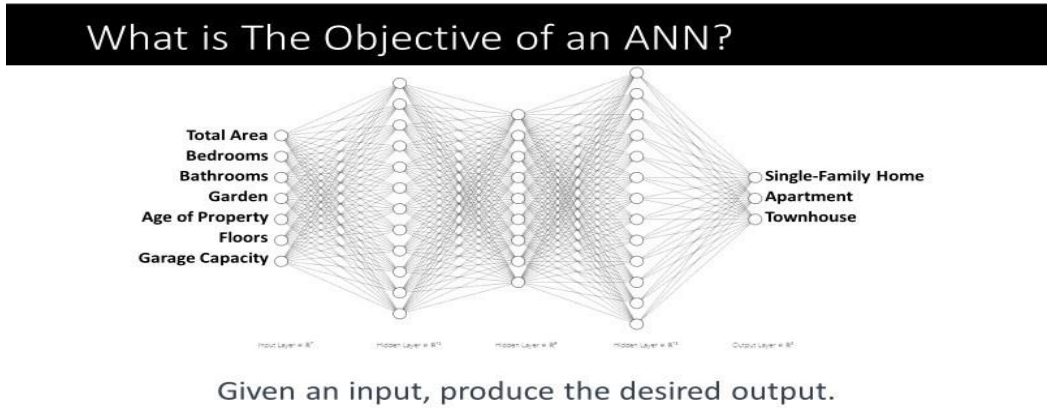
Generative AI & LLM with Python

Generative AI

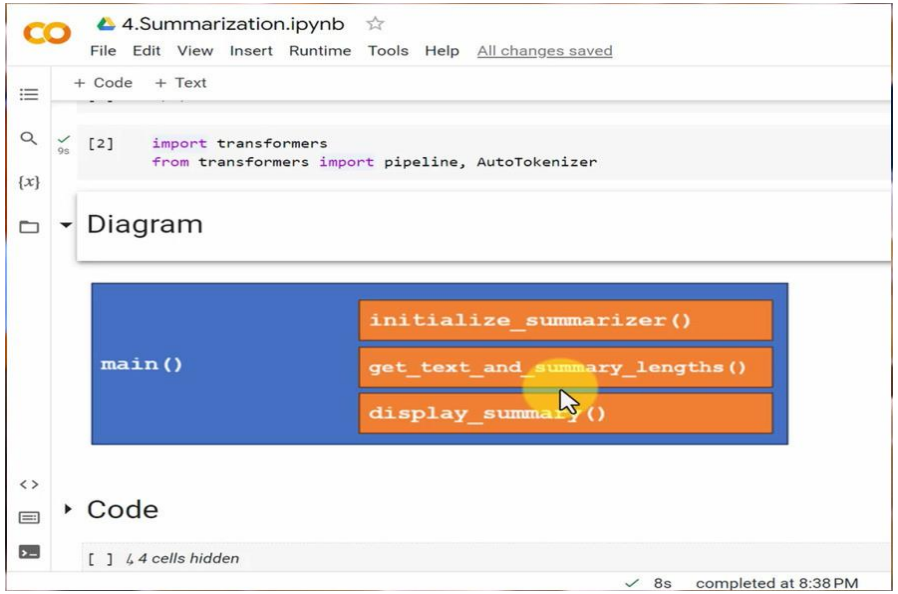
Person: That sounds cool. Can you tell me a joke?
Chatbot: Sure! Why don't scientists trust atoms? Because they make up everything!
Person: Haha, that's a good one! Now, can you help me with a math problem?
Chatbot: Of course! Feel free to tell me the math problem, and I'll do my best to help you solve it. **Person:** What's the square root of 144?
Chatbot: The square root of 144 is 12.
Person: Thanks! Also, I'm planning a trip to Paris. Do you have any travel tips?

Topics

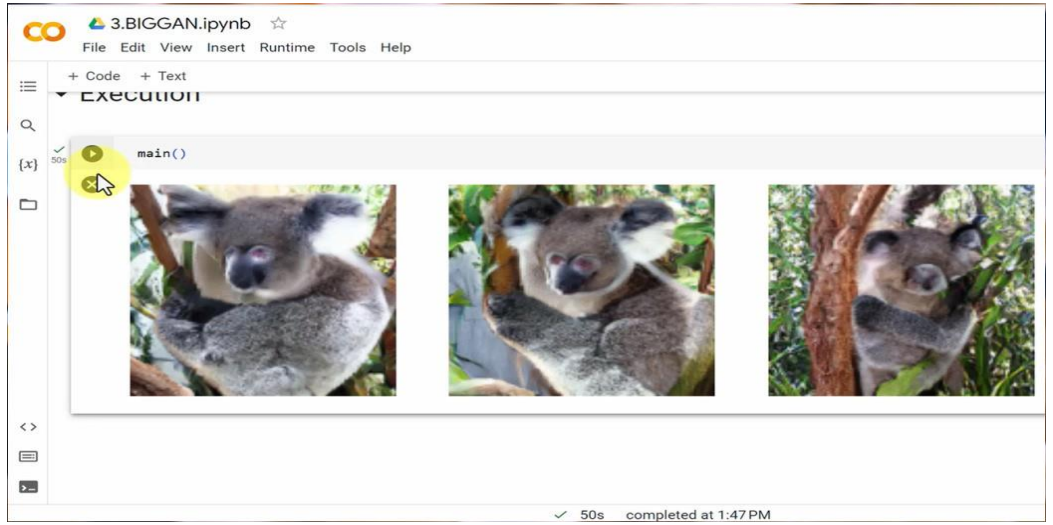
1.Introduction	2. Fundamentals of ANN	3.Getting to Know Transformers
4.Text Generation, Fill the Mask and QEA	5.Summarization, Translation and Chatbot	6.Prompt Engineering for Image Generation
7.Getting to know GANs	8.Image Generation with GANs	9.Getting to Know Diffusers
10.Auto-Image Generation	11.Hight Quality and Conditional Image Generation	12.Other pre-treined Image Generatin Models
13.More Advanced Features	14.Audio and Video Generation	15.Real world Aplication I Chatbot
	16.Real World Application II Image Producer	



1.Introduction	2. Fundamentals of ANN	**3.Getting to Know Transformers**
4.Text Generation, Fill the Mask and QEA	**5.Summarization, Translation and Chatbot**	6.Prompt Engineering for Image Generation
7.Getting to know GANs	8.Image Generation with GANs	9.Getting to Know Diffusers
10.Auto-Image Generation	11.Hight Quality and Conditional Image Generation	12.Other pre-treined Image Generatin Models
13.More Advanced Features	14.Audio and Video Generation	15.Real world Aplication I Chatbot
	16.Real World Application II Image Producer	



1.Introduction	2. Fundamentals of ANN	3.Getting to Know Transformers
4.Text Generation, Fill the Mask and QEA	5.Summarization, Translation and Chatbot	6.Prompt Engineering for Image Generation
7.Getting to know GANs	8.Image Generation with GANs	9.Getting to Know Diffusers
10.Auto-Image Generation	11.Hight Quality and Conditional Image Generation	12.Other pre-treined Image Generatin Models
13.More Advanced Features	14.Audio and Video Generation	15.Real world Aplication I Chatbot
	16.Real World Application II Image Producer	



1.Introduction

2. Fundamentals of ANN

3.Getting to Know Transformers

4.Text Generation, Fill the Mask and QEA

5.Summarization, Translation and Chatbot

6.Prompt Engineering for Image Generation

7.Getting to know GANs

8.Image Generation with GANs

9.Getting to Know Diffusers

10.Auto-Image Generation

11.Hight Quality and Conditional Image Generation

12.Other pre-treined Image Generatin Models

13.More Advanced Features

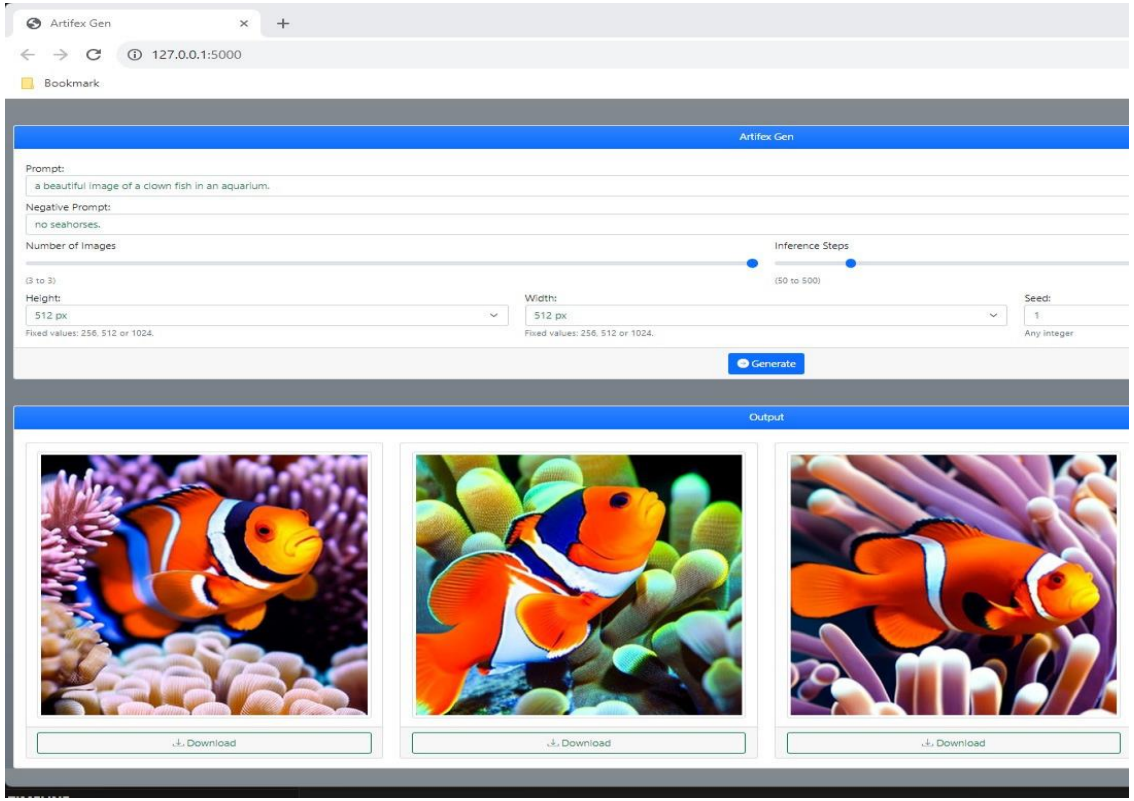
14.Audio and Video Generation

15.Real world Aplication I Chatbot

16.Real World Application II Image Producer

1.Introduction	2. Fundamentals of ANN	3.Getting to Know Transformers
4.Text Generation, Fill the Mask and QEA	5.Summarization, Translation and Chatbot	6.Prompt Engineering for Image Generation
7.Getting to know GANs	8.Image Generation with GANs	9.Getting to Know Diffusers
10.Auto-Image Generation	11.Hight Quality and Conditional Image Generation	12.Other pre-treined Image Generatin Models
13.More Advanced Features	**14.Audio and Video Generation**	15.Real world Aplication I Chatbot
	16.Real World Application II Image Producer	

1.Introduction	2. Fundamentals of ANN	3.Getting to Know Transformers
4.Text Generation, Fill the Mask and QEA	5.Summarization, Translation and Chatbot	6.Prompt Engineering for Image Generation
7.Getting to know GANs	8.Image Generation with GANs	9.Getting to Know Diffusers
10.Auto-Image Generation	11.Hight Quality and Conditional Image Generation	12.Other pre-treined Image Generatin Models
13.More Advanced Features	14.Audio and Video Generation	**15.Real world Aplication I Chatbot**
	16.Real World Application II Image Producer	



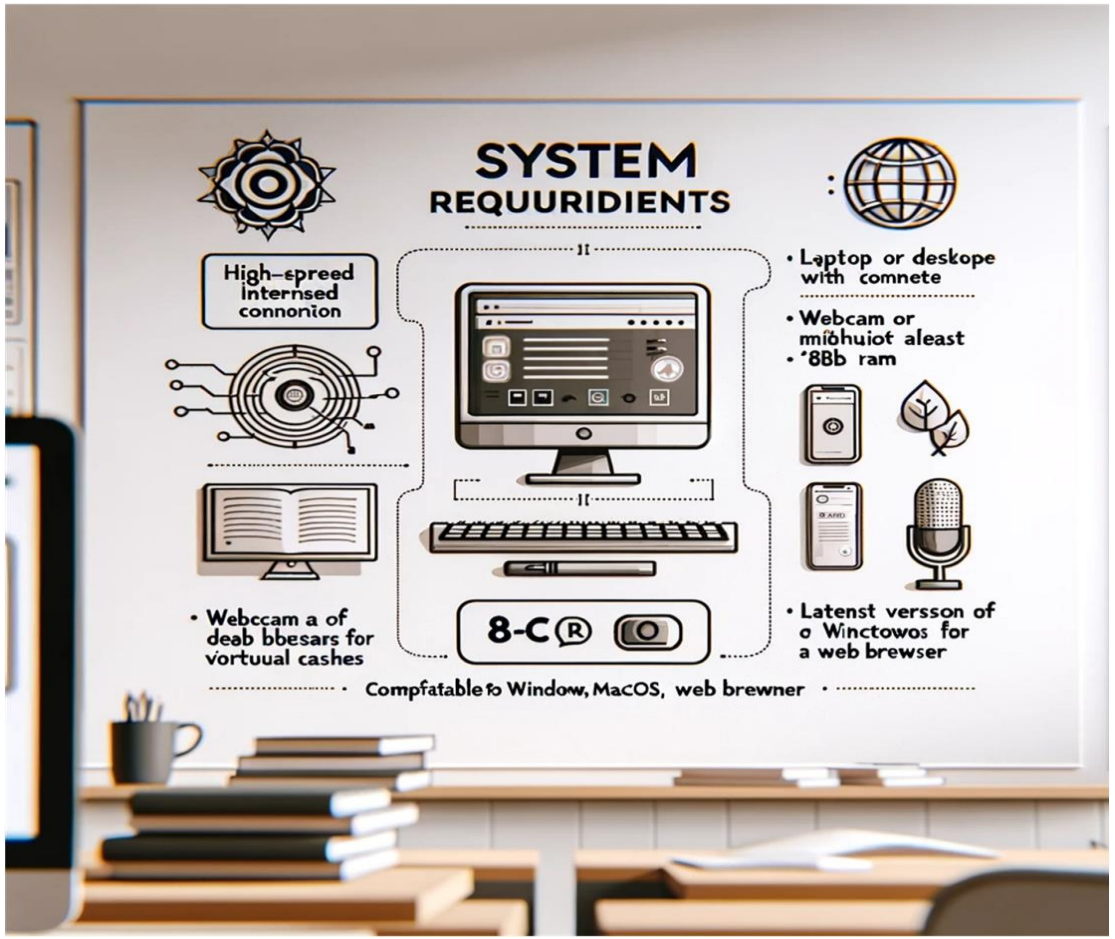
Requirements

- Google Colab
  - https://colab.research.google.com/
  - It’s Free*
  - All you need is a Google Account
  - Free GPU*

- You can Run the Examples on Your own Environment
  - Jupyter / VSCode etc
- GPU is recommended!

Real World Applications

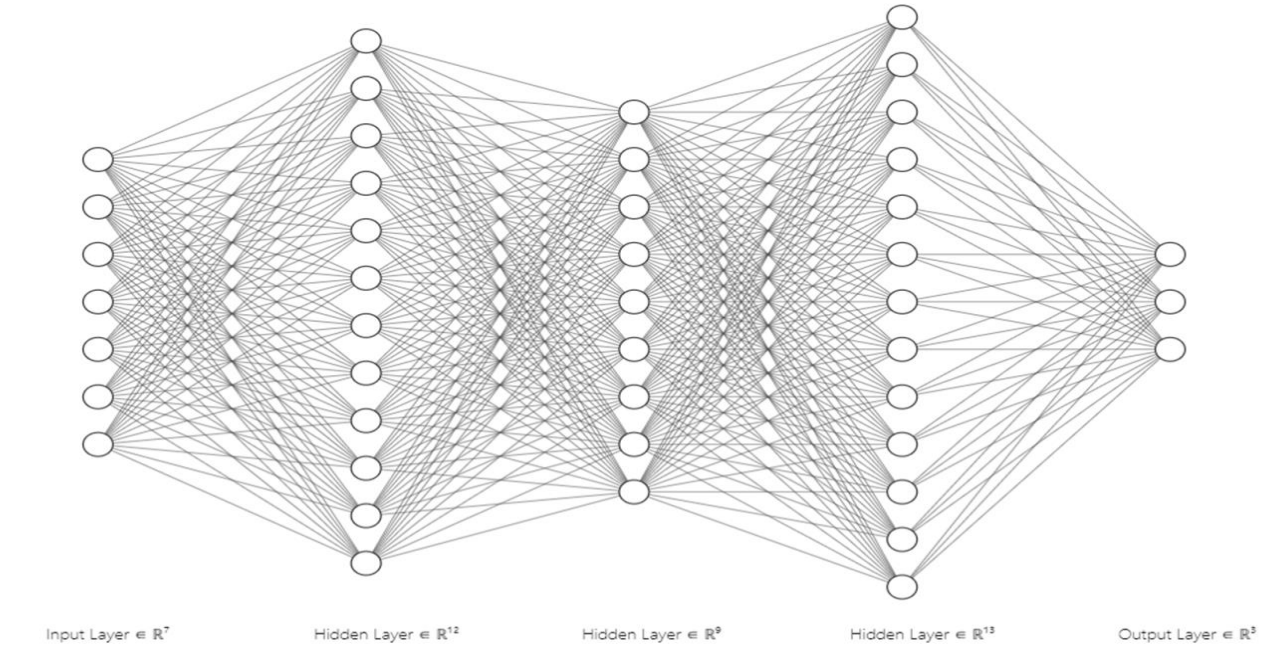
- Needed to run on your Local Machine*
- Recommend using VS Code



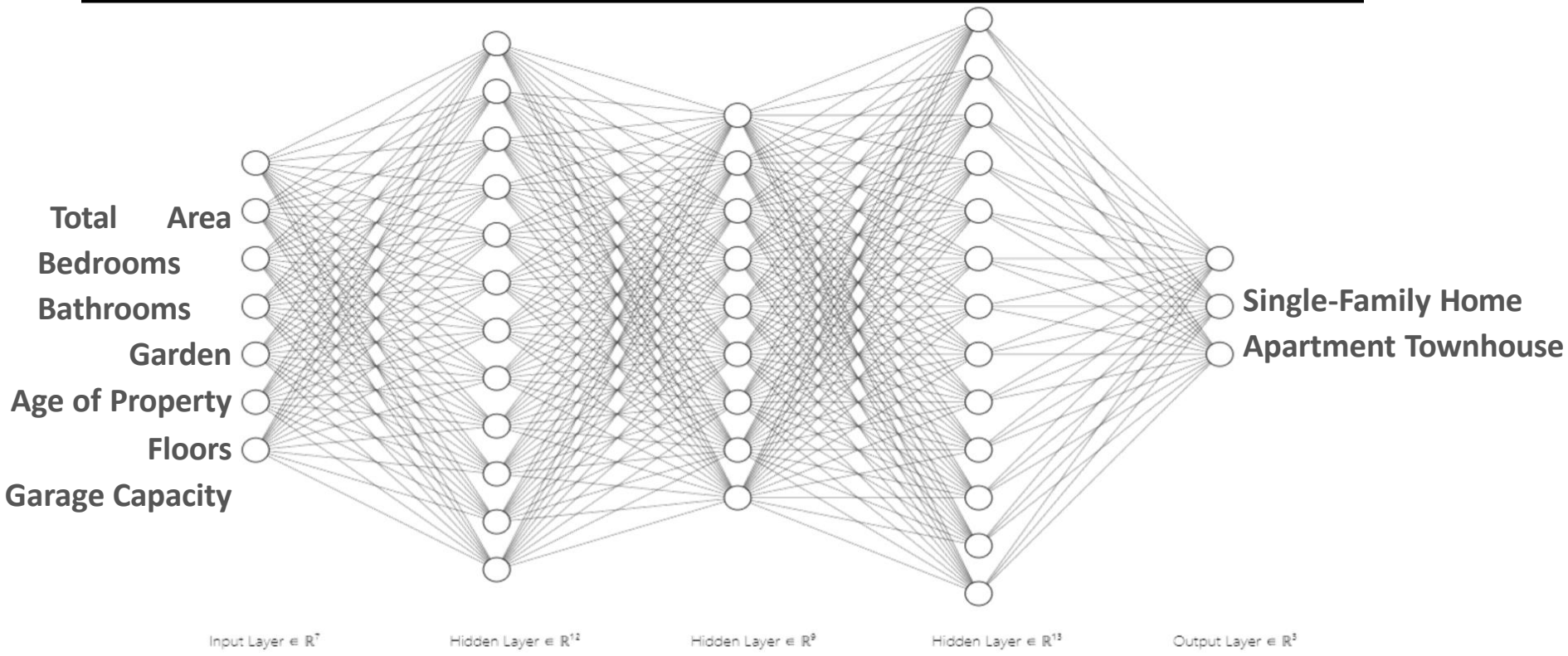
How to Run the Examples

- Option 1:
  - Download the source code from here (next lessons)
  - Upload to your Google Drive
  - From there, open on Google Colab
- Option 2:
  - Fork the repository or use the original Repo
  - From Colab you can open any notebook by the URL
- Option 3
  - Upload one by one

Main Components



What is the Objective of an ANN?

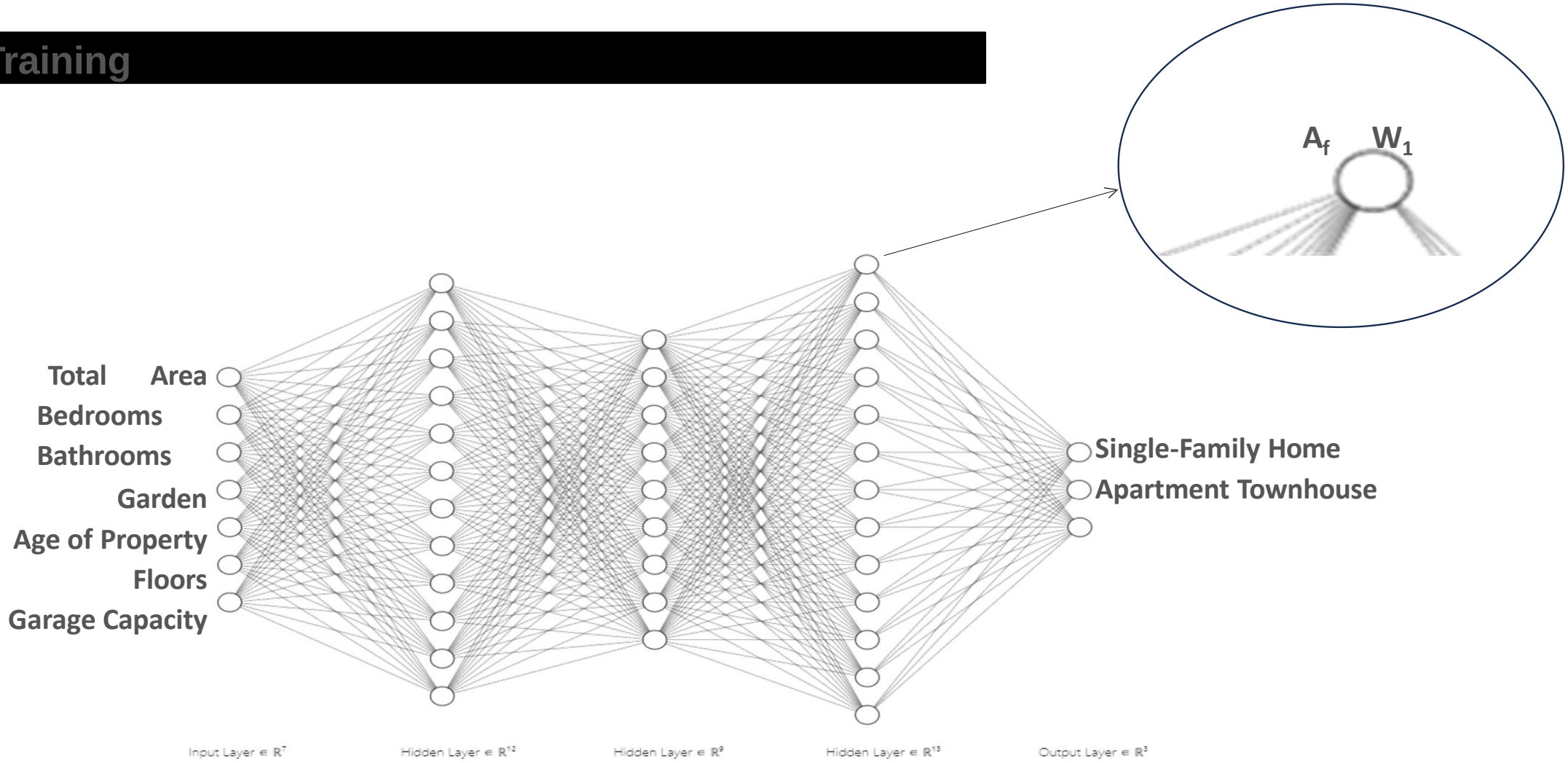


Given an input, produce the desired output.

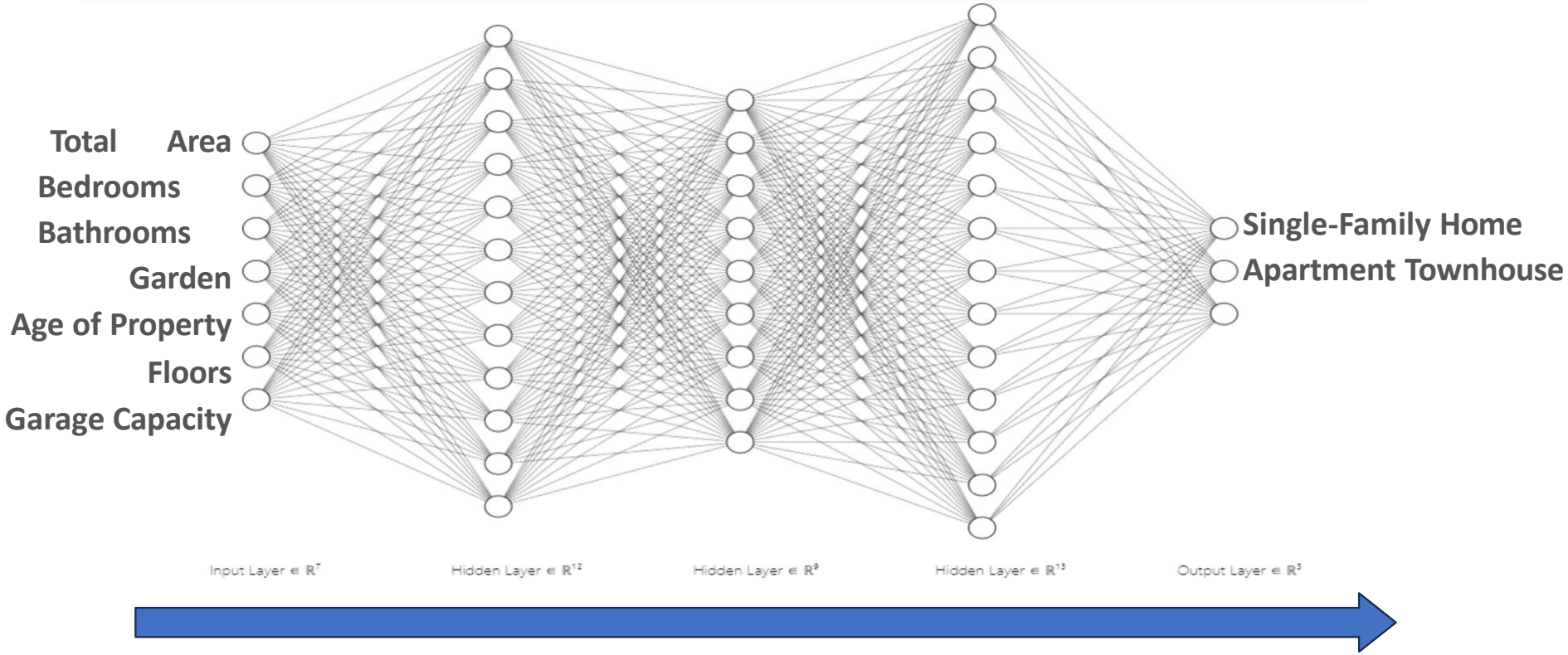
We Need to Train the Network!

Total Area (m^2)	Bedrooms	Bathrooms	Garden	Age (years)	Floors	Garage Capacity	Class
200	3	2	Yes	10	1	2	Single-Family Home
90	2	1	No	5	10	1	Apartment
150	3	3	No	7	3	1	Townhouse
...	...	...	...	...	...	...	...

Training

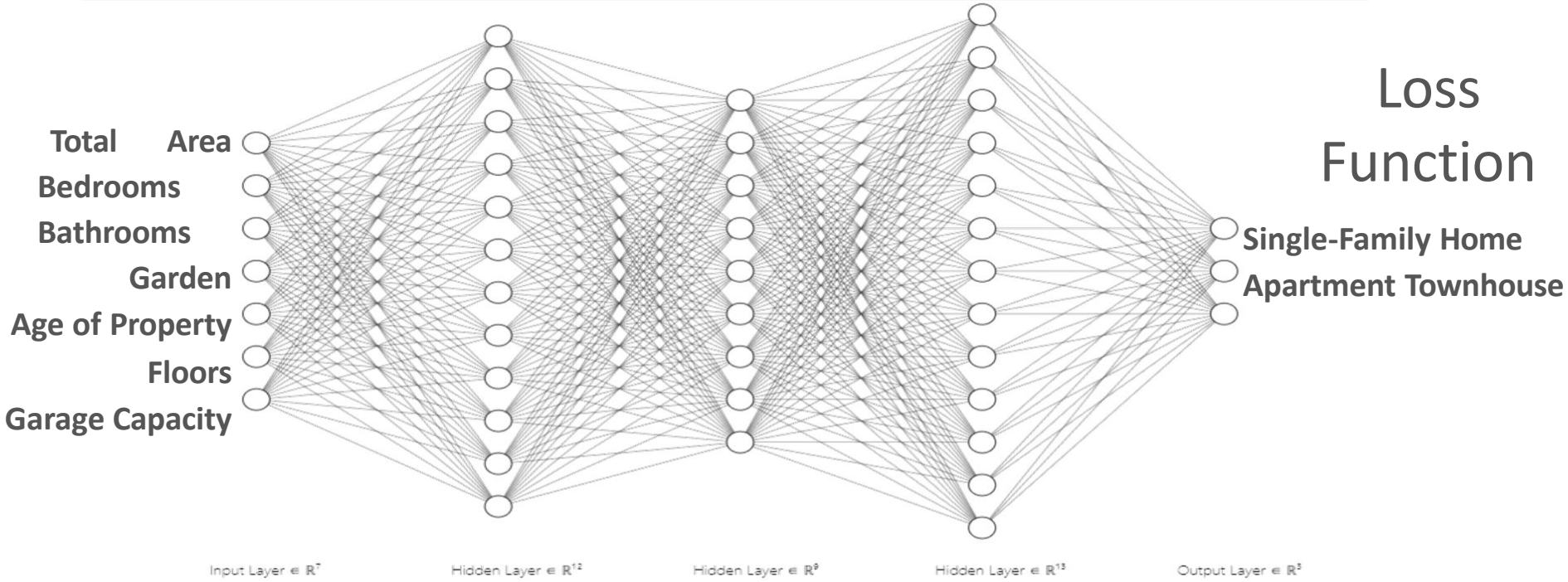


What Is The Objective of an ANN?



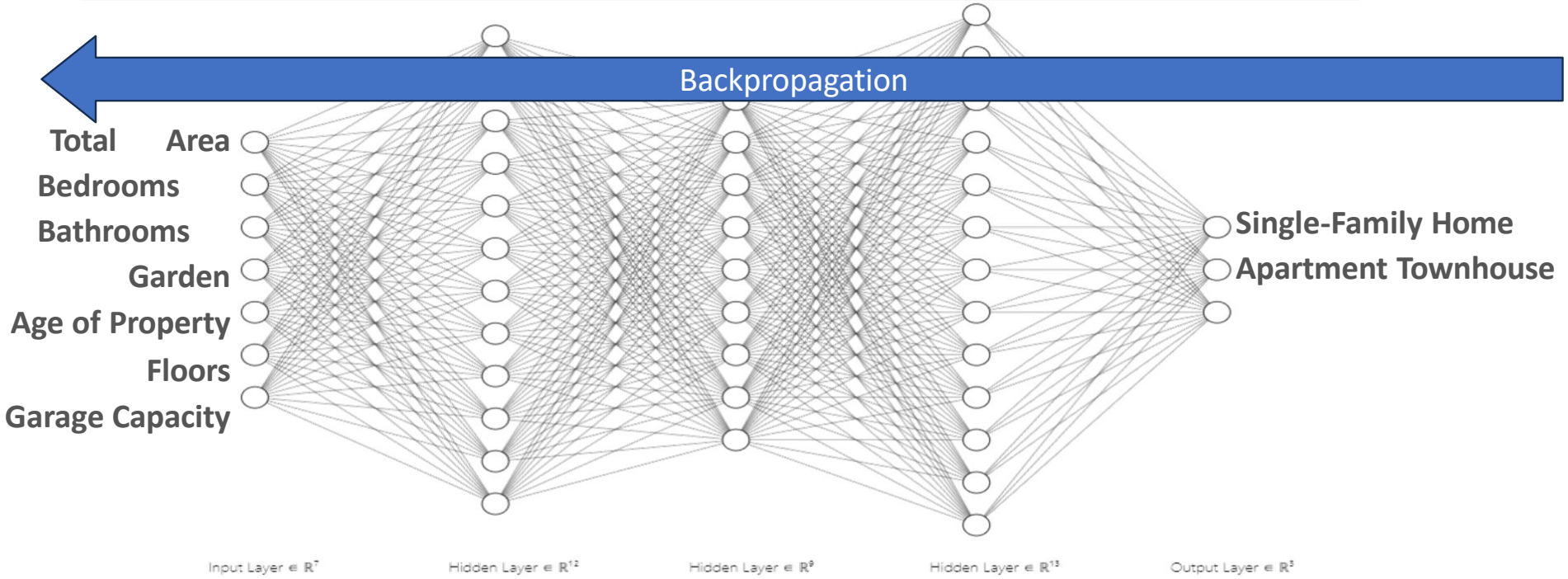
Data Flows Through the Network

What is The Objective of an ANN?

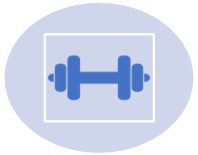


Loss Function

What is the Objective of an ANN?



The Process is Repeated Many Times…



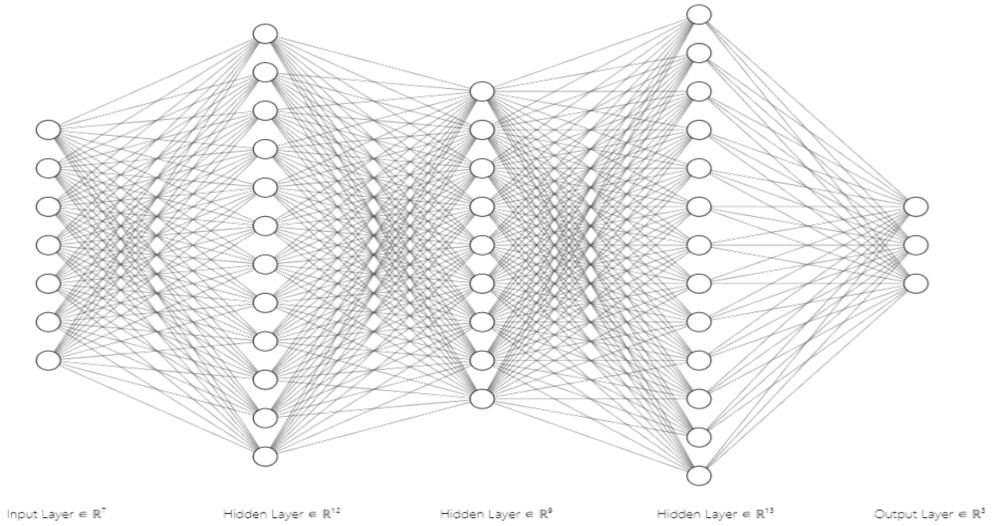
Weights are adjusted

The output tends to improve

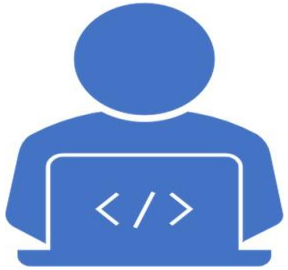
The loss decreases

ANN vs Deep Learning

- Deep Learning is an ANN with 3 or more hidden layers



Concepts



Epochs: The number of times all records will pass through the network.

Batch Size: After how many records pass through the network will the weights be adjusted?

Gradient Descent

Determines the direction and magnitude of the weight adjustments.

Gradient Descent aids in optimizing the training process.

Adam is an advanced implementation of

Learning Rate

The learning rate determines the step size at each iteration while moving towards a minimum in the loss function.

It influences how quickly or slowly a neural network updates its weights during training.

A smaller learning rate might converge slowly, while a larger one might overshoot the optimal solution.

Overfitting

- Overfitting occurs when a model learns the training data too closely, including its noise and outliers, causing it to perform poorly on unseen or new data.
- It essentially means the model is too complex and captures patterns that don't generalize well beyond the training dataset.

- Regularization:
  - Early Stopping
  - Data Augmentation
  - L1 and L2
  - Dropout

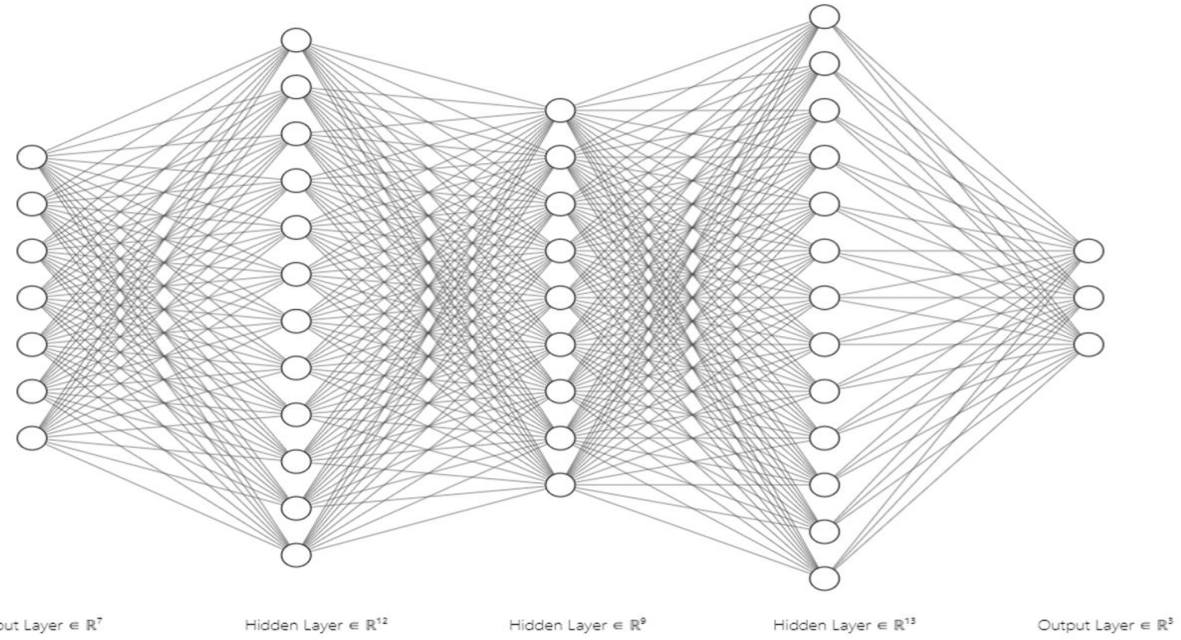
Activation and Loss Functions

- There are many different activation functions:
  - Threshold
  - Sigmoid
  - ReLU
  - Tanh
- There are many different loss functions
  - MSE RMSE
  - Cross-Entropy

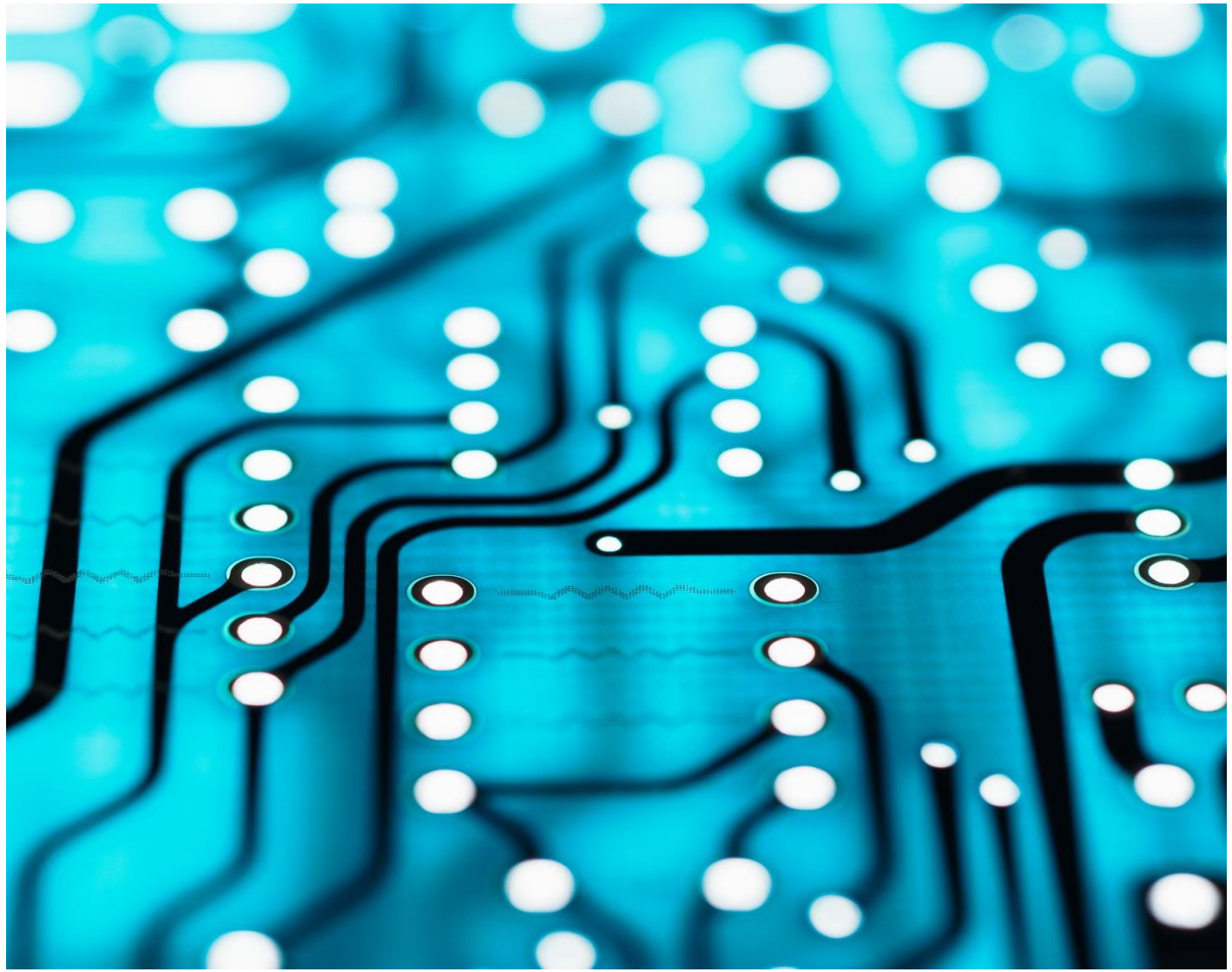
HyperParameter

- A hyperparameter is a parameter of the ANN that you set before training or using a pre-trained model. It influences the output as well as the time the ANN takes to run.

Model


Input Layer ∈ $\mathbb{R}^7$
Hidden Layer ∈ $\mathbb{R}^{12}$
Hidden Layer ∈ $\mathbb{R}^9$
Hidden Layer ∈ $\mathbb{R}^{13}$
Output Layer ∈ $\mathbb{R}^3$

Learning Process



Generative AI

- You can create your own model from scratch
- You can fine-tune a model that has already been trained
- You can use a model that was pre-trained

Transformers

- It emerged in 2017, the article 'Attention is All You Need' by Vaswani et al.

Transformers

Revolutionary Model

Used in GPT-3/4, T5, and BERT

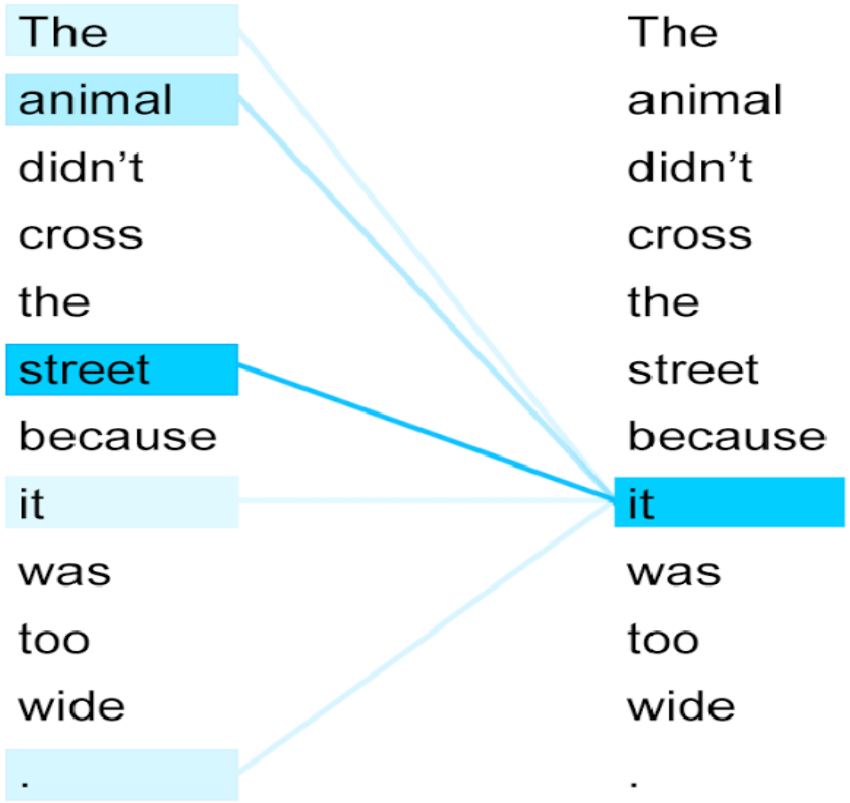
Doesn't use recurrence like RNN

Based on attention mechanism

Encoder/Decoder architecture

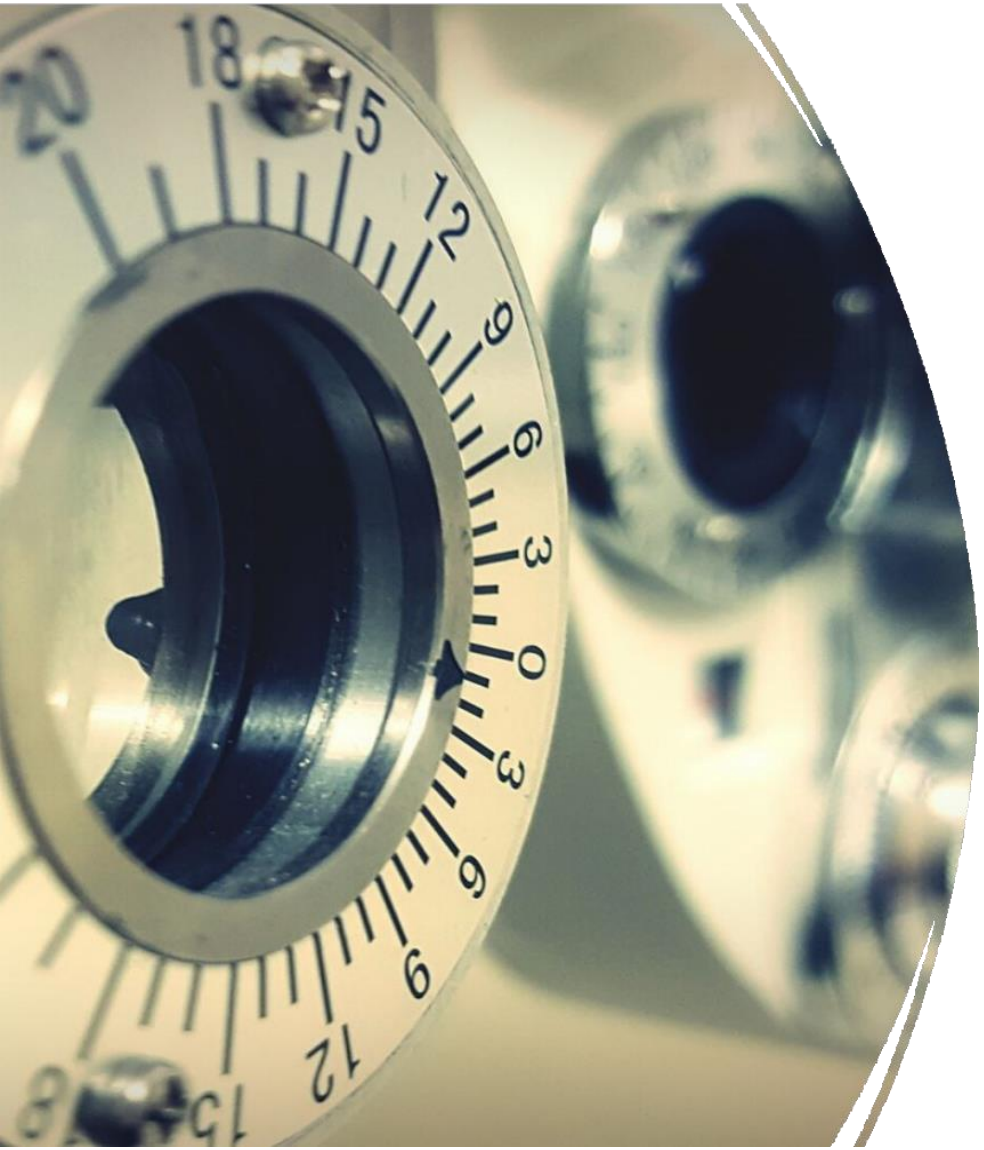
Attention / Self Attention

- Attention learns the contextual relationship between words.



NLP (Natural Language Processing)

- BERT (Bidirectional Encoder Representations from Transformers):
- GPT (Generative Pre-trained Transformer)
- T5 (Text-To-Text Transfer Transformer)
- XLNet
- RoBERTa
- ALBERT (A Lite BERT)
- DistilBERT
- ELECTRA



- ViT (Vision Transformer)
- DETR (DEtection Transformer)

More

- **CLIP (Contrastive Language–Image Pretraining)**
- **DALL-E**
- **MUM (Multitask Unified Model)**
- **Switch Transformer**

Hugging Face

- Transformers Library
- Diffusers
- Hosting of Models
- Much More

Pipeline

- Pipeline method:
  Allows for creating a pipeline for an NLP task
- Main tasks:
  1. Text Generation
  2. Fill the Mask
  3. Q&A
  4. Summarization
  5. Translation
  6. Chat
- Model: Can be a pre-trained model.

Text Generation

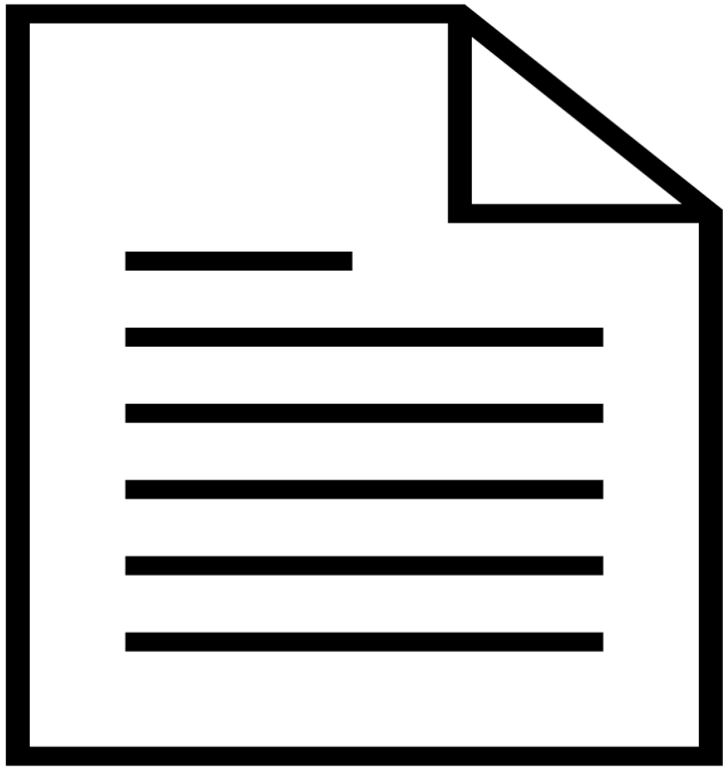
- Given a prompt, the model should complete it...
- Examples:
  - In a world where humans and dragons coexist,
  - The theory of relativity is based on the idea that,
  - By 2050, cities will be designed with,
  - The future of artificial intelligence will lead to,
  - If cats could talk, they would probably say,

NIQ

Models

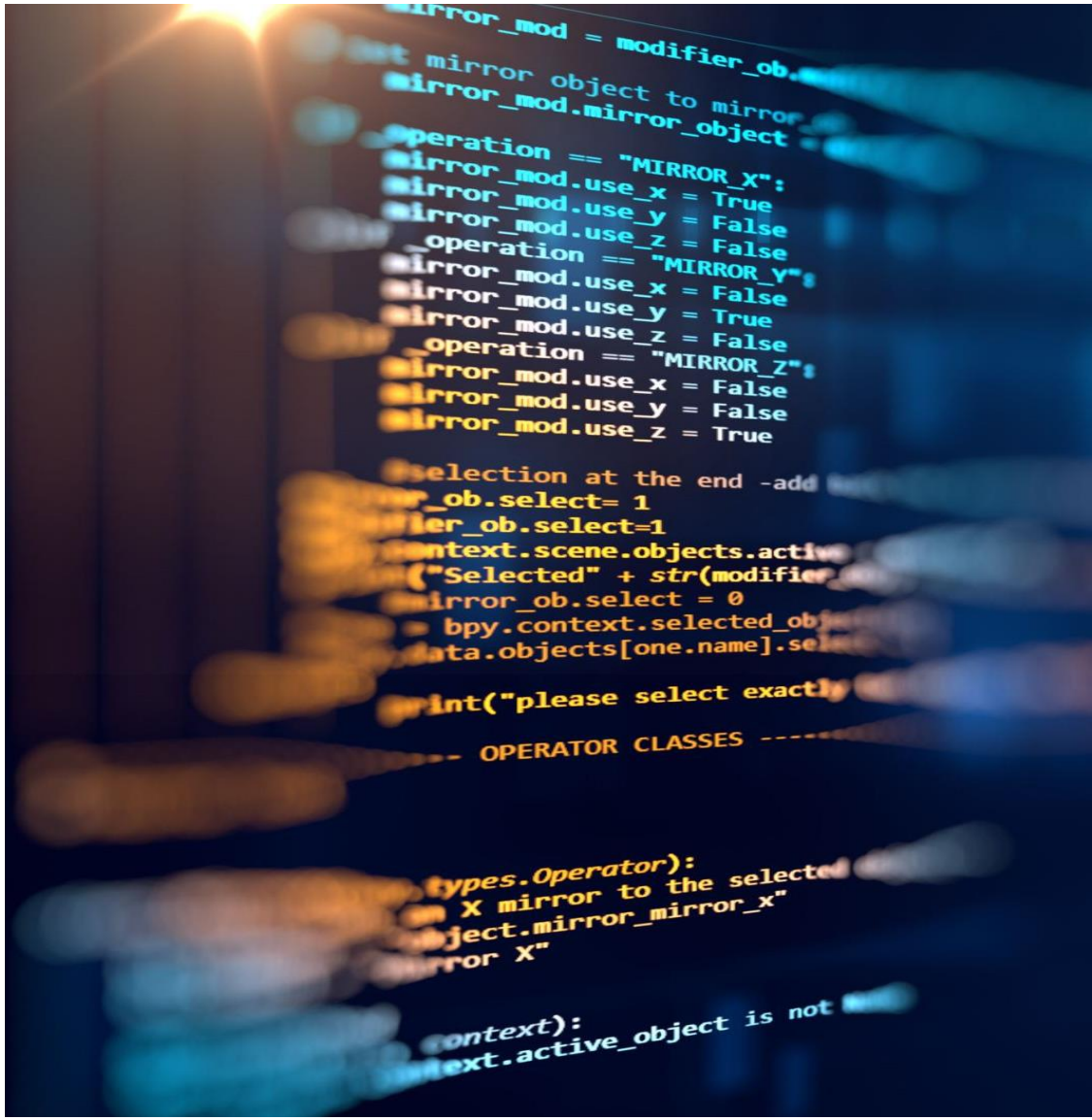
- GPT (gpt2-médium)
- Bert
- T5 (Text-to-Text Transfer Transformer)
- XLNet
- CTRL (Conditional Transformer Language Model)
- RoBERTa
- Transformer-XL

Hyperparameters



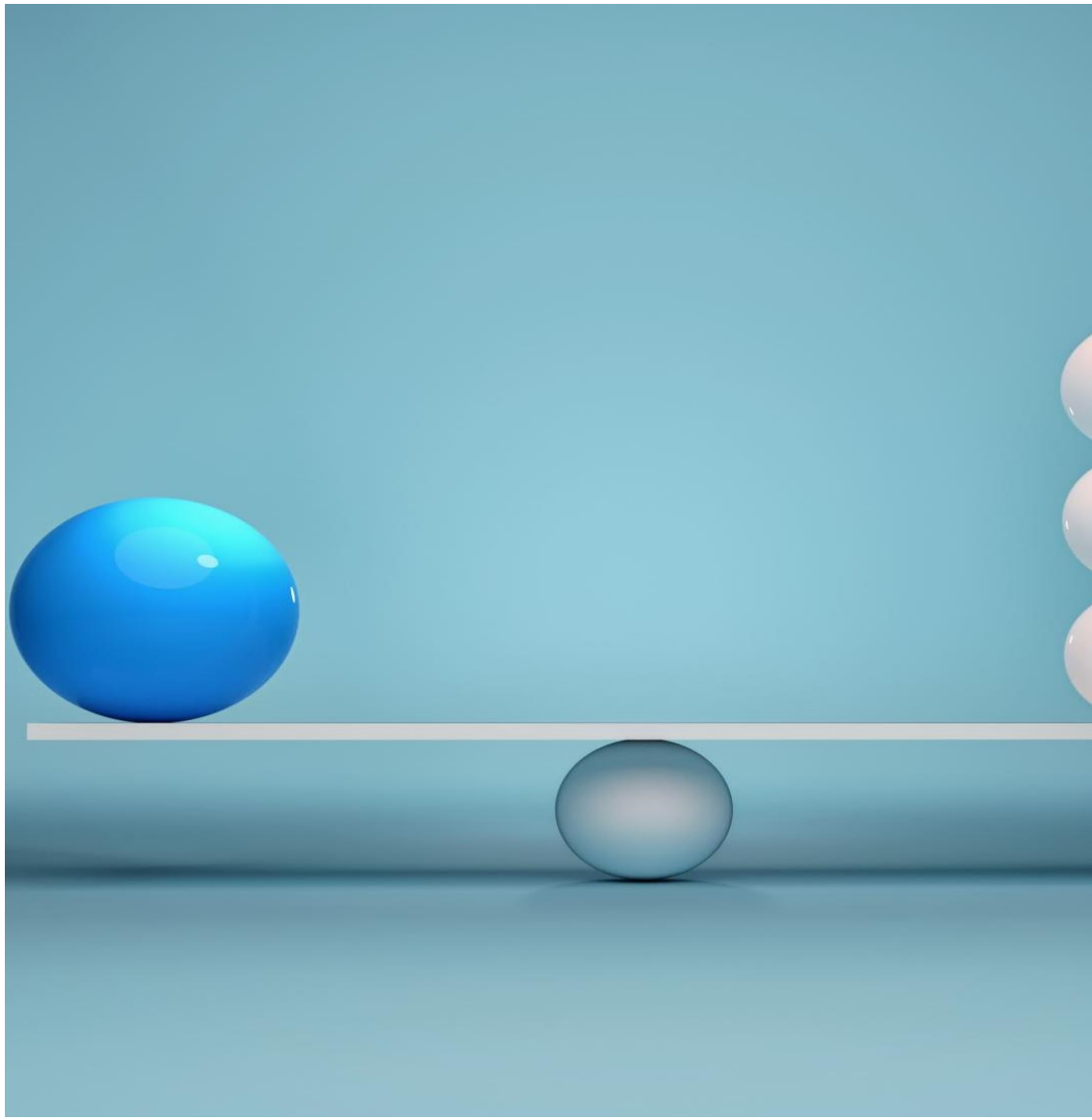
Hyper Parameters

- max_length: The maximum length of the output text will be at most.



length_penalty

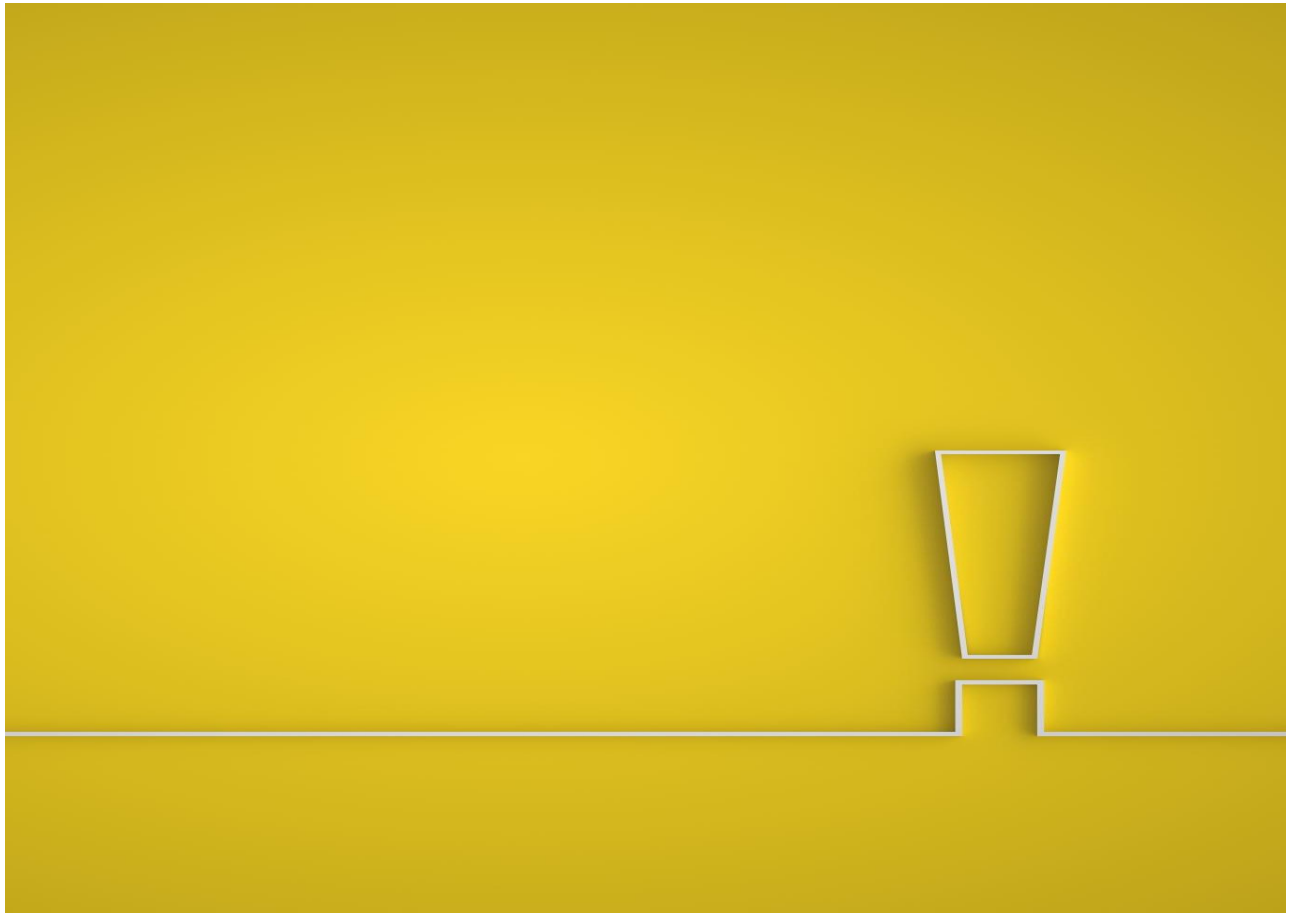
- Values less than 1.0 will favor shorter lengths.
- Values greater than 1.0 will favor longer lengths.



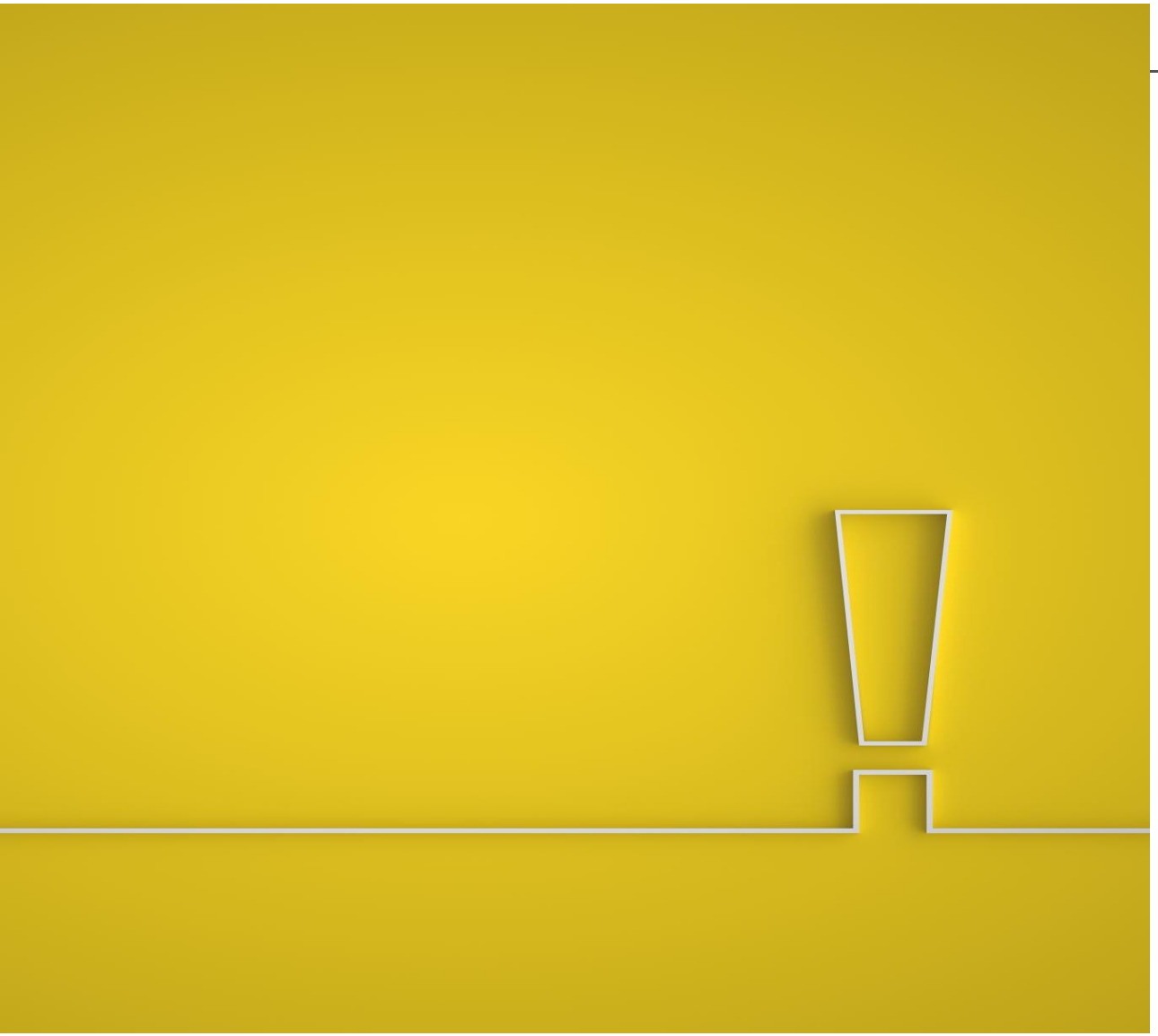
num_return_sequences

- Number of results, or completions, to be created.
- To use a value greater than 1, do_sample must be set to True.

early_stopping



- Stop generation once the end-of-sequence token <EOS> is found.
- Prevent the model from generating text beyond a logical ending.



_token_id

- Token used for padding.
- Padding is needed so that sequences of text have the same length.
- Example: [PAD].

Hyperparameters Related to Variability

		More Variability	Less Variability
do_sample	How the model chooses the next word	True	False
temperature	Control the randomness of predictions	Higher than 1.0	Between 0.0 to 1.0
top_k	Considers only the top K words when deciding on the next word	Larger values	Smaller values
top_p	Samples words from a cumulative probability that exceeds threshold p	Closer to 1	Closer to 0
repetition_penalty	Discourage the model from producing repetitive output	Higher than 1.0	Between 0.0 to 1.0

Real World Applications

- Content Creation
- Chatbots
- Education
- Programming
- Research
- Entertainment

- The goal is to complete the text marked with [MASK]
  - The sun rises in the [MASK].
  - An apple a day keeps the [MASK] away.
  - Rome wasn't built in a [MASK].
  - Every cloud has a [MASK] lining.

- Text Completion and Auto-suggest
- Text Correction
- Named Entity Recognition (NER)
- Language Learning
- Search Engines (complete the search)
- Programming
- Data Analysis (Predict missing values)

- BERT (bert-base-uncased)
- RoBERTa
- DistilBERT
- GPT-2

Hyperparameters

- top_k
- Number of predictions to be returned

Questions and Answers (Q&A):

- The model's goal is to answer a question given a specific context.
- Context: The necessary background for the Answer:
  - A document
  - A text
  - The preceding dialog in a conversation

Models

- distilbert-base-cased-distilled-squad
- pierreguillou/bert-base-cased-squad-v1.1-portuguese
- BERT
- RoBERTa
- DistilBERT
- ALBERT

Toy Examples

1.Context: The Great Wall of China was built to protect Chinese states and empires from invasions and stretches over 13,000 miles.

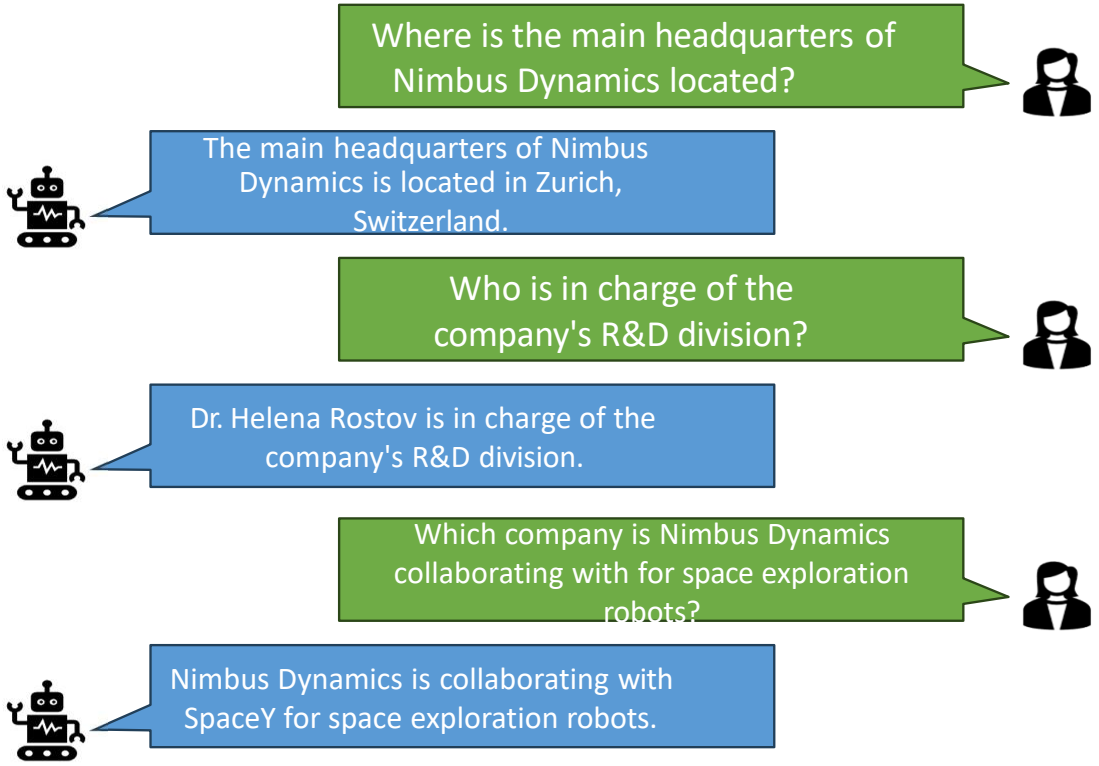
Question 1: Why was the Great Wall of China built?

Question 2: How long is the Great Wall of China?

- **2.Context:** Beethoven became deaf in his later years but continued to compose music, including his famous Ninth Symphony.
- **Question 2:** Did he compose the Ninth Symphony before or after becoming deaf?

Real World Application

Nimbus Dynamics is a multinational corporation established in 1998, renowned for its advanced robotic solutions. With its main headquarters located in Zurich, Switzerland, the company also has regional offices in Tokyo, New York, and Sydney. Over the past two decades, Nimbus Dynamics has expanded its product line to include not only industrial robots but also household robotic assistants and specialized medical robots for surgery. The company's R&D division, spearheaded by Dr. Helena Rostov, has received multiple awards for its innovative designs and sustainable energy solutions. Currently, the company boasts an employee base of approximately 15,000 individuals globally and has an annual revenue of 10 billion USD. In a recent press release, Nimbus Dynamics announced its collaboration with SpaceY, intending to develop robots for space exploration.



Fill The Ma

- The goal is to complete the text marked with [MASK]
  - `The sun rises in the [MASK].`
  - `An apple a day keeps the [MASK] away.`
  - `Rome wasn't built in a [MASK].`
  - `Every cloud has a [MASK] lining.`

- Text Completion and Auto-suggest
- Text Correction
- Named Entity Recognition (NER)
- Language Learning
- Search Engines (complete the search)
- Programming
- Data Analysis (Predict missing values)

- BERT (bert-base-uncased)
- RoBERTa
- DistilBERT
- GPT-2

Hyperparameters

- top_k
- Number of predictions to be returned

- Given a text as input, produces a summary as output

- facebook/bart-large-cnn
- neuralmind/bert-base-portuguese-cased
- BERT
- T5
- PEGASUS
- GPT-2

Translation

- Converting a text from one language (source language) to another (target language) while preserving the meaning and context of the original content.

Model

- Helsinki-NLP/opus-mt
- Series of translation models
- Multi-Language

Prompt Engineering

- "Prompt" is an instruction or stimulus provided to an AI to generate a response.
- In Chat GPT, you input a prompt (question) and it produces an answer.
- Prompt engineering" is a technique that involves creating Prompts (commands)
- It's an approach used to refine and improve the output of an AI model by adjusting the instructions given to the model. The idea is to find the correct formulation of the prompt to obtain the desired response from the model.

Rules for good Prompt Engineering

Clarity
Specificity
Context
Theme or Style
Model Limitations
Direct Instructions
Avoid Contradictions

Good Examples

- ✓ Clarity
- ✓ Specificity
- ✓ Context
- ✓ Theme or Style
- ✓ Model Limitations
- ✓ Direct Instructions
- ✓ Avoid Contradictions

Bad Examples

- **Draw Something**
- **Create an image that makes me happy**
- **Illustrate a sunny and rainy day at the beach.**

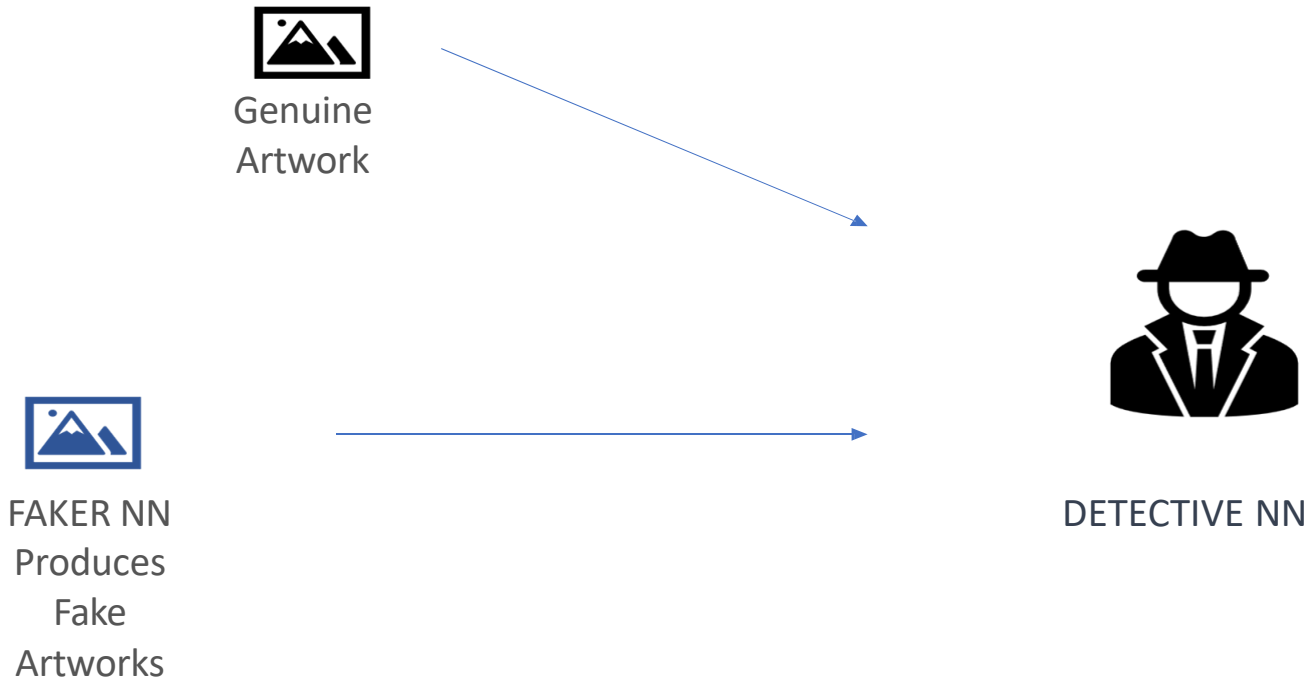
Negative Prompts

- What should not be included
- Avoid or limit the generation of content that could be interpreted as inappropriate, offensive, harmful, or unwanted

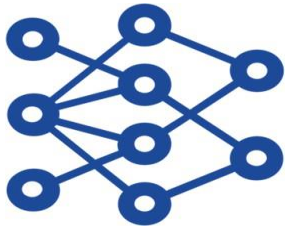
GANs

- Generative Adversarial Network
- GANs are a type of machine learning model composed of two neural networks: the Generator (G) and the Discriminator (D).
- The objective of G: Create data (such as images) that appear real.
- The objective of D: Distinguish between real data and data generated by G.
- Real data are needed to train a GAN.

GANs



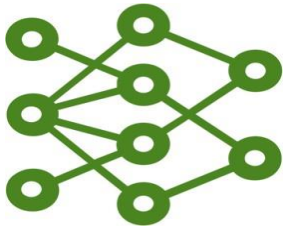
GAN



Generator **(G)**

Objective: Try to deceive D, producing images that D thinks are real

	Loss
High	Not deceiving D
Low	Is deceiving D



Discriminator **(D)**

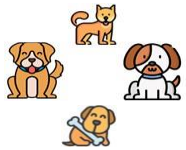
Objective: Be able to distinguish real images from fake ones

	Loss
High	Cannot Distinguish
Low	Can Distinguish

Training the Discriminator D

Objective: Be able to distinguish real images from fake ones

Step 1



Discriminaor **(D)**

Loss

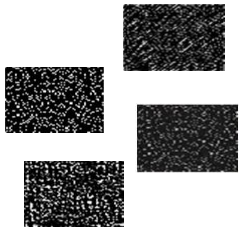
Real Image

Step 2

0.7452, 0.2378, 0.4927, 0.6341
0.2345, 0.4897, 0.1236, 0.7549
0.6431, 0.2875, 0.9652, 0.5164
0.8762, 0.1384, 0.4598, 0.6427

Generator **(G)**

Step 3



Discriminator **(D)**

Loss

Fake Images

Step 4

Loss

Real Images

+

Loss

Fake Images

Total Loss D

Training Loss

Step 5

BackPropagation

Total Loss D

Discriminator **(D)**

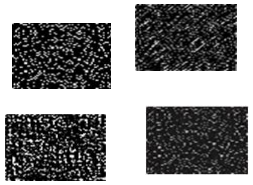
Training the Generator G

Objective: Try to deceive D, producing fake images that D thinks are real

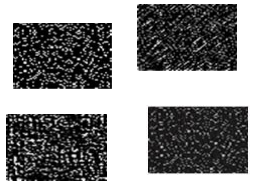
Step 1

0.7452, 0.2378, 0.4927, 0.6341
0.2345, 0.4897, 0.1236, 0.7549
0.6431, 0.2875, 0.9652, 0.5164
0.8762, 0.1384, 0.4598, 0.6427

Generator **(G)**



Step 2



Discriminator **(D)**

Probability of being real

Step 3

Probability of being real

Generator **(G)**

Loss

Training Loss

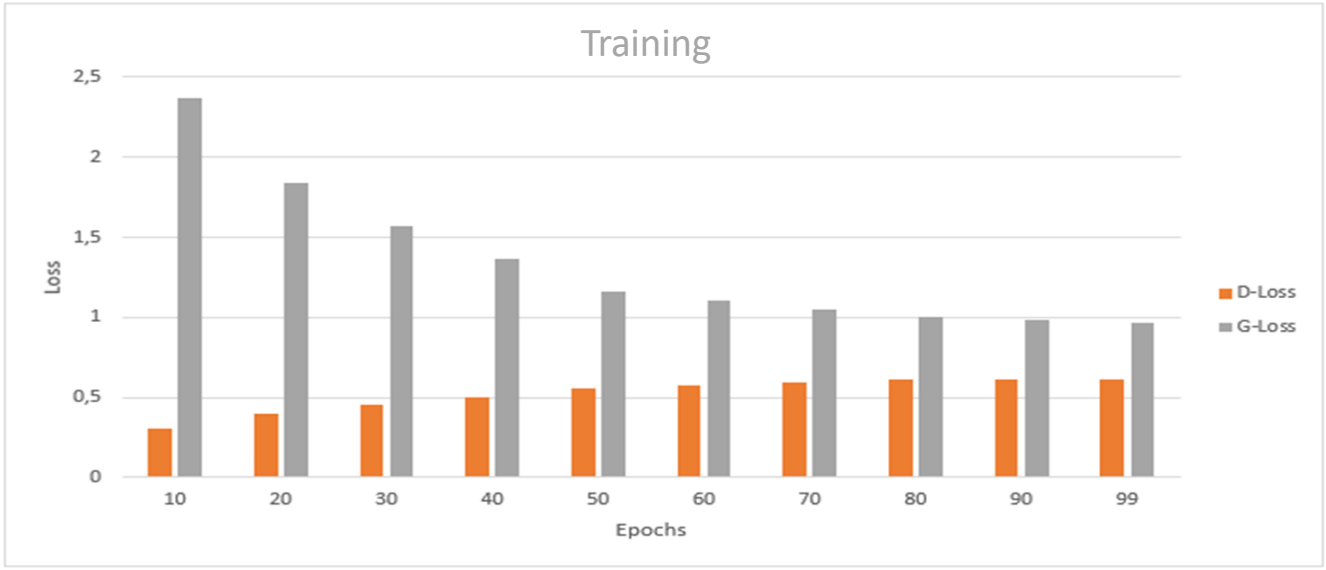
Step 4

BackPropagation Loss G

Generator **(G)**

Many Epochs Later...

- The discriminator will be better at distinguishing fake images from real ones.
- The generator will be better at producing images that resemble real ones.

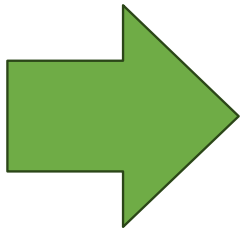


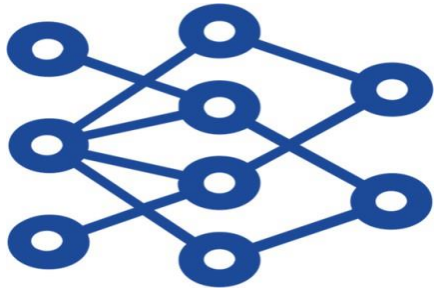
End of The Training

- Epochs
- Observe the Generator and Discriminator Loss
  - Discriminator Loss: If the value is too low, it indicates that it is very good at distinguishing fake images from real ones, which means that the Generator is not managing to deceive it.
  - Generator Loss: If the value is too low, it's a good sign, as the discriminator is having difficulty differentiating real images from fake ones.
- Visual Inspection of Image Quality
- Visual Inspection of Image Variety

Use of the Model

0.7452, 0.2378, 0.4927, 0.6341
0.2345, 0.4897, 0.1236, 0.7549
0.6431, 0.2875, 0.9652, 0.5164
0.8762, 0.1384, 0.4598, 0.6427





Generator **(G)**

Training a GAN

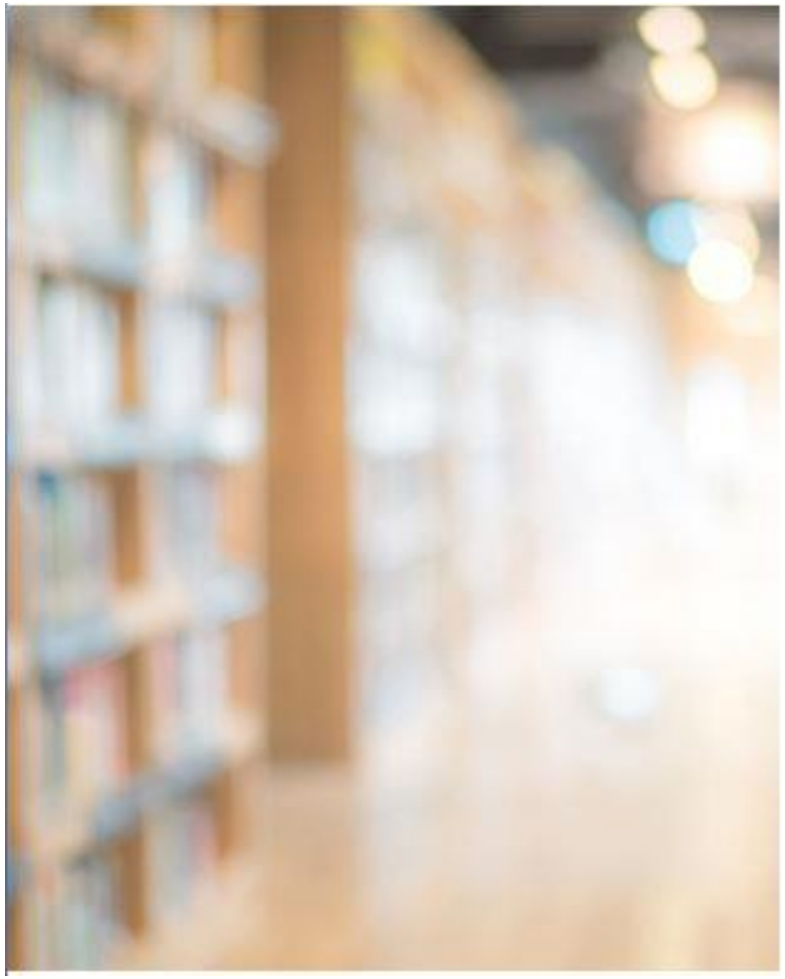
- Complex Process
- High Costs
- Requires Specialized Hardware
- Trial/Error

Final Considerations

Generator: It should learn to create the image, but it will never see a real training image.

Periodically, during training, you might want to save generated images to visually inspect G's progression.

Deep Convolutional Generative Adversarial Networks (DCGANs):



- They use convolutional layers in the generator and discriminator, which helps them handle image data more effectively.
- Can generate high-quality images
- Fundamental architecture for many subsequent GAN variations

DCGANs

- Default model trained on Fashion-MNIST
- Contains 70,000 in 10 categories (T-shirt, Trouser etc.)

PGAN: Progressive Growing of Generative Adversarial Networks

- Type of GAN introduced by NVIDIA in a 2017
- PGANs introduce a training methodology that is designed to stabilize the training of the generator and discriminator by progressively growing both networks, starting from a low resolution and gradually adding layers to increase the resolution during training.
- Utilized for generating realistic and high-quality images, especially faces

PGAN

- Pre-trained model celebAHQ-512
- Trained with CelebA dataset

BigGAN (Big Generative Adversarial Network)

- Is known for its capacity to generate high-quality and high-resolution images.
- Developed by researchers at DeepMind
- Implements a scaled-up version of the GAN architecture, with an increased model size (more parameters), and a greater batch size during training.

BigGAN

- Trained on ImageNet data: https://www.image-net.org/
- Can generate imagens from many different subjects
- 1000 classes
- 128 x 128 (other models available)




Diffusers

- Diffusion
  - Randon Image
  - Applies a Series of Gradual Transformation
  - Increase the Size

Latent Diffusion model created by CompVis, StabilityAI, and LAION

https://stability.ai/

Pipeline

- Diffusion library create by Huggin Face
- Components used to create an image:
  - Model
  - Scheduler
  - Transformer
  - Image Processor

Scheduler

- Controls the diffusion process,

Transformer

- Converts the model's output

Processor

- Generates the image, audio, or video.

Checkpoint

- Stores the state of a model

Pipeline

- Groups all the necessary components to generate the image:
  - Model
  - Scheduler
  - Transformer
  - Image Processor
- DiffusionPipeline is the Base class.

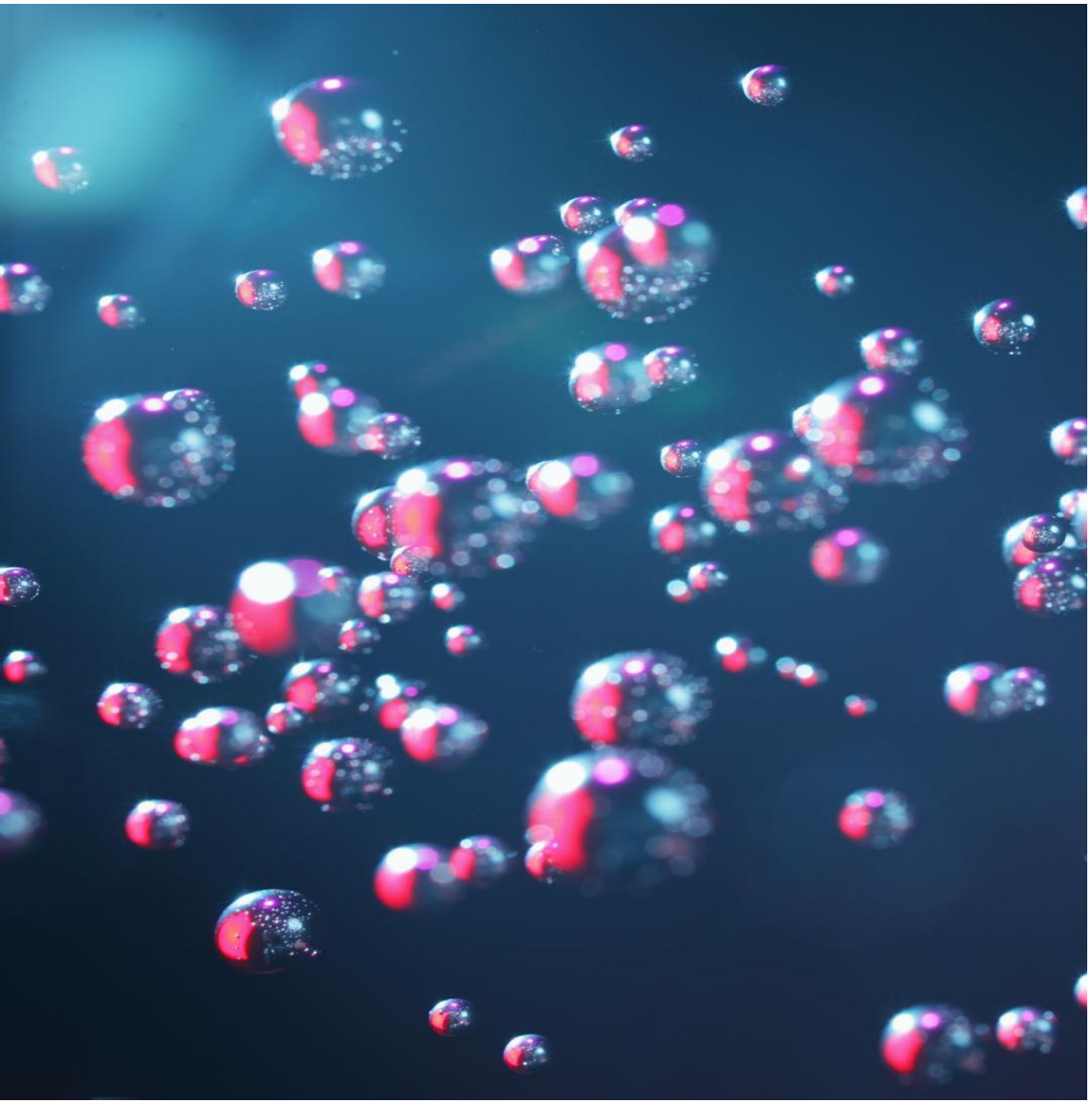
Pipeline	Description
Stable Diffusion	The most popular pipeline, used to generate images, text, music, and other types of creative data
ControlNet	Used to generate high-quality images..
DDIM	Used to generate images in a specific style.

Tasks

- Types of Tasks the Pipeline Will Execute:
  - **Unconditional Image Generation**: generates an image.
  - **Text-Guided Image Generation**: generates an image based on a prompt.
  - **Text-Guided Image-to-Image Translation**: adapts an image based on a text prompt.
  - **Text-Guided Image Inpainting**: fills in a part of the image with a mask, based on a text prompt.
  - **Text-Guided Depth-to-Image Translation**: adapts part of an image based on a text prompt while preserving its structure

Autopipeline

- Simplifies the process
- Automatically loads checkpoints
- Supports
  - Text-to-image
  - Image-to-image
  - Inpanting



Supported Models

- ControlNet
- DeepFloyd IF
- Kandinsky
- Kandinsky 2.2
- Stable Diffusion
- Stable Diffusion XL (SDXL)

Methods

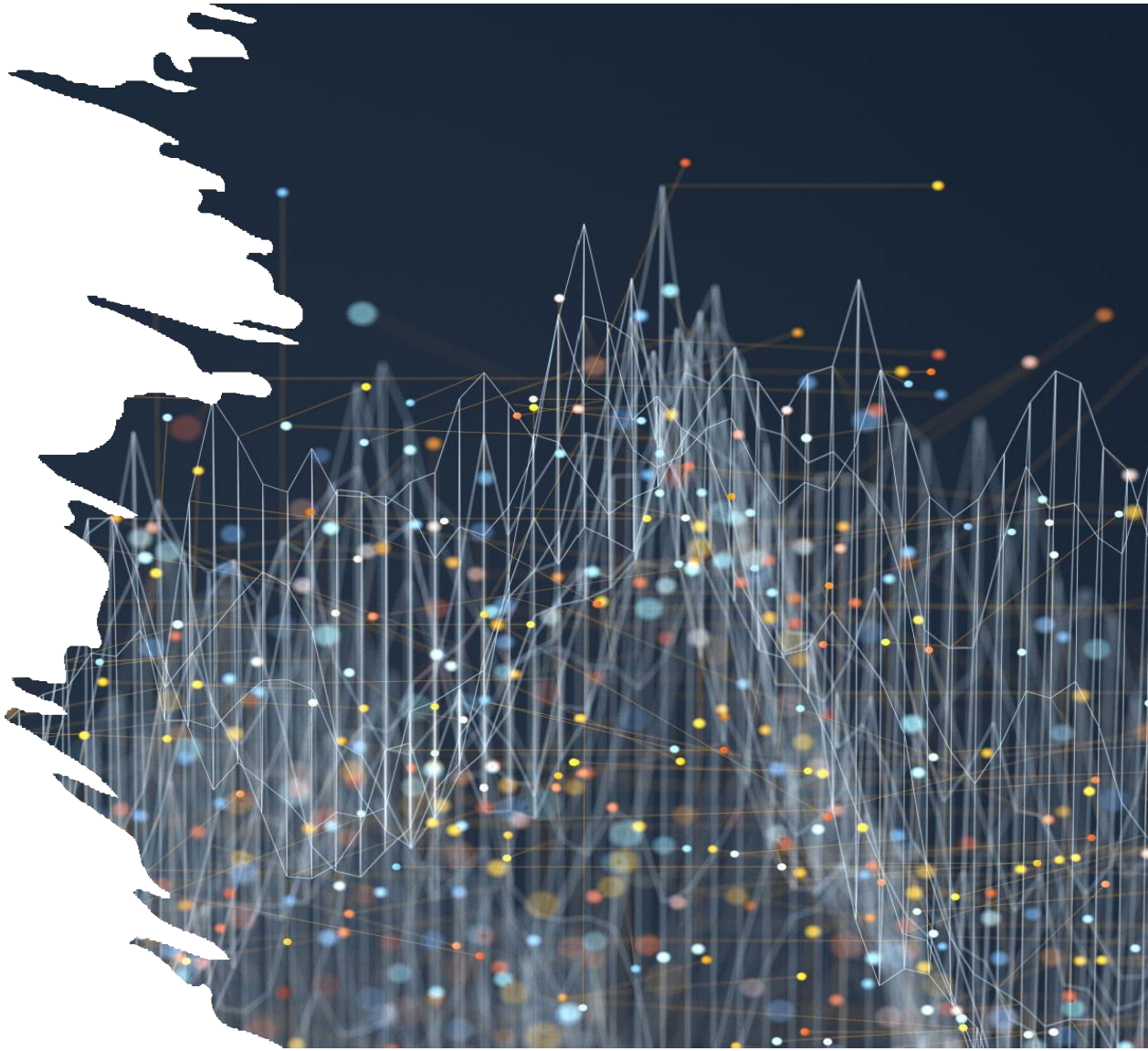
AutoPipelineForText2Image

AutoPipelineForImage2Image

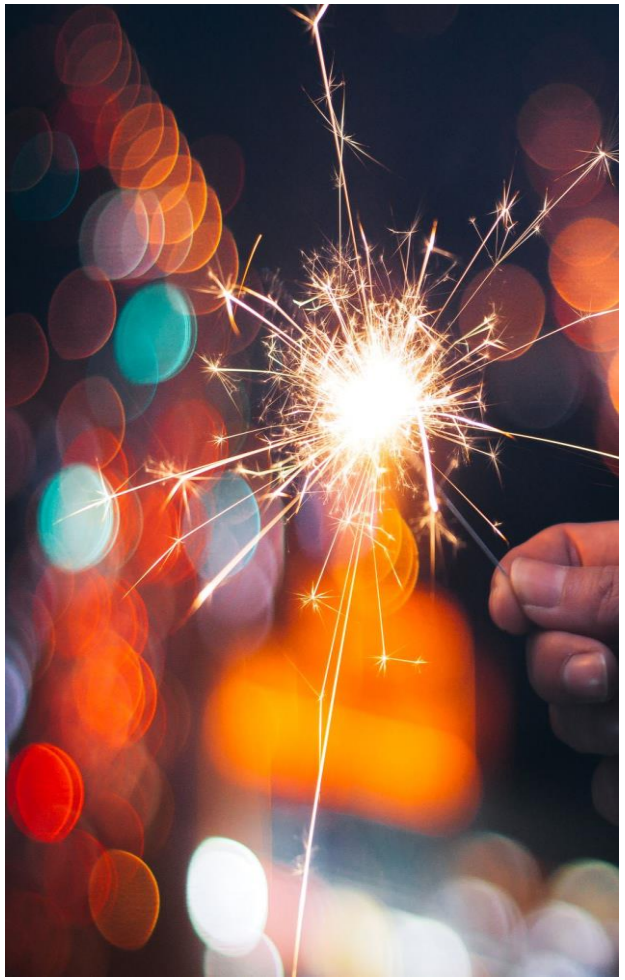
AutoPipelineForInpainting

Unconditional

- Generate a Random Image Based on the Model
- Few hyperparameters



Conditional



- "A serene sunset over a mountain range.“
- "Visual representation of peace and tranquility."
- "The emotion of joy portrayed as an abstract landscape."
- "A chaotic representation of a busy mind."
- "A child's first step, with the parent's encouraging expression in the background."
- "An astronaut playing a guitar on the moon."
- "A cyberpunk detective with neon tattoos and futuristic eyewear."

HyperParameters

- `negative_prompt`
- `num_images_per_prompt`
- `num_inference_steps`
- `Height / width`
- `guidance_scale`
- `image_guidance_scale`
- `manual_seed`

Imagem 1

Imagem 1

Imagem 1

Imagem 1

Image Variations

- Produce n variations of an input image
- `num_images_per_prompt`

Model Specialization

- Train a new model
- Fine-Tunning
- LoRA

Examples

- Pixel Art
- Caricatures
- Robots

Tasks

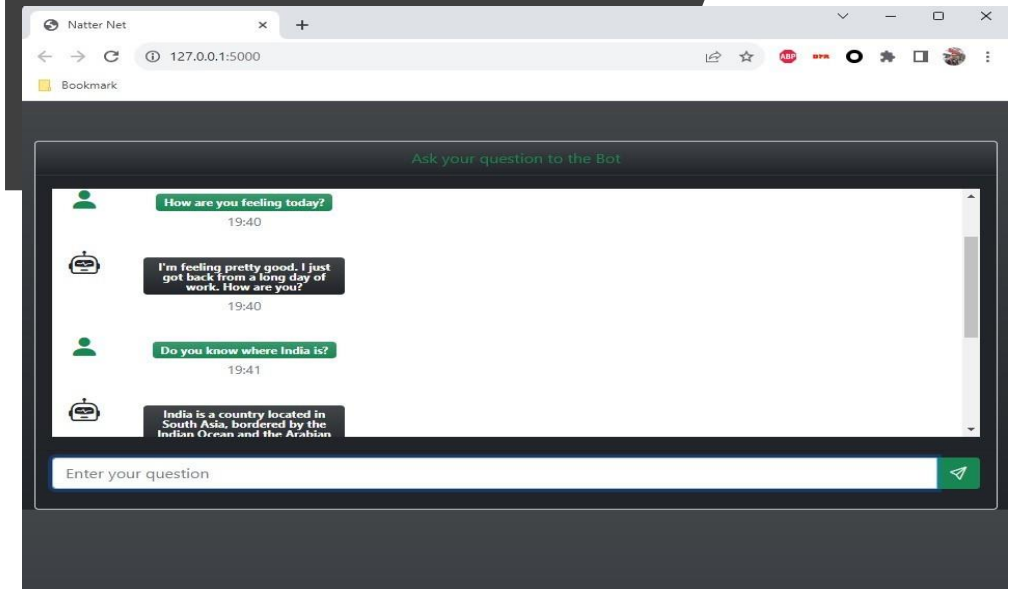
- Text to Audio
- Text to Music
- Text to Speech
- Image to Audio

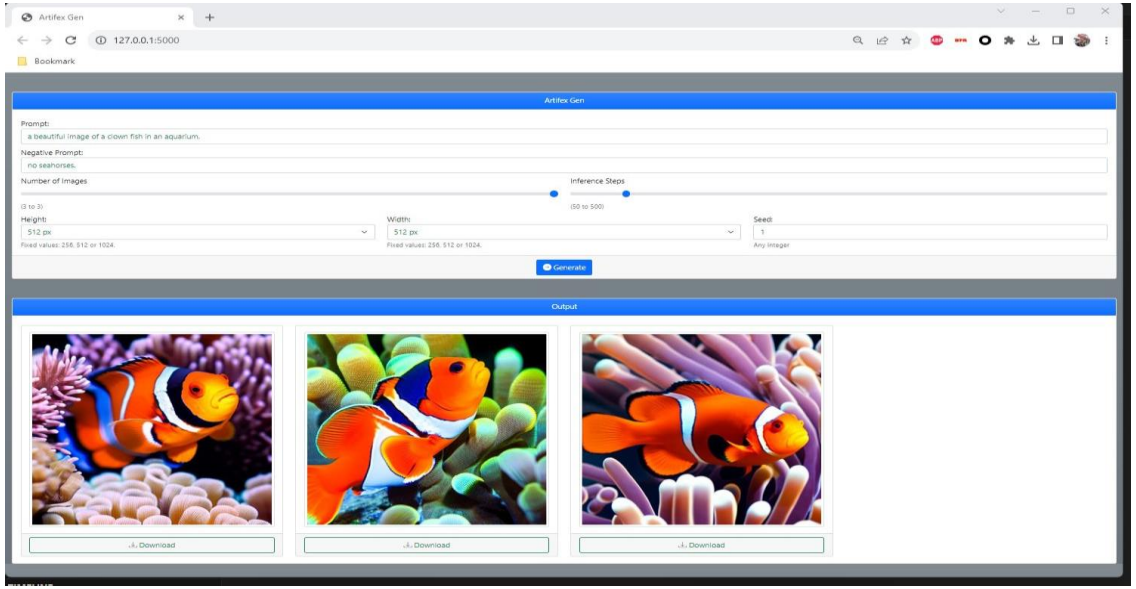
AudioLDM

- AudioLDM and AudioLDM 2
- AudioLDM 2
  - audioldm2: text to audio
  - audioldm2-large: text to audio
  - audioldm2-music: text to music

Applications

- NatterNet: Chatbot
- ArtifexGen: Image Generator





Real World Applications

Back End

Front End

Web Server

Real World Applications

- Running locally
  - Can be deployed to a Webserver

Requirements

- Python 3.10 (or higher)
- VS Code
- Source Code:
  - Download from the course environment
  - Clone the Git Rep