Appendix A
Backpropagation for Non-decomposable Objectives

It worth to mention that the original backpropagation [73] is applicable to objective function, which value on the whole training dataset (or mini-batch) $\mathbb{T}$ can be approximated by the averaged value on every single sample. This is possible due to the stochastic approximation theory [74]

$$E(\mathbb{T}) = \frac{1}{N} \sum_{(\mathbf{x},\mathbf{y})\in\mathbb{T}} E(\mathbf{x},\mathbf{y}), \tag{23}$$

where E is the objective function, pair $(\mathbf{x},\mathbf{y})$ is a sample and its label. These objective functions (23) are known as the *decomposable* [75]. In machine learning we usually apply indirect optimization, regarding the objective function. Technically we try to improve a particular performance measure P (for some practical application), which is calculated on the test dataset. Whereas, during training stage, we optimize model parameters using different objective function E in the hope that in general it will improve performance P. This is indirect optimization, since such metrics are intractable or also known as *non-decomposable functions*. Another ambiguity is that we often optimize the same objective function (MSE, cross-entropy, KL-divergence, e.t.c.) for a different applications and for different performance metrics.

Definition. 1 ***Non-decomposable*** *objective function is not decomposed into expectations over individual examples [76]. For example, these functions are F1-score, Precision/Recall Break Even Point (PRBEP), Precision at k (Prec@k), ROCArea, EER, e.t.c.*

For a *non-decomposable* objective functions, parameters updating rule of backpropagation algorithm is assumed to be the same as for *decomposable* functions, except that *error signal* δ^n is calculated on the whole mini-batch $\mathbb{T}$

$$\mathbf{W}_n \leftarrow \mathbf{W}_n - \eta \frac{\partial E(\mathbb{T})}{\partial \mathbf{W}_n} = \mathbf{W}_n - \eta\delta^n(\mathbb{T}) \frac{\partial \mathbf{z}^n}{\partial \mathbf{W}_n}, \text{ where} \tag{24}$$
$$\delta^n(\mathbb{T}) = \frac{\partial E(\mathbb{T})}{\partial \mathbf{z}^n},$$

here $\mathbf{W}_n$ are neural network parameters for a range of $n = \overline{0, L}$ number of layers; E is a non-decomposable objective function; $\mathbf{z}^n$ is a *pre-activation* output or output of n-th layer of network before applying activation function; η is a learning rate. In the *decomposable* objective function case (23), we can represent the gradient of the function as the averaged gradient calculated per-sample [74]. Whereas, for *non-decomposable* functions in (24), it is similar to the original backpropagation with SGD [73] calculated on a *single sample*, here as a single sample we treat the whole mini-batch $\mathbb{T}$. It is very rough assumption, though it works in practice with *adaptive learning rate methods*.

A. Backpropagation for MFoM-micro-F1

The micro-F1 objective function for multiclass case of M classes in terms of discrete counts *true positive* (TP), *false positive* (FP) and *false negative* (FN)

$$\mathrm{F}_1 = \frac{2\mathrm{TP}}{\mathrm{FP} + 2\mathrm{TP} + \mathrm{FN}} = \frac{2\sum\limits_{\mathrm{k}=1}^{\mathrm{M}} \mathrm{TP}_\mathrm{k}}{\sum\limits_{\mathrm{k}=1}^{\mathrm{M}} \mathrm{FP}_\mathrm{k} + 2\sum\limits_{\mathrm{k}=1}^{\mathrm{M}} \mathrm{TP}_\mathrm{k} + \sum\limits_{\mathrm{k}=1}^{\mathrm{M}} \mathrm{FN}_\mathrm{k}}. \tag{25}$$

The problem with micro-F1 is that it is the discrete function, thus we can not directly differentiate and optimize it. After introducing the MFoM framework: the terms of *discriminative functions* (2), *misclassification measure* (4) and *smooth error function* (6), - we are ready to represent micro-F1 as the smooth continues function $\widehat{F}_1$ and then optimize it. We use *smooth error function* (6) in order to approximate the discrete counts on a mini-batch $\mathbb{T}$

$$\mathrm{TP} \approx \widehat{\mathrm{TP}} \triangleq \sum_{\mathrm{k}=1}^{\mathrm{M}} \widehat{\mathrm{TP}}_\mathrm{k} = \sum_{\mathrm{k}=1}^{\mathrm{M}} \sum_{\mathbf{x}\in\mathbb{T}} (1 - \mathrm{l}_\mathrm{k}(\mathbf{z})) \cdot \mathrm{y}_\mathrm{k}, \tag{26}$$

$$\mathrm{FP} \approx \widehat{\mathrm{FP}} \triangleq \sum_{\mathrm{k}=1}^{\mathrm{M}} \widehat{\mathrm{FP}}_\mathrm{k} = \sum_{\mathrm{k}=1}^{\mathrm{M}} \sum_{\mathbf{x}\in\mathbb{T}} (1 - \mathrm{l}_\mathrm{k}(\mathbf{z})) \cdot \bar{\mathrm{y}}_\mathrm{k}, \tag{27}$$

$$\mathrm{FN} \approx \widehat{\mathrm{FN}} \triangleq \sum_{\mathrm{k}=1}^{\mathrm{M}} \widehat{\mathrm{FN}}_\mathrm{k} = \sum_{\mathrm{k}=1}^{\mathrm{M}} \sum_{\mathbf{x}\in\mathbb{T}} \mathrm{l}_\mathrm{k}(\mathbf{z}) \cdot \mathrm{y}_\mathrm{k}, \tag{28}$$

where y_k and $\bar{y}_k$ are labels of the corresponding sample $\mathbf{x}$ or

$$y_k = \mathbf{1}\left(x \in C_k\right) = \begin{cases} 1, & \text{if } x \in C_k, \\ 0, & \text{if } x \notin C_k. \end{cases} \tag{29}$$

and

$$\bar{y}_k = \mathbf{1}\left(x \notin C_k\right) = \begin{cases} 0, & \text{if } x \in C_k, \\ 1, & \text{if } x \notin C_k. \end{cases} \tag{30}$$

where C_k is the set of samples where k^{th} label is "on" (label k equals 1), indicator functions are $\mathbf{1}\left(\cdot\right)$. We denote the sum of $\widehat{\mathrm{TP}}$ and $\widehat{\mathrm{FN}}$ as

$$\begin{aligned} |C| &\triangleq \sum_{k=1}^{M} |C_k| = \sum_{k=1}^{M} \widehat{\mathrm{TP}}_k + \widehat{\mathrm{FN}}_k = \\ &= \sum_{k=1}^{M} \sum_{\mathbf{x} \in \mathbb{T}} \left(1 - l_k\left(\mathbf{z}\right)\right) \cdot y_k + l_k\left(\mathbf{z}\right) \cdot y_k = \sum_{k=1}^{M} \sum_{\mathbf{x} \in \mathbb{T}} y_k, \end{aligned} \tag{31}$$

Therefore, the term $|C|$ is the count of unit "1" labels of samples $\mathbf{x}$ across all mini-batch $\mathbb{T}$, i.e. the number of positive labels. The $|C|$ is a constant value for the current mini-batch $\mathbb{T}$, because we calculate it one time for the whole $\mathbb{T}$.

Finally, we get the approximation of discrete micro-F1 from (25), we call it *MFoM-micro-F1* objective function

$$\mathrm{F}_1 \approx \widehat{\mathrm{F}}_1 = \frac{2\widehat{\mathrm{TP}}}{\widehat{\mathrm{FP}} + \widehat{\mathrm{TP}} + |\mathrm{C}|} \tag{32}$$

Our task is to minimize the objective function

$$E = 1 - \widehat{\mathrm{F}}_1 \rightarrow \underset{\mathbb{W}}{\arg\min}, \tag{33}$$

where a parameter set $\mathbb{W} = \left\{ \mathbf{W}_n | \, n = \overline{0, L} \right\}$ of the network consisting of $L+1$ layers. We are able to calculate the partial derivatives of all network parameters $\mathbb{W}$, if we find the network *error signal* term $\delta^n\left(\mathbb{T}\right)$, starting from the output layer $n = L$

$$\begin{aligned} \delta^L(\mathbb{T}) = \frac{\partial E}{\partial \mathbf{z}} = -\frac{\partial \widehat{\mathrm{F}}_1}{\partial \mathbf{z}} = -\left[\frac{2\widehat{\mathrm{TP}}}{\widehat{\mathrm{FP}} + \widehat{\mathrm{TP}} + |C|} \right]'_{\mathbf{z}} = \\ = -\frac{2\widehat{\mathrm{TP}}' \cdot \left(\widehat{\mathrm{FP}} + \widehat{\mathrm{TP}} + |C|\right) - 2\widehat{\mathrm{TP}} \cdot \left(\widehat{\mathrm{FP}}' + \widehat{\mathrm{TP}}'\right)}{\left(\widehat{\mathrm{FP}} + \widehat{\mathrm{TP}} + |C|\right)^2} = \\ = -\frac{2\widehat{\mathrm{TP}}' \cdot \widehat{\mathrm{FP}} + 2\widehat{\mathrm{TP}}' \cdot |C| - 2\widehat{\mathrm{TP}} \cdot \widehat{\mathrm{FP}}'}{\left(\widehat{\mathrm{FP}} + \widehat{\mathrm{TP}} + |C|\right)^2} = \\ = -\frac{-2\widehat{\mathrm{FN}}' \cdot \widehat{\mathrm{FP}} - 2\widehat{\mathrm{FN}}' \cdot |C| - 2\widehat{\mathrm{TP}} \cdot \widehat{\mathrm{FP}}'}{\left(\widehat{\mathrm{FP}} + \widehat{\mathrm{TP}} + |C|\right)^2} = \\ = \frac{2 \cdot \left[\left(|C| + \widehat{\mathrm{FP}}\right) \cdot \widehat{\mathrm{FN}}' + \widehat{\mathrm{TP}} \cdot \widehat{\mathrm{FP}}' \right]}{\left(\widehat{\mathrm{FP}} + \widehat{\mathrm{TP}} + |C|\right)^2}. \end{aligned}$$

Thus,

$$\delta^L(\mathbb{T}) = \frac{\partial E}{\partial \mathbf{z}} = A \left(w_1 \frac{\partial \widehat{\mathrm{FN}}}{\partial \mathbf{z}} + w_2 \frac{\partial \widehat{\mathrm{FP}}}{\partial \mathbf{z}} \right), \tag{34}$$

where constants, which are calculated on the whole current mini-batch $\mathbb{T}$

$$A = \frac{2}{\left(\widehat{\mathrm{FP}} + \widehat{\mathrm{TP}} + |C|\right)^2}, \quad w_1 = |C| + \widehat{\mathrm{FP}}, \quad w_2 = \widehat{\mathrm{TP}}. \tag{35}$$

If we plug in the $\widehat{\mathrm{FP}}$ and $\widehat{\mathrm{FN}}$ from (27) and (28) to the δ^L in eq. (34)

$$\begin{aligned} \delta^L(\mathbb{T}) = A \left(w_1 \tfrac{\partial \widehat{\mathrm{FN}}}{\partial \mathbf{z}} + w_2 \tfrac{\partial \widehat{\mathrm{FP}}}{\partial \mathbf{z}} \right) = \\ = A \left(w_1 \cdot \sum_{k=1}^{M} \sum_{\mathbf{x} \in \mathbb{T}} \tfrac{\partial l_k(\mathbf{z})}{\partial \mathbf{z}} \cdot y_k - w_2 \cdot \sum_{k=1}^{M} \sum_{\mathbf{x} \in \mathbb{T}} \tfrac{\partial l_k(\mathbf{z})}{\partial \mathbf{z}} \cdot \bar{y}_k \right) = \end{aligned}$$

$$= A\left(\sum_{\mathbf{x}\in\mathbb{T}}\sum_{k=1}^{M}\frac{\partial l_k(\mathbf{z})}{\partial\mathbf{z}}\left[w_1\cdot y_k - w_2\cdot\bar{y}_k\right]\right),$$

It means that if k^{th} label of a sample $\mathbf{x}$ is unit (i.e., $y_k = 1$ and $\bar{y}_k = 0$), then we multiply by the weight constant w_1, otherwise by the constant $-w_2$. Then we rewrite in the vector form the last result

$$\delta^L(\mathbb{T}) = A\sum_{\mathbf{x}\in\mathbb{T}}\left(\frac{\partial l_1(\mathbf{z})}{\partial\mathbf{z}},\ldots,\frac{\partial l_k(\mathbf{z})}{\partial\mathbf{z}},\ldots,\frac{\partial l_M(\mathbf{z})}{\partial\mathbf{z}}\right)\cdot\begin{pmatrix} w_1\cdot y_1 - w_2\cdot\bar{y}_1\\ \cdots\\ w_1\cdot y_k - w_2\cdot\bar{y}_k\\ \cdots\\ w_1\cdot y_M - w_2\cdot\bar{y}_M\end{pmatrix}. \tag{36}$$

In (36), the left vector under the sum is the transposed Jacobian, because we find the partial derivatives with respect to the vector of multiple variables $\mathbf{z} = (z_1,\ldots,z_m,\ldots,z_M)^\top$

$$\begin{aligned} J_{\mathbf{z}}^\top\mathcal{L} &= \left(\frac{\partial l_1(\mathbf{z})}{\partial\mathbf{z}},\ldots,\frac{\partial l_k(\mathbf{z})}{\partial\mathbf{z}},\ldots,\frac{\partial l_M(\mathbf{z})}{\partial\mathbf{z}}\right) = \\ &= \begin{pmatrix} \frac{\partial l_1(\mathbf{z})}{\partial z_1} & \cdots & \frac{\partial l_M(\mathbf{z})}{\partial z_1}\\ \cdots & \frac{\partial l_k(\mathbf{z})}{\partial z_m} & \cdots\\ \frac{\partial l_1(\mathbf{z})}{\partial z_M} & \cdots & \frac{\partial l_M(\mathbf{z})}{\partial z_M}\end{pmatrix}, \end{aligned} \tag{37}$$

where $m = \overline{1,M}$ is the row index of the variables z_m and $k = \overline{1,M}$ is the column index for the functions $l_k(\mathbf{z})$.

$$\delta^L(\mathbb{T}) = A\cdot\sum_{(\mathbf{x},\mathbf{y})\in\mathbb{T}} J_{\mathbf{z}}^\top\mathcal{L}(\mathbf{x})\cdot(w_1\cdot\mathbf{y} - w_2\cdot\bar{\mathbf{y}}). \tag{38}$$

NOTE: we can interpret (38) as for every pair $(\mathbf{x},\mathbf{y})$ from mini-batch $\mathbb{T}$ we calculate the value of the transposed Jacobian and find the weighted linear combination of its columns with w_1 or $-w_2$. The weight constants w_1 or w_2 are defined by the labels y_k or $\bar{y}_k$ of sample. Then we sum up all weighted Jacobians across all samples in a mini-batch $\mathbb{T}$. Thus, we get *"decomposed"* error signal δ^L per each sample $\mathbf{x}$, i.e. MFoM framework allowed us to archive it.

B. Jacobian for the Units-vs-Zeros Misclassification Measure

In this section, we infer a Jacobian for the *units-vs-zeros* misclassification measure (4). Units-vs-zeros misclassification measure was adopted for multi-label classification and proposed in [32]. First, we consider the example A.1 of the *units-vs-zeros* misclassification measure with sample **x** having multiple label vector **y**.

Example A.1 *Let we have training pair* $(\boldsymbol{x},\boldsymbol{y})$ *and* $\boldsymbol{y} = (1,1,0,0)^\top$*, see Fig. 8. Then, we have units-vs-zeros misclassification as*

$$\begin{aligned}\psi_1(\mathbf{z}) &= -g_1 + \ln\left[\tfrac{1}{2}\left(e^{g_3}+e^{g_4}\right)\right],\\ \psi_2(\mathbf{z}) &= -g_2 + \ln\left[\tfrac{1}{2}\left(e^{g_3}+e^{g_4}\right)\right],\\ \psi_3(\mathbf{z}) &= -g_3 + \ln\left[\tfrac{1}{2}\left(e^{g_1}+e^{g_2}\right)\right],\\ \psi_4(\mathbf{z}) &= -g_4 + \ln\left[\tfrac{1}{2}\left(e^{g_1}+e^{g_2}\right)\right],\end{aligned}$$

where g_k *is the output score of a neural network.*

The *units-vs-zeros* misclassification measure (4) can be rewritten in the convenient form for inference of derivatives

$$\psi_k(\mathbf{z}) = -g_k + \bar{y}_k\cdot u + y_k\cdot v, \tag{39}$$

where Kolmogorov *f-mean* of *unit models*

$$u = \ln\left[\frac{1}{\sum_{i=1}^{M} y_i}\cdot\langle\mathbf{y},\exp(\mathbf{g})\rangle\right] = \ln\left[\frac{1}{\sum_{i=1}^{M} y_i}\cdot\left(y_1e^{g_1}+\ldots+y_ke^{g_k}+\ldots+y_Me^{g_M}\right)\right], \tag{40}$$

and Kolmogorov *f-mean* of *zero models*

$$v = \ln\left[\frac{1}{\sum_{i=1}^{M} \bar{y}_i}\cdot\langle\bar{\mathbf{y}},\exp(\mathbf{g})\rangle\right] = \ln\left[\frac{1}{\sum_{i=1}^{M} \bar{y}_i}\cdot\left(\bar{y}_1e^{g_1}+\ldots+\bar{y}_ke^{g_k}+\ldots+\bar{y}_Me^{g_M}\right)\right], \tag{41}$$

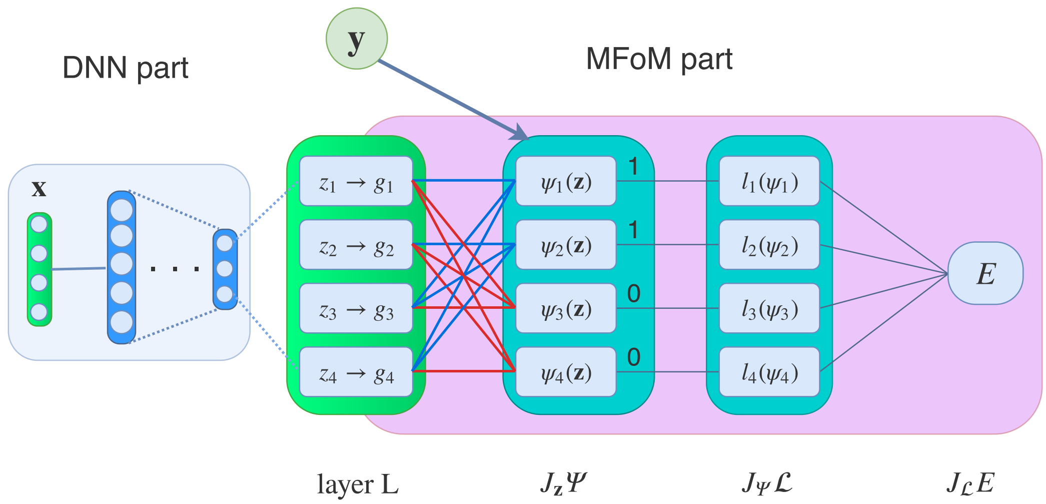

Fig. 8. Example of the extended DNN with the MFoM objective function and *units-vs-zeros* misclassification measure: Ψ is the vector of misclassification measure, $\mathcal{L}$ is the vector of smooth error count, E is the smoothed MFoM-based objective; $J_{\mathcal{L}}E$, $J_{\Psi}\mathcal{L}$, $J_{\mathbf{z}}\Psi$ are Jacobians.

where binary vector of labels is $\mathbf{y}$ and its inverse is $\overline{\mathbf{y}}$. Thus, if the current sample $\mathbf{x}$ is labeled as 1 in the ground-truth for the class C_k (i.e. $y_k = 1$ or $\overline{y}_k = 0$), the competing models will be considered only those with labels 0, and we find an average of these competing zero models, i.e. it is the equation (41). The misclassification measure for that sample is calculated as

$$\psi_k(\mathbf{z}) = -g_k + y_k \cdot v. \tag{42}$$

Otherwise, if the sample $\mathbf{x}$ is labeled as 0 in the ground-truth (i.e. $y_k = 0$ or $\overline{y}_k = 1$) for the class C_k, the competing models are with labels 1, and we find average of these, i.e. the equation (40) and

$$\psi_k(\mathbf{z}) = -g_k + \overline{y}_k \cdot u. \tag{43}$$

We denote Jacobian matrix for *units-vs-zeros* misclassification measure as $J_{\mathbf{z}}\Psi$. Further, we find the partial derivatives $\frac{\partial \psi_k(\mathbf{z})}{\partial z_m}$ of the Jacobian $J_{\mathbf{z}}\Psi$, we have two cases:

a) if $m = k$, diagonal elements of the Jacobian

$$\frac{\partial \psi_k}{\partial z_k} = -g'_k + \bar{y}_k \frac{\partial u}{\partial z_k} + y_k \frac{\partial v}{\partial z_k},$$

$$\frac{\partial u}{\partial z_k} = \frac{1}{\langle \mathbf{y}, \exp(\mathbf{g})\rangle} \cdot y_k g'_k \exp(g_k),$$

$$\frac{\partial v}{\partial z_k} = \frac{1}{\langle \overline{\mathbf{y}}, \exp(\mathbf{g})\rangle} \cdot \overline{y}_k g'_k \exp(g_k).$$

then we get

$$\frac{\partial \psi_k}{\partial z_k} = -g'_k, \tag{44}$$

because $y_k \overline{y}_k = 0$.

b) if $m \neq k$, off-diagonal elements

$$\frac{\partial \psi_k}{\partial z_m} = \bar{y}_k \frac{\partial u}{\partial z_m} + y_k \frac{\partial v}{\partial z_m} = \frac{1}{\langle \mathbf{y}, \exp(\mathbf{g})\rangle} \cdot \bar{y}_k y_m g'_m \exp(g_m) + \frac{1}{\langle \overline{\mathbf{y}}, \exp(\mathbf{g})\rangle} y_k \overline{y}_m g'_m \exp(g_m) =$$

$$= g'_m \exp(g_m) \left[\frac{\bar{y}_k y_m}{\langle \mathbf{y}, \exp(\mathbf{g})\rangle} + \frac{y_k \bar{y}_m}{\langle \overline{\mathbf{y}}, \exp(\mathbf{g})\rangle} \right]$$

Then, we get the Jacobian matrix

$$J_{\mathbf{z}}^{\top}\Psi = \begin{pmatrix} -g'_1 & \cdots & g'_1 \exp(g_1)\left[\frac{\bar{y}_k y_1}{P} + \frac{y_k \bar{y}_1}{B}\right] & \cdots & g'_1 \exp(g_1)\left[\frac{\bar{y}_M y_1}{P} + \frac{y_M \bar{y}_1}{B}\right] \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ g'_m \exp(g_m)\left[\frac{\bar{y}_1 y_m}{P} + \frac{y_1 \bar{y}_m}{B}\right] & \cdots & -g'_k & \cdots & g'_m \exp(g_m)\left[\frac{\bar{y}_M y_m}{P} + \frac{y_M \bar{y}_m}{B}\right] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g'_M \exp(g_M)\left[\frac{\bar{y}_1 y_M}{P} + \frac{y_1 \bar{y}_M}{B}\right] & \cdots & g'_M \exp(g_M)\left[\frac{\bar{y}_k y_M}{P} + \frac{y_k \bar{y}_M}{B}\right] & \cdots & -g'_M \end{pmatrix},$$

where $P = \langle \mathbf{y}, \exp(\mathbf{g}) \rangle$ and $B = \langle \bar{\mathbf{y}}, \exp(\mathbf{g}) \rangle$. Extracting g'_k, we get

$$J_{\mathbf{z}}^\top \Psi = \mathrm{diag}\left(g'_m\right) \cdot \begin{pmatrix} -1 & \cdots & \exp(g_1)\left[\frac{\bar{y}_k y_1}{P} + \frac{y_k \bar{y}_1}{B}\right] & \cdots & \exp(g_1)\left[\frac{\bar{y}_M y_1}{P} + \frac{y_M \bar{y}_1}{B}\right] \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \exp(g_m)\left[\frac{\bar{y}_1 y_m}{P} + \frac{y_1 \bar{y}_m}{B}\right] & \cdots & -1 & \cdots & \exp(g_m)\left[\frac{\bar{y}_M y_m}{P} + \frac{y_M \bar{y}_m}{B}\right] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \exp(g_M)\left[\frac{\bar{y}_1 y_M}{P} + \frac{y_1 \bar{y}_M}{B}\right] & \cdots & \exp(g_M)\left[\frac{\bar{y}_k y_M}{P} + \frac{y_k \bar{y}_M}{B}\right] & \cdots & -1 \end{pmatrix} =$$
$$= \mathrm{diag}\left(g'_m\right) \cdot \mathbf{G}$$

Thus, backpropagation network *error signal*, using *units-vs-zeros* misclassification measure, is

$$\delta^L(\mathbb{T}) = A \sum_{(\mathbf{x},\mathbf{y}) \in \mathbb{T}} J_{\mathbf{z}}^\top E(\mathbf{x}) = A \sum_{(\mathbf{x},\mathbf{y}) \in \mathbb{T}} J_{\mathbf{z}}^\top \Psi \cdot J_{\Psi}^\top \mathcal{L} \cdot J_{\mathcal{L}}^\top E, \tag{45}$$

where

$$\begin{array}{l} J_{\mathbf{z}}^\top \Psi = \mathrm{diag}\left(g'_m\right) \cdot \mathbf{G} \\ J_{\Psi}^\top \mathcal{L} = \mathrm{diag}\left(l'_k\right) \\ J_{\mathcal{L}}^\top E = \left(w_1 \mathbf{y} - w_2 \bar{\mathbf{y}}\right) \end{array} \tag{46}$$