TOÁN CHO KHOA HỌC MÁY TÍNH

HỒI QUY LOGISTIC

TS. Lương Ngọc Hoàng

Nội dung

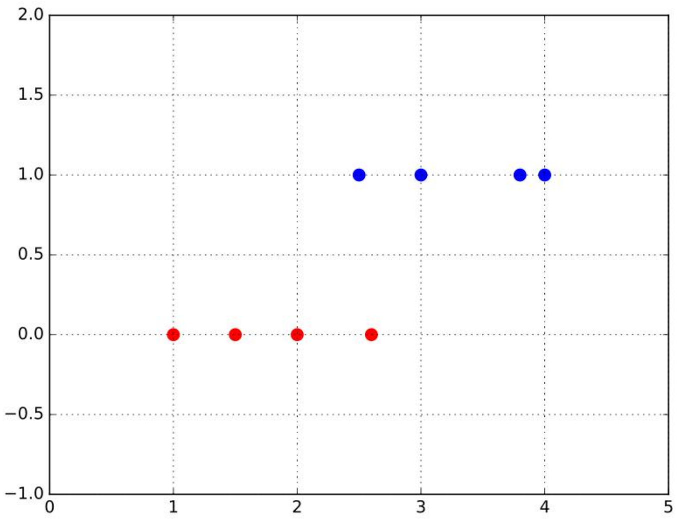
1. Giới thiệu
2. Hàm mất mát của hồi quy logistic
3. Huấn luyện mô hình hồi quy logistic

GIỚI THIỆU

INTRODUCTION

Ví dụ

Xây dựng một mô hình dự đoán kết quả thi của sinh viên (đậu / rớt) dựa vào số giờ sinh viên bỏ ra để ôn bài trước ngày thi.




```
x_train = np.array([[1], [1.5], [2], [2.5], [2.6], [3], [3.8], [4]])
y_train = np.array([0, 0, 0, 1, 0, 1, 1, 1])
```

Tập dữ liệu (Dataset)

Thu thập một tập dữ liệu (dataset) gồm N điểm dữ liệu (data points) $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ của N sinh viên với:

- $x^{(i)} \in \mathbb{R}$ là đặc trưng đầu vào (input feature) – số giờ sinh viên i bỏ ra để ôn bài trước ngày thi.
- $y^{)i(} \in \{0,1\}$ là giá trị nhãn (label) – sinh viên i đậu (nhãn 1) hay rớt (nhãn 0).

Ví dụ - Hồi quy tuyến tính?

Xây dựng một mô hình dự đoán kết quả thi của sinh viên (đậu / rớt) dựa vào số giờ sinh viên bỏ ra để ôn bài trước ngày thi.

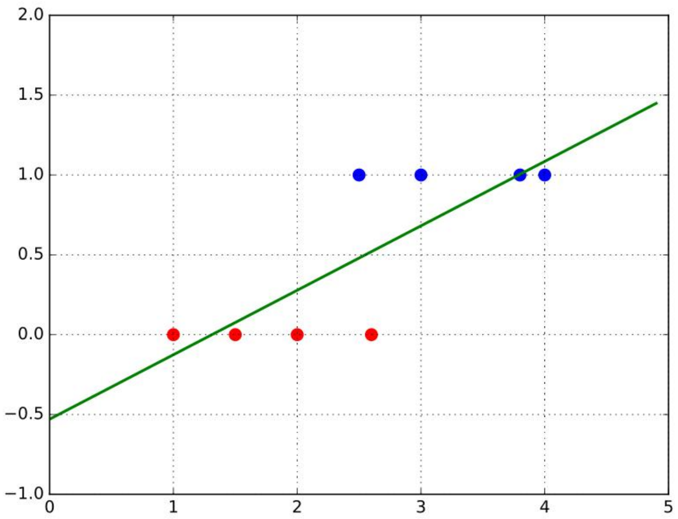
Có thể giải quyết bài toán trên với Hồi quy tuyến tính?

Sử dụng mô hình hồi quy tuyến tính đơn giản với một đặc trưng đầu vào:

$$\hat{y}^{(i)} = f_{\boldsymbol{\theta}}\left(x^{(i)}\right) = wx^{(i)} + b$$

Mô hình này có những vấn đề gì?

- Ta cần dự đoán nhãn (label) $y^{(i)} \in \{0,1\}$ của mỗi điểm dữ liệu chứ không phải một giá trị số thực $\in \mathbb{R}$.
- Diễn giải các giá trị dự đoán $\hat{y}^{(i)} \in \mathbb{R}$ như thế nào? Trong trường hợp bài toán có nhiều hơn 2 nhãn?
- Diễn giải các giá trị dự đoán $\hat{y}^{(i)} \notin [0,1]$?




```
from sklearn.linear_model import LinearRegression
lin_model = LinearRegression()
lin_model.fit(x_train, y_train)

xs = np.arange(0, 5, 0.1)
xs = np.expand_dims(xs, axis=1)
ys_lin = lin_model.predict(xs)
```

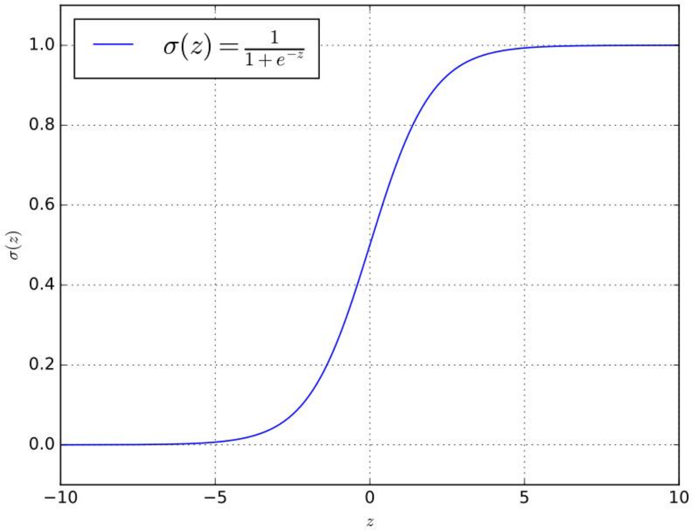
Ví dụ - Định nghĩa lại bài toán

Mô hình hồi quy tuyến tính:

$$\hat{y}^{(i)} = f_{\boldsymbol{\theta}}\left(x^{(i)}\right) = wx^{(i)} + b$$

Giải pháp:

- Thay vì dự đoán nhãn ($y^{(i)} \in \{0,1\}$) thì dự đoán xác suất thi đậu nếu sinh viên bỏ ra $x^{(i)}$giờ để ôn bài → dự đoán xác suất điểm dữ liệu với đặc trưng $x^{(i)}$ thuộc về lớp có nhãn 1 là bao nhiêu?
- Thay đổi miền giá trị đầu ra $\hat{y}^{(i)} \in \mathbb{R}$ thành $\hat{y}^{(i)} \in [0,1]$.




```
import matplotlib.pyplot as plt
sigmoid = lambda z: 1/(1+np.exp(-z))

plt.grid()
plt.xlim(left=-10.0, right=10.0)
plt.ylim(bottom=-0.1, top=1.1)
zs = np.arange(-10, 10, 0.001)
plt.plot(zs, sigmoid(zs))
```

Mô hình đề xuất:

$$\hat{y}^{(i)} = f_{\boldsymbol{\theta}}\left(x^{(i)}\right) = \sigma(wx^{(i)} + b)$$

với:

$$\sigma(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-(wx^{(i)}+b)}}$$

Hàm $\sigma(z)$ gọi là hàm logistic.

Ví dụ - Hồi quy logistic

Mô hình hồi quy logistic:

$$\hat{y}^{(i)} = f_{\boldsymbol{\theta}}\left(x^{(i)}\right) = \sigma\left(wx^{(i)} + b\right)$$

với:

$$\sigma(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-(wx^{(i)} + b)}}$$

```
0 1.1546058654785156
1000 0.3521612584590912
2000 0.2992682158946991
3000 0.2770415842533116
4000 0.2647102475166321
5000 0.25683245062828064
6000 0.25135499238967896
7000 0.24732603132724762
8000 0.24424152076244354
9000 0.24180850386619568
10000 0.2398447121620178
```

```python
import torch
import torch.nn as nn
import torch.optim as optim

class Model(nn.Module):
    def __init__(self):
        super(Model, self).__init__()
        self.linear = nn.Linear(1, 1)

    def forward(self, x):
        y_pred = nn.functional.sigmoid(self.linear(x))
        return y_pred

log_model = Model()
criterion = nn.BCELoss()
optimizer = optim.SGD(log_model.parameters(), lr=0.05)
```

```python
x_data = torch.from_numpy(x_train.astype(np.float32))
y_data = torch.from_numpy(np.expand_dims(y_train, axis=1).astype(np.float32))
for epoch in range(10001):
    y_pred = log_model(x_data)

    loss = criterion(y_pred, y_data)
    if epoch % 1000 == 0:
        print(epoch, loss.item())

    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
```

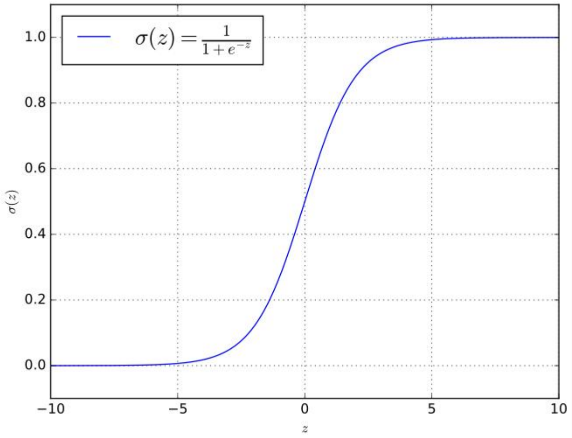
Ví dụ - Hồi quy logistic

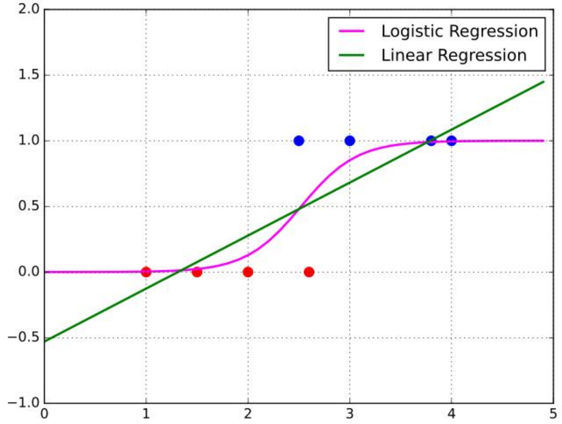
Mô hình hồi quy logistic:

$$\hat{y}^{(i)} = f_{\boldsymbol{\theta}}\left(x^{(i)}\right) = \sigma\left(wx^{(i)} + b\right)$$

với:

$$\sigma(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-(wx^{(i)}+b)}}$$






```
log_model.eval()
with torch.no_grad():
    hour_var = torch.Tensor([[1.5]])
    print('Probability to pass the exam: %.4f' % log_model(hour_var).item())
    print('Predict to pass the exam: ', log_model(hour_var).item() > 0.5)

    hour_var = torch.Tensor([[8.7]])
    print('Probability to pass the exam: ', log_model(hour_var).item())
    print('Predict to pass the exam: ', log_model(hour_var).item() > 0.5)
```

```
Probability to pass the exam: 0.0228
Predict to pass the exam:  False
Probability to pass the exam: 1.0000
Predict to pass the exam:  True
```

```
plt.ylim(-1, 2)
plt.xlim(0,5)
plt.grid()
plt.scatter(x_train[y_train>0], y_train[y_train>0], s=70, color='blue');
plt.scatter(x_train[y_train<1], y_train[y_train<1], s=70, color='red');

xs = np.arange(0, 5, 0.1)
xs = np.expand_dims(xs, axis=1)
with torch.no_grad():
    xs_tensor = torch.from_numpy(xs.astype(np.float32))
    ys_tensor = log_model(xs_tensor)
    plt.plot(xs_tensor, ys_tensor, color='magenta', lw=2, label='Logistic ')
    plt.plot(xs_tensor, ys_lin, color='green', lw=2, label='Linear')
    plt.legend()
```

HÀM MẤT MÁT CỦA HỒI QUY LOGISTIC

LOSS FUNCTION OF LOGISTIC REGRESSION

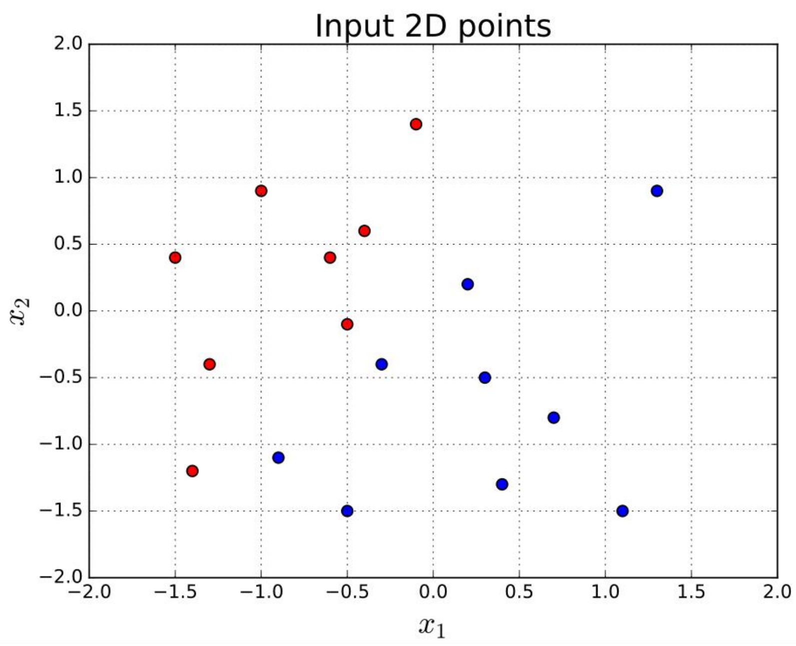
Hồi quy logistic – Phân lớp nhị phân

Bài toán phân lớp nhị phân (binary classification):

Xây dựng một mô hình để phân biệt các điểm dữ liệu thành 2 lớp đỏ và xanh.

Với mỗi điểm dữ liệu đầu vào $\boldsymbol{x} = (x_1, x_2)$, ta muốn đầu ra của mô hình dự đoán nhãn (label) của điểm dữ liệu là nhãn 0 (lớp đỏ) hay nhãn 1 (lớp xanh).

Ta muốn dự đoán xác suất điểm dữ liệu $\boldsymbol{x} = (x_1, x_2)$ thuộc về lớp xanh (nhãn 1) là bao nhiêu.



Ví dụ:

- Dựa trên các chỉ số của bệnh nhân, dự đoán xác suất bệnh nhân sẽ bị đột quỵ.
- Dựa trên thông tin của khách hàng, dự đoán khả năng thanh đoán đúng hạn thẻ tín dụng để xét duyệt hồ sơ cấp thẻ tín dụng.

Hồi quy logistic – Phân lớp nhị phân

Dựa trên thông tin của khách hàng, dự đoán khả năng thanh đoán đúng hạn thẻ tín dụng để xét duyệt hồ sơ cấp thẻ tín dụng.

Xây dựng công thức để chấm điểm thông tin khách hàng với một hàm tuyến tính

$$\text{score} = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

- Nếu đặc trưng x_1 là mức lương, thì w_1 nên là một giá trị dương.
- Nếu đặc trưng x_2 là số tiền nợ, thì w_2 nên là một giá trị âm.
- Nếu đặc trưng x_3 là cân nặng, thì w_3 nên là một giá trị ≈ 0 vì đây là một đặc trưng không quan trọng.
- Giá trị b thể hiện khuynh hướng thiên về lớp nào hơn (bias).

➢ Nếu điểm $\text{score} > 0$, đây là một hồ sơ tốt (khả năng được cấp thẻ lớn hơn 50%).
➢ Nếu điểm $\text{score} < 0$, đây là một hồ sơ xấu (khả năng được cấp thẻ nhỏ hơn 50%).
➢ Ta cần chuyển điểm score thành giá trị xác suất (probability).

Hồi quy logistic – Phân lớp nhị phân

$$\text{score} = w_1 x_1 + w_2 x_2 + b$$

Ta muốn dự đoán xác suất điểm dữ liệu $x = (x_1, x_2)$ thuộc về lớp xanh (nhãn 1) là bao nhiêu.

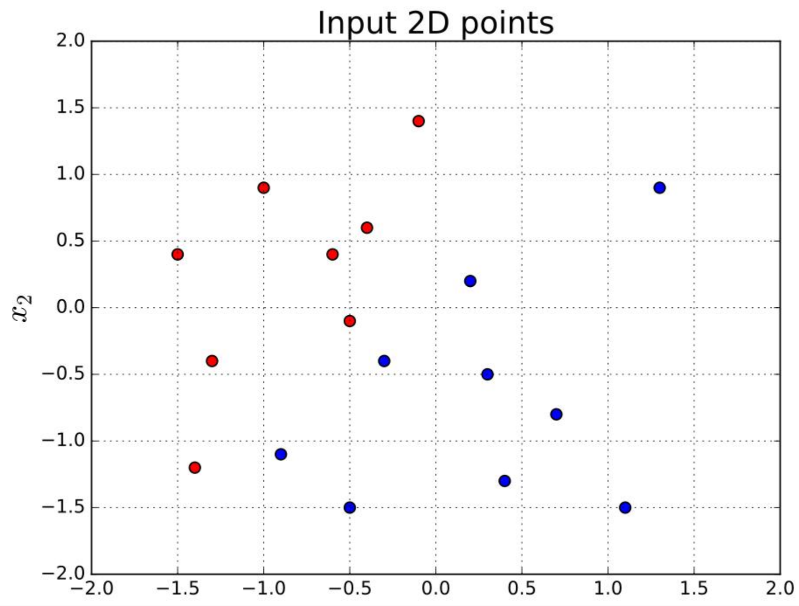
$$a = g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

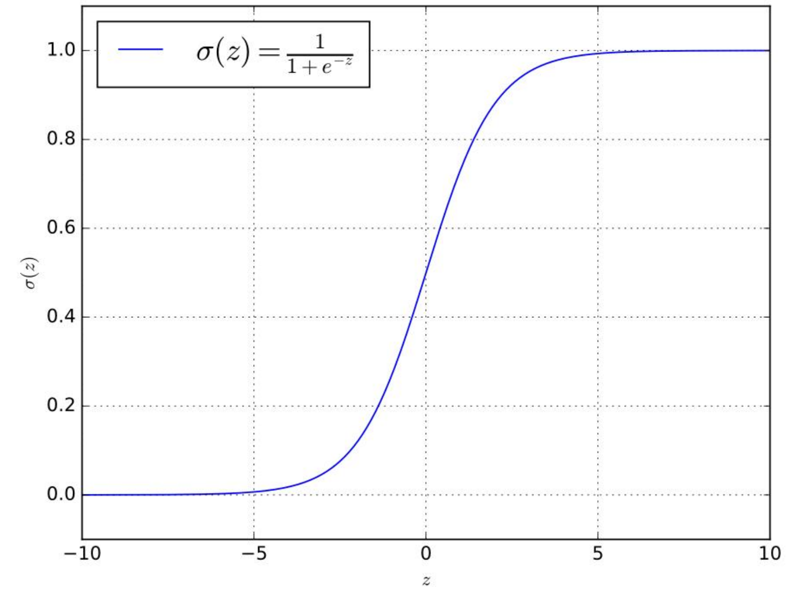
với $z = \text{score} = w_1 x_1 + w_2 x_2 + b = \boldsymbol{w}^T \boldsymbol{x} + b$.

- Ta có $0 < g(z) < 1$ với mọi $z \in \mathbb{R}$.
- Do đó, $0 < g(\boldsymbol{x}) < 1$ với mọi $\boldsymbol{x} \in \mathbb{R}^2$.
- a là xác suất dự đoán của mô hình cho điểm dữ liệu $\boldsymbol{x}$ thuộc về lớp xanh (nhãn 1).

Ta có thể dự đoán nhãn (label) của điểm $\boldsymbol{x} = (x_1, x_2)$

$$\text{Nhãn dự đoán} = \begin{cases} 0, & nếu\ g(\boldsymbol{x}) < 0.5 \\ 1, & nếu\ g(\boldsymbol{x}) \geq 0.5 \end{cases}$$





Hồi quy logistic – Phân lớp nhị phân

Ta muốn dự đoán xác suất điểm dữ liệu $x = (x_1, x_2)$ thuộc về lớp xanh (nhãn 1) là bao nhiêu.

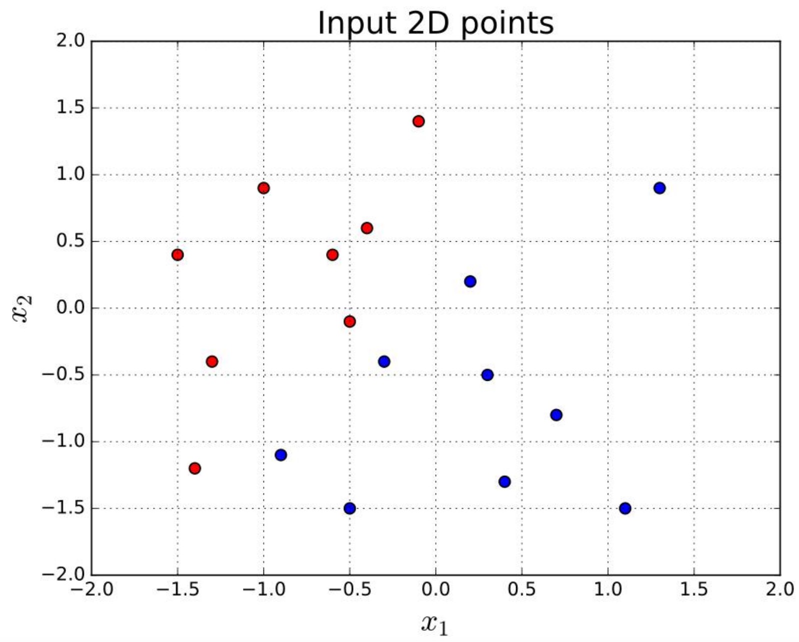
$$a = g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

với $z = \text{score} = w_1 x_1 + w_2 x_2 + b = \boldsymbol{w}^T \boldsymbol{x} + b.$

```
x = np.array([1.1, -0.6])
print(w.dot(x)+b)
print(sigmoid(w.dot(x)+b))
11.485000000000001 0.9999897169153853
```

```
x = np.array([-1.5, 0.0])
print(w.dot(x)+b)
print(sigmoid(w.dot(x)+b))
-7.505000000000001 0.0005500231583811739
```

```
x = np.array([-0.3, -0.0])
print(w.dot(x)+b)
print(sigmoid(w.dot(x)+b))
0.09099999999999997 0.5227343135939515
```



Giả sử sau khi huấn luyện mô hình trên và ta có các tham số mô hình như sau:

$w_1 = 6.33, w_2 =- 4.22,$

$b = 1.99$

Hồi quy logistic – Hàm mất mát

Dự đoán xác suất $\boldsymbol{x} = (x_1, x_2)$ thuộc về lớp xanh (nhãn 1)

$$a = g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

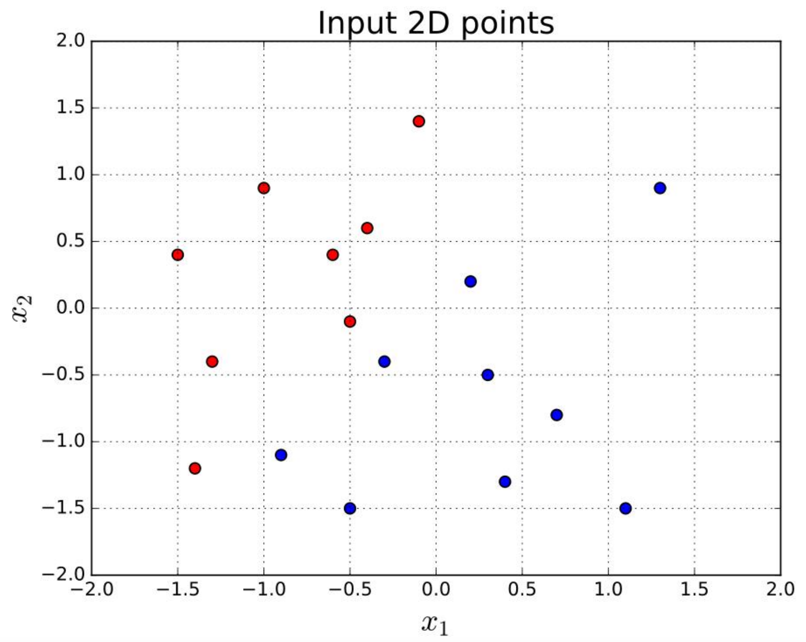
với $z = \text{score} = w_1 x_1 + w_2 x_2 + b = \boldsymbol{w}^T \boldsymbol{x} + b$.

- Ta cần học giá trị $\boldsymbol{w} = (w_1, w_2)$ và b để phân biệt tốt các điểm dữ liệu xanh và đỏ trong tập dữ liệu huấn luyện.
- Giả sử ta có 2 điểm dữ liệu như sau:

$\boldsymbol{x}^{(1)} = [-0.1, 1.4]$ và $y^{(1)} = 0$

$\boldsymbol{x}^{(2)} = [1.3, 0.9]$ và $y^{(2)} = 1$

- Ta muốn học được một bộ phân lớp (classifier) sao cho:
  - ➢ Trả về xác suất a rất lớn đối với điểm dữ liệu có $y = 1$.
  - ➢ Trả về xác suất a rất nhỏ đối với điểm dữ liệu có $y = 0$.
- Ta muốn có giá trị xác suất a gần với y.
  - ➢ Nếu $y = 1$ thì ta muốn *cực đại hóa* a.
  - ➢ Nếu $y = 0$ thì ta muốn *cực đại hóa* $1 - a$.
- Ta muốn *cực đại hóa* $a^y \cdot (1 - a)^{(1-y)}$.



Với mỗi điểm dữ liệu đầu vào $\boldsymbol{x} = (x_1, x_2)$, ta muốn đầu ra của mô hình dự đoán nhãn (label) của điểm dữ liệu là nhãn 0 (lớp đỏ) hay nhãn 1 (lớp xanh).

Hồi quy logistic – Hàm mất mát

Dự đoán xác suất $\boldsymbol{x} = (x_1, x_2)$ thuộc về lớp xanh (nhãn 1)

$$a = g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

với $z = \text{score} = w_1 x_1 + w_2 x_2 + b = \boldsymbol{w}^T \boldsymbol{x} + b$.

- Học giá trị $\boldsymbol{w} = (w_1, w_2)$ và b để *cực đại hóa* $a^y \cdot (1-a)^{(1-y)}$.

$$\widehat{w}, \widehat{b} = \underset{\boldsymbol{w},b}{\operatorname{argmax}}\, a^y \cdot (1-a)^{(1-y)}$$

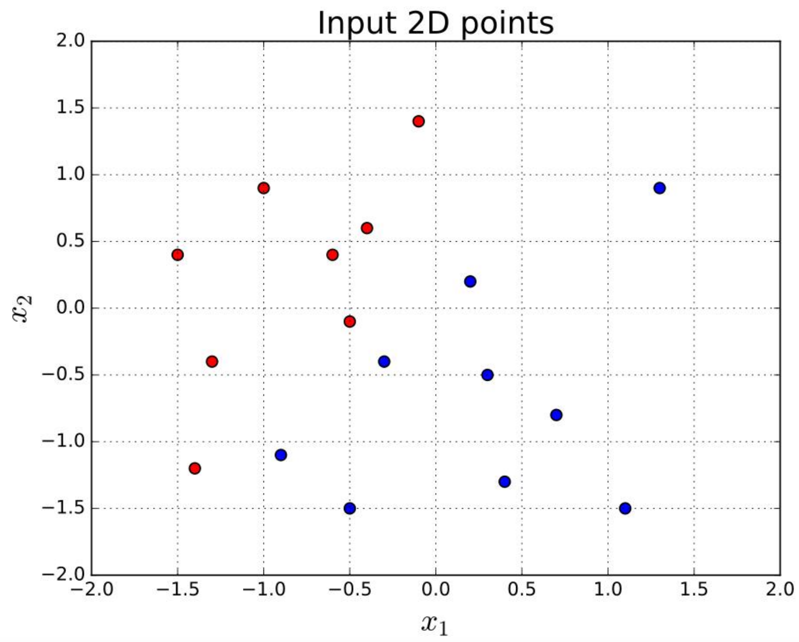
- Cực đại hóa biểu thức trên tương đương với:

$$\begin{aligned}\widehat{w}, \widehat{b} &= \underset{\boldsymbol{w},b}{\operatorname{argmax}} \log\left(a^y \cdot (1-a)^{(1-y)}\right) \\ &= \underset{\boldsymbol{w},b}{\operatorname{argmax}} \left[\log\left(a^y\right) + \log\left((1-a)^{(1-y)}\right)\right] \\ &= \underset{\boldsymbol{w},b}{\operatorname{argmax}} \left[y\log(a) + (1-y)\log(1-a)\right]\end{aligned}$$

- Ta có thể đổi về bài toán *cực tiểu hóa*:

$$\widehat{w}, \widehat{b} = \underset{\boldsymbol{w},b}{\operatorname{argmin}} - \left[y\log(a) + (1-y)\log(1-a)\right] = \underset{\boldsymbol{w},b}{\operatorname{argmin}}\, L(\boldsymbol{w}, b)$$

- Hàm L là *hàm mất mát (loss function)* cho 1 điểm dữ liệu.



Với mỗi điểm dữ liệu đầu vào $\boldsymbol{x} = (x_1, x_2)$, ta muốn đầu ra của mô hình dự đoán nhãn (label) của điểm dữ liệu là nhãn 0 (lớp đỏ) hay nhãn 1 (lớp xanh).

Hồi quy logistic – Ước lượng hợp lý cực đại

Dự đoán xác suất $\boldsymbol{x} = (x_1, x_2)$ thuộc về lớp xanh (nhãn 1)

$$a = g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

với $z = \text{score} = w_1 x_1 + w_2 x_2 + b = \boldsymbol{w}^T \boldsymbol{x} + b$.

Học giá trị $\boldsymbol{w} = (w_1, w_2)$ và b để *cực đại hóa* $a^y \cdot (1-a)^{(1-y)}$ với mỗi điểm dữ liệu

$$\widehat{w}, \widehat{b} = \underset{\boldsymbol{w},b}{\operatorname{argmax}}\, a^y \cdot (1-a)^{(1-y)}$$

- Giả sử ta có 3 điểm dữ liệu có nhãn $y^{(1)} = y^{(3)} = 1$ và $y^{(2)} = 0$.
- Ta có 2 mô hình với dự đoán xác suất 3 điểm dữ liệu trên thuộc về lớp xanh (nhãn 1) như sau:
  - Bộ phân lớp (classifier) 1: $0.9,\ 0.4,\ 0.8$
  - Bộ phân lớp (classifier) 2: $0.7,\ 0.7,\ 0.7$
- ***Mô hình (bộ phân lớp) nào tốt hơn?***
- Ước lượng hợp lý cực đại (maximum likelihood estimation – MLE) xác định khả năng của mỗi mô hình như sau:
  - Bộ phân lớp 1: $0.9^1(1-0.9)^{1-1} \times 0.4^0(1-0.4)^{1-0} \times 0.8^1(1-0.8)^{1-1} = 0.432$
  - Bộ phân lớp 2: $0.7^1(1-0.7)^{1-1} \times 0.7^0(1-0.7)^{1-0} \times 0.7^1(1-0.7)^{1-1} = 0.147$
- Bộ phân lớp 1 có khả năng dự đoán đúng nhãn của 3 điểm dữ liệu cao hơn.

Hồi quy logistic – Ước lượng hợp lý cực đại

Dự đoán xác suất $\boldsymbol{x} = (x_1, x_2)$ thuộc về lớp xanh (nhãn 1)

$$a = g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

với $z = \text{score} = w_1 x_1 + w_2 x_2 + b = \boldsymbol{w}^T \boldsymbol{x} + b$.

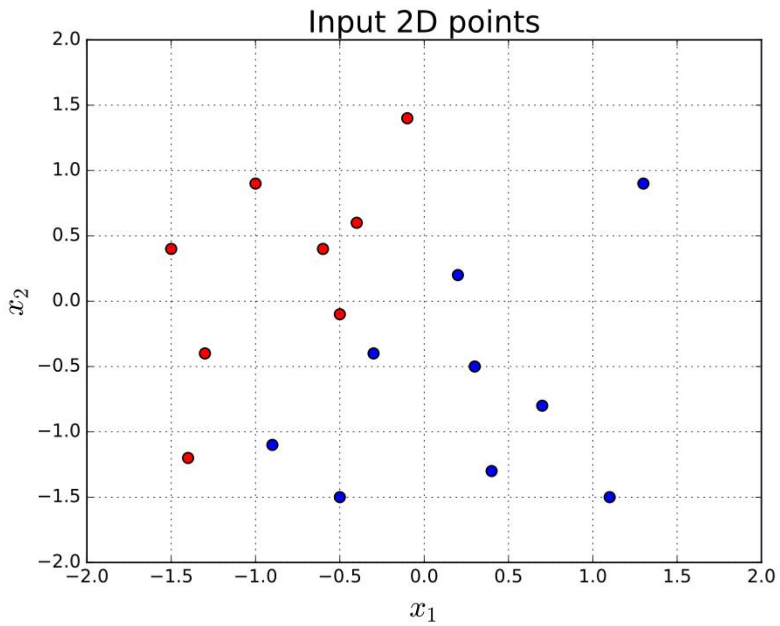
- Học giá trị $\boldsymbol{w} = (w_1, w_2)$ và b để *cực đại hóa* $a^y \cdot (1 - a)^{(1-y)}$ cho mỗi điểm dữ liệu.
- Xét toàn bộ tập dữ liệu gồm N điểm dữ liệu, ước lượng hợp lý cực đại (MLE) cực đại hóa hàm mục tiêu sau:

$$\hat{w}, \hat{b} = \underset{\boldsymbol{w},b}{\operatorname{argmax}} \prod_{i=1}^{N} {a^{(i)}}^{y^{(i)}} \cdot \left(1 - a^{(i)}\right)^{(1-y^{(i)})}$$

$$= \underset{\boldsymbol{w},b}{\operatorname{argmax}} \log \prod_{i=1}^{N} {a^{(i)}}^{y^{(i)}} \cdot \left(1 - a^{(i)}\right)^{(1-y^{(i)})}$$

$$= \underset{\boldsymbol{w},b}{\operatorname{argmax}} \sum_{i=1}^{N} \left[\log\left({a^{(i)}}^{y^{(i)}}\right) + \log\left(\left(1 - a^{(i)}\right)^{(1-y^{(i)})}\right)\right]$$

$$= \underset{\boldsymbol{w},b}{\operatorname{argmax}} \sum_{i=1}^{N} \left[y^{(i)} \log\left(a^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - a^{(i)}\right)\right]$$



Với mỗi điểm dữ liệu đầu vào $\boldsymbol{x} = (x_1, x_2)$, ta muốn đầu ra của mô hình dự đoán nhãn (label) của điểm dữ liệu là nhãn 0 (lớp đỏ) hay nhãn 1 (lớp xanh).

Hồi quy logistic – Ước lượng hợp lý cực đại

- Xét toàn bộ tập dữ liệu gồm N điểm dữ liệu:

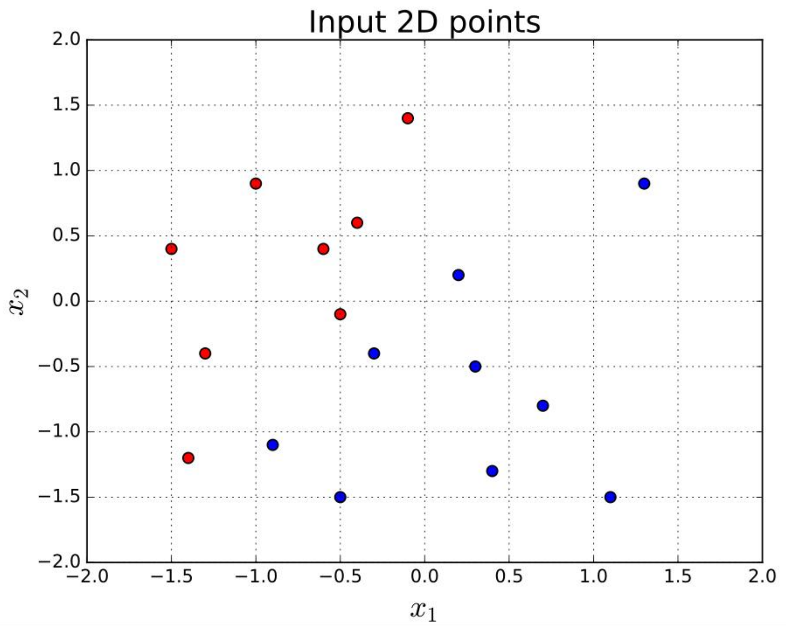
$$\hat{w}, \hat{b} = \underset{\boldsymbol{w},b}{\operatorname{argmax}} \sum_{i=1}^{N} \left[y^{(i)} \log\left(a^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - a^{(i)}\right) \right]$$

$$= \underset{\boldsymbol{w},b}{\arg\min} - \sum_{i=1}^{N} \left[y^{(i)} \log\left(a^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - a^{(i)}\right) \right]$$

$$= \underset{\boldsymbol{w},b}{\arg\min} \sum_{i=1}^{N} L(w, b, \boldsymbol{x}^{(i)}, y^{(i)}) = \underset{\boldsymbol{w},b}{\arg\min}\ J(\boldsymbol{w}, b)$$

với

$$a^{(i)} = \frac{1}{1 + e^{-z^{(i)}}} = \frac{1}{1 + e^{-\left(w_1 x_1^{(i)} + w_2 x_2^{(i)} + b\right)}}$$



Với mỗi điểm dữ liệu đầu vào $\boldsymbol{x} = (x_1, x_2)$, ta muốn đầu ra của mô hình dự đoán nhãn (label) của điểm dữ liệu là nhãn 0 (lớp đỏ) hay nhãn 1 (lớp xanh).

Ta có thể định nghĩa:

- $J(\boldsymbol{w}, b)$ là hàm chi phí (cost function) khi xét tất cả các điểm dữ liệu huấn luyện.
- $L(\boldsymbol{w}, b, \boldsymbol{x}^{(i)}, y^{(i)})$ là hàm mất mát (loss function) khi xét một điểm dữ liệu huấn luyện $(\boldsymbol{x}^{(i)}, y^{(i)})$. ***Binary cross-entropy (BCE) loss.***

HUẤN LUYỆN MÔ HÌNH HỒI QUY LOGISTIC

TRAINING A LOGISTIC REGRESSION MODEL

Hồi quy logistic – Đạo hàm

Dự đoán xác suất $\boldsymbol{x} = (x_1, x_2)$ thuộc về lớp xanh (nhãn 1)

$$a = g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

với $z = \text{score} = w_1 x_1 + w_2 x_2 + b = \boldsymbol{w}^T \boldsymbol{x} + b$.

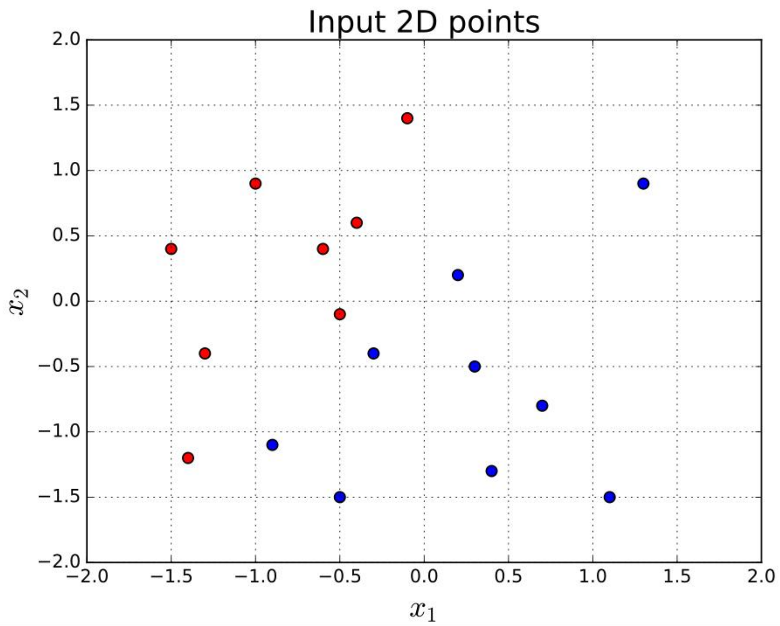
- Học giá trị $\boldsymbol{w} = (w_1, w_2)$ và b để *cực tiểu hóa*

$$L = -(y \log a + (1 - y) \log (1 - a))$$

cho mỗi điểm dữ liệu.

- Để áp dụng thuật toán Gradient Descent, ta cần tính được các đạo hàm riêng của hàm mất mát theo các tham số w_1, w_2, b :

$$\frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2}, \frac{\partial L}{\partial b}$$



Với mỗi điểm dữ liệu đầu vào $\boldsymbol{x} = (x_1, x_2)$, ta muốn đầu ra của mô hình dự đoán nhãn (label) của điểm dữ liệu là nhãn 0 (lớp đỏ) hay nhãn 1 (lớp xanh).

Quy tắc dây chuyền (chain rule)

$$\begin{array}{l} x_1 \searrow \\ x_2 \searrow \\ w_1 \rightarrow \\ w_2 \nearrow \\ b \nearrow \end{array} z = x_1 w_1 + x_2 w_2 + b \longrightarrow a = g(z) \longrightarrow \mathrm{L} = -(y\log(a) + (1-y)\log(1-a))$$

$$z = w_1 x_1 + w_2 x_2 + b \qquad a = g(z) = \frac{1}{1+e^{-z}} \qquad L = -(y\log a + (1-y)\log(1-a)$$

Quy tắc dây chuyền:

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial a}\frac{\partial a}{\partial z}\frac{\partial z}{\partial w_1}$$

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial a}\frac{\partial a}{\partial z}\frac{\partial z}{\partial w_2}$$

$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial a}\frac{\partial a}{\partial z}\frac{\partial z}{\partial b}$$

Do đó:

$$\frac{\partial z}{\partial w_1} = x_1$$

$$\frac{\partial z}{\partial w_2} = x_2$$

$$\frac{\partial z}{\partial b} = 1$$

Do đó:

$$\frac{\partial a}{\partial z} = -\frac{(1+e^{-z})'}{(1+e^{-z})^2}$$

$$= \frac{e^{-z}}{(1+e^{-z})^2} = \frac{1+e^{-z}-1}{(1+e^{-z})^2}$$

$$= g(z) - \left(g(z)\right)^2$$

$$= a(1-a)$$

Do đó:

$$\frac{\partial L}{\partial a} = -\left(\frac{y}{a} - \frac{1-y}{1-a}\right)$$

$$= \frac{-y}{a} + \frac{1-y}{1-a}$$

Quy tắc dây chuyền (chain rule)

$$\begin{matrix} x_1 \\ x_2 \\ w_1 \\ w_2 \\ b \end{matrix} \longrightarrow z = x_1w_1 + x_2w_2 + b \longrightarrow a = g(z) \longrightarrow \mathrm{L} = -(y\log(a) + (1-y)\log(1-a))$$

$$\frac{\partial L}{\partial a}\frac{\partial a}{\partial z} = \left(\frac{-y}{a} + \frac{1-y}{1-a}\right)a(1-a)$$

$$= \frac{-y}{a}a(1-a) + \frac{1-y}{1-a}a(1-a)$$

$$= -y(1-a) + (1-y)a$$

$$= -y + ya + a - ya$$

$$= a - y$$

Quy tắc dây chuyền:

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial a}\frac{\partial a}{\partial z}\frac{\partial z}{\partial w_1} = (a-y)x_1$$

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial a}\frac{\partial a}{\partial z}\frac{\partial z}{\partial w_2} = (a-y)x_2$$

$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial a}\frac{\partial a}{\partial z}\frac{\partial z}{\partial b} = (a-y)$$

Gradient (một điểm dữ liệu)

$$\nabla L = \begin{bmatrix} \frac{\partial L}{\partial b} \\ \frac{\partial L}{\partial w_2} \\ \frac{\partial L}{\partial w_2} \end{bmatrix} = (a-y)\begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}$$

$$= (a-y)\boldsymbol{x}$$

Gradient (tập dữ liệu)

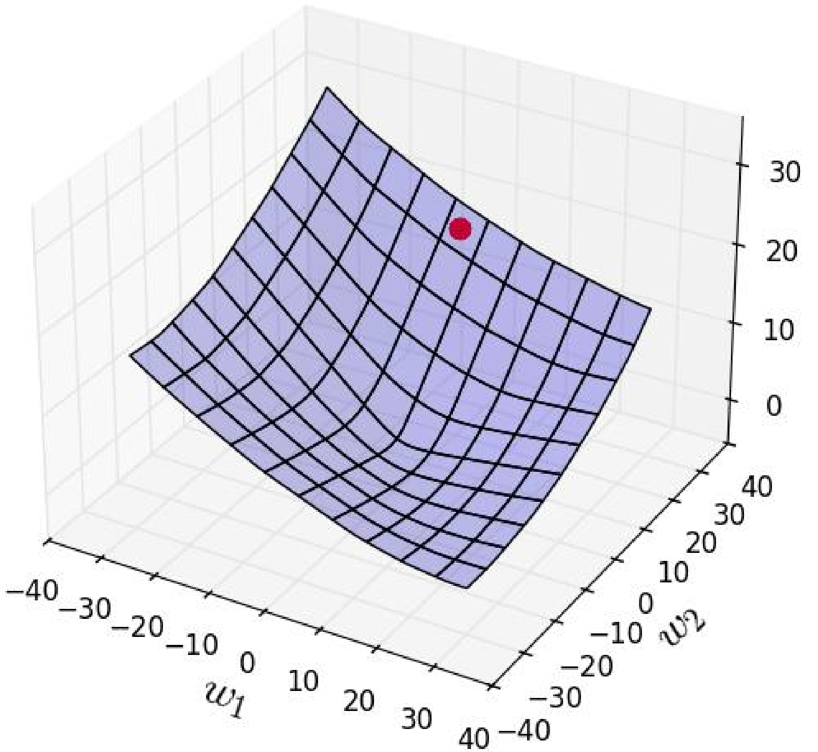
$$\nabla J = \sum_{i=1}^{N}\left(a^{(i)} - y^{(i)}\right)\boldsymbol{x}^{(i)}$$

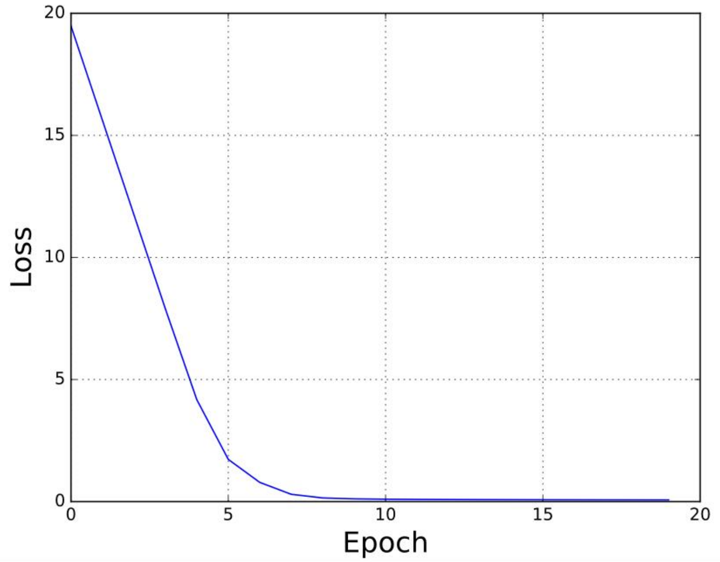
Gradient Descent

```
learning_rate = 8.0
num_epochs = 20
w = np.array([-4.0, 29.0])
b = 0.0
for epoch in range(num_epochs):
    dw = np.zeros(w.shape)
    db = 0.0
    total_loss = 0.0
    for i in range(X.shape[0]):
        # get the i-th data point and its label
        x_i = X[i,:]
        y_i = y[i]

        z_i = w.dot(x_i) + b
        a_i = sigmoid(z_i)
        loss_i = compute_loss(y_i, a_i)
        # derivatives
        dw_i = x_i * (a_i - y_i) # dL / dw_i
        db_i = a_i - y_i  # dL / db

        dw += dw_i
        db += db_i
        total_loss += loss_i
    # scale the derivatives
    dw = (1.0/X.shape[0]) * dw
    db = (1.0/X.shape[0]) * db
    total_loss = (1.0/X.shape[0]) * total_loss
    print(f'Epoch {epoch+1}, Loss = {total_loss:.2f}')
    # gradient descent
    w = w - learning_rate * dw
    b = b - learning_rate * db
```





Biên quyết định (decision boundary)

Dự đoán xác suất $\boldsymbol{x} = (x_1, x_2)$ thuộc về lớp xanh (nhãn 1)

$$a = g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

với $z = \text{score} = w_1 x_1 + w_2 x_2 + b = \boldsymbol{w}^T \boldsymbol{x} + b$.

- Ta dự đoán $\boldsymbol{x} = (x_1, x_2)$ là có nhãn 0 (lớp đỏ) nếu:

$$\frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-(w_1 x_1 + w_2 x_2 + b)}} < 0.5$$

và có nhãn 1 (lớp xanh) trong trường hợp ngược lại.

- Biên quyết định (decision boundary) của mô hình là:

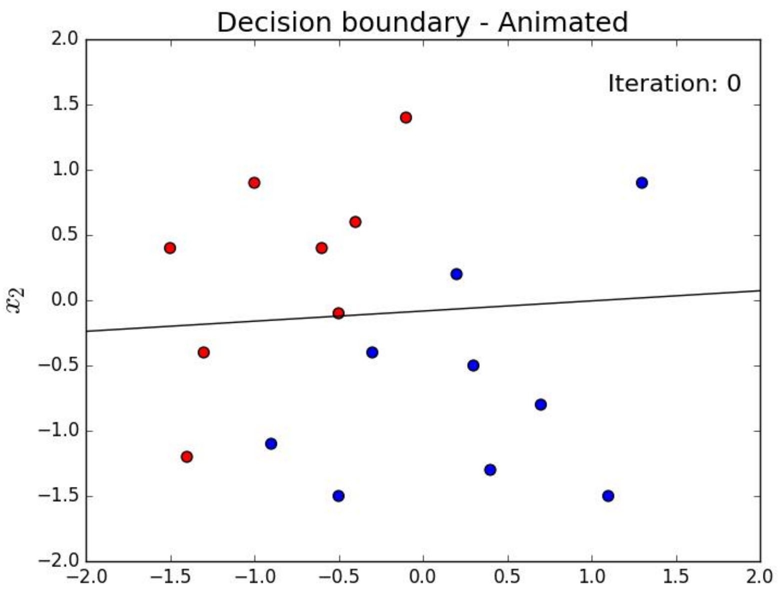
$$\frac{1}{1 + e^{-(w_1 x_1 + w_2 x_2 + b)}} = 0.5$$

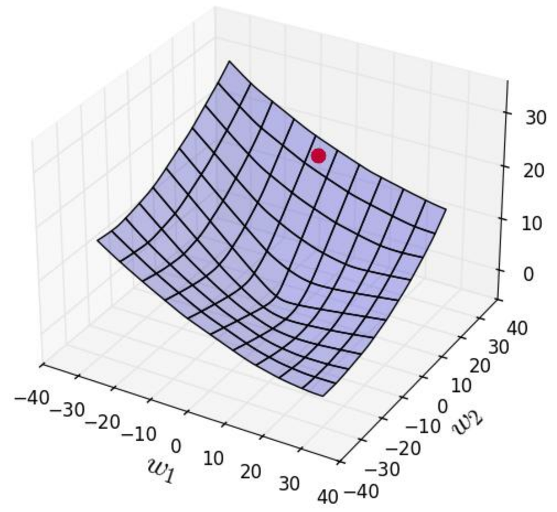
$$1 + e^{-(w_1 x_1 + w_2 x_2 + b)} = 2$$

$$e^{-(w_1 x_1 + w_2 x_2 + b)} = 1$$

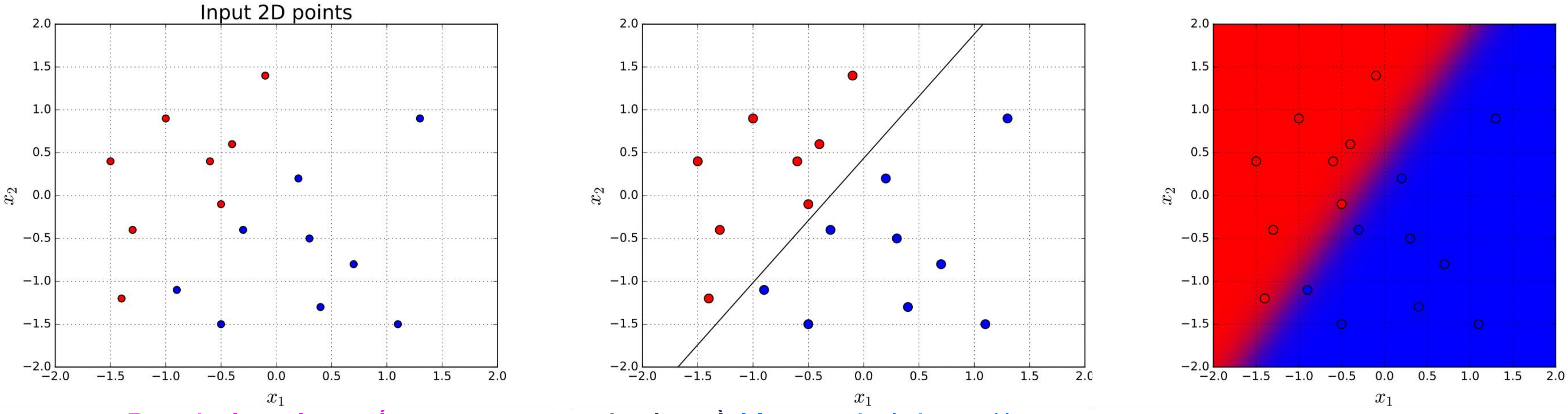
$$w_1 x_1 + w_2 x_2 + b = 0$$

$$x_2 = \frac{-w_1 x_1 - b}{w_2}$$





Biên quyết định (decision boundary)



- Dự đoán xác suất $\boldsymbol{x} = (x_1, x_2)$ thuộc về lớp xanh (nhãn 1)

$$a = g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

với $z = \text{score} = w_1 x_1 + w_2 x_2 + b = \boldsymbol{w}^T \boldsymbol{x} + b.$

- Ta dự đoán $\boldsymbol{x} = (x_1, x_2)$ là có nhãn 0 (lớp đỏ) nếu:

$$\frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-(w_1 x_1 + w_2 x_2 + b)}} < 0.5$$

và có nhãn 1 (lớp xanh) trong trường hợp ngược lại.

Biên quyết định của mô hình là:

$$x_2 = \frac{-w_1 x_1 - b}{w_2}$$