Consider some data $\{(x_i, y_i)\}_{i=1}^n$ and a differentiable loss function $\mathcal{L}(y, F(x))$ and a multiclass classification problem which should be solved by a gradient boosting model. $F(x)$ is a model to predict $\hat{y}$. A loss function is an evaluation metric of a model $F(x)$ and our target variable y. The softmax transfer function is typically used to compute the estimated probability distribution in classification tasks involving multiple classes. Let therefore be the cross-entropy loss function defined by the estimated probability distributions gained from the softmax function so that :

$$\hat{y}_{i,softmax} = \frac{e^{\hat{y}}}{\sum_{k=1}^{N} e^{\hat{y}_k}} \tag{1}$$

$$\mathcal{L}(y_i, F(x)) = -\sum y_i \log \hat{y}_{i,softmax} \tag{2}$$

where y_i defines the the relative frequencies of each class in our target variable y and $haty_i$ defines the predictions of y_i.
We need to initialize our gradient boosting model with an constant value. Let the initial model $F_0(x)$ be defined as :

$$F_0(x) = \frac{1}{N} \sum_{i=1}^{T_j} y_{i,j} \tag{3}$$

where T defines the number of observations in class j and N defines the number of observations of y. So the inital model is the class probability for each class j. Given that $F(x)$ is a model to predict $\hat{y}$ the values of the model $F(x)$ can be used to define $\hat{y}_i$. So that we can calculate $\hat{y}_{i,softmax}$ with 1

Now we show that the loss function is differentiable.

$$D_j \hat{y}_{i,softmax} = \frac{\partial \hat{y}_{i,softmax}}{\partial \hat{y}_j} = \begin{bmatrix} D_1 \hat{y}_{1,softmax} & \times & D_N \hat{y}_{1,softmax} \\ \vdots & \ddots & \vdots \\ D_1 \hat{y}_{N,softmax} & \times & D_N \hat{y}_{N,softmax} \end{bmatrix} \tag{4}$$

$$D_j \hat{y}_{i,softmax} = \frac{\partial \hat{y}_{i,softmax}}{\partial \hat{y}_j} \left\{ \begin{array}{ll} \hat{y}_{i,softmax} - \hat{y}_{j,softmax}^2 & i = j \\ -\hat{y}_{j,softmax} \times \hat{y}_{i,softmax} & i \neq j \end{array} \right\} \tag{5}$$

And the derivative of the cross-entropy loss function w.r.t. $\hat{y}_i$

$$\frac{\partial \mathcal{L}}{\partial \hat{y}_{i,softmax}} = \left(-\sum y_i \log \hat{y}_{i,softmax}\right) = -\sum \frac{y_i}{\hat{y}_{i,softmax}} \tag{6}$$

Combining both gradients leads to the gradient of the loss function

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \hat{y}_{i,softmax}} &= -\frac{y_i}{\hat{y}_{i,softmax}} \hat{y}_{i,softmax} \left(1 - \hat{y}_{i,softmax}\right) + \sum_{j \neq i} -\frac{y_j}{\hat{y}_j} \left(-\hat{y}_{j,softmax} \hat{y}_{i,softmax}\right) \\
&= -y_i + y_i \hat{y}_{i,softmax} + \sum_{j \neq i} y_j \hat{y}_{i,softmax} \\
&= -y_i + \sum_j y_j \hat{y}_{i,softmax} \\
&= \hat{y}_{i,softmax} \underbrace{\sum_j y_j}_{=1} - y_i \\
&= \hat{y}_{i,softmax} - y_i
\end{aligned}
\tag{7}
$$

where y_j is the probability of class j

One obtains the intial residuals $r_{i0} = y_i - F_0(x)$ which are then used to fit a classification tree with R terminal nodes.

The pseudo residuals are obtained through

$$
\begin{aligned}
r_{im} &= -\left[\frac{\partial \mathcal{L}\left(y_i, F\left(x_i\right)\right)}{\partial \hat{y}_{i,softmax}}\right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \ldots, n \\
&= -\sum_{i=1}^{N} (\hat{y}_{i,softmax} - y_i) \\
&= \sum_{i=1}^{N} (y_i - \hat{y}_{i,softmax})
\end{aligned}
\tag{8}
$$

The output values for each terminal node l of each tree m γ_{lm} can be derived using a second-order Taylor approximation so that,

$$
\begin{aligned}
\gamma_{lm} &= \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{lm}} L\left(y_i, F_{m-1}\left(x_i\right) + \gamma\right) \\
&\approx \frac{\partial \mathcal{L}\left(y_i, F\left(x_i\right)\right) + \frac{\partial \mathcal{L}\left(y_i, F\left(x_i\right)\right)}{\partial \hat{y}_{i,softmax}} \gamma + \frac{1}{2} \frac{\partial \mathcal{L}\left(y_i, F\left(x_i\right)\right)}{\partial^2 \hat{y}_{i,softmax}} \gamma^2}{\partial \gamma} \\
&= \hat{y}_{i,softmax} - y_i + \gamma \overset{!}{=} 0 \\
\gamma &= y_i - \hat{y}_{i,softmax}
\end{aligned}
\tag{9}
$$

In the end our updated model should be :

$$
F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{m} \gamma_{jm} I\left(x \in R_{jm}\right) \tag{10}
$$

where ν is a learning rate to weight the output values of the tree m.