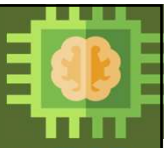
Elective Course

Course Code: CS4103

Autumn 2025-26



Lecture #43

Artificial Intelligence for Data Science

Week-12:

MACHINE LEARNING (Part XI)

A Few Factors to Improve Neural Network Learning

Introduction to Support Vector Machine (SVM)

Course Instructor:

Dr. Monidipa Das

Assistant Professor

Department of Computational and Data Sciences

Indian Institute of Science Education and Research Kolkata, India 741246

Improved Learning

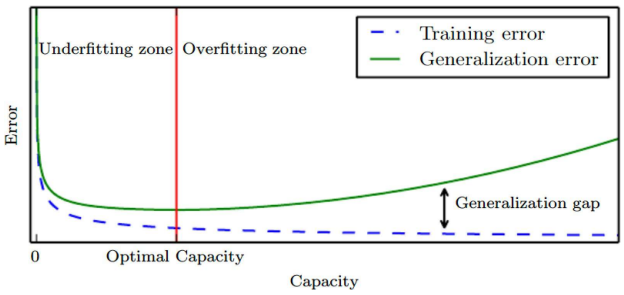
Regularization

- A method for automatically controlling the complexity of the learned hypothesis
- **Idea**: penalize for large values of θ_j
  - Can incorporate into the cost function
  - Works well when we have a lot of features, each that contributes a bit to predicting the label
- Can also address overfitting by eliminating features (either manually or via model selection)

$$J(\Theta) = -\frac{1}{n}\left[\sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik}\log(h_\Theta(\mathbf{x}_i))_k + (1-y_{ik})\log\Big(1-(h_\Theta(\mathbf{x}_i))_k\Big)\right] + \frac{\lambda}{2n}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}(\Theta_{ji}^{(l)})^2$$

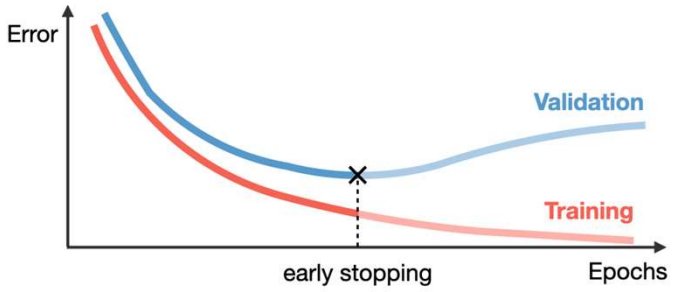
Handling Overfitting: Early Stopping

- Most commonly used in deep learning
- Popularity is due to its effectiveness and its simplicity
- Can think of early stopping as a very efficient hyperparameter selection algorithm
  - In this view number of training steps is just a hyperparameter
  - This hyperparameter has a U-shaped validation set performance curve
  - Most hyperparameters have such a U-shaped validation set performance curve



Early Stopping as Regularization

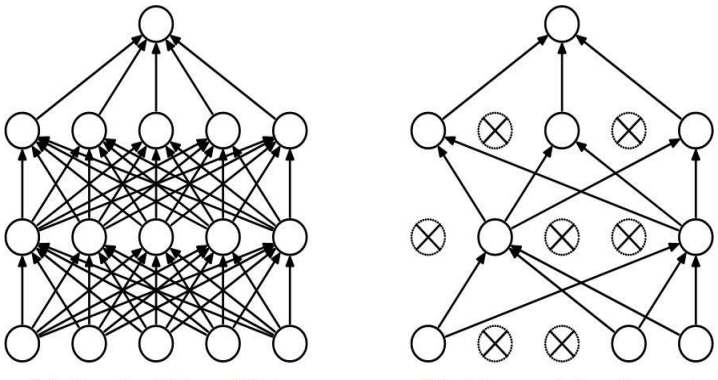
- Early stopping is an unobtrusive form of regularization
- It requires almost no change to the underlying training procedure, the objective function, or the set of allowable parameter values



- As such it is easy to use early stopping without damaging the learning dynamics

Dropout Neural Net

- A simple way to prevent neural net overfitting



(a) Standard Neural Net (b) After applying dropout.

Gradient Checking

- A method used during the implementation of the backward pass of a neural network. It compares the value of the analytical gradient to the numerical gradient at given points and plays the role of a sanity-check for correctness.

Numerical gradient	Analytical gradient
$\frac{df}{dx}(x) \approx \frac{f(x+h) - f(x-h)}{2h}$	$\frac{df}{dx}(x) = f'(x)$

Introduction to Support Vector Machine (SVM): A Few Technical Concepts

Hyperplane

9

- "In a n-dimensional space, a hyperplane is a flat affine subspace of dimension $n - 1$."
  - Hyperplane in 2D:

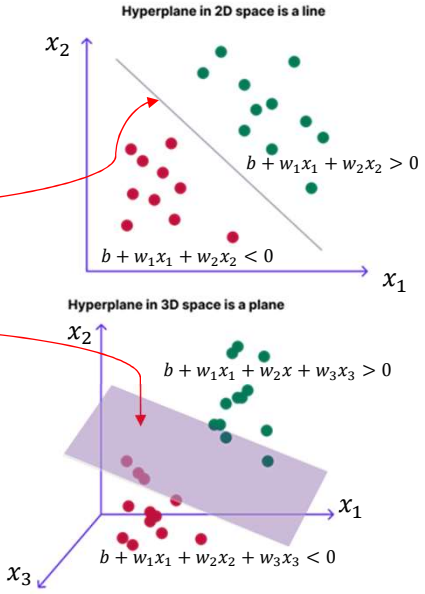
$$b + w_1x_1 + w_2x_2 = 0$$

  - Hyperplane in 3D:

$$b + w_1x_1 + w_2x_2 + w_3x_3 = 0$$

  - Hyperplane in nD: $b + w_1x_1 + w_2x_2 + \cdots + w_nx_n = 0$

$$OR \quad \boldsymbol{w}^T x + b = 0$$



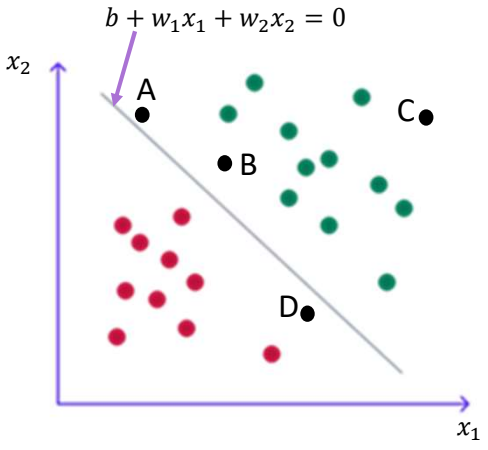
Hyperplane as Classifier

10

- A Hyperplane splits the original n – dimensional space into two half-spaces.
- A dataset is linearly separable if each half space has points only from a single class.
- We classify the test observation x ***based on the sign of*** $f(x^*)$

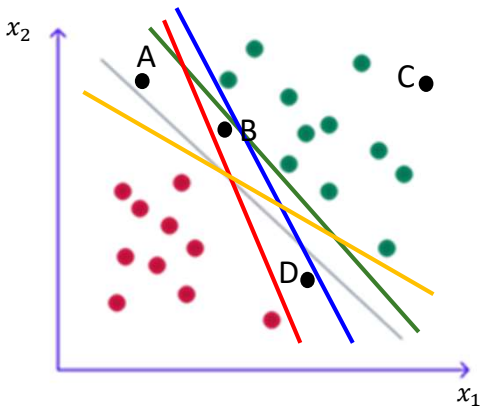
$$f(x^*) = b + w_1x_1^* + w_2x_2^* + \cdots + w_nx_n^*$$

x^* is a test sample.

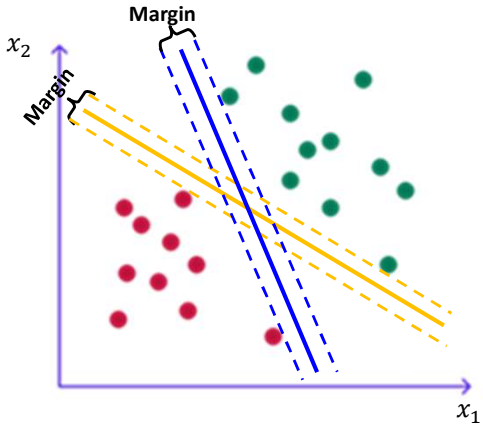


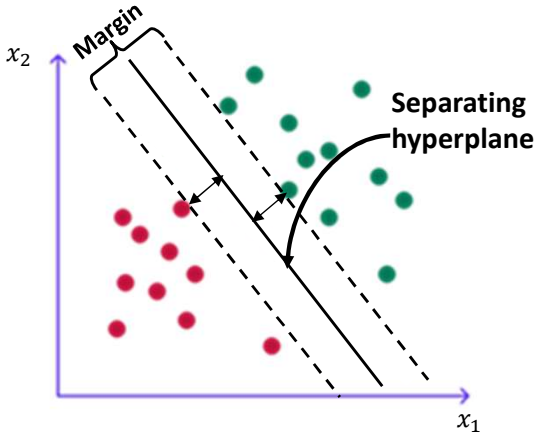
Hyperplane as Classifier

- There exist an infinite number of such hyperplanes.
- Which one of these separating hyperplanes to use?
  - optimal separating hyperplane
    Maximal ***margin*** hyperplane



Maximal Margin Hyperplane



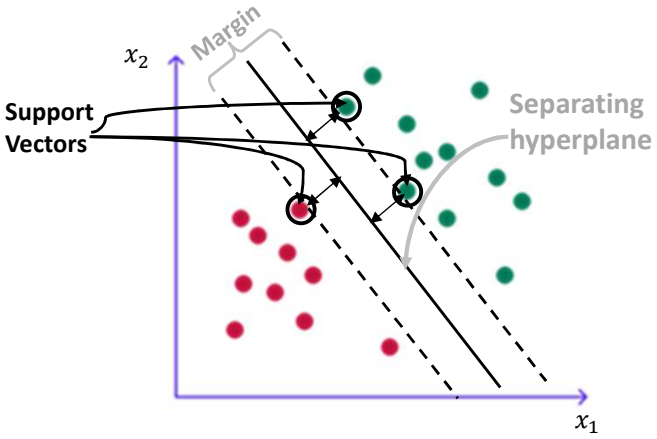


- Maximal Margin Hyperplane:
  - the separating hyperplane for which the ***margin is largest***
  - Margin: the distance between the hyperplane and the nearest data points from each class

Maximal Margin Classifier

13

- Maximal margin hyperplane as the classifier
- **Support Vectors:** "support" the maximal margin hyperplane
  - Slight movements of these would move the maximal margin hyperplane as well

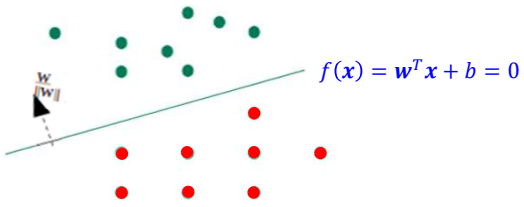


- The maximal margin hyperplane ***depends directly on the support vectors, but not on the other observations***

Maximum Margin: Formalization

14

- $\boldsymbol{w}$: decision hyperplane normal vector
- $\boldsymbol{x}_i$: data point i
- y_i: class of data point i (+1 or -1)

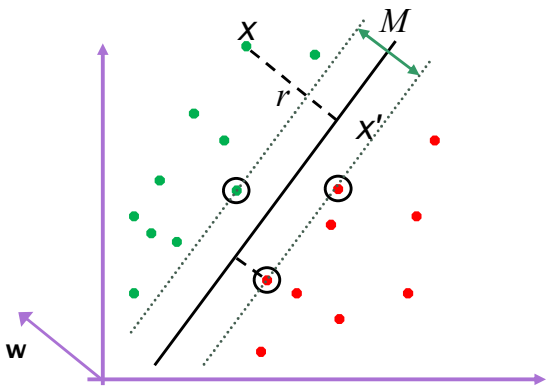


- Classifier is: $f(\boldsymbol{x}_i) = sign(\boldsymbol{w}^T\boldsymbol{x}_i + b)$
- **Functional margin** of $\boldsymbol{x}_i$ is: $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)$
  - But note that we can increase this margin simply by scaling $\boldsymbol{w}$, b….
- **Functional margin of dataset** is twice the minimum functional margin for any point
  - The factor of 2 comes from measuring the whole width of the margin

Geometric Margin

15

- **Distance from example to the separator**/hyperplane is $r = y\frac{\mathbf{w}^T\mathbf{x}+b}{\|\mathbf{w}\|}$
- Examples closest to the hyperplane are ***support vectors***.
- ***Margin*** (***M***) of the separator is the width of separation between support vectors of classes.



Derivation of finding r:

Dotted line $\boldsymbol{x}' - \boldsymbol{x}$ is perpendicular to decision boundary so parallel to $\boldsymbol{w}$.

Unit vector is $\frac{\boldsymbol{w}}{||\boldsymbol{w}||}$, so line is $r\frac{\boldsymbol{w}}{||\boldsymbol{w}||}$.

$\boldsymbol{x}' = \boldsymbol{x} - yr\frac{\boldsymbol{w}}{||\boldsymbol{w}||}$.

$\boldsymbol{x}'$ satisfies $\boldsymbol{w}^T\boldsymbol{x}' + b = 0$.

$$=> \quad \boldsymbol{w}^T(\boldsymbol{x} - yr\frac{\boldsymbol{w}}{||\boldsymbol{w}||}) + b = 0$$

$$=> \quad \boldsymbol{w}^T\boldsymbol{x} - yr||\boldsymbol{w}|| + b = 0$$

So, solving for r gives: $r = \frac{y(\boldsymbol{w}^T\boldsymbol{x}+b)}{||\boldsymbol{w}||}$

Note: y is either -1 or 1

Linear SVM Mathematically

The linearly separable case

16

- Assume that all data is at least distance 1 from the hyperplane,
- Then the following two constraints follow for a training set $\{(\boldsymbol{x_i}, y_i)\}$

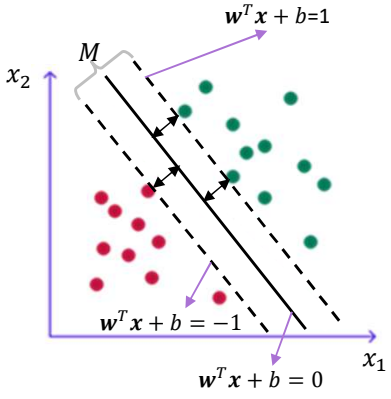
$$\mathbf{w}^T\mathbf{x_i} + b \geq 1 \quad \text{if } y_i = 1$$

$$\mathbf{w}^T\mathbf{x_i} + b \leq -1 \quad \text{if } y_i = -1$$

- For support vectors, the inequality becomes an equality
- Then, since each example's distance from the hyperplane is

$$r = y\frac{\mathbf{w}^T\mathbf{x}+b}{\|\mathbf{w}\|}$$

- The margin is: $M = \frac{2}{\|\mathbf{w}\|}$



Linear SVMs Mathematically (cont.)

- Then we can formulate the *quadratic optimization problem:*

Find $\mathbf{w}$ and b such that

$M = \frac{2}{\|\mathbf{w}\|}$ is maximized;

and for all $\{(\boldsymbol{x_i}, y_i)\}$: $\boldsymbol{w^T x_i} + b \geq 1$ if y_i=1; $\boldsymbol{w^T x_i} + b \leq -1$ if y_i = -1

- A better formulation (min $||\boldsymbol{w}||$ = max $\frac{1}{||w||}$):

Find $\mathbf{w}$ and b such that

$\boldsymbol{\Phi}(\mathbf{w}) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$ is minimized;

and for all $\{(\boldsymbol{x_i}, y_i)\}$: $y_i\,(\boldsymbol{w^T x_i} + b) \geq 1$

Solving the Optimization Problem

Find $\mathbf{w}$ and b such that
$\boldsymbol{\Phi}(\mathbf{w}) = \frac{1}{2}\boldsymbol{w^T w}$ is minimized;
and for all $\{(\mathbf{x_i}, y_i)\}$: $y_i\,(\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b) \geq 1$

Find $\mathbf{w}$ and b such that
$\boldsymbol{\Phi}(\mathbf{w}) = \frac{1}{2}||\boldsymbol{w}||^2$ is minimized;
and for all $\{(\mathbf{x_i}, y_i)\}$: $-y_i\,(\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b) + 1 \leq 0$

- This is now optimizing a *quadratic* function subject to *linear* constraints
- The solution involves ***constructing a dual problem*** where a ***Lagrange multiplier*** α_i is associated with every constraint in the primary problem.
  - **Construct the Lagrangian:**

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{\boldsymbol{i}} \alpha_i [y_i\,(\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b) - 1]$$

$$\nabla_w \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \boldsymbol{w} - \sum_{\boldsymbol{i}} \alpha_i y_i \mathbf{x_i} = \mathbf{0} \Rightarrow \boldsymbol{w} = \sum_{\boldsymbol{i}} \alpha_i y_i \mathbf{x_i} \qquad \nabla_b \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha}) = -\sum_{\boldsymbol{i}} \alpha_i y_i = \mathbf{0} \Rightarrow \sum_{\boldsymbol{i}} \alpha_i y_i = \mathbf{0}$$

Solving the Optimization Problem

– **Plugging back w and b values obtained and simplifying:**

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \sum_{i} \alpha_i \; - \frac{1}{2} \sum_{i,j} \alpha_i \; \alpha_j y_i \; y_j {x_i}^T x_j - b \sum_{i} \alpha_i y_i$$

$$\Rightarrow \; \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \sum_{i} \alpha_i \; - \frac{1}{2} \sum_{i,j} \alpha_i \; \alpha_j y_i \; y_j {x_i}^T x_j$$

- ***Dual optimization problem:***

Find $\alpha_1 \ldots \alpha_N$ such that
$Q(\boldsymbol{\alpha}) = \sum_{\boldsymbol{i}} \alpha_i \; - \frac{1}{2} \sum_{\boldsymbol{i,j}} y_i \; y_j \alpha_i \; \alpha_j {x_i}^T x_j$ *is maximized* and
s.t. $\alpha_i \geq 0$ for all α_i
$\sum_{\boldsymbol{i}} \alpha_i y_i = 0$

The Optimization Problem Solution

- The solution has the form:

$$\boldsymbol{w} = \Sigma \alpha_i y_i \boldsymbol{x_i} \qquad b = y_k - \mathbf{w}^T \mathbf{x_k} \qquad \text{for any } \mathbf{x_k} \text{ such that } \alpha_k \neq 0$$

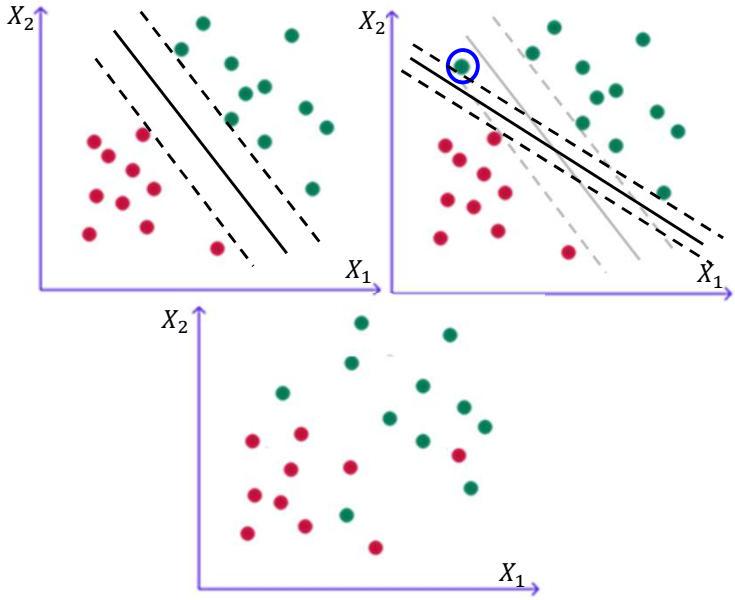
Each non-zero α_i indicates that corresponding $\mathbf{x_i}$ is a support vector.

- Then the classifying function will have the form:

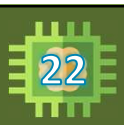
$$f(\boldsymbol{x}) = \Sigma \alpha_i y_i \boldsymbol{x_i}^T \boldsymbol{x} + b$$

- Observations:
  - It relies on an *inner product* between the test point $\boldsymbol{x}$ and the support vectors $\boldsymbol{x_i}$
  - Solving the optimization problem involved computing the inner products $\boldsymbol{x_i}^T \boldsymbol{x_j}$ between all pairs of training points.

Maximal Margin Classifier: Limitations



- Although the maximal margin classifier is often successful, it can also lead to overfitting when n is large.
- In many cases no separating hyperplane exists, hence, no maximal margin classifier
- Remedy: Idea of soft margin



Questions?