Derivation of MBGD-UR-BN

Dongrui Wu, *Senior Member, IEEE*
Key Laboratory of the Ministry of Education for Image Processing and Intelligent Control, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China
Email: drwu@hust.edu.cn.

Let the training dataset be $\mathcal{D}=\{\mathbf{x}_n,y_n\}_{n=1}^N$, in which $\mathbf{x}_n=[x_{n,1},...,x_{n,D}]^T\in\mathbb{R}^{D\times 1}$ is a D-dimensional feature vector, and $y_n\in\{1,2,...,C\}$ the corresponding class label for a C-class classification problem.

Suppose the TSK fuzzy classifier has R rules, in the following form:

$$\begin{aligned}\text{Rule}_r:\ &\text{IF } x_1 \text{ is } X_{r,1} \text{ and } \cdots \text{ and } x_D \text{ is } X_{r,D},\\ &\text{THEN } y_r^1(\mathbf{x})=w_{r,0}^1+\sum_{d=1}^D w_{r,d}^1\cdot x_d \text{ and}\cdots\text{and } y_r^C(\mathbf{x})=w_{r,0}^C+\sum_{d=1}^D w_{r,d}^C\cdot x_d\end{aligned} \tag{1}$$

where $X_{r,d}$ ($r=1,...,R$; $d=1,...,D$) is the membership function (MF) for the d-th antecedent in the r-th rule, and $w_{r,0}^c$ and $w_{r,d}^c$ ($c=1,...,C$) are the consequent parameters for the c-th class.

Different types of MFs can be used in our algorithm, as long as they are differentiable. For simplicity, Gaussian MFs are considered in this paper, and the membership grade of x_d on $X_{r,d}$ is:

$$\mu_{X_{r,d}}(x_d)=\exp\left(-\frac{(x_d-m_{r,d})^2}{2\sigma_{r,d}^2}\right), \tag{2}$$

where $m_{r,d}$ and $\sigma_{r,d}$ are the center and the standard deviation of the Gaussian MF, respectively.

The output of the TSK fuzzy classifier for the c-th class is:

$$y^c(\mathbf{x})=\frac{\sum_{r=1}^R f_r(\mathbf{x})y_r^c(\mathbf{x})}{\sum_{r=1}^R f_r(\mathbf{x})}, \tag{3}$$

where

$$f_r(\mathbf{x})=\prod_{d=1}^D \mu_{X_{r,d}}(x_d)=\exp\left(-\sum_{d=1}^D\frac{(x_d-m_{r,d})^2}{2\sigma_{r,d}^2}\right) \tag{4}$$

is the firing level of Rule r. We can also re-write (3) as:

$$y^c(\mathbf{x})=\sum_{r=1}^R \overline{f}_r(\mathbf{x})y_r^c(\mathbf{x}), \tag{5}$$

where

$$\overline{f}_r(\mathbf{x})=\frac{f_r(\mathbf{x})}{\sum_{i=1}^R f_i(\mathbf{x})} \tag{6}$$

is the normalized firing level of Rule r.

$$q^c(\mathbf{x})=\frac{\exp(y^c(\mathbf{x}))}{\sum_{k=1}^C\exp(y^k(\mathbf{x}))} \tag{7}$$

Once the output vector $\mathbf{y}(\mathbf{x})=[y^1(\mathbf{x}),...,y^C(\mathbf{x})]^T$ is obtained, the input $\mathbf{x}$ is assigned to the class with the largest $y^c(\mathbf{x})$.

To optimize the TSK fuzzy classifier, we need to fine-tune the antecedent MF parameters $m_{r,d}$ and $\sigma_{r,d}$, and the consequent parameters $w_{r,0}^c$ and $w_{r,d}^c$, where $r=1,...,R$, $d=1,...,D$, and $c=1,...,C$.

A. Mini-batch Gradient Descent (MBGD) Based Optimization

For each mini-batch with N_{bs} training samples,

$$\mathcal{L} = \ell_1 + \eta\ell_2 = -\sum_{n=1}^{N_{bs}} \log\left(\frac{\exp(y^{y_n}(\mathbf{x}_n))}{\sum_{c=1}^{C}\exp(y^c(\mathbf{x}_n))}\right) + \eta\sum_{c=1}^{C}\sum_{r=1}^{R}\sum_{d=1}^{D}\left(w_{r,d}^c\right)^2, \tag{8}$$

where ℓ_1 is the cross-entropy loss between the estimated class probabilities [obtained by applying *softmax* to $\mathbf{y}(\mathbf{x})$] and the true class probabilities, and ℓ_2 the L2 regularization of the rule consequent parameters.

The partial derivatives are:

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial m_{r,d}} &= \sum_{n=1}^{N_{bs}}\sum_{c=1}^{C}\frac{\partial\ell_1}{\partial y^c(\mathbf{x}_n)}\frac{\partial y^c(\mathbf{x}_n)}{\partial f_r(\mathbf{x}_n)}\frac{\partial f_r(\mathbf{x}_n)}{\partial\mu_{X_{r,d}}(x_{n,d})}\frac{\partial\mu_{X_{r,d}}(x_{n,d})}{\partial m_{r,d}} \\
&= \sum_{n=1}^{N_{bs}}\sum_{c=1}^{C}\left[\left[q^c(\mathbf{x}_n)-(y_n==c)\right]\frac{y_r^c(\mathbf{x}_n)\sum_{i=1}^{R}f_i(\mathbf{x}_n)-\sum_{i=1}^{R}f_i(\mathbf{x}_n)y_i^c(\mathbf{x}_n)}{\left[\sum_{i=1}^{R}f_i(\mathbf{x}_n)\right]^2}f_r(\mathbf{x}_n)\frac{x_{n,d}-m_{r,d}}{\sigma_{r,d}^2}\right] \\
&= \sum_{n=1}^{N_{bs}}\sum_{c=1}^{C}\left[\left[q^c(\mathbf{x}_n)-I(y_n==c)\right]\left(y_r^c(\mathbf{x}_n)-\sum_{i=1}^{R}\bar{f}_i(\mathbf{x}_n)y_i^c(\mathbf{x}_n)\right)\bar{f}_r(\mathbf{x}_n)\frac{x_{n,d}-m_{r,d}}{\sigma_{r,d}^2}\right]
\end{aligned}\tag{9}$$

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial \sigma_{r,d}} &= \sum_{n=1}^{N_{bs}}\sum_{c=1}^{C}\frac{\partial\ell_1}{\partial y^c(\mathbf{x}_n)}\frac{\partial y^c(\mathbf{x}_n)}{\partial f_r(\mathbf{x}_n)}\frac{\partial f_r(\mathbf{x}_n)}{\partial\mu_{X_{r,d}}(x_{n,d})}\frac{\partial\mu_{X_{r,d}}(x_{n,d})}{\partial \sigma_{r,d}} \\
&= \sum_{n=1}^{N_{bs}}\sum_{c=1}^{C}\left[\left[q^c(\mathbf{x}_n)-(y_n==c)\right]\frac{y_r^c(\mathbf{x}_n)\sum_{i=1}^{R}f_i(\mathbf{x}_n)-\sum_{i=1}^{R}f_i(\mathbf{x}_n)y_i^c(\mathbf{x}_n)}{\left[\sum_{i=1}^{R}f_i(\mathbf{x}_n)\right]^2}f_r(\mathbf{x}_n)\frac{(x_{n,d}-m_{r,d})^2}{\sigma_{r,d}^3}\right] \\
&= \sum_{n=1}^{N_{bs}}\sum_{c=1}^{C}\left[\left[q^c(\mathbf{x}_n)-I(y_n==c)\right]\left(y_r^c(\mathbf{x}_n)-\sum_{i=1}^{R}\bar{f}_i(\mathbf{x}_n)y_i^c(\mathbf{x}_n)\right)\bar{f}_r(\mathbf{x}_n)\frac{(x_{n,d}-m_{r,d})^2}{\sigma_{r,d}^3}\right]
\end{aligned}\tag{10}$$

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial w_{r,d}^c} &= \sum_{n=1}^{N_{bs}}\frac{\partial\ell_1}{\partial y^c(\mathbf{x}_n)}\frac{\partial y^c(\mathbf{x}_n)}{\partial y_r^c(\mathbf{x}_n)}\frac{\partial y_r^c(\mathbf{x}_n)}{\partial w_{r,d}^c}+\eta\frac{\partial\ell_2}{\partial w_{r,d}^c} \\
&= \sum_{n=1}^{N_{bs}}\left[q^c(\mathbf{x}_n)-(y_n==c)\right]\bar{f}_r(\mathbf{x}_n)\cdot x_{n,d}+2\eta w_{r,d}^c\cdot I(d>0)
\end{aligned}\tag{11}$$

where $I(y_n == c)$, $I(i == r)$ and $I(d > 0)$ are indicator functions.

B. Uniform Regularization (UR)

ℓ_{UR} can then be added to the original loss function in MBGD-based training of TSK fuzzy classifiers, i.e., for each mini-batch with N training samples,

$$\mathcal{L} = \ell_1 + \eta\ell_2 + \lambda\ell_{UR} \tag{12}$$

$$= -\sum_{n=1}^{N_{bs}} \log\left(\frac{\exp(y^{y_n}(\mathbf{x}_n))}{\sum_{c=1}^{C}\exp(y^c(\mathbf{x}_n))}\right) + \eta\sum_{c=1}^{C}\sum_{r=1}^{R}\sum_{d=1}^{D}\left(w_{r,d}^c\right)^2 + \lambda\sum_{r=1}^{R}\left(\frac{1}{N_{bs}}\sum_{n=1}^{N_{bs}}\overline{f}_r(\mathbf{x}_n)-\frac{1}{R}\right)^2, \tag{13}$$

The partial derivatives are:

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial m_{r,d}} &= \sum_{n=1}^{N_{bs}} \sum_{c=1}^{C} \frac{\partial \ell_1}{\partial y^c(\mathbf{x}_n)} \frac{\partial y^c(\mathbf{x}_n)}{\partial f_r(\mathbf{x}_n)} \frac{\partial f_r(\mathbf{x}_n)}{\partial \mu_{X_{r,d}}(x_{n,d})} \frac{\partial \mu_{X_{r,d}}(x_{n,d})}{\partial m_{r,d}} + \lambda \sum_{n=1}^{N_{bs}} \frac{\partial \ell_{UR}}{\partial f_r(\mathbf{x}_n)} \frac{\partial f_r(\mathbf{x}_n)}{\partial \mu_{X_{r,d}}(x_{n,d})} \frac{\partial \mu_{X_{r,d}}(x_{n,d})}{\partial m_{r,d}} \\
&= \sum_{n=1}^{N_{bs}} \sum_{c=1}^{C} \left[[q^c(\mathbf{x}_n) - I(y_n == c)] \left(y_r^c(\mathbf{x}_n) - \sum_{i=1}^{R} \bar{f}_i(\mathbf{x}_n) y_i^c(\mathbf{x}_n) \right) \bar{f}_r(\mathbf{x}_n) \frac{x_{n,d} - m_{r,d}}{\sigma_{r,d}^2} \right] \\
&+ \frac{2\lambda}{N_{bs}} \sum_{n=1}^{N_{bs}} \left[\sum_{i=1}^{R} \left[\left(\frac{1}{N_{bs}} \sum_{k=1}^{N_{bs}} \bar{f}_i(\mathbf{x}_k) - \frac{1}{R} \right) \left(I(i == r) - \bar{f}_i(\mathbf{x}_n) \right) \right] \bar{f}_r(\mathbf{x}_n) \frac{x_{n,d} - m_{r,d}}{\sigma_{r,d}^2} \right] \qquad (14)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \sigma_{r,d}} &= \sum_{n=1}^{N_{bs}} \sum_{c=1}^{C} \frac{\partial \ell_1}{\partial y^c(\mathbf{x}_n)} \frac{\partial y^c(\mathbf{x}_n)}{\partial f_r(\mathbf{x}_n)} \frac{\partial f_r(\mathbf{x}_n)}{\partial \mu_{X_{r,d}}(x_{n,d})} \frac{\partial \mu_{X_{r,d}}(x_{n,d})}{\partial \sigma_{r,d}} + \lambda \frac{\partial \ell_{UR}}{\partial f_r(\mathbf{x}_n)} \frac{\partial f_r(\mathbf{x}_n)}{\partial \mu_{X_{r,d}}(x_{n,d})} \frac{\partial \mu_{X_{r,d}}(x_{n,d})}{\partial \sigma_{r,d}} \\
&= \sum_{n=1}^{N_{bs}} \sum_{c=1}^{C} \left[[q^c(\mathbf{x}_n) - I(y_n == c)] \left(y_r^c(\mathbf{x}_n) - \sum_{i=1}^{R} \bar{f}_i(\mathbf{x}_n) y_i^c(\mathbf{x}_n) \right) \bar{f}_r(\mathbf{x}_n) \frac{(x_{n,d} - m_{r,d})^2}{\sigma_{r,d}^3} \right] \\
&+ \frac{2\lambda}{N_{bs}} \sum_{n=1}^{N_{bs}} \left[\sum_{i=1}^{R} \left[\left(\frac{1}{N_{bs}} \sum_{k=1}^{N_{bs}} \bar{f}_i(\mathbf{x}_k) - \frac{1}{R} \right) \left(I(i == r) - \bar{f}_i(\mathbf{x}_n) \right) \right] \frac{(x_{n,d} - m_{r,d})^2}{\sigma_{r,d}^3} \right] \qquad (15)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w_{r,d}^c} &= \sum_{n=1}^{N_{bs}} \frac{\partial \ell_1}{\partial y^c(\mathbf{x}_n)} \frac{\partial y^c(\mathbf{x}_n)}{\partial y_r^c(\mathbf{x}_n)} \frac{\partial y_r^c(\mathbf{x}_n)}{\partial w_{r,d}^c} + \eta \frac{\partial \ell_2}{\partial w_{r,d}^c} \\
&= \sum_{n=1}^{N_{bs}} [q^c(\mathbf{x}_n) - (y_n == c)] \, \bar{f}_r(\mathbf{x}_n) \cdot x_{n,d} + 2\eta w_{r,d}^c \cdot I(d > 0) \qquad (16)
\end{aligned}
$$

where $I(y_n == c)$, $I(i == r)$ and $I(d > 0)$ are indicator functions.

C. Batch Normalization (BN)

BN normalizes the data distribution in each mini-batch to accelerate the training. For a mini-batch $\mathcal{B} = \{\mathbf{x}_n\}_{n=1}^{N_{bs}}$, the output of BN is:

$$
\mathbf{x}'_n = BN(\mathbf{x}_n) = \gamma \frac{\mathbf{x}_n - \bar{\mathbf{m}}_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} + \beta, \qquad (17)
$$

where $\bar{\mathbf{m}}_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$ are the mean and the standard deviation of the samples in the mini-batch, respectively, γ and β are parameters to be learned during training, and ϵ is usually set to $1e-8$ to avoid being divided by zero. During training, exponential weighted averages of $\bar{\mathbf{m}}_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$ are recorded so that they can be used in the test phase.

In the training phase, we first compute the firing level of each rule using the unmodified inputs, as in traditional TSK fuzzy systems. Then, we use BN to normalize the inputs, according to their mean and standard deviation in the current mini-batch. The normalized inputs are then used to compute the rule consequents. The final output is a weighted average of the rule consequents, the weights being the corresponding rule firing levels.

The loss function is

$$
\mathcal{L} = \ell_1 + \eta \ell_2 = -\sum_{n=1}^{N_{bs}} \log \left(\frac{\exp(y^{y_n}(\mathbf{x}_n))}{\sum_{c=1}^{C} \exp(y^c(\mathbf{x}_n))} \right) + \eta \sum_{c=1}^{C} \sum_{r=1}^{R} \sum_{d=1}^{D} \left(w_{r,d}^c \right)^2, \qquad (18)
$$

where

$$
y_r^c(\mathbf{x}) = w_{r,0}^c + \sum_{d=1}^{D} w_{r,d}^c \left(\gamma \frac{x_d - \bar{m}_d}{\sqrt{\sigma_d^2 + \epsilon}} + \beta \right) \qquad (19)
$$

$$
y^c(\mathbf{x}) = \sum_{r=1}^{R} \overline{f}_r(\mathbf{x}) y_r^c(\mathbf{x}) \qquad (20)
$$

The partial derivatives are:

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial m_{r,d}} &= \sum_{n=1}^{N_{bs}} \sum_{c=1}^{C} \frac{\partial \ell_1}{\partial y^c(\mathbf{x}_n)} \frac{\partial y^c(\mathbf{x}_n)}{\partial f_r(\mathbf{x}_n)} \frac{\partial f_r(\mathbf{x}_n)}{\partial \mu_{X_{r,d}}(x_{n,d})} \frac{\partial \mu_{X_{r,d}}(x_{n,d})}{\partial m_{r,d}} \\
&= \sum_{n=1}^{N_{bs}} \sum_{c=1}^{C} \left[[q^c(\mathbf{x}_n) - I(y_n == c)] \frac{y_r^c(\mathbf{x}_n) \sum_{i=1}^{R} f_i(\mathbf{x}_n) - \sum_{i=1}^{R} f_i(\mathbf{x}_n) y_i^c(\mathbf{x}_n)}{\left[\sum_{i=1}^{R} f_i(\mathbf{x}_n)\right]^2} f_r(\mathbf{x}_n) \frac{x_{n,d} - m_{r,d}}{\sigma_{r,d}^2} \right] \\
&= \sum_{n=1}^{N_{bs}} \sum_{c=1}^{C} \left[[q^c(\mathbf{x}_n) - I(y_n == c)] \left(y_r^c(\mathbf{x}_n) - \sum_{i=1}^{R} \bar{f}_i(\mathbf{x}_n) y_i^c(\mathbf{x}_n) \right) \bar{f}_r(\mathbf{x}_n) \frac{x_{n,d} - m_{r,d}}{\sigma_{r,d}^2} \right]
\end{aligned} \tag{21}
$$

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \sigma_{r,d}} &= \sum_{n=1}^{N_{bs}} \sum_{c=1}^{C} \frac{\partial \ell_1}{\partial y^c(\mathbf{x}_n)} \frac{\partial y^c(\mathbf{x}_n)}{\partial f_r(\mathbf{x}_n)} \frac{\partial f_r(\mathbf{x}_n)}{\partial \mu_{X_{r,d}}(x_{n,d})} \frac{\partial \mu_{X_{r,d}}(x_{n,d})}{\partial \sigma_{r,d}} \\
&= \sum_{n=1}^{N_{bs}} \sum_{c=1}^{C} \left[[q^c(\mathbf{x}_n) - I(y_n == c)] \frac{y_r^c(\mathbf{x}_n) \sum_{i=1}^{R} f_i(\mathbf{x}_n) - \sum_{i=1}^{R} f_i(\mathbf{x}_n) y_i^c(\mathbf{x}_n)}{\left[\sum_{i=1}^{R} f_i(\mathbf{x}_n)\right]^2} f_r(\mathbf{x}_n) \frac{(x_{n,d} - m_{r,d})^2}{\sigma_{r,d}^3} \right] \\
&= \sum_{n=1}^{N_{bs}} \sum_{c=1}^{C} \left[[q^c(\mathbf{x}_n) - I(y_n == c)] \left(y_r^c(\mathbf{x}_n) - \sum_{i=1}^{R} \bar{f}_i(\mathbf{x}_n) y_i^c(\mathbf{x}_n) \right) \bar{f}_r(\mathbf{x}_n) \frac{(x_{n,d} - m_{r,d})^2}{\sigma_{r,d}^3} \right]
\end{aligned} \tag{22}
$$

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w_{r,d}^c} &= \sum_{n=1}^{N_{bs}} \frac{\partial \ell_1}{\partial y^c(\mathbf{x}_n)} \frac{\partial y^c(\mathbf{x}_n)}{\partial y_r^c(\mathbf{x}_n)} \frac{\partial y_r^c(\mathbf{x}_n)}{\partial w_{r,d}^c} + \eta \frac{\partial \ell_2}{\partial w_{r,d}^c} \\
&= \sum_{n=1}^{N_{bs}} [q^c(\mathbf{x}_n) - I(y_n == c)] \bar{f}_r(\mathbf{x}_n) \cdot \left(\frac{\gamma(x_{n,d} - \bar{m}_d)}{\sqrt{\sigma_d^2 + \epsilon}} + \beta \right) + 2\eta w_{r,d}^c \cdot I(d > 0)
\end{aligned} \tag{23}
$$

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \gamma} &= \sum_{n=1}^{N_{bs}} \sum_{c=1}^{C} \sum_{r=1}^{R} \frac{\partial \ell_1}{\partial y^c(\mathbf{x}_n)} \frac{\partial y^c(\mathbf{x}_n)}{\partial y_r^c(\mathbf{x}_n)} \frac{\partial y_r^c(\mathbf{x}_n)}{\partial \gamma} \\
&= \sum_{n=1}^{N_{bs}} \sum_{c=1}^{C} \sum_{r=1}^{R} [q^c(\mathbf{x}_n) - I(y_n == c)] \bar{f}_r(\mathbf{x}_n) \cdot \sum_{d=1}^{D} \frac{w_{r,d}^c (x_{n,d} - \bar{m}_d)}{\sqrt{\sigma_d^2 + \epsilon}}
\end{aligned} \tag{24}
$$

$$
\frac{\partial \mathcal{L}}{\partial \beta} = \sum_{n=1}^{N_{bs}} \sum_{c=1}^{C} \sum_{r=1}^{R} \frac{\partial \ell_1}{\partial y^c(\mathbf{x}_n)} \frac{\partial y^c(\mathbf{x}_n)}{\partial y_r^c(\mathbf{x}_n)} \frac{\partial y_r^c(\mathbf{x}_n)}{\partial \beta} = \sum_{n=1}^{N_{bs}} \sum_{c=1}^{C} \sum_{r=1}^{R} [q^c(\mathbf{x}_n) - I(y_n == c)] \bar{f}_r(\mathbf{x}_n) \cdot \sum_{d=1}^{D} w_{r,d}^c \tag{25}
$$

At the testing phase, the BN operation can be merged into the consequent layer. Assume that after training, we obtain a BN layer with learned $\bar{\mathbf{m}} = (\bar{m}_1, ..., \bar{m}_D)^T$, $\sigma = (\sigma_1, ..., \sigma_D)^T$, γ and β. Then, the output y_r of the r-th rule with BN is:

$$
y_r(BN(\mathbf{x}_n)) = w_{r,0} + \gamma \sum_{d=1}^{D} w_{r,d} \frac{x_{n,d} - \bar{m}_d}{\sqrt{\sigma_d^2 + \epsilon}} + \beta D, \tag{26}
$$

which can be re-written as:

$$
y_r(BN(\mathbf{x}_n)) = b'_{r,0} + \sum_{d=1}^{D} b'_{r,d} x_{n,d}, \tag{27}
$$

where

$$
b'_{r,0} = w_{r,0} + \beta D - \gamma \sum_{d=1}^{D} \frac{\bar{m}_d w_{r,d}}{\sqrt{\sigma_d^2 + \epsilon}}, \tag{28}
$$

$$
b'_{r,d} = \gamma \frac{w_{r,d}}{\sqrt{\sigma_d^2 + \epsilon}}. \tag{29}
$$

By doing this, the original architecture of the TSK fuzzy classifier is kept unchanged.