Solution Space of Perceptron

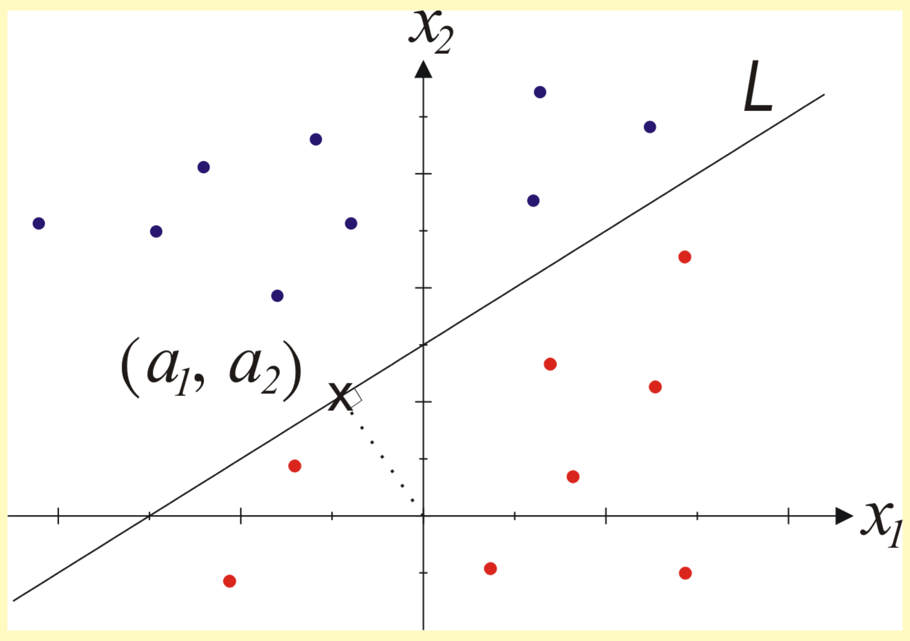
Jau-Chi Huang and Cheng-Yuan Liou*
Department of Computer Science and Information Engineering, National Taiwan University, Taiwan.
*cyliou@csie.ntu.edu.tw

Contribution

We presented a technique to visualize the global structure of the solution space for perceptron. To our knowledge, this solution space is the only one that can be used to explain various training behaviors of perceptron.

Method

The decision line in $2D$ input space can be represented by a perpendicular point.



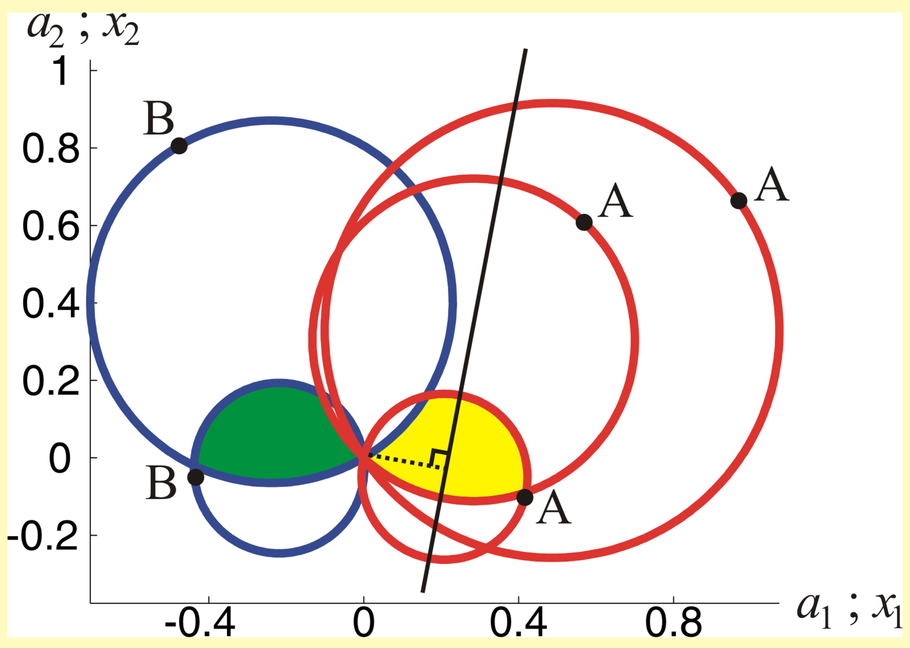
$$L : w_1x_1 + w_2x_2 + w_3 = 0$$

$$a_1 = \frac{-w_1w_3}{w_1^2 + w_2^2} \text{ and } a_2 = \frac{-w_2w_3}{w_1^2 + w_2^2}.$$

Each point $(a_1,\ a_2)$ represents two decision lines with different directions.

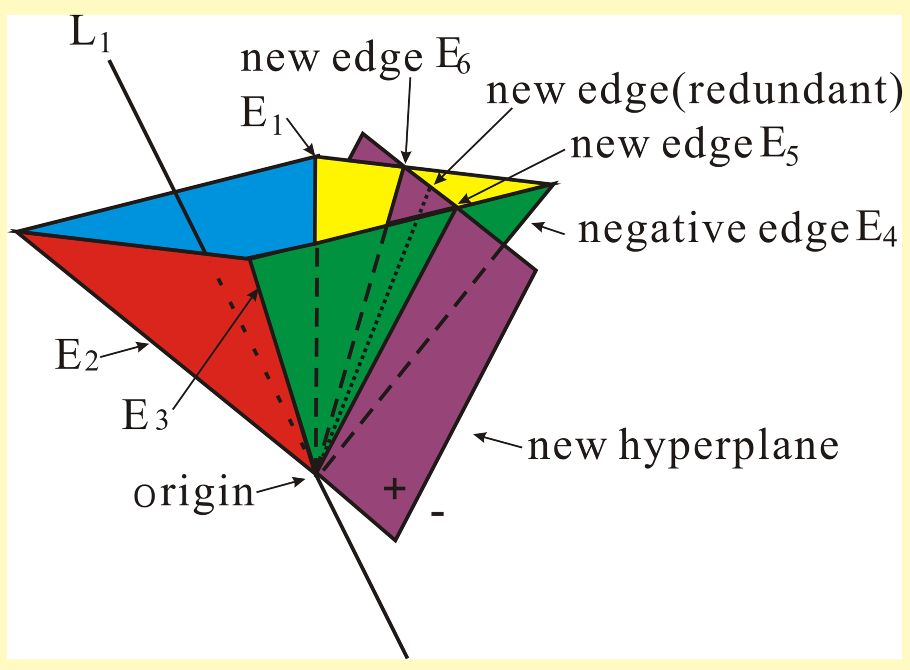
$$S\left[\frac{a_1}{a_1^2 + a_2^2}x_1 + \frac{a_2}{a_1^2 + a_2^2}x_2 - 1\right] = 0,\ S = \pm 1.$$

Solution Space

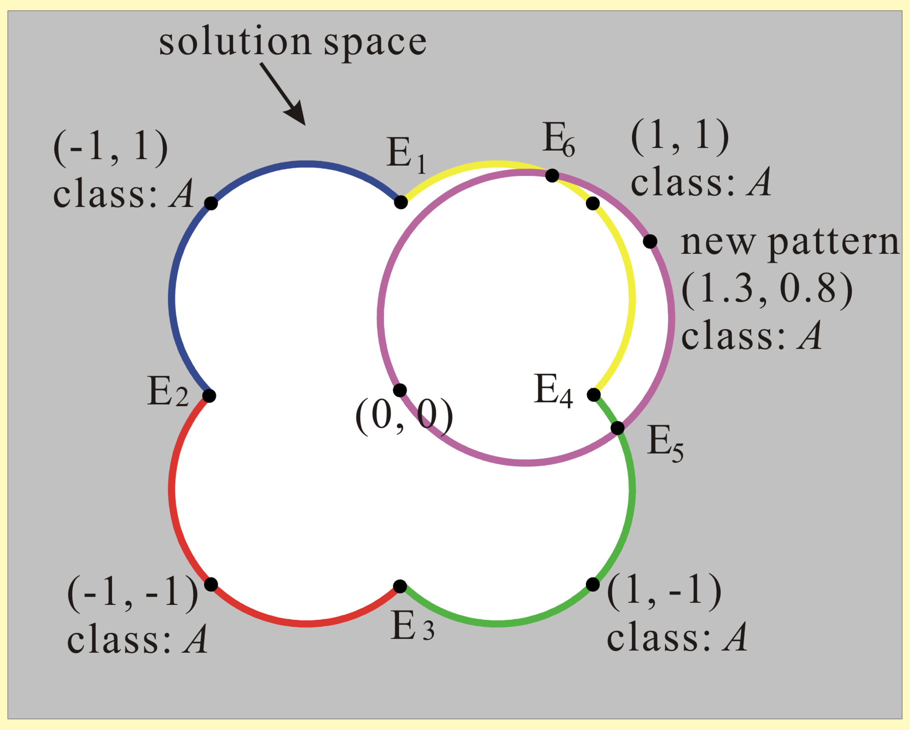


Correspondence

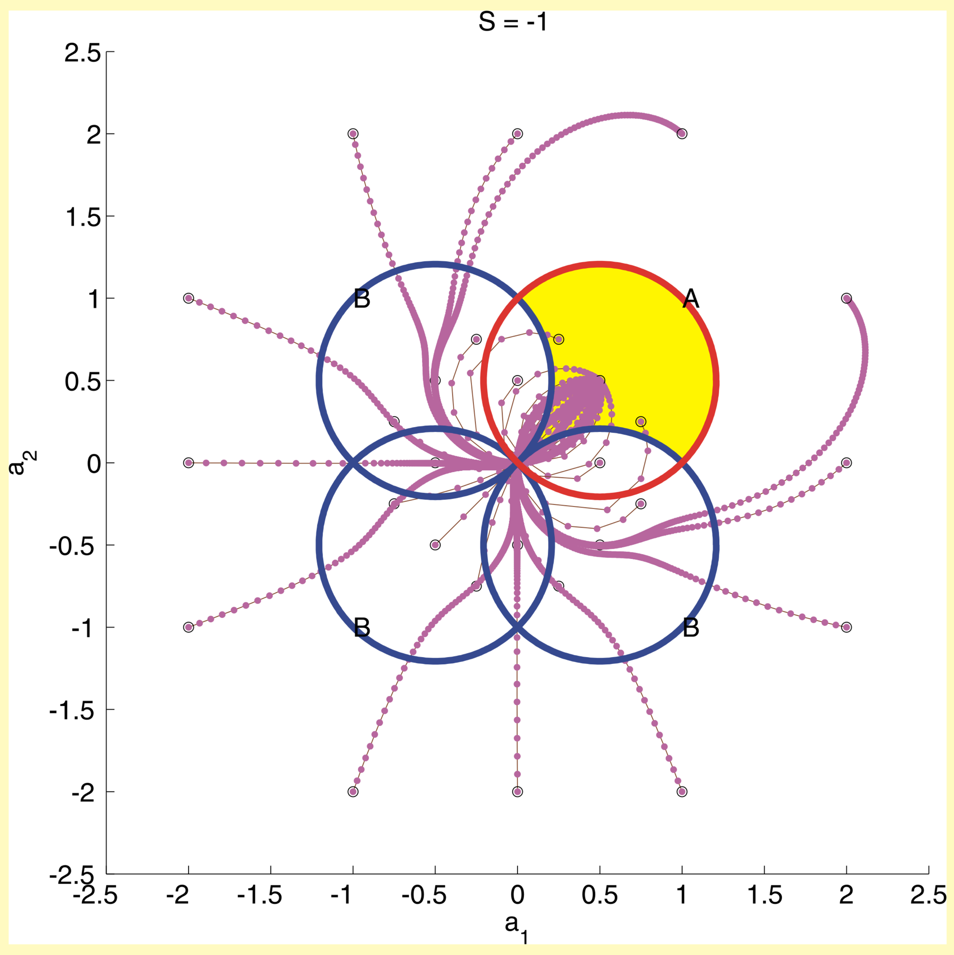
Weight Space:

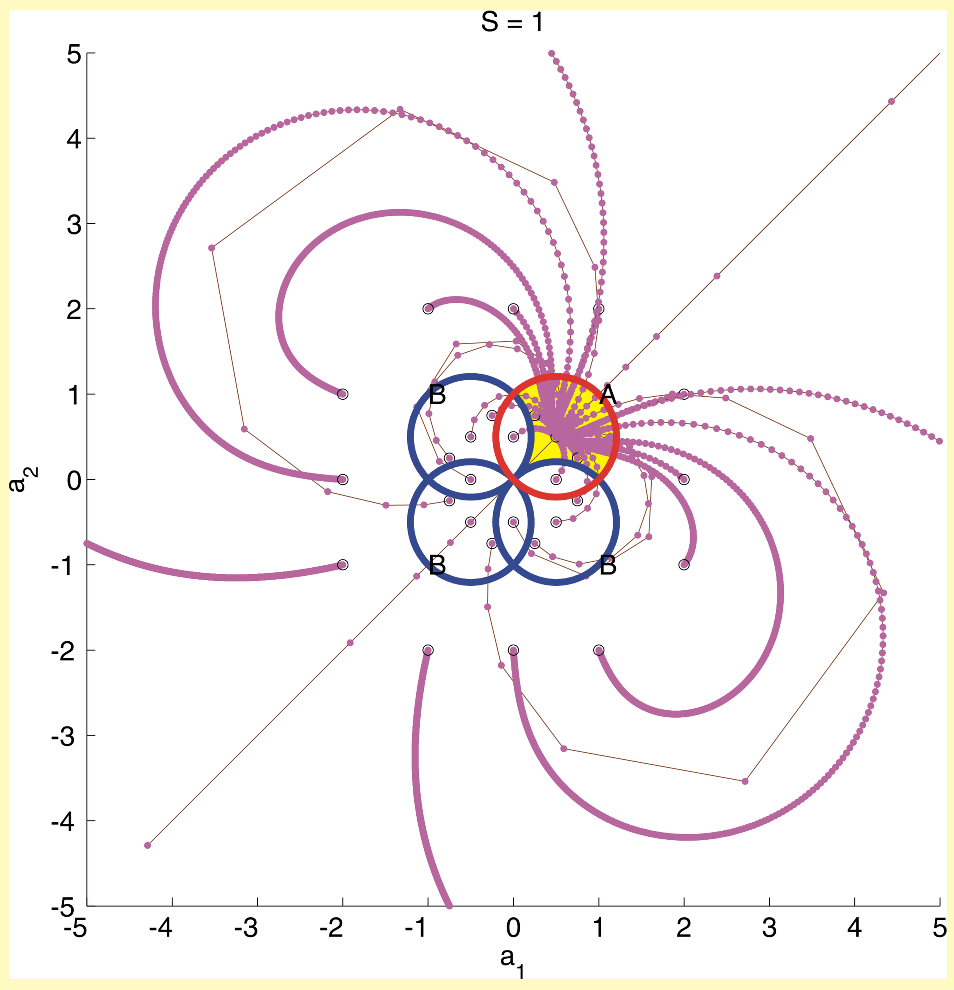


Input Space:

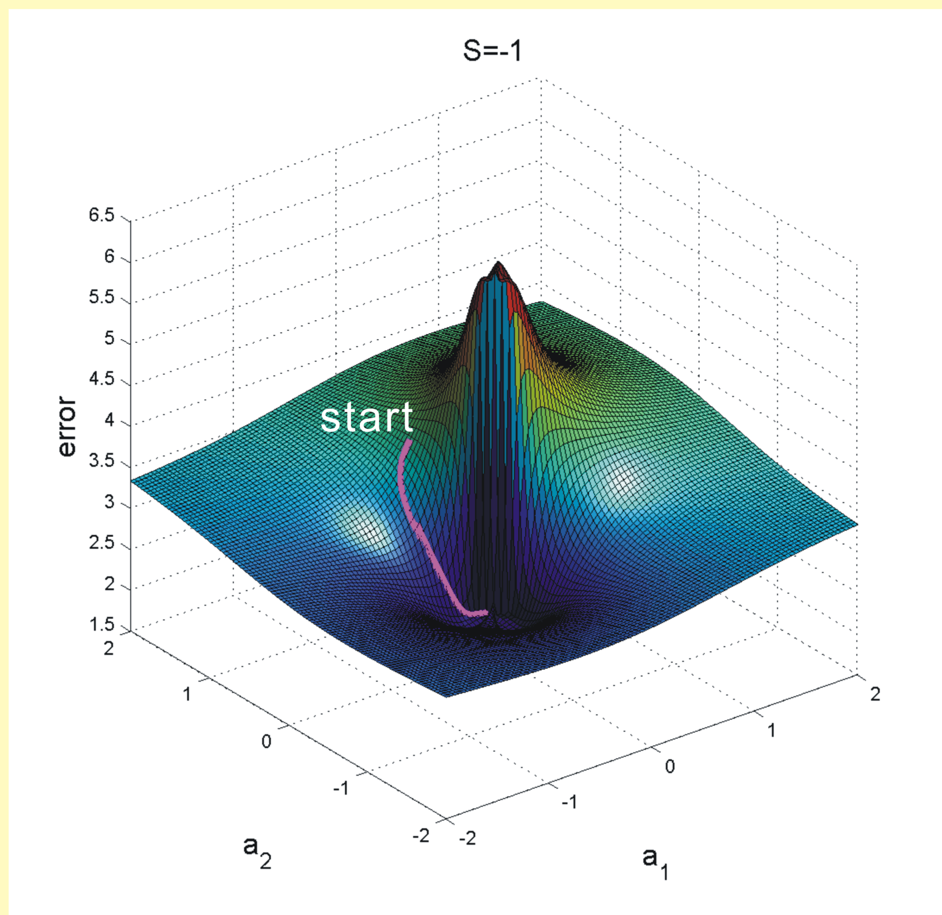


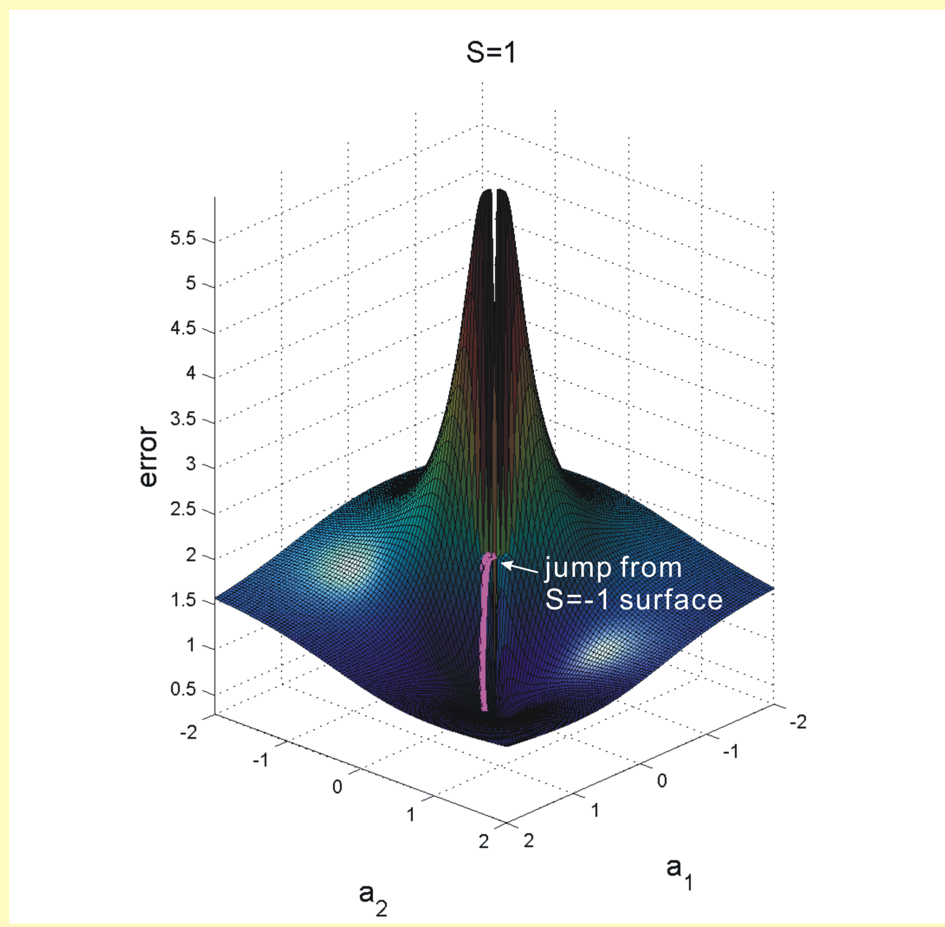
Training Path





Error Surfaces

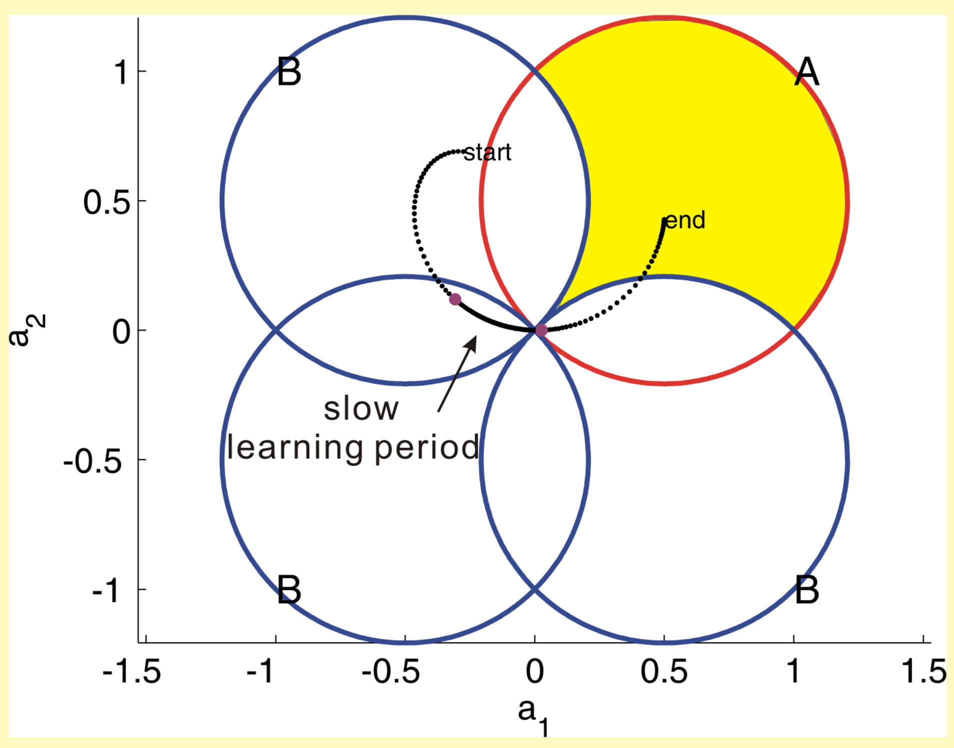


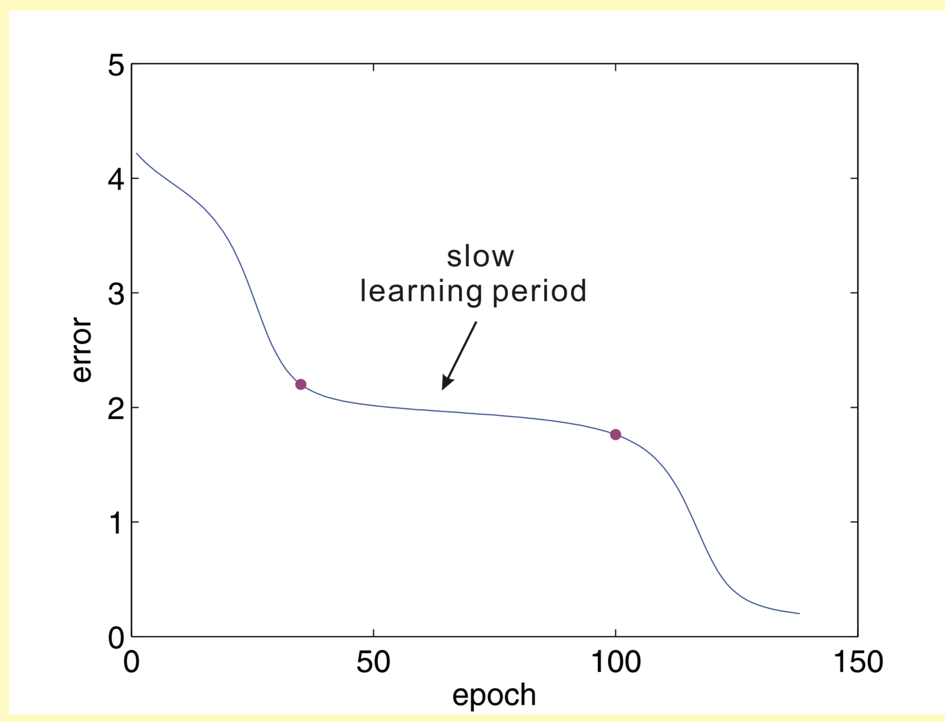


Learning Behavior

Above figures show that if the solution space only exists at S=1 space and the initial weights are at $S = -1$ space, the training path needs to pass the origin to change space. It means w_3 changes from positive value to negative value. Compare the learning behaviors of two different initial weights, $W_a = [1, -2.5, 2]$ and $W_b = 0.5W_a = [0.5, -1.25, 1]$, we'll find the bigger absolute value of w_3 causes slower learning period when changing from positive value to negative value. This can explain why the weights of the perceptron to be trained are typically initialized at small random values.

Initial Weights: $W_a = [1, -2.5, 2]$





Initial Weights: $W_b = [0.5, -1.25, 1]$