May 20, 2021

Scenario: Your regularized linear regression algorithm is making large, unexpected errors. Now what?

→ Get more training examples
→ Try smaller sets of features
→ Try getting additional features
→ Adding polynomial features (i.e. x_1^2, x_2^2, x_1x_2, etc)
→ Increase/decrease λ

☆ Instead of going by gut feeling and potentially wasting time and resources, try the following:

A machine learning diagnostic is a test that you can run to gain insight into what is/isn't working with a learning algorithm and how to improve its performance.

Train/Test Split:

A standard way to evaluate a trained hypothesis is to split a dataset 70/30.

→ If training error $J(\theta)$ is low but test error $J_{test}(\theta)$ is high, overfitting.

Notation:

$$\left.\begin{matrix} (x^{(1)}, y^{(1)}) \\ (x^{(2)}, y^{(2)}) \\ \vdots \\ (x^{(m)}, y^{(m)}) \end{matrix}\right\} \text{training set} \qquad \left.\begin{matrix} (x_{test}^{(1)}, y_{test}^{(1)}) \\ (x_{test}^{(2)}, y_{test}^{(2)}) \\ \vdots \\ (x_{test}^{(m_{test})}, y_{test}^{(m_{test})}) \end{matrix}\right\} \text{test set}$$

* m_{test} = # test examples

Train/Test Procedure:

1. Minimize $J(\theta)$ via training, learn θ.
2. Compute test set error.

↳ For classification, you can use misclassification error.

$$\hookrightarrow err(h_\theta(x), y) = \begin{cases} 1, & h_\theta(x) \geq 0.5 \\ 0, & h_\theta(x) < 0.5 \end{cases}$$

$$\text{Test error} = \frac{1}{m_{test}} \sum_{i=1}^{m_{test}} err(h_\theta(x_{test}^{(i)}), y_{test}^{(i)})$$

For Linear Regression:

$$J_{test}(\theta) = \frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} (h_\theta(x_{test}^{(i)}) - y_{test}^{(i)})^2$$

Model Selection:

$$\left.\begin{array}{lll} \text{Trained models} & & \text{Tested Errors} \\ 1.\ h_\theta(x) = \theta_0 + \theta_1 x & \longrightarrow & J_{test}(\theta^{(1)}) \\ 2.\ h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 & \longrightarrow & J_{test}(\theta^{(2)}) \\ \vdots & & \\ 10.\ h_\theta(x) = \theta_0 + \ldots\ \theta_{10} x^{10} & \longrightarrow & J_{test}(\theta^{(10)}) \end{array}\right\} \text{select one}$$

θ But the selected one is also biased towards the 30% it was tested on, since it performed the best.

↳ Not indicative of performance on new data.

$\therefore$ Split 60/20/20 for train/val/test.

Training Set - trains algorithm

$$\rightarrow J_{train}(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Validation Set - evaluates model, used to select best one

$$\rightarrow J_{cv}(\theta) = \frac{1}{2m_{cv}} \sum_{i=1}^{m_{cv}} (h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$$

Test Set - determines final accuracy/error on new, unseen data

$$\rightarrow J_{test}(\theta) = \frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} (h_\theta(x_{test}^{(i)}) - y_{test}^{(i)})^2$$

Now:

1. $h_\theta(x) = \theta_0 + \theta_1 x \longrightarrow J_{cv}(\theta^{(1)})$
2. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 \longrightarrow J_{cv}(\theta^{(2)})$

$\vdots$

10. $h_\theta(x) = \theta_0 + \ldots \theta_{10} x^{10} \longrightarrow J_{cv}(\theta^{(10)})$

e.g. Pick 4.

Now estimate generalization error $J_{test}(\theta^{(4)})$.