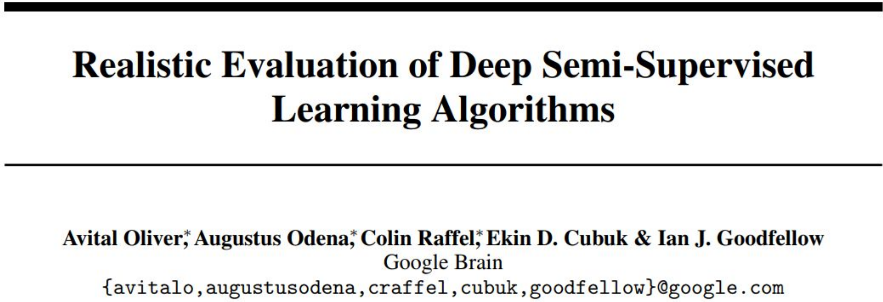
Survey of Semi-supervised Learning (SSL) And Multi-Task Learning (MTL)

Outline

- SSL
- MTL

Google doc: https://docs.google.com/document/d/1LKojD6wUeHtxiyY4ogXW6ZejYikUNnxk-GJ5NHrdIyQ/edit

SSL-overview



Realistic Evaluation of Deep Semi-Supervised Learning Algorithms

Avital Oliver,* Augustus Odena,* Colin Raffel,* Ekin D. Cubuk & Ian J. Goodfellow
Google Brain
{avitalo,augustusodena,craffel,cubuk,goodfellow}@google.com

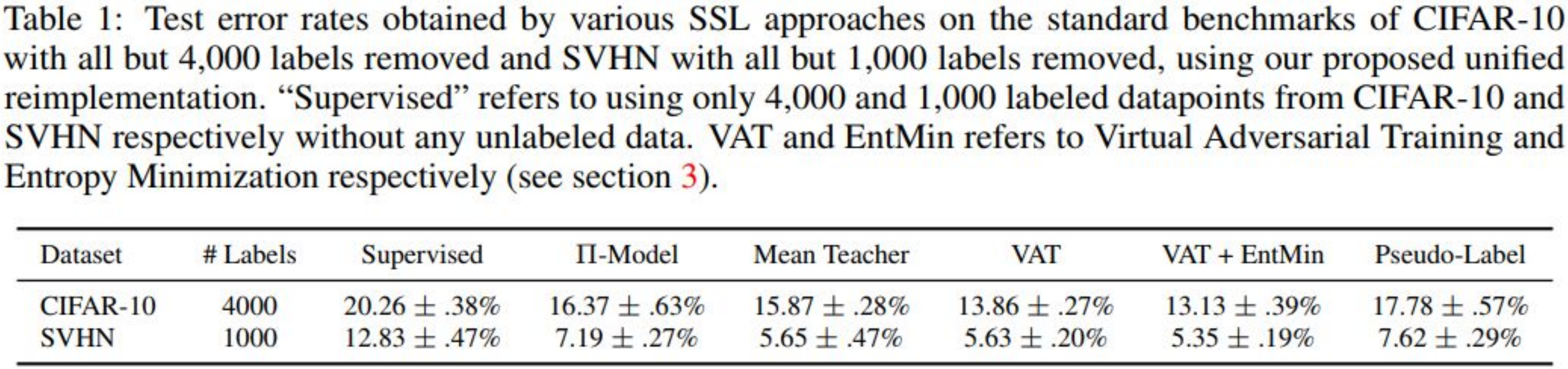


Table 1: Test error rates obtained by various SSL approaches on the standard benchmarks of CIFAR-10 with all but 4,000 labels removed and SVHN with all but 1,000 labels removed, using our proposed unified reimplementation. "Supervised" refers to using only 4,000 and 1,000 labeled datapoints from CIFAR-10 and SVHN respectively without any unlabeled data. VAT and EntMin refers to Virtual Adversarial Training and Entropy Minimization respectively (see section 3).

Dataset	# Labels	Supervised	Π-Model	Mean Teacher	VAT	VAT + EntMin	Pseudo-Label
CIFAR-10	4000	20.26 ± .38%	16.37 ± .63%	15.87 ± .28%	13.86 ± .27%	13.13 ± .39%	17.78 ± .57%
SVHN	1000	12.83 ± .47%	7.19 ± .27%	5.65 ± .47%	5.63 ± .20%	5.35 ± .19%	7.62 ± .29%

SSL overview

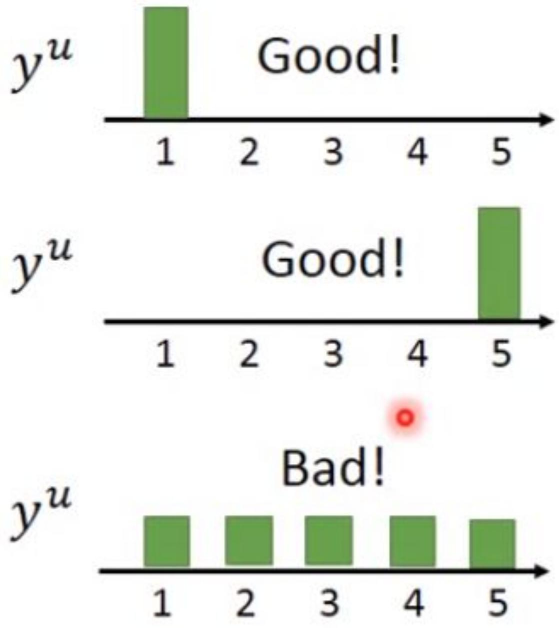
- Semi-supervised learning by entropy minimization. (2004) (EntMin)
- Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. (2013) (Pseudo-label, self-training)
- Label Propagation (LP)
- Temporal Ensembling for Semi-Supervised Learning (ICLR 2017)
- Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results (NIPS 2017) (Mean Teachers)
- Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning (IEEE) (VAT)

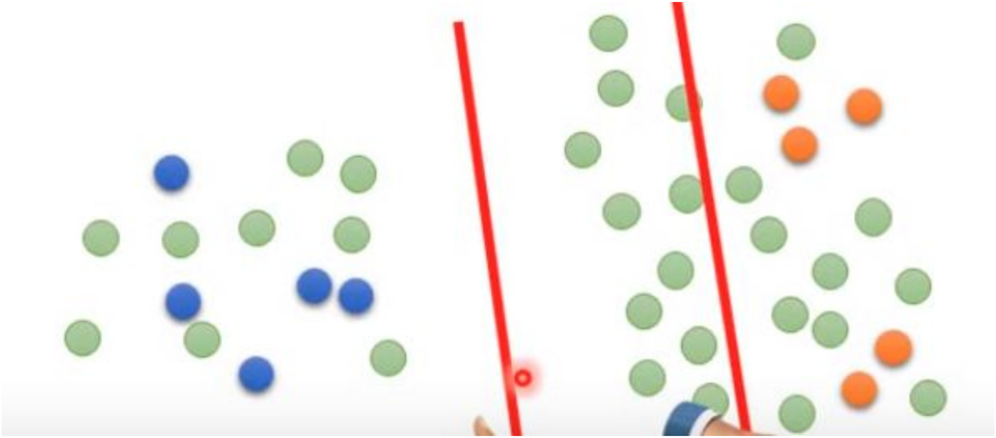
Semi-supervised Learning (SSL)

- Inductive bias / Assumption
  - Low density separation
  - Cluster assumption
  - Smoothness assumption

Entropy Minimization

- Low density separation





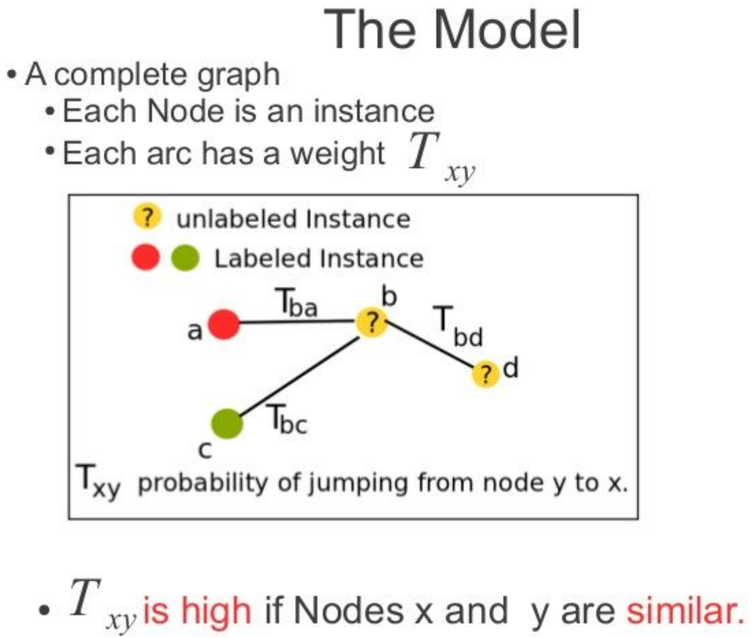
Pseudo-Label (self training, self supervision)

- For unlabeled data, Pseudo-Labels, just picking up the class which has the maximum predicted probability, are used as if they were true labels.
- Equivalent effect of entropy minimization
- Low density separation

Label Propagation graph based method

- Construct graph between data points according to some similarity function
- Propagate label (label are more likely to propagate through similar data points
- Train and propagate until convergence
- Smoothness assumption

(“similar” x has “Similar” y)



The Model

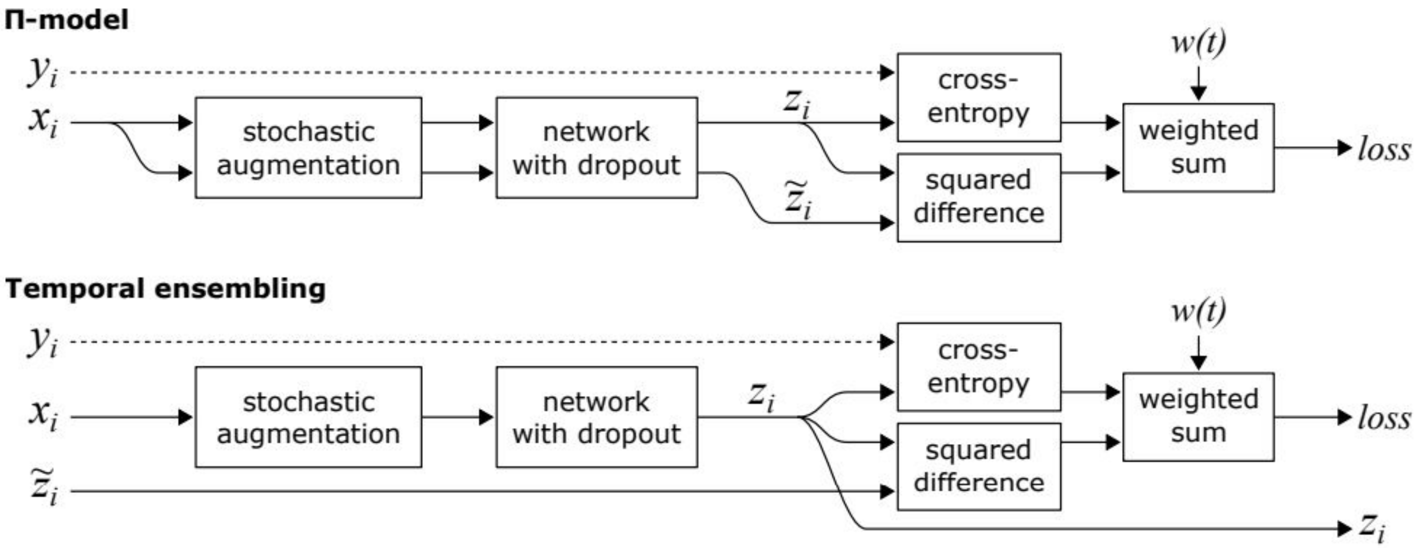
- A complete graph
  - Each Node is an instance
  - Each arc has a weight T_{xy}

T_{xy} probability of jumping from node y to x.

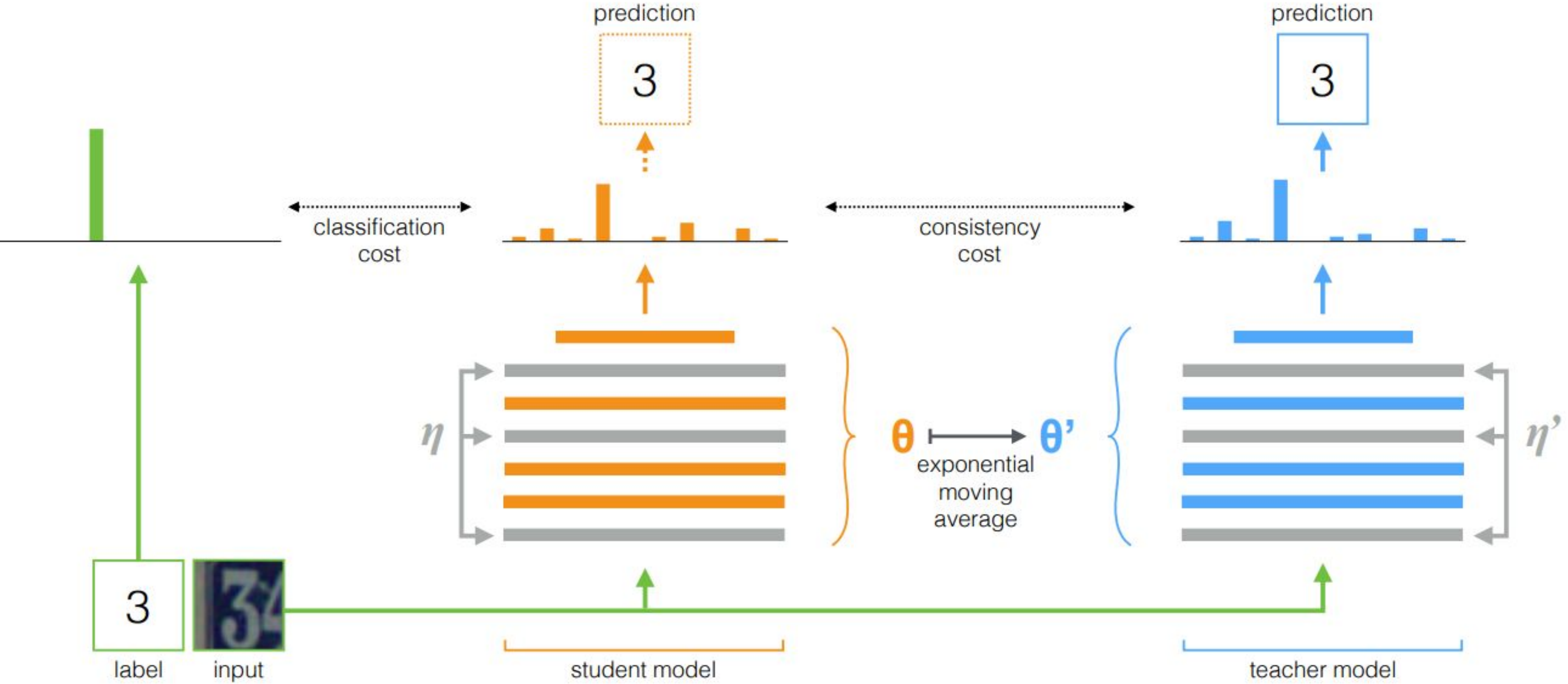
- T_{xy} is high if Nodes x and y are similar.

Temporal Ensembling for Semi-Supervised Learning (ICLR 2017)

- This ensemble prediction can be expected to be a better predictor for the unknown labels than the output of the network at the most recent training epoch, and can thus be used as a target for training.

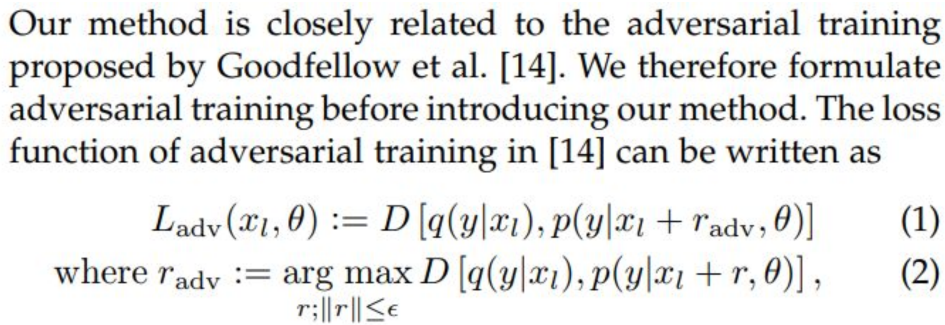


Mean teachers (NIPS 2017)



Virtual Adversarial Training - perturbation-based method (VAT)

Adversarial Training

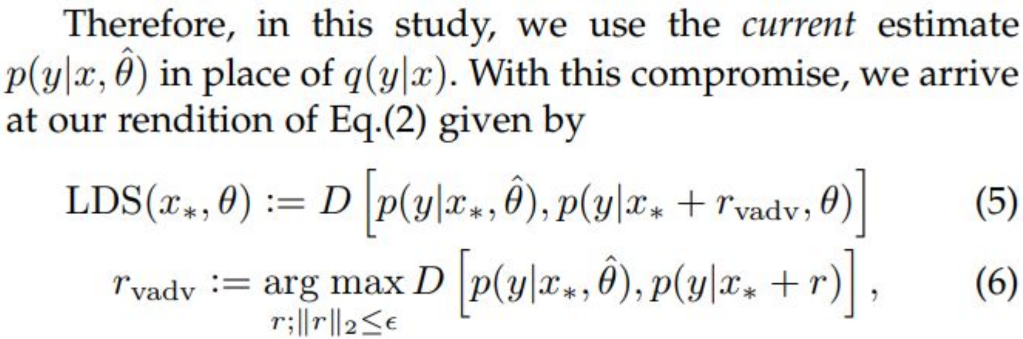


Our method is closely related to the adversarial training proposed by Goodfellow et al. [14]. We therefore formulate adversarial training before introducing our method. The loss function of adversarial training in [14] can be written as

$$L_{\text{adv}}(x_l, \theta) := D\left[q(y|x_l), p(y|x_l + r_{\text{adv}}, \theta)\right] \qquad (1)$$

$$\text{where } r_{\text{adv}} := \underset{r;\|r\|\leq\epsilon}{\arg\max}\, D\left[q(y|x_l), p(y|x_l + r, \theta)\right], \qquad (2)$$

Virtual Adversarial Training

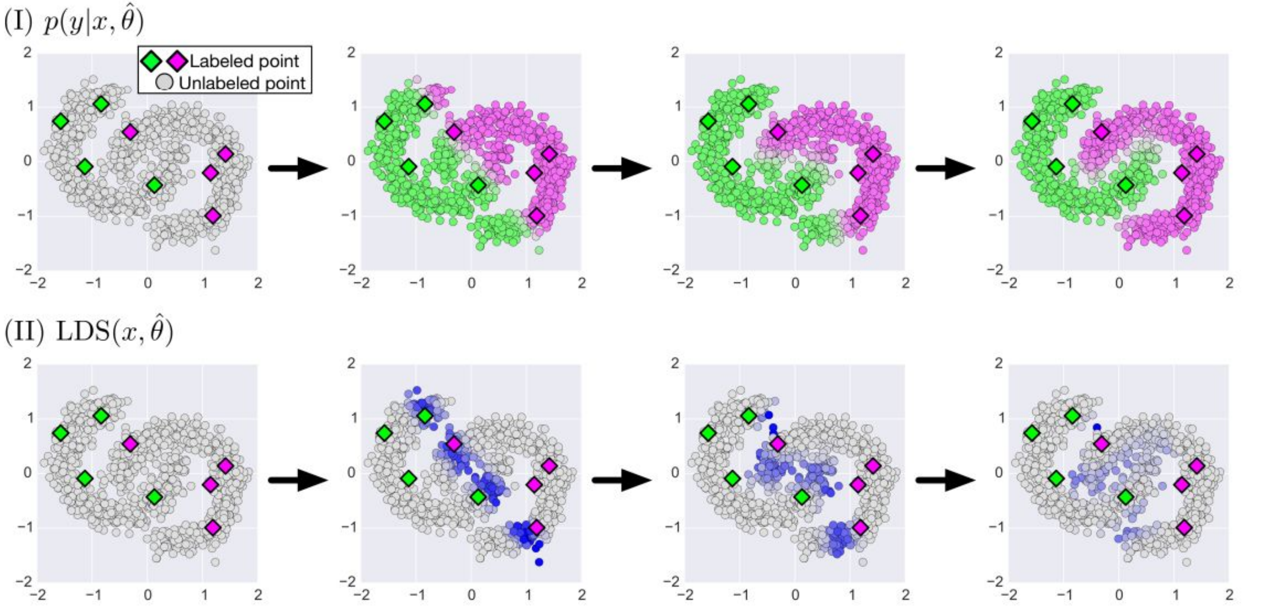


Therefore, in this study, we use the *current* estimate $p(y|x, \hat{\theta})$ in place of $q(y|x)$. With this compromise, we arrive at our rendition of Eq.(2) given by

$$\text{LDS}(x_*, \theta) := D\left[p(y|x_*, \hat{\theta}), p(y|x_* + r_{\text{vadv}}, \theta)\right] \qquad (5)$$

$$r_{\text{vadv}} := \underset{r;\|r\|_2\leq\epsilon}{\arg\max}\, D\left[p(y|x_*, \hat{\theta}), p(y|x_* + r)\right], \qquad (6)$$

Smoothness Assumption



(1) Before training (2) After 10 updates (3) After 100 updates (4) After 1000 updates

Fig. 1: Demonstration of how our VAT works on semi-supervised learning. We generated 8 labeled data points ($y = 1$ and $y = 0$ are green and purple, respectively), and 1,000 unlabeled data points in 2-D space. The panels in the first row (I) show the prediction $p(y = 1|x, \theta)$ on the unlabeled input points at different stages of the algorithm. We used a continuous colormap to designate the predicted values of $p(y = 1|x, \theta)$, with Green, gray, and purple respectively corresponding to the values 1.0, 0.5, and 0.0. The panels in the second row (II) are heat maps of the regularization term $\mathrm{LDS}(x, \hat{\theta})$ on the input points. The values of LDS on blue-colored points are relatively high in comparison to the gray-colored points. We used KL divergence for the choice of D in Eq.(5). Note that, at the onset of training, all the data points have similar influence on the classifier. After 10 updates, the model boundary was still appearing *over* the inputs. As the training progressed, VAT pushed the boundary away from the labeled input data points.

Review

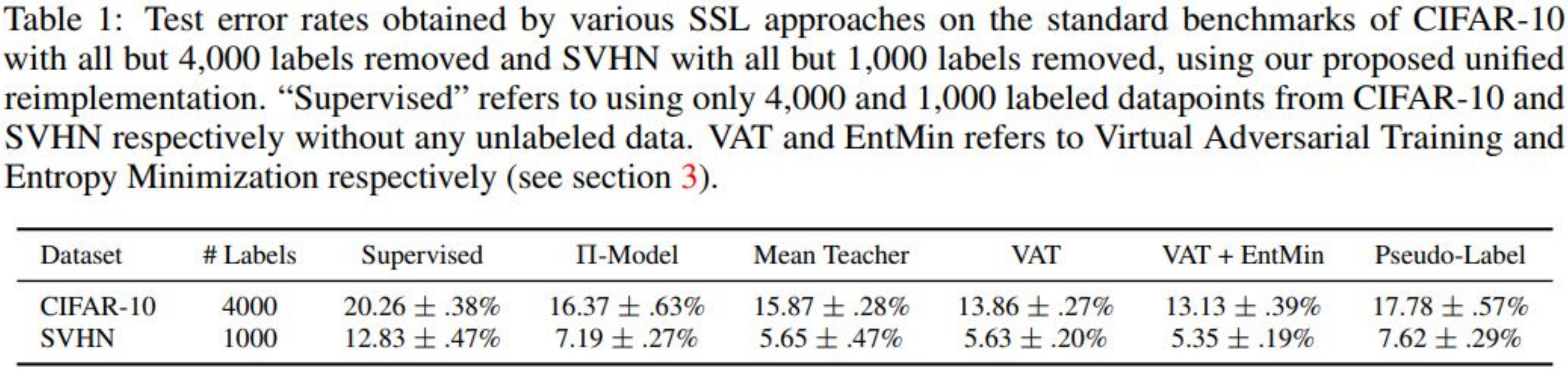


Table 1: Test error rates obtained by various SSL approaches on the standard benchmarks of CIFAR-10 with all but 4,000 labels removed and SVHN with all but 1,000 labels removed, using our proposed unified reimplementation. “Supervised” refers to using only 4,000 and 1,000 labeled datapoints from CIFAR-10 and SVHN respectively without any unlabeled data. VAT and EntMin refers to Virtual Adversarial Training and Entropy Minimization respectively (see section 3).

Dataset	# Labels	Supervised	Π-Model	Mean Teacher	VAT	VAT + EntMin	Pseudo-Label
CIFAR-10	4000	20.26 ± .38%	16.37 ± .63%	15.87 ± .28%	13.86 ± .27%	13.13 ± .39%	17.78 ± .57%
SVHN	1000	12.83 ± .47%	7.19 ± .27%	5.65 ± .47%	5.63 ± .20%	5.35 ± .19%	7.62 ± .29%

Summarize: “Student - Teacher” framework

- Enforce consistency loss between student and teacher
    - Mean_teacher
        - Student: original
        - Teacher: ensemble model (EMA)
    - VAT
        - Student output: perturbed input
        - Teacher output: original
    - Pseudo labelling
        - Student output: original
        - Teacher output: masks data point with high confident

Semi-supervised Regression

Semi-supervised Deep Kernel Learning: Regression with Unlabeled Data by Minimizing Predictive Variance - NIPS 2018 (**SSDKL**)

Baselines

- VAT
- Mean-teachers
- Label propagation
- CoREG (co-training)
  - Uses two k-nearest neighbor (kNN) regressors with different configuration
  - each of which generates labels for the other during the learning process

Semi-supervised Deep Kernel Learning (SSDKL)

$$L_{semisup}(\bar{\theta}) = -\frac{1}{n}\log p(\mathbf{y}_L|X_L,\bar{\theta}) + \frac{\alpha}{m}\sum_{x\in X_U}\mathrm{Var}_{f\sim p}(f(x))$$

Intuition

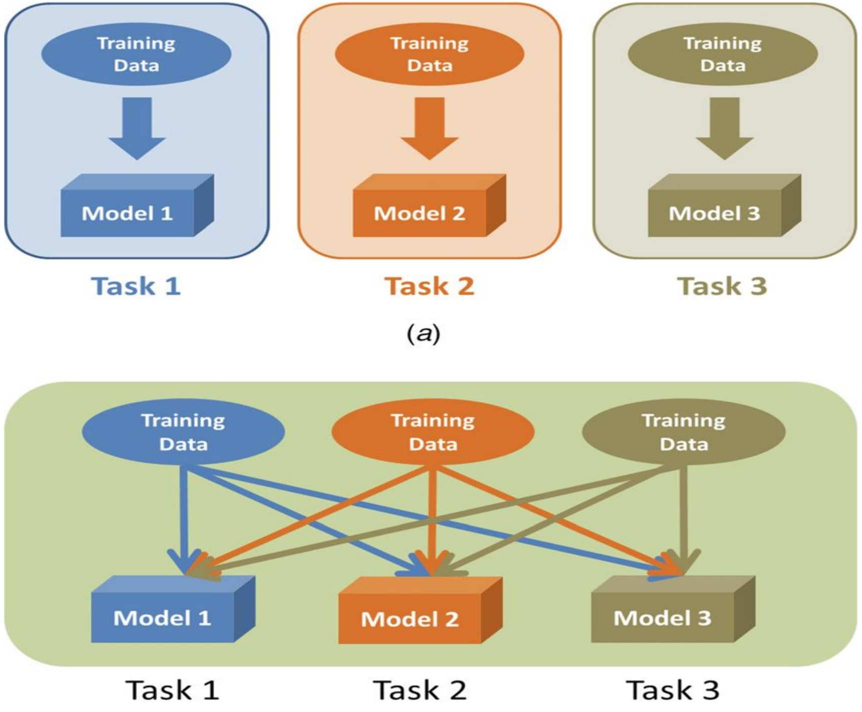
- Unlabeled data tend to cluster around these label supports (as it’s trained jointly)
- Semi-supervised objective serves as a regularizer that reduces overfitting, improve model generalization

Multi-Task Learning (MTL)

Multi-Task Learning (MTL)

Loose Definition: Anytime we optimize more than one loss function at the same time.

Definition: Given m learning tasks $\{T_i\}_{i=1}^{m}$ where all the tasks or a subset of them are related, *multi-task learning* aims to help improve the learning of a model for T_i by using the knowledge contained in all or some of the *m* tasks



(*a*)

(*b*)

	Training	Testing
Transfer Learning	Task 1	Task 2
Multi-task Learning	Task 1 ··· Task N	Task 1 ··· Task N
Lifelong Learning	Task 1 ··· Task N	Task N+1

Why Does MTL Work?

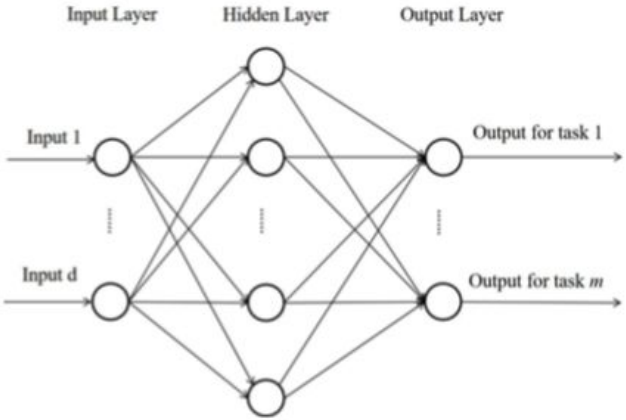
- **Implicit Data Augmentation** - Increase sample size
- **Feature Selection Double Check** - Features that carry across tasks are important
- **Eavesdropping** - Information transfer from different tasks
- **Representation Bias** - Generate flexible representations
- **Regularization** - Reduce risk of overfitting

What to Share?

- **Feature**: Learn common features among different tasks as a way to share knowledge
- **Instance**: Identify useful data instances in a task for other tasks and then shares knowledge via the identified instances
- **Parameter**: Uses model parameters (e.g., coefficients in linear models) in a task to help learn model parameters in other tasks in some ways, for example, regularization.

Feature Based

- **Feature Transformation**: The learned representation is a linear or non-linear transformation of the original representation. Each feature in the learned representation is different from the original.

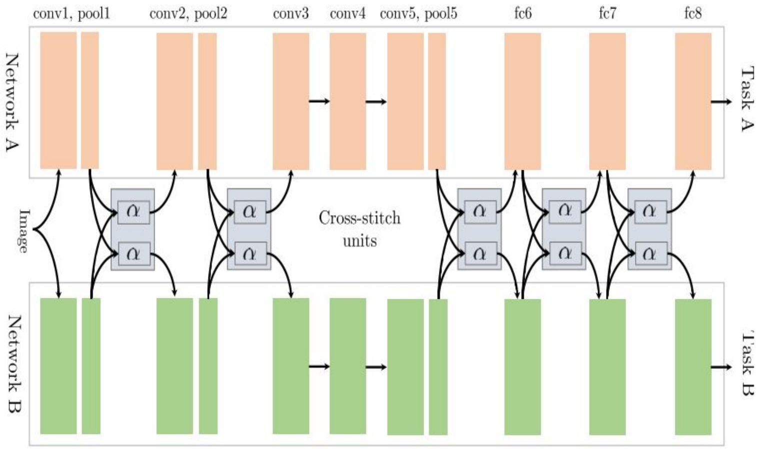


- **Feature Selection:** Select a subset of the original features as the learned representation. Learned representation is similar to the original by eliminating useless features.

Cross Stitch Network (Misra et al., 2016)

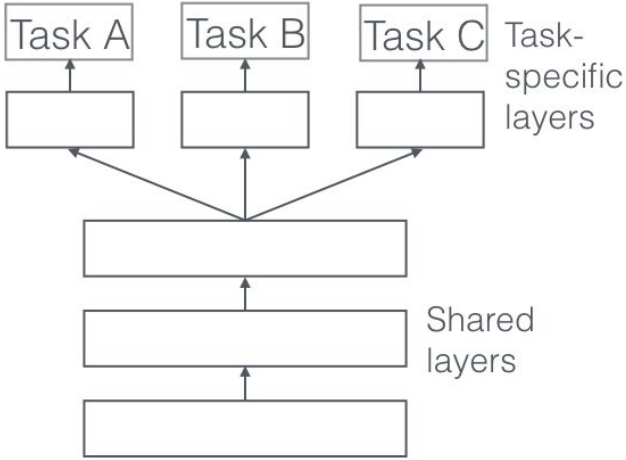
$$\begin{pmatrix} \tilde{x}_A^{i,j} \\ \tilde{x}_B^{i,j} \end{pmatrix} = \begin{pmatrix} \alpha_{AA} & \alpha_{AB} \\ \alpha_{BA} & \alpha_{BB} \end{pmatrix} \begin{pmatrix} x_A^{i,j} \\ x_B^{i,j} \end{pmatrix}$$

When α_{AB} and α_{BA} are both equal to 0 then training the two networks jointly is equivalent to training them independently.

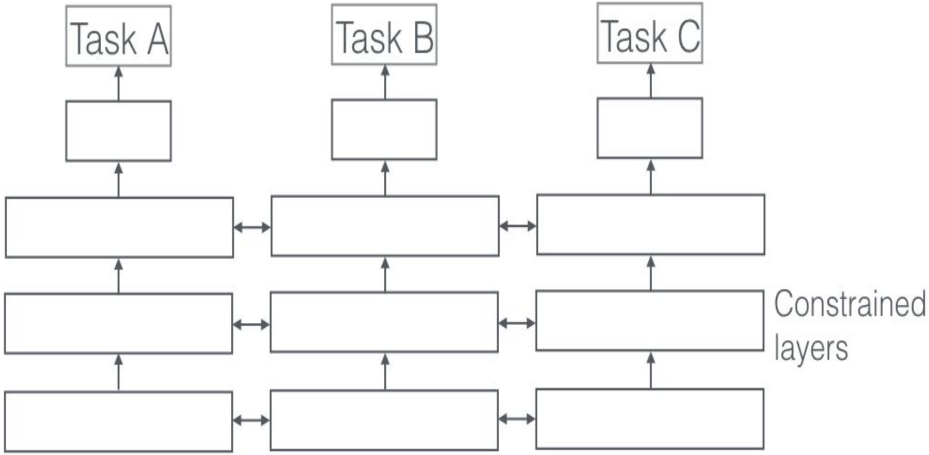


Parameter Based

Hard Parameter Sharing



Soft Parameter Sharing

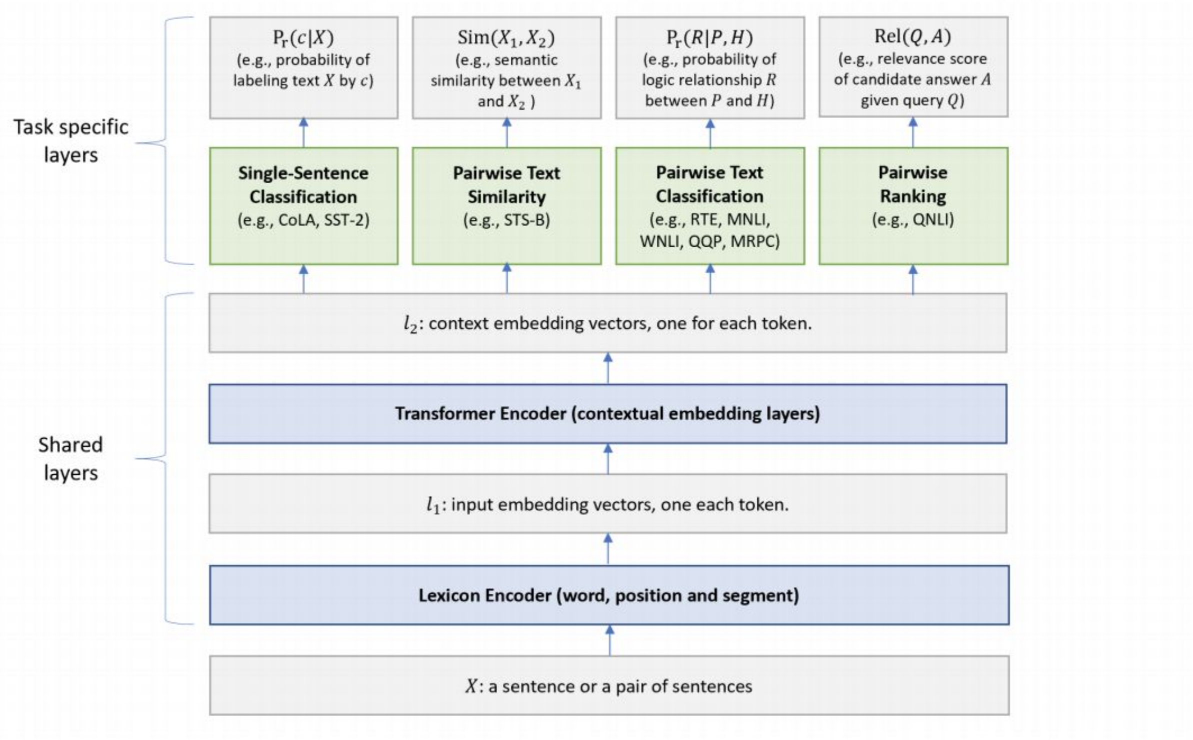


Auxiliary Tasks	Use case
Related Tasks	Similar tasks learn in a group
Adversarial Tasks	Domain Adaptation
Hints	Use easier tasks to help learn complex tasks
Representation Learning	Explicitly create a learned representation(Eg: Language Modelling)

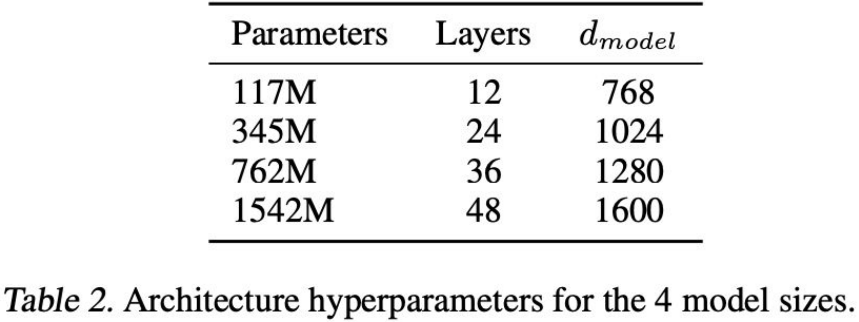
Learning General Purpose Sentence Representations (2018)

- **Idea**: Generate representations of sentences which are generalized and not task specific
- **Method**: Use diverse sentence-representation learning objectives and combine into a single multi-task framework.
  - Skip-thought vectors
  - Neural Machine Translation
  - Constituency Parsing
  - Natural Language Inference
- **Observation**: By utilizing multiple weakly related objectives, the representation encodes the inductive biases of each objective without becoming task-specific.

Multi-Task Deep Neural Networks for Natural Language Understanding(2019)

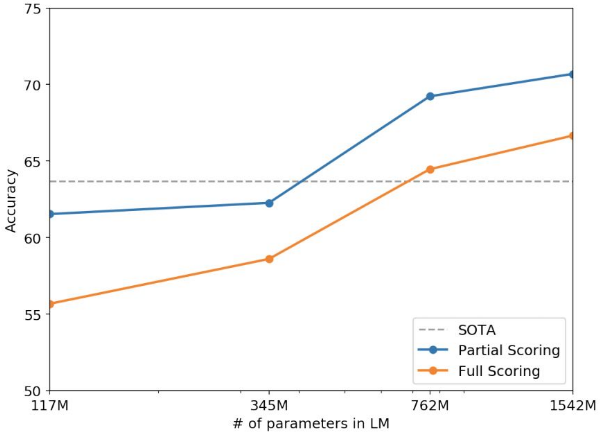

$\mathrm{P_r}(c|X)$ (e.g., probability of labeling text X by c)
$\mathrm{Sim}(X_1, X_2)$ (e.g., semantic similarity between X_1 and X_2)
$\mathrm{P_r}(R|P, H)$ (e.g., probability of logic relationship R between P and H)
$\mathrm{Rel}(Q, A)$ (e.g., relevance score of candidate answer A given query Q)
Task specific layers
Single-Sentence Classification (e.g., CoLA, SST-2)
Pairwise Text Similarity (e.g., STS-B)
Pairwise Text Classification (e.g., RTE, MNLI, WNLI, QQP, MRPC)
Pairwise Ranking (e.g., QNLI)
l_2: context embedding vectors, one for each token.
Transformer Encoder (contextual embedding layers)
Shared layers
l_1: input embedding vectors, one each token.
Lexicon Encoder (word, position and segment)
X: a sentence or a pair of sentences

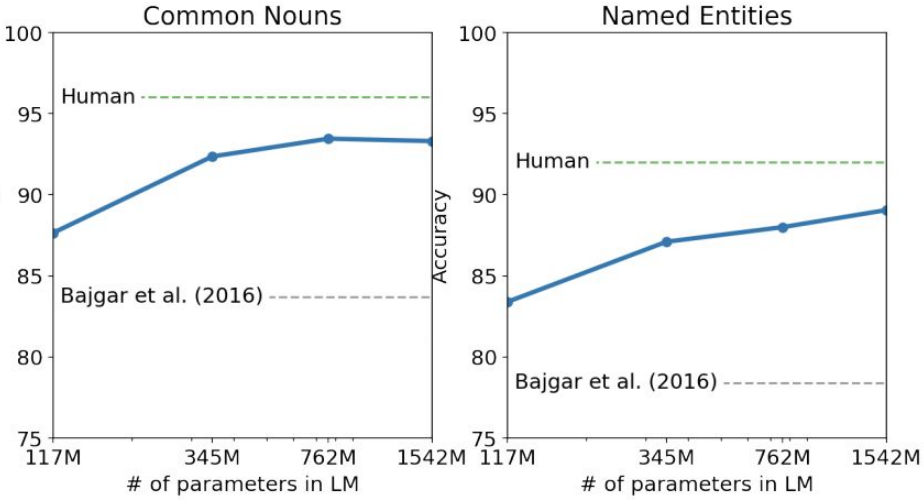
OpenAI GPT-2 (2019)



Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

Table 2. Architecture hyperparameters for the 4 model sizes.





In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.The scientist named the population, after their distinctive horn, Ovid’s Unicorn. These four-horned, silver-white unicorns were previously unknown to science. Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow. Pérez and the others then ventured further into the valley. “By the time we reached the top of one peak, the water looked blue, with some crystals on top,” said Pérez. Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns. While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, “We can see, for example, that they have a common ‘language,’ something like a dialect or dialectic.” Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America. While their origins are still unclear, some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization. According to Pérez, “In South America, such incidents seem to be quite common.” However, Pérez also pointed out that it is likely that the only way of knowing for sure if unicorns are indeed the descendants of a lost alien race is through DNA. “But they seem to be able to communicate in English quite well, which I believe is a sign of evolution, or at least a change in social organization,” said