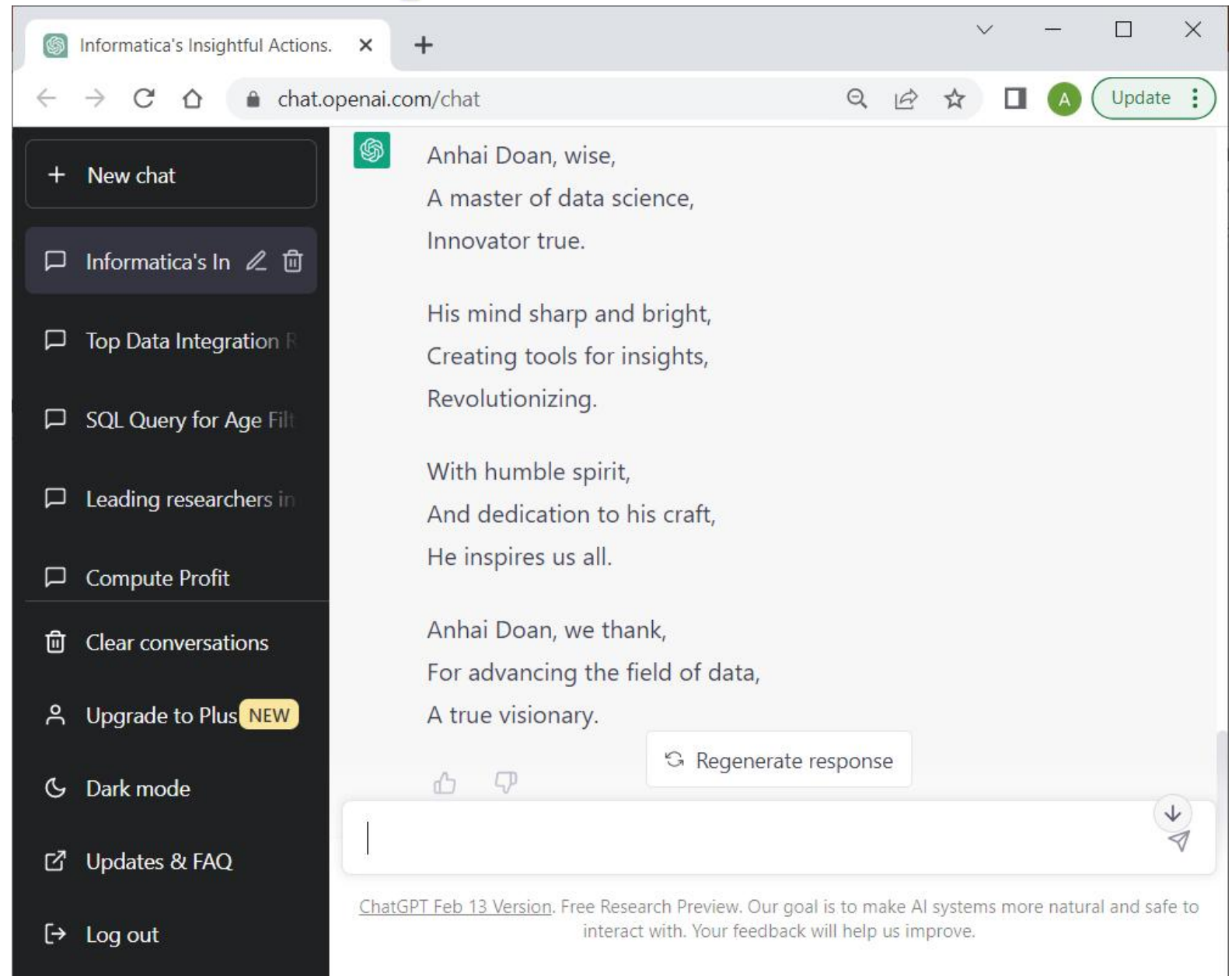
Examples



Examples



Examples



ChatGPT Appears to Know INFA Stuff & Can Calculate

- Q: What does the expression "DATE_DIFF(DATE_PROMISED, DATE_SHIPPED, 'MM')" mean?

- A: This expression is using the **Informatica function DATE_DIFF()** to calculate the difference between two dates, DATE_PROMISED and DATE_SHIPPED, in terms of months ('MM'). The DATE_DIFF() function takes three arguments, the first two are the two dates that you want to find the difference between, and the third argument is the unit of measurement for the difference, in this case 'MM' for months. [...]

Q: what is the result of the expression "DATE_DIFF(DATE_PROMISED, DATE_SHIPPED, 'MM')"

if DATE_PROMISED = 01-JAN-2023 and DATE_SHIPPED = 01-MAY-2021 ?

- A: If DATE_PROMISED = 01-JAN-2023 and DATE_SHIPPED = 01-MAY-2021, then the result of the expression "DATE_DIFF(DATE_PROMISED, DATE_SHIPPED, 'MM')" would be -16. The DATE_DIFF function calculates the difference between two dates in the unit specified in the third argument, in this case, 'MM' for months. If DATE_PROMISED is later than DATE_SHIPPED, then the result will be a negative number, since DATE_SHIPPED is before DATE_PROMISED. So it means that the shipment date is before the promised date by 16 months.

- **All of these generate a lot of buzz about AI and lead many to ask what else can AI do**
 - including in the area of data management

Brief History of AI

- **Started in the 1950s**
- **Many subfields**
 - ML, natural language processing (NLP), vision, planning, etc.
- **We will focus on ML**
 - which dominates AI today
 - and is used heavily in many other subfields, e.g., NLP, vision, speech recognition
- **Three main types of ML**
 - classification, clustering, reinforcement learning
- **Will focus on classification**
 - which dominates ML today

1956 Dartmouth Conference:
The Founding Fathers of AI



John McCarthy



Marvin Minsky



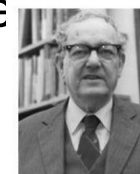
Claude Shannon



Ray Solomonoff



Alan Newell



Herbert Simon



Arthur Samuel



Oliver Selfridge



Nathaniel Rochester



Trenchard More

An Example of Classification

- Learn to classify an animal as DOG or NOT-DOG
- Start with a set of training examples

ID	# of Legs	Need-Daily-Walk	Color	Is-Dog?
1	4	Yes	Black	Yes
2	4	No	Black	No
3	2	No	Yellow	No
4	4	Yes	Yellow	Yes

- Train a ML model M
- Apply M to classify new examples

ID	# of Legs	Need-Daily-Walk	Color	Is-Dog?
5	4	Yes	White	?
6	2	Yes	Black	?

Under the Hood

- **User selects a model template**

- e.g., the ML model M can only have 1 rule of the following form:

IF X = V THEN PREDICT IS-DOG? = YES

where X is a column, V is a value

- **There are many models that fit this template**

- M1: if # of legs = 4 then is-dog? = yes
- M2: if need-daily-walk = yes then is-dog? = yes
- and more

- **Training step finds the model that best fits the training data**

- given a model M, it can compute M's accuracy on the training examples
- then it searches for the model M* that maximizes the accuracy on the training examples

ID	# of Legs	Need-Daily-Walk	Color	Is-Dog?
1	4	Yes	Black	Yes
2	4	No	Black	No
3	2	No	Yellow	No
4	4	Yes	Yellow	Yes

Under the Hood

- **Consider these two models**

- M1: if # of legs = 4 then is-dog? = yes
- M2: if need-daily-walk = yes then is-dog? = yes

- **M1 predicts 3 training examples to be yes, but only 2 are indeed yes**
- **M2 predicts 2 training examples to be yes, and both are indeed yes**
- **So M2 is more accurate than M1**
- **Training step returns the most accurate model (among all models fitting template)**
- **Apply this model to classify new examples**

ID	# of Legs	Need-Daily-Walk	Color	Is-Dog?
1	4	Yes	Black	Yes
2	4	No	Black	No
3	2	No	Yellow	No
4	4	Yes	Yellow	Yes

ID	# of Legs	Need-Daily-Walk	Color	Is-Dog?
5	4	Yes	White	?
6	2	Yes	Black	?

The End-to-End Process

- Create the training examples T (these examples come with correct labels)

ID	# of Legs	Need-Daily-Walk	Color	Is-Dog?
1	4	Yes	Black	Yes
2	4	No	Black	No
3	2	No	Yellow	No
4	4	Yes	Yellow	Yes

- Select a model template V

- Train

- let S be the set of all models that fit template V: $S = \{M1, M2, \dots, M1000\}$
- search for the model M^* in S that best fit training examples T (i.e., is the most accurate for T)
- return model M^*

- Apply

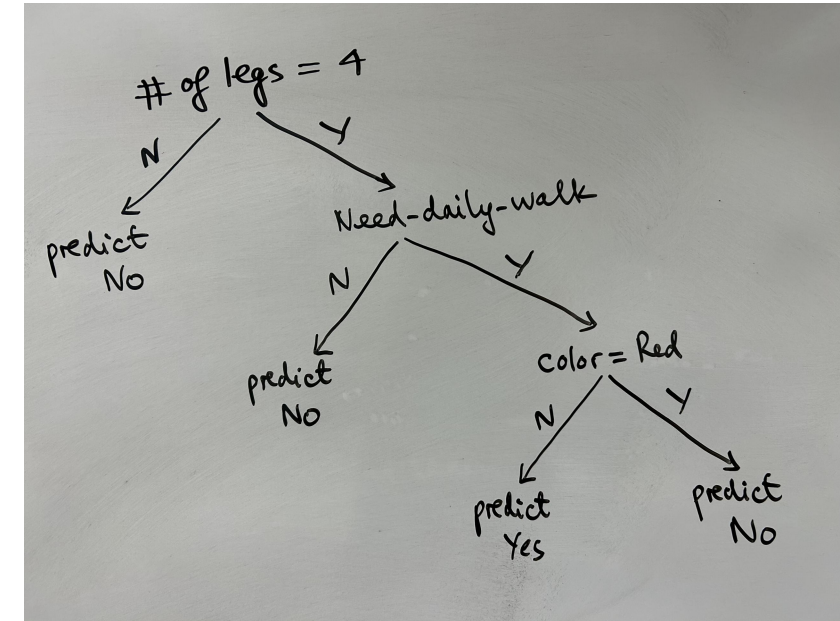
- apply M^* to classify new examples

ID	# of Legs	Need-Daily-Walk	Color	Is-Dog?
5	4	Yes	White	?
6	2	Yes	Black	?

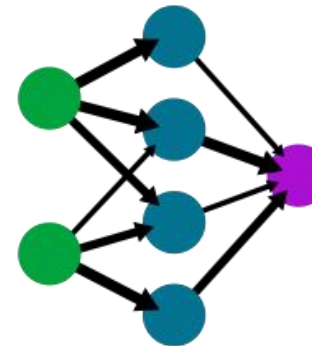
Discussion: Model Template

Can Be Simple or Very Complex

- **Very simple: 1 rule involving 1 column**
 - if a column X has value V, then predict YES
- **A bit more complex: 1 rule involving 2 columns**
 - if a column X has value V and/or a column Y has value U, then predict YES
- **More complex**
 - decision tree, random forest
- **Most complex today**
 - deep neural networks with many layers



A simple neural network
input layer hidden layer output layer



Discussion: How to Pick the Best Model Template?

- **Depends on many factors**

- complexity of the concept we want to learn (dog, hunting dog)
- amount of available training data
- time/hardware available for training

- **Simpler model templates**

- require less training data, take less time to train
- but may not be able to learn complex concepts accurately

- **More complex model templates**

- require a lot more training data, take much longer to train, may require special hardware (GPU)
- but can learn more complex concepts

Discussion: Searching for the Best Model is Difficult

- **A model template can produce numerous models (infinite even)**
- **Defining a notion of accuracy on the training data is difficult**
- **Searching for the most accurate model is even more difficult**
 - the more complex the model template, the more difficult the search is
 - can't just enumerate ALL models, compute its accuracy, then pick the best one
 - need to do something clever, often can't find the best one, but can find a good one
 - can take a lot of time, may require special hardware (to do parallel computations fast)
- **You don't have to worry about these**
 - the training procedure will do the search for you

Discussion: Terminologies

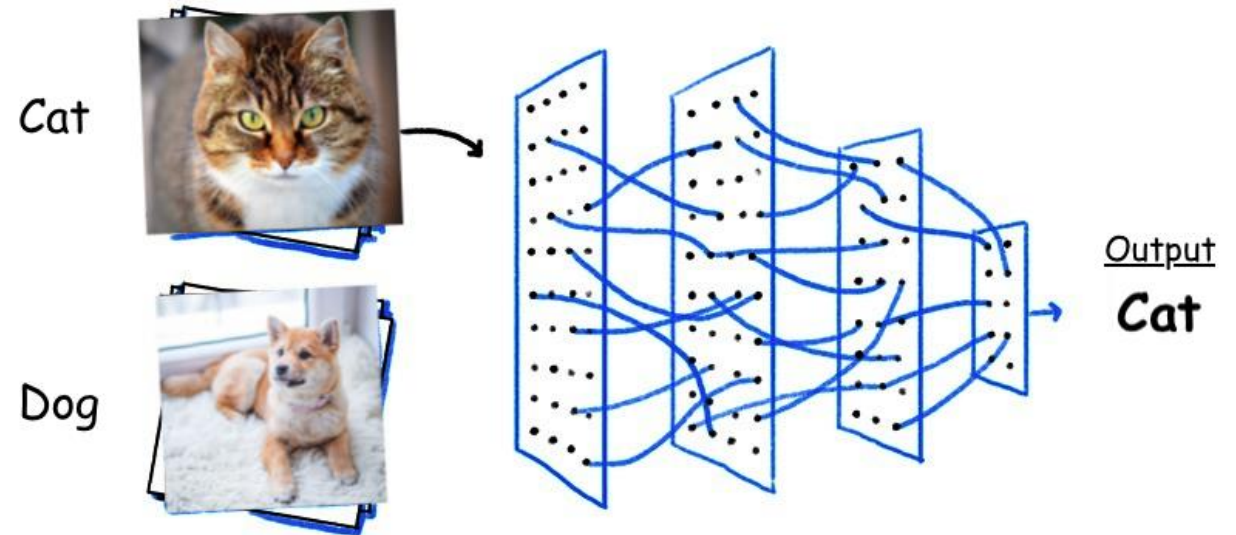
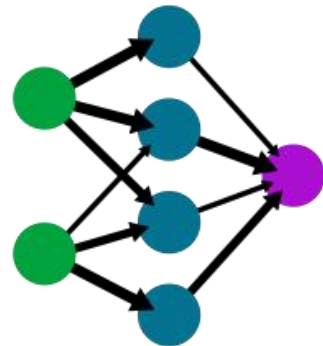
- Create the training examples T (these examples come with correct labels)
- Select a **model template** V
- Train
 - let S be the set of all models that fit template V : $S = \{M_1, M_2, \dots, M_{1000}\}$
 - search for the model M^* in S that best fit training examples T (i.e., is the most **accurate** for T)
- Apply
 - apply M^* to classify new examples
- **Terminologies**
 - model template \rightarrow often called “**model**”, “**algorithm**”, “**solution**” (decision tree, neural networks)
 - train step \rightarrow often called “**training the model**”
 - the best model returned by train step \rightarrow often called “**the trained model**”
 - accuracy on training data \rightarrow often called “**the loss function**”
 - **training is an optimization problem: find the model that minimizes the loss of accuracy on training data**

Popular Non-DL Models

- **Up to 2012, many popular non-DL models (aka model templates) exist**
 - decision tree, random forest
 - SVM, linear regression, logistic regression
 - Naïve Bayesian classifier, probabilistic classifiers
- **DL models are known as neural networks**
- **They existed before 2012, but did not seem to be very accurate then**
 - so were not very popular
- **Then something happened in 2012, and DL really took off**

Neural Networks

A simple neural network



- Graph with weights on edges, multiple layers, often used for classification
- Training: find the weights that minimizes accuracy loss on training data

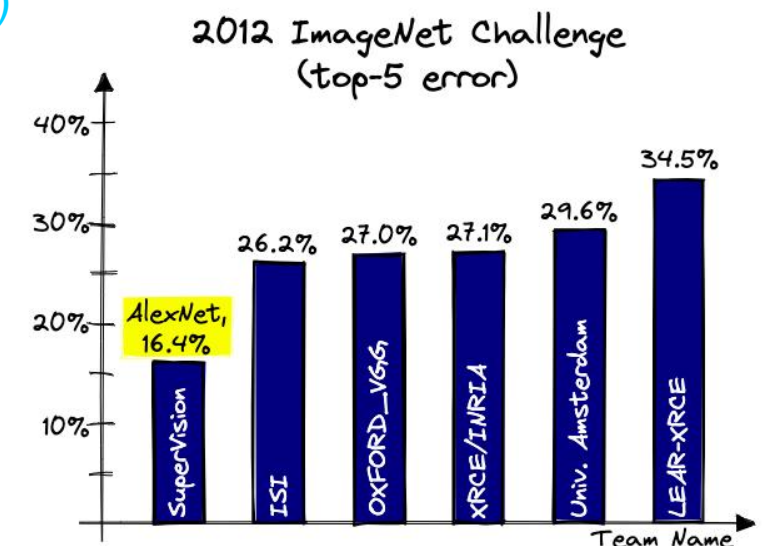
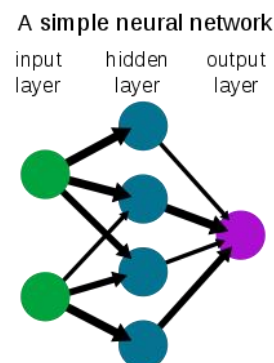
Why Didn't They Prove Popular Before 2012?

- Long story, but briefly

- people experimented with NNs that have just a few layers
- they couldn't experiment with NNs with many layers, because training took way too long
 - didn't have fast training algorithms
- also, fed too few training examples into the network (e.g., just a few hundred thousands)
- overall, these NNs achieved lower accuracy than other solutions, so were ignored

- Then multiple changes converged in 2012

- invented better training algorithms, had better hardware (GPU)
 - can train NNs with many layers (→ deep NNs → deep learning)
- lot more training data became available (tens of millions)
- when feeding these deep NNs a lot of training examples, finally magic happened, accuracy skyrocketed!



DL Took Off in 2012

- **What people learned then**

- DL models are the most complex model templates that we know today
- if we feed them a lot of training data, they can become very accurate
 - beat all other existing methods
- **the deeper the model, the more training data, the more accurate the models seem to become**

- **So two major directions**

- let's **apply DL to all kinds of tasks**: image recognition, self-driving cars, speech, natural language processing, misinformation detection, etc.
- let's **make things bigger**: make models deeper, feed them more training data
 - let's see what happens
 - this led to the recent development of foundation models (aka large language models)
 - **but to pursue this direction of “bigger is better”, need to solve two major challenges**

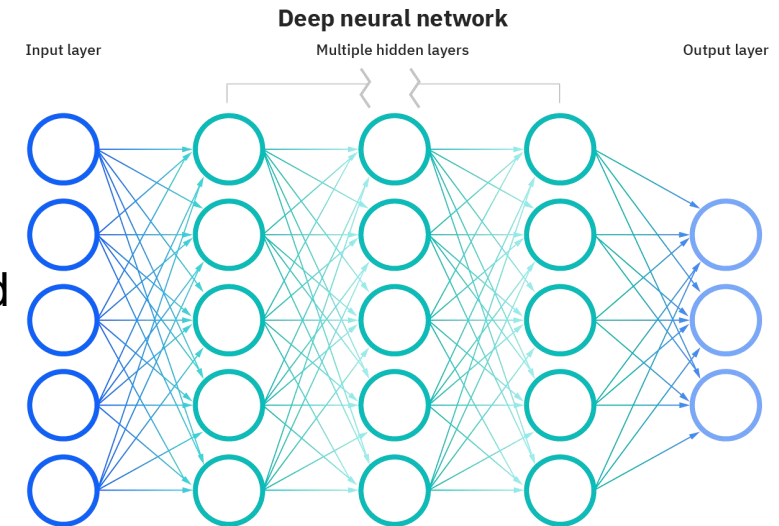
Solving Two Major Challenges for “Bigger is Better”

- **How to generate a lot of training data?**

- (input, correct output) pairs
- self supervised learning
 - e.g., given a sentence with a word masked, predict that word
 - Apple **announced** quarterly earnings today → **announced**
 - training = adjusting the weights on the edges

- **How to process a lot of training data**

- hundreds of billions or trillions of sentences, in tens of billions of documents
- had to process each sentence **sequentially**, took way too long
- Google published a paper in 2017 showing a way to process sentences in parallel
- so it became possible to train on tens of billions of documents → produced foundation models



Foundation Models (a.k.a. Large Language Models)

- **Wikipedia:** A **foundation model** is a large artificial intelligence model
 - trained on a vast quantity of unlabeled data at scale (usually by self-supervised learning)
- resulting in a model that can be adapted to a wide range of downstream tasks
 - this is why they are called “foundation models”
 - also known as “large language models”

Some Popular Foundation Models

Model	Year	Company	# of parameters	Training dataset	Notes
GPT	2018	OpenAI	117M	BooksCorpus	First pretrained Transformer model
BERT _{large}	2018	Google	340M	Wikipedia, Google's BooksCorpus	
GPT-2	2019	OpenAI	1.5B	WebText	Improved and bigger version of GPT
DistilBERT	2019	Hugging Face	66M	Same as BERT	A distilled version of BERT that is 60% faster, 40% lighter in memory but maintains 97% performance
BART	2019	Facebook	400M	Corrupted and noisy text	Bidirectional encoder + left-to-right decoder
T5-11B	2020	Google	11B	Colossal clean crawled corpus (C4)	Newer version FlanT5 (2022)
GPT-3	2020	OpenAI	175B	Filtered Common Crawl, WebText2, Books1, Books2, Wikipedia	Perform well on a various tasks without fine-tuning. InstructGPT (2022, 1.3B) and its sibling ChatGPT are finetuned from GPT-3 with reinforcement learning from human feedback.
Megatron-Turing NLG	2022	Microsoft + NVIDIA	530B	The Pile, Common Crawl snapshots, RealNews, Common Crawl Stories	3D parallelism methodology is used to train the model using MS DeepSeed and NVIDIA Megatron

Fine Tuning a Foundation Model for a Task X

- OpenAI fine tuned GPT3.5 to chat → ChatGPT
- Another example: translating NL query Q to SQL query S
- The aboves are examples of **generative AI**
 - before: output a label (yes/no), now: can produce complex output
 - text → text, text → image, image → text, text → SQL, text → code, etc.
 - very popular right now

Why Does T5 Work Well Here?

- **Foundation models learn the statistical correlations among words**
 - which word typically occurs AFTER a phrase
 - which two words / phrases often occur in the SAME context → synonyms
 - including misspellings
- **So it can handle variations/synonyms in the input**
 - return all tables with at least 3 columns, find all tables ..., get me those table ...
- **It can also produce accurate output text**
 - using correlations it has learned, such as after having seen SELECT and FROM, it typically will see WHERE
- **But these correlations can also mislead the models → statistical parrots**

Statistical Parrot

Who is AnHai Doan?

AnHai Doan is a computer science researcher and professor at the University of Wisconsin-Madison. He is a leading expert in the field of data management and has published numerous papers and articles on topics such as **data integration, data cleaning, data quality, and data provenance**. He is a well-known researcher in the database community and has received several awards for his contributions to the field. His research focuses on developing techniques and systems to make data integration and cleaning more effective and efficient. His team develop system like DataWrangler, Data Cleaner, and Data Profiler etc. He is also known for his contributions to the field of data management in general, including work on data integration, data cleaning, data quality, and data provenance.

- **I haven't worked on "data provenance"**
- **But the phrase "data integration, data cleaning, data quality, and data provenance" very commonly occurs online (in training data that it has seen)**
 - so ChatGPT generated the whole thing

Project Suggestions

- **We focus on classification**
 - but you can do clustering and reinforcement learning if you'd like
- **Projects not using foundation models**
 - **classification using non-DL techniques**
 - take a problem, frame it as a classification problem into a set of labels
 - obtain labeled data, train, apply (maybe using random forests, decision trees)
 - **classification using DL techniques**
 - similar to above, but now use deep neural networks

Project Suggestions

- **Projects using foundation models**

- **investigate how foundation models can help improve the accuracy of data tasks**
- **generative AI**
 - text to image, image to text, text to text, text to SQL, SQL to text, etc.
- **build natural language (NL) interfaces**
- **investigate what else you can do with foundation models**
- **perhaps a combination of some of the aboves**

Further Resources

- Scikit-learn website has a lot of good documentation
- Books

