Báo cáo tiến độ tuần 4.

Người thực hiện: Trần Đức Thọ

Bài 1: Em đang xây dựng mô hình, mô hình của em chưa hoàn thiện do data hơi lớn nên mỗi lần em muốn sửa thì đều rất tốn thời gian.

Bài 2:

Mạng RNN là gì ?

Mạng nơ-ron tuần hoàn là mạng nơ-ron chuyên dùng để xử lý một chuỗi dữ liệu **x(t)= x(1), . . . , x(τ)**với chỉ số bước thời gian **t** khác nhau **1 to τ**. Đối với các tác vụ liên quan đến đầu vào tuần tự, chẳng hạn như giọng nói và ngôn ngữ, thường tốt hơn là sử dụng RNN. Trong một bài toán NLP, nếu bạn muốn dự đoán từ tiếp theo trong câu thì điều quan trọng là phải biết các từ trước nó. RNNs được gọi là tái phát vì họ thực hiện nhiệm vụ tương tự cho mọi phần tử của một chuỗi, với sản lượng được phụ thuộc vào tính toán trước. Một cách khác để nghĩ về RNN là chúng có một “bộ nhớ” ghi lại thông tin về những gì đã được tính toán cho đến nay.

A picture containing text, clock, gauge

Description automatically generated

Bài 3:

What ?

L1 and L2 regularization là 2 phương pháp giúp chúng ta giảm thiểu mô hình bị overfitting.

Why ?

Với các mô hình (học có giám sát), chúng ta luôn muốn cực tiểu hóa hàm loss function, do quá chú trọng đến hàm loss mà nhiều khi chúng ta để vấn để overfitting xảy ra làm ảnh hưởng tới mô hình, do đó cần có regularization để giảm thiểu overfitting.

How ?

regularization: Với W và b là “weight” và “bias” của mô hình:

A picture containing logo

Description automatically generated

Và

A picture containing shape

Description automatically generated

Mô hình Ŷ  của chúng ta xây dựng

A picture containing text, antenna

Description automatically generated

Với bài toán học có giám sát chúng ta xây dựng hàm Loss bằng cách

Logo

Description automatically generated with low confidence

Với L1 regularization chúng ta có công thức

Text

Description automatically generated with low confidence

ở đây  𝝺 là các parameter mà chúng ta thiết lập cho mô hình  𝝺 không nên quá lớn vì nó ảnh hưởng quá lớn tới mô hình, nếu  𝝺 = 0 phần thêm vào là vô nghĩa

Với Gradient Descent chung ta cập nhật trọng số mới

A picture containing graphical user interface

Description automatically generated

Hay chúng ta có thể viết lại là:

Text, letter

Description automatically generated

Với L2 regularization chúng ta có hàm L2 regularization:

A picture containing text

Description automatically generated

Lúc này, áp dụng Gradient Descent chúng ta có thể cập nhật trọng số mới với:

A picture containing diagram

Description automatically generated

Hay chúng ta có thể viết là:

Text, letter

Description automatically generated

So sánh:

* Trong trường hợp L1 regularization cố gắng ước tính mức trung bình của dữ liệu, L2 regularization thực hiện ước tính giá trị trung bình của dữ liệu để tránh overfitting.
* Thông qua việc bao gồm giá trị tuyệt đối của các tham số trọng số, L1 regularization có thể thêm điều khoản phạt trong hàm chi phí. Mặt khác, L2 regularization bổ sung giá trị bình phương của các trọng số trong hàm chi phí.
* L1 regularization đưa ra đầu ra dưới dạng trọng số nhị phân từ 0 đến 1 cho các tính năng của mô hình và được chấp nhận để giảm số lượng các tính năng trong một tập dữ liệu chiều lớn.
* L2 regularization phân tán các điều khoản lỗi trong tất cả các trọng số dẫn đến các mô hình cuối cùng được tùy chỉnh chính xác hơn.

Bài 4:

Phép phân tích thành phần chính (Principal Components Analysis - PCA) là một [thuật toán](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n) [thống kê](https://vi.wikipedia.org/wiki/Th%E1%BB%91ng_k%C3%AA) sử dụng phép biến đổi [trực giao](https://vi.wikipedia.org/wiki/Tr%E1%BB%B1c_giao) để biến đổi một [tập hợp](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_h%E1%BB%A3p) [dữ liệu](https://vi.wikipedia.org/wiki/D%E1%BB%AF_li%E1%BB%87u) từ một [không gian](https://vi.wikipedia.org/wiki/Kh%C3%B4ng_gian) nhiều chiều sang một không gian mới ít chiều hơn (2 hoặc [3 chiều](https://vi.wikipedia.org/wiki/Kh%C3%B4ng_gian_ba_chi%E1%BB%81u)) nhằm tối ưu hóa việc thể hiện [sự biến thiên](https://vi.wikipedia.org/w/index.php?title=S%E1%BB%B1_bi%E1%BA%BFn_thi%C3%AAn&action=edit&redlink=1) của dữ liệu

Phép biến đổi tạo ra những ưu điểm sau đối với dữ liệu:

-Giảm số chiều của không gian chứa dữ liệu khi nó có số chiều lớn, không thể thể hiện trong không gian 2 hay 3 chiều.

-Xây dựng những trục tọa độ mới, thay vì giữ lại các trục của không gian cũ, nhưng lại có khả năng biểu diễn dữ liệu tốt tương đương, và đảm bảo độ biến thiên của dữ liệu trên mỗi chiều mới.

-Tạo điều kiện để các liên kết tiềm ẩn của dữ liệu có thể được khám phá trong không gian mới, mà nếu đặt trong không gian cũ thì khó phát hiện vì những liên kết này không thể hiện rõ.

-Đảm bảo các trục tọa độ trong không gian mới luôn trực giao đôi một với nhau, mặc dù trong không gian ban đầu các trục có thể không trực giao.

Code demo:

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns; sns.set()

rng = np.random.RandomState(1)

X = np.dot(rng.rand(2, 2), rng.randn(2, 200)).T

from sklearn.decomposition import PCA

# pca = PCA(n\_components=2)

# pca.fit(X)

pca = PCA(n\_components=1) #

pca.fit(X)

X\_pca = pca.transform(X)

print("original shape:   ", X.shape)

print("transformed shape:", X\_pca.shape)

X\_new = pca.inverse\_transform(X\_pca)

plt.scatter(X[:, 0], X[:, 1], alpha=0.2)

plt.scatter(X\_new[:, 0], X\_new[:, 1], alpha=0.8)

plt.axis('equal')

Chart, scatter chart

Description automatically generated

Em đã chiếu các điểm dữ liệu (đã bị mờ) thành các điểm dữ liệu màu vàng thu gọn (số chiều) của điểm dữ liệu