

Weekly Report

Riku Gondow

November 8, 2022

1 Progress

- Check the accuracy of Triple Joint Loss[1]
- Read "Proportional Similarity-Based Openmax Classifier for Open Set Recognition in SAR Images"[2]

2 Result

2.1 Softmax + Center Loss

The equations for Softmax Loss and Center Loss are shown below. Softmax Loss increases the inter-class variance, while Center Loss enhances the intra-class aggregation. The sum of Losses uses the hyperparameter α to balance the two Losses.

$$L_{\text{Softmax}} = \frac{1}{N} \sum_{i=1}^N 1 - \log \frac{e^{\|\mathbf{w}_{yi}\| * \|\mathbf{x}_i\| * \cos \theta_{yi} + \mathbf{b}_{yi}}}{\sum_{j=1}^C e^{\|\mathbf{w}_j\| * \|\mathbf{x}_i\| * \cos \theta_j + \mathbf{b}_j}} \quad (1)$$

$$L_{\text{Center}} = \frac{1}{2} \sum_{i=1}^N \left\| \mathbf{x}_i^j - \mathbf{c}^j \right\|_2, (j = 1, 2, \dots, C) \quad (2)$$

$$L_{\text{Total}} = L_{\text{Softmax}} + \alpha L_{\text{Center}}$$

In formula (1), C is the total number of categories in the training set. w_j, w_{yi}, b_j, b_{yi} is the weight matrixes and bias terms of the last fully connected layer of the network, θ_j is the angle between the weight vector and the feature vector in the high dimensional space.

In formula (2), N is the number of samples of the j th class, x_i^j represents the i th sample belonging to the j th class and c^j represents the class center of the j th class.

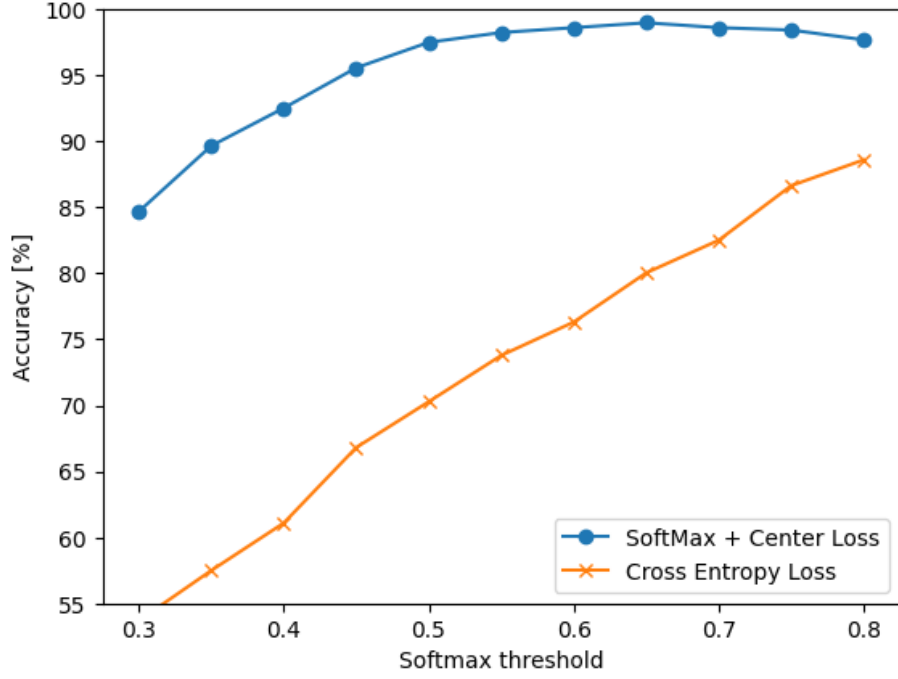


Figure 1: Comparison of using Cross Entropy and Softmax + Center for loss function

Here is a figure. The figure1 shows a comparison of using Cross Entropy and Softmax + Center for loss function. From Figure 1, we can see that higher accuracy is achieved for all threshold values when Softmax+Center is used than when Cross Entropy is used.

Table 1: Comparison with two baselines

Condition	Method	Known/Unknown	Accuracy
Open set	Yan et al.[3] (DDLm)	15/15	93.57%
	Xing et al.[4]	15/15	94.35%
	Softmax + Center Loss	15/15	98.98%

The table1 above shows the comparison between the two baselines. The table shows that the proposed method with Softmax and Center Loss outperforms the two methods.

2.2 Triple Joint Loss(Softmax + Center + Large Margin Cosine Loss)

The equations for Cosine Loss and Triple Joint Loss are shown below. Cosine loss enhances the intra-class aggregation and the inter-class dispersion.

$$L_{\text{Cosine}} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s(\cos \theta_{yi} - m) + \mathbf{b}_{yi}}}{e^{s(\cos \theta_{yi} - m) + \mathbf{b}_{yi}} + \sum_{j=1, j \neq y_i}^C e^{s \cos \theta_j + \mathbf{b}_j}} \quad (3)$$

$$L_{\text{Triple Joint}} = L_{\text{Softmax}} + \alpha L_{\text{Center}} + L_{\text{Cosine}}$$

We set $\|w_{yi}\| = 1, \|x_i\| = s$ in formula (1). It can be seen that after the introduction of s , the classification result is only affected by θ_{yi} , the angle between the weight vector and the feature vector. m is the cosine margin parameter that needs to be optimized according to the training samples. The introduction of m modifies the decision boundary of the neural network. Finally, we combine the Softmax loss, the Center loss, and the Cosine loss to form Triple Joint Loss.

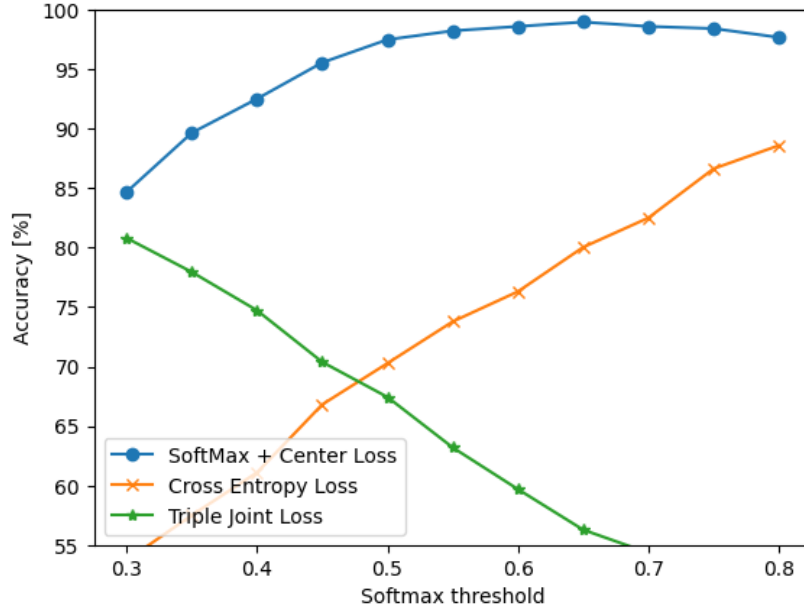


Figure 2: Comparison of using three different losses for loss function

Here is a figure. The figure2 shows a comparison of using three different losses for loss function. From the figure, we can see that Triple Joint Loss is less accurate than Softmax + Center Loss for all threshold values.

2.3 Discussion

For noisy datasets, the method combining Triple Joint Loss and Openmax-based method may have high robustness. This is because the 98.98% accuracy was achieved on a low-noisy dataset, where the known labels can be classified fairly accurately and the Softmax functions output for the known labels is almost 1.

Also, Triple Joint Loss still has room to improve its accuracy by adjusting hyperparameters such as α , s , and m .

3 Openmax vs Proportional Similarity Openmax

In this section, we explain the differences between Openmax and PS Openmax based on the paper[2]. OpenMax is executed in the flow of Algorithm 1 -; 2 (model calibration -; OpenMax score calculation). First, the model calibration is as follows:

Algorithm 1 Model Calibration

Input: $AV_1^{train}, \dots, AV_N^{train}, \eta$
Output: $(Weibull_1^{train}, MAV_1^{train}), \dots, (Weibull_N^{train}, MAV_N^{train})$

- 1: **for** $j = 1, 2, \dots, N$ **do**
- 2: $MAV_j^{train} = \text{mean}(AV_j^{train})$
- 3: $ED_j^{train} = \text{sort}(\text{EuclideanDistance}(AV_j^{train}, MAV_j^{train}))$
- 4: $Weibull_j^{train} = \text{FitHigh}(ED_j^{train}, \eta)$

In the SoftMax layer, the output vector of the last layer of fully connected layers is typically used. This vector is called the **Activation Vector**. AV_1^{train} (Activation Vector) represents the activation vector of the image (input) of the first class out of N classes. η is a hyperparameter that determines how many of the distance values are included in the OpenMax score calculation. The line2 computes the average activation vector for each Known N classes. This is performed using the FitHigg function and Weibull distribution of LibMR, which is a publicly available library, for maximum likelihood estimation. Finally, Algorithm 1 returns the Weibull model and the MAV computed on N Known classes.

Next, the algorithm2 is as follows:

Algorithm 2 Openmax scores calculation

Input: $(\text{Weibull}_1^{\text{train}}, \text{MAV}_1^{\text{train}}), \dots, (\text{Weibull}_N^{\text{train}}, \text{MAV}_N^{\text{train}})$, AV of the test image, N_α
Output: Openmax scores

- 1: $\text{ord} = \text{argsort}(\text{AV}, \text{"descending"})$
- 2: **for** $i = 1, \dots, N_\alpha$ **do**
- 3: $j = \text{ord}(i)$
- 4: $CD = \text{EuclideanDistance}(\text{AV} - \text{MAV}_j^{\text{train}})$
- 5: $\tau, \kappa, \lambda = \text{Weibull}_j^{\text{train}}$
- 6: $w = 1 - e^{-\left(\frac{\|CD - \tau\|}{\lambda}\right)^\kappa}$
- 7: $\alpha = (N_\alpha - i + 1)/N_\alpha$
- 8: $\text{modAV}(j) = \text{AV}(j)(1 - w \times \alpha)$
- 9: $\text{unk} = \sum_{j=1}^N (\text{AV}(j) - \text{modAV}(j))$
- 10: $\text{modAV}(N+1) = \text{unk}$
- 11: **for** $j = 1, 2, \dots, N+1$ **do**
- 12: $s_{\text{open}j} = \frac{e^{\text{modAV}(j)}}{\sum_{k=1}^{N+1} e^{\text{modAV}(k)}}$
- 13: $s_{\text{open}} = [s_{\text{open}1}, s_{\text{open}2}, \dots, s_{\text{open}N+1}]$

From lines 1 to 9, The Openmax subtracts a portion from each element of AV based on the similarity of the test image and the respective training class, sums the subtracted values, and forms a modified AV with one more element appended to its end to represent the unknown class. The N+1th class is then added as unknown at line 10, and the rest of the calculation is the same as SoftMax for the N+1 classes.

Then, the Proportional Similarity-based Openmax is as follows:

Algorithm 3 PS-Openmax scores calculation

Input: $(MAV_1^{train}, \dots, MAV_N^{train})$, AV of the test image
Output: PS-Openmax scores

- 1: **for** $i = 1, \dots, N$ **do**
- 2: $CD_i = \text{EuclideanDistance}(AV - MAV_i^{train})$
- 3: $CD_{Normalized} = \frac{CD}{\sum_{i=1}^N CD_i}$
- 4: $M = -\beta[\min(CD), \dots, \min(CD)]^T$
- 5: $AV_{SAF} = AV - \min(AV)$
- 6: $AV^* = [1 + M \cdot CD_{Normalized}] \circ AV_{SAF}$
- 7: $unk = \sum_{j=1}^N (AV_{SAF}(j) - AV^*(j))$
- 8: $ModifiedAV = [AV^*, unk]$
- 9: **for** $j = 1, 2, \dots, N + 1$ **do**
- 10: $s_{psj} = \frac{e^{ModifiedAV(j)}}{\sum_{k=1}^{N+1} e^{ModifiedAV(k)}}$
- 11: $s_{ps} = [s_{ps1}, s_{ps2}, \dots, s_{psN+1}]$

In algorithm3, the MAV (Mean Activation Vector) is computed in the same way as in OpenMax, forming the Euclidean channel distance vector (CD) from all classes in lines 1 and 2. In line 3, the normalized channel distance vector ($CD_{normalized}$) is computed, effectively avoiding the situation where the absolute value of the last element of the AV becomes too large; in OpenMax1, no interrelationship between the channel distance values is considered, and the channel distance values are used independently to modify the corresponding elements of AV. In line 4, $\min(CD)$ is repeated to form an $N \times 1$ vector, and $\beta \in (0, 1)$ as a hyperparameter is used to balance between the minimum channel distance and the two elements of the normalized channel distance vector. In line 5, a vector-type activation function AV_{SAF} (supplementary activation function) is computed to ensure that each element of AV is not negative.

The proposed method modifies AV_{SAF} on an element-by-element basis using two elements. They are the minimum channel distance M and the normalized channel distance vector $CD_{normalized}$ using the hadamard product (\circ); (see line 6 of Algorithm 3). By computing the sum of the subtraction between AV_{SAF} and AV^* , a new element is formed (see line 7 of Algorithm 3). Like the original Openmax, the new element is added to the end of the modified AV (see line 8 of Algorithm 3), and the modified AV is mapped to the probability region to

produce the PS-Openmax score (see lines 9-11 of Algorithm 3).

The PS (proportional similarity)-Openmax approach substitutes the tail fitting procedure with a proportional similarity measure and uses the relationship between the channel distance values to prevent new elements from always having too large negative or too large positive values. There is also no need to adjust hyperparameters such as η or N_α .

With 10-images open set (called MSTAR dataset, known:8, unknown:2), openmax achieves 90% accuracy and PS-Openmax 93% accuracy.

4 Next Plan

- Visualize the feature mapping of output when using Softmax and Center Loss to see the effect of the loss
- Implement Openmax and Proportional Similarity-based Openmax
- Check the noisy data acquired in our lab

References

- [1] Y. Yang, B. Huang and Z. Ni, "Open-set Person Identification with Triple-Joint Loss Based on Radar Gait Micro-Doppler Signatures," 2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP), 2022, pp. 1787-1791.
- [2] Giusti E, Ghio S, Oveis AH, Martorella M. Proportional Similarity-Based Openmax Classifier for Open Set Recognition in SAR Images. Remote Sensing. 2022; 14(18):4665.
- [3] Yan, Baiju, et al. "Heart signatures: Open-set person identification based on cardiac radar signals." Biomedical Signal Processing and Control 72 (2022): 103306.
- [4] Zelin Xing, Mondher Bouazizi and Tomoaki Ohtsuki. Deep Learning-based Open-set Person Identification using Radar Extracted Cardiac Signals.