Supplementary Materials

1 Optimal Gradient Blending

The theoretical rationale provided in this paper for the optimal gradient blending discussed in Section III.D closely aligns with the explanation given in [1,2]. Let $\mathcal{L}_{train}$ be the model's average training loss over the fixed training set, and $\mathcal{L}_\diamond$ be the true loss with respect to the hypothetical target distribution. The overfitting measure at the training process at epoch n is defined as the gap between $\mathcal{L}_{train}(n)$ and $\mathcal{L}_\diamond(n)$, i.e. $O(n) = \mathcal{L}_{train}(n) - \mathcal{L}_\diamond(n)$. The quality of training between two model checkpoints at the training process at epoch n_0 and n can be measured by the changes in the overfitting measure $\Delta O(n_0, n)$ and the generalization measure $\Delta G(n_0, n)$. We can define the overfitting-to-generalization ratio (OGR) as:

$$ORG = \left|\frac{\Delta O(n_0, n)}{\Delta G(n_0, n)}\right| = \left|\frac{O(n) - O(n_0)}{\mathcal{L}_\diamond(n) - \mathcal{L}_\diamond(n_0)}\right| \tag{1}$$

While it is feasible to reduce the overall OGR throughout the training process, it is not advisable to rely on this metric. This is because very underfit models may still achieve a high OGR, which is misleading. In an alternative perspective, the objective is to address an infinitesimal problem: by combining several gradient estimates, the aim is to blend them to minimize an infinitesimal OGR^2. This ensures that each gradient step yields a gain that is at least as good as that of the single best-task network flow.

Given a single parameter update step with an estimated gradient $\hat{g}$. The distance between two checkpoints allows us to make an approximation: $\Delta O \approx \langle \Delta\mathcal{L}_{train} - \Delta\mathcal{L}_\diamond, \hat{g}\rangle$ and $\Delta G \approx \langle \Delta\mathcal{L}_\diamond, \hat{g}\rangle$. Therefore, OGR^2 for the single vector $\hat{g}$ is given as follows:

$$ORG^2 = \left(\frac{\langle \Delta\mathcal{L}_{train} - \Delta\mathcal{L}_\diamond, \hat{g}\rangle}{\langle \Delta\mathcal{L}_\diamond, \hat{g}\rangle}\right) \tag{2}$$

Let $\hat{g}^{(m)}$ be per-task gradients, $m \in \{1, 2, 3\}$, for three learning tasks in our case, obtained by back-propagation through their specific loss separately (so per-task gradients contain many zeros in other parts of the network), we then aim to blend them into a single vector with better generalization behavior.

Assuming $v^{(m)}$ represents a collection of estimations for $\Delta\mathcal{L}_\diamond$ which is prone to overfitting $\mathbb{E}[\langle \Delta\mathcal{L}_{train} - \Delta\mathcal{L}_\diamond, v^{(m)}\rangle\langle \Delta\mathcal{L}_{train} - \Delta\mathcal{L}_\diamond, v^{(j)}\rangle] = 0$, for $j \neq m$. Given the constraint $\sum_m w^{(m)} = 1$, the optimal weights $w^{(m)}_{optimal} \in \mathbb{R}$ for the optimization problem.

$$w_{optimal} = \underset{w}{\operatorname{argmin}}\, \mathbb{E}\left[\left(\frac{\langle \Delta\mathcal{L}_{train} - \Delta\mathcal{L}_\diamond, \sum_m w^{(m)} v^{(m)}\rangle}{\langle \Delta\mathcal{L}_\diamond, \sum_m w^{(m)} v^{(m)}\rangle}\right)^2\right] \tag{3}$$

are given by

$$w^{(m)}_{optimal} = \frac{1}{Z}\frac{\langle \Delta\mathcal{L}_\diamond, v^{(m)}\rangle}{\sigma^{(m)2}}, \tag{4}$$

where $\sigma^{(m)2} \equiv \mathbb{E}[\langle \Delta\mathcal{L}_{train} - \Delta\mathcal{L}_{\diamond}, v^{(m)} \rangle^2]$ and $Z = \sum_m \frac{\langle \Delta\mathcal{L}_{\diamond}, v^{(m)} \rangle}{2\sigma^{(m)2}}$ is a normalizing factor where the proof is found in the work by [1].

The answer to the optimization problem mentioned above is approximated using the multi-task architecture of MixNet. In each back-propagation step, we compute the gradients $\Delta\mathcal{L}^{(m)}$, $m \in \{1, 2, 3\}$ for individual tasks. This enables us to calculate the gradient of the weighted loss as follows:

$$\mathcal{L}(n) = \sum_{m \in \{1,2,3\}} w^{(m)}(n)\mathcal{L}^{(m)}(n) \tag{5}$$

where the blended gradient can be obtained from $\sum_{m \in \{1,2,3\}} w^{(m)} \Delta\mathcal{L}^{(m)}$. Hence, assigning suitable values to $w^{(m)}$ will provide a convenient method for implementing gradient blending through loss re-weighting.

References

[1] W. Wang, D. Tran, and M. Feiszli, "What makes training multi-modal classification networks hard?" in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 12 695–12 705.

[2] H. Phan, O. Y. Chén, M. C. Tran, P. Koch, A. Mertins, and M. De Vos, "Xsleepnet: Multi-view sequential model for automatic sleep staging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5903–5915, 2022.