Supplementary Material for MixupE

Yingtian Zou[1] **Vikas Verma**[2,3] **Sarthak Mittal**[2] **Wai Hoh Tang**[1] **Hieu Pham**[4] **Juho Kannala**[3]
Yoshua Bengio[2] **Arno Solin**[3] **Kenji Kawaguchi**[1]

[1]National University of Singapore, Singapore
[2]Universite de Montreal, Mila, Canada
[3]Aalto University, Finland
[4]Google Brain, USA

A NOTATIONS

We denote by $z = (\mathbf{x}, \mathbf{y})$ the input and output pair where $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ and $\mathbf{y} \in \mathcal{Y} \subseteq \mathbb{R}^C$. Let $f_\theta(\mathbf{x}) \in \mathbb{R}^C$ be the output of the logits (i.e., the last layer before the softmax or sigmoid) of the model parameterized by θ. We use $\ell(\theta, \mathbf{z}) = h(f_\theta(\mathbf{x})) - \mathbf{y}^\top f_\theta(\mathbf{x})$ to denote the loss function. Let $g(\cdot)$ be the activation function. We use $\mathbf{x}_{(i)}$ to index i-th element of the vector $\mathbf{x}$ and $\mathbf{x}_j$ to represent j-th variable in a set. The notation list is:

- $S = \{\mathbf{x}_i, \mathbf{y}_i\}_{i \in [n]}$ is the fixed training set while $\mathbf{x}'$ is the random test sample.
- ℓ is the loss function for any data point.
- $L_n^{mix}(\theta, S)$: empirical risk of Mixup of size n with parameters θ.
- $\mathcal{L}$: empirical risk of *MixupE*.
- Θ: the constraint set of parameters θ.
- $\mathcal{R}(\Theta, S)$: Empirical Rademacher complexity of set Θ over training set S.
- $\mathbf{J}_a(b)$: Jacobian matrix of a w.r.t b.

B PROOF OF THEOREM 1

Proof. For the cross-entropy loss, we have

$$\ell(\theta, (\mathbf{x}, \mathbf{y})) = -\log \frac{\exp(\mathbf{y}^\top f_\theta(\mathbf{x}))}{\sum_j \exp(f_\theta(\mathbf{x})_{(j)})} = \log\left(\sum_j \exp(f_\theta(\mathbf{x})_{(j)})\right) - \mathbf{y}^\top f_\theta(\mathbf{x}) \tag{1}$$

where $\mathbf{y} \in \mathbb{R}^C$ is a one-hot vector. For the logistic loss,

$$\ell(\theta, (\mathbf{x}, \mathbf{y})) = -\log \frac{\exp(\mathbf{y} f_\theta(\mathbf{x}))}{1 + \exp(f_\theta(\mathbf{x}))} = \log\left(1 + \exp(f_\theta(\mathbf{x})\right) - \mathbf{y} f_\theta(\mathbf{x}). \tag{2}$$

Thus, for both cases, we can write

$$\ell(\theta, (\mathbf{x}, \mathbf{y}) = h(f_\theta(\mathbf{x})) - \mathbf{y}^\top f_\theta(\mathbf{x}) \tag{3}$$

where $h(\mathbf{z}) = \log\left(\sum_j \exp(\mathbf{z}_j)\right)$ for the cross-entropy loss and $h(\mathbf{z}) = \log(1 + \exp(\mathbf{z}))$ for the logistic loss. Using this and equation (9) of [Zhang et al., 2021], we have that

$$L_n^{\text{mix}}(\theta, S) = \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{\lambda \sim \mathcal{D}_\lambda} \mathbb{E}_{\mathbf{x}' \sim \mathcal{D}_X} l(\theta, (r_i(\mathbf{x}'), \mathbf{y}_i)),$$

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

where $\mathcal{D}_X$ is the empirical distribution induced by training samples, and

$$r_i(\mathbf{x}) = \lambda \mathbf{x}_i + (1-\lambda)\mathbf{x}. \tag{4}$$

Define $a_\lambda = 1-\lambda$. Then,

$$r_i(\mathbf{x}') = (1-a_\lambda)\mathbf{x}_i + a_\lambda \mathbf{x}' = \mathbf{x}_i + a_\lambda(\mathbf{x}' - \mathbf{x}_i). \tag{5}$$

Define

$$\varphi_i(a_\lambda) := f_\theta(\mathbf{x}_i + a_\lambda(\mathbf{x}' - \mathbf{x}_i)) \tag{6}$$

Assume f_θ lies in the C^K manifold (K-times differentiable), then there exists a function ψ_i such that $\lim_{a_\lambda \to 0} \psi_i(a_\lambda) = 0$ and with Taylor expansion at $a_\lambda = 0$, we have

$$\begin{aligned}\varphi_i(a_\lambda) &= \varphi_i(0) + \sum_{k=1}^{K} \frac{a_\lambda^k}{k!}\varphi_i^{(k)}(0) + a_\lambda^K \psi_i(a_\lambda)\\ &= f_\theta(\mathbf{x}_i) + \sum_{k=1}^{K} \frac{a_\lambda^k}{k!}\varphi_i^{(k)}(0) + a_\lambda^K \psi_i(a_\lambda)\end{aligned} \tag{7}$$

where $\varphi_i^{(k)}(0)$ is the k-th order derivative at $a_\lambda = 0$, $\psi_i(a_\lambda)$ is the remainder term:

$$\psi_i(a_\lambda) = \int_{\mathbb{R}} \varphi_i^{(K)}(a_\lambda) da_\lambda - \frac{1}{k!}\varphi_i^{(K)}(0) \tag{8}$$

Here, for any $k \in \mathbb{N}^+$, we have

$$\begin{aligned}\varphi_i^{(k)}(0) = \varphi_i^{(k)}(a_\lambda)|_{a_\lambda=0} &= \frac{\partial^k f_\theta(\mathbf{x}_i + a_\lambda(\mathbf{x}'-\mathbf{x}_i))}{\partial(\mathbf{x}_i + a_\lambda(\mathbf{x}'-\mathbf{x}_i))^k}(\mathbf{x}'-\mathbf{x}_i)^{\otimes k}\Big|_{a_\lambda=0}\\ &= \frac{\partial^k f_\theta(\mathbf{x}_i)}{\partial(\mathbf{x}_i)^k}(\mathbf{x}'-\mathbf{x}_i)^{\otimes k}\end{aligned} \tag{9}$$

where $\otimes$ denotes Kronecker product and thus $(\mathbf{x}'-\mathbf{x}_i)^{\otimes k} \in \mathbb{R}^{d^k}$. We can then rewrite $\varphi_i^{(k)}(0)$ as

$$\varphi_i^{(k)}(0) = \mathbf{J}_{f_\theta}^k(\mathbf{x}_i)(\mathbf{x}'-\mathbf{x}_i)^{\otimes k} \tag{10}$$

Plug back into the (7), we have

$$\begin{aligned}f_\theta(\mathbf{x}_i + a_\lambda(\mathbf{x}'-\mathbf{x}_i)) &= f_\theta(\mathbf{x}_i) + \sum_{k=1}^{K}\frac{a_\lambda^k}{k!}\mathbf{J}_{f_\theta}^k(\mathbf{x}_i)(\mathbf{x}'-\mathbf{x}_i)^{\otimes k} + a_\lambda^K\psi_i(a_\lambda)\\ &= f_\theta(\mathbf{x}_i) + a_\lambda \underbrace{\left(\sum_{k=1}^{K}\frac{a_\lambda^{k-1}}{k!}\mathbf{J}_{f_\theta}^k(\mathbf{x}_i)(\mathbf{x}'-\mathbf{x}_i)^{\otimes k} + a_\lambda^{K-1}\psi_i(a_\lambda)\right)}_{\Delta_i}\end{aligned} \tag{11}$$

Above equation will be

$$\begin{aligned}\ell(\theta, (r_i(\mathbf{x}), \mathbf{y}_i)) &= \ell[\theta, (\mathbf{x}_i + a_\lambda(\mathbf{x}'-\mathbf{x}_i), \mathbf{y}_i)]\\ &= h(f_\theta(\mathbf{x}_i + a_\lambda(\mathbf{x}'-\mathbf{x}_i))) - \mathbf{y}_i^\top f_\theta(\mathbf{x}_i + a_\lambda(\mathbf{x}'-\mathbf{x}_i))\\ &= h(f_\theta(\mathbf{x}_i) + a_\lambda\Delta_i) - \mathbf{y}_i^\top(f_\theta(\mathbf{x}_i) + a_\lambda\Delta_i).\end{aligned} \tag{12}$$

Analogously, we can define $\hat{\varphi}_i^{(k)}(a_\lambda) := h(f_\theta(\mathbf{x}_i) + a_\lambda\Delta_i)$ and the parallel notation $\hat{\psi}_i(a_\lambda)$, then

$$h(f_\theta(\mathbf{x}_i) + a_\lambda\Delta_i) = h(f_\theta(\mathbf{x}_i)) + \sum_{k=1}^{K}\frac{a_\lambda^k}{k!}\mathbf{J}_{h\circ f_\theta}^k(\mathbf{x}_i)\Delta_i^{\otimes k} + a_\lambda^K\hat{\psi}_i(a_\lambda) \tag{13}$$

Combining these,

$$\begin{aligned}\ell(\theta,(r_i(\mathbf{x}),\mathbf{y}_i)) &= h(f_\theta(\mathbf{x}_i)) - \mathbf{y}_i^\top f_\theta(\mathbf{x}_i) - a_\lambda \mathbf{y}_i \Delta_i + \sum_{k=1}^{K} \frac{a_\lambda^k}{k!} \mathbf{J}_{h\circ f_\theta}^k(\mathbf{x}_i)\Delta_i^{\otimes k} + a_\lambda^K \hat{\psi}_i(a_\lambda) \\ &= \ell(\theta,(\mathbf{x},\mathbf{y}_i)) - a_\lambda \mathbf{y}_i^\top \Delta_i + \sum_{k=1}^{K} \frac{a_\lambda^k}{k!} \mathbf{J}_{h\circ f_\theta}^k(\mathbf{x}_i)\Delta_i^{\otimes k} + a_\lambda^K \hat{\psi}_i(a_\lambda)\end{aligned} \tag{14}$$

Thus, the implicit regularization of Mixup can be unfolded as

$$\begin{aligned}L_n^{\text{mix}}(\theta,S) &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\lambda\sim\mathcal{D}_\lambda}\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_X} l(\theta,(r_i(\mathbf{x}),\mathbf{y}_i)) \\ &= L_n^{std}(\theta,S) + \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\lambda\sim\mathcal{D}_\lambda}\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_X}\left(\sum_{k=1}^{K}\frac{a_\lambda^k}{k!}\mathbf{J}_{h\circ f_\theta}^k(\mathbf{x}_i)\Delta_i^{\otimes k} - a_\lambda \mathbf{y}_i^\top \Delta_i + a_\lambda^K \hat{\psi}_i(a_\lambda)\right),\end{aligned} \tag{15}$$

where

$$\Delta_i = \sum_{k=1}^{K} \frac{a_\lambda^{k-1}}{k!} \mathbf{J}_{f_\theta}^k(\mathbf{x}_i)(\mathbf{x}' - \mathbf{x}_i)^{\otimes k} + a_\lambda^{K-1}\psi_i(a_\lambda). \tag{16}$$

Note that with probability 1, we have

$$\lim_{a_\lambda\to 0} \hat{\psi}_i(a_\lambda) = 0,\ \lim_{a_\lambda\to 0} \psi_i(a_\lambda) = 0$$

□

C PROOF OF THEOREM 2

The Rademacher generalization bound is widely applied where the empirical Rademacher complexity of a function class Θ is given by:

$$\mathcal{R}_n(\Theta, \{\mathbf{x}_i\}_{i\in[n]}) = \mathbb{E}\left[\sup_{\theta\in\Theta}\frac{1}{n}\sum_{i=1}^{n} f_\theta(\mathbf{x}_i)\,\epsilon_i\right] \tag{17}$$

where, Rademacher r.v ϵ_i independently takes values in $\{-1,+1\}$ with equal probability.

Lemma 1. *(Bartlett and Mendelson [2002]). For any B-uniformly bounded and L Lipchitz function ζ, for all $\phi\in\Phi$, with probability at least $1-\delta$,*

$$\mathbb{E}\zeta(\phi(\mathbf{x}_i)) \le \frac{1}{n}\sum_{i=1}^{n}\zeta(\phi(\mathbf{x}_i)) + 2L\mathcal{R}_n(\Phi,S) + B\sqrt{\frac{\log(1/\delta)}{2n}}$$

Proof. Consider GLM that $h(f_\theta(\mathbf{x})) = A(\theta^\top\mathbf{x})$ and training set S, the constraint of $\Theta = \{\mathbf{x}\to f_\theta(\mathbf{x})|\sup_{\mathbf{x}}\hat{q}(\mathbf{x})\le\gamma\}$ implies that

$$\sup_{\mathbf{x}}|\hat{q}_i(\mathbf{x})| = \sup_{\mathbf{x}}(\mathbf{y} - A'(\theta^\top\mathbf{x}))^\top(\theta^\top\mathbf{x}) \le \gamma \tag{18}$$

Rearranging the terms, and by Cauchy–Schwarz inequality we have

$$\begin{aligned}\gamma &\ge \sup_{\mathbf{x}}(\mathbf{y} - A'(\theta^\top\mathbf{x}))^\top(\theta^\top\mathbf{x}) \\ &= \sup_{\mathbf{x}}\langle\mathbf{y},\theta^\top\mathbf{x}\rangle - \sup_{\mathbf{x}}\langle A'(\theta^\top\mathbf{x}),\theta^\top\mathbf{x}\rangle \\ &\ge \sup_{\mathbf{x}}\langle\mathbf{y},\theta^\top\mathbf{x}\rangle - \sup_{\mathbf{x}}\|A'(\theta^\top\mathbf{x})\|_2\|\theta^\top\mathbf{x}\|_2\end{aligned} \tag{19}$$

Due to the fact that $A(\cdot)$ is a L_A Lipchitz function, then it's trivial to prove

$$\|A'(\theta^\top\mathbf{x})\|_2 \le L_A \tag{20}$$

Let $\mathbf{y} = (\theta^*)^\top \mathbf{x} = (\Sigma\theta)^\top \mathbf{x}$ where Σ is the diagonal matrix. Thus the above relation will be

$$\begin{aligned}\gamma &\geq \sup_{\mathbf{x}} \langle \mathbf{y}, \theta^\top \mathbf{x} \rangle - \sup_{\mathbf{x}} \|A'(\theta^\top \mathbf{x})\|_2 \|\theta^\top \mathbf{x}\|_2 \\ &\geq \sup_{\mathbf{x}} \langle (\Sigma\theta)^\top \mathbf{x}, \theta^\top \mathbf{x} \rangle - L_A \sup_{\mathbf{x}} \|\theta^\top \mathbf{x}\|_2\end{aligned} \tag{21}$$

Let $\mathbf{v} = \sup_{\mathbf{x}} \theta^\top \mathbf{x}$ and $\overline{\sigma}$ be the expected value that $\overline{\sigma} = \mathbb{E}_{j \in [d]} \sup_{\mathbf{x}_i} \Sigma_i(j) = \sup_{\mathbf{x}} \frac{\mathrm{tr}(\Sigma)}{d}$, then we have

$$\gamma \geq \overline{\sigma} \|\mathbf{v}\|_2^2 - L_A \|\mathbf{v}\|_2 \tag{22}$$

which implies

$$\frac{L_A - \sqrt{L_A^2 + 4\gamma\overline{\sigma}}}{2\overline{\sigma}} \leq \|\mathbf{v}\|_2 \leq \frac{L_A + \sqrt{L_A^2 + 4\gamma\overline{\sigma}}}{2\overline{\sigma}} \tag{23}$$

Obviously,

$$\left| \frac{L_A + \sqrt{L_A^2 + 4\gamma\overline{\sigma}}}{2\overline{\sigma}} \right| > \left| \frac{L_A + \sqrt{L_A^2 - 4\gamma\overline{\sigma}}}{2\overline{\sigma}} \right| \tag{24}$$

Denote $\mathbf{v}_i = \theta^\top \mathbf{x}_i$, we have the Rademacher complexity $\mathcal{R}(\Theta, S)$ that

$$\begin{aligned}\mathcal{R}(\Theta, S) &= \mathbb{E}_\epsilon \sup_{\mathbb{E}_{\mathbf{x}} \hat{q}(\mathbf{x}) \leq \gamma} \frac{1}{n} \sum_{i=1}^n \epsilon_i \theta^\top \mathbf{x}_i \\ &\leq \mathbb{E}_\epsilon \sup_{\|\mathbf{v}_i\|_2^2 \leq \left(\frac{L_A + \sqrt{L_A^2 + 4\gamma\overline{\sigma}}}{2\overline{\sigma}} \right)^2} \frac{1}{n} \sum_{i=1}^n \epsilon_i \mathbf{v}_i \\ &\leq \frac{1}{n} \cdot \frac{L_A + \sqrt{L_A^2 + 4\gamma\overline{\sigma}}}{2\overline{\sigma}} \cdot \sqrt{\mathbb{E}_\epsilon \left(\sum_{i=1}^n \epsilon_i \right)^2} \\ &= \frac{1}{\sqrt{n}} \frac{L_A + \sqrt{L_A^2 + 4\gamma\overline{\sigma}}}{2\overline{\sigma}} \\ &\leq \frac{1}{\sqrt{n}} \frac{2L_A + 2\sqrt{\gamma\overline{\sigma}}}{2\overline{\sigma}} \\ &= \frac{L_A + \sqrt{\gamma\overline{\sigma}}}{\overline{\sigma}\sqrt{n}}\end{aligned} \tag{25}$$

Consequently, we have

$$\mathcal{R}(\Theta, S) \leq \frac{L_A + \sqrt{\gamma\overline{\sigma}}}{\overline{\sigma}\sqrt{n}} \tag{26}$$

Recall the objective of *MixupE*,

$$\mathcal{L}(\theta, S) := \hat{\eta} \left(L_n^{mix}(\theta, S) + \eta R(\theta, S) \right) \tag{27}$$

$$\hat{\eta} = \frac{|L_n^{mix}(\theta, S)|}{|L_n^{mix}(\theta, S) + \eta R(\theta, S)|} \tag{28}$$

With Lemma 1, we can get

$$\begin{aligned}\mathcal{L}(\theta, S) &\leq \hat{\eta} L_n^{mix}(\theta, S) + 2\hat{\eta}\eta L \mathcal{R}(\Theta, S) + B\sqrt{\frac{\log(1/\delta)}{2n}} \\ &\leq \hat{\eta} L_n^{mix}(\theta, S) + \frac{2\hat{\eta}\eta L L_A (L_A + \sqrt{\gamma\overline{\sigma}})}{\overline{\sigma}\sqrt{n}} + B\sqrt{\frac{\log(1/\delta)}{2n}}\end{aligned} \tag{29}$$

$\square$

C.1 COMPARISON TO VANILLA MIXUP

As a comparison, for vanilla Mixup with parameter space $\hat{\Theta} = \{\theta | \|\theta\|_2^2 \leq \gamma\}$ and assume $\|\mathbf{x}_i\|^2 \leq \mathcal{X}, \forall i \in [n]$ the Rademacher complexity will be

$$\begin{aligned}
\mathcal{R}(\hat{\Theta}, S) &= \mathbb{E}_\epsilon \sup_{\|\theta\|_2^2 \leq \gamma} \frac{1}{n} \sum_{i=1}^n \epsilon_i \theta^\top \mathbf{x}_i \\
&= \frac{1}{n} \mathbb{E}_\epsilon \sup_{\|\theta\|_2^2 \leq \gamma} \sqrt{\sum_{i=1}^n \epsilon_i^2 \|\theta\|^2 \|\mathbf{x}_i\|^2} \\
&= \frac{\sqrt{\gamma}}{n} \mathbb{E}_\epsilon \sqrt{\sum_{i=1}^n \epsilon_i^2 \|\mathbf{x}_i\|^2} \\
&\leq \frac{\sqrt{\gamma \mathcal{X}}}{\sqrt{n}}
\end{aligned} \tag{30}$$

Compared to the Rademacher complexity of Mixup, we found that *MixupE* don't need to bound the norm of input data by $\mathcal{X}$ which may cause a large term. However, if considering normalized input space where $\mathcal{X} \leq 1$, the condition to have a shrink parameter space is

$$\frac{L_A + \sqrt{\gamma \overline{\sigma}}}{\overline{\sigma}} \leq \sqrt{\gamma} \Rightarrow \frac{L_A}{\overline{\sigma} - \sqrt{\overline{\sigma}}} \leq \sqrt{\gamma} \quad \text{and} \quad \overline{\sigma} > 1 \tag{31}$$

Thus, when the above condition is satisfied, our regularization reduces the norm of parameter space for the case where input space is normalized $\mathcal{X} \leq 1$. In general, the $\overline{\sigma}$ is the average entry value of the maximum correction matrix to the ground truth which can be quite large. Scaling by σ, it is probably satisfied in most cases.

D IMPLEMENTATION

The code implementation in PyTorch is shown as Listing 1.

```
def beta_mean(alpha, beta):
    return alpha/(alpha+beta)

lam_mod_mean = beta_mean(alpha+1, alpha) # mean of beta distribution

# y1, y2 should be one-hot vectors
for (x1, y1), (x2, y2) in zip(loader1, loader2):
    lam = numpy.random.beta(alpha, alpha)
    x = Variable(lam * x1 + (1. - lam) * x2)
    y = Variable(lam * y1 + (1. - lam) * y2)
    loss = loss_function(net(x), y)  # mixup loss
    loss_scale = torch.abs(loss.detach().data.clone())
    f = net(x1)
    b = y1 - torch.softmax(f, dim=1)
    loss_new = torch.sum(f * b, dim=1)
    loss_new = (1.0 - lam_mod_mean) * torch.sum(torch.abs(loss_new)) / batch_size  #
    additional loss term
    loss = loss + (mixup_eta * loss_new)  # total loss
    loss_new_scale = torch.abs(loss.detach().data.clone())
    loss = (loss_scale / loss_new_scale) * loss  # loss after scaling
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
```

Listing 1: One epoch *MixupE* training in PyTorch

References

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Linjun Zhang, Zhun Deng, and Kenji Kawaguchi. How does mixup help with robustness and generalization? In *International Conference on Learning Representations (ICLR)*, 2021.