UNIVERSITÀ PARTHENOPE
DiST
DIPARTIMENTO DI SCIENZE E TECNOLOGIE

Machine Learning (part II)

Error Functions

Angelo Ciaramella

Introduction

Goal in network training

- Model the underlying generator of the data
- General and complete description of the generator joint probability input-target space

$$p(\mathbf{x}, \mathrm{t}) = p(\mathbf{t}|\mathbf{x})p(\mathbf{x})$$

$$p(\mathbf{x}) = \int p(\mathbf{t}, \mathbf{x})d\mathbf{t}$$

Likelihood

- Training data

$$\{\mathbf{x}^n, \mathbf{t}^{\mathbf{n}}\}$$

- Likelihood

$$\mathcal{L} = \prod_n p(\mathbf{x}^n, \mathbf{t}^n) = \prod_n p(\mathbf{t}^n|\mathbf{x}^n)p(\mathbf{x}^n)$$

- Maximizing the likelihood

$$E = -\ln \mathcal{L} = -\sum_n \ln p(\mathbf{t}^n|\mathbf{x}^n) - \sum_n \ln p(\mathbf{x}^n)$$

The term does not depend on the network parameters

sum-of-squares error

- Consider c targets t_k

$$p(\mathbf{t}|\mathbf{x}) = \prod_{k=1}^{c} p(t_k|\mathbf{x})$$

- Assume distribution of target data is Gaussian

deterministic function

$$t_k = h_k(\mathbf{x}) + \epsilon_k$$

Gaussian noise

- Distribution of the error

$$p(\epsilon_k) = \frac{1}{(2\pi\sigma^2)^{1/2}} e^{-\frac{e_k^2}{2\sigma^2}}$$

sum-of-squares error

- Considering

$$y_k(\mathbf{x},\mathbf{w}) \equiv \; h_k(\mathbf{x})$$

- Probability distribution of target

$$p(t_k|\mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{1/2}} e^{-\frac{\{y_k(\mathbf{x},\mathbf{w})-t_k\}^2}{2\sigma^2}}$$

- The overall error function

Terms independent from the weights

$$E = \frac{1}{2\sigma^2}\sum_{n=1}^{N}\sum_{k=1}^{c}\{y_k(\mathbf{x}^n,\mathbf{w}) - t_k^n\}^2 + Nc\,\mathbf{ln}\sigma + \frac{Nc}{2}\ln(2\pi)$$

sum-of-squares error

- Sum of squares error function

$$
\begin{aligned}
E &= \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{c}\{y_k(\mathbf{x}^n,\mathbf{w}) - t_k^n\}^2 \\
&= \frac{1}{2}\sum_{n=1}^{N}\|y(\mathbf{x}^n,\mathbf{w}) - \mathbf{t}^n\|^2
\end{aligned}
$$

Interpretation of NN output

- Limit of number N to infinity

$$E = \lim_{N\to\infty} \frac{1}{2N} \sum_{n=1}^{N} \sum_{k} \{y_k(\mathbf{x}^n, \mathbf{w}) - t_k^n\}^2 =$$

$$\frac{1}{2} \sum_{k} \int \int \{y_k(\mathbf{x}^n, \mathbf{w}) - t_k^n\}^2 p(t_k, \mathbf{x}) d t_k d\mathbf{x}$$

- Moreover

$$E =$$

$$\frac{1}{2} \sum_{k} \int \int \{y_k(\mathbf{x}^n, \mathbf{w}) - t_k^n\}^2 p(t_k, | \mathbf{x}) p(\mathbf{x}) d t_k d\mathbf{x}$$

Interpretation of NN output

- Defining

$$\langle t_k | \, \mathbf{x} \rangle \equiv \int t_k p(t_k|\mathbf{x}) dt_k$$

$$\langle t_k^2 | \, \mathbf{x} \rangle \equiv \int t_k^2 p(t_k|\mathbf{x}) dt_k$$

- Moreover

$$\{y_k - t_k\}^2 = \{y_k - \langle t_k | \, \mathbf{x} \rangle + \langle t_k | \, \mathbf{x} \rangle \; - t_k\}^2$$

$$= \{y_k - \langle t_k | \, \mathbf{x} \rangle\}^2 + 2\{y_k - \langle t_k | \, \mathbf{x} \rangle\}\{\langle t_k | \, \mathbf{x} \rangle - t_k\} + \{\langle t_k | \, \mathbf{x} \rangle - t_k\}^2$$

Interpretation of NN output

- The error

$$E =$$

$$\frac{1}{2}\sum_{k}\int\{y_k(\mathbf{x}^n,\mathbf{w}) - \langle t_k|\,\mathbf{x}\rangle\}^2 p(\mathbf{x})d\mathbf{x} +$$

$$\frac{1}{2}\sum_{k}\int\{\langle t_k^2|\,\mathbf{x}\rangle \;-\; \langle t_k|\,\mathbf{x}\rangle^2\}\;\; p(\mathbf{x})d\mathbf{x} +$$

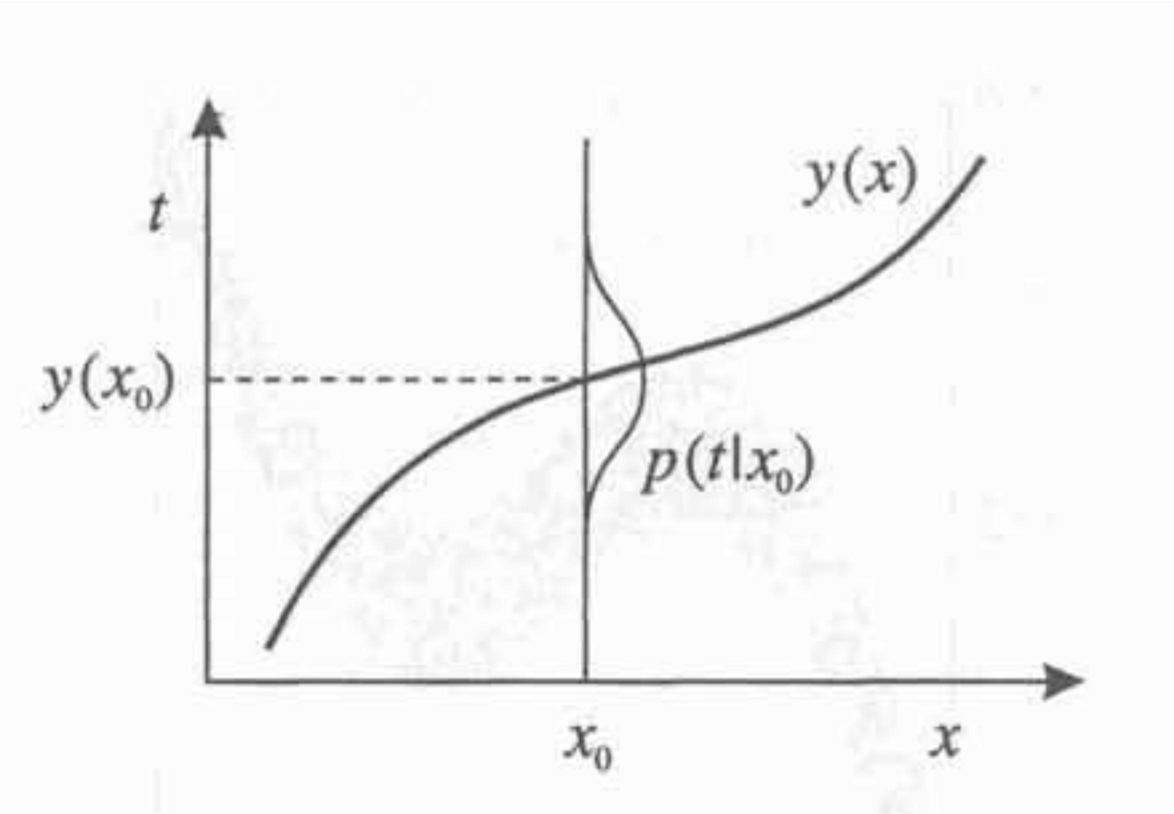
- First integrand positive

$$y_k(\mathbf{x}^n,\mathbf{w}^*) = \langle t_k|\,\mathbf{x}\rangle$$

Regression ot t_k conditioned on **x**

Interpretation of NN output

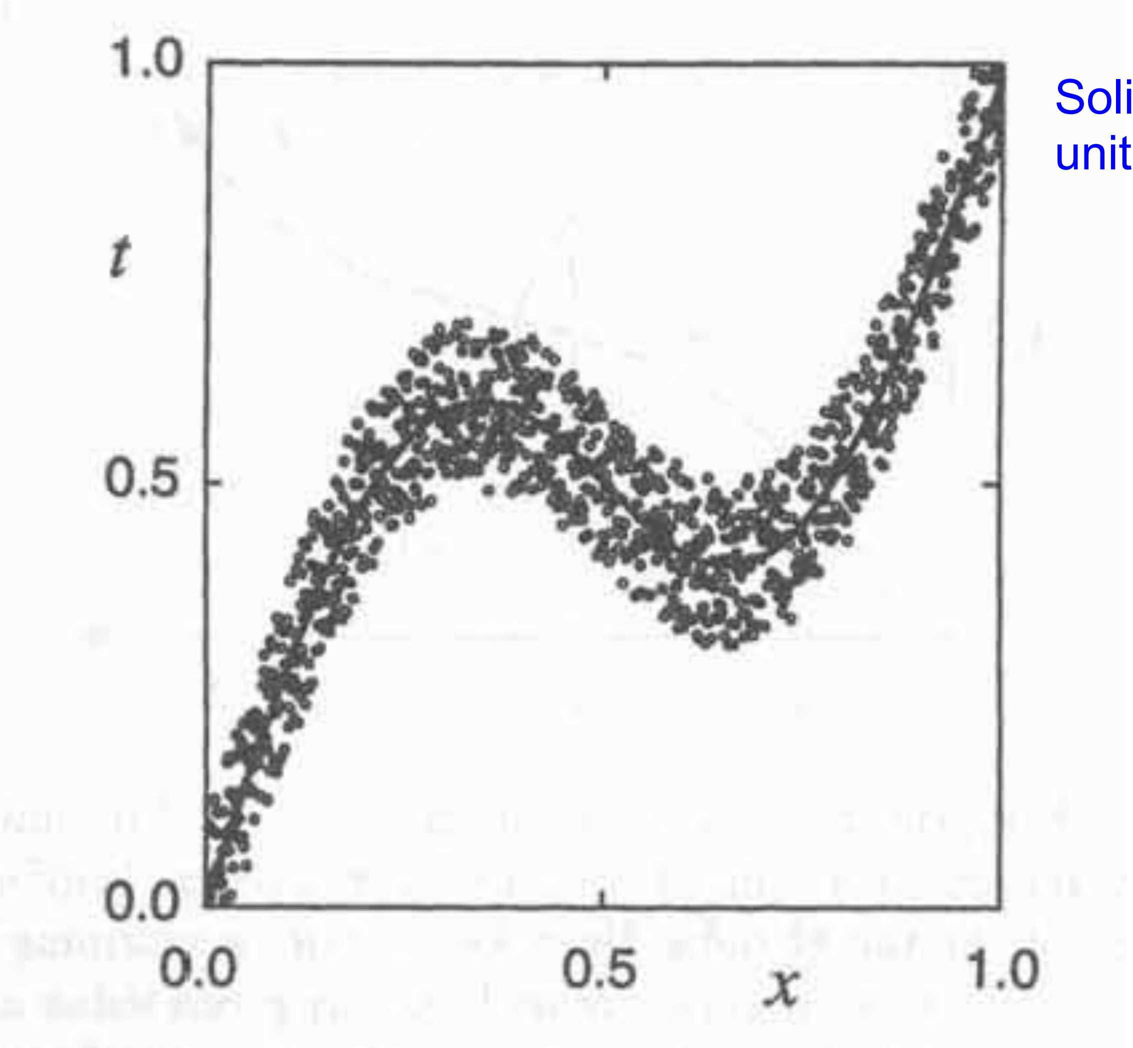
- Regression



Considerations

- Three key assumptions
  - Data set must be sufficiently large that is approximates an infinite data set
  - The network output function must be sufficiently general that there exists a choice of parameters (hidden units sufficiently large) which makes th error sufficiently small
  - The optimization of the network parameters is performed in such a way as to find the appropriate minimum of the cost function

Example



Solid curve: MLP with 5 hidden units and sum-of-squares error

$$t = x + 0.3\sin(2\,\pi\,x) + \epsilon$$

General error functions

- Generalization of the Gaussian distribution

$$p(\epsilon) = \frac{R\beta^{1/R}}{2\Gamma(1/R)} e^{-\beta|\epsilon|^R}$$

Gamma function

- By negative log-likelihood the error function is

$$E = \sum_{n=1}^{N} \sum_{k=1}^{c} |y_k(\mathbf{x}^n, \mathbf{w}) - t_k^n|^R$$

Minkowski-R error

General error functions

- The derivatives

$$\frac{\partial E}{\partial w_{ji}} = \sum_{n=1}^{N} \sum_{k=1}^{c} |y_k(\mathbf{x}^n, \mathbf{w}) - t_k^n|^{R-1} sign(y(\mathbf{x}^n, \mathbf{w}) - \mathbf{t}^n) \frac{\partial y_k^n}{\partial w_{ji}}$$

- Evaluated using the standard back-propagation procedure
  - sensitivity to outliers

Classification

- Density function

1-of-c coding

$$p(t_k|\mathbf{x}) = \sum_{l=1}^{c} \delta(t_k - \delta_{kl}) P(C_l|\mathbf{x})$$

- by sum-of-squares error function the conditional average

$$y_k(\mathbf{x}) = \langle t_k|\, \mathbf{x}\, \rangle = \int t_k p(t_k|\mathbf{x}) d_{t_k} = P(C_k|\, \mathbf{x})$$

Hidden units

- Considering the error

$$E = \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{c} \left\{ \sum_{j=1}^{M} w_{kj}\, z_j^n - t_k^n \right\}^2$$

- Defining

$$(\mathbf{T})_{nk} = t_k^n \qquad (\mathbf{W})_{kj} = w_{kj} \qquad (\mathbf{Z})_{nj} = z_j^n$$

- Mimimizing the error with respect the weights

$$\frac{\partial E}{\partial w_{kj}} = \sum_{n=1}^{N} \left\{ \sum_{j=1}^{M} w_{kj} z_j^n - t_k^n \right\} z_j^n = 0$$

Hidden units

- In matrix notation

$$\mathbf{Z}^T\mathbf{Z}\,\mathbf{W}^T - \mathbf{Z}^T\mathbf{T} \quad = 0$$

- with solution

$$\mathbf{W}^T = \mathbf{Z}^{\dagger}\mathbf{T} \quad = 0$$

- The error becomes

$$E = \frac{1}{2} Tr\{(\mathbf{Z}\mathbf{W}^T - \mathbf{T})(\mathbf{Z}\mathbf{W}^T - \mathbf{T})^T\}$$

$$E = \frac{1}{2} Tr\left\{\left(\mathbf{Z}\mathbf{Z}^{\dagger}\mathbf{T} - \mathbf{T}\right)\left(\mathbf{Z}\mathbf{Z}^{\dagger}\mathbf{T} - \mathbf{T}\right)^T\right\}$$

Hidden units

- After simple matrix manipultation

$$E = \frac{1}{2} Tr\{\mathbf{T}^T\mathbf{T} - \mathbf{S}_B\mathbf{S}_T^{-1}\}$$

- with

covariance matrix of the last hidden layer

$$\mathbf{S}_T = \mathbf{Z}^T\mathbf{Z} \quad = \sum_n (\mathbf{z}^n - \bar{\mathbf{z}})(\mathbf{z}^n - \bar{\mathbf{z}})^T$$

between class covariance matrix

$$\mathbf{S}_B = \mathbf{Z}^T\mathbf{T}\mathbf{T}^T\mathbf{Z}$$

Hidden units

- Minimizing the sum-of-squares error is equivalent to maximizing the discriminant

$$E = \frac{1}{2} Tr\{\mathbf{S}_B \mathbf{S}_T^{-1}\}$$

Similar to Fisher discriminant

- Result
  - The weights in the final layer are adjusted to produce an optimum discrimination of the classes of input vectors by means of a linear transformation

Hidden units

- Minimizing the sum-of-squares error is equivalent to maximizing the discriminant

$$E = \frac{1}{2} Tr\{\mathbf{S}_B \mathbf{S}_T^{-1}\}$$

Similar to Fisher discriminant

- Result
  - The weights in the final layer are adjusted to produce an optimum discrimination of the classes of input vectors by means of a linear transformation

Cross-Entropy

- Network with single output *y*

$$P(C_1|\mathbf{x}) = y \qquad\qquad P(C_2|\mathbf{x}) = 1 - y$$

- Probability

$$p(t|\mathbf{x}) = y^t(1-y)^{1-t}$$

Bernoulli distribution

- Likelihood

$$\mathcal{L} = \prod_n (y^n)^{t^n}(1-y^n)^{1-t^n}$$

ML – Error functions

Cross-Entropy

- Cross-entropy error function

$$E = -\ln \mathcal{L}$$
$$= -\sum_{n} \{t^{n} \ln y^{n} + (1 - t^{n}) \ln(1 - y^{n})\}$$

- Differentiating

$$\frac{\partial E}{\partial y^{n}} = \frac{(y^{n} - t^{n})}{y^{n}(1 - y^{n})}$$

- Absolute minimum

$$y^{n} = t^{n} \qquad \forall\, n$$

ML – Error functions

Cross-Entropy for multiple class

- Conditional distribution

$$p(t^n|\mathbf{x}^n) = \prod_{k=1}^{c} (y_k^n)^{t_k^n}$$

- Differentiating

$$E = -\sum_{n} \sum_{k=1}^{c} t_k^n \ln y_k^n$$