

Bài 1

Cho mạng nơron nhân tạo sau

$$\begin{aligned}o_0 &= x, \\z_\ell &= W_\ell o_{\ell-1} + b_\ell, \ell = 1, 2, \dots, L, \\o_\ell &= \sigma_\ell(z_\ell), \ell = 1, 2, \dots, L-1, \\o &= z_L.\end{aligned}$$

Trong đó,

- $x \in \mathbb{R}^{d_0}$ là đầu vào,
- $o_\ell \in \mathbb{R}^{d_\ell}$ là đầu ra của tầng thứ ℓ (với $d_L = C$ là số lớp phân loại),
- $o = z_L \in \mathbb{R}^C$ là đầu ra của tầng cuối cùng và của mạng nơ-ron,
- $W_\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ và $b_\ell \in \mathbb{R}^{d_\ell}$ lần lượt là ma trận trọng số và véc-tơ độ lệch của tầng thứ ℓ ,
- $\sigma_\ell : \mathbb{R} \rightarrow \mathbb{R}$ là hàm kích hoạt áp dụng trên từng phần tử của z_ℓ .

Xác suất dự đoán của mẫu đầu vào x thuộc về lớp c được tính bằng hàm softmax

$$P(c|x) = \frac{\exp(o_c)}{\sum_{c'=1}^C \exp(o_{c'})}, \quad c = 1, 2, \dots, C. \quad (1)$$

Với nhãn $y \in \{1, 2, \dots, C\}$, giả sử ta có hàm mất mát $L(o, y)$ là hàm entropy chéo

$$L(o, y) = - \sum_{c=1}^C \mathbb{I}(y=c) \log P(c|x), \quad (2)$$

Hãy thực hiện các bước sau

- Viết công thức tính đạo hàm của hàm mất mát theo đầu ra o_a , tức là $\frac{\partial L}{\partial o_a}$. Sau đó, viết ở dạng véc-tơ $\delta_L = \nabla_{o_a} L$.
- Viết công thức tính đạo hàm của hàm mất mát đối với đầu ra thứ a của tầng thứ ℓ , tức là $\frac{\partial L}{\partial o_{\ell,a}}$ dựa trên đạo hàm của hàm mất mát đối với đầu ra tầng thứ $\ell + 1$. Từ đó, viết ở dạng véc-tơ $\delta_\ell = \nabla_{o_\ell} L$.
- Viết công thức tính đạo hàm của hàm mất mát đối với trọng số $W_{\ell,a,b}$ và độ lệch $b_{\ell,a}$ của tầng thứ ℓ , tức là $\frac{\partial L}{\partial W_{\ell,a,b}}$ và $\frac{\partial L}{\partial b_{\ell,a}}$. Sau đó, viết ở dạng ma trận và véc-tơ
$$\nabla_{W_\ell} L \in \mathbb{R}^{d_\ell \times d_{\ell-1}},$$

$$\nabla_{b_\ell} L \in \mathbb{R}^{d_\ell}.$$
- Viết công thức cập nhật tham số W_ℓ và b_ℓ sử dụng thuật toán SGD (kích thước lô bằng 1) với tốc độ học là η .

Bài 2

Hãy sử dụng các thuật toán phân lớp sau

- Hồi quy logistic (Logistic Regression),
- Cây quyết định (Decision Tree),
- K-láng giềng gần nhất (K-Nearest Neighbors),
- Mạng nơ-ron nhân tạo (Artificial Neural Network).

để phân lớp tập dữ liệu Digits (có sẵn trong thư viện `sklearn` của Python). Thực hiện các bước sau

- Chia ngẫu nhiên tập dữ liệu thành tập huấn luyện (train), tập kiểm thử (validation) và tập kiểm tra (test) với tỷ lệ 60% - 20% - 20%. Lưu ý, sử dụng hạt giống ngẫu nhiên cố định để chia tập dữ liệu.

2. Huấn luyện từng mô hình trên tập huấn luyện.
3. Tối ưu siêu tham số của từng mô hình sử dụng tập kiểm thử.
4. Khi chọn được siêu tham số tốt nhất, đánh giá độ chính xác của từng mô hình trên tập kiểm tra.
5. So sánh kết quả thu được từ các mô hình.

Yêu cầu đầu ra:

1. Mã nguồn (code) thực hiện các bước trên. Gửi link GitHub của file mã nguồn bằng IPython Notebook (.ipynb).
2. Một bảng tổng hợp kết quả độ chính xác của từng mô hình trên tập kiểm tra. Trong đó mỗi dòng ứng với một mô hình và thông tin về bộ siêu tham số tốt nhất tìm được. Các cột (11 cột) gồm có
 - Độ chính xác trên tập kiểm tra (Accuracy).
 - Độ chính xác trên tập kiểm tra của từng lớp (Recall theo lớp).