Introduction to Deep Learning

ITM528 Deep learning

Taemoon Jeong

Course Introduction

Instructor & Teaching Assistant

- **Instructor**
  - Taemoon Jeong, taemoon-jeong@korea.ac.kr

- **Teaching Assistant**
  - SeulBi Lee, seulbi@ds.seoultech.ac.kr

Syllabus

- Introduction
- Linear model
- Multilayer Perceptron
- Convolutional Neural Networks
- Recurrent Neural Networks
- Attention and Transformers
- GPT Siblings
- Deep Generative Models

Evaluation

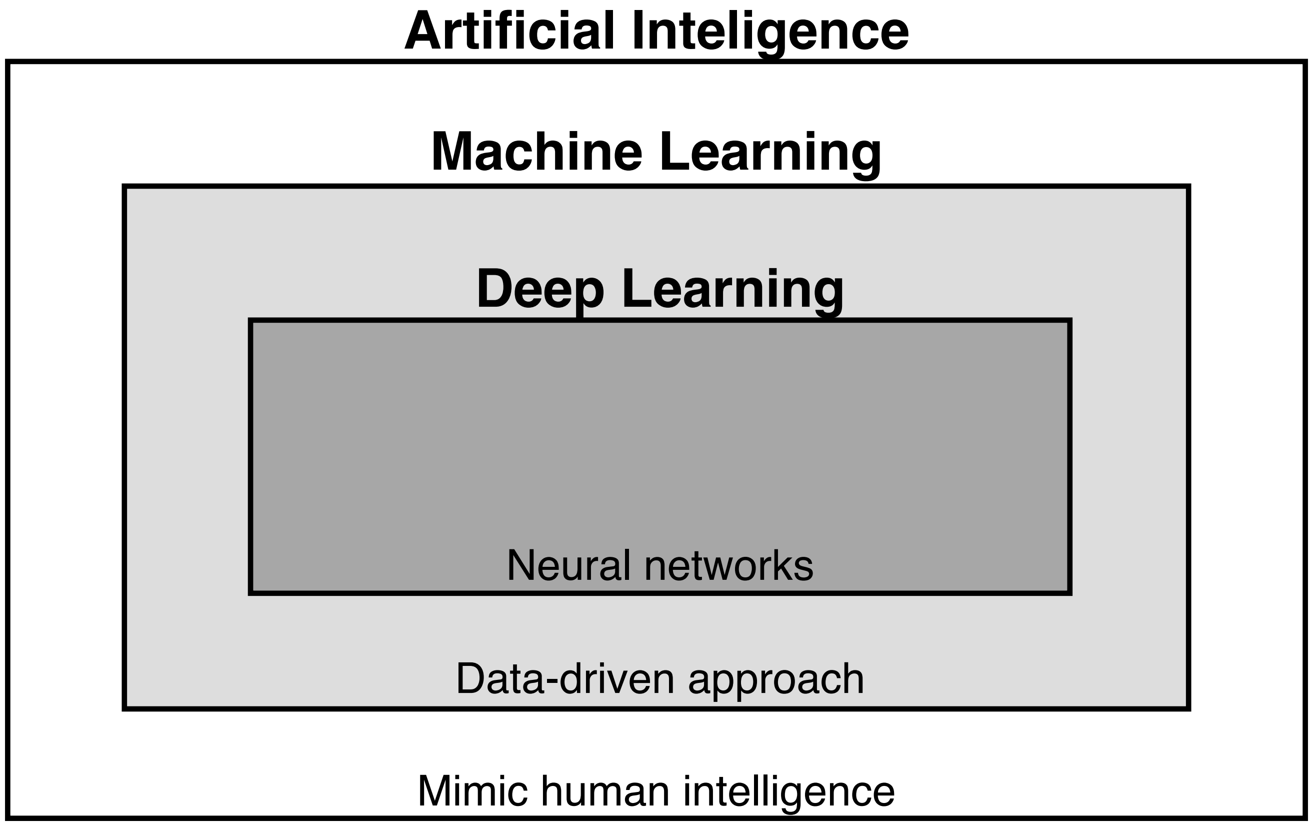
- **Midterm Exam** (35%)
  - Scheduled for October 30th
  - ~ Convolutional Neural Networks
- **Final Exam** (35%)
  - Scheduled for December 18th
  - ~Deep Generative Models
- **Assignment** (30%)
  - Four code Implementation (CNN, RNN, Transformer, Generative Models)
  - In these assignments, we will use Pytorch(https://pytorch.org/) and Google Colab(https://colab.google/).

Introduction to Deep Learning

Introduction

What is Deep Learning?

Introduction



Introduction

We will be learning about Good-old-Fashioned Deep Learning

Introduction

- Key Components of Deep Learning
  - The **data** that we can learn from
  - A **model** of how to transform the data
  - An **objective function** that quantifies how well the model is doing.
  - An **algorithm** to adjust the model's parameters to optimize the objective function

Data

How do we **represent** the data?

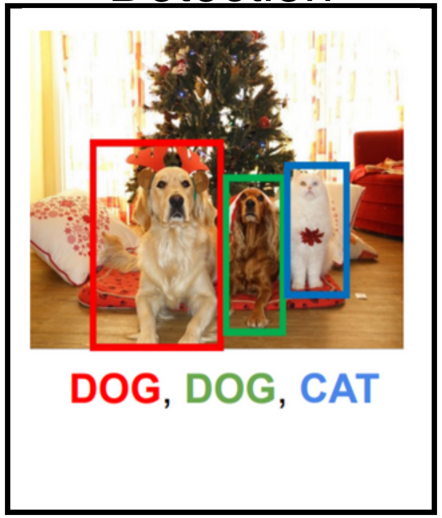
Data

- Data depend on the type of the problem to solve.

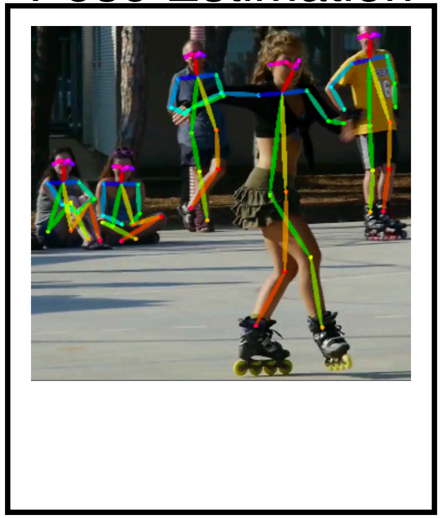
Classification

Semantic Segmentation

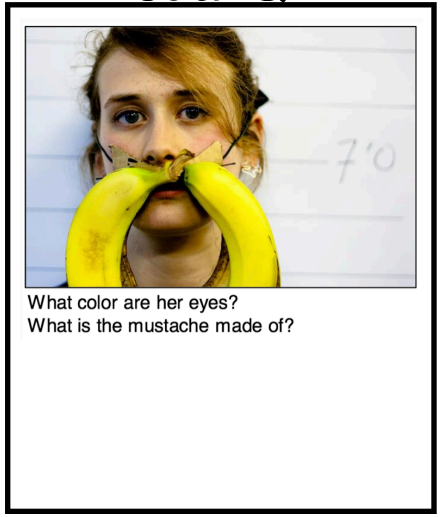
Detection



Pose Estimation



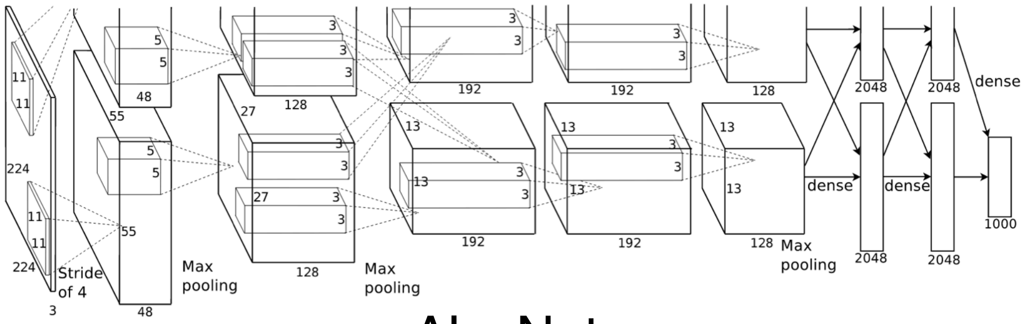
Visual QnA



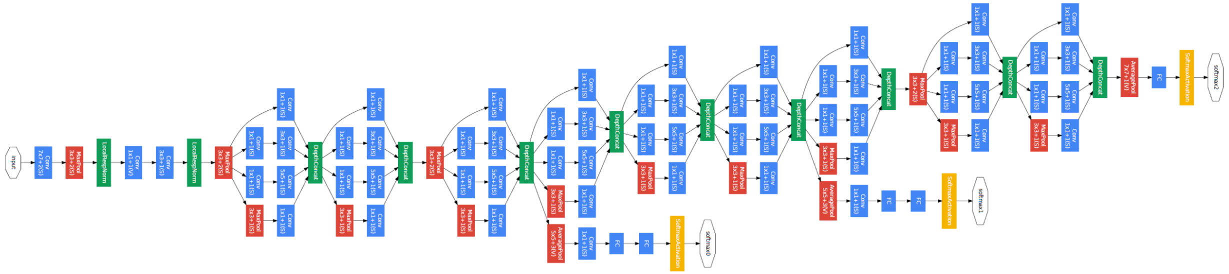
Model

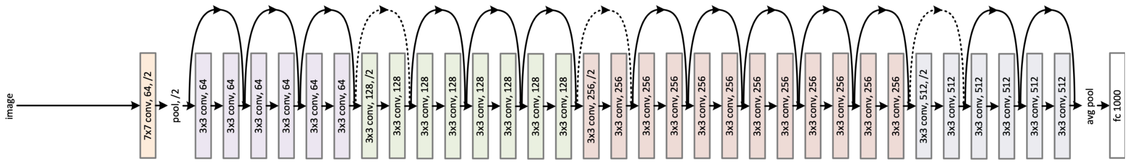
How do we transform an input to a corresponding target?

Model

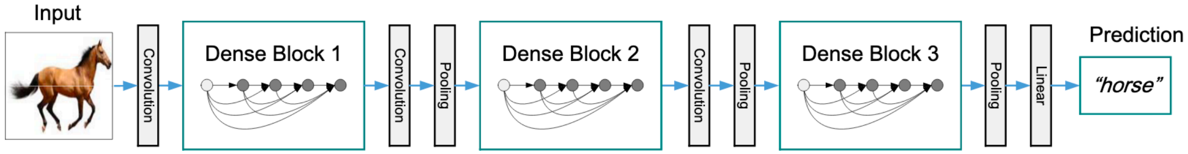


AlexNet

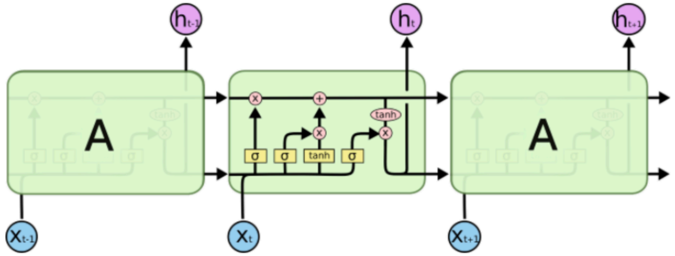

GoogLeNet



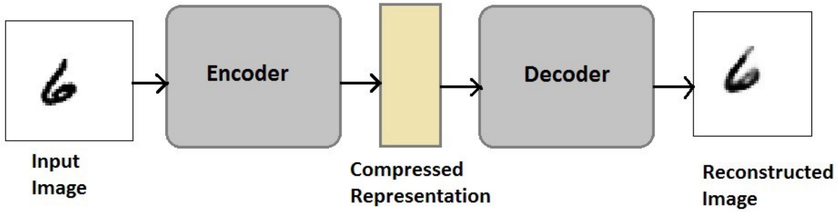
ResNet



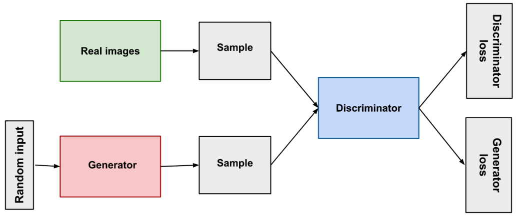
DenseNet



LSTM



Deep AutoEncoders



GAN

Objective Function

How can we evaluate how good our model is?

Objective Function

- The **Objective** function is a proxy of what we want to achieve.

Regression Task

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\sum_{d=1}^{D}(y_i^{(d)} - \hat{y}_i^{(d)})^2$$

Classification Task

$$\mathrm{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{d=1}^{D} y_i^{(d)} \log \hat{y}_i^{(d)}$$

Probabilistic Task

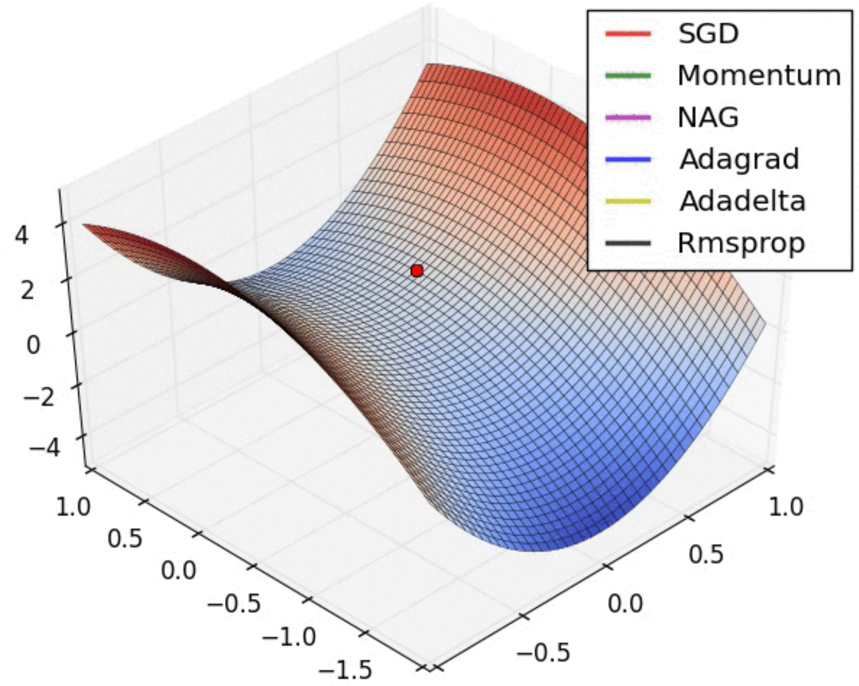
$$\mathrm{MLE} = \frac{1}{N}\sum_{i=1}^{N}\sum_{d=1}^{D} \log \mathcal{N}(y_i^{(d)}; \hat{y}_i^{(d)}, 1)$$

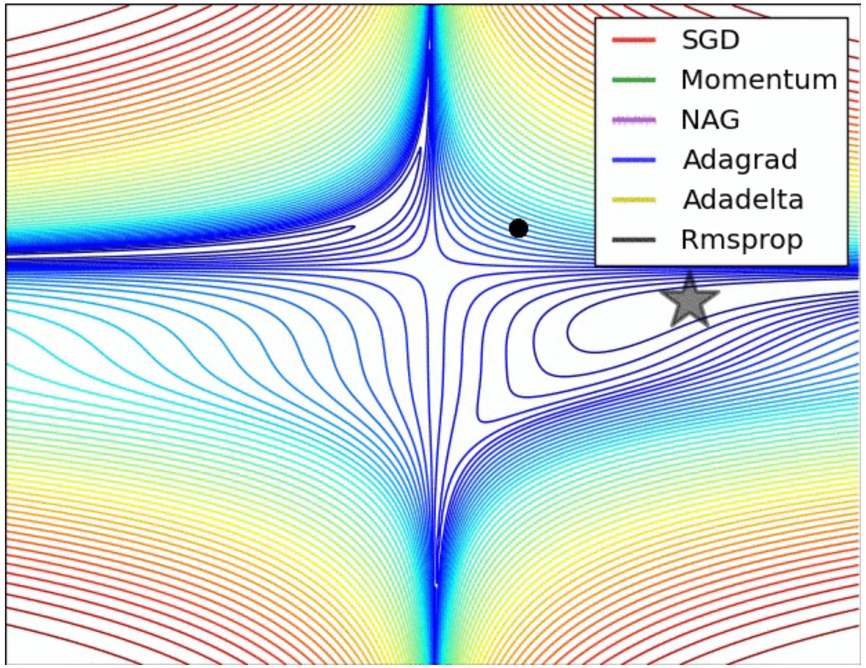
(=MSE)

Algorithm

How can we learn (or optimize) the parameters of a neural network?

Algorithm





Dropout

Early stopping

k-fold validation

Weight decay

Batch normalization

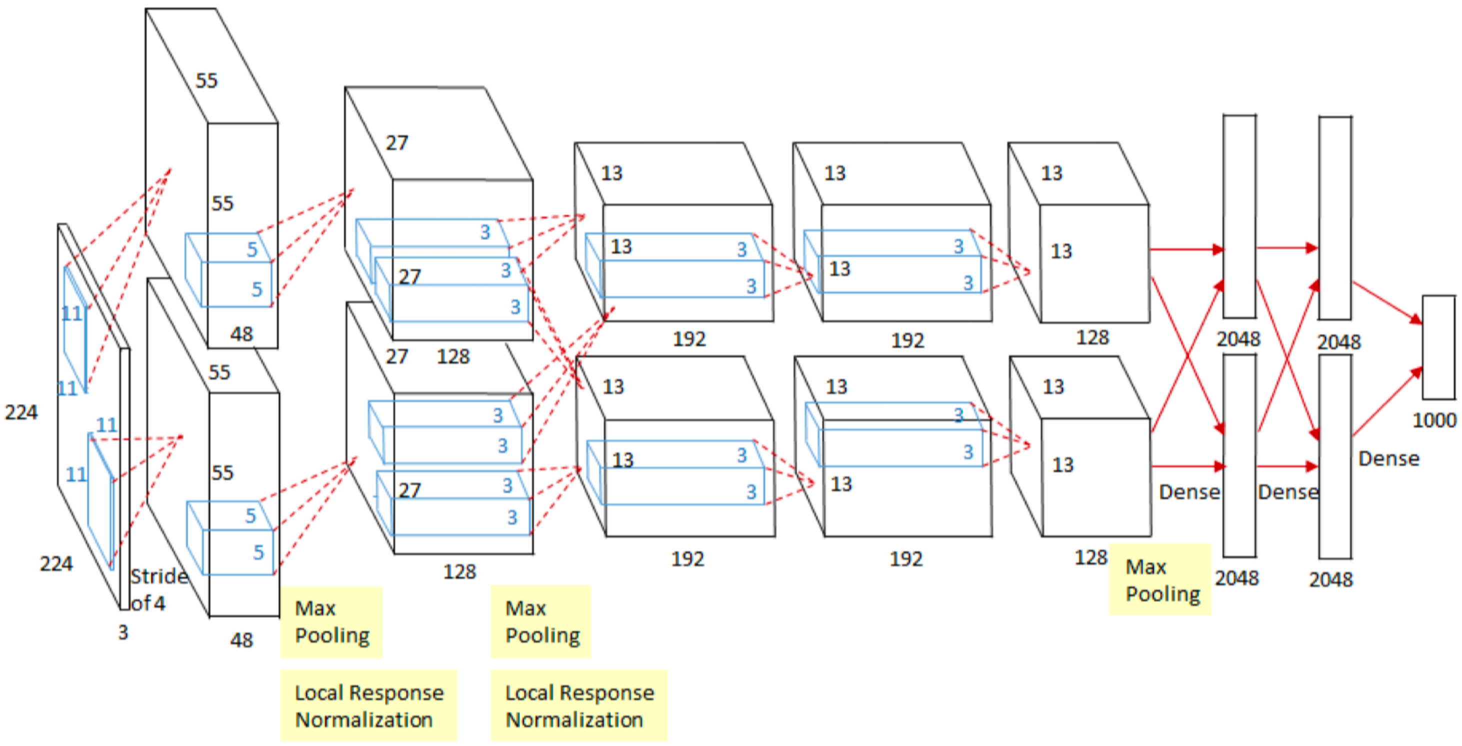
MixUp

Ensemble

Bayesian Optimization

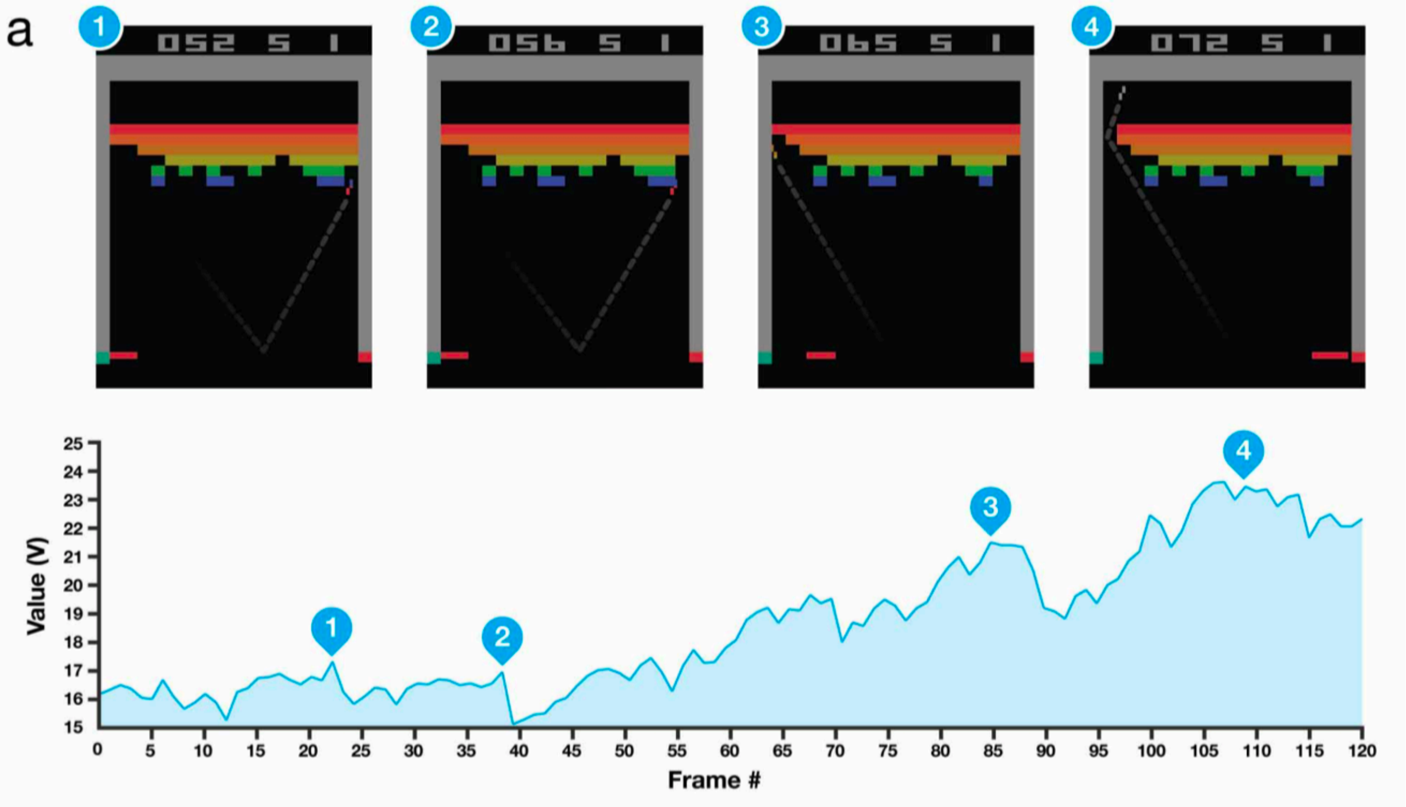
Historical Review

2012 - AlexNet



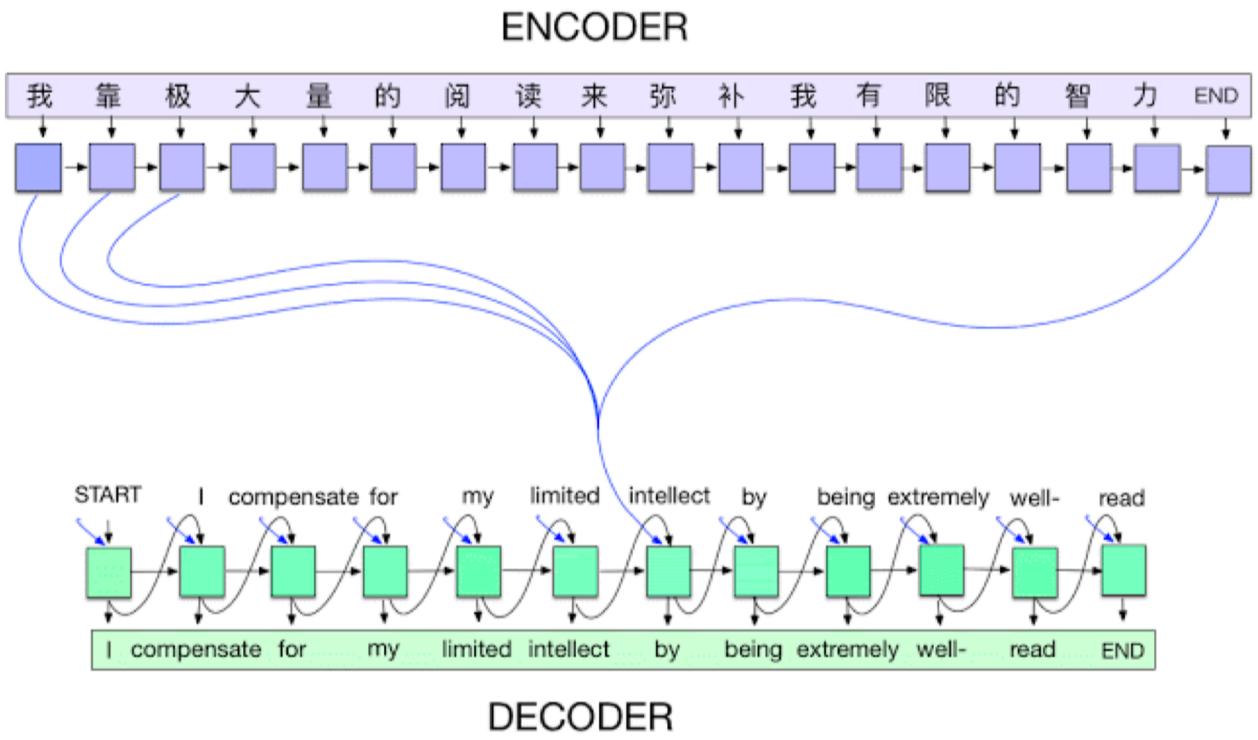
“ImageNet Classification with Deep Convolutional Neural Networks,” 2012

2013 - DQN



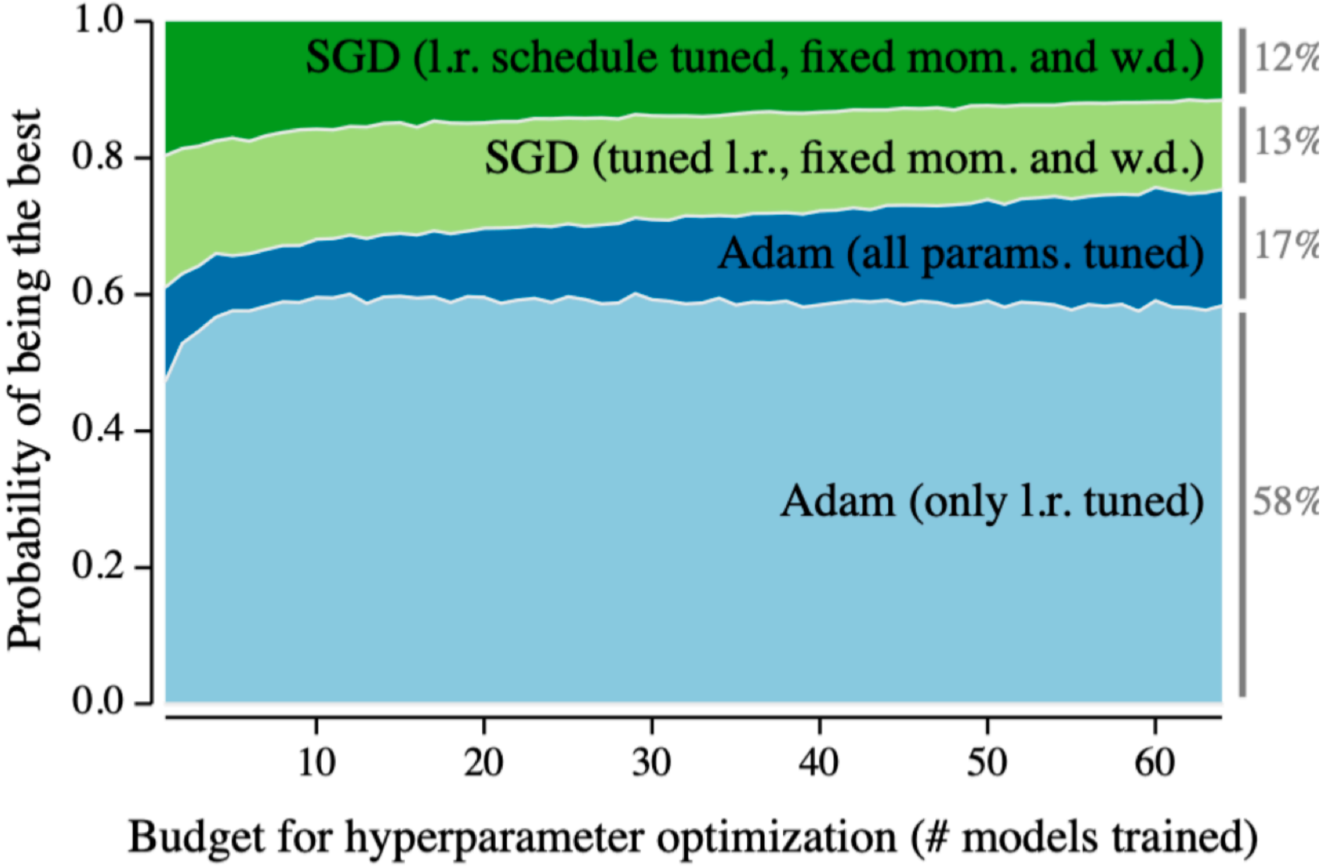
"Playing Atari with Deep Reinforcement Learning," 2013

2014 - Encoder & Decorder



"Neural Machine Translation by Jointly Learning to Align and Translate," 2014

2014 - Adam Optimizer



"Adam: A Method for Stochastic Optimization," 2014

2015 - Generative Adversarial Network

"Generative Adversarial Networks," 2015

2015 - Generative Adversarial Network

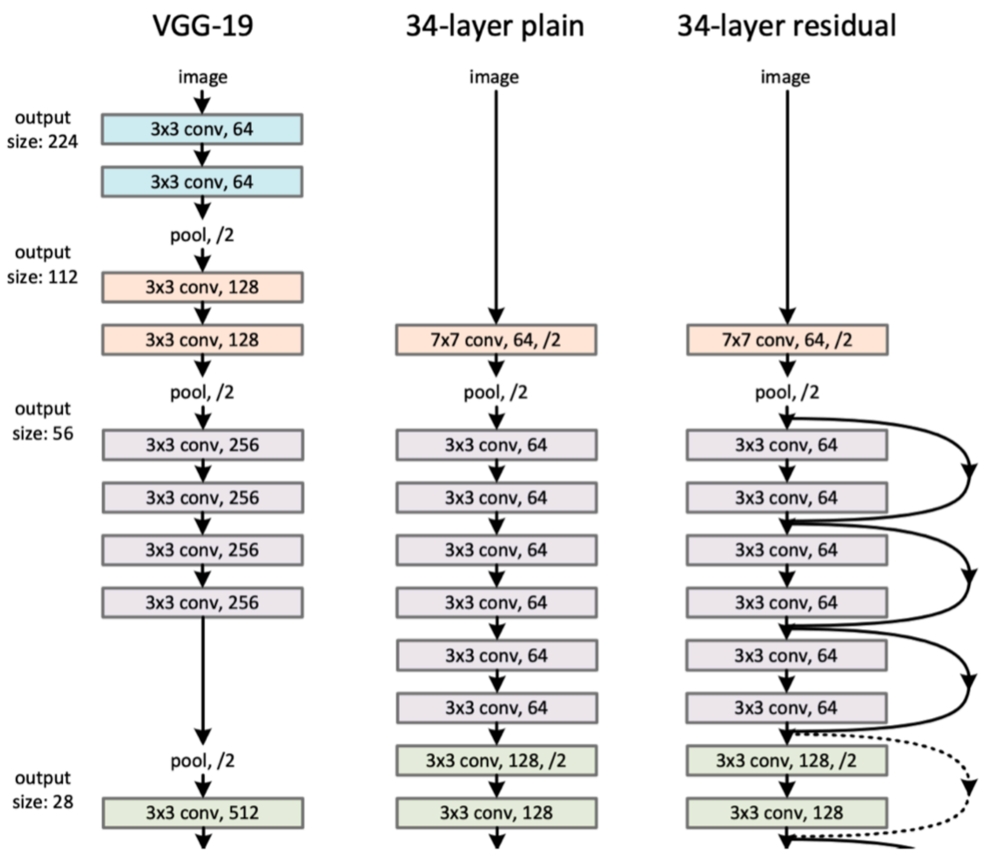
Acknowledgments

We would like to acknowledge Patrice Marcotte, Olivier Delalleau, Kyunghyun Cho, Guillaume Alain and Jason Yosinski for helpful discussions. Yann Dauphin shared his Parzen window evaluation code with us. We would like to thank the developers of Pylearn2 [12] and Theano [7, 1], particularly Frédéric Bastien who rushed a Theano feature specifically to benefit this project. Arnaud Bergeron provided much-needed support with LaTeX typesetting. We would also like to thank CIFAR, and Canada Research Chairs for funding, and Compute Canada, and Calcul Québec for providing computational resources. Ian Goodfellow is supported by the 2013 Google Fellowship in Deep Learning. Finally, we would like to thank Les Trois Brasseurs for stimulating our creativity.

"Finally, we would like to thank **Les Trois Brasseurs** for stimulating our creativity."

"Generative Adversarial Networks," 2015

2015 - Residual Networks



"Deep Residual Learning for Image Recognition," 2015

2017 - Transformer

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

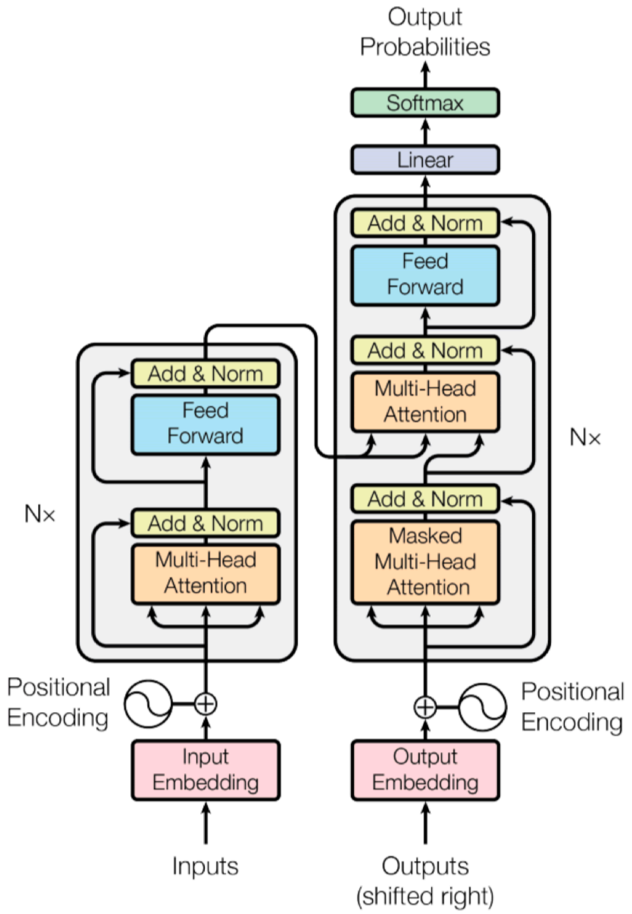
Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

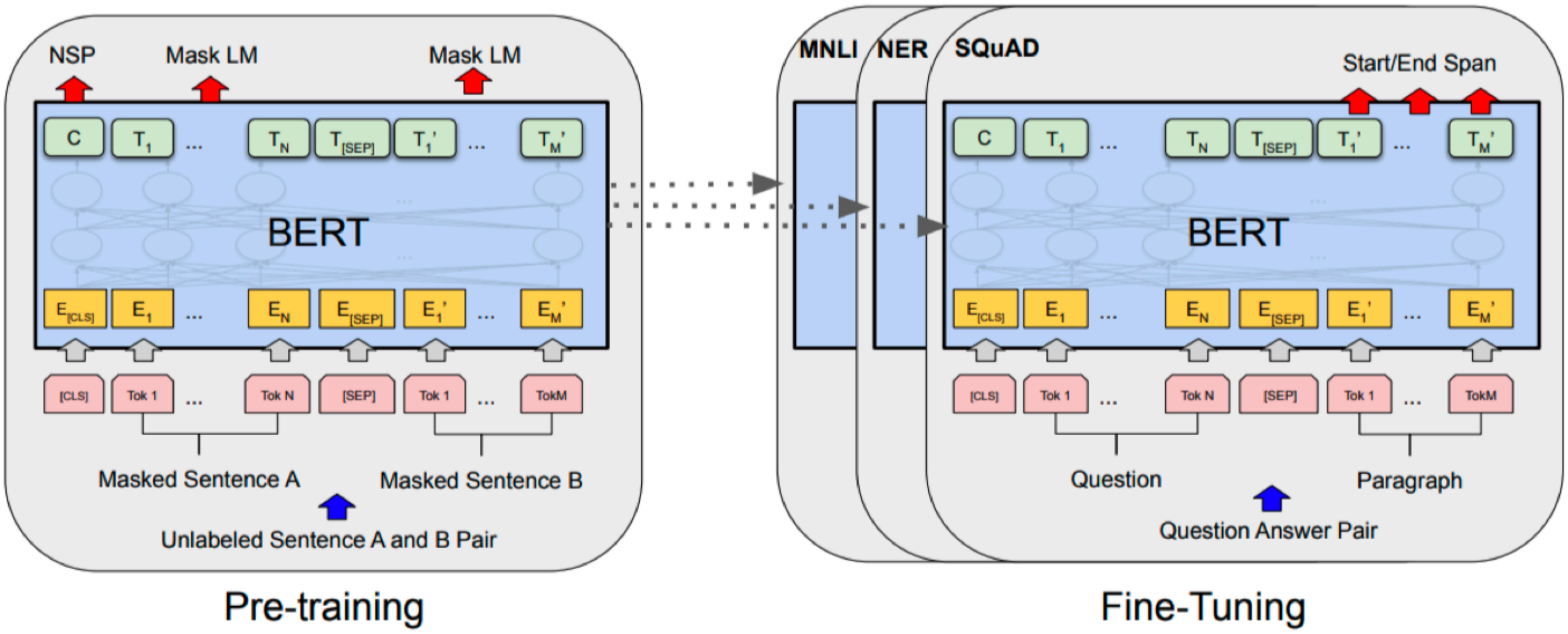
Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com



"Attention Is All You Need," 2017

2018 - BERT (fine-tuned NLP models)



Bidirectional **E**ncoder **R**epresentations from **T**ransformers

"BERT: Pre-training of Deep Bidiectional Transformers for Language Understanding," 2018

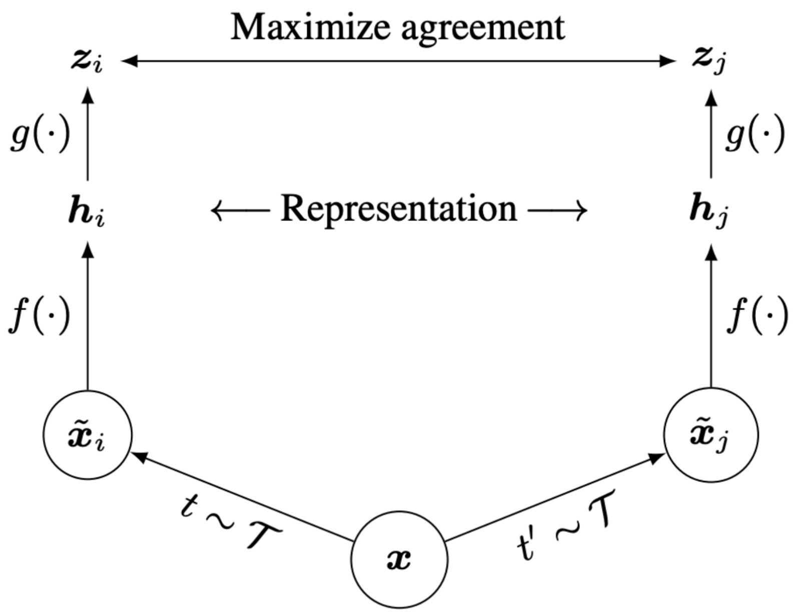
2019 - Large Language Models

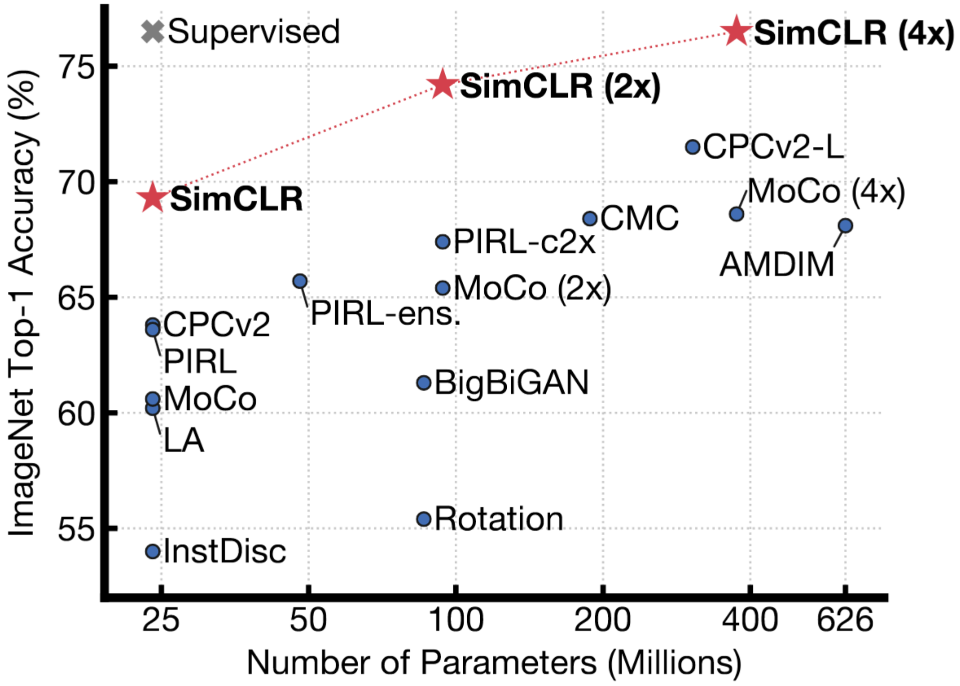
OpenAI

GPT-3, an autoregressive language model with 175 billion parameters

"Language Models are Unsupervised Multitask Learners," 2019
"Language Models are Few-Shot Learners," 2020

2020 - Self Supervised Learning

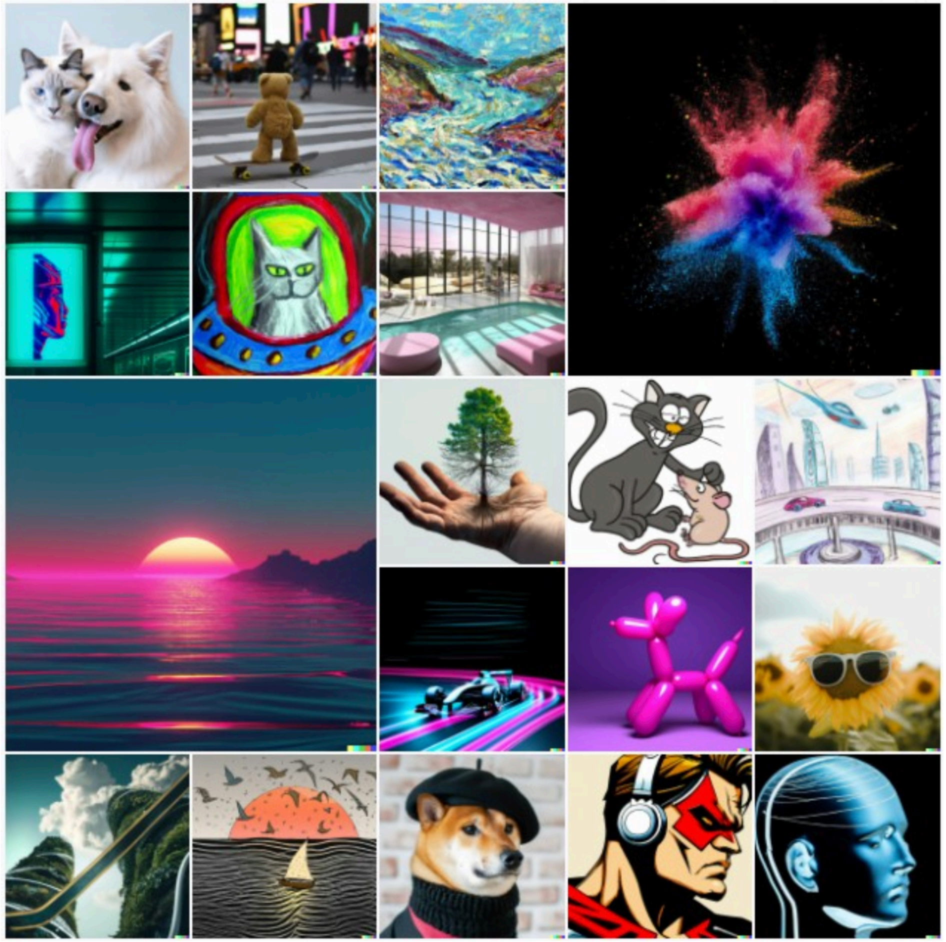




SimCLR: a simple framework for contrastive learning of visual representations

"A Simple Framework for Contrastive Learning of Visual Representations," 2020

2022 - Conditional Generative Models



"Zero-Shot Text-to-Image Generation," 2022

2024 - Generative Models (Text to Video)

a toy robot wearing **a green dress and a sun hat** taking a pleasant stroll in **Johannesburg, South Africa** during **a winter storm**

https://openai.com/index/sora/