GNR638 Course Project

December 2020

Few Shot Image Classification

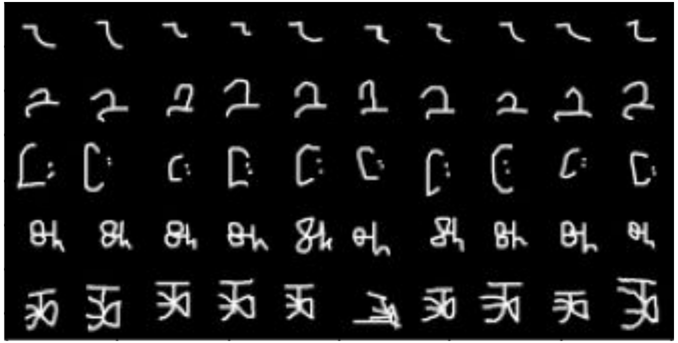
MAML + Student-Teacher Framework

Team Members:

- Arjit Jain, 170050010
- Yash Jain, 170050055
- Soumya Chatterjee, 170070010
- Aditya Vavre, 170050089

Task Description

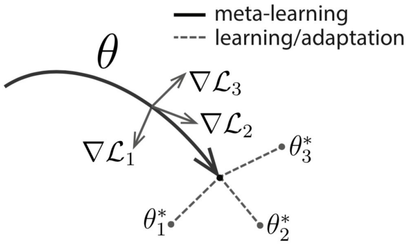
- **Few shot learning** is an example of meta learning. Here, a learner is trained on different related tasks such that it is able to generalize well on an unseen task after seeing only a few training examples.
- **N-way K-shot** image classification is a common few shot learning problem. Here the task is image classification with **N** classes, with only **K** labels per class.



Background

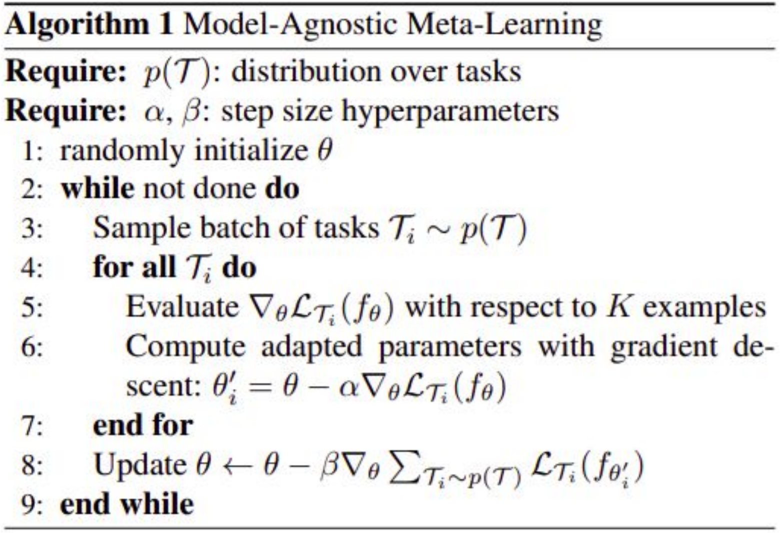
MAML
Self Training with Noisy Student
Rethinking Pre-training and Self-training

MAML



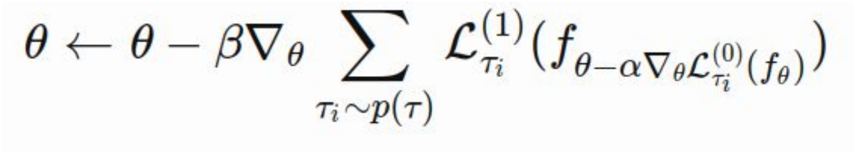
To goal is to find model parameters that are *sensitive* to changes in the task

Small changes in the parameters produce large improvements in the loss of any task from p(T) when altered in the direction of the loss



Algorithm 1 Model-Agnostic Meta-Learning

```
Require: $p(\mathcal{T})$: distribution over tasks
Require: $\alpha$, $\beta$: step size hyperparameters
randomly initialize $\theta$
while not done do
    Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
    for all $\mathcal{T}_i$ do
        Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ with respect to $K$ examples
        Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
    end for
    Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$
end while
```



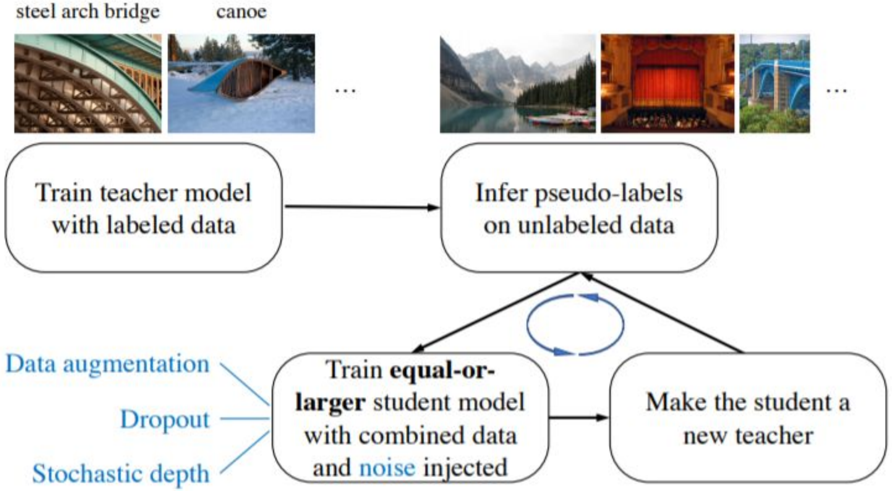
$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\tau_i \sim p(\tau)} \mathcal{L}_{\tau_i}^{(1)}\left(f_{\theta - \alpha \nabla_\theta \mathcal{L}_{\tau_i}^{(0)}(f_\theta)}\right)$$

Self Training with Noisy Student

An iterative semi-supervised learning algorithm.

Student is trained on the combination of labeled data and unlabeled data with pseudo labels from the teacher.

Student is typically larger than the teacher, and is trained with input noise (data augmentation), and model noise (stochastic depth, dropout)


steel arch bridge
canoe
...
...
Train teacher model with labeled data
Infer pseudo-labels on unlabeled data
Data augmentation
Dropout
Stochastic depth
Train **equal-or-larger** student model with combined data and noise injected
Make the student a new teacher

Rethinking Pre-training and Self-training

Pre Training

- Pre-training hurts performance when stronger data augmentation is used
- More labeled data diminishes the value of pre-training

Self-Training

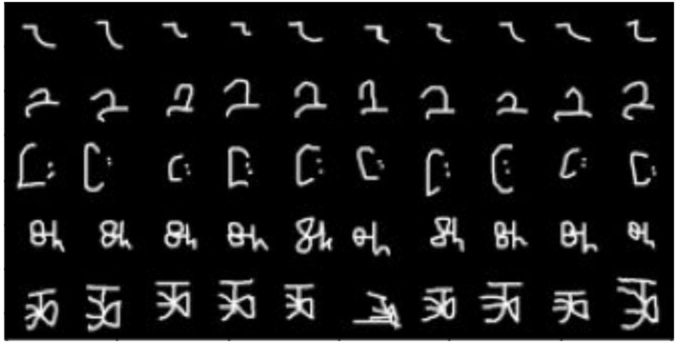
- **Self-training helps in high data/strong augmentation regimes, even when pre-training hurts**
- Self-training works across dataset sizes and is additive to pre-training

Motivation

- In Model Agnostic Meta Learning, the model sizes used are generally very small due to the need for calculating second order gradients which is expensive
- On image classification tasks, it is known that larger models give good results but these are domain specific
- We try to combine the benefits of these two ideas to meta-learn a small teacher network using MAML, fine tune it to a specific task and use that train a larger student network

Task Description

- 5 way 5 shot image classification
- Datasets Used:
  - Omniglot, CIFARFS and Mini-Imagenet
- Teacher Model: 4 layer CNN
  - Trained using MAML
- Student Model: Resnet18
  - Trained using Pseudo labels from the teacher



Method

Similar to other meta learning algorithms, our method is divided into two parts: Meta Training, and Meta Testing

Meta Training: This phase is dependent on the meta learning algorithm used by the teacher, and independent of the student training. For instance, in MAML, this phase would correspond to learning weight such that fine-tuning those weights gives low error on the test set.

Meta Testing:

- The teacher model performs “fast adaptation” to unseen tasks.
- Student training: Once we have adapted the teacher network, we use it to generate pseudo labels. A student network is then trained using the labeled training labels and pseudo labels following the traditional supervised training procedure.

The data for each task follows a three way split into training data D_{train}, validation data D_{val}, and testing data D_{test}.

The teacher network (T) is trained on D_{train} , and evaluated on D_{test}.

After adapting on D_{train}, the teacher network generated pseudo labels ($L = T(D_{val})$) on D_{val}. Finally, the student network (S) is trained on D_{train}, and D_{val} with the pseudo labels.

The student network is also evaluated on D_{test}.

Note that, we do not use the label information in D_{val}

Student Training

- Sources of Noise:

  Data augmentation and dropout could be used as the sources of noise for student training

- Pseudo Labels:

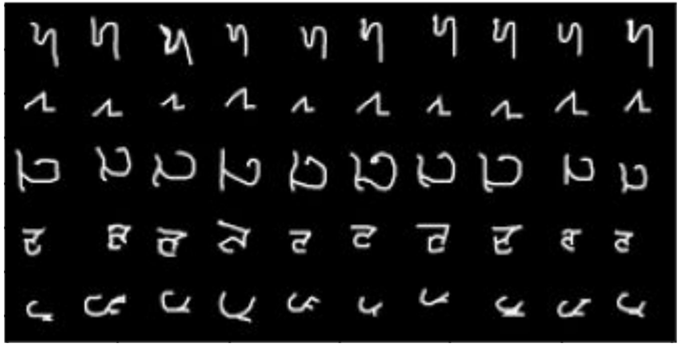
  We can have hard or soft pseudo labels. Hard pseudo labels correspond to performing an argmax operation on the teacher, whereas soft pseudo labels correspond to a softmax operation.

  For hard pseudo labels, we use the Cross Entropy loss, whereas for soft pseudo labels, we use the KL Divergence loss.

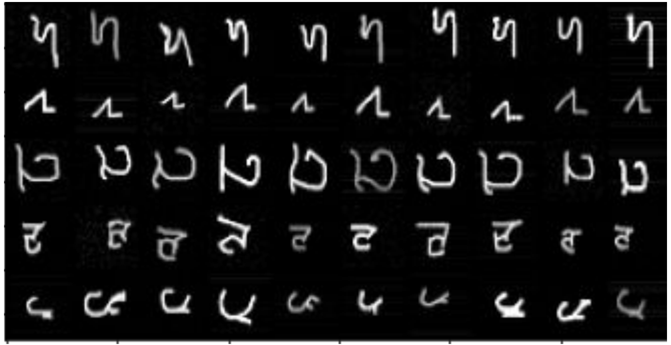
Augmentation

Since Omniglot has images of characters, Rotation, flipping etc cannot be used.

We used the following augmentations:
- Random crop leaving out 0-10% of the image
- Gaussian blur with 0.5 probability
- Contrast change to 0.75-1.5 times the original
- Adding Gaussian Noise
- Multiplying pixel values by a number in 0.8-1.2



Original Images



Images after Augmentation

Soft Labels vs Hard Labels

Teacher Accuracy = 96.6%

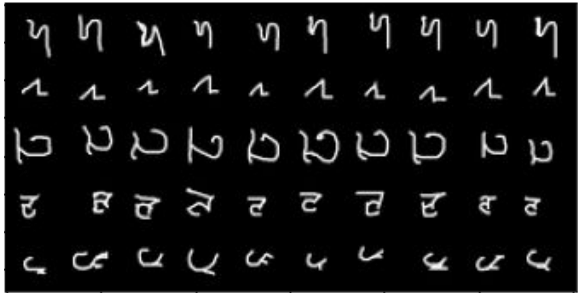
Configuration	Hard Labels (Accuracy in %)	Soft Labels(Accuracy in %)
ResNet-18 (not pre-trained)	**93.2**	92.4
ResNet-18 (train only final layer)	95.6	**97.9**
ResNet-18 (full fine-tune)	95.6	**97.7**

Effect of varying k and n in n-way-k-shot

Configuration	5 ways 1 shot (Accuracy in %)	5 ways 5 shots (Accuracy in %)	20 ways 5 shots (Accuracy in %)
Teacher	96.87	98.37	87.56
Not pretrained	80.62	96.50	77.31
Pre-trained (only fc)	84.37	98.24	86.96
Pre-trained (full ft)	86.25	98.24	85.34

Omniglot

- 5 ways 5 shot training for student model
- Teacher is a 4 layer standard CNN model for Omniglot dataset
- Student is ResNet-18 model
- 5 Adaptation labels + 5 Pseudo labels used for training the student model (1:1)



Configuration	Training Epochs	Meta Test Acc. (%)
Teacher	1000	98.37
Not pretrained	200	96.50
Pre-trained (only fc)	300	98.24
Pre-trained (full ft)	200	98.24

Other Datasets - Mini Imagenet & CIFARFS

Configuration	Mini Imagenet	CIFARFS
Teacher	58.12	54.62
Not pretrained	43.24	49.25
Pre-trained (only fc)	55.37	51.25
Pre-trained (full ft)	55.37	53.99

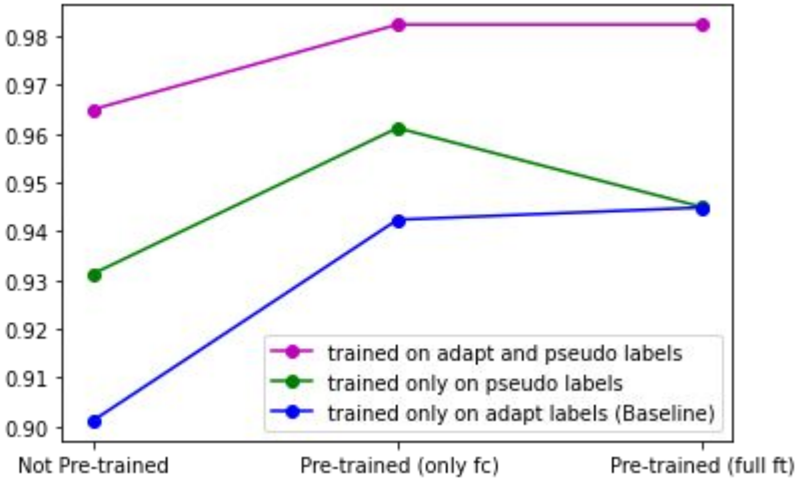
Ratio of labeled to unlabeled data (noisy labels)

Performance comparison of student model trained with 1:1 vs 1:2 ratio of labeled to unlabeled data

Configuration	**Meta Test Acc (1:1) (in %)**	**Meta Test Acc (1:2) (in %)**
Teacher	98.37	97.49
Not pretrained	96.50	96.62
Pre-trained (only fc)	98.24	97.49
Pre-trained (full ft)	98.24	**99.12**

Proof of concept

We tested our training regime by training student only with true labels instead of noisy labels and found that it always performs subpar to our proposed method



Different Teacher Algorithms

Configuration	MAML (accuracy in %)	First Order MAML (accuracy in %)	Reptile (accuracy in %)
Teacher	94.2	92.2	68.8
Not pretrained	92.1	87.3	88.8
Pre-trained (only fc)	94.6	90.6	96.8
Pre-trained (full ft)	94.7	92.1	95.2

Related Work

Meta Pseudo Labels (https://arxiv.org/abs/1906.00562)

- Train a teacher network such that, a student trained on pseudo labels from this teacher performs well on test data. Note that this is NOT in the context of meta learning.

Learning to Self-Train for Semi-Supervised Few-Shot Classification (https://arxiv.org/abs/1906.00562)

- This work is in the intersection of Meta Learning and Semi-Supervised Learning. While it does use Self-training, it does not use a noisy student framework. Similar model size restrictions, like MAML, apply to this work too

Conclusion

We demonstrated the effectiveness of the self-training framework in the context of meta-learning, in that, a student model trained with pseudo labels by the teacher, can outperform the teacher network.

We showed that this framework allows us to use deep neural networks architectures for the student without the computational overhead from gradient based meta learning algorithms. Pretraining the student can further increase the performance.

The student-teacher framework also provides a natural way to exploit unlabeled data.

Contributions

- Arjit Jain 25%
- Yash Jain 25%
- Soumya Chatterjee 25%
- Aditya Vavre 25%

Everyone contributed equally in performing various experiments and developing the code.

Thank You

Only pseudo labels for training the student

Yash

TRAINING DATA (5 img) EVAL DATA (5 img) TEST DATA(5 img)

Student,

Robustness

Iterative student teacher training

MAML (CNN) > Resnet 12

Resnet12 - > Resnet18