COURSE 02457

Signal Processing in Non-linear Systems:

Lecture 7

- Exam dates
- The signal detection problem
- Maximum likelihood estimation
- Two class estimation
- Properties of the costfunction
- Multiple classes
- Reduced output coding
- Pruning and weight decay

Signal detection: Bayes decision theory

- A signal detection system (or pattern classifier) provides a rule for assigning a measurement to a given signal category (class)
- Hence, a classifier divides measument space (feature space) into disjoint regions $\mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_c$, such that measurements that fall into region $\mathcal{R}_k$ are assigned with class $\mathcal{C}_k$.
- Boundaries between regions are denoted decision surfaces or decision boundaries

Signal Detection: Bayes decision theory

Figure 1: Schematic plot of the densities for a measured signal drawn from either of two populations $\mathcal{C}_1, \mathcal{C}_2$

$$
\begin{aligned}
P(\text{error}) &= P(x \in \mathcal{R}_2, \mathcal{C}_1) + P(x \in \mathcal{R}_1, \mathcal{C}_2) \\
&= P(x \in \mathcal{R}_2 | \mathcal{C}_1) P(\mathcal{C}_1) + P(x \in \mathcal{R}_1 | \mathcal{C}_2) P(\mathcal{C}_2) \\
&= \left(\int_{\mathcal{R}_2} p(x|\mathcal{C}_1) dx \right) P(\mathcal{C}_1) + \left(\int_{\mathcal{R}_1} p(x|\mathcal{C}_2) dx \right) P(\mathcal{C}_2)
\end{aligned}
$$

- The probability of error is minimized if we assign points to $\mathcal{R}_1$, whenever $p(x|\mathcal{C}_1)P(\mathcal{C}_1) > p(x|\mathcal{C}_2)P(\mathcal{C}_2)$
- Using Bayes' theorem, this is equivalent to assign points to $\mathcal{R}_1$, whenever $p(\mathcal{C}_1|x) > p(\mathcal{C}_2|x)$, since we can divide by $p(\mathbf{x})$ on both sides of the inequality. Hence, the Bayes optimal signal detection system chooses the most probable class given the measurement.

The learning problem

- Supervised learning: Learning relations between sets of variables e.g. between input and output variables, conditional distributions $p(\text{output}|\text{input})$.
- Unsupervised learning: Learning the distribution of a set of variables $p(\text{input})$.

The Bayesian paradigm

- The output density of the measured signals $(t, \mathbf{x})$ is modeled by a parameterized density: $p(t|\mathbf{x}) \sim p(t|\mathbf{x}, \mathbf{w})$.
- Let $\chi = \{(t^1, \mathbf{x}^1), (t^2, \mathbf{x}^2), ..., (t^N, \mathbf{x}^N)\}$ be a *training set*
- Objective: Find the distribution of the parameter vector, $p(\mathbf{w}|\chi)$, hence the parameters are considered stochastic.

The likelihood function

- Let $\chi = \{(t^1, \mathbf{x}^1), (t^2, \mathbf{x}^2), ..., (t^N, \mathbf{x}^N)\}$ be the $training\ set$
- We use Bayes theorem

$$p(\mathbf{w}|\chi) = \frac{p(\chi|\mathbf{w})p(\mathbf{w})}{p(\chi)}$$

- The function $p(\chi|\mathbf{w})$ is called the likelihood function (more correct the likelihood of the parameter vector θ). The density $p(\mathbf{w})$ is called the $a\ priori$ or $prior$ parameter distribution.
- If the prior is "flat" in the neighborhood of the peak of $p(\chi|\mathbf{w})$, we have

$$p(\mathbf{w}|\chi) \propto p(\chi|\mathbf{w})$$

- ...and finding the most probable parameters is equivalent to finding the maximum likelihood parameters.

Maximum likelihood & optimization

- For independent examples, $\chi = \{(t^1, \mathbf{x}^1), (t^2, \mathbf{x}^2), ..., (t^N, \mathbf{x}^N)\}$, the likelihood function factorizes

$$p(\chi|\mathbf{w}) = \prod_{n=1}^{N} p(t^n|\mathbf{x}_n, \mathbf{w})p(\mathbf{x}^n) = p(\chi_t|\chi_\mathbf{x}, \mathbf{w}) * p(\chi_\mathbf{x})$$

- Many algorithms are based on minimizing an index or cost function

$$E(\mathbf{w}) = -\log p(\chi_t|\chi_\mathbf{x}, \mathbf{w}) = \sum_{n=1}^{N} -\log p(t^n|\mathbf{x}_n\mathbf{w})$$

Least squares as maximum likelihood

- We seek a conditional density model of the form

$$
\begin{aligned}
t &= f_{\mathbf{w}}(\mathbf{x}) + \nu \\
p(t|\mathbf{x}, \sigma^2, \mathbf{w}) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(t - f_{\mathbf{w}}(\mathbf{x}))^2\right) \\
p(\chi_t|\chi_{\mathbf{x}}, \sigma^2, \theta) &= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^N \exp\left(-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(t^n - f_{\mathbf{w}}(\mathbf{x}^n)^2\right) \\
E(\mathbf{w}, \sigma^2) &= \frac{N}{2}\log 2\pi\sigma^2 + \frac{1}{2\sigma^2}\sum_{n=1}^{N}(t_n - f_{\mathbf{w}}(\mathbf{x}_n))^2
\end{aligned}
$$

- Hence, maximizing the likelihood for Gaussian noise leads to a least squares problem (for $\mathbf{w}$).
- Note, the noise variance is always given trivially by

$$
\widehat{\sigma^2} = \frac{1}{N}\sum_{n=1}^{N}(t_n - f_{\widehat{\mathbf{w}}}(\mathbf{x}_n))^2
$$

The generalization error: "The Hidden agenda"

- Let a training set of independent examples be given by $\mathcal{D} = \{(t^1, \mathbf{x}^1), ..., (t^N, \mathbf{x}^N)\}$.
- The *training error pr. example* of the model $p(t|\mathbf{x}, \mathbf{w})$ is given by

$$E = \frac{1}{N} \sum_{n=1}^{N} - \log p(t^n|\mathbf{x}^n, \mathbf{w})$$

  this is what we use to find good parameters $\mathbf{w}$.
- However, what we really want is that the probability of future data points is high, i.e., that the typical cost

$$E^k = - \log p(t^k|\mathbf{x}^k, \mathbf{w})$$

  is low. A model that assigns high probability to all future data point is close to the true model, hence, *a good generalizer*.
- So, let us define *the generalization error*:

$$\begin{aligned} E &= \lim_{M \to \infty} \frac{1}{M} \sum_{k=1}^{M} - \log p(t^k|\mathbf{x}^k, \mathbf{w}) \\ &= \int \int - \log[p(t^k|\mathbf{x}^k, \mathbf{w})] p(t|\mathbf{x}) dt p(\mathbf{x}) d\mathbf{x} \end{aligned}$$

  This is the average (or expected) error on a test datum $(t, \mathbf{x})$.

Two class problem

- Let the labels for a two class problem be $t_n = 0$ for one class and $t_n = 1$ for the other.
- Let the network output be $0 \leq y \leq 1$ be the probability of $t = 1$, then
- we can write the likelihood as

$$\begin{aligned} p(\chi_t|\chi_x, \mathbf{w}) &= \prod_{n=1}^{N} p(t_n|\mathbf{x}_n, \mathbf{w}) \\ &= \prod_{n=1}^{N} y(\mathbf{x}_n|\mathbf{w})^{t_n}[1 - y(\mathbf{x}_n|\mathbf{w})]^{(1-t_n)} \end{aligned}$$

- and the costfunction becomes

$$E(\mathbf{w}) = -\sum_{n=1} t_n \log y(\mathbf{x}_n|\mathbf{w}) + (1 - t_n) \log[1 - y(\mathbf{x}_n|\mathbf{w})]$$

- this is called the *entropic costfunction*

Properties of the costfunction

- The cost is minimal if $y_n = t_n$

$$E(\mathbf{w}) = -\sum_{n=1} t_n \log y(\mathbf{x}_n|\mathbf{w}) + (1-t_n)\log[1-y(\mathbf{x}_n|\mathbf{w})]$$

- the derivative w.r.t. y is

$$\frac{\partial E}{\partial y_n} = -\left[\frac{t_n}{y_n} - \frac{1-t_n}{1-y_n}\right]$$

Figure 2: Entropic costfunction: dependence on y_n

Properties of the costfunction cont'd

- The entropic costfunction penalizes wrong decisions more heavily than least squares method:

$$E(\mathbf{w}) = -\sum_{n=1} t_n \log y(\mathbf{x}_n|\mathbf{w}) + (1-t_n)\log[1-y(\mathbf{x}_n|\mathbf{w})]$$

- Let $t_n = 1$ and $y_n = 1 - \epsilon_n$ (correct decision)

$$E^n = -\log y_n = -\log[1-\epsilon_n] \approx \epsilon_n$$

- Let $t_n = 0$ and $y_n = 1 - \epsilon_n$ (very wrong decision!)

$$E^n = -\log[1-y_n] = -\log[1-(1-\epsilon_n)] = \log \epsilon_n$$

- Now for the sum of squares costfunction we have for the same situations

$$E^n = (1-(1-\epsilon_n))^2 = (\epsilon_n)^2$$
$$E^n = (0-(1-\epsilon_n))^2 = (1-\epsilon_n)^2$$

Properties of the costfunction cont'd

- We use $0 \leq y \leq 1$ coding of output unit based on linear standard MLP ($a_o(\mathbf{x}|\mathbf{w})$)

$$y(\mathbf{x}|\mathbf{w}) = \frac{1}{1 + \exp(-a_o(\mathbf{x}|\mathbf{w}))}$$

- Backprop rule

$$\frac{\partial E^n}{\partial \mathbf{w}_{jk}} = \delta_j^n z_k^n$$

- The output weight derivative is given by

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

  the derivative is given by $g'(a) = g(a)[1 - g(a)]$

- and the output error is then simply

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

Multiple classes

- We use $0 \leq y\ 1$ coding for C classes and we want the outputs to be the posterior probabilities $P(C|\mathbf{x})$, hence they "should sum to one"

$$y_k(\mathbf{x}) = \frac{\exp a_k(\mathbf{x})}{\sum_k \exp a_k(\mathbf{x})}$$

- Targets are represented by 0-1 vectors:

$$\mathbf{t}_k = [0, 0, 0, ..., 1, 0, 0]$$

- The likelihood function is given by

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

Multiple classes cont'd

- The likelihood function is given by

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

$$E = -\sum_n \sum_k t_k^n \log y_k^n$$

- The derivatives are relatively simple again

$$\frac{\partial E^n}{\partial a_k} = \sum_{k'} \frac{\partial E^n}{\partial y_{k'}} \frac{\partial y_{k'}}{\partial a_k}$$

$$\frac{\partial y_{k'}}{\partial a_k} = \delta_{kk'} y_k - y_{k'} y_k$$

$$\frac{\partial E^n}{\partial y_{k'}} = -\frac{t_{k'}}{y_{k'}}$$

- and we find

$$
\begin{aligned}
\frac{\partial E^n}{\partial a_k} &= \sum_{k'} \frac{t_{k'}}{y_{k'}} (\delta_{kk'} y_k - y_k y_{k'}) \\
&= -(t_k - y_k \sum_{k'} t_{k'}) \\
&= y_k - t_k
\end{aligned}
$$

Expansion around a minimum

- At a minimum $\nabla E = 0$

$$E(\mathbf{w}) \approx E(\mathbf{w}^*) + \frac{1}{2}(\mathbf{w} - \mathbf{w}^*)^T H(\mathbf{w}^*)(\mathbf{w} - \mathbf{w}^*)$$

- The Hessian is real and symmetric, hence a has a set of orthonormal eigenvectors

$$\mathbf{H}\mathbf{u}_j = \lambda_j \mathbf{u}_j$$
$$\mathbf{u}_i^\top \mathbf{u}_j = \delta_{ij}$$

- At a minimum the Hessian is positive: $\mathbf{v}^\top \mathbf{H}\mathbf{v} > 0$
- in particular for all eigenvectors

$$\mathbf{u}_j^\top \mathbf{H}\mathbf{u}_j = \lambda_j > 0$$

- hence, when the Hessian is positive all eigenvalues are positive

Regularization by weight decay

- Weight decay is a means of soft capacity control

$$\tilde{E}(\mathbf{w}) = E(\mathbf{w}) + \frac{1}{2}\nu \mathbf{w}^T \mathbf{w}$$

- Analysis of weight decay: Second order Taylor expansion of the costfunction

$$\begin{aligned} E(\mathbf{w}) &\approx E(\mathbf{w}_0) + \sum_j \frac{\partial E}{\partial w_j}\left(w_j - w_{0,j}\right) \\ &\quad + \frac{1}{2}\sum_{j,k} \frac{\partial^2 E}{\partial w_j \partial w_k}\left(w_j - w_{0,j}\right)\left(w_k - w_{0,k}\right) \\ E(\mathbf{w}) &\approx E(\mathbf{w}_0) + \frac{\partial E}{\partial \mathbf{w}}\left(\mathbf{w} - \mathbf{w}_0\right) \\ &\quad + \frac{1}{2}\left(\mathbf{w} - \mathbf{w}_0\right)^T \frac{\partial^2 E}{\partial \mathbf{w} \partial \mathbf{w}^T}\left(\mathbf{w} - \mathbf{w}_0\right) \end{aligned}$$

- The matrix $\mathbf{H} = \frac{\partial^2 E}{\partial \mathbf{w} \partial \mathbf{w}^T}$ is called the $Hessian$

Weight decay

- Analysis of weight decay:

$$E(\mathbf{w}) \approx E(\mathbf{w}_0) + \frac{\partial E}{\partial \mathbf{w}}(\mathbf{w} - \mathbf{w}_0) + \frac{1}{2}(\mathbf{w} - \mathbf{w}_0)^T \mathbf{H}(\mathbf{w} - \mathbf{w}_0)$$

- Hence the minimum solves

$$\frac{\partial E}{\partial \mathbf{w}} + \mathbf{H}(\mathbf{w} - \mathbf{w}_0) = 0$$

$$\mathbf{w}^* = \mathbf{w}_0 - \mathbf{H}^{-1}\frac{\partial E}{\partial \mathbf{w}}$$

Weight decay

- Now if there is a non-zero weight decay

$$\frac{\partial \tilde{E}(\mathbf{w})}{\partial \mathbf{w}} = \frac{\partial E}{\partial \mathbf{w}} + \nu \mathbf{w}$$

- Hence the new minimum solves

$$\frac{\partial E}{\partial \mathbf{w}_0} + \mathbf{H}_0(\mathbf{w}^* - \mathbf{w}_0) + \nu(\mathbf{w}^* - \mathbf{w}_0) = 0$$

- while the old minimum solves

$$\frac{\partial E}{\partial \mathbf{w}_0} + \mathbf{H}_0(\mathbf{w} - \mathbf{w}_0) = 0$$

- this means that the new and the old minima are related as

$$\mathbf{w}^* - \mathbf{w}_0 = (\mathbf{H}_0 + \nu \mathbf{1})^{-1} \mathbf{H}_0(\mathbf{w} - \mathbf{w}_0)$$

Saliency: Optimal Brain Damage

- How much does the training error increase if we delete a weight
- Second order expansion:

$$E(\mathbf{w}) \approx E(\mathbf{w}^*) + \frac{\partial E}{\partial \mathbf{w}}(\mathbf{w} - \mathbf{w}^*) + \frac{1}{2}(\mathbf{w} - \mathbf{w}^*)^T \mathbf{H} (\mathbf{w} - \mathbf{w}^*)$$

- Deletion of the j'th weight: $\mathbf{w} - \mathbf{w}^* = w_j \mathbf{e}_j$

$$E(\mathbf{w}) \approx E(\mathbf{w}^*) + \frac{\partial E}{\partial \mathbf{w}} w_j \mathbf{e}_j + \frac{1}{2} w_j \mathbf{e}_j^T \mathbf{H} w_j \mathbf{e}_j$$

$$E(\mathbf{w}) \approx E(\mathbf{w}^*) + \frac{\partial E}{\partial w_j} w_j + \frac{1}{2} \mathbf{H}_{j,j} w_j^2$$

Saliency: Optimal Brain Damage

- However, in the minimum the first derivative is zero, hence

$$\Delta E(\mathbf{w})_{\mathrm{obd}} \approx \frac{1}{2}\mathbf{H}_{jj}w_j^2$$

  defining the OBD *saliency*

- If the retraining contribution is included (the un-pruned weights are not optimal after pruning) we get instead the OBS saliency

$$\Delta E(\mathbf{w})_{\mathrm{obs}} \approx \frac{1}{2}\frac{w_j^2}{(\mathbf{H}^{-1})_{jj}}$$

Pruning and example

- We use pruning by OBD
- Stop pruning based on generalization error