

Supervised Learning Cheatsheet

Introduction to Supervised Learning

Supervised learning is a type of machine learning paradigm where the model is trained on labeled data. The training process continues until the model achieves a desired level of accuracy. Mathematically, the goal is to find a function that, given a sample of data and corresponding outputs, best maps the input data to the output.

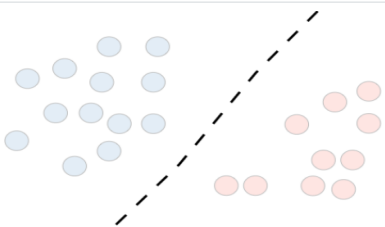
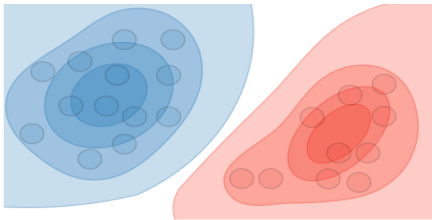
Math expression: ($f(x) = y$) where (x) is the input and (y) is the output.

Type of Prediction

Model Type	Description
Regression	Outcome is continuous. Examples include predicting house prices, temperature, sales amounts, etc.
Classification	Outcome is categorical. Examples include predicting whether an email is spam or not, if a tumor is benign or malignant, etc.

Type of Model

Model Type	Description
Discriminative Model	Directly estimates the output given an input. It learns the boundaries between classes.
Generative Model	It learns how the data is generated. It tries to model how the classes are distributed.

	Discriminative model	Generative model
Goal	Directly estimate $P(y x)$	Estimate $P(x y)$ to then deduce $P(y x)$
What's learned	Decision boundary	Probability distributions of the data
Illustration		
Examples	Regressions, SVMs	GDA, Naive Bayes

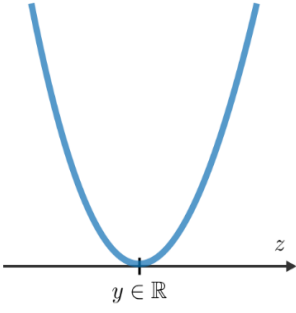
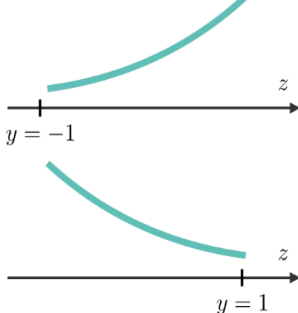
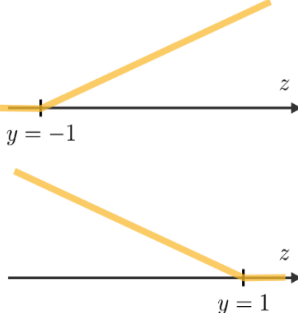
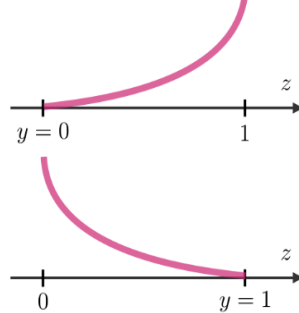
Notations and General Concepts

Hypothesis

The hypothesis is noted (h_θ) and represents the model we choose. Given input data $x^{(i)}$, the model's predicted output is $h_\theta(x^{(i)})$.

Loss Function

A loss function is a function defined as $L: (z, y) \in \mathbb{R} \times Y \rightarrow L(z, y) \in \mathbb{R}$. It measures the discrepancy between the predicted value z and the actual data value y . Common loss functions are detailed in the table below:

Least squared error	Logistic loss	Hinge loss	Cross-entropy
$\frac{1}{2}(y - z)^2$	$\log(1 + \exp(-yz))$	$\max(0, 1 - yz)$	$-\left[y \log(z) + (1 - y) \log(1 - z)\right]$
			
Linear regression	Logistic regression	SVM	Neural Network

Cost Function

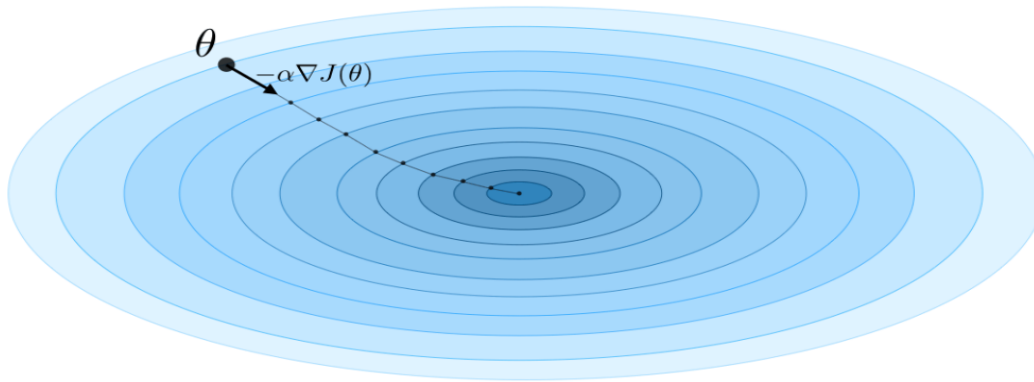
The cost function, denoted as J , is commonly used to evaluate a model's performance. It is defined using the loss function L as:

$$J(\theta) = \sum_{i=1}^m L(h_\theta(x^{(i)}), y^{(i)})$$

Gradient Descent

Given the learning rate $\alpha \in \mathbb{R}$, the update rule for gradient descent is expressed using both the learning rate and the cost function (J).

$$\theta \leftarrow \theta - \alpha \nabla J(\theta)$$



Likelihood

The likelihood of a model, $L(\theta)$, given parameters θ , is employed to determine the optimal parameters θ via likelihood maximization. The relationship is:

$$\theta^{\text{opt}} = \arg \max_{\theta} L(\theta)$$

Newton's Algorithm

Newton's algorithm is a numerical approach to find θ such that $\ell'(\theta) = 0$. The algorithm's update rule is:

$$\theta \leftarrow \theta - \frac{\ell'(\theta)}{\ell''(\theta)}$$

Linear models

Linear regression

Given:

$$y|x; \theta \sim N(\mu, \sigma^2)$$

Normal Equations:

The solution for θ that minimizes the cost function in linear regression can be found using the normal equation:

$$\theta = (X^T X)^{-1} X^T y$$

Where:

- X is the design matrix.
- y is the vector of observed outputs.

This equation provides the optimal values for θ without the need for iterative optimization, given the assumptions of linear regression are met.

Locally Weighted Regression (LWR):

For each training example, it assigns a weight:

$$w^{(i)}(x) = \exp \left(-\frac{(x^{(i)} - x)^2}{2\tau^2} \right)$$

Where:

x is the input value for which we're predicting the output.

$x^{(i)}$ is the input value from the i^{th} training example.

τ is a bandwidth parameter that determines the range of influence of the training examples.

The closer x is to $x^{(i)}$, the larger the weight $w^{(i)}(x)$ will be. The parameter τ controls how quickly the weights drop off for training examples that are farther from x .