MixMatch: A Holistic Approach to Semi-Supervised Learning

2019.06.20
김보섭

Agenda

1. Abstract
2. Introduction & Related Work
3. MixMatch
4. Experiment
5. Conclusion

Abstract

본 논문에서는 기존의 Semi-Supervised Learning에서 활용되는 approach들을 하나로 통합한 방법론을 제시

MixMatch: A Holistic Approach to Semi-Supervised Learning

David Berthelot
Google Research
dberth@google.com

Nicholas Carlini
Google Research
ncarlini@google.com

Ian Goodfellow
Work done at Google
ian-academic@mailfence.com

Avital Oliver
Google Research
avitalo@google.com

Nicolas Papernot
Google Research
papernot@google.com

Colin Raffel
Google Research
craffel@google.com

Abstract

Semi-supervised learning has proven to be a powerful paradigm for leveraging unlabeled data to mitigate the reliance on large labeled datasets. In this work, we unify the current dominant approaches for semi-supervised learning to produce a new algorithm, MixMatch, that works by guessing low-entropy labels for data-augmented unlabeled examples and mixing labeled and unlabeled data using MixUp. We show that MixMatch obtains state-of-the-art results by a large margin across many datasets and labeled data amounts. For example, on CIFAR-10 with 250 labels, we reduce error rate by a factor of 4 (from 38% to 11%) and by a factor of 2 on STL-10. We also demonstrate how MixMatch can help achieve a dramatically better accuracy-privacy trade-off for differential privacy. Finally, we perform an ablation study to tease apart which components of MixMatch are most important for its success.

Introduction & Related Work (1/2)

근래의 SSL (Semi-Supervised Learning) 방법론은 unlabeled data에 대해서 계산된 loss term을 기존 task loss에 추가하는 형태

Semi-supervised learning [6] (SSL) seeks to largely alleviate the need for labeled data by allowing a model to leverage unlabeled data. Many recent approaches for semi-supervised learning add a loss term which is computed on unlabeled data and encourages the model to generalize better to unseen data. In much recent work, this loss term falls into one of three classes (discussed further in Section 2): entropy minimization [17, 28]—which encourages the model to output confident predictions on unlabeled data; consistency regularization—which encourages the model to produce the same output distribution when its inputs are perturbed; and generic regularization—which encourages the model to generalize well and avoid overfitting the training data.

In this paper, we introduce MixMatch, an SSL algorithm which introduces a single loss that gracefully unifies these dominant approaches to semi-supervised learning. Unlike previous methods, MixMatch targets all the properties at once which we find leads to the following benefits:

Introduction & Related Work (2/2)

논문에서 제안하는 MixMatch는 entropy minimization, consistency regularization, generic regularization을 모두 통합하는 방법론

Consistency Regularization

- MixMatch utilizes a form of consistency regularization through the us of standard data augmentation for images (random horizontal flips and crops)

> *Consistency regularization* applies data augmentation to semi-supervised learning by leveraging the idea that a classifier should output the same class distribution for an unlabeled example even after it has been augmented. More formally, consistency regularization enforces that an unlabeled example x should be classified the same as $\text{Augment}(x)$, where $\text{Augment}(x)$ is a stochastic data augmentation function—like a random spatial translation or adding noise.

Entropy Minimization

- MixMatch also implicitly achieves entropy minimization through the use of a "sharpening" function on the target distribution for unlabeled data

> A common underlying assumption in many semi-supervised learning methods is that the classifier's decision boundary should not pass through high-density regions of the marginal data distribution. One way to enforce this is to require that the classifier output low-entropy predictions on unlabeled data. This is done explicitly in [17] by simply adding a loss term which minimizes the entropy of $\text{p}_{\text{model}}(y \mid x; \theta)$ for unlabeled data x. This form of entropy minimization was combined with

Traditional Regularization

- MixMatch utilizes MixUp both as a regularizer (applied to labeled datapoints) and a semi-supervised learning method (applied to unlabeled datapoints)

MixMatch (1/2)

어느 정도 학습된 backbone network에 unlabeled data와 labeled data에 MixMatch를 적용하여, 새로운 MiniBatch를 만들고 아래의 loss를 적용

Algorithm 1 MixMatch ingests a batch of labeled data $\mathcal{X}$ and a batch of unlabeled data $\mathcal{U}$ and produces a collection $\mathcal{X}'$ of processed labeled examples and a collection $\mathcal{U}'$ of processed unlabeled examples with "guessed" labels.

Input: Batch of labeled examples and their one-hot labels $\mathcal{X} = \big((x_b, p_b); b \in (1, \dots, B)\big)$, batch of unlabeled examples $\mathcal{U} = \big(u_b; b \in (1, \dots, B)\big)$, sharpening temperature T, number of augmentations K, Beta distribution parameter α for MixUp.
for $b = 1$ **to** B **do**
  $\hat{x}_b = \text{Augment}(x_b)$ // *Apply data augmentation to* x_b
  for $k = 1$ **to** K **do**
    $\hat{u}_{b,k} = \text{Augment}(u_b)$ // *Apply* k^{th} *round of data augmentation to* u_b
  end for
  $\bar{q}_b = \frac{1}{K} \sum_k \text{p}_{\text{model}}(y \mid \hat{u}_{b,k}; \theta)$ // *Compute average predictions across all augmentations of* u_b
  $q_b = \text{Sharpen}(\bar{q}_b, T)$ // *Apply temperature sharpening to the average prediction (see eq. (7))*
end for
$\hat{\mathcal{X}} = \big((\hat{x}_b, p_b); b \in (1, \dots, B)\big)$ // *Augmented labeled examples and their labels*
$\hat{\mathcal{U}} = \big((\hat{u}_{b,k}, q_b); b \in (1, \dots, B), k \in (1, \dots, K)\big)$ // *Augmented unlabeled examples, guessed labels*
$\mathcal{W} = \text{Shuffle}\big(\text{Concat}(\hat{\mathcal{X}}, \hat{\mathcal{U}})\big)$ // *Combine and shuffle labeled and unlabeled data*
$\mathcal{X}' = \big(\text{MixUp}(\hat{\mathcal{X}}_i, \mathcal{W}_i); i \in (1, \dots, |\hat{\mathcal{X}}|)\big)$ // *Apply* MixUp *to labeled data and entries from* $\mathcal{W}$
$\mathcal{U}' = \big(\text{MixUp}(\hat{\mathcal{U}}_i, \mathcal{W}_{i+|\hat{\mathcal{X}}|}); i \in (1, \dots, |\hat{\mathcal{U}}|)\big)$ // *Apply* MixUp *to unlabeled data and the rest of* $\mathcal{W}$
return $\mathcal{X}', \mathcal{U}'$

$$\mathcal{X}', \mathcal{U}' = \text{MixMatch}(\mathcal{X}, \mathcal{U}, T, K, \alpha)$$

$$\mathcal{L}_{\mathcal{X}} = \frac{1}{|\mathcal{X}'|} \sum_{x,p \in \mathcal{X}'} \text{H}(p, \text{p}_{\text{model}}(y \mid x; \theta))$$

$$\mathcal{L}_{\mathcal{U}} = \frac{1}{L|\mathcal{U}'|} \sum_{u,q \in \mathcal{U}'} \|q - \text{p}_{\text{model}}(y \mid u; \theta)\|_2^2$$

$$\mathcal{L} = \mathcal{L}_{\mathcal{X}} + \lambda_{\mathcal{U}} \mathcal{L}_{\mathcal{U}}$$

practice that most of MixMatch's hyperparameters can be fixed and do not need to be tuned on a per-experiment or per-dataset basis. Specifically, for all experiments we set $T = 0.5$ and $K = 2$. Further, we only change $\lambda_{\mathcal{U}}$ and α on a per-dataset basis; we found that $\lambda_{\mathcal{U}} = 100$ and $\alpha = 0.75$ are good starting points for tuning.

MixMatch (2/2)

어느 정도 학습된 backbone network에 unlabeled data와 labeled data에 MixMatch를 적용하여, 새로운 MiniBatch를 만들고 아래의 loss를 적용

Algorithm 1 MixMatch ingests a batch of labeled data $\mathcal{X}$ and a batch of unlabeled data $\mathcal{U}$ and produces a collection $\mathcal{X}'$ of processed labeled examples and a collection $\mathcal{U}'$ of processed unlabeled examples with "guessed" labels.

Input: Batch of labeled examples and their one-hot labels $\mathcal{X} = \big((x_b, p_b); b \in (1, \dots, B)\big)$, batch of unlabeled examples $\mathcal{U} = \big(u_b; b \in (1, \dots, B)\big)$, sharpening temperature T, number of augmentations K, Beta distribution parameter α for MixUp.

for $b = 1$ **to** B **do**

  $\hat{x}_b = \text{Augment}(x_b)$ // *Apply data augmentation to* x_b

  for $k = 1$ **to** K **do**

    $\hat{u}_{b,k} = \text{Augment}(u_b)$ // *Apply* k^{th} *round of data augmentation to* u_b

  end for

  $\bar{q}_b = \frac{1}{K} \sum_k \text{p}_{\text{model}}(y \mid \hat{u}_{b,k}; \theta)$ // *Compute average predictions across all augmentations of* u_b

  $q_b = \text{Sharpen}(\bar{q}_b, T)$ // *Apply temperature sharpening to the average prediction (see eq. (7))*

end for

$\hat{\mathcal{X}} = \big((\hat{x}_b, p_b); b \in (1, \dots, B)\big)$ // *Augmented labeled examples and their labels*

$\hat{\mathcal{U}} = \big((\hat{u}_{b,k}, q_b); b \in (1, \dots, B), k \in (1, \dots, K)\big)$ // *Augmented unlabeled examples, guessed labels*

$\mathcal{W} = \text{Shuffle}\big(\text{Concat}(\hat{\mathcal{X}}, \hat{\mathcal{U}})\big)$ // *Combine and shuffle labeled and unlabeled data*

$\mathcal{X}' = \big(\text{MixUp}(\hat{\mathcal{X}}_i, \mathcal{W}_i); i \in (1, \dots, |\hat{\mathcal{X}}|)\big)$ // *Apply* MixUp *to labeled data and entries from* $\mathcal{W}$

$\mathcal{U}' = \big(\text{MixUp}(\hat{\mathcal{U}}_i, \mathcal{W}_{i+|\hat{\mathcal{X}}|}); i \in (1, \dots, |\hat{\mathcal{U}}|)\big)$ // *Apply* MixUp *to unlabeled data and the rest of* $\mathcal{W}$

return $\mathcal{X}', \mathcal{U}'$

For consistency regularization
Data Augmentation
Label Guessing:

For entropy minimization

$$\text{Sharpen}(p, T)_i := p_i^{\frac{1}{T}} \Big/ \sum_{j=1}^{L} p_j^{\frac{1}{T}}$$

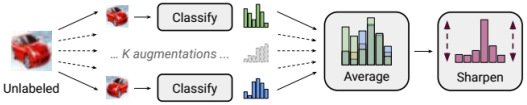

Figure 1: Diagram of the label guessing process used in MixMatch. Stochastic data augmentation is applied to an unlabeled image K times, and each augmented image is fed through the classifier. Then, the average of these K predictions is "sharpened" by adjusting the distribution's temperature. See algorithm 1 for a full description.

For generic regularization
MixUp

$$\lambda \sim \text{Beta}(\alpha, \alpha)$$
$$\lambda' = \max(\lambda, 1 - \lambda)$$
$$x' = \lambda' x_1 + (1 - \lambda') x_2$$
$$p' = \lambda' p_1 + (1 - \lambda') p_2$$

Experiments (1/2)

기존의 방법론들을 능가하는 성능, 특히 labeled data를 많이 취득하는 것보다 unlabeled data를 많이 취득하여 성능을 끌어올리는 것이 가능한 시나리오임을 보임

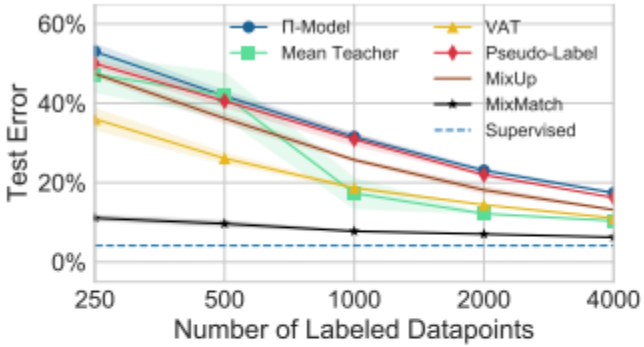

Figure 2: Error rate comparison of MixMatch to baseline methods on CIFAR-10 for a varying number of labels. Exact numbers are provided in table 5 (appendix). "Supervised" refers to training with all 50000 training examples and no unlabeled data. With 250 labels MixMatch reaches an error rate comparable to next-best method's performance with 4000 labels.

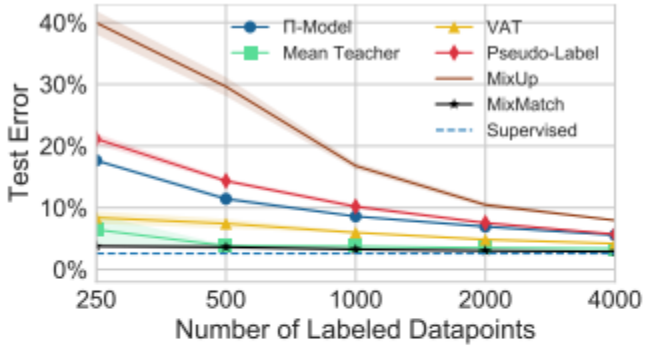

Figure 3: Error rate comparison of MixMatch to baseline methods on SVHN for a varying number of labels. Exact numbers are provided in table 6 (appendix). "Supervised" refers to training with all 73257 training examples and no unlabeled data. With 250 examples MixMatch nearly reaches the accuracy of supervised training for this model.

Labels	250	500	1000	2000	4000	All
SVHN	3.78 ± 0.26	3.64 ± 0.46	3.27 ± 0.31	3.04 ± 0.13	2.89 ± 0.06	2.59
SVHN+Extra	2.22 ± 0.08	2.17 ± 0.07	2.18 ± 0.06	2.12 ± 0.03	2.07 ± 0.05	1.71

Table 3: Comparison of error rates for SVHN and SVHN+Extra for MixMatch. The last column ("All") contains the fully-supervised performance with all labels in the corresponding training set.

Experiments (2/2)

ablation study를 통해서 MixMatch의 기법을 구성하는 요소 중, 특히 unlabeled data간의 MixUp이 중요함을 확인

Ablation	250 labels	4000 labels
MixMatch	11.80	6.00
MixMatch without distribution averaging ($K = 1$)	17.09	8.06
MixMatch without temperature sharpening ($T = 1$)	27.83	10.59
MixMatch with parameter EMA	11.86	6.47
MixMatch without MixUp	39.11	10.97
MixMatch with MixUp on labeled only	32.16	9.22
MixMatch with MixUp on unlabeled only	12.35	6.83
MixMatch with MixUp on separate labeled and unlabeled	12.26	6.50
Interpolation Consistency Training [44]	38.60	6.81

Table 4: Ablation study results. All values are error rates on CIFAR-10 with 250 or 4000 labels. ICT uses EMA parameters and unlabeled mixup and no sharpening.

Conclusion

5 Conclusion

We introduced MixMatch, a semi-supervised learning method which combines ideas and components from the current dominant paradigms for semi-supervised learning. Through extensive experiments on semi-supervised and privacy-preserving learning, we found that MixMatch exhibited significantly improved performance compared to other methods in all settings we studied, often by a factor of two or more reduction in error rate. In future work, we are interested in incorporating additional ideas from the semi-supervised learning literature into hybrid methods and continuing to explore which components result in effective algorithms. Separately, most modern work on semi-supervised learning algorithms is evaluated on image benchmarks; we are interested in exploring the effectiveness of MixMatch in other domains.

Q & A

감사합니다.