Modification in META-LEARNING FOR LOW-RESOURCE SPEECH EMOTION RECOGNITION

Yash Chandra (211187)

1 Additions and code summary

We tried the following two modifications to the base MAML algorithm

- **Reptile Algorithm:** This approach is mathematically similar to the First- order MAML used by the paper. The key difference lies in the meta step: instead of calculating the loss with updated interim parameters on a randomly sampled batch, we directly take a step in the direction of the interim parameters. Thus, FOMAML requires sampling two set of tasks b_0, b_1 but Reptile only requires b_0 as the calculation of loss in the meta step is omitted. For our case we have multiple task so the meta steps become,
  Meta Step for First order MAML(FOMAML):

  $$\theta_0' = \theta_0 - \alpha * \frac{1}{n} \sum_{i=1}^{n} G_1(\theta_{1,i}), \tag{1}$$

  Meta Step for Reptile:

  $$\theta_0' = \theta_0 - \alpha * \frac{1}{n} \sum_{i=1}^{n} (\theta_{1,i} - \theta_0), \tag{2}$$

  where

  - n is number of tasks in a sample
  - $\theta_{1,i}$ is the interim parameters after updating θ_0 on the task t_i

- **Regularized MAML for generalization(Reg-MAML):** This idea follows from the mathematical decomposition of MAML. We identify the component of gradient responsible for promoting generalization in the model and incorporate this component into the computed gradient. This helps us to control the weightage of terms. The equation for meta step update changes to the following,

  $$G(\theta_0) = G_1(\theta_1) + \lambda * \frac{(G_1(\theta_1) - G_1(\theta_0))}{\alpha}, \tag{3}$$

  $$\theta_0' = \theta_0 - \alpha * G(\theta_0), \tag{4}$$

  For n tasks in a sample

  $$G(\theta_{0,i}) = G_1(\theta_{1,i}) + \lambda * \frac{(G_1(\theta_{1,i}) - G_1(\theta_0))}{\alpha}, \tag{5}$$

  $$\theta_0' = \theta_0 - \alpha * \frac{1}{n} \sum_{i=1}^{n} G(\theta_{0,i}), \tag{6}$$

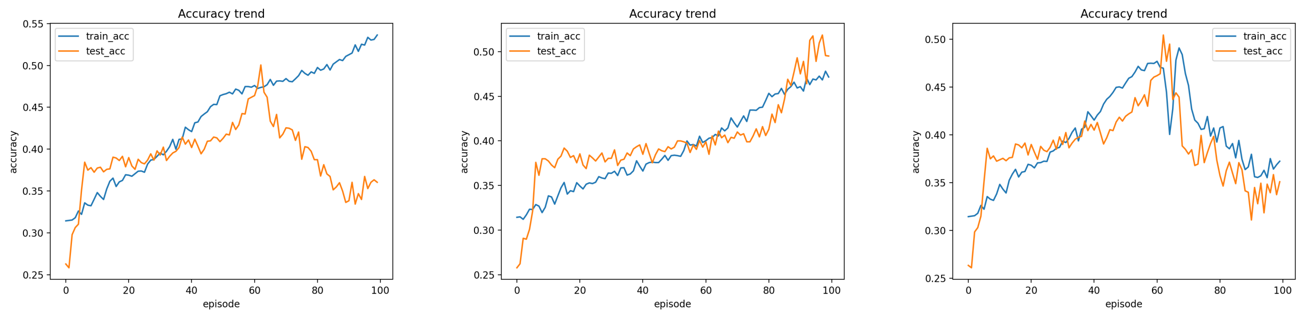
  For detailed mathematical proof of above refer to the mathematical POC pdf submitted.

2 Results

Table 1: Metrics

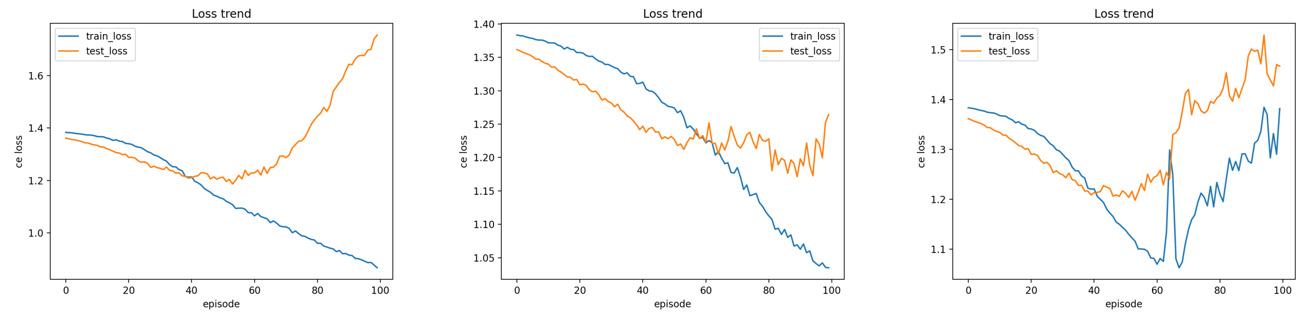
Algorithm	**F1 Score**	**Time taken per 100 epochs**
FOMAML(original)	0.5296	30.39 s
Reptile	0.5661	22.27 s
Reg-MAML	0.5145	37.20 s

Figure 1: Accuracy Trends



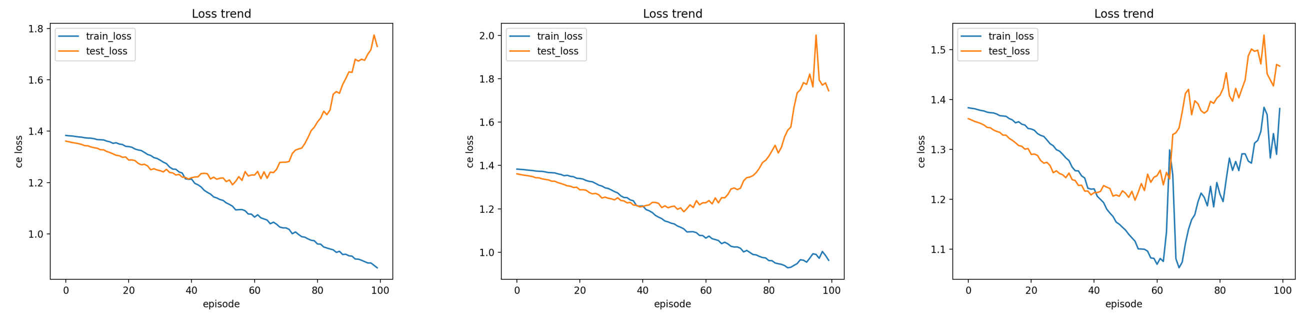
(a) FOMAML (b) Reptile (c) Reg-MAML

Figure 2: Loss Trends



(a) FOMAML (b) Reptile (c) Reg-MAML

Figure 3: Reg-MAML for different Regularization constants(λ)



(a) $\lambda = 0.0001$ (b) $\lambda = 0.001$ (c) $\lambda = 0.01$

- Reptile Algorithm achieved the highest accuracy in 10000 epochs. It took less time per epoch but more epochs to reach optimal value.
- FOMAML took the least time(epochs $*$ time per epochs) to attain its optimal value.
- Reg-MAML took more time and achieved maximum accuracy slightly less than FOMAML.
- In Reg-MAML, the peaks in training and test accuracy, as well as the minima in training and test loss, occur in close proximity. This alignment suggests that optimal model performance can be identified based solely on training trends, eliminating the need for periodic test evaluations to locate the best model.
- Figure 3 shows the effect of regularization in loss trends. As regularization increases, the training loss optimum shifts closer to the optimum of the test loss.

3 Conclusion

We conclude that Reg-MAML algorithm is effective in preventing overfitting. It can be used to omit the need of periodic validation or evaluation of test data. Optimal performance can be identified directly from training metrics, making this approach a strong regularization method for MAML.