**1. Hàm khoảng cách (từ Lesson 11 KNN.pdf, trang 19)**

**a. Khoảng cách Manhattan (L1 norm)**

Công thức:

Ảnh có chứa văn bản, Phông chữ, đồng hồ, Đồ họa

Nội dung do AI tạo ra có thể không chính xác.

x,y: Hai điểm dữ liệu

n: Số đặc trưng

Mô tả: Đo tổng chênh lệch tuyệt đối giữa các chiều. Thường hiệu quả khi dữ liệu có phân phối không chuẩn (trang 20).

Code:

def manhattan\_distance(x, y):

"""Tính khoảng cách Manhattan (slide Lesson 11, trang 19)"""

return sum(abs(a - b) for a, b in zip(x, y))

**b. Khoảng cách Euclidean (L2 norm)**

Công thức:

Ảnh có chứa văn bản, Phông chữ, đồng hồ, Đồ họa

Nội dung do AI tạo ra có thể không chính xác.

Code:

import math

def euclidean\_distance(x, y):

"""Tính khoảng cách Euclidean (slide Lesson 11, trang 19)"""

return math.sqrt(sum((a - b) \*\* 2 for a, b in zip(x, y)))

**c. Khoảng cách Minkowski (tổng quát hóa)**

Công thức:

**Ảnh có chứa Phông chữ, văn bản, thiết kế, thuật in máy

Nội dung do AI tạo ra có thể không chính xác.**

Trong đó:

* P: Tham số quyết định loại khoảng cách
* P = 1: Manhattan
* P = 2: Eculidean

Code:

def minkowski\_distance(x, y, p=2):

"""Tính khoảng cách Minkowski (slide Lesson 11, trang 19)"""

return sum(abs(a - b) \*\* p for a, b in zip(x, y)) \*\* (1 / p)

**2. Hàm mất mát (từ Lesson 13 ANN 1.pdf)**

**a. Cross-Entropy Loss (phân loại đa lớp, trang 20-22)**

**Công thức:**

Ảnh có chứa Phông chữ, văn bản, chữ viết tay, thuật in máy

Nội dung do AI tạo ra có thể không chính xác.

Trong đó:

N: số mẫu

C: Số lớp

y(i,c): Nhãn thực tế

y^(I,c): Xác suất dự đoán từ softmax

Code:

import numpy as np

def cross\_entropy\_loss(y\_true, y\_pred):

"""Tính mất mát cross-entropy (slide Lesson 13, trang 20)"""

n\_samples = len(y\_true)

y\_pred = np.clip(y\_pred, 1e-15, 1 - 1e-15) # Tránh log(0)

loss = -np.sum(y\_true \* np.log(y\_pred)) / n\_samples

return loss

**b. Mean Squared Error (MSE, ngầm định cho hồi quy, trang 5)**

Công thức:

Ảnh có chứa Phông chữ, văn bản, Đồ họa, thuật in máy

Nội dung do AI tạo ra có thể không chính xác.

Trong đó:

Yi: Giá trị thực

Y^i: Giá trị dự đoán

N: số mẫu

Code:

def mean\_squared\_error(y\_true, y\_pred):

"""Tính mất mát MSE (slide Lesson 13, trang 5, ngầm định)"""

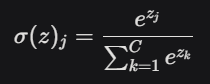
n\_samples = len(y\_true)

return np.sum((y\_true - y\_pred) \*\* 2) / n\_samples

**3. Bổ sung:**

**a. Softmax (Lesson 13 ANN 1.pdf, trang 20-22):**

Công thức:



Trong đó:

Z: Vecto đầu vào

J: Chỉ số lớp

C: Số lớp

Softmax: Xác suất dự đoán cho lớp j

Code:

import numpy as np

def softmax(z):

"""Tính hàm Softmax (slide Lesson 13, trang 20)"""

exp\_z = np.exp(z - np.max(z, axis=1, keepdims=True)) # Tránh tràn số

return exp\_z / np.sum(exp\_z, axis=1, keepdims=True)

**b. ReLU (Rectified Linear Unit, Lesson 13 ANN 1.pdf, trang 19):**

Công thức:

Ảnh có chứa Phông chữ, Đồ họa, biểu tượng, màu trắng

Nội dung do AI tạo ra có thể không chính xác.

Mô tả:

Hàm kích hoạt phi tuyến, dùng trong các tầng ẩn của ANN.

Đơn giản, giúp tránh vanishing gradient, tăng tốc hội tụ.

Code:

def relu(x):

"""Tính hàm ReLU (slide Lesson 13, trang 19)"""

return np.maximum(0, x)

**c. Sigmoid (Lesson 13 ANN 1.pdf, trang 5, ngầm định)**

Công thức:

Ảnh có chứa Phông chữ, biểu tượng, số, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

Mô tả:

* Hàm kích hoạt, ánh xạ đầu vào sang [0, 1].
* Ngầm định Sigmoid là hàm kích hoạt phổ biến.
* Thường dùng trong phân loại nhị phân hoặc tầng output (trước khi Softmax phổ biến).

Code:

def sigmoid(x):

"""Tính hàm Sigmoid (slide Lesson 13, trang 5, ngầm định)"""

return 1 / (1 + np.exp(-np.clip(x, -500, 500))) # Tránh tràn số

**d. Tanh (Hyperbolic Tangent, Lesson 13 ANN 1.pdf, trang 5, ngầm định)**

Công thức:

Ảnh có chứa Phông chữ, hàng, thuật in máy, thiết kế

Nội dung do AI tạo ra có thể không chính xác.

Mô tả:

* Hàm kích hoạt, ánh xạ đầu vào sang [−1,1].
* Tương tự Sigmoid, được ngầm định trong ANN (trang 5).
* Tốt hơn Sigmoid trong một số trường hợp do đầu ra cân bằng quanh 0.

Code:

def tanh(x):

"""Tính hàm Tanh (slide Lesson 13, trang 5, ngầm định)"""

return np.tanh(x)