Supplementary Materials: MentorNet Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels

Lu Jiang Zhengyuan Zhou Thomas Leung Li-Jia Li Li Fei-Fei

A. Derivation of Remark 1

Our objective function is:

$$\min_{\mathbf{w}\in\mathbb{R}^d,\mathbf{v}\in[0,1]^n} \mathbb{F}(\mathbf{w},\mathbf{v}) = \frac{1}{n}\sum_{i=1}^{n} v_i \mathbf{L}(y_i, g_s(\mathbf{x}_i,\mathbf{w})) + G(\mathbf{v};\lambda) + \theta\|\mathbf{w}\|_2^2 \tag{1}$$

Let $\ell_i = \mathbf{L}(y_i, g_s(\mathbf{x}_i, \mathbf{w}))$ denote the loss of the i-th sample ($\ell_i \geq 0$). The predefined curriculum is defined as:

$$G(\mathbf{v};\lambda) = \sum_{i=1}^{n} \frac{1}{2}\lambda_2 v_i^2 - (\lambda_2 + \lambda_1)v_i. \tag{2}$$

Denote $\mathbb{F}_{\mathbf{w}}$ as the objective function when the $\mathbf{w}$ is fixed. We have

$$\begin{aligned}
\mathbb{F}_{\mathbf{w}}(\mathbf{v}) &= \frac{1}{n}\sum_{i=1}^{n} v_i\ell_i + G(\mathbf{v};\lambda) + \theta\|\mathbf{w}\|_2^2 \\
&= \frac{1}{n}\sum_{i=1}^{n} v_i\ell_i + \frac{1}{2}\lambda_2 v_i^2 - (\lambda_2+\lambda_1)v_i + \theta\|\mathbf{w}\|_2^2 \\
&= \frac{1}{n}\sum_{i=1}^{n} f(v_i) + \theta\|\mathbf{w}\|_2^2
\end{aligned} \tag{3}$$

where $f(v_i) = v_i\ell_i + \frac{1}{2}\lambda_2 v_i^2 - (\lambda_2+\lambda_1)v_i$.

As $f(v_i)$ is convex with respect to v_i ($\lambda_2 \geq 0$). Its minimum is obtained at

$$\begin{aligned}
&\arg\min_{\mathbf{v}\in[0,1]^n} \nabla_{\mathbf{v}}\mathbb{F}_{\mathbf{w}}(\mathbf{v}) = 0 \\
&\Rightarrow \frac{\partial f(v_i)}{\partial v_i} = \ell_i + \lambda_2 v_i - \lambda_2 - \lambda_1 = 0,\ (\forall i \in [1,n], v_i \in [0,1])
\end{aligned} \tag{4}$$

Since $\lambda_1, \lambda_2 \geq 0$ and v_i is bounded in $[0,1]$. When $\lambda_2 \neq 0$, the optimal v_i^* is calculated from:

$$v_i^* = \begin{cases} 1 & (\ell_i \leq \lambda_1) \wedge (\lambda_2 \neq 0) \\ 1 - \frac{\ell_i - \lambda_1}{\lambda_2} & (\lambda_1 < \ell_i < \lambda_2 + \lambda_1) \wedge (\lambda_2 \neq 0) \\ 0 & (\ell_i \geq \lambda_2 + \lambda_1) \wedge (\lambda_2 \neq 0) \end{cases}, \tag{5}$$

When $\lambda_2 = 0$, the optimal weight writes as:

$$v_i^* = \begin{cases} 1 & (\ell_i < \lambda_1) \wedge (\lambda_2 = 0) \\ 0 & (\ell_i \geq \lambda_1) \wedge (\lambda_2 = 0) \end{cases}. \tag{6}$$

Combined Eq. (5) and Eq. (6), we have

$$v_i^* = \begin{cases} \mathbb{1}(\ell_i \leq \lambda_1) & \lambda_2 = 0 \\ \min(\max(0, 1 - \frac{\ell_i - \lambda_1}{\lambda_2}), 1) & \lambda_2 \neq 0 \end{cases} \tag{7}$$

According to the definition of Θ, we have $g_m(\phi(\mathbf{x}_i, y_i, \mathbf{w}); \Theta^*) = v_i^*$. Incorporating Eq. (7), we have

$$g_m(\phi(\mathbf{x}_i, y_i, \mathbf{w}); \Theta^*) = \begin{cases} \mathbb{1}(\ell_i \leq \lambda_1) & \lambda_2 = 0 \\ \min(\max(0, 1 - \frac{\ell_i - \lambda_1}{\lambda_2}), 1) & \text{otherwise} \end{cases} \tag{8}$$

Now we derive its underlying objective. First, we define a function $v^*(\lambda, x)$ and incorporate the above optimal solution:

$$v^*(\lambda, x) = \arg\min_{v \in [0,1]} \; vx + G(v, \lambda) = \begin{cases} \mathbb{1}(x \leq \lambda_1) & \lambda_2 = 0 \\ \min(\max(0, 1 - \frac{x - \lambda_1}{\lambda_2}), 1) & \text{otherwise} \end{cases} \tag{9}$$

Let $\epsilon > 0$ denote a small positive constant. Using the condition $\lambda_2 \geq 0$, we incorporate Eq. (7):

$$v^*(\lambda, x + \epsilon) - v^*(\lambda, x) \leq 0 \tag{10}$$

That indicates $v^*(\lambda, x)$ decreases with respect to x. According to (Meng et al., 2015), given the fixed hyperparameter λ (*i.e.* λ_1, λ_2), its underlying objective function has the form of:

$$F_\lambda(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \int_0^{\ell_i} v^*(\lambda; x) dx, \tag{11}$$

After incorporating Eq. (5) and Eq. (6) into Eq. (11), we have when $\lambda_2 = 0$

$$F_\lambda(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \min(\ell_i, \lambda_1) \tag{12}$$

When $\lambda_2 \neq 0$, we have:

$$\begin{aligned} F_\lambda(\mathbf{w}) &= \frac{1}{n} \sum_{i=1}^{n} \begin{cases} \ell_i & \ell_i \leq \lambda_1 \\ \ell_i + \frac{\lambda_1}{\lambda_2} \ell_i - \frac{1}{2\lambda_2} \ell_i^2 - \frac{\lambda_1^2}{2\lambda_2} & \lambda_1 < \ell_i < \lambda_2 + \lambda_1 \\ \frac{\lambda_2 + 2\lambda_1}{2} & \ell_i \geq \lambda_2 + \lambda_1 \end{cases} \\ &= \frac{1}{n} \sum_{i=1}^{n} \begin{cases} \ell_i & \ell_i \leq \lambda_1 \\ \theta \ell_i - \ell_i^2/(2\lambda_2) - \frac{(\theta-1)^2 \lambda_2}{2} & \lambda_1 < \ell_i < \lambda_2 + \lambda_1 \\ (\lambda_2 + 2\lambda_1)/2 & \ell_i \geq \lambda_2 + \lambda_1 \end{cases} \end{aligned}, \tag{13}$$

where $\theta = (\lambda_2 + \lambda_1)/\lambda_2$ and $\lambda_1, \lambda_2 \geq 0$. We $\lambda_2 \neq 0$, we have the Eq.(10) in the Remark 1 in the paper.

$$F_\lambda(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \begin{cases} \ell_i & \ell_i \leq \lambda_1 \\ (\lambda_2 + 2\lambda_1)/2 & \ell_i \geq \lambda_2 + \lambda_1 \\ \theta \ell_i - \ell_i^2/(2\lambda_2) - \frac{(\theta-1)^2 \lambda_2}{2} & \text{otherwise} \end{cases}, \tag{14}$$

When $\theta = 1$ we have $\lambda_1 = 0$ and the above equation becomes:

$$F_\lambda(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \begin{cases} \ell_i - \ell_i^2/(2\lambda_2) & \ell_i < \lambda_2 \\ \lambda_2/2 & \ell_i \geq \lambda_2 \end{cases}. \tag{15}$$

The above equation is equivalent to the minimax concave penalty (MCP) (Gong et al., 2013; Zhang, 2010). Below is its formula presented in (Gong et al., 2013) (with the regularization hyperparameter setting to 1):

$$\int_0^{\ell} [1 - \frac{x}{t}] dx = \begin{cases} \ell - \ell^2/(2t) & \ell < t \\ t/2 & \ell \geq t \end{cases}, \tag{16}$$

B. Proof of Theorem 1

Theorem 1 *Let the objective $\mathbb{F}(\mathbf{w}, \mathbf{v})$ defined in Eq.* (1) *be differentiable, $L(\cdot)$ be Lipschitz continuous in $\mathbf{w}$ and $\nabla_{\mathbf{v}} G(\cdot)$ be Lipschitz continuous in $\mathbf{v}$. Let $\mathbf{w}^t, \mathbf{v}^t$ be iterates from Algorithm 1 and $\sum_{t=0}^{\infty} \alpha_t = \infty, \sum_{t=0}^{\infty} \alpha_t^2 < \infty$. Then, $\lim_{t\to\infty} \mathbb{E}[\|\nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^t, \mathbf{v}^t)\|_2^2] = 0$.*

Proof 1 *First, by the definition of the objective function $\mathbb{F}(\mathbf{w}, \mathbf{v})$, it can be easily checked that $L(\cdot)$ being Lipschitz continuous in $\mathbf{w}$ implies that $\nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}, \mathbf{v})$ is a Lipschitz function in $\mathbf{w}$ for every $\mathbf{v}$. Similarly, $\nabla_{\mathbf{v}} G(\cdot)$ being Lipschitz continuous in $\mathbf{v}$ implies that $\nabla_{\mathbf{v}} \mathbb{F}(\mathbf{w}, \mathbf{v})$ is a Lipschitz function in $\mathbf{v}$ for all $\mathbf{w}$. Further, without loss of generality, we assume all the Lipschitz constants are (upper bounded by) L.*

Throughout the proof, as in the main text, n is the size of the training dataset. Define the n-dimensional vector e^k as $e_i^t = 1$ if $(x_i, y_i) \in \Xi_t$ and 0 otherwise and denote by $\otimes$ the point-wise product operation between two vectors: $[a_1, a_2] \otimes [b_1, b_2] = [a_1 b_1, a_2 b_2]$. When G is used, the update in each iteration is two consecutive gradient steps as follows:

$$\begin{aligned} \mathbf{w}^{t+1} &= \mathbf{w}^t - \alpha_t \nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^t, \mathbf{v}^t)|_{\Xi_t}, \\ \mathbf{v}^{t+1} &= \mathbf{v}^t - \alpha_t \nabla_{\mathbf{v}} \mathbb{F}(\mathbf{w}^{t+1}, \mathbf{v}^t)|_{\Xi_t}. \end{aligned}$$

Since the mini-batch Ξ_t is draw uniformly at random, we can rewrite the update as:

$$\begin{aligned} \mathbf{w}^{t+1} &= \mathbf{w}^t - \alpha_t [\nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^k, \mathbf{v}^t) + \xi^t], \\ \mathbf{v}^{t+1} &= \mathbf{v}^t - \alpha_t [e^t \otimes \nabla_{\mathbf{v}} \mathbb{F}(\mathbf{w}^{t+1}, \mathbf{v}^t)], \end{aligned}$$

*where $\xi^t = \nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^t, \mathbf{v}^t)|_{\Xi_t} - \nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^k, \mathbf{v}^t)$. Note that both ξ^t and e^t are **iid** random variables with finite variance, since Ξ_t are drawn **iid** with a finite number (b) of samples. Further, $\mathbb{E}[\nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^t, \mathbf{v}^t)|_{\Xi_t} - \nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^k, \mathbf{v}^t)] = 0$, since samples are drawn uniformly at random.*

By Lipschitz continuity of $\nabla_{\mathbf{w}} F(\mathbf{w}, \mathbf{v})$ (and L being the Lipschitz constant), we obtain the following:

$$\begin{aligned} &\mathbb{F}(\mathbf{w}^{t+1}, \mathbf{v}^t) - \mathbb{F}(\mathbf{w}^t, \mathbf{v}^t) \le \langle \nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^t, \mathbf{v}^t), \mathbf{w}^{t+1} - \mathbf{w}^t \rangle + \frac{L}{2} \|\mathbf{w}^{t+1} - \mathbf{w}^t\|_2^2 \\ &= \langle \nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^t, \mathbf{v}^t), -\alpha_t [\nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^t, \mathbf{v}^t) + \xi^t] \rangle + \frac{L}{2} \|\mathbf{w}^{t+1} - \mathbf{w}^t\|_2^2 \\ &= -\alpha_t \{\|\nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^t, \mathbf{v}^t)\|_2^2 + \langle \nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^t, \mathbf{v}^t), \xi^t \rangle\} + \frac{L}{2} \|\mathbf{w}^{t+1} - \mathbf{w}^t\|_2^2 \\ &= -\alpha_t \{\|\nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^t, \mathbf{v}^t)\|_2^2 + \langle \nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^t, \mathbf{v}^t), \xi^t \rangle\} + \frac{L\alpha_t^2}{2} \|\nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^t, \mathbf{v}^t) + \xi^t\|_2^2 \\ &= -(\alpha_t - \frac{L\alpha_t^2}{2}) \|\nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^t, \mathbf{v}^t)\|_2^2 + \frac{L\alpha_t^2}{2} \|\xi^t\|_2^2 - (\alpha_t - L\alpha_t^2) \langle \nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^t, \mathbf{v}^t), \xi^t \rangle. \end{aligned}$$

Similarly, by the Lipschitz continuity of $\nabla_{\mathbf{v}} F(\mathbf{w}, \mathbf{v})$, we have:

$$\begin{aligned} &\mathbb{F}(\mathbf{w}^{t+1}, \mathbf{v}^{t+1}) - \mathbb{F}(\mathbf{w}^{t+1}, \mathbf{v}^t) \le \langle \nabla_{\mathbf{v}} \mathbb{F}(\mathbf{w}^{t+1}, \mathbf{v}^t), \mathbf{v}^{t+1} - \mathbf{v}^t \rangle + \frac{L}{2} \|\mathbf{v}^{t+1} - \mathbf{v}^t\|_2^2 \\ &= \langle \nabla_{\mathbf{v}} \mathbb{F}(\mathbf{w}^{t+1}, \mathbf{v}^t), -\alpha_t e^t \otimes \nabla_{\mathbf{v}} \mathbb{F}(\mathbf{w}^{t+1}, \mathbf{v}^t) \rangle + \frac{L}{2} \|\mathbf{v}^{t+1} - \mathbf{v}^t\|_2^2 \\ &= -\alpha_t \langle \nabla_{\mathbf{v}} \mathbb{F}(\mathbf{w}^{t+1}, \mathbf{v}^t), e^k \otimes \nabla_{\mathbf{v}} \mathbb{F}(\mathbf{w}^{t+1}, \mathbf{v}^t) \rangle + \frac{L\alpha_t^2}{2} \|e^t \otimes \nabla_{\mathbf{v}} \mathbb{F}(\mathbf{w}^{t+1}, \mathbf{v}^t)\|_2^2. \end{aligned}$$

Note that when G is not used, the bound for $\mathbb{F}(\mathbf{w}^{t+1}, \mathbf{v}^t) - \mathbb{F}(\mathbf{w}^t, \mathbf{v}^t)$ is still the same, but the bound for $\mathbb{F}(\mathbf{w}^{t+1}, \mathbf{v}^{t+1}) - \mathbb{F}(\mathbf{w}^{t+1}, \mathbf{v}^t)$ is now simply $\mathbb{F}(\mathbf{w}^{t+1}, \mathbf{v}^{t+1}) - \mathbb{F}(\mathbf{w}^{t+1}, \mathbf{v}^t) \le 0$.

Combining the above two equations, we then have:

1. *If G is used,*

$$\begin{aligned}
&\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t+1})-\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})=\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t+1})-\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t})+\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t})-\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\\
&\leq-(\alpha_t-\frac{L\alpha_t^2}{2})\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\|_2^2+\frac{L\alpha_t^2}{2}\|\xi^t\|_2^2-(\alpha_t-L\alpha_t^2)\langle\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t}),\xi^t\rangle\\
&-\alpha_t\langle\nabla_{\mathbf{v}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{k}),e^t\otimes\nabla_{\mathbf{v}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t})\rangle+\frac{L\alpha_t^2}{2}\|e^t\otimes\nabla_{\mathbf{v}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t})\|_2^2.
\end{aligned}$$

If G is not used,

$$\begin{aligned}
&\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t+1})-\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\leq\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t})-\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\\
&\leq-(\alpha_t-\frac{L\alpha_t^2}{2})\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\|_2^2+\frac{L\alpha_t^2}{2}\|\xi^t\|_2^2-(\alpha_t-L\alpha_t^2)\langle\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t}),\xi^t\rangle
\end{aligned}$$

Taking expectation of both sides and since $\mathbf{E}[\xi^t]=0$*, we have if G is used:*

$$\begin{aligned}
&\mathbf{E}[\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t+1})]-\mathbf{E}[\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})]\\
&\leq-(\alpha_t-\frac{L\alpha_t^2}{2})\mathbf{E}[\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\|_2^2]+\frac{L\alpha_t^2}{2}\mathbf{E}[\|\xi^t\|_2^2]-\alpha_t\mathbf{E}[\langle\nabla_{\mathbf{v}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t}),e^t\otimes\nabla_{\mathbf{v}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t})\rangle]\\
&+\frac{L\alpha_t^2}{2}\mathbf{E}[\|e^t\otimes\nabla_{\mathbf{v}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t})\|_2^2]\\
&=-(\alpha_t-\frac{L\alpha_t^2}{2})\mathbf{E}[\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\|_2^2]+\frac{L\alpha_t^2}{2}\mathbf{E}[\|\xi^t\|_2^2]-\frac{\alpha_t b}{n}\mathbf{E}[\|\nabla_{\mathbf{v}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t})\|_2^2]\\
&+\frac{L\alpha_t^2}{2}\sum_{i=1}^{n}\mathbf{E}[\|e_i^t\frac{\partial_{v_i}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t})}{\partial v_i}\|_2^2]\\
&\leq-(\alpha_t-\frac{L\alpha_t^2}{2})\mathbf{E}[\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\|_2^2]+\frac{L\alpha_t^2}{2}\mathbf{E}[\|\xi^t\|_2^2]-\frac{\alpha_t b}{n}\mathbf{E}[\|\nabla_{\mathbf{v}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t})\|_2^2]\\
&+\frac{L\alpha_t^2}{2}\sum_{i=1}^{n}\mathbf{E}[\|\frac{\partial_{v_i}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t})}{\partial v_i}\|_2^2]\\
&=-\alpha_t\mathbf{E}[\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\|_2^2]-\frac{\alpha_t b}{n}\mathbf{E}[\|\nabla_{\mathbf{v}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t})\|_2^2]\\
&+\frac{L\alpha_t^2}{2}\{\mathbf{E}[\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\|_2^2]+\mathbf{E}[\|\nabla_{\mathbf{v}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t})\|_2^2]+\mathbf{E}[\|\xi^t\|_2^2].\}
\end{aligned}$$

where the second equality follows from $\mathbf{E}[\langle\nabla_{\mathbf{v}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t}),e^t\otimes\nabla_{\mathbf{v}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t})\rangle]=\frac{b}{n}\mathbf{E}[\|\nabla_{\mathbf{v}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t})\|_2^2]$*, which can be checked by a straightforward combinatorial argument.*

Following a similar chain of steps, we have the following bound if G is not used: $\mathbf{E}[\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t+1})]-\mathbf{E}[\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})]\leq-\alpha_t\mathbf{E}[\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\|_2^2]+\frac{L\alpha_t^2}{2}\{\mathbf{E}[\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\|_2^2]+\mathbf{E}[\|\xi^t\|_2^2].\}$ *Since there are only a finite number of training samples, all the random quantities have bounded support and all the second moments are upper bounded, leading to* $\mathbf{E}[\|\xi^t\|_2^2]<\infty,\mathbf{E}[\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\|_2^2]<\infty,\mathbf{E}[\|\nabla_{\mathbf{v}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t})\|_2^2]<\infty$*, and let* B *be an upper bound on* $\mathbf{E}[\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\|_2^2]+\mathbf{E}[\|\nabla_{\mathbf{v}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t})\|_2^2]+\mathbf{E}[\|\xi^t\|_2^2]$*.*

By telescoping, if G is used, we have: $\mathbf{E}[\mathbb{F}(\mathbf{w}^{T+1},\mathbf{v}^{T+1})]-\mathbb{F}(\mathbf{w}^{0},\mathbf{v}^{0})=\sum_{k=0}^{T}\left\{\mathbf{E}[\mathbb{F}(\mathbf{w}^{k+1},\mathbf{v}^{k+1})]-\mathbf{E}[\mathbb{F}(\mathbf{w}^{k},\mathbf{v}^{k})]\right\}\leq-\sum_{k=0}^{T}\alpha_k\mathbf{E}[\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{k},\mathbf{v}^{k})\|_2^2]-\sum_{k=0}^{T}\frac{\alpha_k b}{n}\mathbf{E}[\|\nabla_{\mathbf{v}}\mathbb{F}(\mathbf{w}^{k+1},\mathbf{v}^{k})\|_2^2]+\frac{LB}{2}\sum_{k=0}^{T}\alpha_k^2$*. Taking the limit* $T\to\infty$ *of both sides, we obtain:*

$$\begin{aligned}
&-\sum_{k=0}^{\infty}\alpha_k\mathbf{E}[\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{k},\mathbf{v}^{k})\|_2^2]-\frac{b}{n}\sum_{k=0}^{\infty}\alpha_k\mathbf{E}[\|\nabla_{\mathbf{v}}\mathbb{F}(\mathbf{w}^{k+1},\mathbf{v}^{k})\|_2^2]+\frac{LB}{2}\sum_{k=0}^{\infty}\alpha_k^2\\
&\geq\lim_{T\to\infty}\mathbf{E}[\mathbb{F}(\mathbf{w}^{T+1},\mathbf{v}^{T+1})]-\mathbf{E}[\mathbb{F}(\mathbf{w}^{0},\mathbf{v}^{0})]\geq\min_{\mathbf{w},\mathbf{v}}\mathbb{F}(\mathbf{w},\mathbf{v})]-\mathbb{F}(\mathbf{w}^{0},\mathbf{v}^{0})>-\infty.
\end{aligned}$$

Since $\sum_{k=0}^{\infty}\alpha_k^2 < \infty$*, the above inequality immediately implies that* $\sum_{k=0}^{\infty}\alpha_k\mathbf{E}[\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^k,\mathbf{v}^k)\|_2^2] < \infty$ *and* $\sum_{k=0}^{\infty}\alpha_k\mathbf{E}[\|\nabla_{\mathbf{v}}\mathbb{F}(\mathbf{w}^{k+1},\mathbf{v}^k)\|_2^2] < \infty$.

If G *is not used, then by a similar argument, we have* $\sum_{k=0}^{\infty}\alpha_k\mathbf{E}[\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^k,\mathbf{v}^k)\|_2^2] < \infty$.

By Lemma A.5 in (Mairal, 2013), to show $\lim_{k\to\infty}\mathbf{E}[\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^k,\mathbf{v}^k)\|_2^2] = 0$*, since* α_k *is not summable, it suffices to show* $\left|\mathbf{E}[\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{k+1},\mathbf{v}^{k+1})\|_2^2] - \mathbf{E}[\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^k,\mathbf{v}^k)\|_2^2]\right| \le C\alpha_k$ *for some constant* C*. To do so, we first recall a useful fact: for any two vectors* $\mathbf{a},\mathbf{b}$ *and any finite-dimensional vector norm* $\|\cdot\|$*,*

$$\Big|(\|a\|+\|b\|)(\|a\|-\|b\|)\Big| \le \|a+b\|\|a-b\|. \tag{17}$$

We have:

$$\begin{aligned}
&\left|\mathbf{E}[\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t+1})\|_2^2] - \mathbf{E}[\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\|_2^2]\right| \\
&= \left|\mathbf{E}\Big[\big(\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t+1})\|_2 + \|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\|_2\big)\big(\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t+1})\|_2 - \|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\|_2\big)\Big]\right| \\
&\le \mathbf{E}\left[\Big|\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t+1})\|_2 + \|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\|_2\Big| \cdot \Big|\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t+1})\|_2 - \|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\|_2\Big|\right] \\
&\le \mathbf{E}\left[\Big\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t+1}) + \|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\Big\|_2 \cdot \Big\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t+1})\|_2 - \|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\|_2\Big\|\right] \\
&\le 2LB\mathbf{E}\Big[\|(\mathbf{w}^{t+1},\mathbf{v}^{t+1})_-(\mathbf{w}^{t},\mathbf{v}^{t})\|_2\Big] \\
&\le 2LB\alpha_t\mathbf{E}\left[\Big\|\Big(\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t}) + \xi^k, e^t\otimes\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t})\Big)\Big\|_2\right] \\
&= 2LB\alpha_t\mathbf{E}\left[\sqrt{\Big\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t}) + \xi^k\Big\|_2^2} + \sqrt{\Big\|e^t\otimes\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t})\Big\|_2^2}\right] \\
&\le 2LB\alpha_t\sqrt{\mathbf{E}\left[\Big\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t}) + \xi^k\Big\|_2^2\right] + \mathbf{E}\left[\Big\|e^t\otimes\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t})\Big\|_2^2\right]} \\
&\le 2LB\alpha_k\sqrt{2\mathbf{E}\left[\Big\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t},\mathbf{v}^{t})\Big\|_2^2\right] + 2\mathbf{E}\left[\Big\|\xi^t\Big\|_2^2\right] + \mathbf{E}\left[\Big\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^{t+1},\mathbf{v}^{t})\Big\|_2^2\right]} \\
&\le 2LB\alpha_t\sqrt{5B^2} = 2\sqrt{5}B^2L\alpha_t,
\end{aligned}$$

where the first inequality is an application of Jensen's inequality, the second inequality follows from Equation (17) *the third inequality follows from Lipschitz continuity and the sixth inequality follows from another application of Jensen's inequality. Consequently, by Lemma and the above chain of inequalities, it follows that* $\lim_{t\to\infty}\mathbf{E}[\|\nabla_{\mathbf{w}}\mathbb{F}(\mathbf{w}^t,\mathbf{v}^t)\|_2^2] = 0$*. Note that if* G *is used, then by a similar argument, it follows that* $\lim_{t\to\infty}\mathbf{E}[\|\nabla_{\mathbf{v}}\mathbb{F}(\mathbf{w}^t,\mathbf{v}^t)\|_2^2] = 0$*. Consequently, if* G *is used, then* $\lim_{t\to\infty}\mathbf{E}[\|\nabla\mathbb{F}(\mathbf{w}^t,\mathbf{v}^t)\|_2^2] = 0$.

C. Proof of Proposition 1

Proposition 1 *Suppose* $(\mathbf{x},y)$ *denotes a training sample and its corrupted label. For simplicity, let the MentorNet input* $\phi(\mathbf{x},y,\mathbf{w}) = \ell$ *be the loss computed by the StudentNet model parameter* $\mathbf{w}$*. The MentorNet* $g_m(\ell;\Theta) = v$*, where* v *is the sample weight. If* g_m *decreases with respect to* ℓ*, then there exists an underlying robust objective* F*:*

$$F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}\rho(\ell_i),$$

where $\rho(\ell_i) = \int_0^{\ell_i} g_m(x;\Theta)dx$*. In the special cases,* $\rho(\ell)$ *degenerates to the classical robust M-estimator: Huber (Huber et al., 1964) and the log-sum penalty (Candes et al., 2008).*

Proof 2 *Given a MentorNet g_m and its fixed parameter Θ. As $\phi(\mathbf{x}, y, \mathbf{w}) = \ell$, we have $g_m(\phi(\mathbf{x}, y, \mathbf{w}); \Theta) = g_m(\ell; \Theta)$.*

g_m is a neural network and hence is continuous with respect to ℓ. Define $\rho(\ell)$ as

$$\rho(\ell) = \int_0^{\ell} g_m(x; \Theta) dx \tag{18}$$

Given the condition in the proposition, g_m is decreasing with respect to ℓ. The function ρ is then bounded by its 1st term of the Taylor series about a point $\mathbf{w}^*$. We have:

$$\rho(\phi(\mathbf{x}, y, \mathbf{w})) \leq \rho(\phi(\mathbf{x}, y, \mathbf{w}^*)) + g_m(\phi(\mathbf{x}, y, \mathbf{w}^*); \Theta)(\phi(\mathbf{x}, y, \mathbf{w}) - \phi(\mathbf{x}, y, \mathbf{w}^*)) \tag{19}$$

According to (Meng et al., 2015), the right-hand side in Eq. (19) *is a tractable surrogate for $\rho(\phi(\mathbf{x}, y, \mathbf{w}))$ and there exists an underlying robust objective. For the i-th sample, we have:*

$$\rho(\phi(\mathbf{x}_i, y_i, \mathbf{w})) = \rho(\ell_i) = \int_0^{\ell_i} g_m(x; \Theta) dx \tag{20}$$

Finally, we have a robust objective derived from:

$$F(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \rho(\phi(\mathbf{x}_i, y_i, \mathbf{w})) = \frac{1}{n} \sum_{i=1}^{n} \rho(\ell_i) \tag{21}$$

Now we show the connection between Eq. (20) *to the robust M-estimator. For simplicity, we assume that the loss $\ell \geq 0$ is non-negative for every training sample. For the Huber loss, there exists an Θ^* such that:*

$$g_m(\ell; \Theta^*) = \begin{cases} \frac{1}{2} & \ell \leq \lambda^2 \\ \frac{\lambda}{2\sqrt{\ell}} & \textit{otherwise} \end{cases}, \tag{22}$$

where $\lambda > 0$. It is easy to verify g_m is decreasing with respect to ℓ, and we have:

$$\rho(\ell) = \int_0^{\ell} g_m(x; \Theta^*) dx = \begin{cases} \frac{1}{2}\ell & \ell \leq \lambda^2 \\ \lambda(\sqrt{\ell} - \frac{1}{2}\lambda) & \textit{otherwise} \end{cases} \tag{23}$$

Therefore we have:

$$\rho(\ell^2) = \begin{cases} \frac{1}{2}\ell^2 & \ell \leq \lambda^2 \\ \lambda(\ell - \frac{1}{2}\lambda) & \textit{otherwise} \end{cases} \tag{24}$$

Eq. (24) *has a similar form of the Huber M-estimator (Huber et al., 1964).*

Likewise, there exists an Θ^* and a positive ϵ such that

$$g_m(\ell; \Theta^*) = \frac{\lambda}{\ell + \epsilon} \tag{25}$$

Its underlying objective is identical to the log-sum penalty (Candes et al., 2008):

$$\rho(\ell) = \int_0^{\ell} g_m(x; \Theta^*) dx = \lambda \log(\ell + \epsilon) - \lambda \log(\epsilon) = \lambda \log(1 + \frac{\ell}{\epsilon}) \tag{26}$$

It leads to the following log-sum penalty:

$$F(\mathbf{w}) = \lambda \frac{1}{n} \sum_{i=1}^{n} \log(1 + \frac{\ell_i}{\epsilon}) \tag{27}$$

For the Lorentzian (Black & Anandan, 1996). We have

$$g_m(\ell;\Theta^*) = \frac{2\ell}{2\delta^2+\ell^2} \tag{28}$$

In special cases, we assume that δ is a small positive such that all sample loss $\ell \geq \sqrt{2}\delta$, the underlying objective is

$$\rho(\ell) = \int_0^{\ell} g_m(x;\Theta^*)dx = \log(2\delta^2+\ell^2) - \log 2\delta^2 = \log(1+\frac{1}{2}(\frac{\ell}{\delta})^2) \tag{29}$$

The above objective becomes Lorentzian (Black & Anandan, 1996), also known as Lorentzian/Cauchy.

D. Comparison on MentorNet Architectures

This section examines *MentorNet*'s architecture in terms of learning existing predefined curriculums (or weighting scheme). To generate the data for training MentorNet, we enumerate the feasible input space of $\mathbf{z}_i$ including the loss, difference to the loss moving average, label, and epoch percentage. For this experiment, the dataset contains a total of 300k samples. For each sample, we label a weight according to the weighting scheme in the predefined curriculum. For example, we compute the weight for focal loss by

$$v_i^* = [1-\exp\{-\ell_i\}]^{\gamma}, \tag{30}$$

where γ is a hyperparameter for smoothing the distribution. We consider the following predefined curriculums: self-paced (Kumar et al., 2010), hard negative mining (Felzenszwalb et al., 2010), linear weighting (Jiang et al., 2015), focal loss (Lin et al., 2017), and prediction variance (Chang et al., 2017). Besides, we also consider a temporal mixture weighting which is a mix of the self-paced (when the epoch percentage < 50) and the hard negative mining (when the epoch percentage ≥ 50). In some curriculums, the form G is unknown. We directly minimize the mean squared error between the MentorNet's output and the true weight.

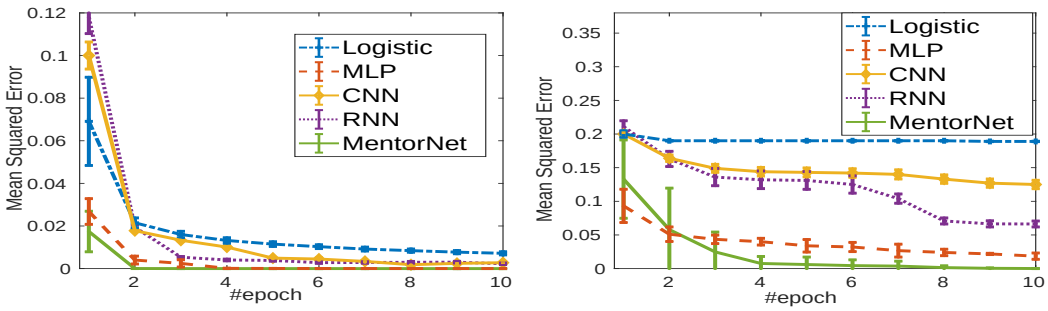
We compare the MentorNet architecture in Fig. 1 in the paper to a simple logistic regression and 3 classical networks: a 2-layer Multiple Layer Perceptron (MLP), a 2-layer CNN with mean pooling, and an LSTM network (RNN) to sequentially encode the features of every example in a mini-batch. The same features are used across all networks. The objective is minimized by the Adam optimizer (Kingma & Ba, 2015). We use a very simple network for the proposed MentorNet. The bidirectional LSTM has a single layer of 10 base LSTM units (step size = 10). We use an embedding layer (size = 2) for the labels and an embedding layer (size = 5) for the integer epoch percentage between 0 and 99. The fully connected layer fc_1 uses the tangent activation function and has 20 hidden nodes.

The performance is evaluated by the Mean Squared Error (MSE) to the true weight produced by each curriculum. Each experiment is repeated 5 times using random starting values, and the average MSE (with the 90% confidence interval) is reported. Table 1 shows the comparison results. As we see, the simple network structure MLP performs well except for complex weighting schemes that are prediction variance (Chang et al., 2017) and Temporal mixture weighting. Nevertheless, the bi-LSTM structure in Fig.1 of the paper performs better than other classical network architectures. Fig. 1 illustrates the error curve of different MentorNet architecture during training, where the x-axis is the training epoch and y-axis is the MSE.

Table 1. The MSE comparison of different MentorNet architecture on predefined curriculums.

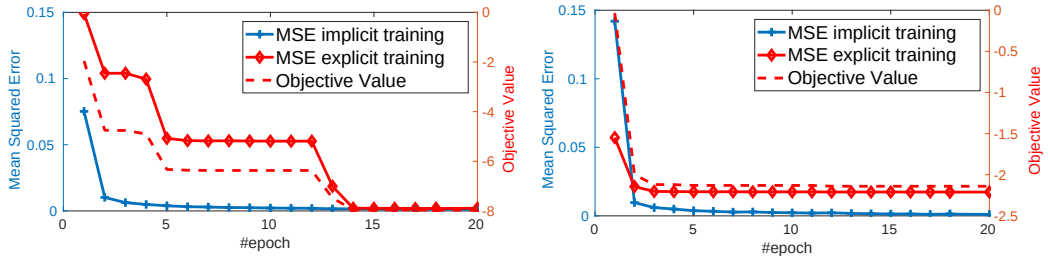
Weighting Scheme	Logistic	MLP	CNN	RNN	MentorNet (bi-LSTM)
Self-paced (Kumar et al., 2010)	8.9±0.8E-3	1.1±0.3E-5	4.9±1.0E-2	1.6±1.0E-2	**1.6±0.5E-6**
Hard negative mining (Felzenszwalb et al., 2010)	7.1±0.7E-3	1.6±0.6E-5	2.7±0.6E-3	2.2±0.4E-3	**6.6±4.5E-7**
Linear weighting (Jiang et al., 2015)	9.2±0.1E-4	1.2±0.4E-4	1.1±0.2E-4	2.0±0.3E-2	**4.4±1.3E-5**
Prediction variance (Chang et al., 2017)	6.8±0.1E-3	4.0±0.1E-3	2.8±0.4E-2	6.2±0.2E-3	**1.4±0.7E-3**
Focal loss (Lin et al., 2017)	1.7±0.0E-3	6.0±3.5E-5	1.2±0.3E-2	1.2±0.3E-2	**1.5±0.8E-5**
Temporal mixture weighting	1.8±0.0E-1	1.9±0.4E-2	1.2±0.6E-1	6.6±0.4E-2	**1.2±1.1E-4**

In Table 1, we learn a MentorNet by minimizing the MSE between its output and the true weight. We call it implicit training. When the form of G is known, we can learn a MentorNet by directly minimizing Eq.(4) in the paper, called explicit training. These two training approaches are theoretically identical and we empirically compare them on two curriculums of known G in Fig. 2. As shown, we found implicit training converges faster.



(a) Hard negative mining (b) Temporal mixture weighting

Figure 1. Comparison of explicit and implicit MentorNet training.



(a) Self-paced (b) Linear weighting

Figure 2. Convergence comparison of different MentorNet architectures.

E. Implementation Details

E.1. Dataset and StudentNet

CIFAR-10 and CIFAR-100 (Krizhevsky & Hinton, 2009) consist of 32 $\times$ 32 color images arranged in 10 and 100 classes. Both datasets contain 50,000 training and 10,000 validation images. The inception (Szegedy et al., 2016) and wide-resnet-101 (He et al., 2016; Zagoruyko & Komodakis, 2016) are used as the StudentNet. Their implementations are based on the TensorFlow slim implementation[1], where the inception is detailed in (Zhang et al., 2017) and the wider resnet is in (Zagoruyko & Komodakis, 2016).

In training, the batch size is set to 128 and we train 39K iterations for resnet model and 120k for the inception model. The Step 12 in Algorithm 1 in the paper is implemented by the momentum SGD optimizer (momentum = 0.9) on a single GPU. We use the common learning rate scheduling strategy and set the starting learning rate as 0.1 and multiply it by a factor of 0.1 at the 19.5k, 25k and 30k iteration for the resnet model, and 78k for the inception model. Training in this way on the clean training dataset, the validation accuracy reaches about 81.4% and 95.5% for inception and resnet-101 on CIFAR-10, and about 49.2% and 78.8% on CIFAR-100.

By default, the StudentNet incorporates three types of regularization: 1) the weight decay which adds an l_2 norm of the model parameters into the learning objective; 2) data augmentation (Krizhevsky et al., 2012) which augments the training images via domain-specific transformations random cropping, perturbation and contrast; 3) dropout (Srivastava et al., 2014) masks out network fully-connected layer outputs randomly. We use the best hyperparameter found on the clean training data. In the inception network, weight decay is set $4e{-}3$ and the dropout keep probability is 0.5. In the resnet-101, weight decay is $2e{-}4$ and the dropout keep probability is 1.0. In the Forgetting baseline (Arpit et al., 2017), we search the dropout parameter in the range of (0.2-0.9) and report the best accuracy on the validation set. The data augmentation is the same for the inception and the resnet-101: we pad 4 pixels to each side and randomly sampling a 32 x 32 crop, randomly flip the image horizontally (left to right), and then linearly scale the image to have zero mean and unit norm. Unless specified otherwise, the StudentNet of the same hyperparameter, discussed above, is used in all baseline and the proposed model.

ImageNet ILSVRC2012 (Deng et al., 2009) contains about 1.2 million training and 50k validation images, split into 1,000 classes. Each image is resized to 299 $\times$ 299 with 3 color channels. For the ImageNet, we use the inception-resnet v2 slim

[1]https://github.com/tensorflow/models/tree/master/research/slim

implementation[2] as our StudentNet. We train the model on the ImageNet of 40% noise. The Step 12 in Algorithm 1 in the paper is implemented as a distributed asynchronized momentum SGD optimizer (momentum = 0.9) on 50 GPUs. We set the batch size to 32 and train the model until it converges. That is 500 thousand steps for the StudentNet without any regularization and 1 million steps for the StudentNet with full regularization (weight decay, dropout and data augmentation). The starting learning rate is 0.05 and is decreased by a factor of 0.1 every 10 training epochs. The default dropout-keep-probability hyperparameter set to 0.8. In the Forgetting baseline, we set it to 0.2, which is the best parameter found on CIFAR-100. The weight decay is $4e{-}5$. The batch norm is used and its decay is set to 0.9997 and the epsilon is set to 0.001. The default data augmentation in the slim implementation is used. Training in this way on the clean training dataset, the validation accuracy is Hit@1=0.765.

E.2. Baselines

Regarding the baseline method. FullMode is the standard StudentNet trained using l_2 weight decay, dropout (Srivastava et al., 2014) and data augmentation (Krizhevsky et al., 2012). The same parameters discussed in Appendix E.1 are used. Forgetting was introduced in (Arpit et al., 2017). It is same as the FullModel except that the dropout parameter is tuned in the range of (0.2-0.9). For the self-paced learning (Kumar et al., 2010), we gradually increase λ in training by 20%. Following (Jiang et al., 2014) we tune the parameter in the range of the 50th, 60th and 75th percentile of average sample loss. For the focal loss (Lin et al., 2017), we tune its γ in the range of $\{1, 2, 3\}$ in our experiments. It is easy to verify that Eq. (30) leads to the same classification objective in the focal loss (Lin et al., 2017), an award-winning method for object detection. For the Reed (Reed et al., 2014), we implement two versions: the soft and hard version. Let $q = [q_1, \ldots, q_m]$ be the prediction (logits after softmax) for m classes:

$$\ell_i = -(\beta \sum_{j=1}^{m} \mathbb{1}(y_i = j) \log(q_j) + (1 - \beta) \sum_{j=1}^{m} q_j \log(q_j)), \tag{31}$$

where $\mathbb{1}$ is the indicator function and a hard version:

$$\ell_i = -(\beta \sum_{j=1}^{m} \mathbb{1}(y_i = j) \log(q_j) + (1 - \beta) \max_j \log(q_j)), \tag{32}$$

We tune the parameters β in the range of $\{0.7, 0.8, 0.9, 0.95\}$. Goldberger (Goldberger & Ben-Reuven, 2017) is a recent baseline weakly-supervised learning method. We implement the S-Model in the paper by appending an additional layer to the StudentNet.

E.3. Setups of Our Model

The details about the MentorNet architecture (Fig. 1 in the paper) are discussed in Appendix D. MentorNet PD is learned using the curriculum in Eq. (5) in the paper. The loss moving average ℓ_{pt} is set to the 75th-percentile loss in a mini-batch. We tune the hyperparameter λ_1 and λ_2. MentorNet DD is the learned *data-driven* curriculum. It is trained on 5,000 images of true labels, randomly sampled from CIFAR-10. We learn MentorNet DD on this dataset and apply it to CIFAR-100 on which no true labels are used. We use the CIFAR-10 subset of the same level of the noise fraction corresponding to the CIFAR-100. CIFAR-10 and CIFAR-100 are two different datasets that have not only different classes but also the different number of classes. As CIFAR-100 and CIFAR-10 have the different number of classes, to apply a MentorNet, we fix the class label to 0.

Algorithm 1 is used to train the MentorNet together with the StudentNet. The decay factor in computing the loss moving average is set to 0.95. As mentioned in the paper, a burn-in process is used in the first 20% training epoch for both MentorNet DD and MentorNet PD. We update and learn MentorNet twice after the learning rate is changed. That is on the 21% and 75% of the total epoch. More updates lead to insignificant performance difference. For each mini-batch, the weight decay parameter θ in Eq. (1) in the paper is normalized by the sum of the weight in a mini-batch. That is $\theta^t = \frac{1}{b}\theta^0 \sum_{i=1}^{b} \mathbf{v}_{\Xi_i}$, where θ^0 is the original weight decay parameter. The same learning rate scheduling strategy in the StudentNet is used in Algorithm 1.

[2] https://github.com/tensorflow/models/blob/master/research/slim/nets/inception_resnet_v2.py

References

Arpit, D., Jastrzkebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. A closer look at memorization in deep networks. In *ICML*, 2017.

Black, M. J. and Anandan, P. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer vision and image understanding*, 63(1):75–104, 1996.

Candes, E. J., Wakin, M. B., and Boyd, S. P. Enhancing sparsity by reweighted l1 minimization. *Journal of Fourier analysis and applications*, 14(5-6):877–905, 2008.

Chang, H.-S., Learned-Miller, E., and McCallum, A. Active bias: Training a more accurate neural network by emphasizing high variance samples. *NIPS*, 2017.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

Goldberger, J. and Ben-Reuven, E. Training deep neural-networks using a noise adaptation layer. In *ICLR*, 2017.

Gong, P., Zhang, C., Lu, Z., Huang, J. Z., and Ye, J. A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In *ICML*, 2013.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Huber, P. J. et al. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

Jiang, L., Meng, D., Yu, S.-I., Lan, Z., Shan, S., and Hauptmann, A. Self-paced learning with diversity. In *NIPS*, 2014.

Jiang, L., Meng, D., Zhao, Q., Shan, S., and Hauptmann, A. G. Self-paced curriculum learning. In *AAAI*, 2015.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. 2009.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

Kumar, M. P., Packer, B., and Koller, D. Self-paced learning for latent variable models. In *NIPS*, 2010.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. *ICCV*, 2017.

Mairal, J. Stochastic majorization-minimization algorithms for large-scale optimization. In *NIPS*, 2013.

Meng, D., Zhao, Q., and Jiang, L. What objective does self-paced learning indeed optimize? *arXiv preprint arXiv:1511.06049*, 2015.

Reed, S., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., and Rabinovich, A. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *BMVC*, 2016.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

Zhang, C.-H. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, pp. 894–942, 2010.