EVIDYA DSML

More on deep learning models

Hariprasad Kodamana, Agam Gupta, Manojkumar Ramteke
IIT DELHI

Images courtesy : internet

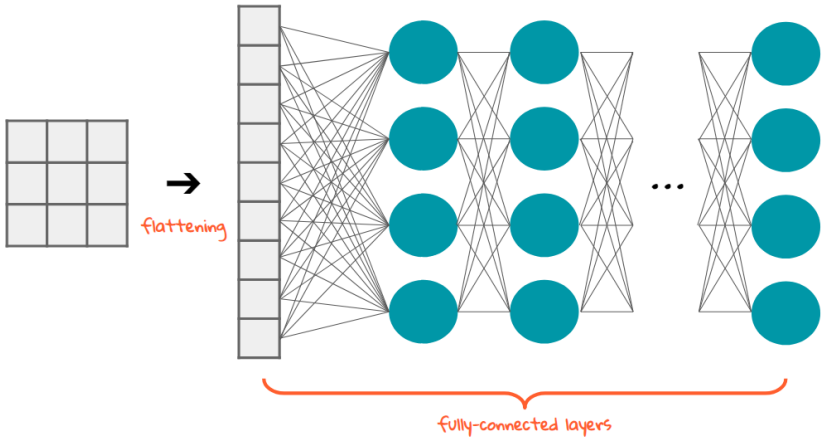
Overview of Presentation

The problem of data tensors

- Image classification is the task of taking an input image and outputting a class (a cat, dog, etc) or a probability of classes that best describes the image.
- When a computer akes an image as input, it will see an array of pixel values, let us consider 480 pixel JPG , it will see a 480 x 480 x 3 array of numbers (The 3 refers to RGB values)
- Each of these numbers is given a value from 0 to 255 which describes the pixel intensity at that point
- These numbers, while meaningless to us when we perform image classification, are the only inputs available to the computer

The problem of data tensors

- The machine has to process this array of numbers and output numbers that describe the probability of the image being a certain class (.80 for cat, .15 for dog, .05 for bird, etc)
- Training MLP is near impossible (480 x 480 x 3=691200 parameters)



Convolution operation

- In its most general form, convolution is an operation on two functions of a real valued argument
- Suppose we are tracking the location of a spaceship with a laser sensor. Our laser sensor provides a single output $x(t)$, the position of the spaceship at time t. Both x and t are real valued, that is, we can get a different reading from the laser sensor at any instant in time.
- Now suppose that our laser sensor is somewhat noisy and to obtain a less noisy estimate of the spaceship's position, we would like to average several measurements
- We can do this with a weighting function $w(a)$, where a is the age of a measurement

$$s(t) = \int x(a)w(t-a)da \tag{1}$$

- This operation is called convolution denoted as $s = (x * w)(t)$

Convolution operation: Importance of features

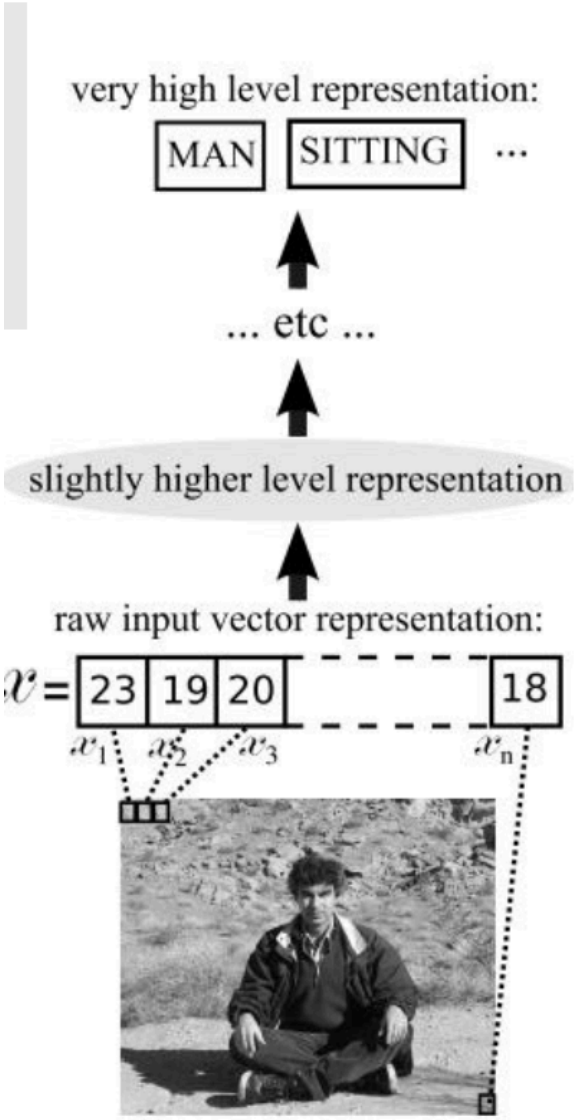
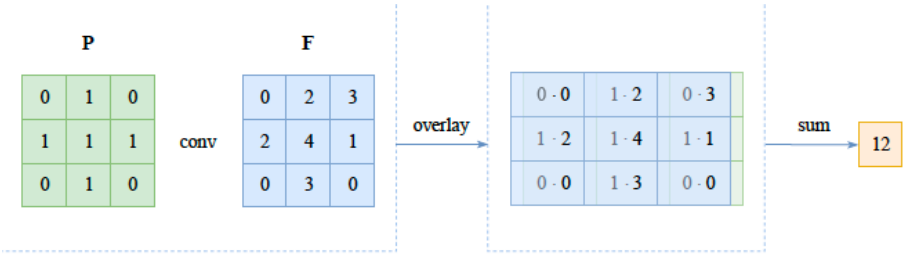


Figure is from Yoshua Bengio

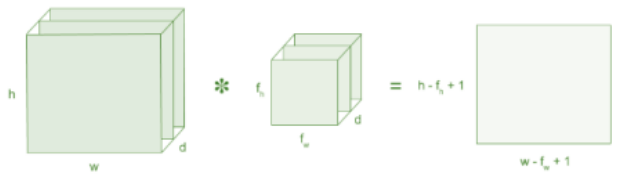
- Learning feature hierarchies where features from higher levels of the hierarchy are formed by lower level features
- Low level representation: pixel numbers
- slightly high level representation: edges, local shapes, object parts
- High level representation: Full object, posture

Convolution of matrices

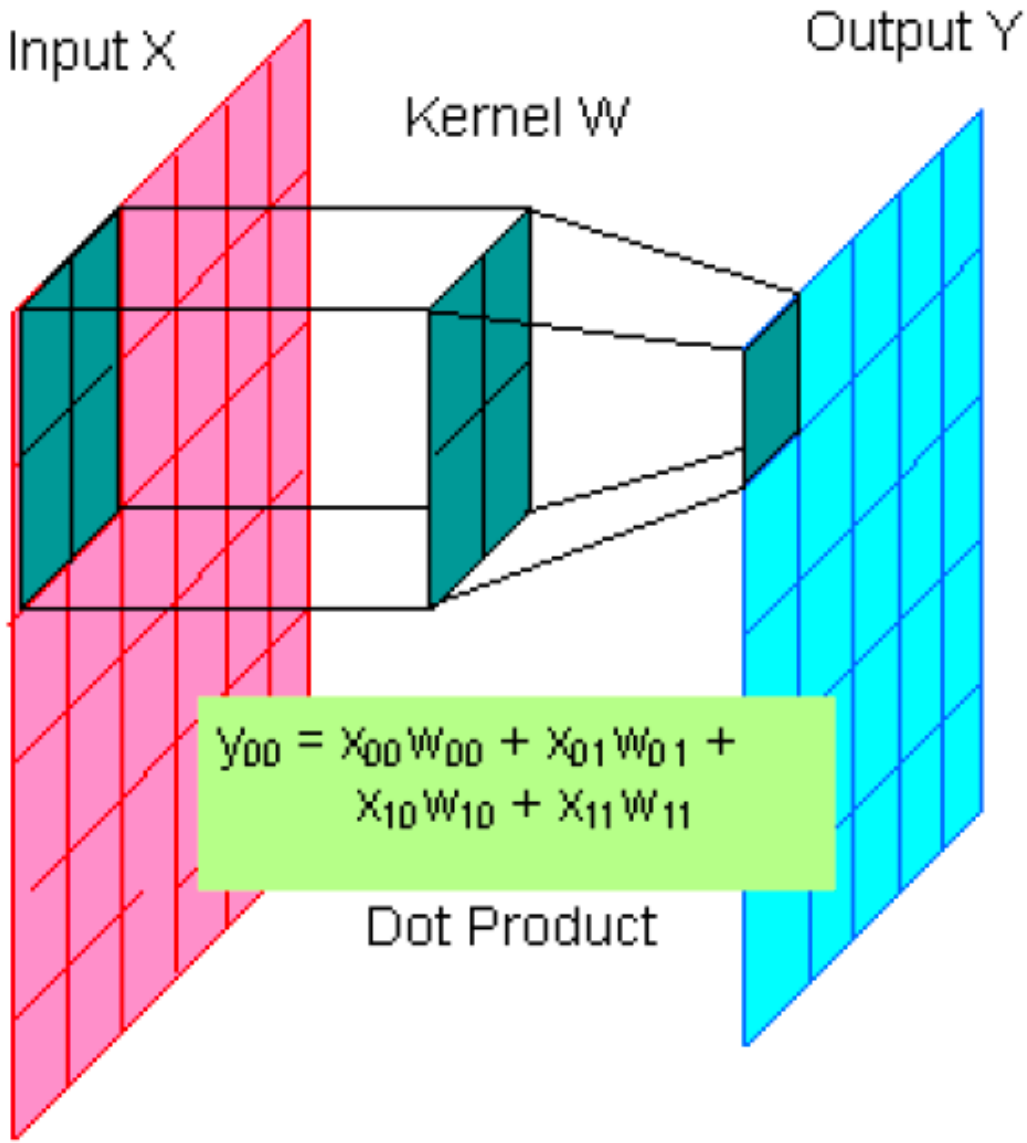
- Convolution is the first layer to extract features from an input image
- Convolution preserves the relationship between pixels by learning image features using small squares of input data
- It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel



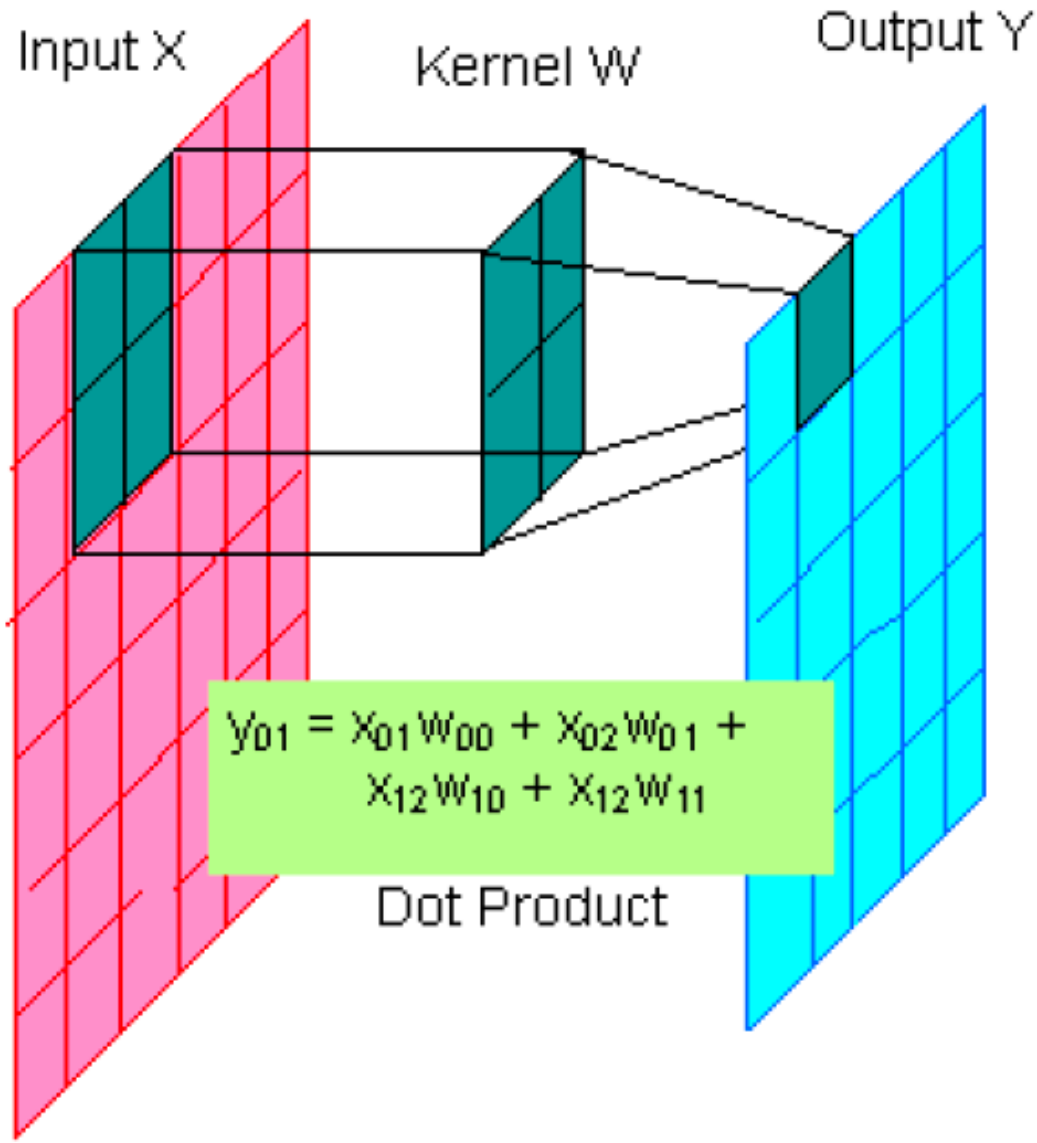
- An image matrix (volume) of dimension **(h x w x d)**
- A filter **(f_h x f_w x d)**
- Outputs a volume dimension **(h - f_h + 1) x (w - f_w + 1) x 1**



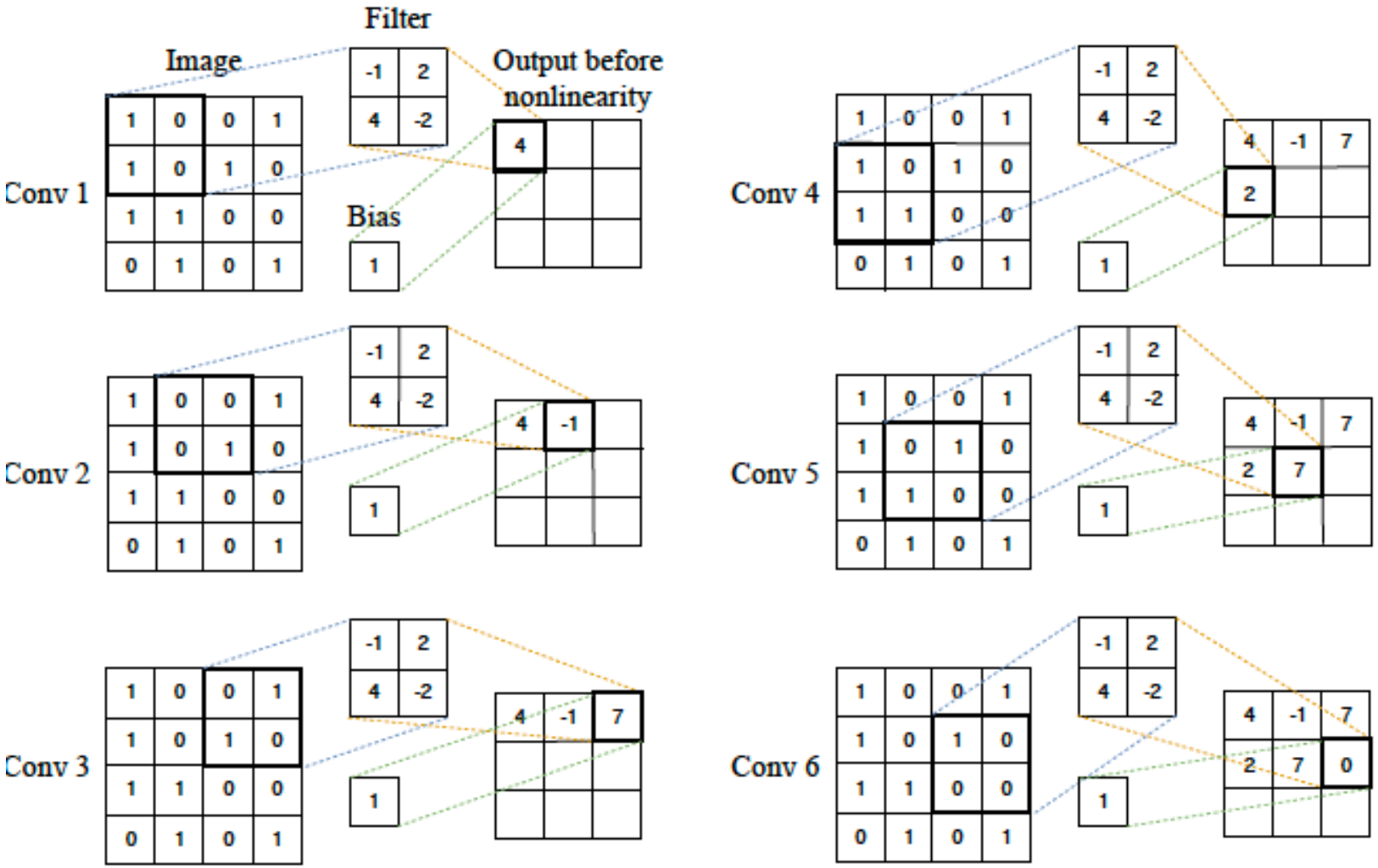
Convolution operation



Convolution operation



Convolution operation



Convolution operation: Importance of features

Input image

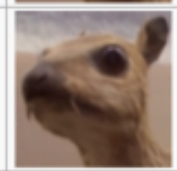
Convolution Kernel

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Feature map

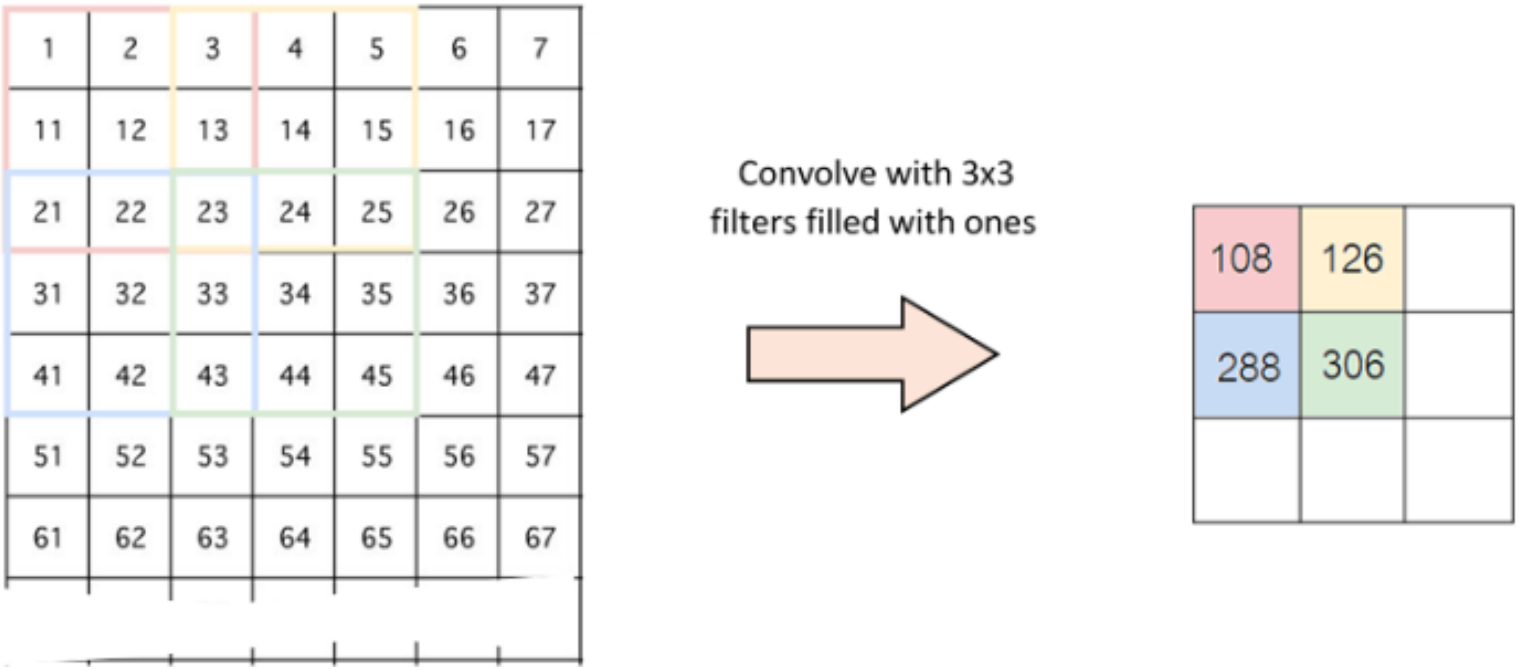
- Not all data/image segments are important in decision making
- Features are most relevant information extracted from data/image that are helpful decision making
- Traditional algorithms consider feature extraction and learning as independent tasks
- If one can optimize the feature selection also, then learning can be improved tremendously

Convolution operation: Importance of features

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

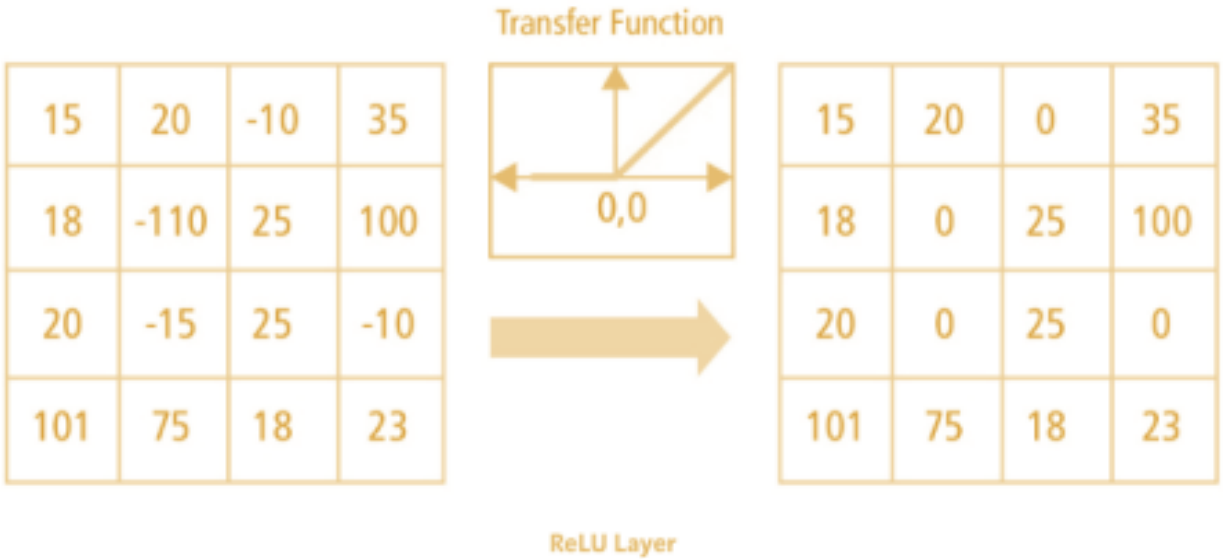
Stride

- Stride is the number of pixels shifts over the input matrix
- When the stride is 1 then we move the filters to 1 pixel at a time
- When the stride is 2 then we move the filters to 2 pixels at a time and so on
- The below figure shows convolution would work with a stride of 2.



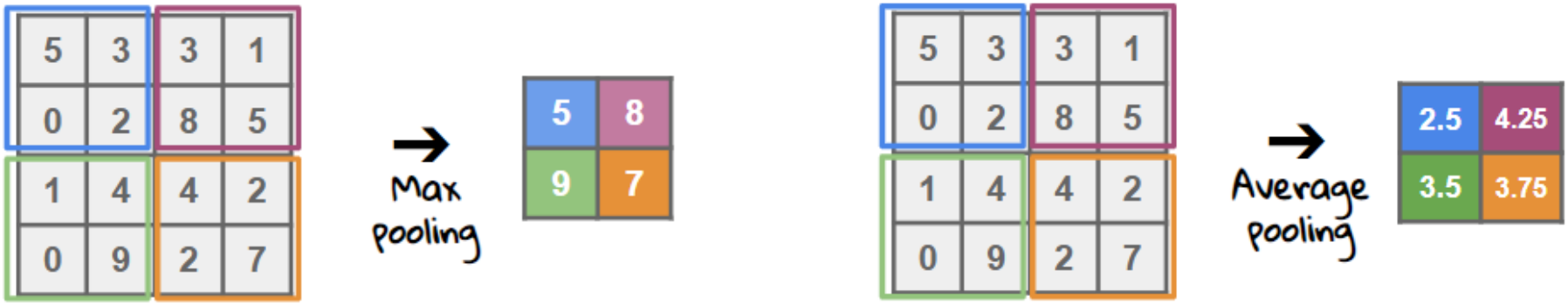
Nonlinear activation

- Various non-linear functions such as tanh or sigmoid or ReLU can be used
- Most of the data scientists use ReLU since performance (both speed and accuracy) wise ReLU is better than the other two
- ReLU stands for Rectified Linear Unit for a non-linear operation. The output is $f(x) = max(0, x)$



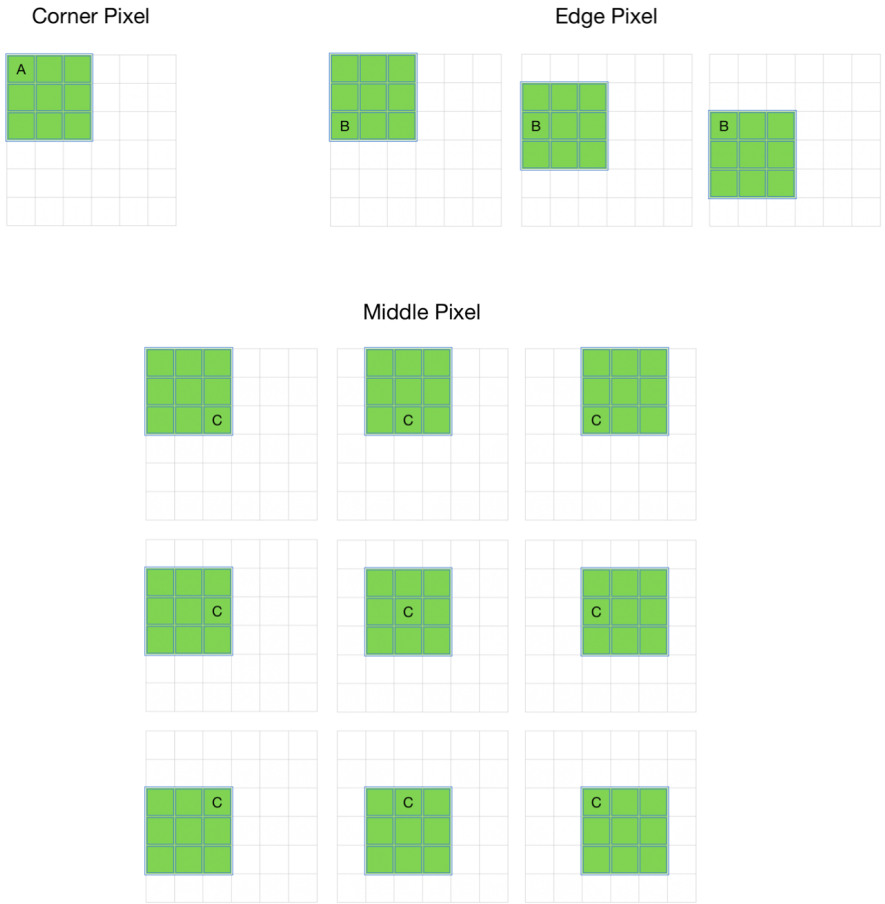
Pooling layer

- Pooling layers section would reduce the number of parameters when the images are too large
- Spatial pooling also called subsampling or downsampling which reduces the dimensionality of each map but retains important information
  - Max Pooling-takes the largest element from the rectified feature map
  - Average Pooling-mean of all elements in the feature map
  - Sum Pooling-sum of all elements in the feature map



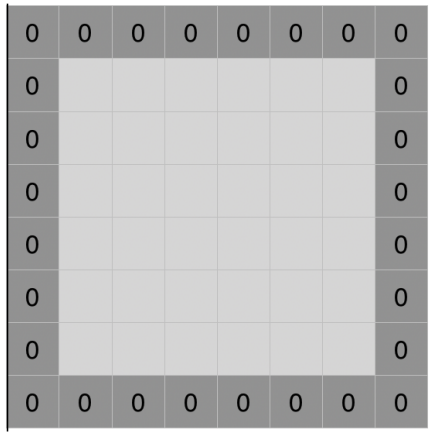
Padding operation

- In general, pixels in the middle are used more often than pixels on corners and edges. Consequently, the information on the borders of images are not preserved as well as the information in the middle.



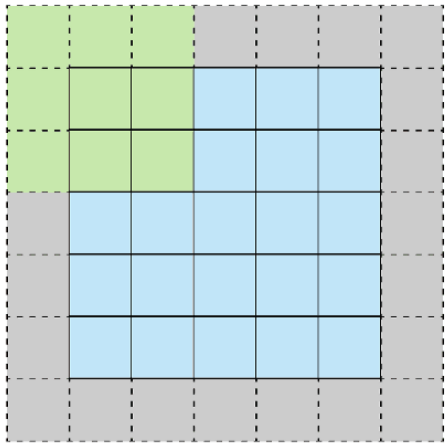
Padding operation

- Padding is simply a process of adding layers of zeros to the input images
- If p = number of layers of zeros added to the border of the image, then our (nxn) image becomes $(n+2p)x(n+2p)$ image after padding.
- So, applying convolution-operation (with (fxf) filter) outputs $(n+2p–f+1)x(n+2p–f+1)$ images
- For example, adding one layer of padding to an $(8x8)$ image and using a $(3x3)$ filter we would get an $(8x8)$ output after performing convolution operation.

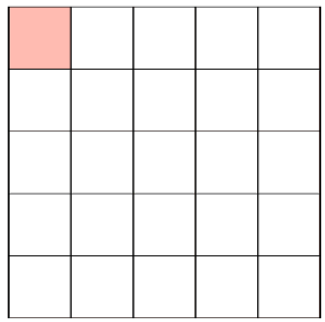


Padding operation

- Valid Padding : It implies no padding at all. The input image is left in its valid/unaltered shape. So, $[(nxn)image] * [(fxf)filter] \rightarrow [(n–f+1)x(n–f+1)image]$
- Same Padding : In this case, we add $'p'$ padding layers such that the output image has the same dimensions as the input image. So, $[(n+2p)x(n+2p)image] * [(fxf)filter] \rightarrow [(nxn)image]$ or
- This increases the contribution of the pixels at the border of the original image by bringing them into the middle



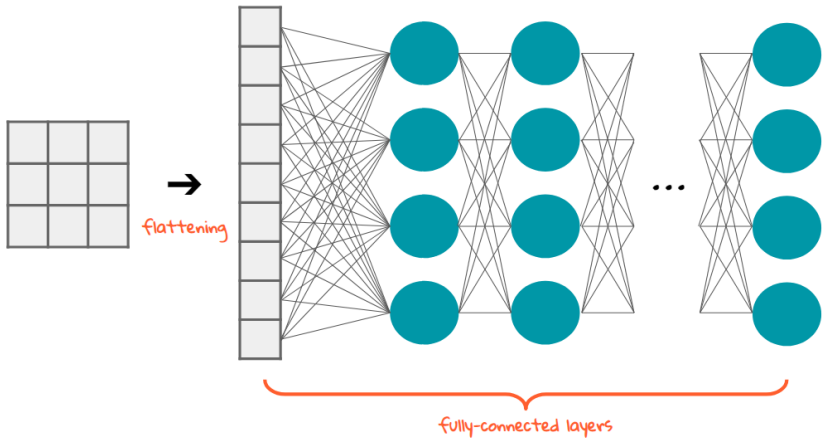
Stride 1 with Padding



Feature Map

Fully connected layer

- The layer we call as FC layer, we flattened our matrix (whose size is much much lesser than original image) into vector and feed it into a fully connected layer like a neural network

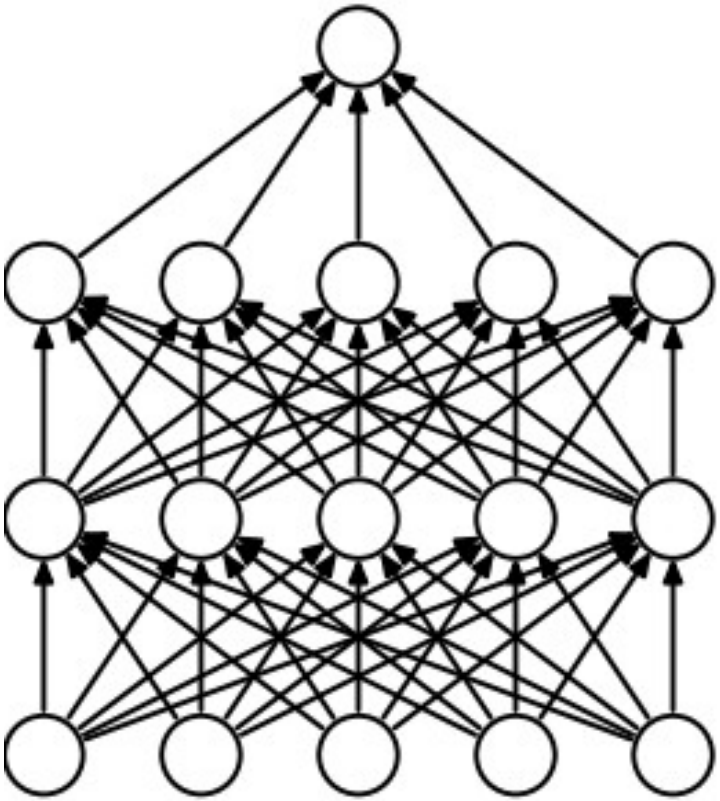


- In the above diagram, the feature map matrix will be converted as vector
- With the fully connected layers, we combined these features together to create a model
- Finally, we have an activation function such as softmax or sigmoid to classify the outputs as cat, dog, car, truck etc.,

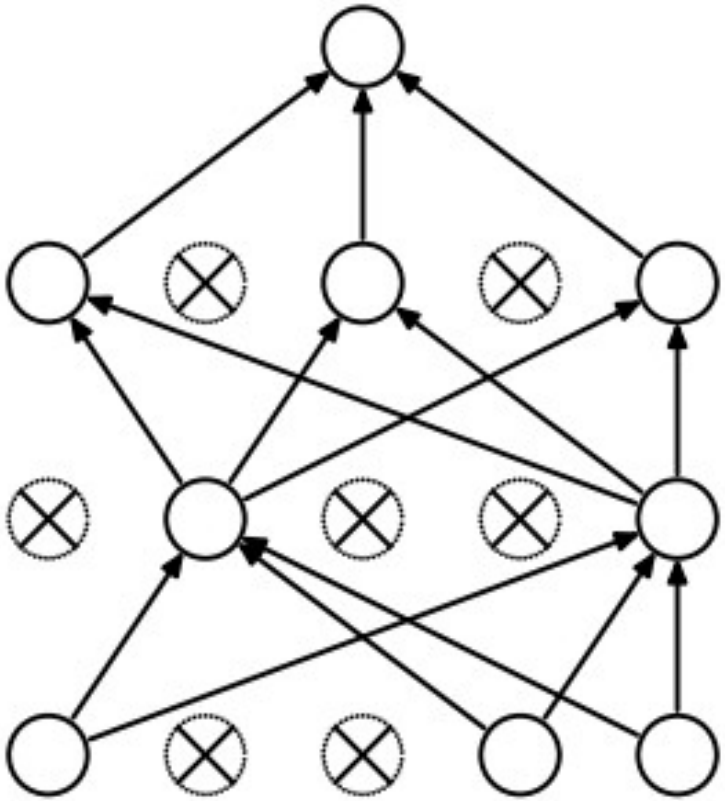
Dropout

- Dropout: set the output of each hidden neuron to zero with probability p
- The neurons which are "dropped out" in this way do not contribute to the forward pass and do not participate in backpropagation
- So every time an input is presented, the neural network samples a different architecture, but all these architectures share weights
- This technique reduces complex co-adaptations of neurons, since a neuron cannot rely on the presence of particular other neurons
- It is, therefore, forced to learn more robust features that are useful in conjunction with many different random subsets of the other neurons
- Without dropout, the network may exhibit substantial overfitting

Dropout

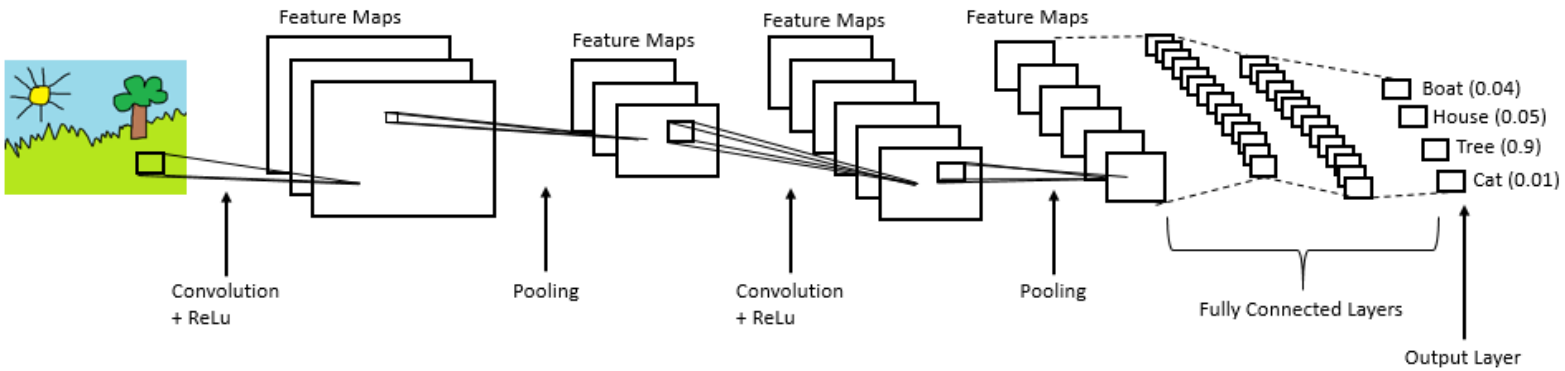


(a) Standard Neural Net



(b) After applying dropout.

CNN-full network

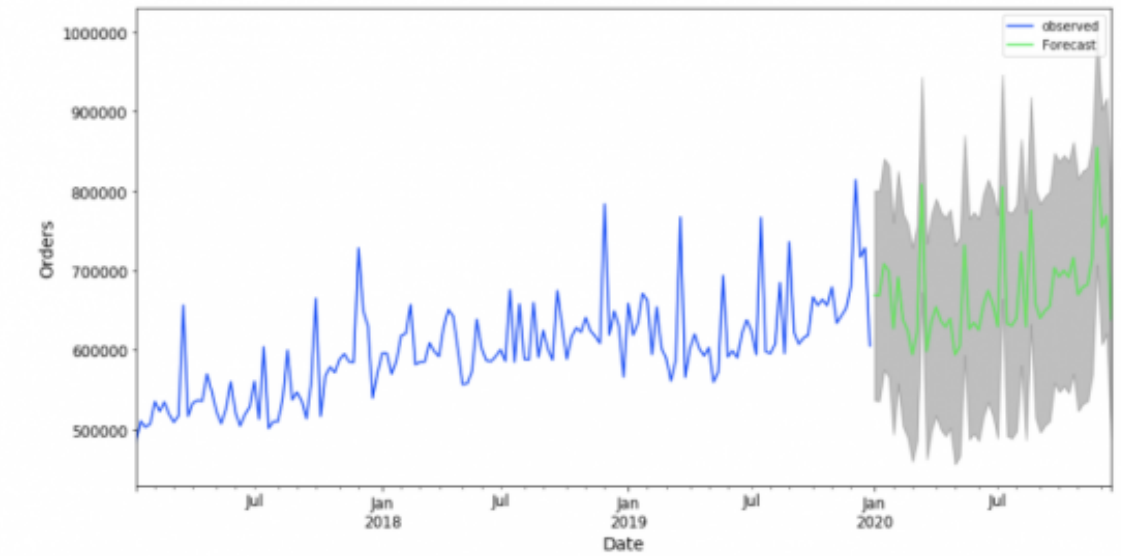


- Provide input image into convolution layer
- Choose parameters, apply filters with strides, padding if requires
- Perform convolution on the image and apply activation to the matrix.
- Perform pooling to reduce dimensionality size
- Add as many convolutional layers until satisfied
- Flatten the output and feed into a fully connected layer (FC Layer)
- Output the class using an activation function (Logistic Regression with cost functions) and classifies images

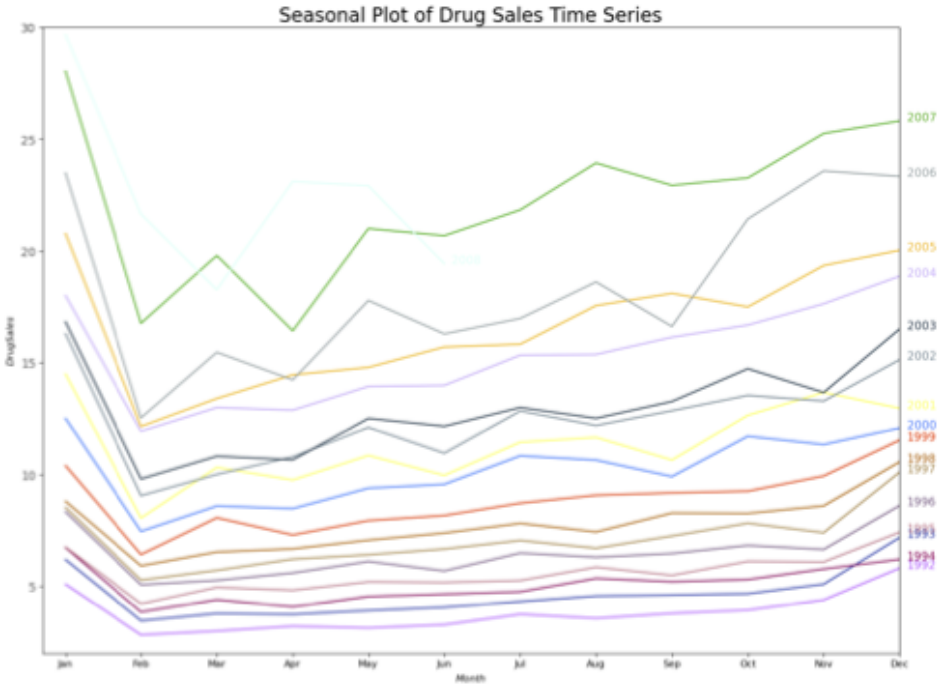
Overview of Presentation

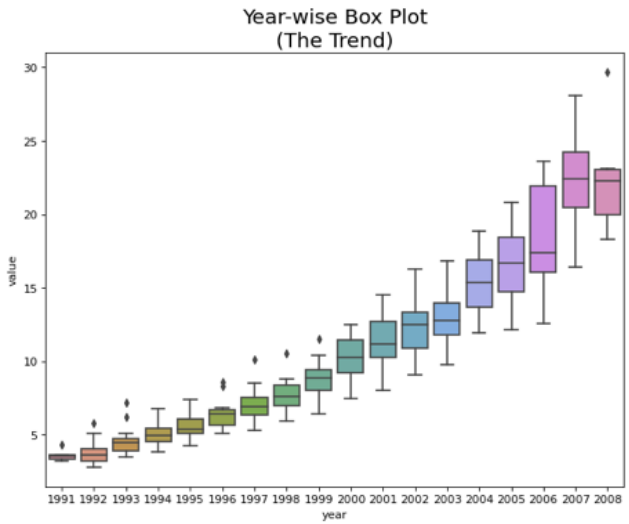
Time series data

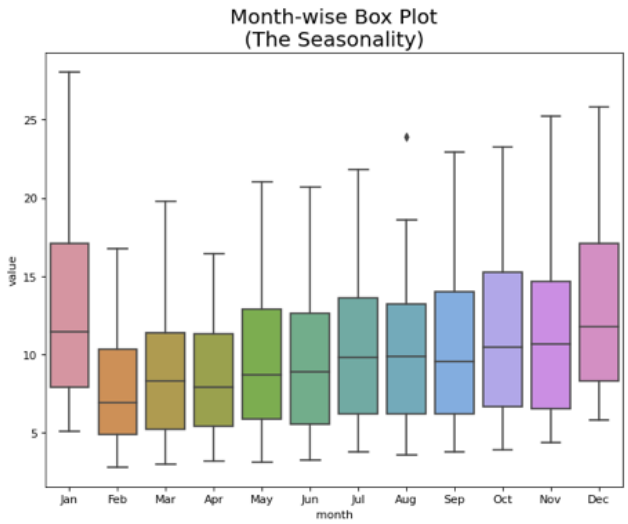
- A time series analysis focuses on a series or sequential data points ordered in time
- Possibly repetitive and seasonal
- Applications : Forecast daily sales volumes, forecast stock prices, weather forecasting



Visualization





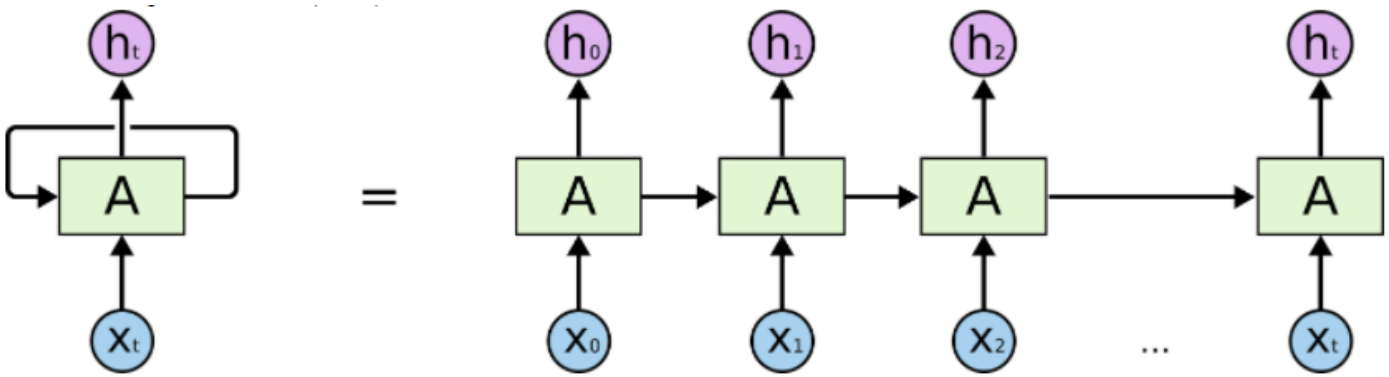


Overview of Presentation

Need of RNNs

- CNNs can also learn sequential series, but only when the required information is very near
- RNNs are basedd on hidden Markov models.
- First order Markov property : $X_t = f(X_{t-1})$
- They supply the output of the last time step as an additional input thus providing higher memory

RNN

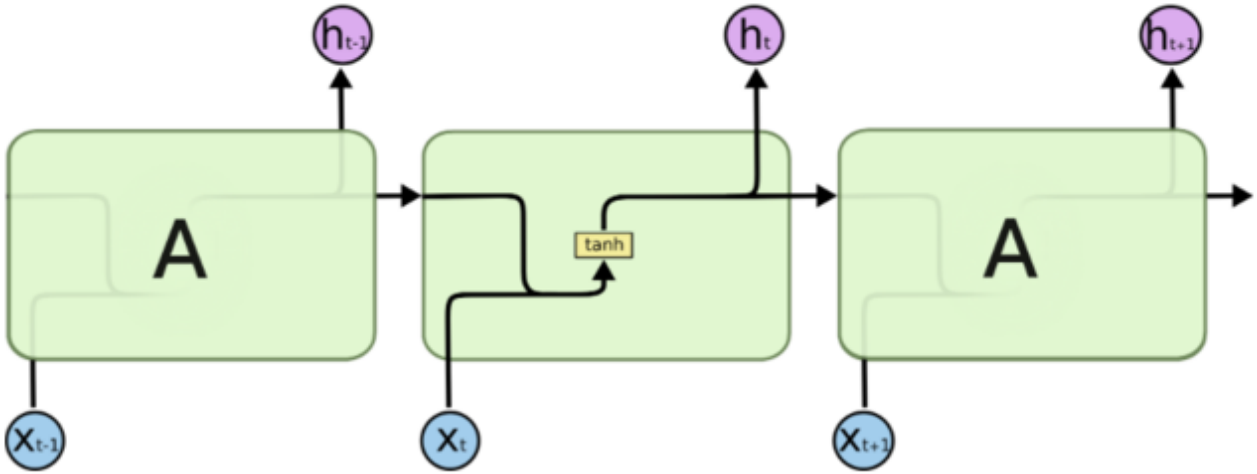


Need

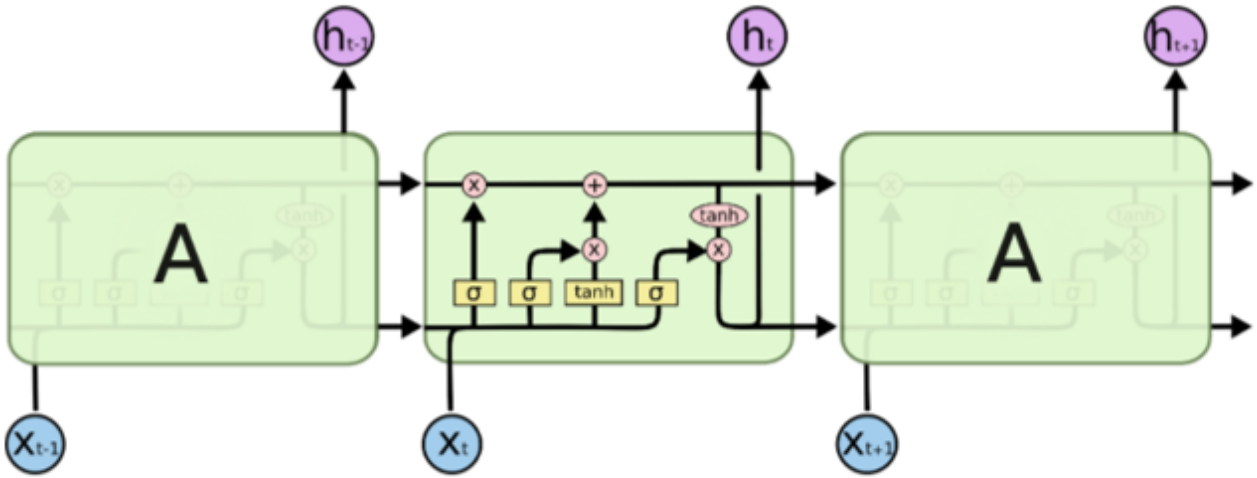
- Needed when older data is required to predict the next step
- Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems.
- *"The clouds are in the sky"*
- Does not require a long range memory to predict the word "sky"
- "I grew up in France ..... I speak fluent French."
- It's easy to suggest that the last word is the name of a language, but which language requires context from earlier on.
- Other example: Predicting the temperature of day 48 hours ahead
- LSTMs solve such long range dependencies

Structure

- The feature differentiating LSTMs from other RNNs is the gates

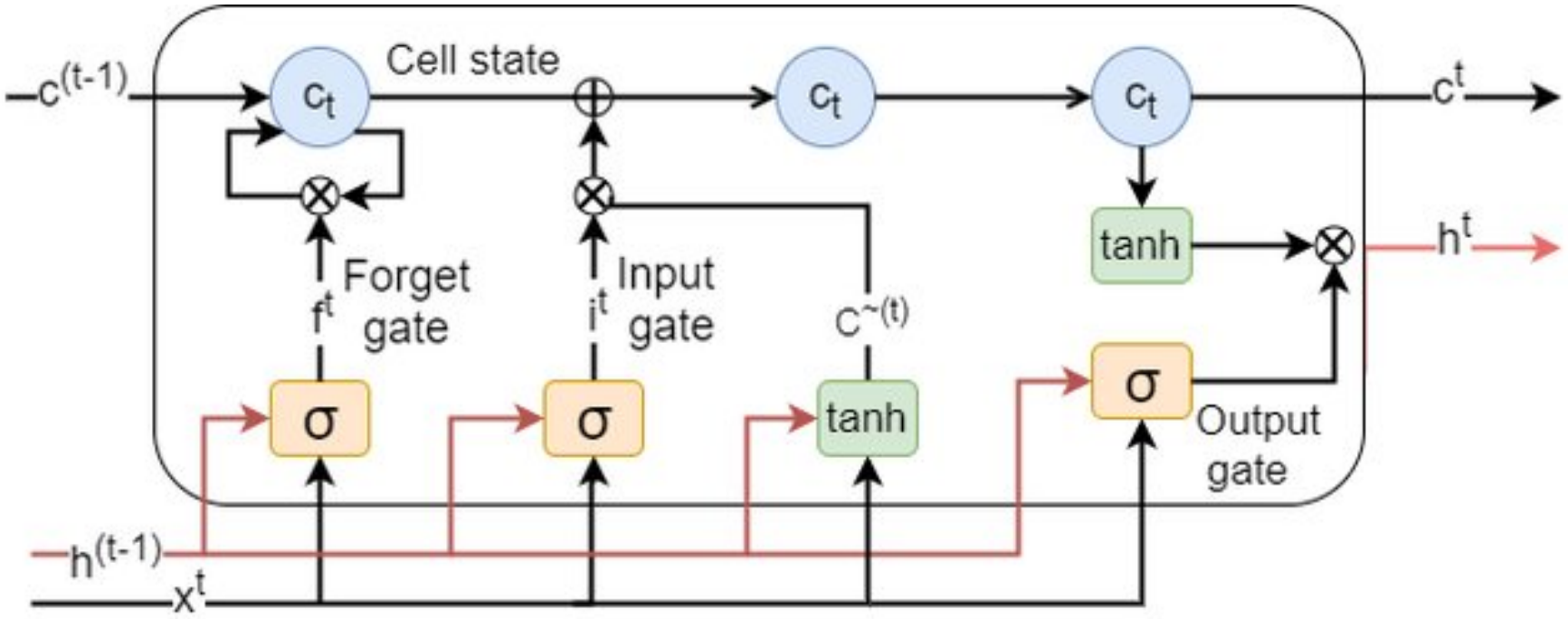


The repeating module in a standard RNN contains a single layer.



The repeating module in an LSTM contains four interacting layers.

LSTM structure



Terminology

- **Hidden State** - Working memory that carries information from immediately previous events and overwrites at every step uncontrollably -present at RNNs and LSTM
- **Cell state**- Long term memory capability that stores and loads information of not necessarily immediately previous events present in LSTMs

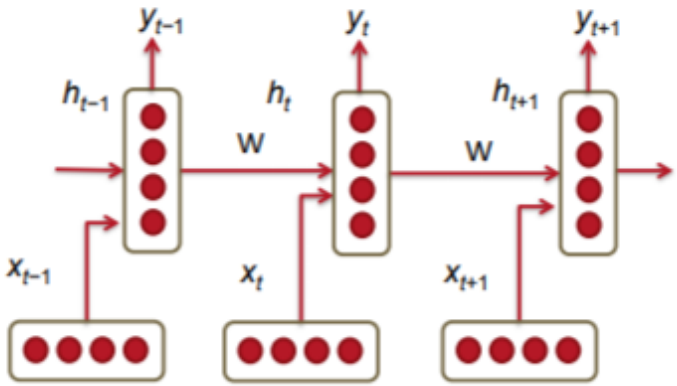
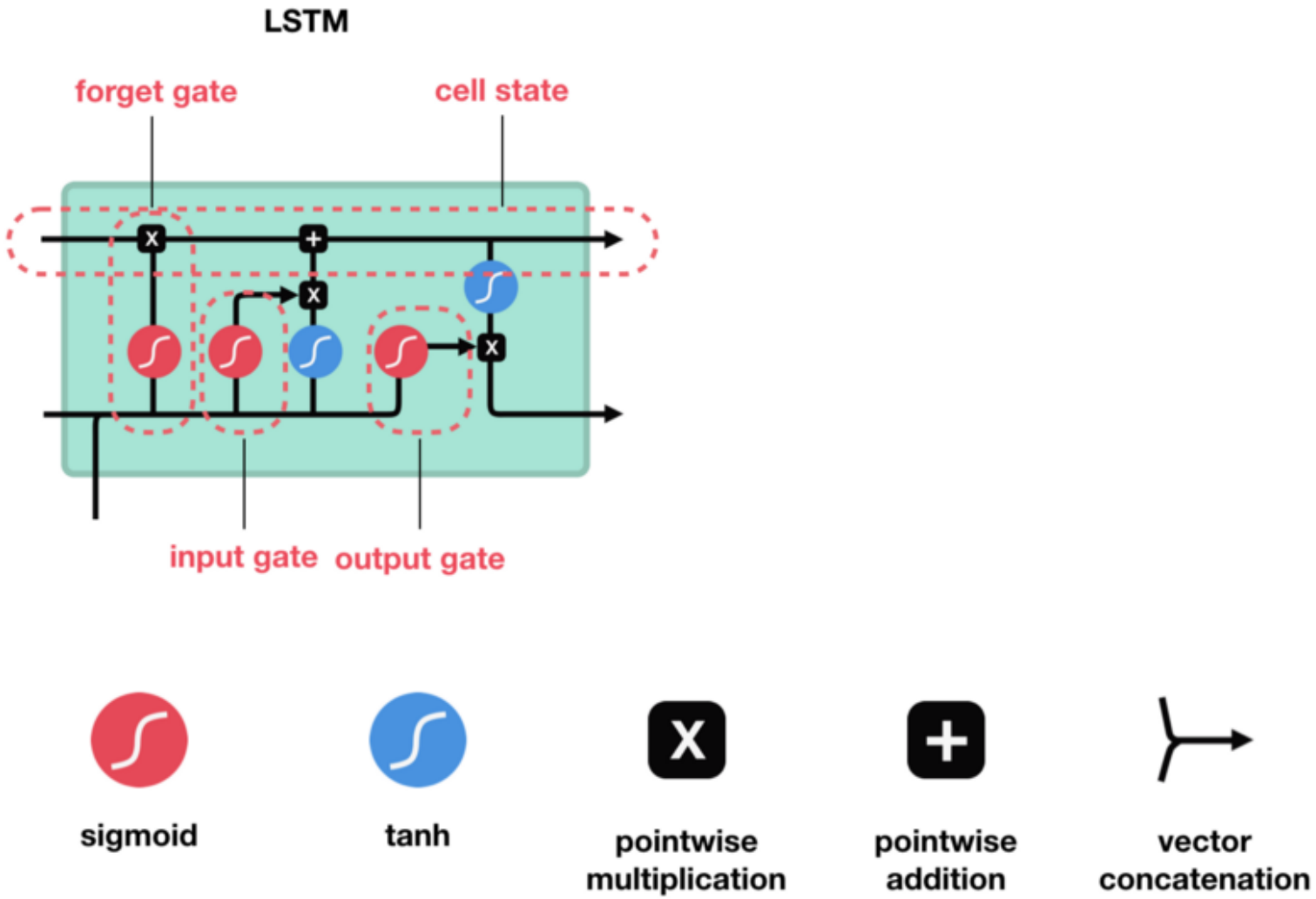


Figure 2: A Recurrent Neural Network (RNN). Three time-steps are shown.

Structure

- A single LSTM cell has 4 different components. Forget gate, input gate, output gate and the cell state.

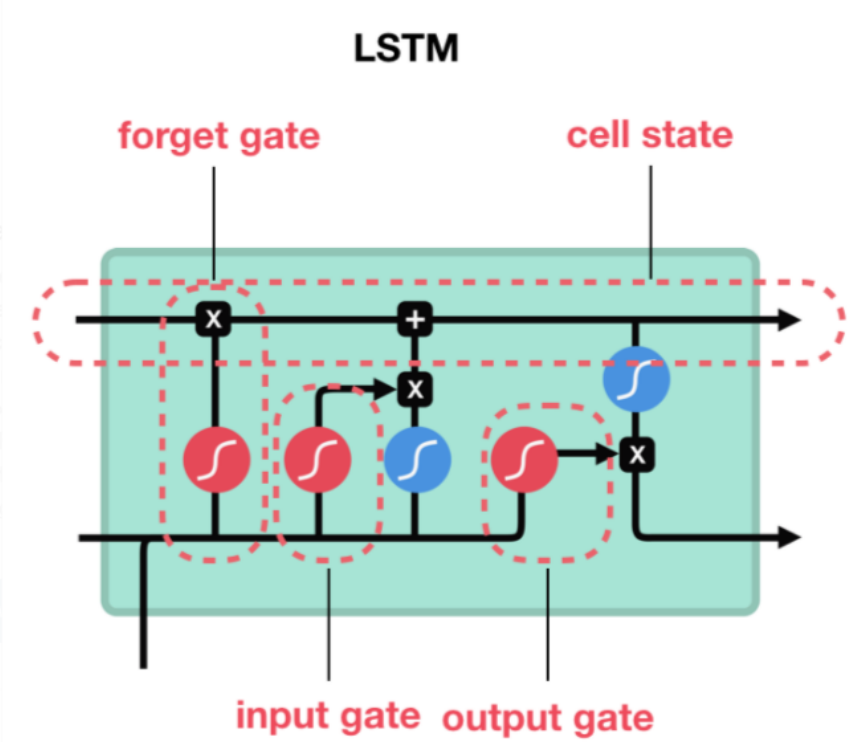


Structure

- A Gate is a sigmoid or tanh activation function. They provide values between 0 to 1 or -1 to 1, providing the capability to filter important information
- The gates control what is added to the cell state from the stored information and how much of it is added
- They do this using weights and biases

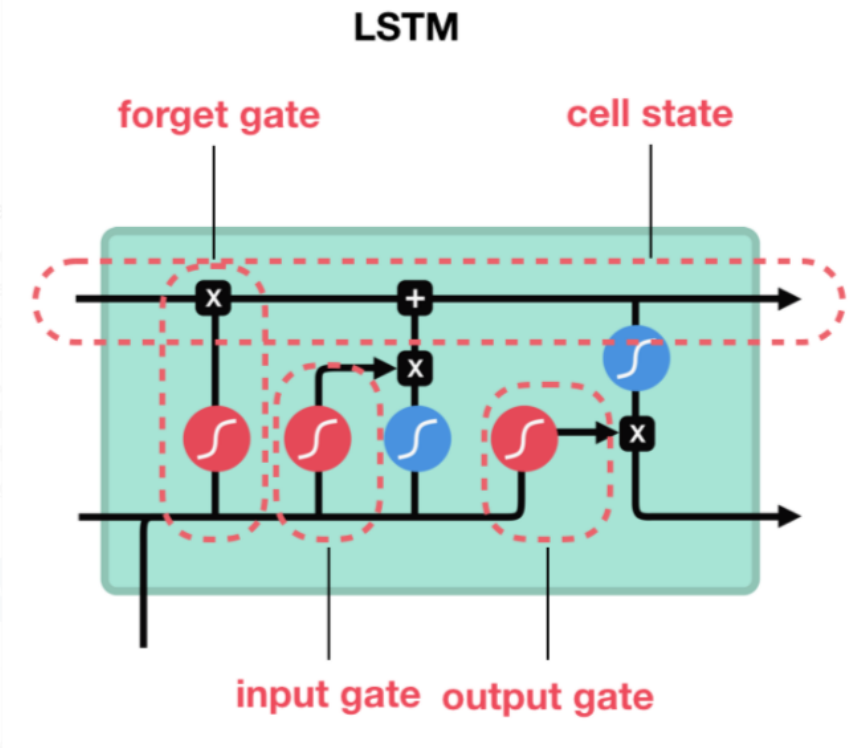
Structure

- Forget Gate- the first gate that interacts with the flow of information
- It decides what information to keep and what to forget
- Naturally, it is multiplied to the current cell state



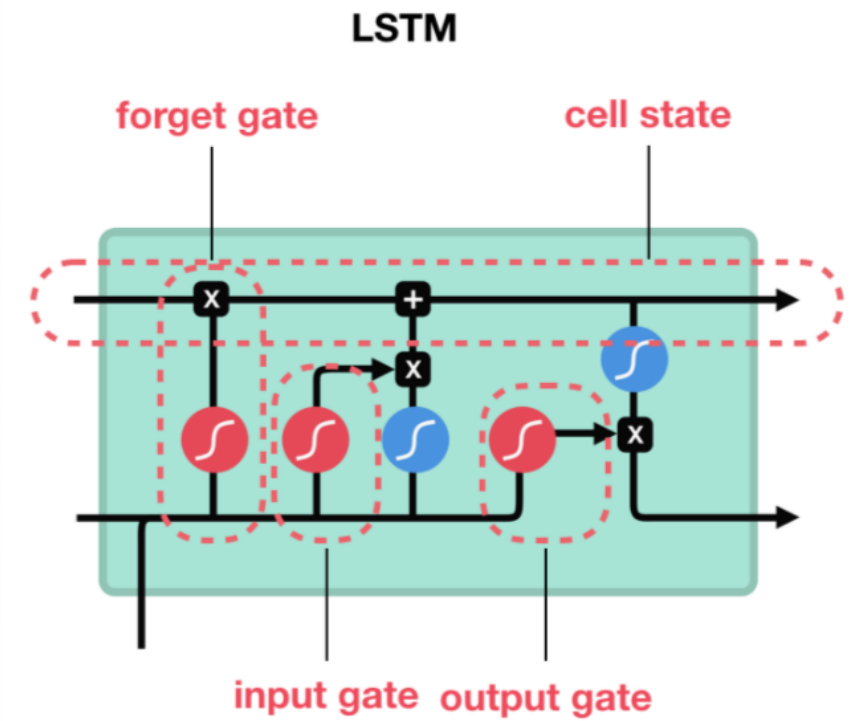
Structure

- Input Gate - the gate that adds the hidden information to the cell state
- Here also, the flow is regulated by sigmoid activation, however the information is also squashed through tanh and then added to cell state

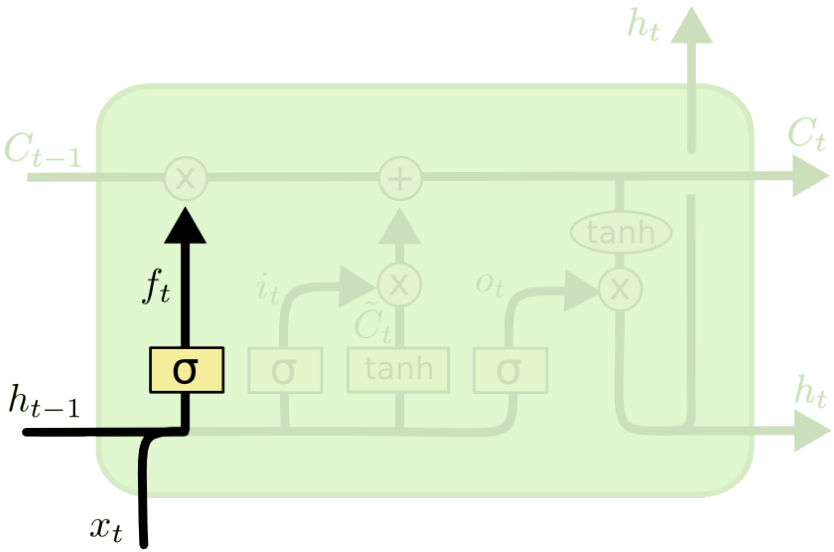


Structure

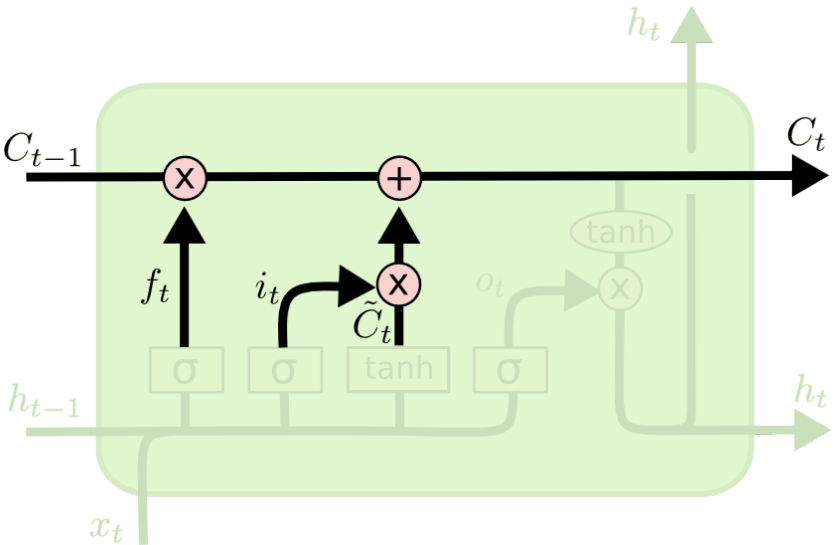
- Cell State runs straight down the entire chain, with only some minor linear interactions. It's very easy for information to just flow along it unchanged
- The cell state serves as the LSTM's long term memory, making it better than traditional RNNs



LSTM operation



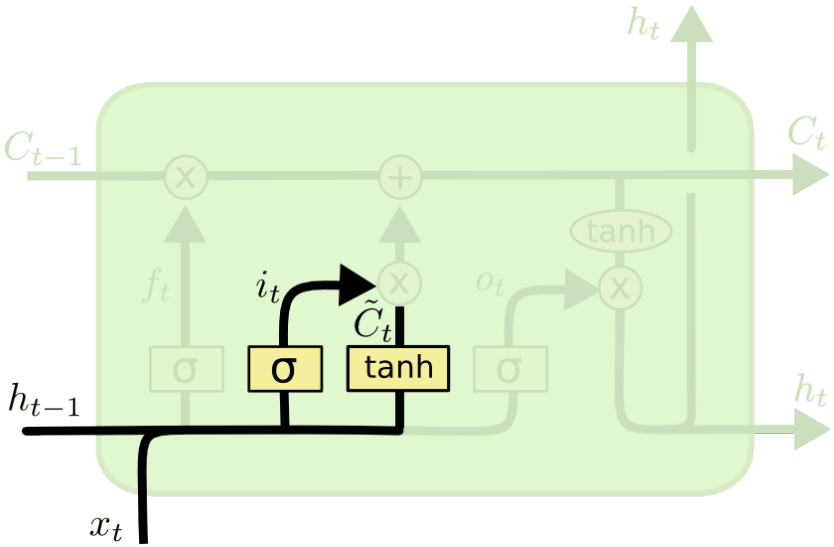
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

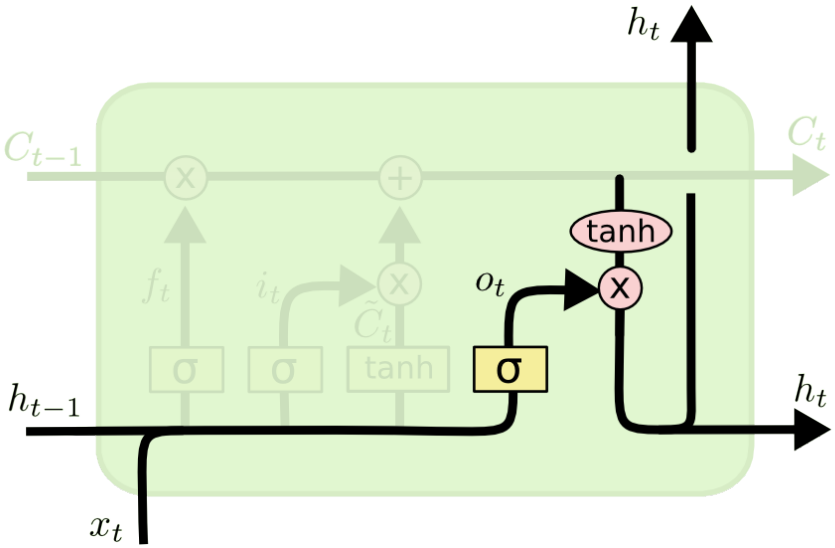
$* \implies$ Hadamard product

LSTM operation



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$



$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

$*$ $\Longrightarrow$ Hadamard product

Problems with LSTMs

- LSTMs may have solved vanishing gradients with their gated structure, but that came at the expense of hardware
- As for long term memory LSTMs can deal with longer sequences than RNNs, but the usecases might actually require even more than they can remember.
- RNN models, were considerably heavier than others for running on low key everyday use hardwares, like mobile phones and voice assistants.
- With the rise of Attention and encoder decoder like networks, we were able to get better and faster results, which is now preferred over of LSTMs

Overview of Presentation

Discriminative v/s Generative

- Discriminative Model
- Generative Model

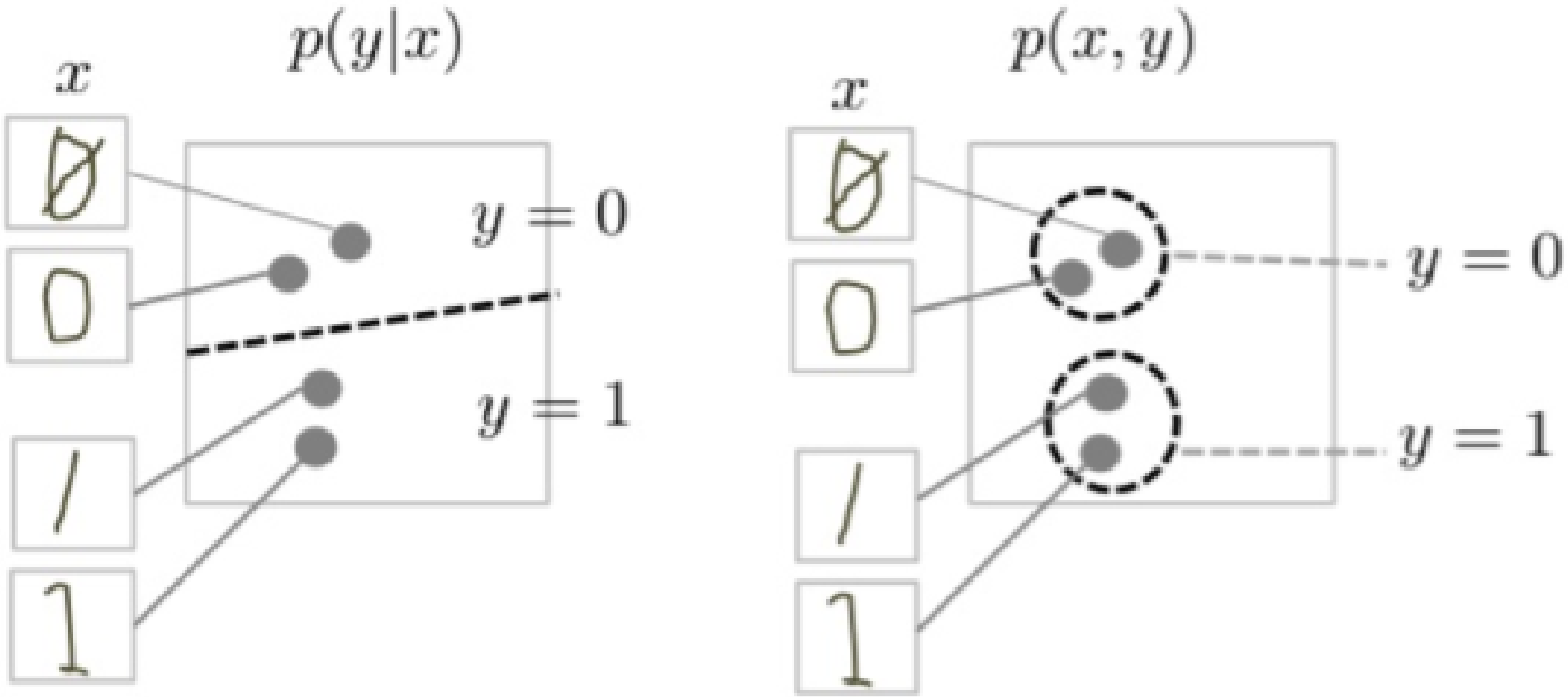
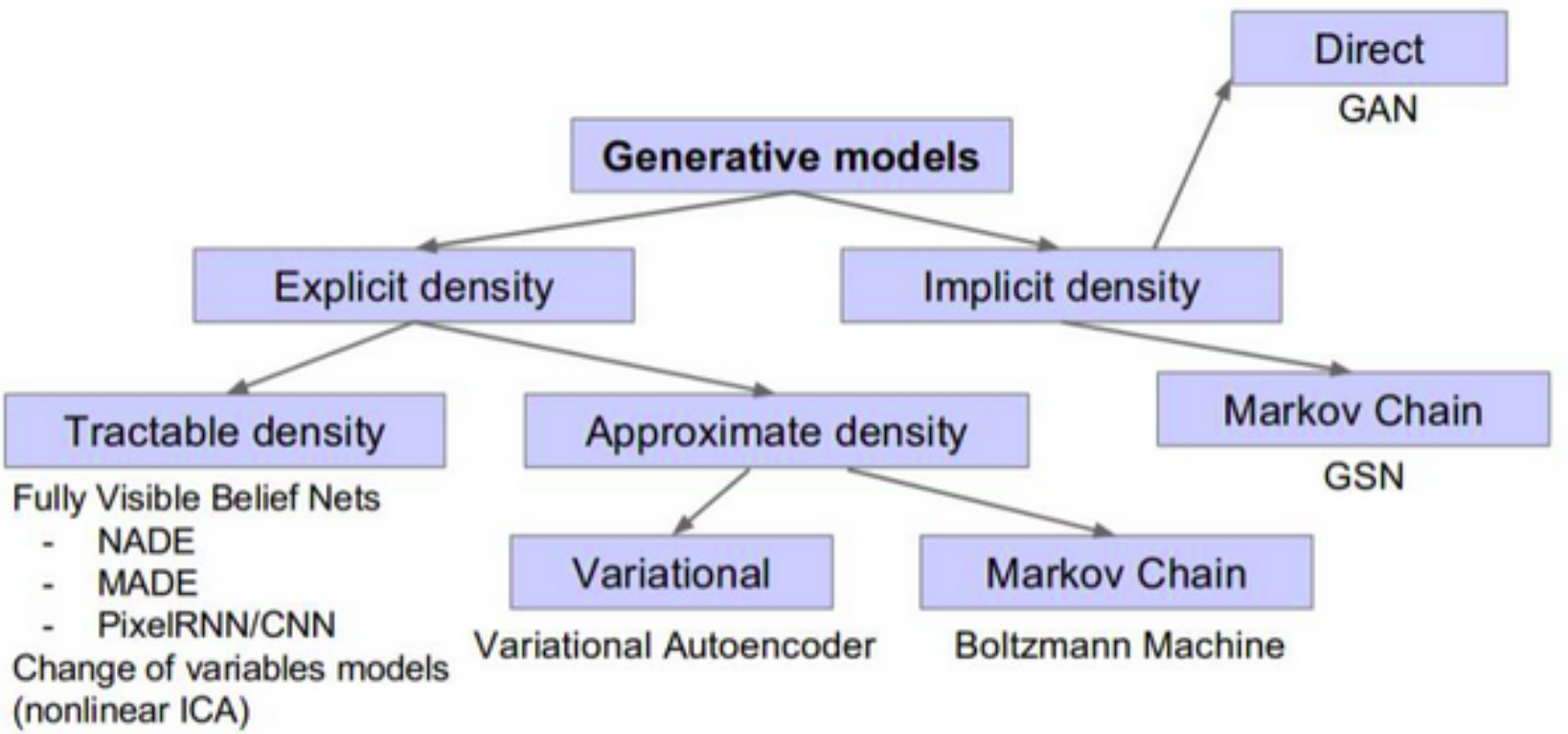


Figure 1: Discriminative and generative models of handwritten digits.

Applications

- Generating Data
- Image to Image Translation
- Face Aging and De-Aging
- 3D object generation

Types of Generative Models



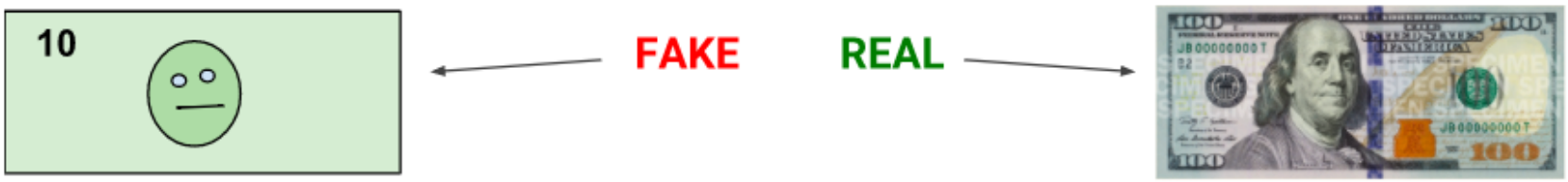
GANs

A generative adversarial network (GAN) has two parts:

- The generator learns to generate plausible data. The generated instances become negative training examples for the discriminator.
- The discriminator learns to distinguish the generator's fake data from real data. The discriminator penalizes the generator for producing implausible results.

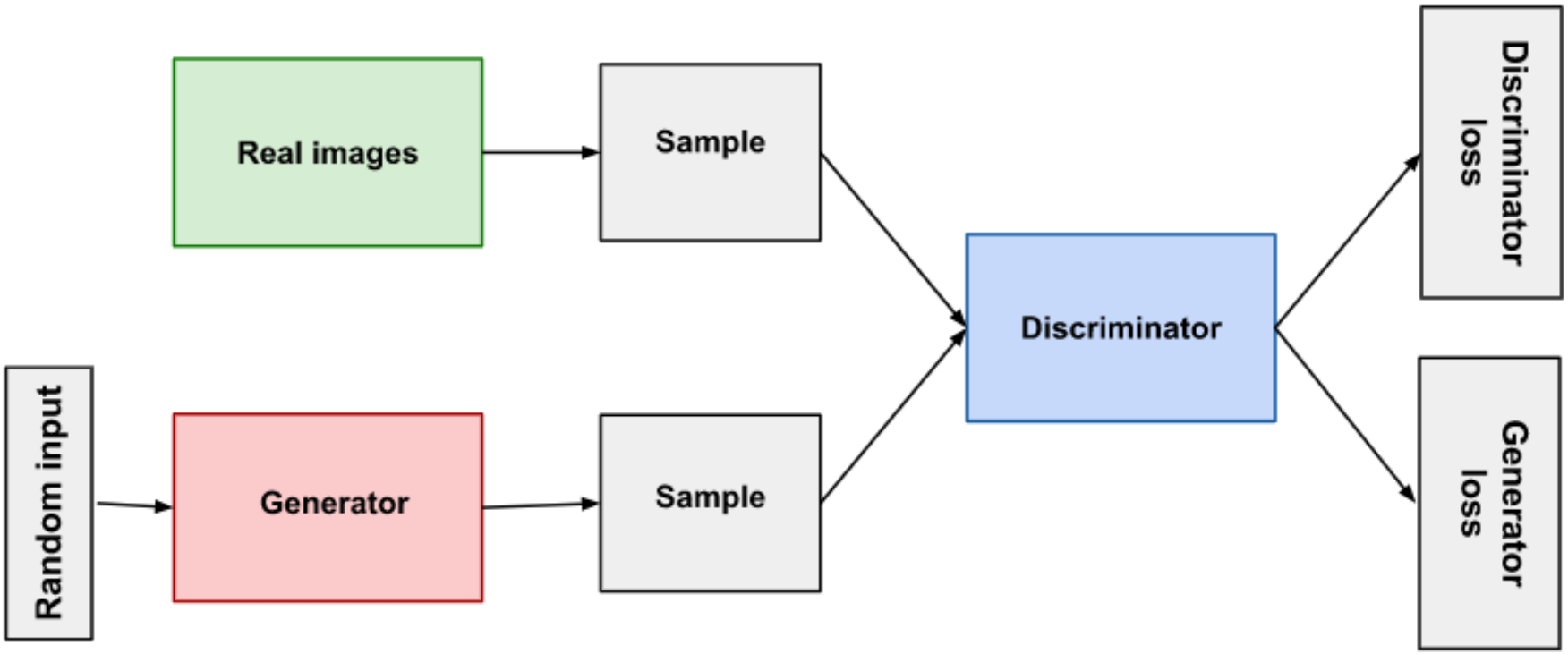
GANs

As training progresses, the generator gets closer to producing output that can fool the discriminator:

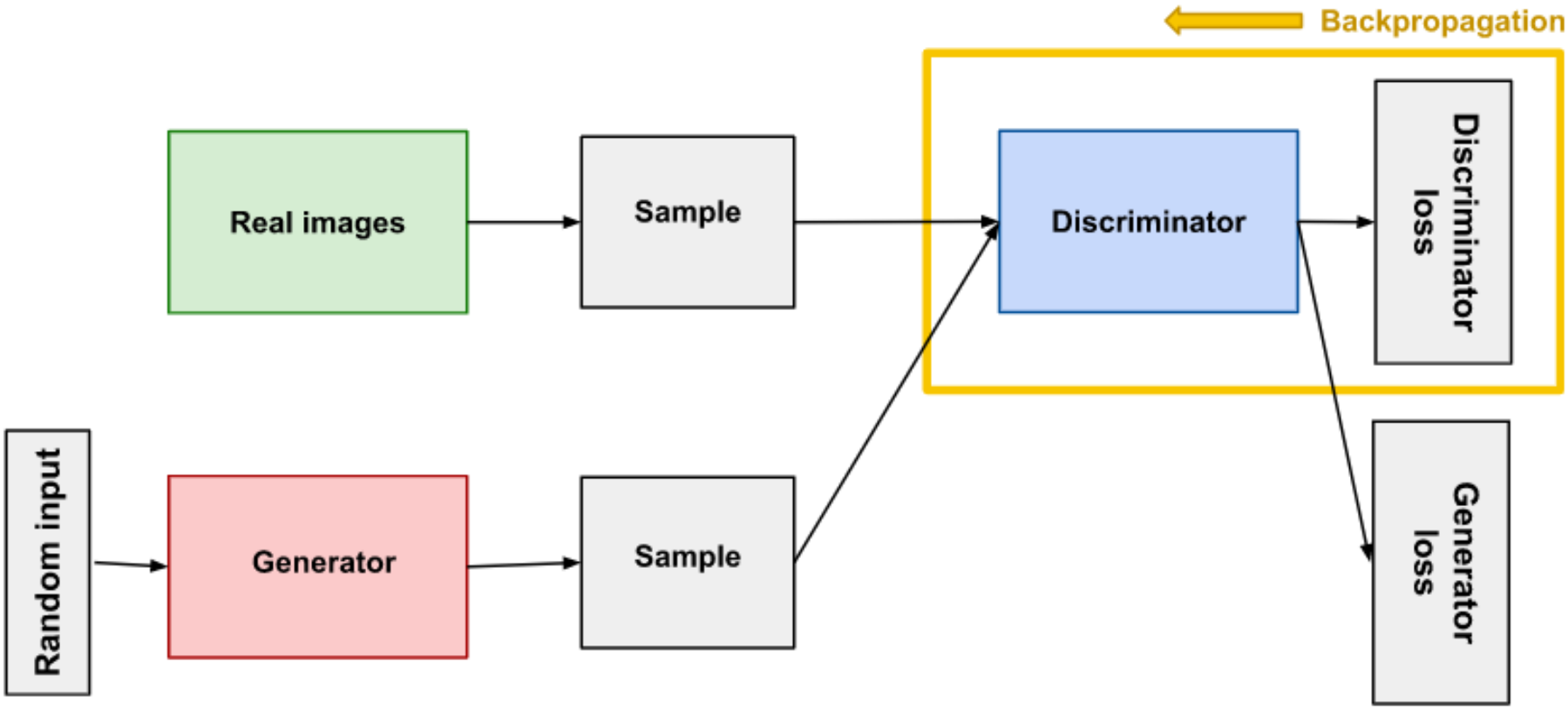


Finally, if generator training goes well, the discriminator gets worse at telling the difference between real and fake. It starts to classify fake data as real.

GANs


Real images
Sample
Random input
Generator
Sample
Discriminator
Discriminator loss
Generator loss

Discriminator

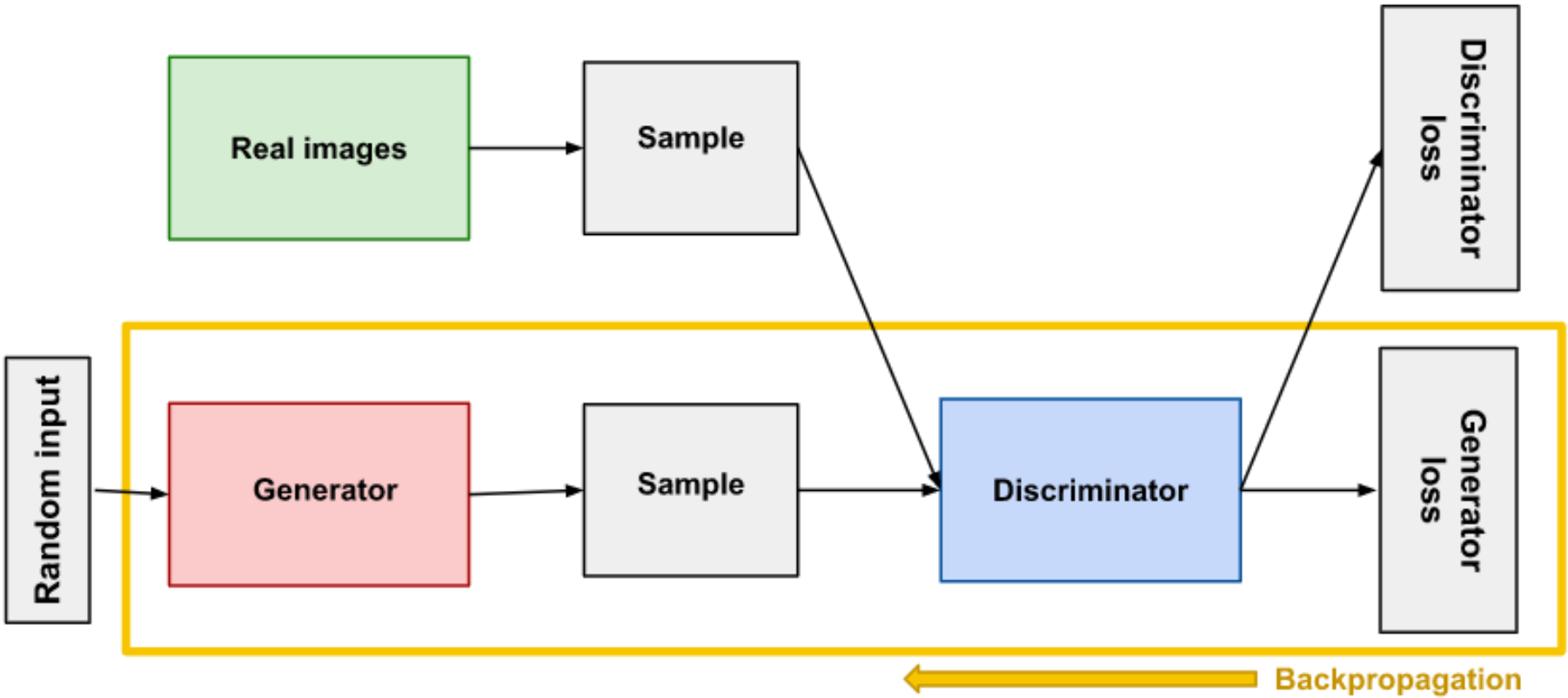


Discriminator

Training Discriminator :

1. The discriminator classifies both real data and fake data from the generator.
2. The discriminator loss penalizes the discriminator for misclassifying a real instance as fake or a fake instance as real.
3. The discriminator updates its weights through backpropagation from the discriminator loss through the discriminator network.

Generator


Real images
Sample
Discriminator loss
Random input
Generator
Sample
Discriminator
Generator loss
Backpropagation

Generator

Training Generator :

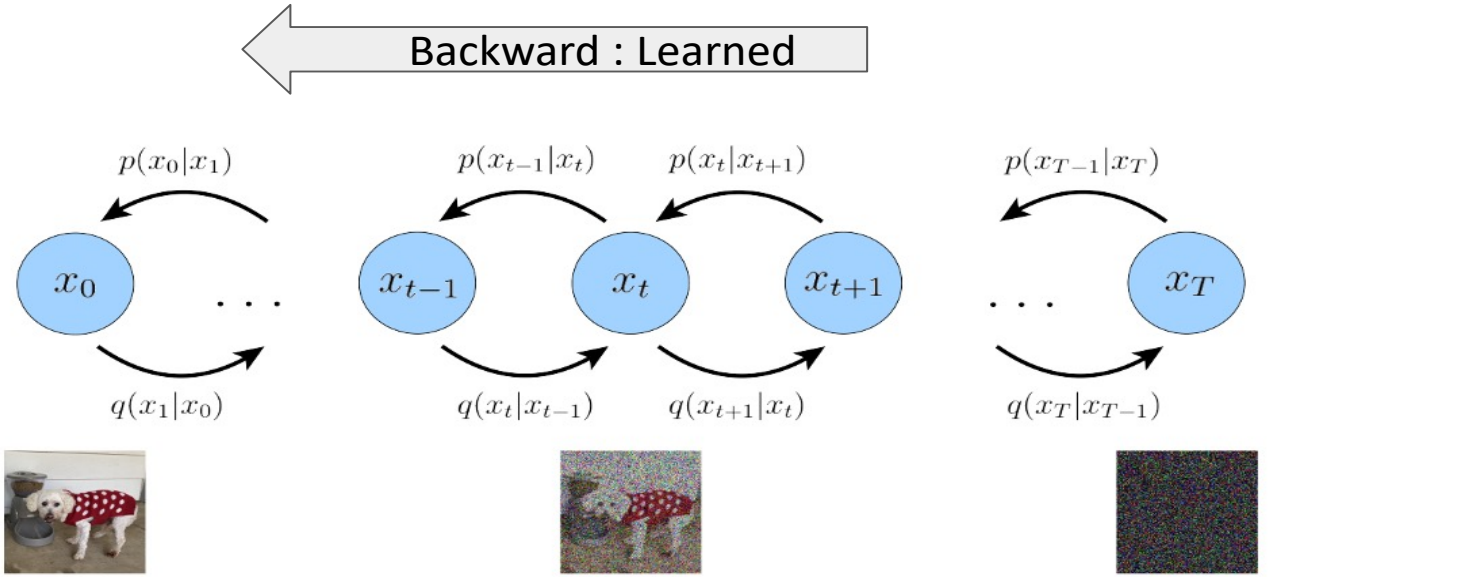
1. Sample random noise.
2. Produce generator output from sampled random noise.
3. Get discriminator "Real" or "Fake" classification for generator output.
4. Calculate loss from discriminator classification.
5. Backpropagate through both the discriminator and generator to obtain gradients.
6. Use gradients to change only the generator weights.

GAN Training

GAN training proceeds in alternating periods:

1. The discriminator trains for one or more epochs.
2. The generator trains for one or more epochs.
3. Repeat steps 1 and 2 to continue to train the generator and discriminator networks.

Diffusion models



$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)\mathbf{I})$$

$$\text{as t} \implies T, \alpha_t \implies 0$$

Image Source: Luo 2022

GAN vs Diffusion

1. **GAN-Based Models (Adversarial Training)**
   - **StyleGAN (NVIDIA):** High-resolution face & artistic image generation.
   - **BigGAN (Google):** Large-scale, detailed images.
   - **CycleGAN:** Unpaired image-to-image translation (e.g., horse zebra).
   - **Pix2Pix:** Paired image translation (e.g., sketch real image).
2. **Diffusion-Based Models (Denoising Process)**
   - **Stable Diffusion (Stability AI):** Text-to-image AI art.
   - **DALL · E 2 (OpenAI):** Advanced text-image synthesis.
   - **Imagen (Google):** Photorealistic AI-generated images.
   - **DeepFloyd IF:** High-quality AI-generated visuals.

GAN vs Diffusion

3. Comparison:

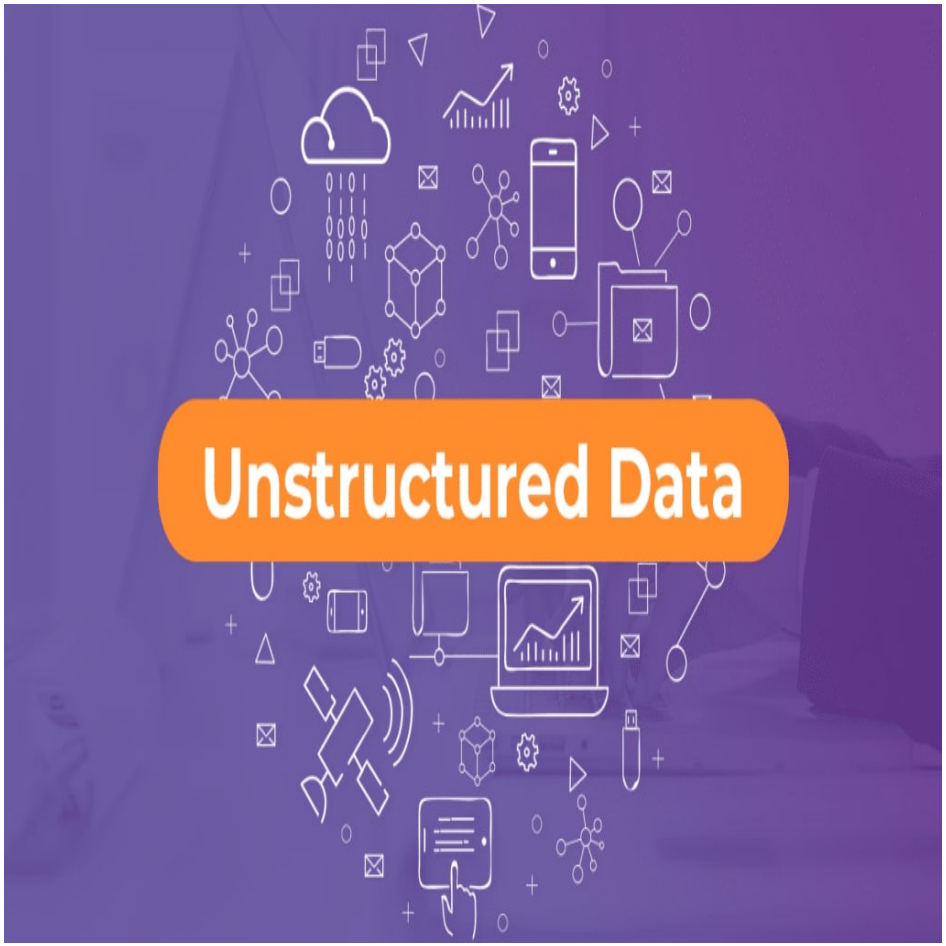
Feature	**GANs**	**Diffusion**
Stability	Hard to train	More stable
Realism	High (StyleGAN)	Ultra-realistic (Imagen)
Compute Cost	Lower	Higher
Use Cases	Deepfakes, Faces	AI Art, Text-to-Image

Overview of Presentation

Human languages

- Every language have different set of words.
- Combination of words is sentence.
- To make meaningful sentence, grammar is necessary.

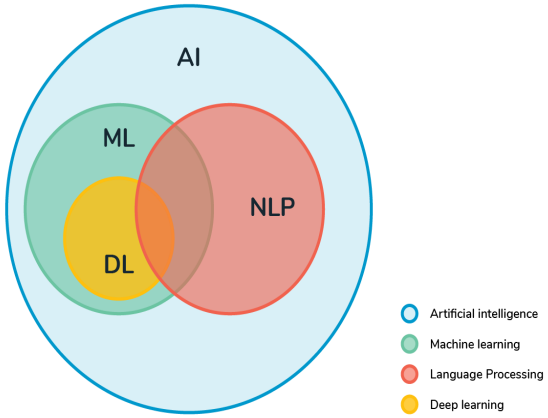
Text mining and NLP

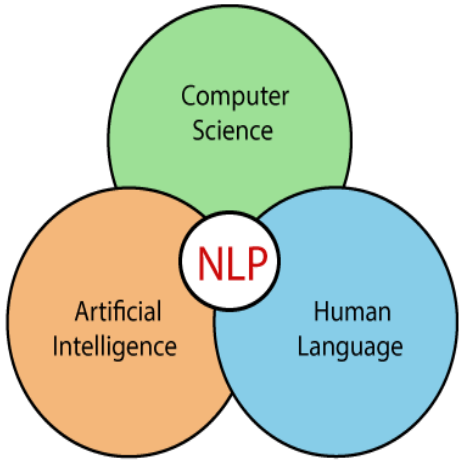


- Text mining refers to the process of deriving high quality information from the texts.
- The main aim is, essentially turn the text into data for analysis, via application of natural language processing.

Natural language processing

- Natural Language Processing or NLP gives the machines the ability to read, understand and derive meaning from human languages.





- It is Also known as Computational Linguistics (CL), Human Language Technology (HLT), Natural Language Engineering (NLE).

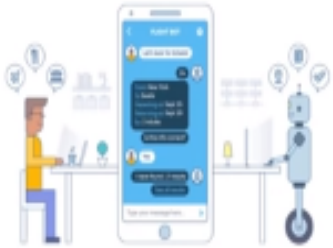
History of NLP

- In 1950, Alan Turing published an article titled "Machine and Intelligence" which advertised what is now called the Turing test as a subfield of intelligence
- Some beneficial and successful Natural language systems were developed in the 1960s were SHRDLU, a natural language system working in restricted "blocks of words" with restricted vocabularies was written between 1964 to 1966

Applications of NLP

Sentimental Analysis

Chatbot

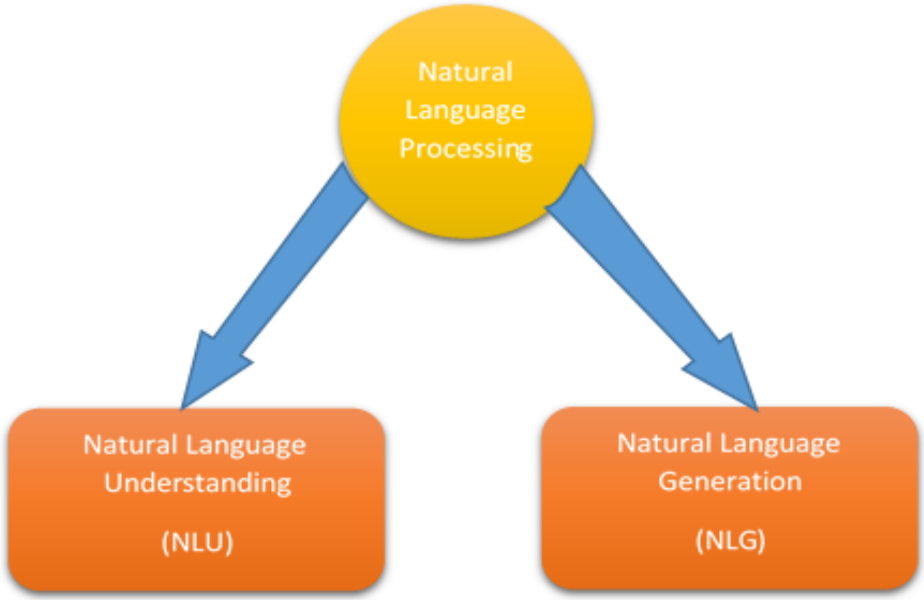


Speech Recognition

Machine Translation

- Spell checking, Keyword search, advertisement matching, Information extraction and many more.

Components of NLP



- Mapping input to meaning
- analyzing different aspects of the language

- Text planning
- Sentence planning
- Text realization

- NL Understanding problem is much harder than NL Generation.

NLU applications

- Text Categorization & Classification: Few applications of text categorization include Spam filtration in emails, script compliance, etc.
- Automatic Summarization: NLU creates compact, fluent summaries from long text documents
- Voicebot: NLU enables a voicebot to understand the intent behind the customer's speech and extract important entities from that.
- Question Answering & Semantic Parsing: Question Answering (QA) systems enable machines to answer the questions asked automatically in natural human language.
- Sentiment Analysis/Emotion Mining: measure the sentiment behind an opinion or context, which can be really helpful in driving purchase decisions.

Natural Language Generation (NLG)

- Producing output in the natural language from some internal interpretation.
- NLG can be viewed as the reverse process of NL understanding
- Different level of synthesis required:
  - Lexical Selection (word selection)
  - syntactic generation (make sense of combination of words)
  - Discourse planner (make posterior based on prior)

NLG applications

- generating the responses of chatbots and voice assistants such as Google's Alexa and Apple's Siri
- automating lead nurturing email, messaging and chat responses;
- personalizing responses to customer emails and messages;
- generating and personalizing scripts used by customer service representatives;
- creating product descriptions for e-commerce webpages and customer messaging.

Tools of NLP

- Basic name of tools:
  - Tokenization, Stemming, Lemmatization, Part Of Speech (POS), Syntax Tree etc.

Sentiment Analysis: Task & Approaches

Definition: Sentiment Analysis (Opinion Mining) is an NLP task that classifies text into sentiment labels (Positive, Negative, Neutral).

Task Types:

- **Binary Classification:** Positive vs. Negative
- **Multi-class Sentiment:** Very Positive, Neutral, Very Negative
- **Regression-based Sentiment Scoring**

Approaches:

1. **Lexicon-Based:** Uses predefined word lists (e.g., SentiWordNet, VADER).
2. **Traditional ML:** Naïve Bayes, SVM, Logistic Regression with feature extraction (TF-IDF, BoW, Word2Vec).
3. **Deep Learning:** LSTMs, CNNs, and Transformer-based models (BERT, XLNet).

Transformer-Based Models & Challenges

State-of-the-Art Models:

- **BERT (Bidirectional Encoder Representations):** Captures bidirectional context.
- **RoBERTa, ALBERT, DistilBERT:** Optimized versions of BERT.
- **XLNet:** Overcomes BERT's masked token limitation.
- **GPT-based Models:** Generative sentiment analysis.

Challenges in Sentiment Analysis:

- **Sarcasm Detection:** "Great! Another Monday morning..."
- **Domain Adaptation:** Finance sentiment differs from movie reviews.
- **Ambiguity in Expressions:** "The product is sick" (positive or negative?).
- **Multilingual Sentiment Analysis:** Varies across languages.

Tokenization

- Tokenization is the first step in NLP.
  - Break a sentence into words.
  - Understand the importance of each of words with respect to the sentence.
  - Produce a structural description on an input sentence.
- It have Unigram, Bigram, Trigram, n-gram.
- *For example*, "statistics" is a unigram (n = 1), "machine learning" is a bigram (n = 2), "natural language processing" is a trigram (n = 3) and so on.

Stemming

- Normalize the words into its base form or root form.

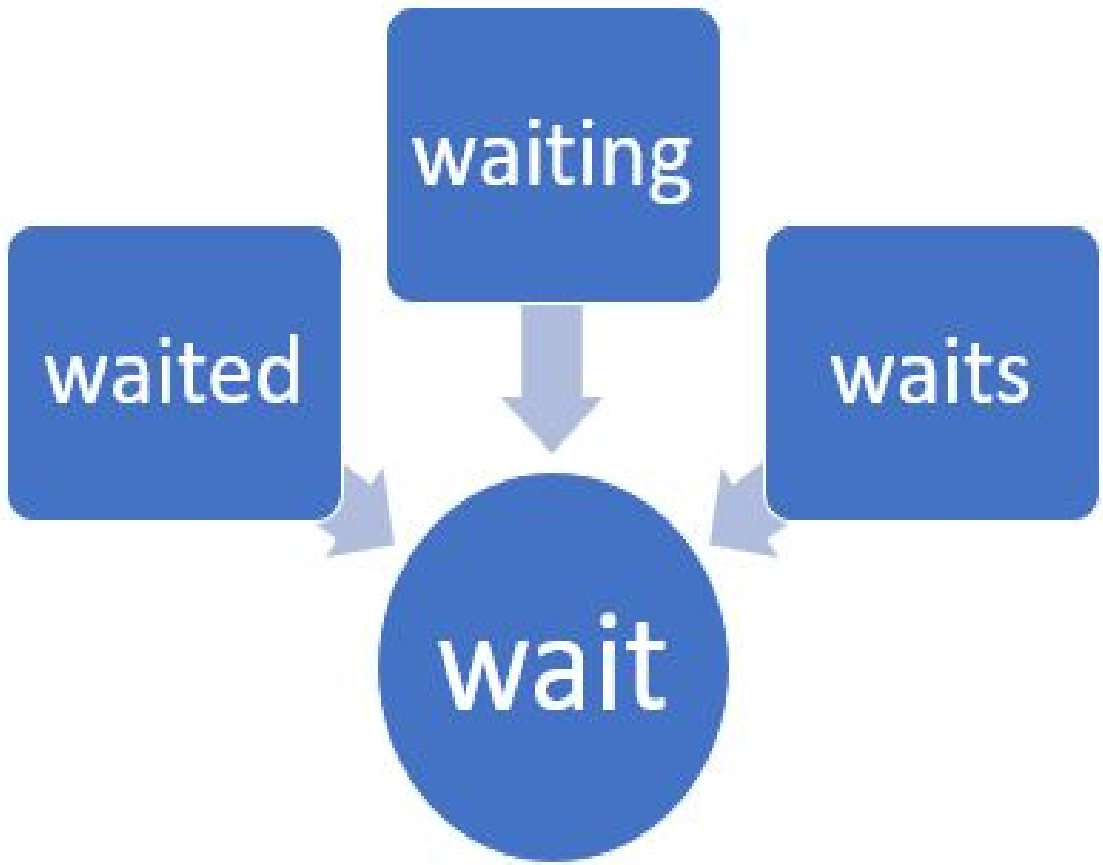


Figure: stemming

Lemmatization

- Lemmatization is the process wherein the context is used to convert a word to its meaningful base or root form.
- Somehow similar to stemming, as it maps several words into one common root.
- For example, a Lemmatizer should map gone, going and went into go.
- The stemmed words may result in invalid words but lemmatized words always result in meaningful words.
- For example, the word “computer” was stemmed to the word “comput”.

Stop words

- Stopwords are the words in any language which does not add much meaning to a sentence.
- They can safely be ignored without sacrificing the meaning of the sentence.

Figure: Stop words

Part of speech (POS)

- Each language is made up of a number of parts of speech such as verbs, nouns, adverbs, adjectives and so on.
- PoS is all about tagging specific parts of a speech on a text.

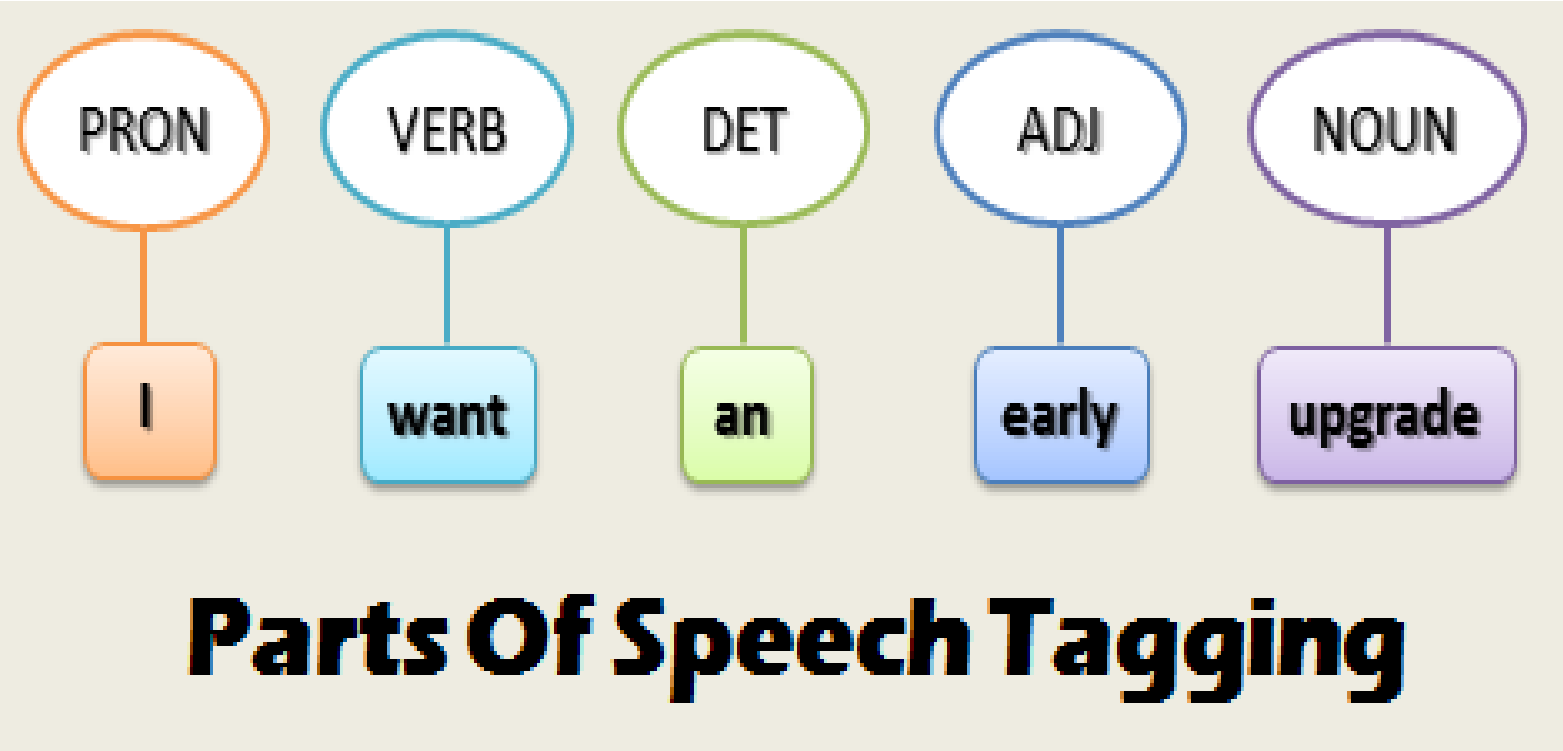


Figure: Part of speech words

Part of speech (POS)

- Each language is made up of a number of parts of speech such as verbs, nouns, adverbs, adjectives and so on.
- PoS is all about tagging specific parts of a speech on a text.

1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	*to*
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb

Figure: Part of speech words

NER: Named Entity Recognition

- Named Entity Recognition (NER) is a standard NLP problem which involves spotting named entities (people, places, organizations etc.) from a chunk of text, and classifying them into a predefined set of categories.

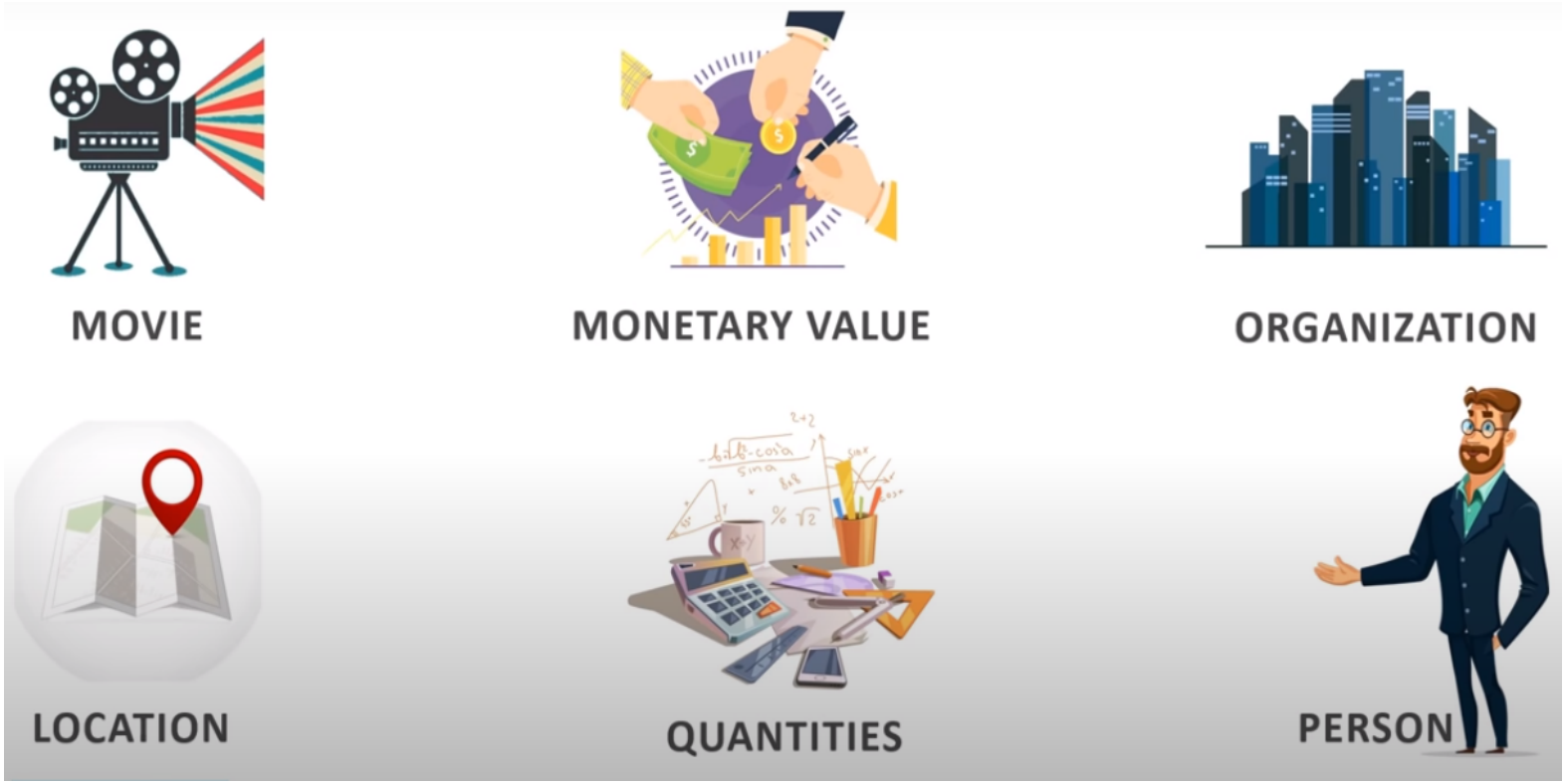


Figure: Named Entity Recognition

Chunk

- Picking up individual pieces of information and grouping them into bigger pieces.

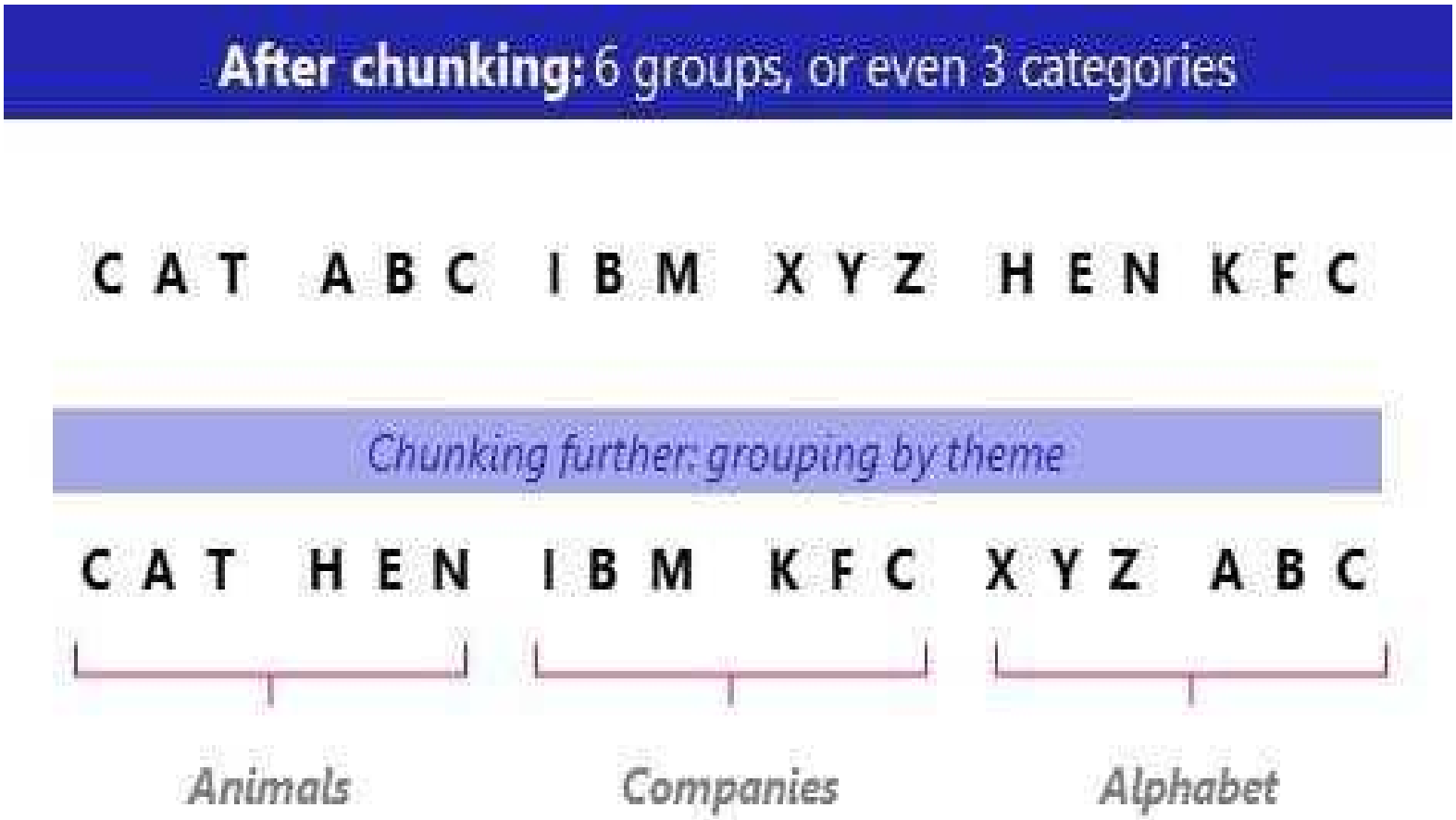
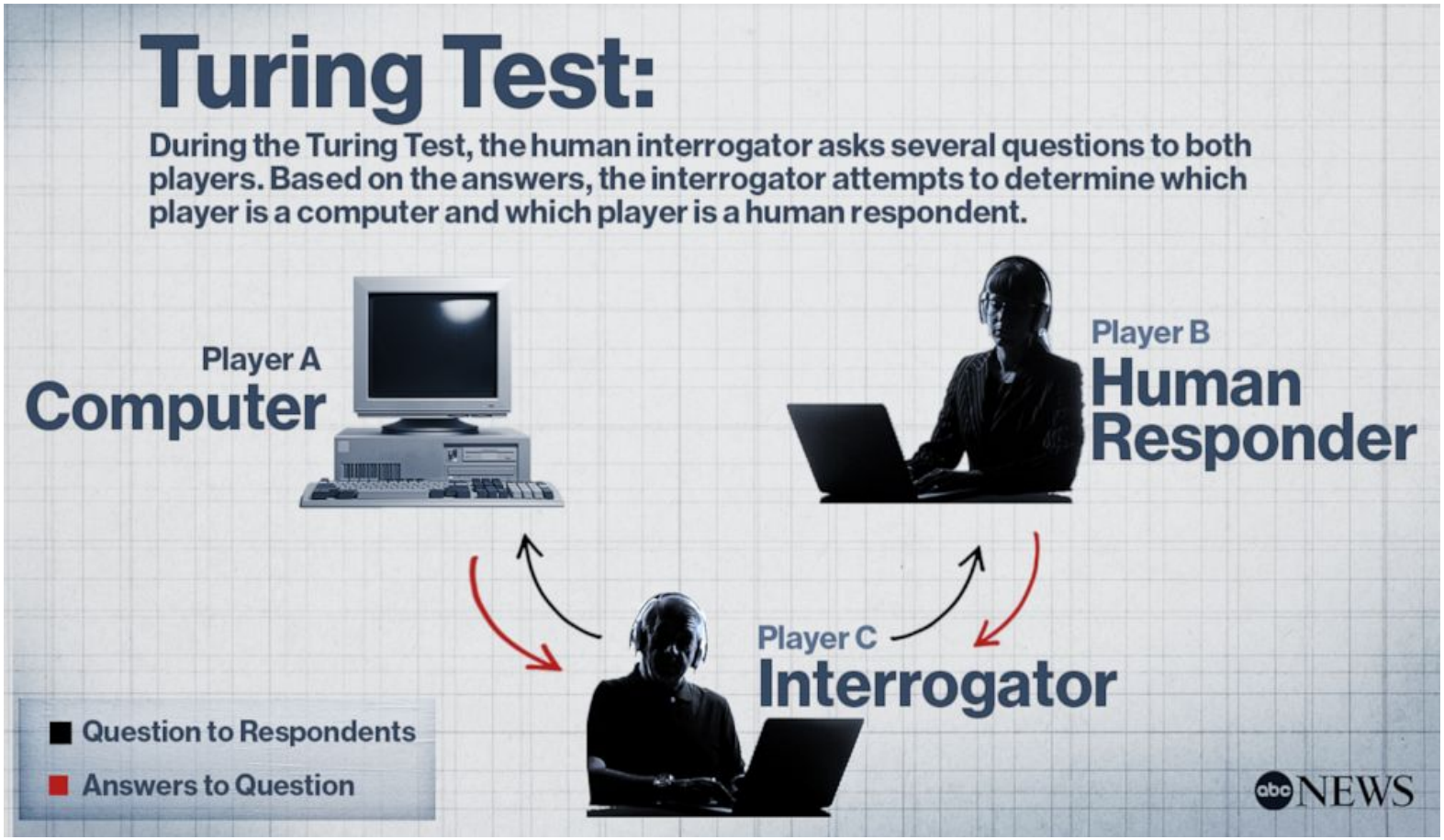
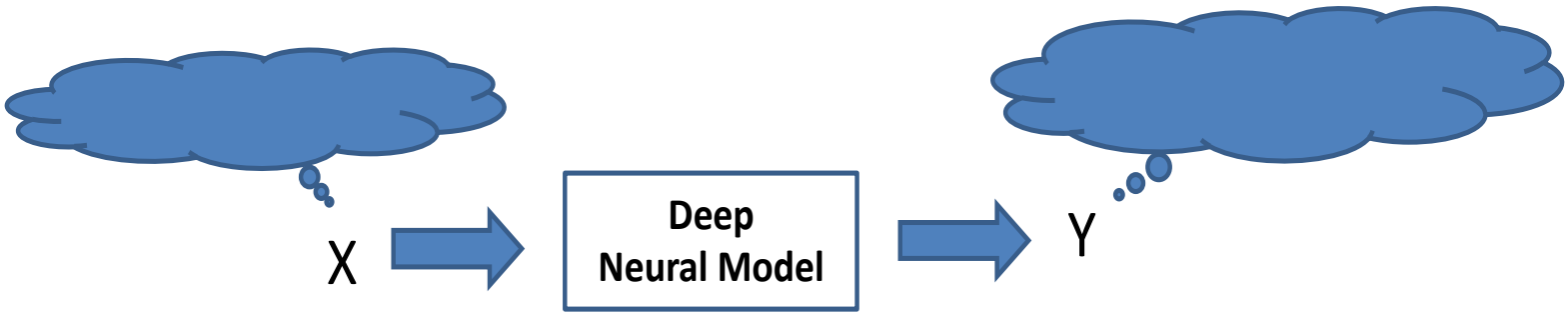


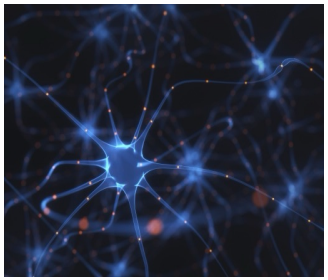
Figure: chunk functioning

Turing Test


Turing Test:
During the Turing Test, the human interrogator asks several questions to both players. Based on the answers, the interrogator attempts to determine which player is a computer and which player is a human respondent.
Player A
Computer
Player B
Human Responder
Player C
Interrogator
Question to Respondents
Answers to Question
abc NEWS

NN - Prediction





Inspired by Computations in Brain.
Neurons organized layer by layer

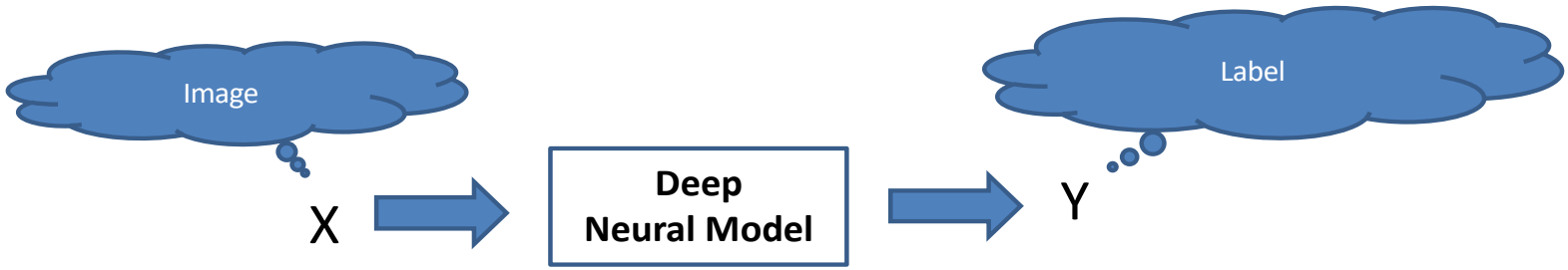
Seen as a function approximator.

$(f: X \rightarrow Y)$

Learn the model from Data: $\{X_i, Y_i\}_{i=1}^{m}$

X and Y can be fairly complex.

NN - Classification



Label Prediction

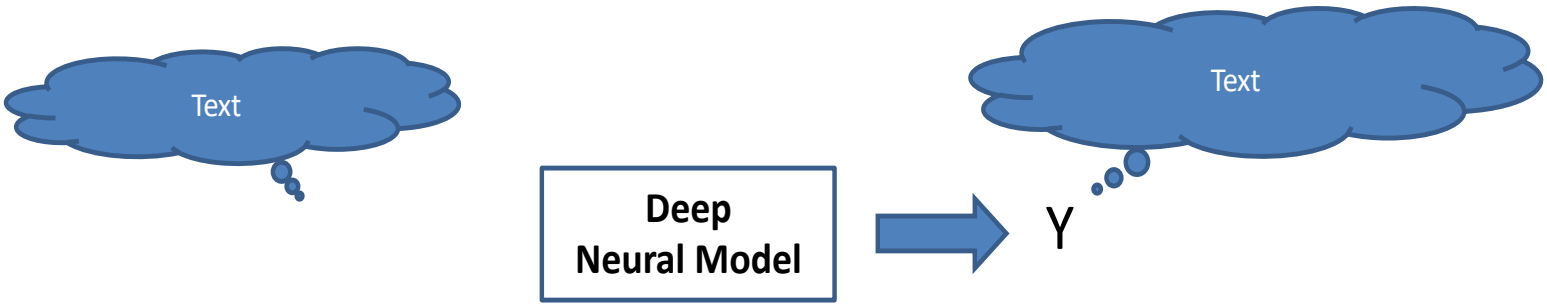
Typically one or few correct answers.

f

Leopard (label)

Source: ImageNet

NN - Generation



Text Generation

Practically infinite number of possible correct answers.

Complete the following sentence:
To be proud of one's country is

f

To be proud of one's country is to appreciate its heritage, celebrate its achievements, and strive for its continuous progress, while also acknowledging its imperfections and actively working towards a more inclusive and equitable society for all.

Source: ChatGPT

NN - Text Generation

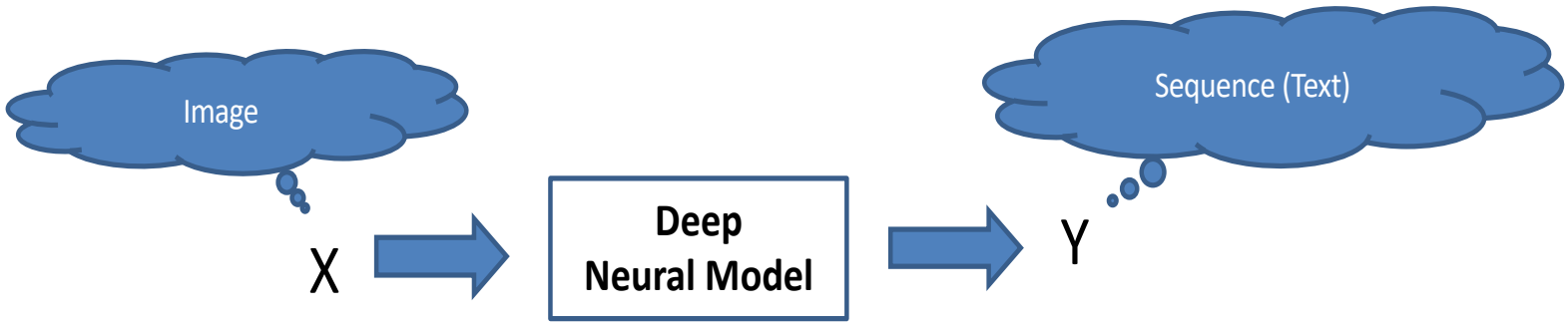
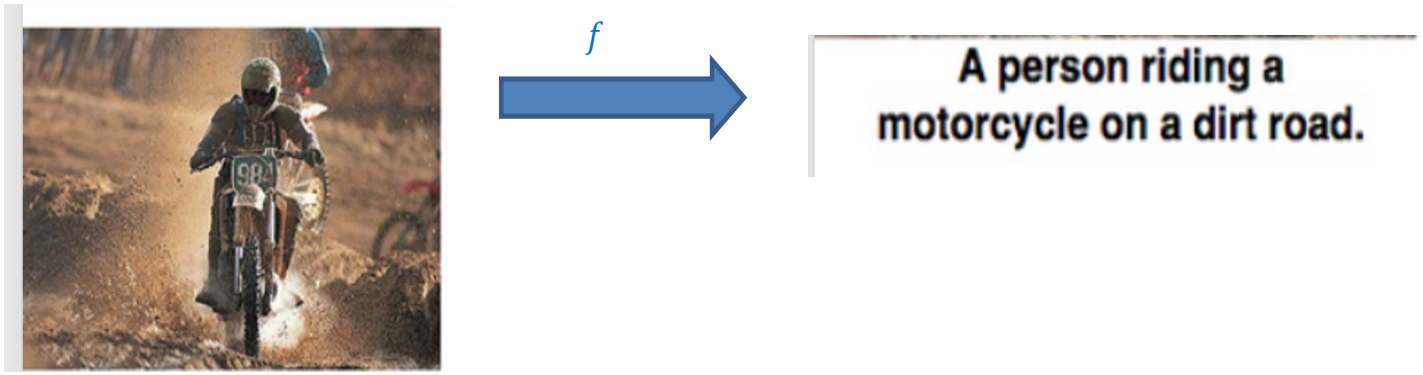
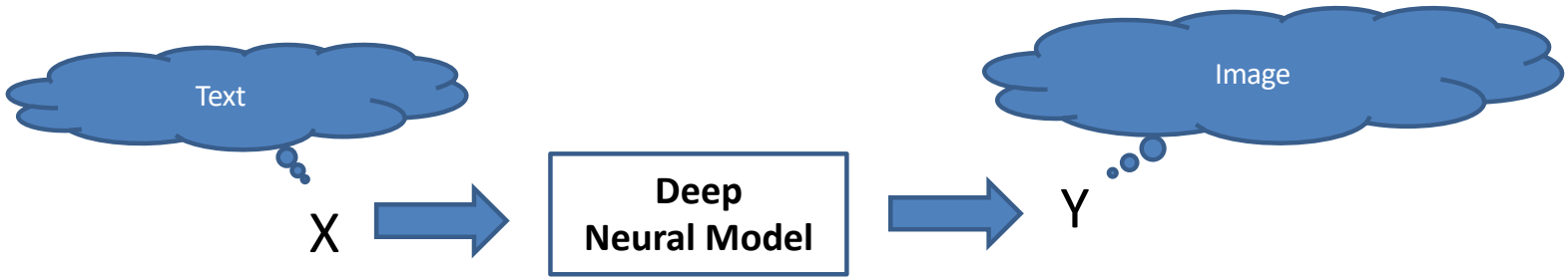


Image conditioned Text Generation



NN - Image Generation



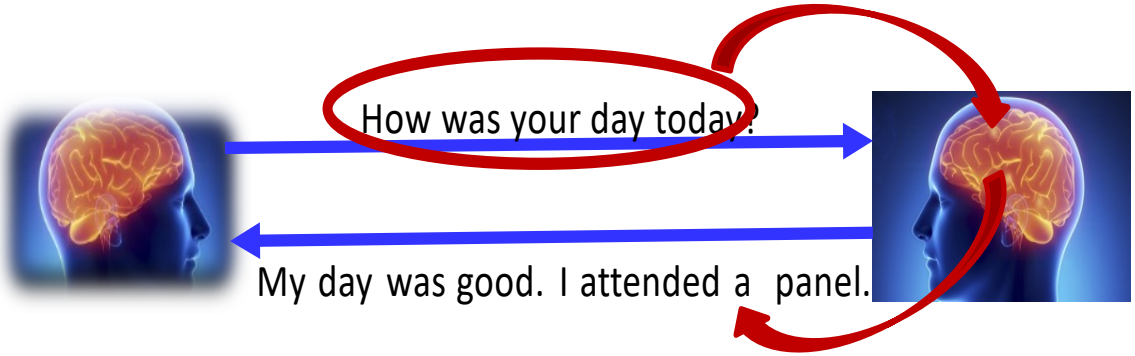
Text Conditioned Image Generation

A brain riding a rocketship heading towards the moon.

f

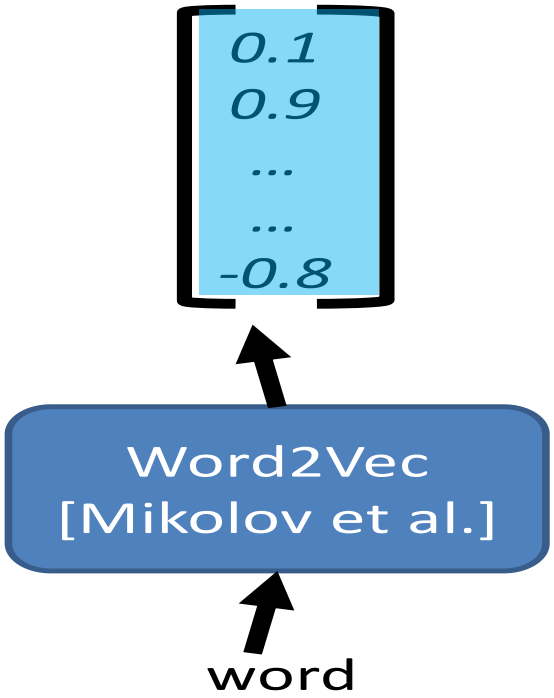
Source: Google Imagen

NN - Structure



- Encoding: Input words are processed by the brain
  - Words are discrete; brain processing uses signals (continuous)
- Reasoning: Brain performs internal reasoning to decide a response
- Decoding: Brain verbalizes the response one word at a time

NN - Word2Vec

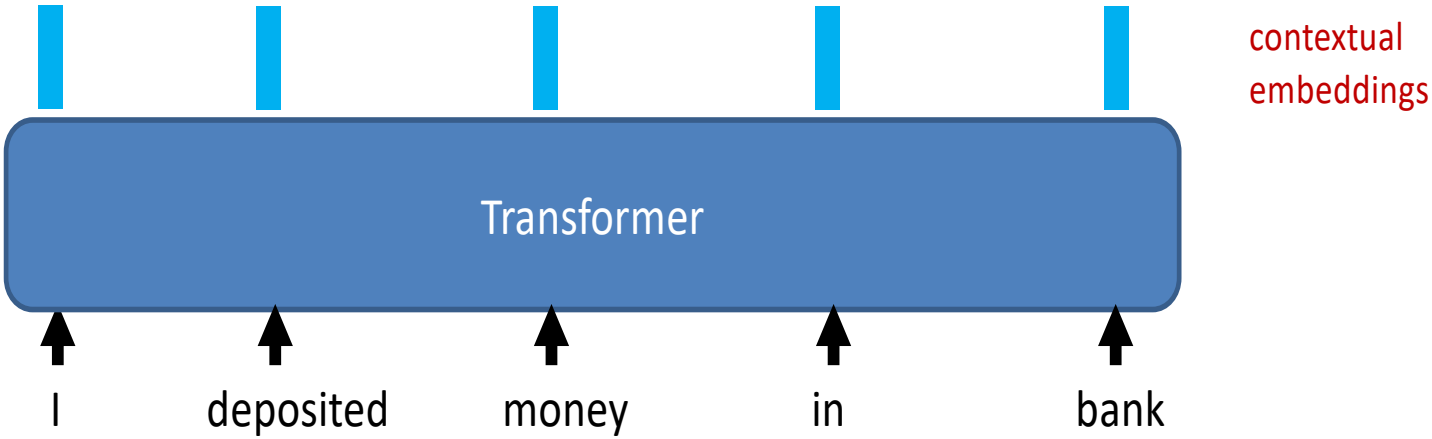


Discrete objects (words) need to be embedded for neural models

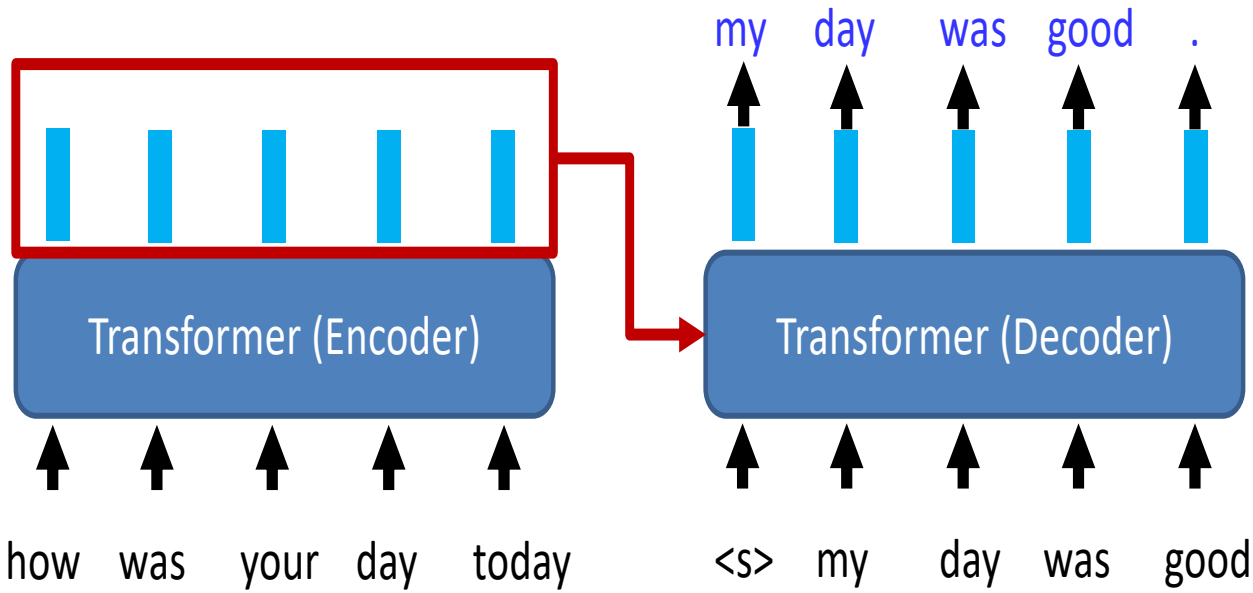
Key Idea: Similar words have similar representations.
E.g., : Chair and Table or King and Queen

NN - Transformers

- One embedding not enough for words that have multiple meanings
  - Bank – financial institution or river bank
- Transformers [Vasvani et al. 2017]: a novel neural architecture to generate context-based word embeddings



NN - Transformers

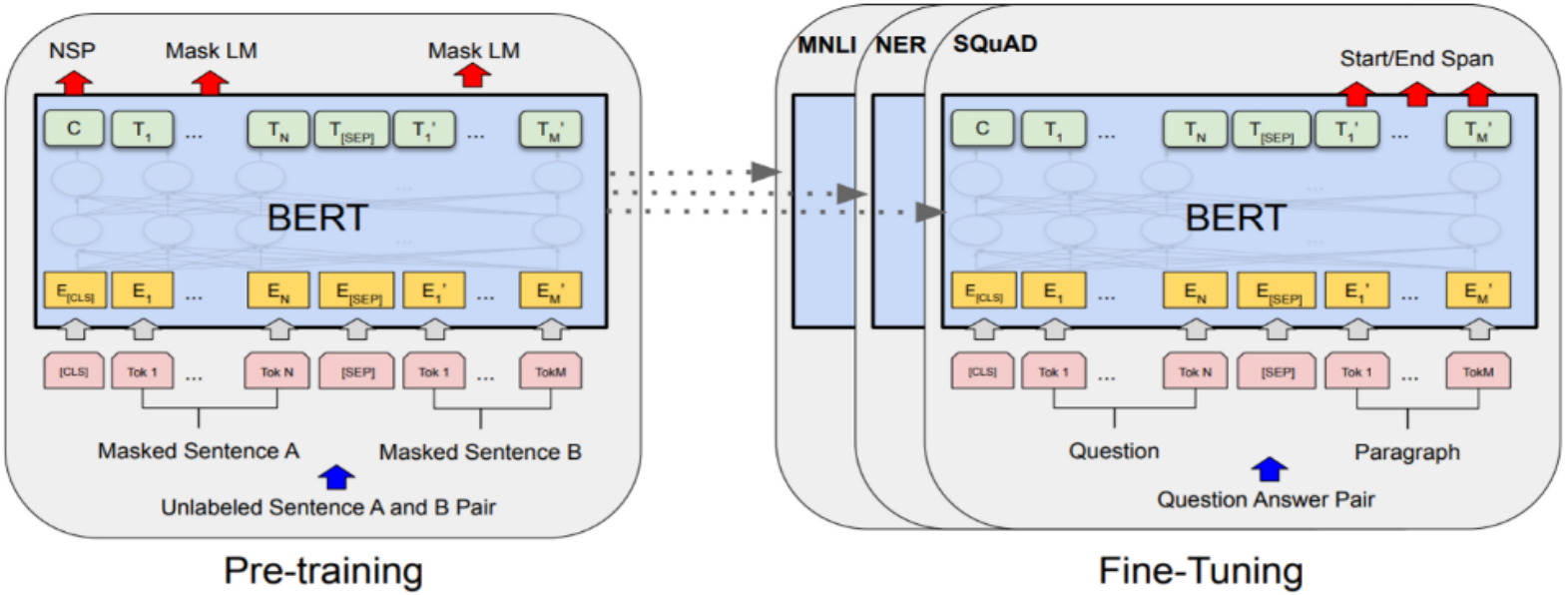


Challenge: How to train? -- using annotated training data
limited availability of (input-output) pairs ☹

BERT - a deep learning model for NLP

- BERT (Bidirectional Encoder Representations from Transformers) is a Natural Language Processing Model proposed by researchers at Google Research in 2018.
- It is fundamentally an LSTM
- BERT is basically an Encoder stack of transformer architecture.
- A transformer architecture is an encoder-decoder network that uses self-attention on the encoder side and attention on the decoder side.

Continued...



[1]

Figure: Pre-trained bert tasks

- Firstly train the model on the pre-training tasks.
- Once the pre-training is complete, the same model can be fine-tuned for a variety of downstream tasks.

[1] Bert: Pre-training of deep bidirectional transformers for language understanding

NN - Self supervised learning

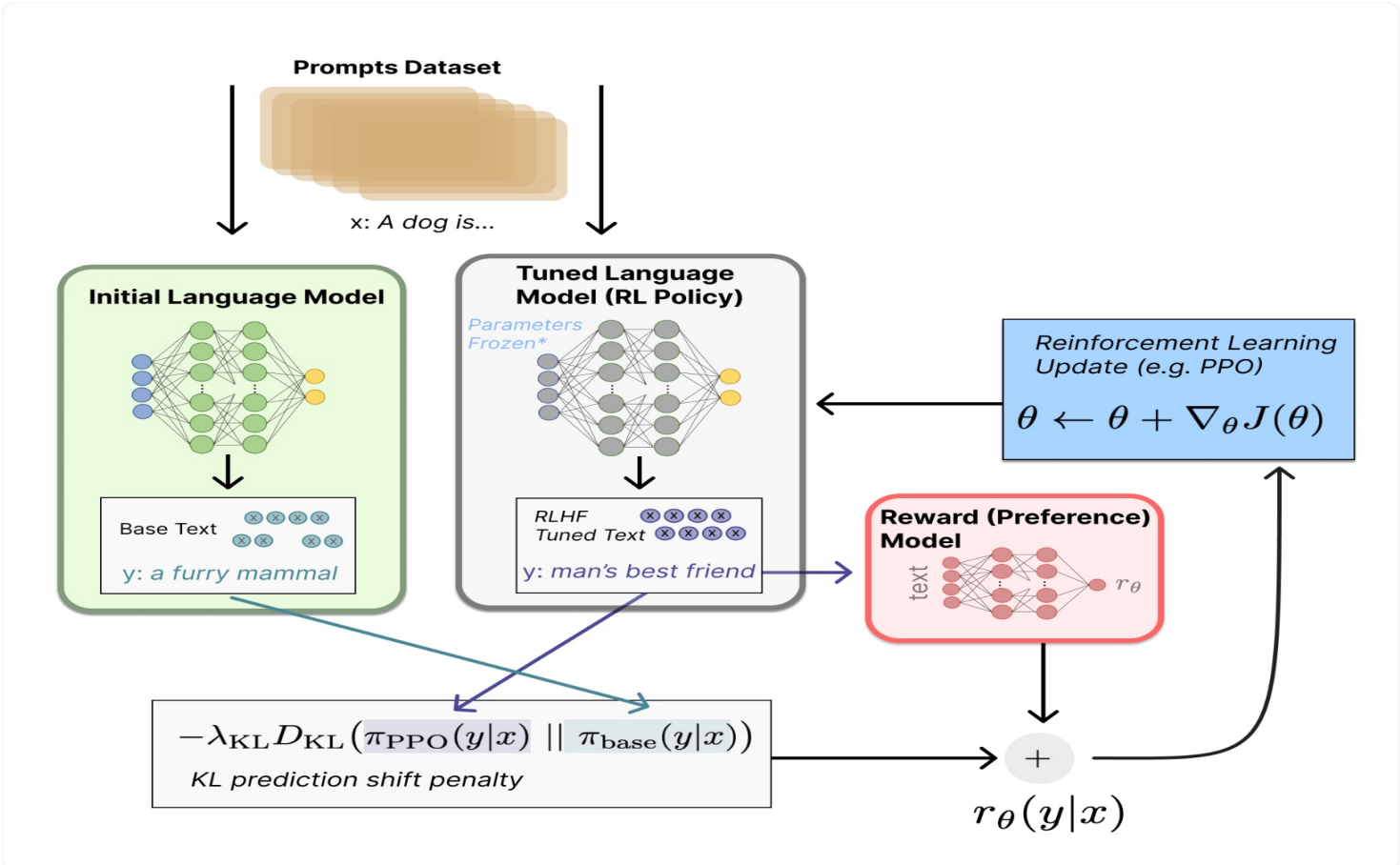
- Convert general text on the Web into huge number of (input-output) pairs

> **Linguistics** is the scientific study of human language.[1][2] It entails the comprehensive, systematic, objective, and precise analysis of all aspects of language[3] — cognitive, social, environmental, biological as well as structural.[4]

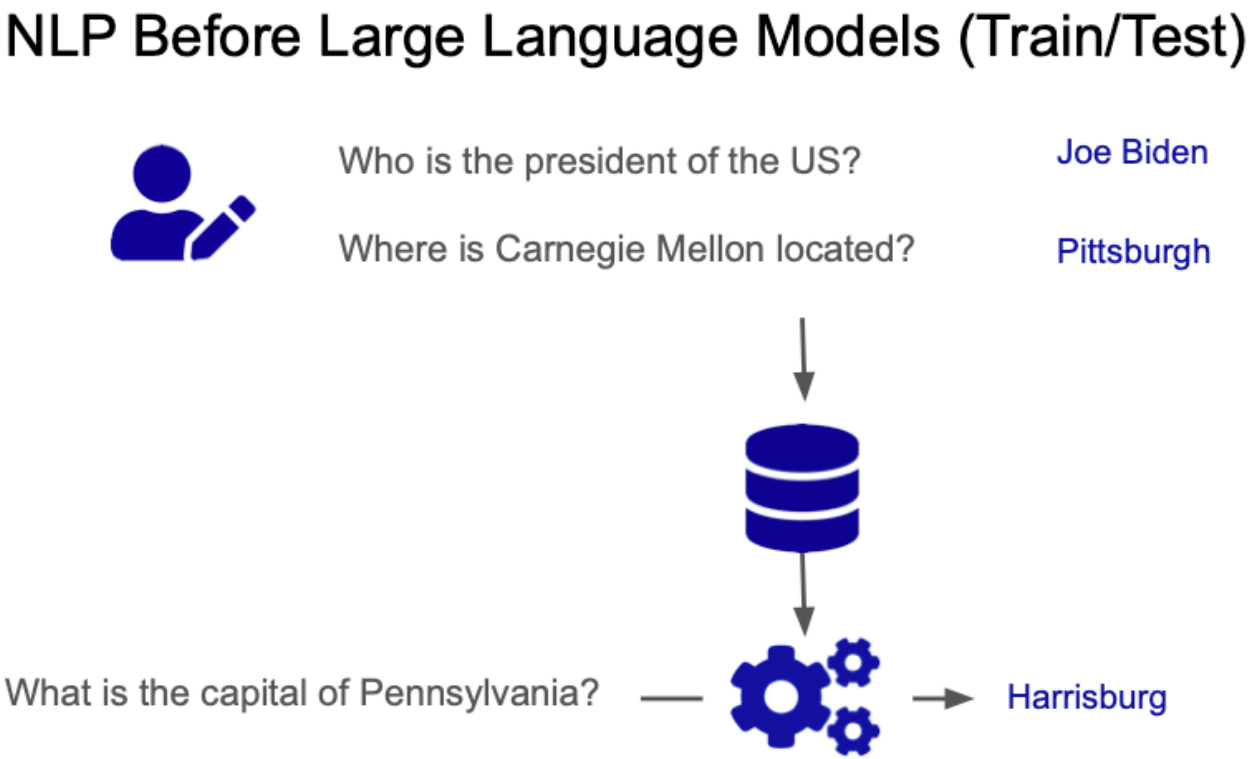
- Linguistics is the _____
- Linguistics is the scientific _____
- Linguistics is the scientific study _____
- Linguistics is the scientific study of _____
- Linguistics is the scientific study of human _____

- scientific
- study
- of
- human
- language

LLM - RLHF


Prompts Dataset
x: A dog is...
Initial Language Model
Base Text
y: a furry mammal
Tuned Language Model (RL Policy)
Parameters Frozen*
RLHF Tuned Text
y: man's best friend
Reinforcement Learning Update (e.g. PPO)
$\theta \leftarrow \theta + \nabla_\theta J(\theta)$
Reward (Preference) Model
text
r_θ
$-\lambda_{\mathrm{KL}} D_{\mathrm{KL}}(\pi_{\mathrm{PPO}}(y|x) \,||\, \pi_{\mathrm{base}}(y|x))$
KL prediction shift penalty
+
$r_\theta(y|x)$

Large Language Models



Large Language Models

NLP With Large Language Models (Prompting)

Q: What is the capital of Pennsylvania?

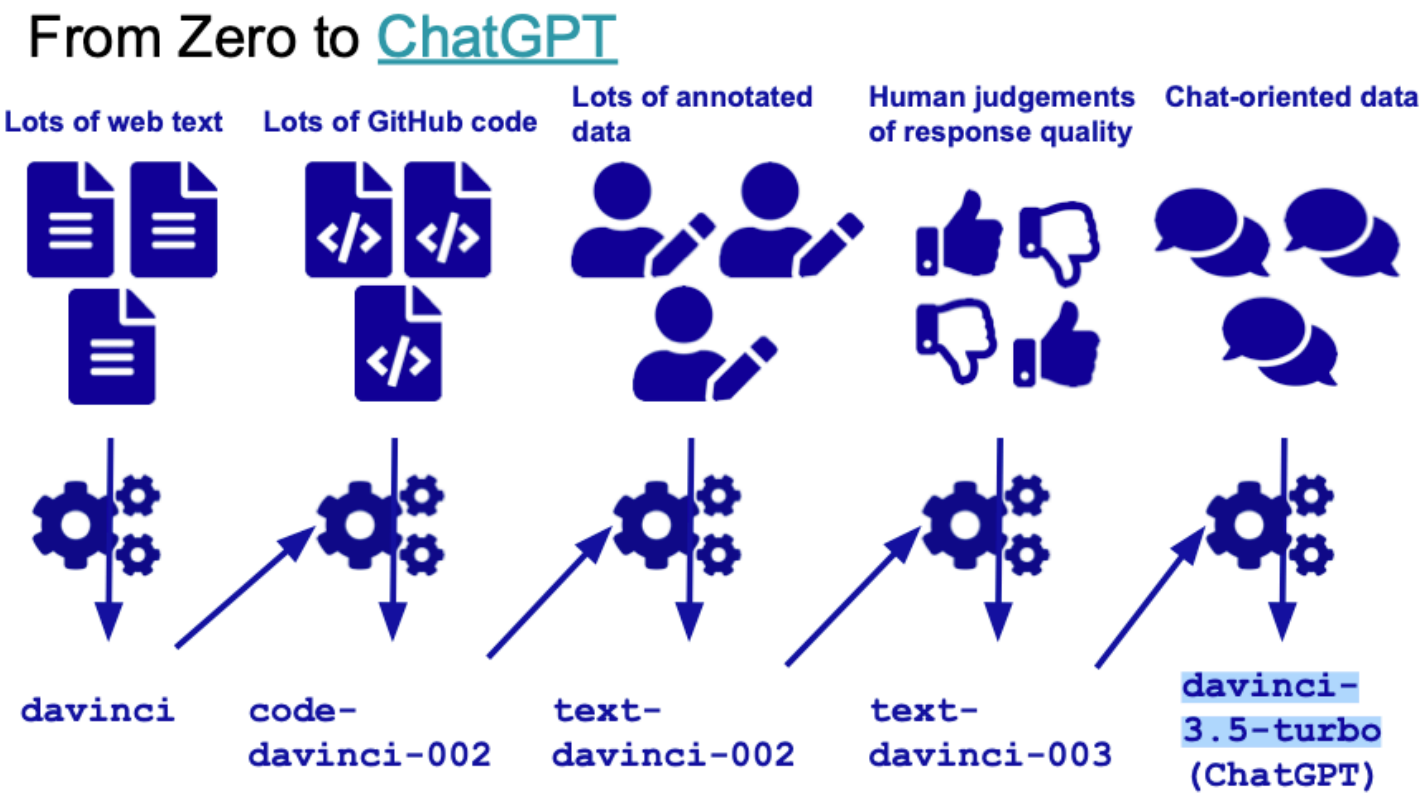
A:  Harrisburg

It's for real this time. After months of legal drama, bad memes and will-they-or-won't-they-chaos to put your favorite rom-com to shame, Elon Musk has closed his $44 billion acquisition of Twitter. Musk sealed the deal Thursday night, taking Twitter private and ousting a handful of top executives — CEO Parag Agrawal included in the process.

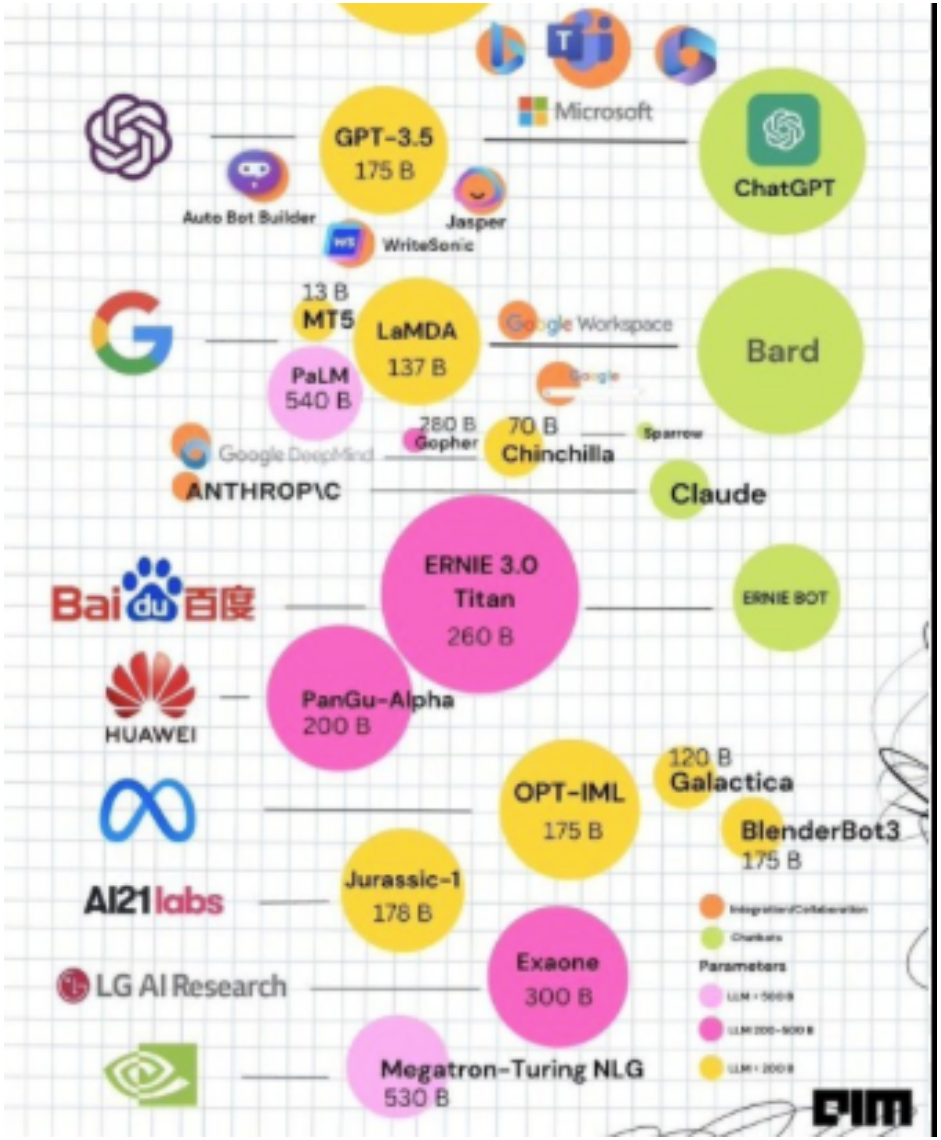
TL;DR:  Elon Musk has bought Twitter.

- Mostly through APIs such as GPT, Cohere, PaLM

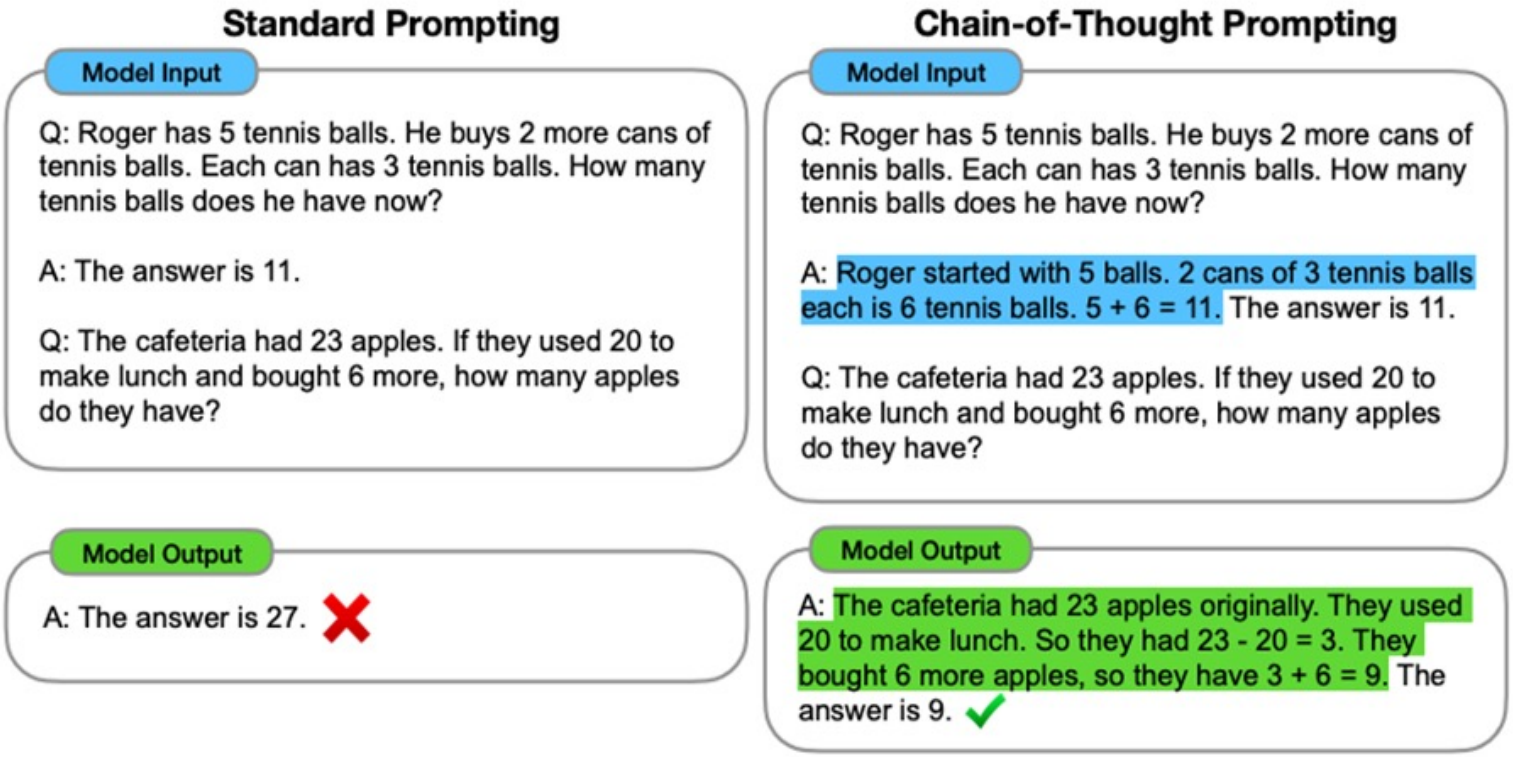
Large Language Models



Large Language Models



LLM - Prompting - Chain of Thought



Source: Wei et al. 2023

LLM - Prompting - Step by Step

(a) Few-shot

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A:

(Output) The answer is 8. ✗

(b) Few-shot-CoT

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are 16 / 2 = 8 golf balls. Half of the golf balls are blue. So there are 8 / 2 = 4 blue golf balls. The answer is 4. ✓

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: The answer (arabic numerals) is

(Output) 8 ✗

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: ***Let's think step by step.***

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

Source: Kojima et al. 2023

Big Gen AI players

- ChatGPT (OpenAI): A conversational AI chatbot based on OpenAI's GPT models, widely used for writing, coding, and customer support.
- Google Gemini (formerly Bard): Google's flagship AI assistant and LLM, integrated across Google products like Search, Docs, and Workspace.
- Microsoft Copilot: AI-powered assistant embedded in Microsoft 365 (Word, Excel, Outlook) and Windows, leveraging OpenAI's GPT models.

Big Gen AI players

- Perplexity AI: AI-powered search engine and chatbot that retrieves and summarizes real-time web information with citations. Known for fact-checking, academic research, and live updates.
- DALL · E (OpenAI): AI for generating images from text descriptions, widely used in marketing and digital art.
- Adobe Firefly: AI-powered tools for creative professionals, integrated into Photoshop and Illustrator for generative design.
- Meta AI (Llama Models): AI-powered chatbot and foundation model used across Facebook, Instagram, and WhatsApp.
- Amazon Bedrock: AWS's AI service providing access to multiple foundation models for enterprises.

Lang Chain

- LangChain is a framework designed for building applications powered by Large Language Models (LLMs)
- It simplifies the process of integrating LLMs with external data sources, retrieval mechanisms, memory, and agent-like behavior.
- Facilitates the composition of multiple LLM calls into structured workflows.

Thank you!