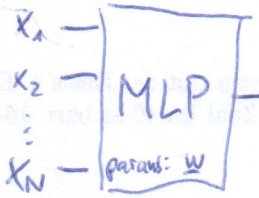
Multilayer Perceptron (MLP): Cost functions

(*) $\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 = \frac{1}{2}$ vs. $\left(\frac{1}{4}\right)^2 + \left(\frac{3}{4}\right)^2 = \frac{10}{16}$ for distance 1 between labels

$1 = \frac{1}{2} + \frac{1}{2}$ ∧ $\frac{1}{4} + \frac{7}{4}$

a) What will be the effect of the error measure choice?



$$\underline{w} \in \operatorname{argmin}_{\underline{w}} E^T(\underline{w}) = \operatorname{argmin}_{\underline{w}} \frac{1}{P} \sum_{\alpha=1}^{P} e^{(\alpha)}(\underline{w})$$

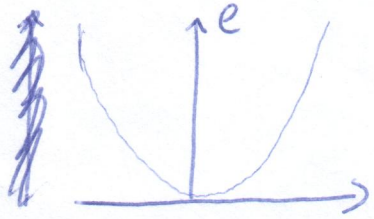
E^T: batch training error; $e^{(\alpha)}$: training error of data point α

$(\underline{x}^{(\alpha)}, y_T^{(\alpha)})$, $\alpha = 1, \dots, P$ with $y_T^{(\alpha)} \in \mathbb{R}$, $\underline{x}^{(\alpha)} \in \mathbb{R}^N$

COST FUNCTIONS FOR REGRESSION

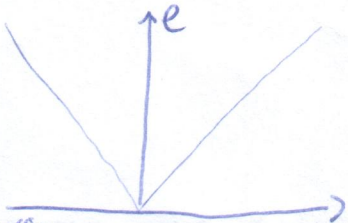
1) quadratic error function:

$$e^{(\alpha)} = \frac{1}{2}\left(y(\underline{x}^{(\alpha)}; \underline{w}) - y_T^{(\alpha)}\right)^2$$

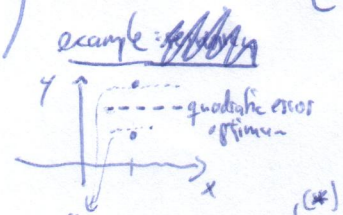


2) linear error

$$e^{(\alpha)} = \left| y(\underline{x}^{(\alpha)}; \underline{w}) - y_T^{(\alpha)} \right|$$



example:



⇒ linear error less robust w.r.t. outliers

b) Relation between quadratic error and Gaussian labels → Maximum likelihood.

remark:

if labels are conditionally Gaussian,

i.e., $P_{data}(y|\underline{x}) = \mathcal{N}(y | y(\underline{x}; \underline{w}), \sigma^2)$,

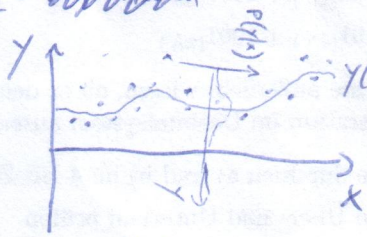
then: $\underline{w}_{opt} \in \operatorname{argmin}_{\underline{w}} E^T(\underline{w})$

likelihood fct. ($\mathcal{L}(\underline{w})$)

$$\Leftrightarrow \underline{w}_{opt} \in \operatorname{argmax}_{\underline{w}} \prod_{\alpha=1}^{P} \mathcal{N}(y^{(\alpha)} | y(\underline{x}^{(\alpha)}; \underline{w}), \sigma^2)$$

⇒ min. squared error corresponds to max. likelihood solution for Gaussian labels (Hint: log-likelihood...)

example:



outliers:

This property makes the quadratic cost function the standard choice for regression!

$$\underline{w}_{opt} \in \operatorname{argmin}_{\underline{w}} \left(-\ln\left[\mathcal{L}(\underline{w})\right]\right)$$

MLP cost functions (cont'd)

c) Derive the cross entropy cost function for binary classification

Data: $\{(\underline{x}^{(\alpha)}, y_T^{(\alpha)})\}_{\alpha=1}^{P}$ with $\underline{x}^{(\alpha)} \in \mathbb{R}^N$, $y_T^{(\alpha)} \in \{0,1\}$

generated by $(\underline{x}^{(\alpha)}, y_T^{(\alpha)}) \overset{iid}{\sim} p_{data}(\underline{x}, y)$

Model: MLP, output interpreted as probability that $y=1$:

$$y(\underline{x};\underline{w}) =: p_{model}(y=1|\underline{x}) \Rightarrow p_{model}(y=0|\underline{x}) = 1 - y(\underline{x};\underline{w})$$

Cost function derivation via minimization of Kullback-Leibler divergence between data and model distribution. ($D_{KL} = 0$ for identical and $D_{KL} > 0$ for different distributions, not symmetric!)

→ see lecture.

$$D_{KL}(p_{data}(\underline{x},y) \,||\, p_{model}(\underline{x},y)) = \int_{\mathbb{R}^N} d\underline{x} \sum_{y\in\{0,1\}} p_{data}(\underline{x},y) \ln\left(\frac{p_{data}(\underline{x},y)}{p_{model}(\underline{x},y)}\right) = \int_{\mathbb{R}^N} d\underline{x} \sum_{y\in\{0,1\}} p_{data}(\underline{x}) p_{data}(y|\underline{x}) \ln \frac{p_{data}(\underline{x}) p_{data}(y|\underline{x})}{p_{model}(\underline{x}) p_{model}(y|\underline{x})}$$

(p_{data}: discriminative not generative model)

$$= \underbrace{\int_{\mathbb{R}^N} d\underline{x}\, p_{data}(\underline{x}) \sum_{y\in\{0,1\}} p_{data}(y|\underline{x}) \ln\left[p_{data}(y|\underline{x})\right]}_{\text{independent of model parameters } \underline{w}} - \underbrace{\int_{\mathbb{R}^N} d\underline{x}\, p_{data}(\underline{x}) \sum_{y\in\{0,1\}} p_{data}(y|\underline{x}) \ln\left[p_{model}(y|\underline{x})\right]}_{\text{cross entropy between (conditional) data and model distribution (for given } \underline{x})}$$

Empirical risk minimization:

$$E^T = -\frac{1}{P}\sum_{\alpha=1}^{P} \sum_{y\in\{0,1\}} p_{data}(y|\underline{x}^{(\alpha)}) \ln\left[p_{model}(y|\underline{x}^{(\alpha)})\right]$$

$$= \frac{1}{P}\sum_{\alpha=1}^{P} -\Big(p_{data}(y=1|\underline{x}^{(\alpha)}) \ln\left[p_{model}(y=1|\underline{x}^{(\alpha)})\right] + p_{data}(y=0|\underline{x}^{(\alpha)}) \ln\left[\underbrace{p_{model}(y=0|\underline{x}^{(\alpha)})}_{1-y(\underline{x}^{(\alpha)};\underline{w})}\right] \Big)$$

$$= \frac{1}{P}\sum_{\alpha=1}^{P} \Big(\underbrace{-y_T^{(\alpha)} \ln\left[y(\underline{x}^{(\alpha)};\underline{w})\right] - (1-y_T^{(\alpha)}) \ln\left[1-y(\underline{x}^{(\alpha)};\underline{w})\right]}_{\text{cross entropy cost function, } e^{(\alpha)}} \Big)$$

multi-class version: softmax output / multiclass cross entropy, cf. lecture

MLP cost fcts (cont'd)

d)

Cost functions & output layers → show after backprop derivation

task	data	output layer	cost function
scalar regression	$\{(\underline{x}^{(\alpha)}, y_T^{(\alpha)})\}$ $\in \mathbb{R}^N \quad \in \mathbb{R}$	linear $y = \sum_{k=0}^{N} w_{1k}^{L,L-1} s_k^{L-1}$ $= h_1^L$	mean squared error $e^{(\alpha)} = \frac{1}{2}\left[y(\underline{x}^{(\alpha)}; \underline{w}) - y_T^{(\alpha)}\right]^2$
vectorial regression	$\{(\underline{x}^{(\alpha)}, \underline{y}_T^{(\alpha)})\}$ $\in \mathbb{R}^M$	linear $y_i = \sum_{k=0}^{N} w_{ik}^{L,L-1} s_k^{L-1}$ $= h_i^L$	mean squared Euclidean distance $e^{(\alpha)} = \frac{1}{2}\left\| \underline{y}(\underline{x}^{(\alpha)}; \underline{w}) - \underline{y}_T^{(\alpha)} \right\|_2^2$
binary classification	$\{(\underline{x}^{(\alpha)}, y_T^{(\alpha)})\}$ $\in \{0,1\}$	logistic sigmoidal $y = \frac{1}{1+\exp(-h_1^L)}$	cross entropy $e^{(\alpha)} = -y_T^{(\alpha)} \ln y(\underline{x}^{(\alpha)}; \underline{w})$ $-(1-y_T^{(\alpha)}) \ln\left[1 - y(\underline{x}^{(\alpha)}; \underline{w})\right]$
classification with M classes	$\{(\underline{x}^{(\alpha)}, \underline{y}_T^{(\alpha)})\}$ $\in \{0,1\}^M$ 1-hot-code e.g. $\underline{y}_T = (0,0,1,0,0,0)^T$ $\hat{=}$ class 3 of 6	softmax $y_i = \frac{\exp(h_i^L)}{\sum_{k=1}^{M} \exp(h_k^L)}$ $\Rightarrow \sum_{k=1}^{M} y_i = 1$	cross entropy (multi class case) $e^{(\alpha)} = -\sum_{i=1}^{M} y_{T,i}^{(\alpha)} \ln y_i(\underline{x}^{(\alpha)}; \underline{w})$

M-class case: next lecture(s)

Remarks

→ all classes explicitly included: $y_i = P(y=i \mid \underline{x})$

→ binary classification implicit: $y = P(y=1 \mid \underline{x})$

$1-y = P(y=0 \mid \underline{x})$