

# Exercise 7 - Tensorflow

Đặng Linh Anh

January 2021

5. (Optional) Thử chạy lại các ví dụ trên với giá trị khởi tạo cho theta lớn. Ví dụ:

```
theta = tf.Variable(tf.random.normal((5, 3), dtype=tf.float64)*5)
```

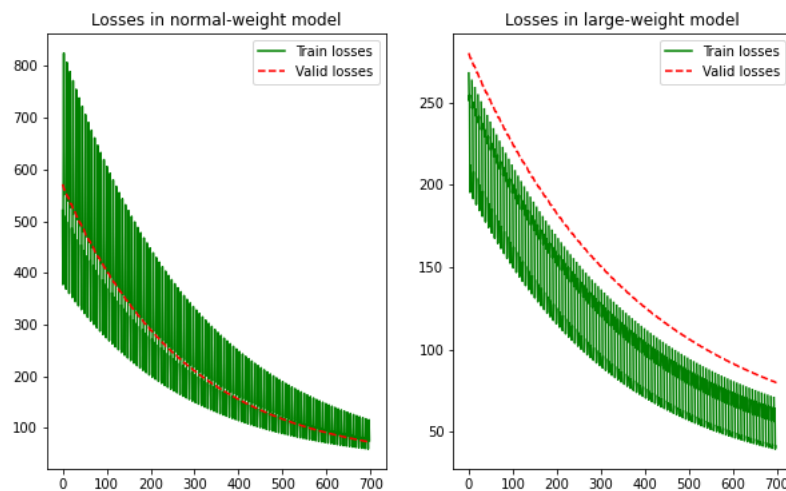
Lúc này quá trình train không còn ổn định nữa

- Tìm hiểu nguyên nhân tại sao lại có hiện tượng này.
- Đề xuất cách giải quyết.

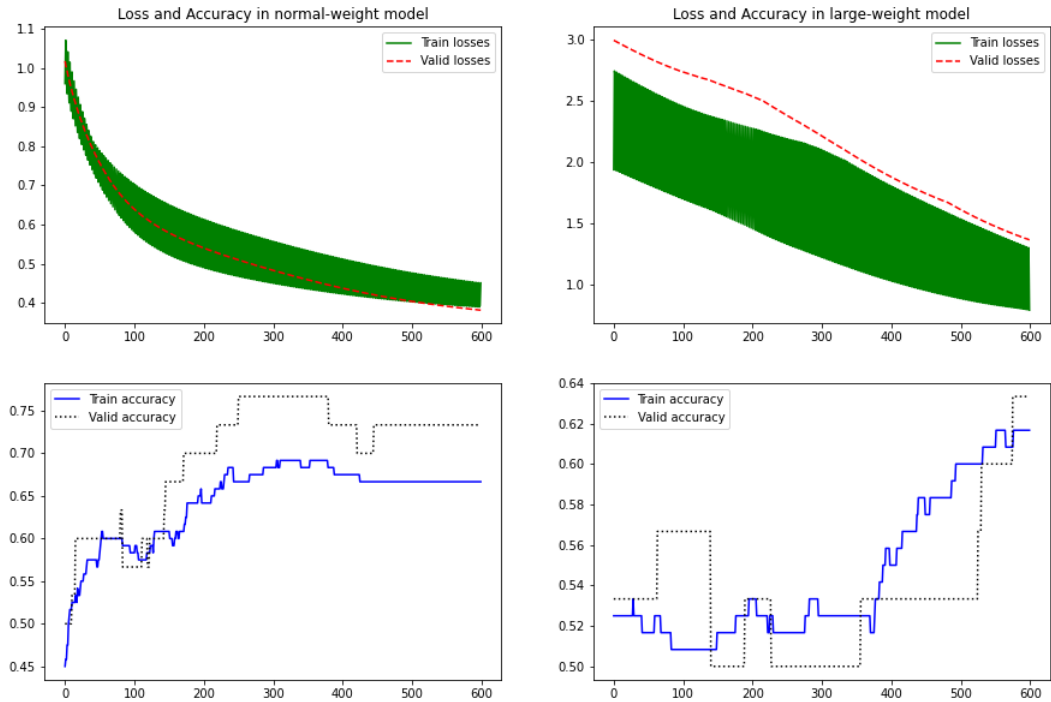
## Lời giải

Code cài đặt và chạy thử nghiệm: [LINK NOTEBOOK](#)

Dưới đây là kết quả *loss* và *accuracy* sau khi chạy.



Hình 1: Linear regression: So sánh *loss* và *accuracy* giữa 2 model



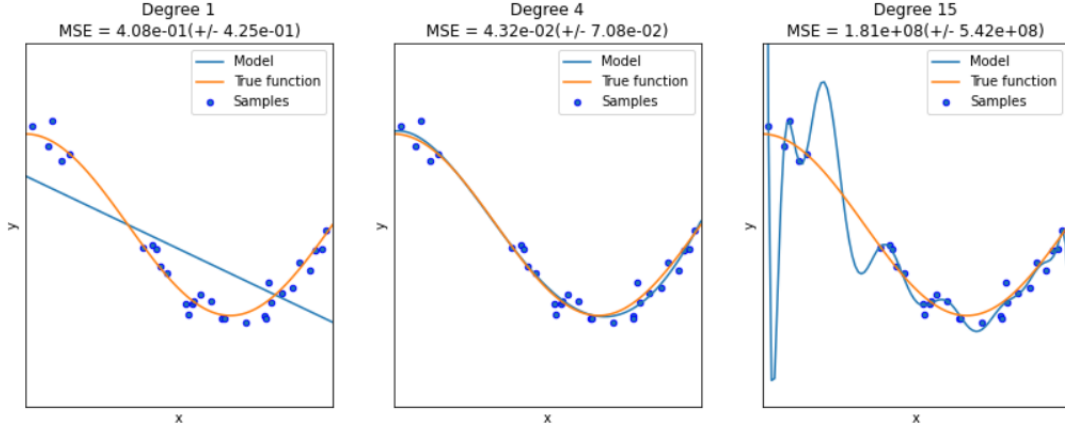
Hình 2: Logistic regression: So sánh *loss* và *accuracy* giữa 2 model

Có thể thấy, khi khởi tạo *weights* với giá trị lớn, *loss* ở tập train và tập valid không tương quan nhau (sai số xác thực giữa tập train và valid lớn và tăng dần theo *epochs*).

- Đối với linear regression, vì được train trên dataset `advertising.csv` chỉ có 3 *features* nên việc huấn luyện trên tập train và valid vẫn đem lại kết quả khá tốt.
- Đối với logistic regression, bên cạnh sự khác biệt về *loss* của 2 phần dataset đã nói trên, kết quả đem lại khá kém trên cả *loss* và *accuracy*. *Weights* lớn ảnh hưởng rất nhiều đến model thông qua hàm *sigmoid* ( $\mathbf{z} = \mathbf{\Theta}^T \mathbf{x}$  lớn  $\rightarrow \text{sigmoid}(\mathbf{z}) \approx 1 \vee 0$ ) nên kết quả đem lại kém hơn hẳn so với model dùng *weights* lớn trong linear regression.

Về nguyên nhân của sự bất ổn trong quá trình train, ta cần đề cập tới hiện tượng *Quá khớp (Overfitting)*. Trong quá trình train ta luôn mong muốn đưa ra hàm số  $f$  sao cho  $y_i \approx f(\mathbf{x}_i)$ . Trong đó, có 2 trường hợp khả dĩ đều đưa ra kết quả *loss* rất tốt là:

- Mô hình trung gian (bậc không quá cao cũng không quá thấp): Khớp tốt cho cả tập train và tập valid (tính *generalization* và độ chính xác *accuracy* cao)
- Mô hình bị quá khớp: Khớp rất tốt cho tập train, tuy nhiên với tập valid thì rất kém (*high variance*, không có tính *generalization*)



Hình 3: Mô hình chưa khớp, tốt và quá khớp

Bậc của model sau khi train được tính toán dựa trên số chiều ở đó có *weights* có giá trị đáng kể. Khi tất cả các *weights* được khởi tạo ở giá trị lớn, đồng nghĩa với việc model được khởi tạo ở bậc cao nhất. Vì vậy, model sẽ có xu hướng hội tụ tại các vị trí *quá khớp*.

Để khắc phục vấn đề này, ta có thể thêm số hạng vào hàm *loss* (còn được gọi là *regularization*):

$$\mathcal{L}_{reg}(\theta) = \mathcal{L}(\theta) + \lambda R(\theta)$$

Trong đó:

$\lambda$  (*scalar*) thường là số dương nhỏ

$R(\theta)$  là số hạng kiểm soát chỉ phụ thuộc vào  $\theta$

Xây dựng  $\mathcal{L}_{reg}$ :

$$\mathcal{L}_{reg} = \frac{1}{m} \sum_{i=1}^m \sum_{j=1}^k [-y_j^{(i)} \log(\hat{y}_j^{(i)}) + (1 - y_j^{(i)}) (1 - \log(\hat{y}_j^{(i)}))] + \frac{\lambda}{2m} \sum_{i=1}^k \sum_{j=1}^n (\Theta_{ij})^2 \quad (1)$$

(với  $n$  là số *features*,  $k$  là số *labels*)

Phần chứa hệ số  $\lambda$  sẽ khiến các tham số của mô hình không quá lớn, tránh việc phụ thuộc quá mạnh vào một đặc trưng nào đó.

Bên cạnh đó, trong những trường hợp như ví dụ đã cài đặt cho linear regression ở trên, ta có thể dùng cơ chế *kết thúc sớm* (*early stopping*). Khi huấn luyện, ta tính toán cả sai số huấn luyện và sai số xác thực, nếu sai số huấn luyện vẫn có xu hướng giảm nhưng sai số xác thực có xu hướng tăng lên thì ta kết thúc quá trình huấn luyện.