Time-Consistent Self-Supervision for Semi-Supervised Learning
(2020)

Tianyi Zhou, Shengjie Wang, Jeff A. Bilmes

Summary

Contributions

Semi-supervised learning (SSL) uses a large number of unlabeled data to enhance the performance of a given model when the labeled training data is insufficient. In such a setting, the model is often trained with combination of supervised and unsupervised losses. The unsupervised loss can consistent of: A normal cross-entropy, where the model's predictions are considered as ground truth if the model is confident enough. A consistency loss, where the model is encouraged to have a local smooth on the data manifold, where an image and its transformed version have similar prediction, and contrastive loss (or a triplet loss), that pushed the representations of different samples away, while two similar inputs (an image and its transformed version) are pulled closer.

However, one problem of such types of losses, is that they depend upon the network's prediction and use them as targets or anchors for the unsupervised loss. If these output are time inconsistent, where the prediction change from one training step to the other, even if the model output confident prediction, they might be erroneous. Such a consistency over time might slow the training or even make diverge.

In this work, the authors propose a **time-consistency** (TC) metric that measure the change in the output probability distribution between consecutive two time steps; t and $t-1$, in terms of distance and confidence, and use such a metric to select stable unlabeled sample to be used in the unsupervised loss.

Method

Time-consistency TC of an individual unlabeled sample x at training step t, $c^t(x)$, is the exponential moving average of individual consistency measures $a^t(x)$ between two consecutive time steps:

$$c^t(x) = \gamma_c(-a^t(x)) + (1-\gamma_c)c^{t-1}(x)$$

with

$$a^t(x) = D_{KL}(p^{t-1}(x)\|p^t(x)) + \left|\log \frac{p^{t-1}_{y^{t-1}(x)}(x)}{p^t_{y^{t-1}(x)}(x)}\right|$$

and $\gamma_c \in [0,1]$ is the discount factor, in $c(t)$. The individual measures are negated (i.e., $-a(x)$) so that larger $c(t)$ will refer to a consistent example x, since $a(x) > 0$ and smaller $a(x)$ means that x is stable between the two consecutive steps. The KL divergence term measure how consistent is the model between two time steps for a given example x, and the second term measure the change in confidence. A sample x is TC if it has both a consistent prediction, and a consistent confidence. The measure $a^t(x)$ was derived by measuring the change in the supervised loss of the model, with and without the a single unlabeled example x. If the added sample x is TC, i.e., with high $c(x)$, the authors showed that the change in the loss will be small with no catastrophic forgetting of learned data will occur.

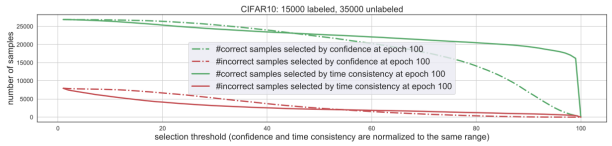

Figure 1. Computed time-consistency and confidence at epoch 100. The x-axis shows the validation samples selected using different thresholds on the two metrics (normalized to $[0, 100]$). The y-axis reports correct v.s. incorrect predictions over the selected samples.

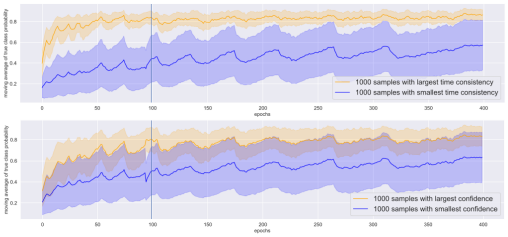

Figure 2. Computed time-consistency (top) and confidence (bottom) at epoch 100. Select the top 1000 and bottom 1000 validation samples based on the two metrics. Compare the moving average of true class probability of the selected samples across epochs.

The two figures above present some empirical evidence on the usefulness of TC. In Figure 1 we see, if large thresholds, use time consistency as a criterion to select unlabeled example results in a higher number of true positives, where the selected samples have the correct label. Additionally, in Figure 2, if we select the top and bottom 1000 samples based, we see the time consistency is quite stable across training time for the top 1000, and above a given epoch (e.g., epoch 100), it stays the same, indicating that the top 1000 samples are consistent.

TC-SSL loss For training, the authors propose to use a combination of supervised losses, contrastive loss and consistency loss with a mean teacher to generate the pseudo-targets, in addition to TC to select the consistency unlabeled samples over time. Given an augmentation function $G(.)$, a network $f(.)$ and its exponentially moving average $\overline{f}(.)$. The losses used are as follows:

- Cross-entropy loss for labeled examples: $\ell_{ce}(x, y; \theta) = -\log \frac{\exp(f(G(x);\theta)[y])}{\sum_{y'} \exp(f(G(x);\theta)[y'])}$
- Cross-entropy loss for labeled examples using pseudo-labels: $\ell^t_{ce}(x; \theta) = -\log \frac{\exp\left(f(G(x);\theta)\left[y^{t-1}(x)\right]\right)}{\sum_{y'} \exp(f(G(x);\theta)[y'])}$
- Consistency loss: $\ell^t_{cs}(x; \theta) = \|f(G(x); \theta) - \overline{f^t}(x)\|_p$
- Contrastive loss: $\ell^t_{ct}(x; \theta) = -\log \frac{\exp\left(\cos\left[f(G(x);\theta), \overline{f^t}\left(G'(x)\right)\right]\right)}{\sum_{z \in \{G'(x)\} \cup D} \exp(\cos[f(G(x);\theta), \overline{f^t}(z)])}$

For the unsupervised losses, the unsupervised samples to be used are updated at a given step, where we chose the top k most consistent example which is gradually increased, but at the start, the authors propose a weighted sampling for more exploration, and later use top k for more exploitation. During the beginning of training, we only train using the supervised loss, and keep accumulating the TC metric $c(x)$, where $a(x)$ are also computed with learning rate calibration to reflect the parameter change in the TC metric, so if the model took a big step, $a(x)$ is also increased. To summarize, the TC-SSL algorithm is as follows:

Algorithm 1 Time-Consistent SSL (TC-SSL)

input: $\mathcal{U}, \mathcal{L}, \pi(\cdot; \eta), \eta^{1:T}, f(\cdot; \theta), G(\cdot)$;
hyperparameters: $T_0, T, \lambda_{cs}, \lambda_{ct}, \lambda_{ce}, \gamma_\theta, \gamma_c, \gamma_k$;
initialize: θ^0, k^1;
for $t \in \{1, \cdots, T\}$ **do**
  if $t \leq T_0$ **then**
    $\theta^t \leftarrow \theta^{t-1} + \pi\left(\sum_{(x,y)\in\mathcal{L}} \nabla_\theta \ell_{ce}(x, y; \theta^{t-1}); \eta^t\right)$
  else
    $S^t = \text{argmax}_{S:S\subseteq\mathcal{U},|S|=k^t} \sum_{x\in S} c^t(x)$ or
    Draw k^t samples from $\Pr(x \in S^t) \propto \exp(c^t(x))$;
    $\theta^t \leftarrow \theta^{t-1} + \pi\left(\nabla_\theta L^t(\theta^{t-1}); \eta^t\right)$ (ref. Eq. (11));
  end if
  $p^t(x) \leftarrow \frac{\exp(f(x;\theta^t)[y])}{\sum_{y'=1}^{C} \exp(f(x;\theta^t)[y'])}$, $\forall y \in [C], x \in \mathcal{U}$;
  if $t = 1$ **then**
    $\overline{\theta^t} \leftarrow \theta^t, c^t(x) \leftarrow 0, \ \forall x \in \mathcal{U}$
  else
    Compute $a^t(x)$ (ref. Eq (1)), $\forall x \in \mathcal{U}$;
  end if
  $c^{t+1}(x) \leftarrow \gamma_c(-a^t(x)) + (1-\gamma_c)c^{t-1}(x), \ \forall x \in \mathcal{U}$;
  $\overline{\theta^{t+1}} \leftarrow \gamma_\theta \theta^t + (1-\gamma_\theta)\overline{\theta^t}$;
  $k^{t+1} \leftarrow (1+\gamma_k) \times k^t$;
end for

Results

Table 1. Test error rate (mean±variance) of SSL methods training a small WideResNet and a large WideResNet on **CIFAR10**. Baselines: Pseudo Label (Lee, 2013), Π-model (Sajjadi et al., 2016), VAT (Miyato et al., 2019), Mean Teacher (Tarvainen & Valpola, 2017), MixMatch (Berthelot et al., 2019), ReMixMatch (Berthelot et al., 2020).

Benchmark	CIFAR10 (small WideResNet-28-2)				CIFAR10 (large WideResNet-28-135)			
labeled/unlabeled	500/44500	1000/44000	2000/43000	4000/41000	500/44500	1000/44000	2000/43000	4000/41000
Pseudo Label	40.55 ± 1.70	30.91 ± 1.73	21.96 ± 0.42	16.21 ± 0.11	-	-	-	-
Π-model	41.82 ± 1.52	31.53 ± 0.98	23.07 ± 0.66	5.70 ± 0.13	-	-	-	-
VAT	26.11 ± 1.52	18.68 ± 0.40	14.40 ± 0.15	11.05 ± 0.31	-	-	-	-
Mean Teacher	42.01 ± 5.86	17.32 ± 4.00	12.17 ± 0.22	10.36 ± 0.25	-	-	-	-
MixMatch	9.65 ± 0.94	7.75 ± 0.32	7.03 ± 0.15	6.24 ± 0.06	8.44 ± 1.04	7.38 ± 0.63	6.51 ± 0.48	5.12 ± 0.31
ReMixMatch	-	$\mathbf{5.73 \pm 0.16}$	-	$5.14 \pm \mathbf{0.04}$	-	-	-	-
TC-SSL (ours)	$\mathbf{9.14 \pm 0.88}$	6.15 ± 0.23	$\mathbf{5.85 \pm 0.10}$	$\mathbf{5.07} \pm 0.05$	$\mathbf{6.04 \pm 0.39}$	$\mathbf{3.81 \pm 0.19}$	$\mathbf{3.79 \pm 0.21}$	$\mathbf{3.54 \pm 0.06}$

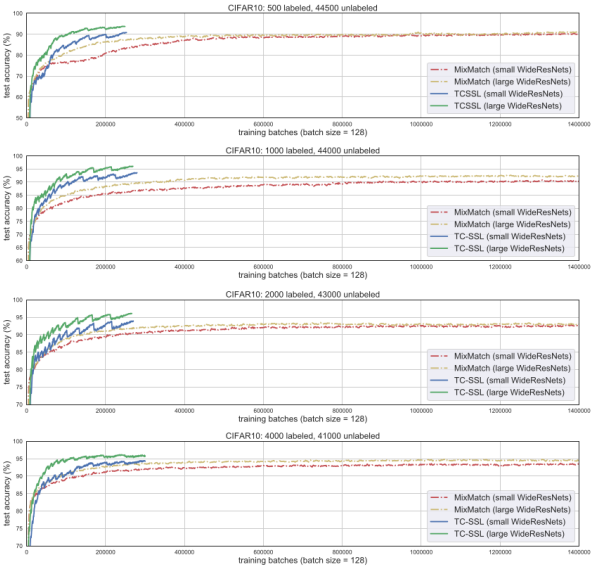

Figure 4. Test accuracy (%) during training small/large WideResNet by MixMatch and TC-SSL on four splittings of CIFAR10.

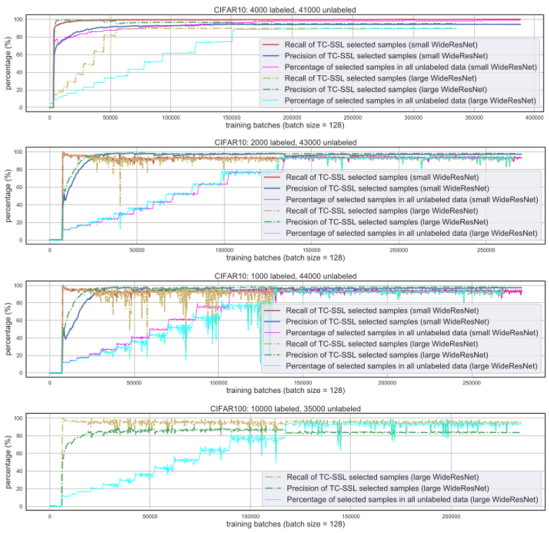

Figure 5. Precision and recall (%) of pseudo labels produced by the model during training for the unlabeled samples selected by TC-SSL. We also provide the percentage (%) of the selected samples in the ground set of unlabeled data $\mathcal{U}$.

Table 4. **Ablation Study:** test error rate (mean±variance) of TC-SSL variants for training WideResNet-28-135 on CIFAR10: "no consistency" removes the consistency loss in Eq. (7); "no contrastive" removes the contrastive loss in Eq. (8); "no PseudoLabel" removes the cross entropy loss for unlabeled data in Eq. (10); "no TC-selection" replaces Line 8-9 of Algorithm 1 with uniform sampling.

labeled/unlabeled	500/44500	1000/44000	2000/43000	4000/41000
TC-SSL (ours)	$\mathbf{6.04 \pm 0.39}$	$\mathbf{3.81 \pm 0.19}$	$\mathbf{3.79 \pm 0.21}$	$\mathbf{3.54 \pm 0.06}$
TC-SSL (no consistency)	7.51 ± 0.56	5.31 ± 0.23	3.82 ± 0.20	$3.58 \pm \mathbf{0.06}$
TC-SSL (no contrastive)	7.56 ± 0.52	5.35 ± 0.28	3.96 ± 0.25	3.66 ± 0.08
TC-SSL (no PseudoLabel)	41.05 ± 2.32	23.64 ± 1.17	14.37 ± 0.69	9.87 ± 0.22
TC-SSL (no TC-selection)	12.25 ± 0.81	6.39 ± 0.44	4.68 ± 0.35	4.05 ± 0.13