5.1 Consider a two-layer network function of the form (5.7) in which the hidden-unit non-linear activation functions $g(\cdot)$ are given by logistic sigmoid functions of the form $\sigma(a)=\{1+\exp(-a)\}^{-1}$. Show that there exists an equivalent work, which computes exactly the Same function, but with hidden unit activation functions given by $\tanh(a)$ where the tanh function is defined by (5.59). Hint: first find the relation between $\sigma(a)$ and $\tanh(a)$, and then show that the parameters of the two networks differ by linear transformations.

Sol. Sigmoid function $f(x)=\frac{1}{1+e^{-x}}$, tanh function $\tanh x=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$

So. $\tanh x=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}+1-1=\frac{e^{x}-e^{-x}+e^{x}+e^{-x}}{e^{x}+e^{-x}}-1=\frac{2e^{x}}{e^{x}+e^{-x}}-1=\frac{2}{1+e^{-2x}}-1=2f(2x)-1$

$\therefore f(x)=\frac{1}{2}\tanh\frac{1}{2}x+\frac{1}{2}$

$\therefore$ from (5.7). $y_k(x,w)=\sigma\left(\sum_{j=1}^{M}w_{kj}^{(2)}h\left(\sum_{i=1}^{D}w_{ji}^{(1)}x_i+w_{j0}^{(1)}\right)+w_{k0}^{(2)}\right)$

$$h\left(\sum_{i=1}^{D}w_{ji}^{(1)}x_i+w_{j0}^{(1)}\right)=\frac{1}{2}\tanh\left[\frac{1}{2}\left(\sum_{i=1}^{D}w_{ji}^{(1)}x_i+w_{j0}^{(1)}\right)\right]+\frac{1}{2}=\frac{1}{2}\tanh\left(\frac{1}{2}\sum_{i=1}^{D}w_{ji}^{(1)}x_i+\frac{1}{2}w_{j0}^{(1)}\right)+\frac{1}{2}$$

$$\therefore y(x,w)=\sigma\left(\sum_{j=1}^{M}w_{kj}^{(2)}\cdot\frac{1}{2}\cdot\tanh\left(\sum_{i=1}^{D}\frac{1}{2}w_{ji}^{(1)}x_i+\frac{1}{2}w_{j0}^{(1)}\right)+\frac{1}{2}w_{kj}^{(2)}+w_{k0}^{(2)}\right)$$
$$=\sigma\left(\sum_{j=1}^{M}\mu_{kj}^{(2)}\tanh\left(\sum_{i=1}^{D}\mu_{ji}^{(1)}x_i+\mu_{j0}^{(1)}\right)+\mu_{k0}^{(2)}\right)$$

where. $\mu_{kj}^{(2)}=\frac{1}{2}w_{kj}^{(2)}$ $\quad\mu_{ji}^{(1)}=\frac{1}{2}w_{ji}^{(1)}$ $\quad\mu_{j0}^{(1)}=\frac{1}{2}w_{j0}^{(1)}$ $\quad\mu_{k0}^{(2)}=\frac{1}{2}\sum_{j=1}^{M}w_{kj}^{(2)}+w_{k0}^{(2)}$

5.2. Show that maximizing the likelihood function under the conditional distribution (5.16) for a multioutput neural network is equivalent to minimizing the sum-of-squares error function.

Sol. from 5.16 $p(t|x,w)=\mathcal{N}(t|y(x,w),\beta^{-1}I)$

So, the maximum likelihood is $p(t|X,w)=\prod_{n=1}^{N}\mathcal{N}(t_n|y(x_n,w),\beta^{-1}I)$

remove the terms independent with w, we get

$$p(t|X,w)\propto\exp\left\{\sum_{n=1}^{N}-(t_n-y(x_n,w))^{T}\beta I(t_n-y(x_n,w))\right\}$$
$$=\exp\left\{\sum_{n=1}^{N}-\beta(t_n-y(x_n,w))^2\right\}=\exp\left\{-\beta\sum_{n=1}^{N}[t_n-y(x_n,w)]^2\right\}$$

So, the problem changes into maximize $-\sum_{n=1}^{N}[t_n-y(x_n,w)]^2$ which is equivalent to minimize the sum-squares error function (5.11).

5.4. Consider a binary classification problem in which the target values are $t \in \{0,1\}$, with a network output $y(x,w)$ that represents $p(t=1|x)$, and suppose that there is a ~~prop~~ probability ϵ that the class label on a training data point has been incorrectly set. Assuming independ~~ata~~ and identically distributed data, write down the error function corresponding to the negative log likelihood. Verify that the error function (5.21) is obtained when $\epsilon = 0$, note that this error function makes the model robust to incorrectly labelled data. in contrast to the usual error function.

Sol. from. 5.20. $p(t|x,w) = y(x,w)^t \{1-y(x,w)\}^{1-t}$

so. $\begin{cases} p(t=1|x,w) = y(x,w) \\ p(t=0|x,w) = 1-y(x,w) \end{cases}$

and if it exists a ~~error~~ probability of error label of ϵ. so.

$$\begin{cases} p(t=1|x,w) = y(x,w)(1-\epsilon) + [1-y(x,w)]\epsilon \\ p(t=0|x,w) = [1-y(x,w)](1-\epsilon) + y(x,w)\epsilon \end{cases}$$

$\because$ $p(t_n|x,w) = p(t=1|x,w)^{t_n} \cancel{\cdot} \, p(t=0|x,w)^{1-t}$

$\therefore$ $p(t|x,w,\epsilon) = \{y(x,w)(1-\epsilon) + [1-y(x,w)]\epsilon\}^t \cdot \{[1-y(x,w)](1-\epsilon) + y(x,w)\epsilon\}^{1-t}$

$$-\ln p(t|X,w,\epsilon) = -\ln \prod_{n=1}^{N} p(t_n|x_n,w,\epsilon) = \ln \prod_{n=1}^{N} \{y(x_n,w)(1-\epsilon) - [1-y(x_n,w)]\epsilon\}^{t_n} \cdot \{[1-y(x_n,w)](1-\epsilon) + y(x_n,w)\epsilon\}^{1-t_n}$$

$$= -\sum_{n=1}^{N} \{t_n \ln\{y(x_n,w)(1-\epsilon) - [1-y(x_n,w)]\epsilon\} + (1-t_n)\ln\{[1-y(x_n,w)](1-\epsilon) + y(x_n,w)\epsilon\}$$

$$\text{So. } E(w) = -\sum_{n=1}^{N} \{t_n \ln\{y(x_n,w)(1-\epsilon) - [1-y(x_n,w)]\epsilon\} + (1-t_n)\ln\{[1-y(x_n,w)](1-\epsilon) + y(x_n,w)\epsilon\}\}$$

5.9. The error function (5.21) for binary classification problems was derived for a network having a logistic-sigmoid output activation function, so that $0 \le y(x,w) \le 1$, and data ~~driving~~ having target values $t \in \{0,1\}$. Derive the corresponding error function if we consider a network having an output $-1 \le y(x,w) \le 1$ and target values $t=1$ for class C_1 and $t=-1$ for class C_2. What should be the appropriate choice of output unit activation function?

Sol. Because we need to project the output into -1 to 1, so. the tanh activation function will be a very good choice, ~~bee~~ $\tanh a \in (-1,1)$

$\tanh a = 2\,\mathrm{sigmoid}(2a) - 1$ so. ~~tah~~ $\tanh \frac{a}{2} = 2 \cdot \mathrm{sigmoid}\, a - 1$

$0 < \sigma(a) < 1$

$0 < 2\sigma(a) < 2$

$-1 < 2\sigma(a) - 1 < 1$

if the activation function is Sigmoid function, we can get

(5.21). $E(w) = -\sum_{n=1}^{N}\{t_n y_n + (1-t_n)\ln(1-y_n)\}$

~~So we need to transform the t_n, y_n from (0,1) to (-1,1).~~

~~So, t_n~~ So, if the target is (-1,1), in order to satisfy the above condition, we need trans the t_n, y_n from (-1,1) to (0,1) so,

$t_n' = \frac{1+t_n}{2} \quad y_n' = \frac{1+y_n}{2}$

$$
\begin{aligned}
\therefore E(w) &= -\sum_{n=1}^{N}\left\{\frac{1+t_n}{2}\ln\frac{1+y_n}{2} + \left(1-\frac{1+t_n}{2}\right)\ln\left(1-\frac{1+y_n}{2}\right)\right\} \\
&= -\sum_{n=1}^{N}\left\{\frac{1+t_n}{2}\left[\ln(1+y_n) - \ln 2\right] + \frac{1-t_n}{2}\left[\ln(1-t_n) - \ln 2\right]\right\} \\
&= -\frac{1}{2}\sum_{n=1}^{N}\left\{(1+t_n)\ln(1+y_n) + (1-t_n)\ln(1-t_n) - 2\ln 2\right\} \\
&= -\frac{1}{2}\sum_{n=1}^{N}\left\{(1+t_n)\ln(1+y_n) + (1-t_n)\ln(1-t_n)\right\} + N\ln 2
\end{aligned}
$$

From the first analyse, we can change sigmoid function to tanh function to ~~satif~~ satisfy the conditions better.

5.16 The ~~output~~ outer ~~pro~~ product approximation to the Hessian matrix for a neural network using a sum-of-squares error function is given by (5.84). Extend this result to the case of multiple outputs.

Sol. The multiple form of loss function is $E = \frac{1}{2}\sum_{n=1}^{N}(y_n - t_n)^T(y_n - t_n) = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K}(y_{nk} - t_{nk})^2$

$$
\text{So, } \frac{\partial E_n}{\partial w_i} = \sum_{k=1}^{K}\frac{\partial E_n}{\partial y_{nk}}\cdot\frac{\partial y_{nk}}{\partial w_i} = \sum_{k=1}^{K}(y_{nk} - t_{nk})\frac{\partial y_{nk}}{\partial w_i}
$$

$$
\begin{aligned}
\therefore \frac{\partial^2 E_n}{\partial w_i \partial w_j} &= \sum_{k=1}^{K}\left[\frac{\partial y_{nk}}{\partial w_j}\cdot\frac{\partial y_{nk}}{\partial w_i} + (y_{nk} - t_{nk})\cdot\frac{\partial^2 y_{nk}}{\partial w_i \partial w_j}\cdot \cancel{\frac{\partial y_{nk}}{}}\right] \\
&= \sum_{k=1}^{K}\left[\frac{\partial y_{nk}}{\partial w_j}\cdot\frac{\partial y_{nk}}{\partial w_i} + (y_{nk} - t_{nk})\frac{\partial^2 y_{nk}}{\partial w_i \partial w_j}\right]
\end{aligned}
$$

$$
\therefore \frac{\partial E}{\partial w_i \partial w_j} = \sum_{n=1}^{N}\left\{\frac{\partial y_n}{\partial w_j}^T\cdot\frac{\partial y_n}{\partial w_i} + (y_n - t_n)^T\cdot\frac{\partial^2 y_n}{\partial w_i \partial w_j}\right\}
$$

$\therefore H = \sum_{n=1}^{N} B_n \cdot B_n^T$. if we neglect the second term for the univariate case.

and H is a M×M matrix. where M is the dimention of W

~~$\frac{\partial^2 E_n}{\partial w_i \partial w_j} = \sum_{k=1}^{K}\left[\frac{\partial y_{nk}}{\partial w_j}\frac{\partial y_{nk}}{\partial w_j}\right]$~~ $\frac{\partial^2 E_n}{\partial w_i \partial w_j} = \sum_{k=1}^{K}\left[\frac{\partial y_{nk}}{\partial w_j}\cdot\frac{\partial y_{nk}}{\partial w_i}\right] = (B_n \cdot B_n^T)_{ij}$

~~$(B_n)_{ij} = \frac{\partial y_{nk}}{\partial}$~~ $\therefore (B_n)_{mk} = \frac{\partial y_{nk}}{\partial w_m}$, B_n is a matrix of M×K

5.19. Derive the expression (5.85) for the outer product approximation to the Hessian matrix for a network having a single output with a logistic sigmoid output-unit activation function and a cross-entropy error function, corresponding to the result (5.84) for the sum-of-squares error function.

Sol. Loss function is cross-entropy. error function. so

$$E(w) = -\sum_{n=1}^{N} \{t_n \ln y_n + (1-t_n)\ln(1-y_n)\}$$

$$\frac{\partial E_n}{\partial w_i} = \frac{\partial E_n}{\partial y_n} \cdot \frac{\partial y_n}{\partial a_n} \cdot \frac{\partial a_n}{\partial w_i} = -\left(\frac{t_n}{y_n} - \frac{1-t_n}{1-y_n}\right) \times (1-y_n) y_n \times \frac{\partial a_n}{\partial w_i} = (-t_n + y_n)\frac{\partial a_n}{\partial w_i}$$

$$\therefore \frac{\partial^2 E_n}{\partial w_i \partial w_j} = \frac{\partial y_n}{\partial w_j} \cdot \frac{\partial a_n}{\partial w_i} + (-t_n + y_n) \cdot \frac{\partial^2 a_n}{\partial w_i \partial w_j}$$

$$= y_n(1-y_n) \cdot \frac{\partial a_n}{\partial w_j} \cdot \frac{\partial a_n}{\partial w_i} + (-t_n + y_n) \cdot \frac{\partial^2 a_n}{\partial w_i \partial w_j}$$

So. if we neglect the second term. we can get

$$\frac{\partial^2 E_n}{\partial w_i \partial w_j} = y_n(1-y_n) \cdot \frac{\partial a_n}{\partial w_j} \cdot \frac{\partial a_n}{\partial w_i}$$

$$\therefore H = \frac{\partial E}{\partial w} = \sum_{n=1}^{N} y_n(1-y_n) \cdot b_n \cdot b_n^T. \quad \text{where. } b_n = \frac{\partial a_n}{\partial w}$$

5.34. Derive the result (5.155) for the derivate of the error function with respect to the network output activations controlling the componet means in the mixture density network.

Sol. $\frac{\partial E_n}{\partial a_k^{\pi}} = \sum_{j=1}^{K} \frac{\partial E_n}{\partial \pi_j} \cdot \frac{\partial \pi_j}{\partial a_k^{\pi}}$

from. (5.153). $\frac{\partial E_n}{\partial \pi_j} = \frac{N_{nj}}{-\sum_{l=1}^{K} \pi_l N_{nl}} = -\frac{\gamma_{nj}}{\pi_j}$

from (4.106). $\frac{\partial \pi_j}{\partial a_k^{\pi}} = \pi_j (I_{jk} - \pi_k)$

$$\therefore \frac{\partial E_n}{\partial a_k^{\pi}} = -\sum_{j=1}^{K} \frac{\gamma_{nj}}{\pi_j} \cdot \pi_j (I_{jk} - \pi_k) = -\sum_{j=1}^{K} \gamma_{nj} (I_{jk} - \pi_k)$$

$$= -\gamma_{nk} + \sum_{j=1}^{K} \gamma_{nj} \pi_k = \pi_k - \gamma_{nk}$$

5.37. Verify the results (5.158) and (5.160) for the conditional mean and variance of the mixture density network model.

Sol. $E[t|x] = \int t\, p(t|x)\, dt$

$$= \int t \sum_{k=1}^{K} \pi_k(x) \cdot N(t|\mu_k, \sigma_k^2 I)\, dt$$

$$= \sum_{k=1}^{K} \pi_k(x) \cdot \int t \cdot N(t|\mu_k, \sigma_k^2 I)\, dt$$

$$= \sum_{k=1}^{K} \pi_k(x) \cdot \mu_k(x)$$

$$s^2(x) = E\left[\|t - E[t|x]\|^2 \,|\, x\right] = \int \|t - E[t|x]\|^2 \cdot p(t|x)\, dt$$

assum $\bar{t} = E[t|x] = \sum_{k=1}^{K} \pi_k(x) \cdot \mu_k(x)$

$$\therefore s^2(x) = \int \|t-\bar{t}\|^2 p(t|x)\, dt = \int (t-\bar{t})^T (t-\bar{t})\, p(t|x)\, dt$$

$$= \int (t^T t - t^T \bar{t} - \bar{t}^T t + \bar{t}^T \bar{t}) \cdot \sum_{k=1}^{K} \pi_k(x) N(t|\mu_k, \sigma_k^2 I)\, dt$$

$$= \sum_{k=1}^{K} \pi_k(x) \int (t^T t - t^T \bar{t} - \bar{t}^T t + \bar{t}^T \bar{t}) \cdot N(t|\mu_k, \sigma_k^2 I)\, dt$$

$$= \sum_{k=1}^{K} \pi_k(x) \cdot (\mu_k^T \cdot \mu_k + \sigma_k^2 - \mu_k^T \bar{t} - \bar{t}^T \cdot \mu_k + \bar{t}^T \cdot \bar{t})$$

$$= \sum_{k=1}^{K} \pi_k(x) \left[\sigma_k^2 + (\mu_k - \bar{t})^T (\mu_k - t)\right]$$

$$= \sum_{k=1}^{K} \pi_k(x) \left[\sigma_k^2 + \|\mu_k - \bar{t}\|^2\right] = \sum_{k=1}^{K} \pi_k(x) \left\{\sigma_k^2(x) + \left\|\mu_k(x) - \sum_{l=1}^{K} \pi_l(x) \mu_l(x)\right\|^2\right\}$$

5.38. Using the general result result (2.115), derive the predictive distribution 5.172 for the laplace approximation to the Bayesian neural network model.

Sol. $P(w|D) = N(w|w_{MAP}, A^{-1})$

$$p(t|x, w, \beta) \simeq N(t|y(x, w_{MAP}) + g^T(w - w_{MAP}), \beta^{-1})$$

from 2.113. and 2.114. $\begin{cases} P(x) = N(x|\mu, \Lambda^{-1}) \\ P(y|x) = N(y|Ax+b, L^{-1}) \end{cases}$

So, $\mu \Rightarrow w_{MAP}$, $\Lambda^{-1} \Rightarrow A^{-1}$, $x \Rightarrow w$, $A \Rightarrow g^T$, $y \Rightarrow t$, $b = y(x, w_{MAP}) - g^T w_{MAP}$. $L^{-1} \Rightarrow \beta^{-1}$

$\therefore$ from 2.115 $P(y) = N(y|A\mu+b, L^{-1} + A\Lambda^{-1}A^T)$

$$\therefore A\mu + b = g^T \cdot w_{MAP} + y(x, w_{MAP}) - g^T w_{MAP}$$

$$L^{-1} + A\Lambda^{-1}A^T = \beta^{-1} + g^T \cdot A^{-1} \cdot g$$

$$\therefore P(y) = N(y|g^T w_{MAP} + y(x, w_{MAP}) - g^T w_{MAP}, \beta^{-1} + g^T A^{-1} g)$$

5.39. Make use of the Laplace approximation result (4.135) to show that the evidence function for the hyperparameters α and β in the Bayesian neural network model can be approximated by (5.175).

Sol. from (4.135) ~~$p(z) =$~~ $Z = \int f(z)dz \simeq f(z_0)\int \exp\{-\frac{1}{2}(z-z_0)^T A(z-z_0)\}dz = f(z_0)\cdot\frac{(2\pi)^{\frac{M}{2}}}{|A|^{\frac{1}{2}}}$

~~$p(D|\alpha,\beta) =$~~ assum. $Z \Rightarrow P(D|\alpha,\beta) = \cancel{p(D)}$

$f(z) \Rightarrow p(D|w,\beta)p(w|\alpha)$

$\because z_0 \Rightarrow W_{MAP}$.

$\therefore \int p(D|w,\beta)\,p(w|\alpha)dw = p(D|w_{MAP},\beta)\,p(w_{MAP}|\alpha)\cdot\frac{(2\pi)^{\frac{M}{2}}}{|A|^{\frac{1}{2}}}$

~~$\ln p(D)$~~ $\ln\int p(D|w,\beta)\,p(w|\alpha)dw = \ln p(D|w_{MAP},\beta) + \ln p(w_{MAP}|\alpha) + \frac{M}{2}\ln 2\pi - \frac{1}{2}\ln|A|$

$\ln p(D|w_{MAP},\beta) = -\frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n,w_{MAP})-t_n\}^2 + \frac{N}{2}\ln\beta - \frac{N^2}{2}\ln 2\pi$

$\ln p(w_{MAP}|\alpha) = -\frac{\alpha}{2}w_{MAP}^T w_{MAP} + \frac{W}{2}\ln\alpha - \cancel{\frac{W}{2}\ln 2\pi}\ \frac{W}{2}\ln 2\pi$

$\therefore \ln\int p(D|w,\beta)\,p(w|\alpha)dw = -\frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n-w_{MAP})-t_n\}^2 - \frac{\alpha}{2}W_{MAP}^T W_{MAP} + \frac{N}{2}\ln\beta + \frac{W}{2}\ln\alpha - \frac{1}{2}\ln|A|$
$- \frac{N^2}{2}\ln 2\pi$

$\therefore \ln p(D|\alpha,\beta) = -\frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n-w_{MAP})-t_n\}^2 - \frac{\alpha}{2}w_{MAP}^T w_{MAP} + \frac{N}{2}\ln\beta + \frac{W}{2}\ln\alpha - \frac{1}{2}\ln|A| - \frac{N^2}{2}\ln(2\pi)$