ĐẠI HỌC QUỐC GIA TPHCM

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

BỘ MÔN CÔNG NGHỆ TRI THỨC

Phương pháp toán trong phân tích dữ liệu thị giác

Đề tài: Tên Báo Cáo Gì Đấy

Môn học: Môn Học Gì Đấy

Sinh viên thực hiện:

Nguyễn Quốc Anh - 21127006

Giáo viên hướng dẫn:

GS. TS. Lý Quốc Ngọc

Ngày 30 tháng 10 năm 2024

Mục lục

Danh sách bảng

Danh sách hình vẽ

1 Lecture 1. Introduction to Mathematical Method in VDA

Introduction

the fourth industrial revolution computer vision

1.1 Intelligent Vision System

1.1.1 The concept of Intelligence

Some indications of intelligence that are of interest include:

- memory, recall, creativity
- computation, inference, recognition, prediction
- retrieval, localization & moving, reasoning

1.1.2 The intelligence levels of Vision System's output

1.1.3 The three basic levels of Vision Systems:

- Basic methods for data processing

  Q: Những khám phá nào ở cấp độ 1 mà làm thay đổi ngoạn mục cấp độ 2 và 3?

  A: Fast Fourier Transform (FFT) and Convolutional Neural Network (CNN).
- Single task processing
- Complex applications processing.

1.1.4 Some Intelligent Vision Systems

- Intelligent Transportation System (ITS)
- Intelligent Monitor System (IMS)
- Autonomous Vehicle System (AVS)
- Fault Inspection System (FIS)
- Disease Diagnosis System based on Imaging

- Harvesting System in Agriculture
- Intelligent Image-Video Retrieval

1.1.5 Learning Method

Everything advances slowly Ngta sẽ nghiên cứu các mô hình học máy để cải tiến hơn nữa.

- Supervised learning (semi-, self-)
- Unsupervised learning
- Reinforcement learning: học tăng cường, nổi lên thông qua AlphaGo, bây giờ ứng dụng trong xe tự hành, ChatGPT,... RL is a machine learning technique that focouses on training and algorithm following the cut-and-try approach. The algotirithm
  - The agent or the learner
  - The environment

  Examples:
  - any real-world problem where an agent must interact with an uncertain environment to meet a specific goal: robotics, AlphaGo, autonomous driving, logistics,...

  Benefits:
  - Artificial General Intelligence (AGI)
  - does not need a separate data collection step
- Continual learning: học
- Federated learning
- Deep learning
- Transfer learning

- Meta learning Deep neural networks can achieve great succes when presented wiht large datasets and sufficient computational resources. However, their ability to learn new concepts quickly is limited. It is one of the defining aspects of human intelligence (Jankowski, 2018). Meta learning is one approach to this issue by enabling the network to learn how to learn.

  - Image Classification
  - Facial Recognition and Face Antispoofing
  - Person-specific talking head generateion for unseen

Traditional Programming data -> computer -> results set of rules (proram)
Machine Learning data -> computer -> set of rules (model) results (Optional)
Machine learning as a field is "concerned with the question of how to construct computer programs that automatically improve with experience." - Tom Mitchell 1997 he presents the formal definition of machine learning as follows:
"A computer program is said to learn from **experience E** with respect to some **class of tasks T** and **performance measure P**, if its performance at tasks in T, as measured by P, improves with experience E."
There are three main machine learning paradigms:

- Supervised learning: learning properties of data using labelled data
- Unsupervised learning: learning properties of data using unlabelled data
- Reinforcement learning: learning properties of an environment through trial and error

learning Method
advanced deep neural network system advanced deep neural network architecture

- Le-Net, AlexNet, VGG, GoogleNet
- ResNet, SeNet, EfficientNet
- Graph Neural Network (GNN)
- Generative Adversarial Network (GAN)

- Vision Transformer (ViT)

Q: Có mạng học máy nào mới gần đây không?

A:

- **Swin Transformer**: Một loại mạng Transformer được thiết kế cho các tác vụ thị giác máy tính, nổi bật với khả năng xử lý hình ảnh hiệu quả và chính xác.
- **MLP-Mixer**: Một kiến trúc mạng mới sử dụng các lớp MLP (Multi-Layer Perceptron) để trộn các đặc trưng không gian và kênh, thay vì sử dụng các lớp convolution.
- **ConvNeXt**: Một biến thể hiện đại của mạng convolutional neural network (CNN) truyền thống, được thiết kế để cạnh tranh với các mô hình Transformer trong các tác vụ thị giác.
- **DALL-E 2**: Một mô hình học sâu của OpenAI có khả năng tạo ra hình ảnh từ mô tả văn bản, cải tiến từ phiên bản DALL-E ban đầu.
- **CLIP**: Một mô hình của OpenAI kết hợp giữa hình ảnh và văn bản, cho phép hiểu và tạo ra các mô tả văn bản từ hình ảnh và ngược lại.

1.2 Generative AI

1.2.1 Generative AI in Computer Vision

- Generative model
  - GAN
  - DIFFUSION
- Image-tasks
  - Text2Image, Image2Text
  - Style2Image
  - HumanBrainSignal2Image
- Video-tasks
  - Text2Video, Video2Text

  - Text2Animation

- Computer Graphics-tanks
  - Text23DScene
  - Text23DObjectAnimation

Q: Ví dụ về lĩnh vực, chủ đề mà Generative AI có thể hỗ trợ, không có không được
A: Image2Text (e.g. tóm tắt video)

1.2.2 Diffusion Model

"Diffusion Models are a class of probabilistic generative models that turn noise to a representative data sample."
Using Diffusion models, we can generate images either conditionally or unconditionally.
1. Unconditional image generation simply means that the model converts noise into any "random representative data sample." The generation process is not controlled or guided, and the model can generate an image of any nature.
Q: Nếu như đánh context "flamingos standing on water, red sunset, pink-red water reflection" thì máy có generate được hình ảnh khác không?
A: Có, các mô hình học sâu hiện đại như DALL-E 2 của OpenAI có khả năng tạo ra hình ảnh từ mô tả văn bản. Khi cung cấp mô tả như "flamingos standing on water, red sunset, pink-red water reflection", mô hình có thể tạo ra một hình ảnh tương ứng với mô tả này. Các mô hình này sử dụng các kỹ thuật tiên tiến trong học sâu và xử lý ngôn ngữ tự nhiên để hiểu và chuyển đổi mô tả văn bản thành hình ảnh.

- **DALL-E 2**: Một mô hình học sâu của OpenAI có khả năng tạo ra hình ảnh từ mô tả văn bản. Với mô tả "flamingos standing on water, red sunset, pink-red water reflection", DALL-E 2 có thể tạo ra một hình ảnh với các đặc điểm như vậy.
- **Imagen**: Một mô hình của Google cũng có khả năng tương tự, tạo ra hình ảnh từ mô tả văn bản với độ chính xác và chi tiết cao.

Các mô hình này đã được huấn luyện trên hàng triệu cặp hình ảnh và mô tả văn bản, giúp chúng có khả năng hiểu và tạo ra hình ảnh từ các mô tả phức tạp.

Q: mỗi lần tôi gõ cùng một câu thì nó ra một kết quả khác nhau hay giống nhau?

A: tuỳ vào model

Q: Nếu Intelligence System chỉ dừng ở mức thông minh mà không phải ở mức thông tuệ thì sẽ dừng ở lĩnh vực nào?

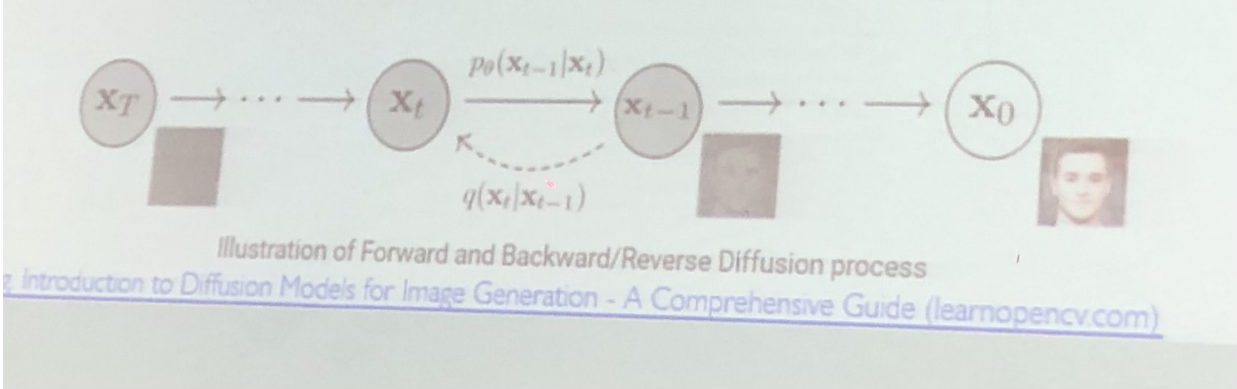
A: Nó sẽ chủ yếu dừng lại ở các lĩnh vực yêu cầu khả năng xử lý thông tin và ra quyết định dựa trên các quy tắc và dữ liệu có sẵn, nhưng không yêu cầu sự hiểu biết sâu sắc, khả năng suy luận phức tạp, hoặc khả năng đưa ra các quyết định mang tính chiến lược và đạo đức. Cụ thể bao gồm:

- **Tự động hóa và điều khiển**: Các hệ thống thông minh có thể thực hiện các nhiệm vụ tự động hóa trong sản xuất, quản lý năng lượng, và điều khiển các thiết bị điện tử.
- **Xử lý ngôn ngữ tự nhiên**: Các ứng dụng như dịch máy, nhận dạng giọng nói, và chatbot có thể hoạt động hiệu quả mà không cần đến mức độ thông tuệ cao.
- **Phân tích dữ liệu**: Các hệ thống thông minh có thể phân tích dữ liệu lớn để tìm ra các mẫu và xu hướng, hỗ trợ ra quyết định trong kinh doanh và tiếp thị.
- **Thị giác máy tính**: Nhận dạng hình ảnh và video, phát hiện đối tượng, và phân loại hình ảnh là những nhiệm vụ mà hệ thống thông minh có thể thực hiện tốt.
- **Hệ thống khuyến nghị**: Các hệ thống này có thể đề xuất sản phẩm, dịch vụ, hoặc nội dung dựa trên sở thích và hành vi của người dùng.

Tuy nhiên, để đạt được mức thông tuệ, hệ thống cần có khả năng hiểu biết sâu sắc, suy luận phức tạp, và đưa ra các quyết định mang tính chiến lược và đạo đức, điều này đòi hỏi sự phát triển vượt bậc trong các lĩnh vực như trí tuệ nhân tạo tổng quát (AGI) và học sâu (deep learning).

An idea used in non-equilibrium statistical physics is that we can **gradually convert one distribution into another**. In 2015, Sohl-Dicktein et al., inspired by this, created "Diffusion Probabilistic models" or "Diffusion models" in short, building on this essential idea.

They build - **"A generative Markov chain which converts a simple known distribution (e.g., a Gaussian) into a target (data) distribution using a diffusion process."**



Hình 1: Illustration of Forward and Backward/Reverse Diffusion process

In 2015, Sohl-Dickstein et al. published paper "Deep Unsupervised Learning using Nonequilibrium Thermodynamics" and Diffusion models in deep learning were first introduced.

In 2019, Song et al. published a paper, "Generative Modeling by Estimating Gradients of the Data Distribution," using the same principle but a different approach.

In 2020, Ho et al. published the paper, now-popular "Denoising Diffusion Probabilistic Models" (DDPM for short).

After 2020, research in diffusion models took off. Much progress has been made in creating, training, and improving diffusion-based generative modeling in a relatively short time.

Some of the Diffusion-based Image Generation models that became famous over the past few months. Some typical famous Diffusion-based Image Generation models include:

- **DALL-E 2** by OpenAI
- **Imagen** by Google
- **Stable Diffusion** by StabilityAI
- **Midjourney**

D and G play the following two-player minimax game with value function $V(G, D)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

Discriminator output for real data x

Discriminator output for generated fake data G(z)

Min Max V(D,G) = Ex-Plate [logD(x)] + Ez-P2(2) [log(1 – D(G(z)))] G

Error from the discriminator model training

Error from the combined model training

Explainable Al inComputer Vision

Role of XAI in Computer Vision

The absence of explicability and transparency in certain areas is not invariably a problem since state- of-the-art models have an extremely high accuracy.

However, in areas such as autonomous cars, financial transactions and mainly medical applications, failures are unacceptable, considering that erroneous decisions can have disastrous con-sequences, such as the loss of human lives. Due to this fact, these application areas have extreme interest in explaining and interpreting each decision made by deep learning models.

It is possible to use Explaining Artifcial Intelligence to improve deep learning models performance.

1.3 Vision-Language Pre-trained Model

1.3.1 Vision-Language Pre-trained

Why is the Vision-Language Pre-trained Model necessary?

Most visual recognition studies rely heavily on crowd-labelled data in deep neural networks (DNNs) training, and they usually train a DNN for each single visual recognition task, leading to a laborious and time-consuming visual recognition paradigm.

To address the two challenges, Vision-Language Models (VLMs) have been intensively investigated recently, which learns rich vision-language correlation from web-scale image-text pairs that are almost infinitely available on the Internet and enables zero-shot predictions on various visual recognition tasks with a single VLM.

1.3.2 Text-Image Tasks

Model Architecture

Given an image-text pair, a VL model first extracts text features w = W1, , WN and visual features v = (V1, VM) via a text encoder and a vision encoder, respectively. Here, N is the number of tokens in a sentence, and M is the number of visual features for an image, which can be the number

of image regions/grids/patches, depending on the specific vision encoder being used. The text and visual features are then fed into a multimodal fusion module to produce cross-modal representations, which are then optionally fed into a decoder before generating the final outputs.

Core VisionTasks

Q: A:

Dataset for VLM

2 Lecture 2. Metric Space

2.1 The Role of Metric Space in VDA

2.2 The Basic Concepts in Metric Space

2.2.1 Metric Space

Metric space (X, d) is space (X) together with a real-valued function d, $d : X \times X \to \mathbb{R}$, which measures the distance between pairs of points x and y in X.
d obeys the following axioms:

(i) $0 < d(x, y) < \infty \quad \forall x, y \in X, x \neq y$

(ii) $d(x, x) = 0 \quad \forall x \in X$

(iii) $d(x, y) = d(y, x) \quad \forall x, y \in X$

(iv) $d(x, y) \leq d(x, z) + d(z, y) \quad \forall x, y, z \in X$

Such a function d is called a metric.

2.2.2 Cauchy sequences

A sequence of points $\{x_n\}_{n=1}^{\infty}$ in a metric space X, d is called a Cauchy sequence if:

$$\forall \epsilon > 0, \exists N > 0 \text{ such that } d(x_n, x_m) < \epsilon \quad \forall n, m > N$$

2.2.3 Convergent sequence

A sequence of points $\{x_n\}_{n=1}^{\infty}$ is said to converge to a point $x \in X$ in metric space X, d if:

$$\forall \epsilon > 0, \exists N > 0 \text{ so that } d(x_n, x) < \epsilon \quad \forall n > N$$

$x \in X$, to which the sequence converges, is called the limit of the sequence, and we use the notation:

$$x = \lim_{n \to \infty} x_n$$

2.2.4 Complete Metric Space

Theorem 1. (*Convergent sequence & Cauchy sequence*)
A sequence of points $\{x_n\}_{n=1}^{\infty}$ in metric space X, d converges to a point $x \in X$, then $\{x_n\}_{n=1}^{\infty}$ is a Cauchy sequence.

Definition 4. (*Complete metric space*)
A metric space X, d is complete if every Cauchy sequence $\{x_n\}_{n=1}^{\infty}$ in X has a limit $x \in X$.

2.2.5 Metric space Hausdorff

Let X, d be a complete metric space. Then $\mathcal{H}(X)$ denotes the space whose points are the compact subsets of X, other than the empty set.

Definition 6. (*Metric in space Hausdorff*)
Let X, d be a complete metric space. Let $A, B \in \mathcal{H}(X)$. The Hausdorff distance between points A and B in $\mathcal{H}(X)$ is defined by:

$$h(A, B) = \max\{d(A, B), d(B, A)\}$$

where

$$d(A, B) = \max\{d(x, B) : x \in A\}$$

$$d(x, B) = \min\{d(x, y) : y \in B\}$$

Theorem 2. (*The completeness of metric space Hausdorff*)
Let X, d be a complete metric space. Then $(\mathcal{H}(X), h_d)$ is a complete metric space. Moreover, if $\{A_n \in \mathcal{H}(X)\}_{n=1}^{\infty}$ is a Cauchy sequence then:

$$A = \lim_{n\to\infty} A_n \in \mathcal{H}(X)$$

$$A = \{x \in X : \exists \text{ a Cauchy sequence } \{x_n \in A_n\} \to x\}$$

2.2.6 Contraction mapping and fixed point

Definition 1. (*Contraction mapping*)

A transformation $f : X \to X$ on a metric space X, d is called contractive or a contraction mapping if:

$$\exists s, 0 \leq s < 1 \text{ such that } d(f(x), f(y)) \leq s \cdot d(x, y) \quad \forall x, y \in X$$

Any such number s is called a contractivity factor for f.

Theorem 3. (*Contraction mapping*)

Let $f : X \to X$ be a contraction mapping on a complete metric space X, d. Then f possesses exactly one fixed point $x_f \in X$ and moreover for any point $x \in X$, the sequence $\{f^n(x)\} \to x_f$. That is:

$$\lim_{n \to \infty} f^n(x) = x_f \quad \forall x \in X$$

Theorem 4. (*Fixed point approximation*)

Let $f : X \to X$ be a contraction mapping on a complete metric space X, d with contractivity factor s. The fixed point x_f is approximated by the following expression:

$$d(f^n(x), x_f) \leq \frac{s^n}{1 - s} d(x, f(x)) \quad \forall x \in X$$

Theorem 5. (*Approximate x by fixed point*)

Let $f : X \to X$ be a contraction mapping on a complete metric space X, d with contractivity factor s, fixed point $x_f \in X$. Then:

$$d(x, x_f) \leq \frac{1}{1 - s} d(x, f(x)) \quad \forall x \in X$$

2.2.7 Contraction mapping on metric space Hausdorff

Lemma 1. (*Contraction mapping*)

Let $f : X \to X$ be a contraction mapping on a complete metric space X, d with contractivity factor s. Then $w : \mathcal{H}(X) \to \mathcal{H}(X)$ defined by:

$$w(B) = \{w(x) : x \in B\} \quad \forall B \in \mathcal{H}(X)$$

is a contraction mapping on $\mathcal{H}(X), h(d)$ with contractivity factor s.

Lemma 2. (*Contraction mapping sequence*)

Let X, d be a metric space. Let $\{w_n\}_{n=1}^{\infty}$ be contraction mappings on $\mathcal{H}(X), h(d)$ with contractivity factor for w_n denoted by s_n for each n. Define $W : \mathcal{H}(X) \to \mathcal{H}(X)$ by:

$$W(B) = w_1(B) \cup w_2(B) \cup \ldots \cup w_N(B)$$

$$= \bigcup_{n=1}^{N} w_n(B) \quad \text{for each } B \in \mathcal{H}(X)$$

Then W is a contraction mapping with contractivity factor:

$$s = \max\{s_n : n = 1, 2, \ldots, N\}$$

Theorem 6. (*Fixed set in metric space Hausdorff*)

Let $X; w_n, n = 1, 2, \ldots, N$ be an iterated function system with contractivity factor s. Then the transformation $W : \mathcal{H}(X) \to \mathcal{H}(X)$ defined by:

$$W(B) = \bigcup_{n=1}^{N} w_n(B) \quad \forall B \in \mathcal{H}(X)$$

is a contraction mapping on the complete metric space $\mathcal{H}(X), h(d)$ with contractivity factor s. Its unique fixed set, $A \in \mathcal{H}(X)$, obeys:

$$A = W(A) = \bigcup_{n=1}^{N} w_n(A)$$

$$A = \lim_{n\to\infty} W^n(B), \quad B \in \mathcal{H}(X)$$

Q: Ý nghĩa của tính chất này?

A: self-similarity (đặc tính tự tương tự)

$$\begin{aligned} A = W(A) &= \bigcup_{n=1}^{N} w_n(A) \\ &= \bigcup_{n=1}^{N} w_n(W(A)) \\ &= \bigcup_{n=1}^{N} \bigcup_{n=1}^{N} (w_n w_n(A)) \end{aligned}$$

$\rightarrow$ tính chất tự phân hình (tính chất fractal)

Q: Lí giải vì sao cho B nào cx quay về A?

A:

Theorem 7. (*Approximate fixed set in metric space Hausdorff*)

Let $\{w_n\}_{n=1}^N$ be an iterated function system with contractivity factor s. Then the transformation $W : \mathcal{H}(X) \rightarrow \mathcal{H}(X)$ defined by:

$$W(B) = \bigcup_{n=1}^{N} w_n(B) \quad \text{for all } B \in \mathcal{H}(X)$$

$A \in \mathcal{H}(X)$ is a fixed set approximated (A không bao giờ có giá trị chính xác mà chỉ có giá trị xấp xỉ) by:

$$h(W^{on}(B), A) \leq \frac{s^n}{1-s} h(B, W(B)) \quad \forall B \in \mathcal{H}(X)$$

Q: Ý nghĩa của tính chất này?

A: cho các xấp xỉ

$$h(W^{on}(B), A) \leq \frac{s^n}{1-s} h(B, W(B)) \quad \forall B \in \mathcal{H}(X)$$
$$\exists \varepsilon : h(W^{on}(B), A) \leq \varepsilon \quad \forall B \in \mathcal{H}(X)$$
$$\text{Let } C = h(B, W(B))$$

Theorem 8. (*Approximate O by fixed set*)

Let O be a subset of $\mathcal{H}(X)$. Let $\{w_n\}_{n=1}^N$ be an iterated function system with contractivity factor s. Then the transformation $W : \mathcal{H}(X) \to \mathcal{H}(X)$ defined by:

$$W(B) = \bigcup_{n=1}^{N} w_n(B) \quad \text{for all } B \in \mathcal{H}(X)$$

$A \in \mathcal{H}(X)$ is a fixed set of W,

$$h(O, A) \leq \frac{1}{1-s} h(O, W(O))$$

Q: Ý nghĩa của tính chất này?

A: khẳng định điều kiện khi nào A gần giống O

2.3 Applications of Metric Space in VDA

Q: Nếu d không thoả điều kiện trong phương trình iii: tính bắc cầu thì gây khó khăn gì?

A:

Cho ví dụ về không gian metric

A: Cho không gian toạ độ ảnh, ta có: d(x,y): khoảng cách giữa 2 điểm x và y trong không gian Euclid, x = (x_1, x_2) và $y = (y_1, y_2)$ với $x_1, x_2, y_1, y_2 \in \mathbb{N}$

$$0 < d(x,y) = \sqrt{(x_1-y_1)^2+(x_2-y_2)^2} < \infty, \forall x \neq y, \forall x_1, x_2, y_1, y_2 \in \mathbb{N}$$

$$d(x,y) = \sqrt{(x_1-y_1)^2+(x_2-y_2)^2} = 0 \Leftrightarrow x = y$$

$$d(x,y) = \sqrt{(y_1-x_1)^2+(y_2-x_2)^2} = \sqrt{(x_1-y_1)^2+(x_2-y_2)^2} = d(y,x)$$

$$Letz = (z_1, z_2) \in \mathbb{N},$$

$$d(x,y) = \sqrt{(x_1-y_1)^2+(x_2-y_2)^2}$$

$$\leq d(x,z) + d(z,y) = \sqrt{(x_1-z_1)^2+(x_2-z_2)^2} + \sqrt{(z_1-y_1)^2+(z_2-y_2)^2}$$

$$\Leftrightarrow (y_1-x_1)^2+(y_2-x_2)^2 \leq (y_1-z_1)^2+(y_2-z_2)^2+(z_1-x_1)^2+(z_2-x_2)^2+$$

$$2\sqrt{(x_1-z_1)^2+(x_2-z_2)^2}\sqrt{(z_1-y_1)^2+(z_2-y_2)^2}$$

$$\Leftrightarrow (y_1-z_1+z_1-x_1)^2+(y_2-z_2+z_2-x_2)^2$$

$$\leq (y_1-z_1)^2+(z_1-x_1)^2-2(y_1-z_1)(z_1-x_1)$$

$$+(y_2-z_2)^2+(z_2-x_2)^2+2(y_2-z_2)(z_2-x_2)$$

$$\leq (y_1-z_1)^2+(z_1-x_1)^2+(y_2-z_2)^2+(z_2-x_2)^2$$

$$\Leftrightarrow [(y_1-z_1)(z_1-x_1)+(y_2-z_2)(z_2-x_2)]$$

$$\leq [(y_1-z_1)^2+(y_2-z_2)^2][(z_1-x_1)^2+(z_2-x_2)^2]$$

Bất đẳng thức Bunjakowski-Schwarz

Q: Cho ví dụ về dãy Cauchy mà không phải dãy hội tụ

A:

Q: Trong hằng hà sa số các ánh xạ, tại sao ánh xạ co được chú ý nhất?

A:

- **Tính ổn định và hội tụ**: Định lí ánh xạ co của Banach đảm bảo rằng mọi ánh xạ co trên một không gian metric đầy đủ đều có một điểm cố định duy nhất. Điều này rất hữu ích trong việc giải các bài toán tìm điểm cố định và đảm bảo sự hội tụ của các phương pháp lặp.
- **Ứng dụng rộng rãi**: Ánh xạ co được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau như giải tích, phương trình vi phân, phương trình tích phân, lý thuyết trò chơi, và khoa học máy tính. Khả năng áp dụng rộng rãi này làm cho ánh xạ co trở thành một công cụ quan trọng trong nhiều bài toán thực tế.
- **Đơn giản và hiệu quả**: Các ánh xạ co thường có cấu trúc đơn giản và dễ hiểu, nhưng lại mang lại hiệu quả cao trong việc giải quyết các bài toán phức tạp. Điều này làm cho chúng trở thành lựa chọn ưu tiên trong nhiều tình huống.
- **Khả năng mở rộng**: Định lí ánh xạ co không chỉ áp dụng cho các không gian metric đầy đủ mà còn có thể được mở rộng và áp dụng cho các không gian khác, chẳng hạn như không gian Banach. Điều này làm tăng tính linh hoạt và khả năng ứng dụng của ánh xạ co.

Vì những lý do trên, ánh xạ co được chú ý và nghiên cứu nhiều trong toán học và các lĩnh vực liên quan.

Q: Giải thích định lí ánh xạ co và ứng dụng của nó

A: Định lí ánh xạ co, hay còn gọi là định lí Banach, phát biểu rằng trong một không gian metric đầy đủ, mọi ánh xạ co đều có một điểm cố định duy nhất. Cụ thể, nếu (X, d) là một không gian metric đầy đủ và $T : X \to X$ là một ánh xạ co, tức là tồn tại một hằng số $0 \leq k < 1$ sao cho:

$$d(T(x), T(y)) \leq k \cdot d(x, y) \quad \forall x, y \in X,$$

thì tồn tại duy nhất một điểm $x^* \in X$ sao cho $T(x^*) = x^*$.

Ứng dụng:

- **Giải phương trình**: Định lí ánh xạ co được sử dụng để chứng minh sự tồn tại và duy nhất của nghiệm cho nhiều loại phương trình, bao gồm phương trình vi phân và phương trình tích phân.
- **Phân tích số liệu**: Trong phân tích số liệu, định lí ánh xạ co được sử dụng để thiết kế các thuật toán lặp để tìm nghiệm của các bài toán tối ưu hóa.

- **Lý thuyết trò chơi**: Định lí ánh xạ co cũng được áp dụng trong lý thuyết trò chơi để chứng minh sự tồn tại của điểm cân bằng Nash trong các trò chơi không hợp tác.
- **Khoa học máy tính**: Trong khoa học máy tính, định lí ánh xạ co được sử dụng trong các thuật toán học máy và trí tuệ nhân tạo để đảm bảo sự hội tụ của các thuật toán lặp.

Generative AI in Computer Vision

Generative Model

D and G play the following two-player minimax game with value function V (G, D):

Min Max V(D,G) = Ex-pan [logD(x)] + E-P(z) [log(1-D(G(z)))]

Discriminator output for real data x

Discriminator output for generated fake data G(z)

Min Max V(D,G) = Ex-Plate [logD(x)] + Ez-P2(2) [log(1 – D(G(z)))] G

Error from the discriminator model training

Error from the combined model training

HOHUI Quốc Ngpo 12023

57

2.4 Lecture 3. Apply metric space to VDA

2.4.1 Drawing based on Fractal geometry

3.1.1. Computing Fractals from Iterated Function Systems

Consider IFS $\{ \mathbb{R}^2, w_n: n = 1, 2, \ldots, N \}$

$$w_i \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix} A_i x + t_i$$

p_i associated with w_i

$$p_i \approx \frac{|\det A_i|}{\sum_{i=1}^{N} |\det A_i|} = \frac{|a_i d_i - b_i c_i|}{\sum_{i=1}^{N} (a_i d_i - b_i c_i)}, i = 1, 2, \ldots, N$$

Q: Vẽ 1 giao diện cho phép tạo W để vẽ hình fractal

A:

Deterministic Algorithm

Let, $X; w_1, w_2, \ldots, w_N$ be an IFS,

Compute directly a sequence of sets $A_n = W^{on}(B)$, starting from an initial set $A \subset \mathbb{R}^2$

$$\begin{aligned} & B \\ & A_1 = W^{01}(B) \\ & A_2 = W^{02}(B) = W^{01}(W^{01}(B)) = W^{01}(A_1) \\ & \ldots \\ A_{n+1} &= W^{01}(A_n) \\ &= \bigcup_{j=1}^{N} w_j(A_n), n = 1, 2, \ldots \end{aligned}$$

Sequence $\{A_n\}$ converges to the attractor A of the IFS in the Hausdorff metric.

Q: Nguyên lí vẽ hình Fractal?
A: Định lí ánh xạ co và hệ hàm lặp IFS

.

Deterministic Algorithm

1. Initialize $s[M, M]$, $t[M, M]$
2. Setup the values of IFS
   - $a[1] = 0.5; b[1] = 0; c[1] = 0; d[1] = 0.5; e[1] = 1; f[1] = 1$
   - $a[2] = 0.5; b[2] = 0; c[2] = 0; d[2] = 0.5; e[2] = 50; f[2] = 1$
   - $a[3] = 0.5; b[3] = 0; c[3] = 0; d[3] = 0.5; e[3] = 50; f[3] = 50$
3. Input the initial set $A(0)$ into $t[M, M]$
4. Repeat
5. For $i = 1$ to M /Apply W to $A(n)$ to make $A(n + 1)$ in $s[i, j]$/
6. For $j = 1$ to M
7. If $t[i, j] = 1$ then
   - $s[a[1] * i + b[1] * j + e[1], c[1] * i + d[1] * j + f[1]] = 1$
   - $s[a[2] * i + b[2] * j + e[2], c[2] * i + d[2] * j + f[2]] = 1$
   - $s[a[3] * i + b[3] * j + e[3], c[3] * i + d[3] * j + f[3]] = 1$
8. End if
9. End /for j/
10. End /for i/
11. For $i = 1$ to M
12. For $j = 1$ to M
13. $t[i, j] = s[i, j]$ /Put $A(n + 1)$ into the array $t[i, j]$/
14. $s[i, j] = 0$ /Reset the array $s[i, j]$ to 0/
15. If $t[i, j] = 1$ then

16. setpixel(i, j) /Plot $A(n+1)$/

17. End if

18. End /for j/

19. End /for i/

20. Until $A(n+1) = W(A(n+1))$

Q: Ý nghĩa của thuật toán này?

A:

Random Iteration Algorithm

Let $\{X; w_1, w_2, \ldots, w_N\}$ be an IFS, where $p_i > 0$ has been $1, 2, \ldots, N$ assigned to $w_i > 0$, $\sum p_i = 1$

Choose $x_0 + - \in X$ and then choose recursively, independently,

$x_n \in \{w_1(x_{n-1}), w_2(x_{n-1}), \ldots, w_N(x_{n-1})\}$, $n = 1, 2, 3, \ldots$

Where the probability of the event $x_n = w_i(x_{n-1})$ is p_i

Sequence $\{x_n\}$ be constructed converges to the attractor of the IFS.

1. Initialize $s[M, M]$, $t[M, M]$

2. Setup the values of IFS

$a[1] = 0.5; b[1] = 0; c[1] = 0; d[1] = 0.5; e[1] = 1; f[1] = 1$

$a[2] = 0.5; b[2] = 0; c[2] = 0; d[2] = 0.5; e[2] = 50; f[2] = 1$

$a[3] = 0.5; b[3] = 0; c[3] = 0; d[3] = 0.5; e[3] = 50; f[3] = 50$

3. $x = 0; y = 0; numits = N$ /Initialize (x, y) and the number of iterations/

4. For $n = 1$ to $numits$ /Random Iteration/

5. $k = \text{int}(3 * \text{rnd} - 0.0001) + 1$ /Choose one of the numbers 1,2,3/

6. $newx = a[k] * x + b[k] * y + e[k]$;

7. $newy = c[k] * x + d[k] * y + f[k]$;

8. $x = newx$;

9. $y = newy$;

10. setpixel(x, y);

11. End /for n/

2.5 Fractal image compression

Method

Based on theorem of contraction mapping sequence in metric space Hausdorff

Let the Original image be O

The compression image will be $\{w_n\}$

The ideal uncompressed will be A

The real uncompressed be $W^{0n}(B)$

Q: Ứng dụng vào ảnh Fractal?

A:

Q: Ảnh tụ có phụ thuộc vào ảnh ban đầu không?

A:

Q: Viết chương trình nén/giải nén bằng kĩ thuật Fractal?

A:

Q: Nguyên lí nén ảnh Fractal?

A:

không gian metric hiểu các yếu tố cơ bản , định lí ánh xạ co & ứng dụng trong tạo hình & nén ảnh Fractal, không gian vector

Q: Trình bày lí thuyết nền tảng tạo hình Fractal?

A:

Q:

A:

3 Vector Space

3.1 Definition (Vector space)

A vector space E over the field T is a non-empty set that satisfies the following 8 properties:

1. $x + y = y + x$
2. $(x + y) + z = x + (y + z)$
3. $\exists 0 \in E : x + 0 = x$
4. $\forall x \in E, \exists (-x) \in E : x + (-x) = 0$
5. $(\alpha + \beta) \cdot x = \alpha \cdot x + \beta \cdot x$
6. $\alpha \cdot (x + y) = \alpha \cdot x + \alpha \cdot y$
7. $(\alpha \cdot \beta) \cdot x = \alpha \cdot (\beta \cdot x)$
8. $1 \cdot x = x$

Note:

- 0 here isn't a number, it's an identity element (phần tử trung hoà)
- 4 tính chất đầu là cộng tích, 4 tính chất sau là nhân tích
- def 5: dấu cộng là cộng 2 value trong trường T, phép nhân là nhân 2 value trong trường T
- def 8: số 1 là value trong trường T

Q: Cho ví dụ về không gian vector

A: *Ví dụ 1: Không gian vector* $\mathbb{R}^2$

Không gian $\mathbb{R}^2$ bao gồm tất cả các cặp số thực (x, y) với $x, y \in \mathbb{R}$. Các phép toán cộng vector và nhân vô hướng được định nghĩa như sau:

- Cộng vector: $(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2)$
- Nhân vô hướng: $c \cdot (x, y) = (c \cdot x, c \cdot y)$

Ví dụ 2: Không gian vector các đa thức bậc không quá n

Không gian P_n bao gồm tất cả các đa thức bậc không quá n với hệ số thực. Một đa thức $p(x)$ trong P_n có dạng:

$$p(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$$

với $a_0, a_1, \ldots, a_n \in \mathbb{R}$. Các phép toán cộng đa thức và nhân vô hướng được định nghĩa như sau:

- Cộng đa thức: $(a_0 + a_1x + \ldots + a_nx^n) + (b_0 + b_1x + \ldots + b_nx^n) = (a_0 + b_0) + (a_1 + b_1)x + \ldots + (a_n + b_n)x^n$
- Nhân vô hướng: $c \cdot (a_0 + a_1x + \ldots + a_nx^n) = (c \cdot a_0) + (c \cdot a_1)x + \ldots + (c \cdot a_n)x^n$

Ví dụ 3: Không gian vector các ma trận

Không gian $M_{m\times n}$ bao gồm tất cả các ma trận kích thước $m \times n$ với các phần tử là số thực. Các phép toán cộng ma trận và nhân vô hướng được định nghĩa như sau:

- Cộng ma trận: $(A + B)_{ij} = A_{ij} + B_{ij}$
- Nhân vô hướng: $(c \cdot A)_{ij} = c \cdot A_{ij}$

3.2 Linear map, Linear Transformation

3.2.1 Definition (Linear map) - Ánh xạ tuyến tính

Given vector spaces E and F over the field T.
The linear map from E to F is the mapping from E to F that obeys f:

$$f(x_1 + x_2) = f(x_1) + f(x_2) \quad \forall x_1, x_2 \in E$$

$$f(k \cdot x) = k \cdot f(x) \quad \forall x \in E, \forall k \in T$$

3.2.2 Theorem (Linear map)

Let $e_1, e_2, \ldots, e_n$ be a basis of E and $f_1, f_2, \ldots, f_n$ be arbitrary vectors of F.

Then there exists a unique linear mapping $f : E \to F$ such that:

$$f(e_i) = f_i, \quad i = 1, \ldots, n$$

3.2.3 The matrix associated with a linear map

Let E be an n-dimensional vector space,

F be an m-dimensional vector space,

and f be the linear map from E to F,

$e_1, e_2, \ldots, e_n$ be the basis vectors in E

and $g_1, g_2, \ldots, g_m$ be the basis vectors in F.

$\to$ The mapping f maps $e_1, e_2, \ldots, e_n$ to $f(e_1), f(e_2), \ldots, f(e_n)$.

Express $f(e_i)$ in terms of $g_1, g_2, \ldots, g_m$:

$$\begin{aligned} f(e_1) &= a_{11}g_1 + a_{12}g_2 + \ldots + a_{1m}g_m \\ f(e_2) &= a_{21}g_1 + a_{22}g_2 + \ldots + a_{2m}g_m \\ &\vdots \\ f(e_n) &= a_{n1}g_1 + a_{n2}g_2 + \ldots + a_{nm}g_m \end{aligned}$$

or

$$\begin{bmatrix} f(e_1) \\ f(e_2) \\ \vdots \\ f(e_n) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ a_{21} & a_{22} & \ldots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nm} \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_m \end{bmatrix}$$

The matrix created by the row coordinates of $f(e_1), f(e_2), \ldots, f(e_n)$ is called the matrix associated with f in the basis $e_1, e_2, \ldots, e_n$ and $g_1, g_2, \ldots, g_m$.

$\to$ This can be written as:

$$f(e) = A \cdot g$$

Calculate the coordinates of $f(x)$ in F:

$$f(x) = f\left(\sum_{i=1}^{n} x_i e_i\right) = \sum_{i=1}^{n} x_i \cdot f(e_i) = \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij} g_j = \sum_{j=1}^{m}\left(\sum_{i=1}^{n} x_i \cdot a_{ij}\right) g_j$$

$$\Rightarrow f(x) = x \cdot A$$

or

$$\begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{bmatrix}$$

Q: Ứng dụng của phương trình trong khoa học thị giác?

A: "tìm vị trí định vị" của robot

3.2.4 Definition (Linear Transformation)

The linear transformation A of vector space E is a linear mapping from E to E.

The linear mapping A from E to E satisfies:

$$A(x_1 + x_2) = A(x_1) + A(x_2) \quad \forall x_1, x_2 \in E$$

$$A(k \cdot x) = k \cdot A(x) \quad \forall x \in E, \forall k \in T$$

3.2.5 Theorem (The invariant subspace of linear transformation)

Every linear transformation in an n-dimensional real vector space E ($n \geq 1$) has at least one 1-dimensional or 2-dimensional invariant subspace.

3.3 Eigenvalue - Eigenvector

3.3.1 Definition (Eigenvalue - Eigenvector)

Let A be the linear transformation of an n-dimensional vector space E.

The subspace E_1 of E is the invariant subspace of A if A transforms every vector x of E_1 to a vector of E_1:

$$Ax = \lambda x$$

Finding the eigenvector and the one-dimensional invariant subspace are equivalent problems:

$$A(kx) = kA(x) = k(\lambda x) = \lambda(kx)$$

3.3.2 Calculation method (Eigenvalue – Eigenvector)

How to calculate eigenvalues and eigenvectors:

1. Let $e_1, e_2, \ldots, e_n$ be the basis of E, and let the matrix $A = [a_{ij}]$ be associated with the linear transformation A.

2. Let vector x be an eigenvector of A corresponding to the eigenvalue λ, and its row coordinate $(x_1, x_2, \ldots, x_n)$.

$$Ax = \lambda x$$

$$(x)A = \lambda(x)$$

$$(x) = \begin{bmatrix} x_1 & x_2 & \ldots & x_n \end{bmatrix}, \quad A = \begin{bmatrix} a_{11} & a_{21} & \ldots & a_{n1} \\ a_{12} & a_{22} & \ldots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \ldots & a_{nn} \end{bmatrix}$$

$$\begin{aligned}(x)A = [&(x_1a_{11} + x_2a_{21} + \ldots + x_na_{n1}), (x_1a_{12} + x_2a_{22} + \ldots + x_na_{n2}), \\ &\ldots, (x_1a_{1n} + x_2a_{2n} + \ldots + x_na_{nn})] \\ &= \lambda[x_1, x_2, \ldots, x_n] = \lambda(x)\end{aligned}$$

$$\begin{cases} x_1a_{11} + x_2a_{21} + \ldots + x_na_{n1} = \lambda x_1 \\ x_1a_{12} + x_2a_{22} + \ldots + x_na_{n2} = \lambda x_2 \\ \vdots \\ x_1a_{1n} + x_2a_{2n} + \ldots + x_na_{nn} = \lambda x_n \end{cases} \Rightarrow \begin{cases} x_1(a_{11} - \lambda) + x_2a_{21} + \ldots + x_na_{n1} = 0 \\ x_1a_{12} + x_2(a_{22} - \lambda) + \ldots + x_na_{n2} = 0 \\ \vdots \\ x_1a_{1n} + x_2a_{2n} + \ldots + x_n(a_{nn} - \lambda) = 0 \end{cases}$$

$$\Rightarrow \begin{vmatrix} a_{11}-\lambda & a_{21} & \dots & a_{n1} \\ a_{12} & a_{22}-\lambda & \dots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \dots & a_{nn}-\lambda \end{vmatrix} = 0 \Rightarrow \begin{vmatrix} a_{11}-\lambda & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22}-\lambda & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn}-\lambda \end{vmatrix} = 0$$

Suppose that the linear transform has n independent eigenvectors.

Suppose that A is the linear transform, and e_i are its n independent eigenvectors.

$$Ae_i = \lambda_i e_i, \quad i = 1, 2, \dots, n$$

If n eigenvectors are chosen as the basis vectors, then the matrix of the linear transformation is diagonal.

$$A = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}$$

3.4 Linear form, Bilinear form, Quadratic form

3.4.1 Definition (Linear forms)

Given vector space E over the field T.

The linear form on E is the linear map from E to T that obeys:

$$\varphi(x_1 + x_2) = \varphi(x_1) + \varphi(x_2) \quad \forall x_1, x_2 \in E$$

$$\varphi(k \cdot x) = k \cdot \varphi(x) \quad \forall x \in E, \forall k \in T$$

3.4.2 Definition (Bilinear forms)

Given vector spaces E and F over the field T.

$\rightarrow$ The mapping φ from $E \times F$ to T.

φ is the bilinear form on $E \times F$ if it is the linear form of x when y is fixed, and the linear form of

y when x is fixed.

$$\varphi(x_1 + x_2, y) = \varphi(x_1, y) + \varphi(x_2, y)$$
$$\varphi(k \cdot x, y) = k \cdot \varphi(x, y)$$
$$\varphi(x, y_1 + y_2) = \varphi(x, y_1) + \varphi(x, y_2)$$
$$\varphi(x, k \cdot y) = k \cdot \varphi(x, y)$$

Suppose that E is an n-dimensional vector space, and F is an m-dimensional vector space.

$e_1, e_2, \ldots, e_n$ are the basis vectors of E,

$f_1, f_2, \ldots, f_n$ are the basis vectors of F.

$$\begin{aligned}\varphi(x, y) &= \varphi(x_1 e_1 + x_2 e_2 + \ldots + x_n e_n, y_1 f_1 + y_2 f_2 + \ldots + y_n f_n) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{m} x_i y_j \varphi(e_i, f_j) \\ &\Rightarrow a_{ij} = \varphi(e_i, f_j)\end{aligned}$$

The matrix $A = [a_{ij}]$ is said to be the matrix representing the bilinear form in the basis vectors $e_1, e_2, \ldots, e_n$ and $f_1, f_2, \ldots, f_n$.

(x) is the row coordinate of x in the basis vectors $e_1, e_2, \ldots, e_n$.

(y) is the row coordinate of y in the basis vectors $f_1, f_2, \ldots, f_n$.

$$\varphi(x, y) = (x)A(y^T)$$

Suppose that e, f are the bases of E and F respectively, the matrix representing the bilinear form is A.

In the new base e' of E, f' of F, the matrix representing the bilinear form is A':

$$A' = TAS^T$$

T, S are the transformation matrices from the old basis to the new ones.

$$T \cdot e = e'$$

$$S \cdot f = f'$$

If φ is the bilinear form on $E \times E$ then:

$$A' = TAT^T$$

T is the transition matrix from the old basis to the new one in E.

$$\varphi(x, y) = (x)A(y^T) = (x')TA(y'^T)T = (x')(TAT^T)(y')$$

3.4.3 Definition (Quadratic forms)

Given vector space E over the field T.
$\rightarrow \varphi$ is the bilinear form on $E \times E$. The quadratic form w associated with φ is the mapping w from E to T:

$$w(x) = \phi(x, x) \quad \forall x \in E, \quad w(x) \in T$$

Suppose that E is an n-dimensional vector space, and $e_1, e_2, \ldots, e_n$ are the basis vectors of E.

$$w(x) = \varphi(x, x) = \varphi\left(\sum_{i=1}^{n} x_i e_i, \sum_{j=1}^{n} x_j e_j\right) = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j \varphi(e_i, e_j)$$

$$w(x) = (x)A(x^T)$$

where $A = [\varphi(e_i, e_j)]$.
If in E, T is the transition matrix from the old basis to the new basis, then the quadratic form in the new basis is:

$$w(x) = (x)A(x^T) = (x')TA((x')T)^T = (x')TAT^T(x')^T$$

where $A' = TAT^T$.

3.4.4 Transform the quadratic form to the canonical form

Consider the quadratic form on an n-dimensional vector space E over the field T.

Prove that in E, there are basis vectors $f_1, f_2, \ldots, f_n$ such that

if

$$x = \sum_{i=1}^{n} u_i f_i$$

then

$$w(x) = k_1 u_1^2 + k_2 u_2^2 + \ldots + k_n u_n^2$$

where $k_1, k_2, \ldots, k_n$ are elements of the field T.

3.5 Symmetry Operators

3.5.1 Definition (Euclidean space)

A vector space E is called a Euclidean space if:

For each pair (x, y) of E corresponding to a real number called the dot product of x, y obeys:

$$\langle x, y\rangle = \langle y, x\rangle$$
$$\langle x_1 + x_2, y\rangle = \langle x_1, y\rangle + \langle x_2, y\rangle$$
$$\langle kx, y\rangle = k\langle x, y\rangle \quad \forall k$$
$$\langle x, x\rangle > 0 \quad \text{if} \quad x \neq 0$$
$$\langle x, x\rangle = 0 \quad \text{if} \quad x = 0$$

3.5.2 Dot product is the bilinear form

In an n-dimensional vector space E, let $e_1, e_2, \ldots, e_n$ be the basis vectors.

Then:

$$x = \sum_{i=1}^{n} x_i e_i, \quad y = \sum_{j=1}^{n} y_j e_j$$

$$\langle x, y\rangle = \left\langle \sum_{i=1}^{n} x_i e_i, \sum_{j=1}^{n} y_j e_j \right\rangle = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i y_j \langle e_i, e_j\rangle$$

3.5.3 Definition (Symmetric operator)

The linear transformation A of Euclidean space E is said to be symmetric if:
For every pair of vectors x, y of E, we have:

$$\langle Ax, y\rangle = \langle x, Ay\rangle$$

3.5.4 Diagonalize the transformation matrix of the symmetric operator

A is symmetric $\rightarrow |A-\lambda I| = 0$ has a real root $\rightarrow$ There exists a 1-dimensional invariant subspace of eigenvectors $\rightarrow$ construct an orthogonal basis of eigenvectors $e_1, e_2, \ldots, e_n \rightarrow$ transformation matrix is diagonalized.

- The characteristic polynomial of the symmetric transformation has real roots.
- Every symmetric transformation has at least one 1-dimensional invariant subspace because it has a real root of the characteristic polynomial.
- A is symmetric $\rightarrow$ Find out the orthogonal matrix T such that TAT^{-1} is diagonal, diagonal elements are the characteristic root of A, each root taken a number of times equal to its multiplicity.

3.5.5 Reduce the quadratic form to the canonical form

Theorem:
In the n-dimensional Euclidean space E, every quadratic form can be reduced to a unique canonical form

$$w(x) = \sum_{i=1}^{n} \lambda_i u_i^2$$

by the orthogonal transformation, λ_i are the eigenvalues of w
In the n-dimensional Euclidean space E, the quadratic form $w(x)\varphi(x, y)$ is the symmetric bilinear form for $w(x)$

in E, choose the orthogonal basis $e_1, e_2, \ldots, e_n$:

$$\varphi(x,y) = \langle \sum_{i=1}^{n} x_i e_i, \sum_{j=1}^{n} y_j e_j \rangle = \sum_{j=1}^{n} \left(\sum_{i=1}^{n} x_i a_{ij} \right) y_j = \sum_{j=1}^{n} z_j y_j$$

$$a_{ij} = \varphi(e_i, e_j) = \varphi(e_j, e_i) = a_{ji}$$

$$z_j = \sum_{i=1}^{n} x_i a_{ij}, \quad j = 1, 2, \ldots, n$$

$$\Rightarrow z = Ax$$

The linear transformation A is symmetric because its transformation in the orthogonal basis system is symmetric. According to the invariant subspace theorem of the symmetric transformation, in E there is an orthogonal basis of the eigenvectors of A: $f_1, f_2, \ldots, f_n$.
In this basis:

$$x = \sum_{i=1}^{n} u_i f_i, \quad y = \sum_{j=1}^{n} v_j f_j$$

$$\begin{aligned} \varphi(x,y) = \langle Ax, y \rangle &= \left\langle A \left(\sum_{i=1}^{n} u_i f_i \right), \sum_{j=1}^{n} v_j f_j \right\rangle \\ &= \left\langle \left(\sum_{i=1}^{n} u_i (A f_i) \right), \sum_{j=1}^{n} v_j f_j \right\rangle \\ &= \left\langle \left(\sum_{i=1}^{n} \lambda_i u_i v_i \right), \sum_{j=1}^{n} v_j f_j \right\rangle \\ &= \sum_{i=1}^{n} \lambda_i u_i v_i \end{aligned}$$

$$w(x) = \phi(x,x) = \sum_{i=1}^{n} \lambda_i u_i^2$$

3.6 Applications

4 Chapter 4. Optimization Method

4.1 The role of optimization method in VDA

4.2 Unconstrained optimization method

4.3 Constrained optimization method

4.4 Applications of optimization method in VDA

5 Chapter 5. Method of Solving a System of Equations

5.1 The role of system of equations in VDA

5.2 Method of solving system of linear equations

5.3 Method of solving system of non-linear equations

5.4 Applications of system of equations in VDA

6 Chapter 6. Method of Solving Partial Differential Equations

6.1 The role of PDE in VDA

6.2 Method of solving PDE

6.3 Applications of PDE in VDA

7 Chapter 7. Deep Learning

7.1 The role of DL in VDA

7.2 2D and 3D Deep Convolution Neural Network

7.3 Deep Recurrent Neural Network

7.4 Applications of DL in VDA

- Để gõ thuật toán: `algorithm` và `algpseudocode`
- Để nhúng (chèn) code: `listings`

Các gói này được cài đặt thông qua lệnh

```
sudo apt-get install texlive-full
```

Tuy nhiên kích thước gói đâu đó vào khoảng 5GB (!). Vì vậy tốt nhất nên xài Overleaf.

8.1.2 Sử dụng font khác

Tham khảo font typefaces tại link này.

8.1.3 Đánh số chỉ mục bằng chữ số La Mã

Mở file `main.tex` và bỏ comment dòng

```
% \renewcommand{\thesection}{\Roman{section}}
% \renewcommand{\thesubsection}{\thesection.\Roman{subsection}}
```

8.2 Ví dụ

Ngày xửa ngày xưa, ở vương quốc VNUHCM - US, có một chàng hoàng tử ngồi cắm đầu viết doc[1].

Mặc định muốn xuống dòng chỉ cần dùng \\ (2 lần dấu xẹt huyền).

Nếu bạn muốn thụt đầu dòng khi bắt đầu paragraph mới, vào `main.tex` và disble dòng

```
\setlength{\parindent}{0pt}
```

8.3 First subsection

8.3.1 First sub-subsection

Subsection để ví dụ thôi. Thêm vài ví dụ:

- Dùng itemize
- Vẫn là itemize

Sau đó xài enumerate:

[1] Dây là footnote, chú thích lại những gì cần chú ý.

Hình 2: Hình ví dụ (logo HCMUS - updated 30/11/2022)

1. Dùng enumerate

2. Vẫn là enumerate

Nhỏ hơn subsubsection thì xài `paragraph`:

Đây là ví dụ cho paragraph Lưu ý là paragraph không nằm trong Mục lục.

8.4 Chia nhỏ nội dung

Bạn có thể chia nhỏ nội dung của báo cáo thành các file `.tex` và dùng lệnh `input` để chèn vào báo cáo chính. Ví dụ có trong file `main.tex`.

9 Hình ảnh

Hình ảnh được thể hiện như hình 2, lưu ý flag `[H]` để disable floating (hình được hiển thị đúng vị trí, không trôi lên đầu trang).
Hình 3 cũng là hình ví dụ nhưng có tag `[H]`. Lưu ý là có tag `[H]` thì code ở đâu hình sẽ nằm ở đó, không quan trọng nội dung ít hay nhiều (trang giấy sẽ thừa 1 khúc như bạn thấy). Để hiểu hơn về positioning trong LaTeX, xin tham khảo bài này.

Tên con vật	Số chân
Gà	2
Chó	4
Trần Hoàng Tử	2

Bảng 1: Số chân của một số con vật, không có tag `[H]`

Hình 3: Hình ví dụ (logo HCMUS - updated 30/11/2022)

10 Bảng biểu

Bảng biểu được thể hiện như bảng 1, lưu ý flag `[H]` để disable floating (bảng được hiển thị đúng vị trí, không trôi lên đầu trang). Bảng 1 là một trường hợp không sử dụng tag `[H]` và bảng bị trôi tít lên đầu trang:

Bảng 2 thể hiện bảng biểu với tag `[H]`[2]. Để không phải mất thời gian tuổi trẻ ngồi chỉnh table, xài https://www.tablesgenerator.com.

Tên con vật	Số chân
Gà	2
Chó	4
Trần Hoàng Tử	2

Bảng 2: Số chân của một số con vật, có tag `[H]`

[2]Tương tự cách sử dụng tag `[H]` với hình

.

11 Công thức toán

Công thức toán gõ chung 1 dòng thì dùng 2 lần dấu dollar: $f(x) = x^2 + 2x + 1$. Với công thức nằm riêng 1 dòng thì gõ 2 cặp dấu dollar:

$$ReLU(x) = \max(0, x)$$

Siêu việt hơn, gõ hệ phương trình thì nên dùng tag `equation`

$$a_1x_1 + a_2x_2 + .. + a_nx_n = u$$
$$b_1x_1 + b_2x_2 + .. + b_nx_n = v$$
$$c_1x_1 + c_2x_2 + .. + c_nx_n = w$$

Tham khảo cách gõ equation trên Overleaf nhé!

12 Thuật toán

Dùng gói `algorithm` và `algpseudocode` để gõ đoạn thuật toán 1[3]

13 Code

Dùng gói `listings` để gõ code, ví dụ cho C++:

```cpp
#include <iostream>

int main() {
  std::cout << "Hello, world!\n";
  return 0;
}
```

Cho Python:

[3]Tất nhiên đây là dùng katana mổ ruồi!

Algorithm 1 Thuật toán đếm xem nhiều gà hay nhiều chó hơn

function GaChoSoNaoLonHon(ga, cho)
  $soGa \leftarrow 0$
  $soCho \leftarrow 0$
  for $i \in [0, |ga| - 1]$ **do**
    $soGa \leftarrow soGa + 1$
  end for
  for $i \in [0, |cho| - 1]$ **do**
    $soCho \leftarrow soCho + 1$
  end for
  if $soGa > soCho$ **then**
    return $soGa$
  else if $soGa < soCho$ **then**
    return $soCho$
  else
    return "bang nhau"
  end if
end function

```python
import turtle

def draw_fractal_tree(branch_length, t):
    if branch_length > 5:
        t.forward(branch_length)
        t.right(20)
        draw_fractal_tree(branch_length - 15, t)
        t.left(40)
        draw_fractal_tree(branch_length - 15, t)
        t.right(20)
        t.backward(branch_length)

def main():
    screen = turtle.Screen()
    screen.bgcolor("white")

    t = turtle.Turtle()
    t.color("green")
    t.speed(0)
    t.left(90)
    t.up()
```

```
    t.backward(100)
    t.down()
    t.color("brown")

    draw_fractal_tree(100, t)

    screen.mainloop()

if __name__ == "__main__":
    main()
```

Đặc biệt: code có comment bằng tiếng Việt

```
# In các số chẵn trong đoạn [1, 10]

for i in range(1, 11):
    if i % 2 == 0:
        print(i)
```

14 Ngôn ngữ

Ngôn ngữ mặc định của template là Tiếng Việt, config ở file `main.tex` với lệnh

```
\usepackage[utf8]{vietnam}
```

Để chuyển sang Tiếng Anh (e.g. nhiều khi bạn muốn label trong các bảng bằng Tiếng Anh; bạn muốn viết report bằng Tiếng Anh thay vì Tiếng Việt), khi đó có 2 lựa chọn:

- Chuyển xang xài package `babel` và xài tag `\uselanguage`.
- Bỏ xài package `vietnam`

Hướng dẫn thì mời bạn xem link này

15 Sử dụng tài liệu tham khảo

File BibTeX tài liệu tham khảo nằm ở đường dẫn `ref/ref.bib`. Sửa tên file `.bib` sẽ phải sửa lại nội dung file `ref.tex`.
Đây là ví dụ cite một tài liệu[?].

A Phụ lục

- Template này **không phải** là template chính thức của Khoa Công nghệ thông tin - Trường Đại học Khoa học Tự nhiên.
- Các hình ảnh, bảng biểu, thuật toán trong template chỉ mang tính chất ví dụ.
- Nhóm tác giả phân phối **miễn phí** template này trên GitHub và trên Overleaf với Giấy phép GNU General Public License v3.0. Nhóm tác giả không chịu trách nhiệm với các bản phân phối không nằm trong hai kênh phân phối chính thức nêu trên.