# Examples of successful applications

Ckeck the State of the Art site (papers with code)

# Major domains and tasks

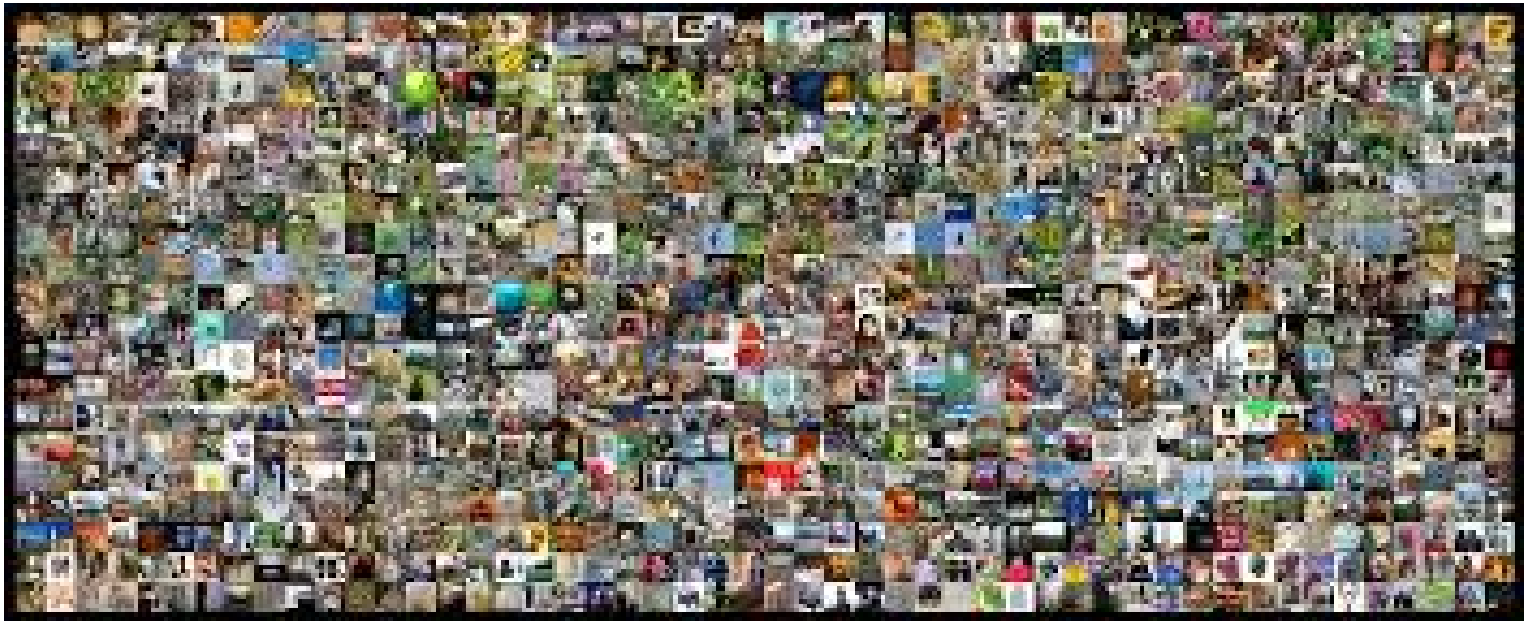
- **Image Processing**
  - Image Classification
  - Object Detection and Segmentation
  - Image Generation
    - latent space and generators
    - conditioning
    - video and multimedia
- **Natural Language Processing**
  - Lamguage Understanding and Modeling
  - Embeddings
  - Learning to complete
  - Auotregressive generation
  - The stochastic parrot metaphor

# Image Processing

# Image Classification

ImageNet (@Stanford Vision Lab)

- high resolution color images covering 22K object classes
- over 15 million labeled images from the web



# ImageNet competition

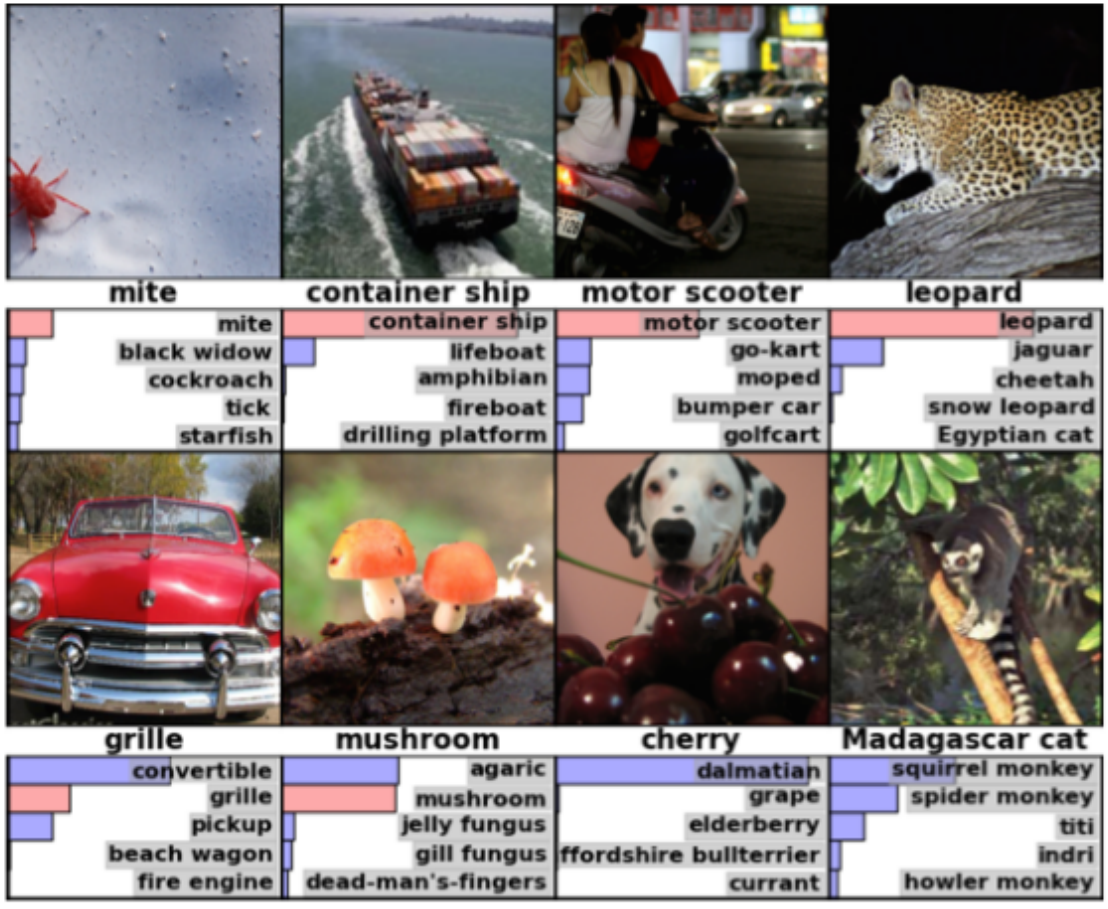
Annual competition of image classification: 2010-2017.

- 1.2 Million images, covering 1K different categories
- make five guesses about image label, ordered by confidence

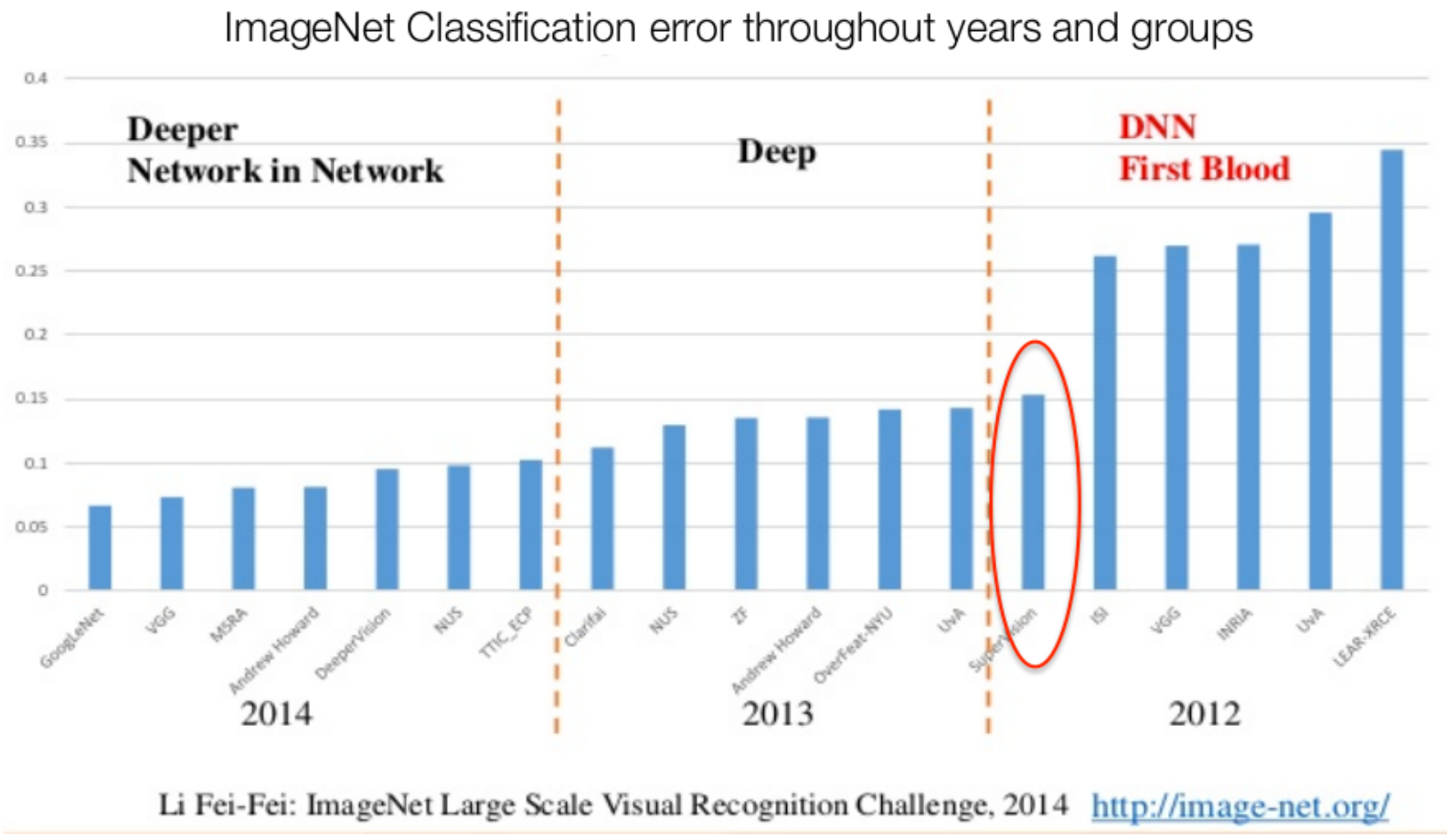
EntleBucher

Appenzeller

# ImageNet samples



# ImageNet results



# Why is ImageNet still important

Classification models are typically composed of two parts:

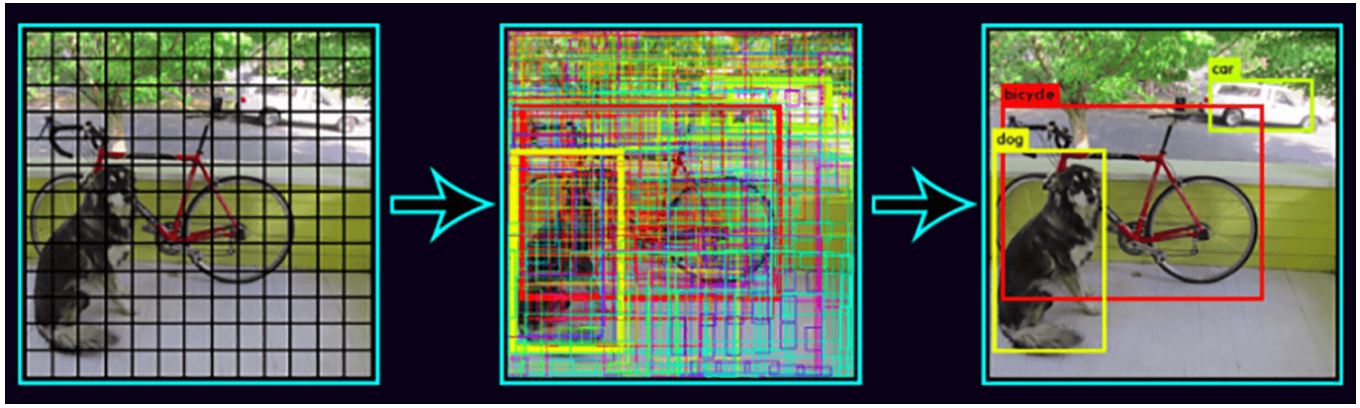
- a front-end extracting features from the input image and converting them into a vector
- a back-end, using the extracted features for classification

The front-end, **pre-trained over ImageNet**, can be reused for a lot of different applications (transfer learning).

You easily find on line the weights for these models.

# Image Detection

YOLO (You Only Look Once) is a state-of-the-art Object Detection system.




- First introduced in 2015. A new version every year.
- On a Pascal Titan X it processes images at 30 FPS and has a mAP of 57.9% on COCO test-dev (version 4).

# Image Segmentation - Scene understanding

## Video-to-Video Synthesis

# Key-points detection

Suggested reading: Keypoint Detection with Transfer Learning

# Main area of applications

- medical imaging
- autonomous driving
- pose estimation
- activity recognition
- video surveillance
- person reidentification
- satellite observations
- ...

# Image Generation

# Generative modeling

The objective of generative modeling is to learn the **distribution P(X)** of training data - that is how points are distributed inside the feature space they inhabit.

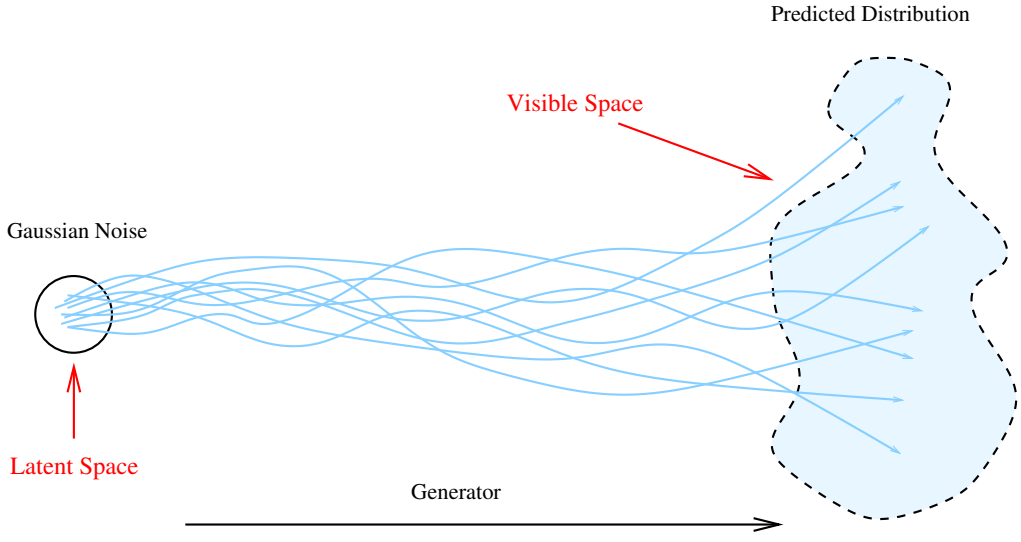
Typically, we aim to build a **generator** able to **sample** points according to the learned distribution.

But neural networks are **deterministic systems**, so how can simlulate a stochastic sampling procedure?

# The generator

We know how to build **pseudo-random generators** for simple, know distributions, such as e.g Gaussian Distributions.

So the problem reduces to learn a **transformation** mapping the known distribution to the actual distribution $P(X)$.

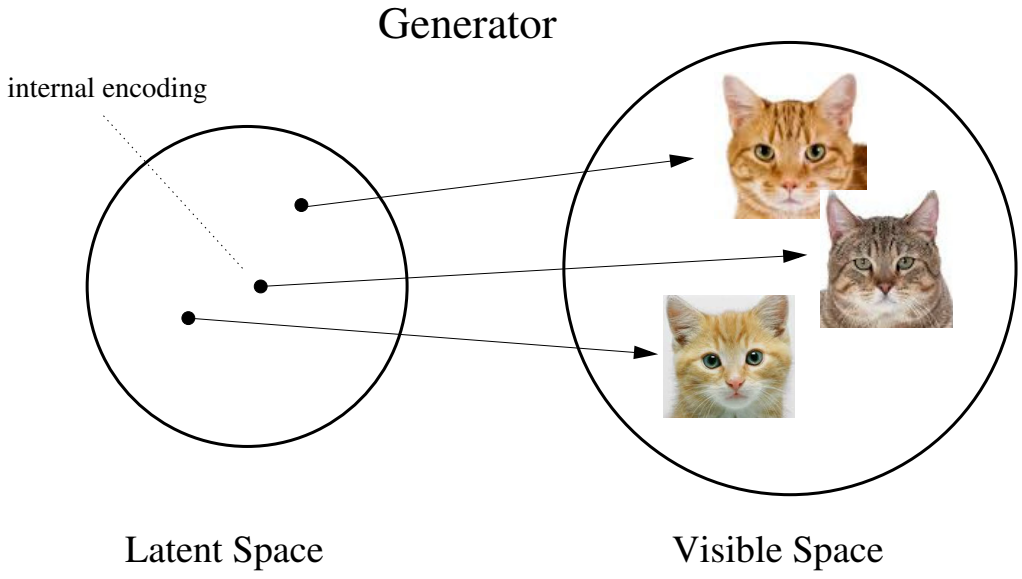



Morally, the generator learns $P(X|z)$, where z is the "**latent**" representation of X.

# Ancestral sampling

Generation is thus a two stage process (ancestral sampling):

- sample z according to a **known prior distribution**
- pass z as input to the generator, and process it to get a siginificant output




Generative models (GANs,VAEs,Diffusion,. . . ) differ in the way the generator is trained.

# The latent space

The source space is the so called latent space.

Each latent point z contains **all information** needed to generate a complete sample, hence it can be seen as an **internal encoding** (latent representation) of the given sample.

Latent values must be disseminated with a **known, regular distribution** in their space (the so called prior distribution)

Generation is a continuous process: small modifications of the seed produce small modifications of the output

Suggested reading: Comparing the latent space of generative models

# Editing trajectories and interpolation

Important points to keep in mind:

- **any** face is somewhere in the latent space
- generation is a **continuous** process: small modifications of the encoding produce small modifications of the output

**Example**:
Face generation video by Nvidia

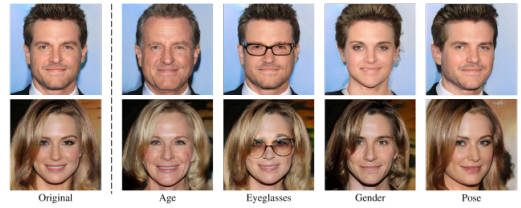
$\Rightarrow$ study of **editing trajectories** and **interpolation**

# Conditional generation

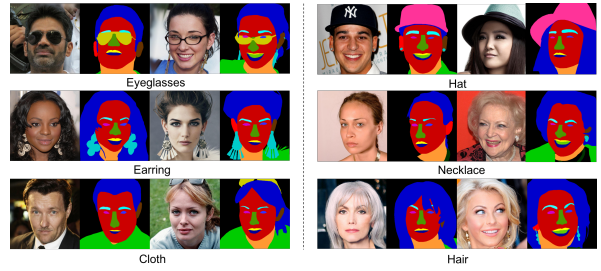
In case of conditional generation, the generator try to model $P(X|z, c)$ where $c$ is a condition integrating the latent encoding $z$.

The condition $c$ can be arbitrary:

- a label
- a segmentation
- a textual prompt
- another image
- a sequence of frames
- ...




Semantic Face Editing




Interactive Facial Image Manipulation

# Multi-modality and Video generation

Sora by OpenAI

Generative models are expanding beyond images to handle:

- **Text-to-Video**: generating coherent video sequences from text (e.g. Gen-2 by Ronway, Sora).
- **Audio**: synthesizing realistic soundscapes.
- **Multimodal Systems**: Combining text, image, video and audio.

# Natural Language Processing

# Main subfields in NLP

- **Natural Language Understanding**. Focuses on interpreting and extracting meaning from human language.
  Includes tasks such as
  - sentiment analysis,
  - named entity recognition (NER)
  - question answering
- **Natural Language Modeling** Focuses on producing human-like text based on structured or unstructured input.
  Includes tasks such as
  - text summarization,
  - dialogue generation,
  - story writing
  - retrieval augmented generation

**Speech recognition/generation** is additional topic bridging text and audio.

# NLU and language embeddings

Natural Language understanding instruct models to understand context, ambiguity, and nuances in language.

All previous tasks require a meaningful representation of words, provided by so called **embeddings**.

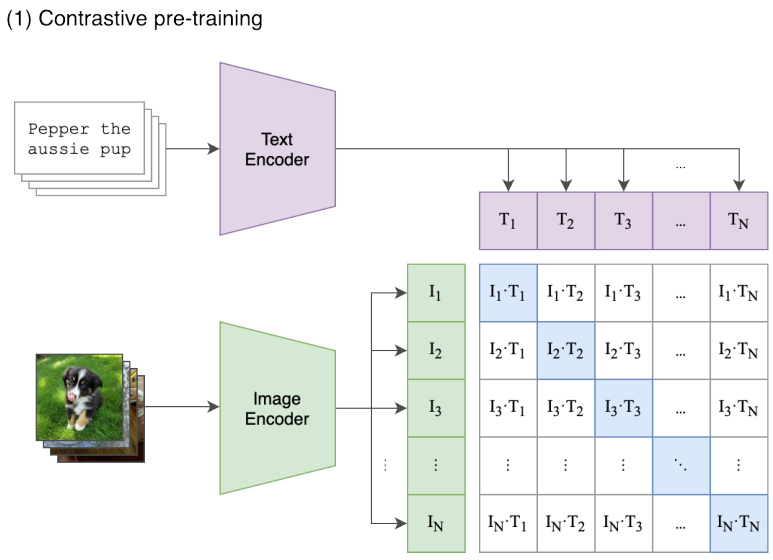
Embeddings like Word2Vec, GloVe, FastText, and Transformer based embeddings are designed to map words/sentences into a high-dimensional vector space.

They help models understand relationships between words, such as similarity, analogy, and context.

# Bridging Text and Images

A recent technology allowing to jointly analyze text and images is CLIP: Contrastive Language-Image Pretraining.

- Trained on a massive dataset of text-image pairs
- Maps text and images to a shared latent space, where similar text and image pairs have closer embeddings.
- Key for aligning textual descriptions with visual features.



# Language Modeling

# Difference between language and image generation

Images involve spatial relationships, often modeled **holistically** or in parallel.

NLP operates on **sequences** of tokens (e.g., words or subwords), where order is critical.

$x$ is a sequence of tokens $x_1 x_2 \ldots x_n$ and we are interested to model the mechanisms underlying their concatenation.

# Modeling $P(x_1 x_2 \ldots x_n)$


- Generative models estimate the joint probability of a text sequence,

$$P(x_1 x_2 \ldots x_n)$$

- Using the **chain rule**, this is broken into **conditional probabilities**:

$$P(x_1) \cdot P(x_2|x_1) \cdot P(x_3|x_1 x_2) \ldots P(x_n|x_1 \ldots x_{n-1})$$

- In practice you train the model to guess the next token completing the given sequence
- This enables sequential generation, where each token depends on its predecessor

# Autoregression in action


| **current output** | | **alternatives** |
|---|---|---|
| $\langle start \rangle$ | next? | . . . The . . . A . . . There . . . Once . . . |

# Autoregression in action


| **current output** | | **alternatives** |
|---|---|---|
| $\langle start \rangle$ | next? | . . . The . . . A . . . There . . . Once . . . |
| The | next? | . . . day . . . cat . . . night . . . |

# Autoregression in action


| **current output** | | **alternatives** |
|---|---|---|
| $\langle start \rangle$ | next? | . . . The . . . A . . . There . . . Once . . . |
| The | next? | . . . day . . . cat . . . night . . . |
| The cat | next? | . . . sleeps . . . came . . . is . . . went . . . |

# Autoregression in action


| **current output** | | **alternatives** |
|---|---|---|
| $\langle start \rangle$ | next? | . . . The . . . A . . . There . . . Once . . . |
| The | next? | . . . day . . . cat . . . night . . . |
| The cat | next? | . . . sleeps . . . came . . . is . . . went . . . |
| The cat is | next? | |

And so on.

# From n-grams to Transformers

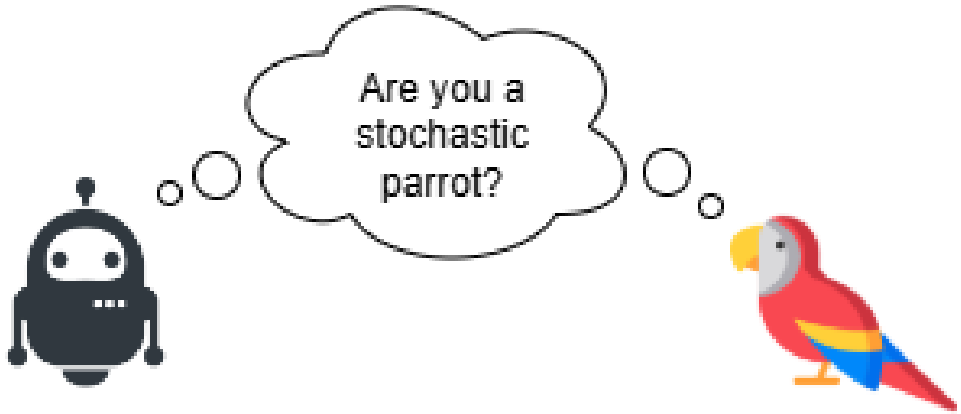
- **n-grams**
  - Estimate probabilities based on fixed window size.
  - Example: Bigram model uses $P(x_n|x_{n-1})$
  - Limitation: Can't capture long-range dependencies.

- **Deep Neural Networks approach**
  - **Recursive Neural Networks** (RNNs) introduced sequence processing but struggled with long-term dependencies.
  - **Transformers**, with **self-attention**, handle long-range context efficiently and are the foundation of modern NLP.

# The stochastic parrot metaphor

Generative models predict the most probable next token, mimicking patterns from training data.

This "stochastic parrot" behavior raises questions: are these models truly intelligent, or are they sophisticated mimics?




Intelligence as pattern recognition: does human intelligence also emerge from advanced pattern processing?