Notation Reference Sheet

Hypotheses

h: hypothesis

$H = \{h_1, h_2, h_3, \dots\}$: hypothesis space

Data

$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_M \end{pmatrix}$: data point corresponding to a column vector of M features

$\overline{\boldsymbol{x}} = \begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix} = \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \dots \\ x_M \end{pmatrix}$: concatenation of 1 with the vector $\boldsymbol{x}$

$\boldsymbol{X} = \begin{pmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & \ddots & \vdots \\ x_{M1} & \cdots & x_{MN} \end{pmatrix}$: dataset consisting of N data points of M features

$\overline{\boldsymbol{X}} = \begin{pmatrix} 1 & \dots & 1 \\ x_{11} & \cdots & x_{1N} \\ \vdots & \ddots & \vdots \\ x_{M1} & \cdots & x_{MN} \end{pmatrix}$: concatenation of a vector of 1's with the matrix $\boldsymbol{X}$

y: output target (regression) or label (classification)

$\boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{pmatrix}$: vector of outputs for a dataset of N points

N: # of data points in a dataset

n: index of a data point in a dataset

M: # of features in a data point

m: index of a feature in a data point

Weights

$\boldsymbol{w} = \begin{pmatrix} w_1 \\ w_2 \\ \dots \\ w_M \end{pmatrix}$: vector of weights

$\boldsymbol{w}^T = (w_1, w_2, \dots, w_M)$ or $(w_0, w_1, w_2, \dots, w_M)$ depending on the context (here w_0 is an additional weight that multiplies the first entry of $\overline{\boldsymbol{x}}$ when computing $\boldsymbol{w}^T \overline{\boldsymbol{x}}$)

Mixture of Gaussians

π: mixture probability of a Gaussian

$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_M \end{pmatrix}$: mean of a Gaussian

σ: Standard deviation of a univariate Gaussian

$\boldsymbol{\Sigma} = \begin{pmatrix} & \cdots & \\ \vdots & \ddots & \vdots \\ & \cdots & \end{pmatrix}$: covariance matrix of a multivariate Gaussian

Regularization

λ: weight determining the importance of the penalty term

Neural networks

$a_j = \sum_i w_{ji} z_i$: linear combination of inputs fed to unit j

$h(a_j)$: activation function (identity, sigmoid, Gaussian, tanh, etc.) applied to unit j

$\boldsymbol{W}^{(k)}$: matrix of weights connecting layer k to layer $k+1$

η: step length (learning rate) in gradient descent

E_n: error at output node n

$\delta_j = \frac{\partial E_n}{\partial a_j}$: partial derivative of the error at output node n with respect to linear combo a_j at unit j