**Class Project Report**

***Implementation of SoftAdapt in Deep Multiview Clustering by Contrasting Cluster Assignments***

**1. Introduction**

In the realm of neural network (NN) models and machine learning, when attempting to train a NN, the process will produce a loss value, which usually indicates how a model behaves after an iteration of optimization. There are a few ways of calculating this loss value when dealing with NN, the most relevant one involves the creation of a linear combination in order to provide an aggregate of the total loss of a system. For our assigned paper, the Cross-View Contrastive Learning (CVCL) model [2] used this kind of linear combination for determining the loss with the multiview data present in the NN training. However, since the process involves contrastive learning, certain views of a single data point will be a negative or positive match, the extent of which matters in classifying the data in question. Since classification is the end goal for the CVCL model, we want to ensure the classifications are as accurate as possible. Therefore, we sought a way to improve the accuracy of the CVCL model. As mentioned before, a single linear combination is used in determining the total loss. We sought a method to enhance the calculation of the loss by weighting the beneficial views of a particular data point higher than those that do not contribute to successful classification. This is achieved through a dynamic weight assignment based on the SoftAdapt scheme.

**2. Related Work**

Multiview Clustering was initially mostly based on simpler methods like graph,multiple-kernel, and co-training, but as the complexity grew the deep learning-based MVC methods became more popular.

Adaptive loss functions, as the name suggests, the total loss value is changed based on certain parameters. These are useful when the loss functions have different contribution percentages. Some of the most common challenges are model convergence difficulties and the need to manually adjust the loss function parameter settings.

Conventional optimization algorithms like gradient descent with fixed weights, might not work great with multiple loss functions.SoftAdapt is a new method that uses a dynamic approach to alter weights based on the model's previous iteration performance. The dynamic technique helps to overcome some of the limitations of static methods by focusing on model robustness and convergence speed.

**3. Methodology**

In this project, we have implemented a SoftAdapt algorithm which focuses on adjusting the weights of the Loss components in the loss function. This algorithm helps to improve the learning rate of the model as it is efficient and straightforward to implement.

—---(1)

------(2)

Where 𝝰i= Final Weight,  
 m= Number of Loss Components,  
 si= rate of change  
 fi = mean of resp. Loss Components

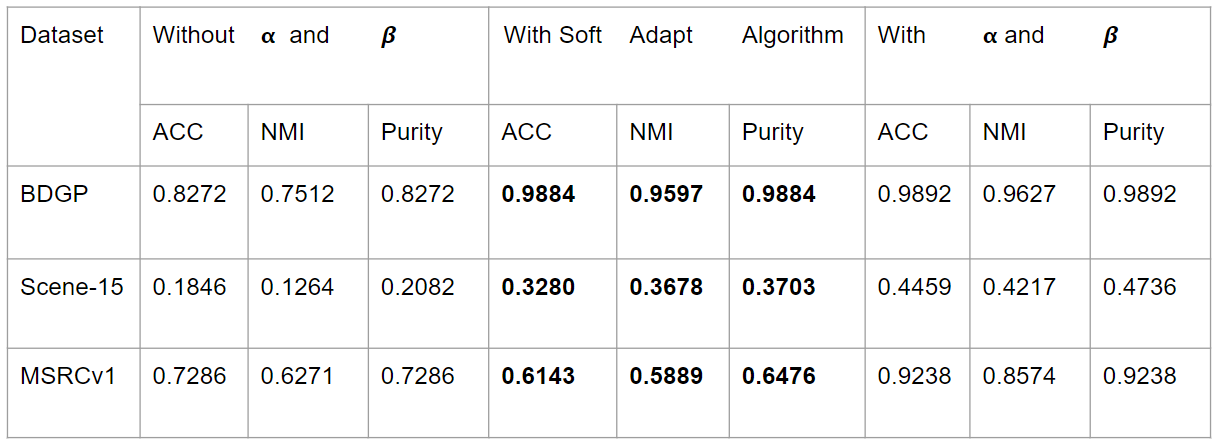
In the soft Adapt algorithm, to calculate the lost components' updated weights in the loss function, we need to calculate the rate of change of the past loss components until ( n-1) th order. For simplicity, we are considering a fifth-order finite difference in our implementation. The mean of respective loss components is also calculated. Using (2) formulae, we can calculate the updated weights.

During epochs less than n+1, where n denotes the nth order of finite difference, we just store the values of loss components. When our epoch reaches n+1, we start calculating the rate of change and average weight values for the last n lost values. These two parameters are employed to estimate the updated weights. We pass these weight values to our contrastive training function, which calculates and optimizes the loss function.

Three datasets—the BDGP, Scene-15, and MSRCv1 datasets—that were supplied by the original authors were used to evaluate this adaptive loss technique. Three criteria were used to evaluate the cluster assignments: purity, normalized mutual information (NMI), and accuracy (ACC). The details of each dataset's results are provided below.

**4.Experimental Results**

Table1. Experimental results on BDGP, Scene-15, MSRCv1.



The initial experiments were performed on three different datasets including : BDGP, Scene-15, and MSRCv1. The performance was measured based on accuracy (ACC), normalized mutual information (NMI), and purity, which are some of the standard metrics for evaluating the effectiveness of clustering algorithms. The sofAdapt Algorithm is implemented with default value of it’s hyperparameter =0.1.

In the BDGP and Scene-15 even though there was no improvement, the results were comparable to the original algorithm with not more than 0.15 units difference across all the metrics. However on MSRCv1 it performed much poorer compared to the original algorithm

Further experiments were conducted to see how the original algorithm performed without α and β hyperparameters, by making them equal to 1, this means total loss now is just a linear sum of individual loss functions without any weighting. Interestingly, in these tests, the adaptive loss method outperformed the original algorithm, suggesting that the method has potential to drive neural networks towards the desired optimization.

The results of the softAdapt algorithm when compared to the original algorithm with α and β were unexpected and made us further analyze and understand the underlying reasons. The adaptive method appears to be extremely sensitive to the datasets' intrinsic properties, which could result in overfitting or poor adaptation during the training process. This sensitivity could be related to dynamic weight modifications, which, while potentially favorable, require careful tuning to assure their effectiveness across a wide range of data situations.Therefore we tried to tune hyperparameter for the least performing dataset MSRCv1.

Table 2. Experimental results for different 𝞫 values on MSRCv1 dataset

| Dataset | Different values | With Soft | Adapt | Algorithm |
| --- | --- | --- | --- | --- |
|  |  | ACC | NMI | Purity |
| MSRCV1 | 0.3 | 0.6808 | 0.6486 | 0.7124 |
|  | 1.9 | 0.6964 | 0.6564 | 0.7132 |
|  | 6 | 0.8244 | 0.7989 | 0.8244 |

The experiments with different values on the MSRCv1 dataset clearly demonstrate how hyperparameter tuning influences the algorithm's effectiveness. The best performance was at a value of 6, where the accuracy reached 82.4%. This underlines the critical role of hyperparameter selection and adjustment in the SoftAdapt framework, showcasing direct correlations between hyperparameter settings and the adaptive loss method's success in clustering scenarios. The peak might have reached at a point earlier than 𝞫soft=6, but because of the limited time and resources, we could conduct experiments with only certain beta values.

**5. Conclusion**

Unfortunately, our adaptive loss method was not as successful as we desired. Compared against the original algorithm, which had the α and β hyperparameters, the adaptive loss method performed worse for all three metrics for each dataset. For the BDGP dataset, the algorithm does perform well, but it did not provide any improvement over the original algorithm. For the MSRCv1 and Scene-15 datasets, the algorithm did not perform well in general, and in the case of the MSRCv1 dataset, performed significantly worse than the original algorithm. However , in our testing, the adaptive loss algorithm did perform better than the original algorithm when excluding the α and β hyperparameters. These hyperparameters are passed to the contrastive train function when calculating the loss per view. They aid in finding the loss from the forward labels as well as the forward probabilities. Therefore, the adaptive loss algorithm does aid in reducing loss when excluding these control hyperparameters.

Although this algorithm was not successful, there is still an opportunity to implement adaptive loss calculations for the CVCL model. The original paper still uses a single aggregate linear combination for the total loss. Thus the issue may lie in implementation rather than logic. A different method may be required to successfully implement the adaptive loss calculation on CVCL. SoftAdapt did include two other variants of the algorithm; either of these could be applied to CVCL and reported on in the future.We still believe adaptive loss will produce successful results, just not using our current SoftAdapt implementation.

**6. References**

[1] Heydari, A.A., Thompson, C.A. and Mehmood, A., 2019. Softadapt: Techniques for adaptive loss weighting of neural networks with multi-part loss functions. *arXiv preprint arXiv:1912.12355*.

[2] Chen, J., Mao, H., Woo, W.L. and Peng, X., 2023. Deep multiview clustering by contrasting cluster assignments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 16752-16761).