Investigating CoTTA: Validating Real-Time Neural Network Adaptations

Ansh Mujral
amujral@ucsd.edu

Ifunanya Okoroma
iokoroma@ucsd.edu

Keenan Serrao
kserrao@ucsd.edu

Nicholas Swetlin
nswetlin@ucsd.edu

Jun-Kun Wang
jkw005@ucsd.edu

Abstract

Machine learning models encounter the problem of distribution shifts, where the data used to train the model differs from the data used when the model is deployed at test time. When training new systems without access to the test data, it's important that these models still perform well under those circumstances. Without access to these labels at test time, an approach is to use CoTTA (Continual Test-Time Adaptation). CoTTA has had major success in adapting to distribution shifts. With that success, our group strives to take an under-the-hood approach to better our understanding — assessing model architecture and the affect it can have on CoTTA performance.

Website: https://abc.github.io/
Code: https://github.com/nickswetUCSD/Capstone-Project-Q2

1 Introduction

As machine learning becomes more widely used in today's world, there are various issues that arise with it, one of them being addressing of distribution shifts within data. Distribution shifts occur when data the model was trained on vastly differs from the data performed on at test time. These shifts can lead to miscalculations in models, making them perform worse and reducing their efficiency.

One example of the importance of addressing distribution shifts: self-driving cars. They need to be able to adapt in real time to different lighting conditions. Otherwise, common environments with lots of rapid lighting changes, such as driving through a short tunnel or on a curving road during sunset, would be significantly more unsafe than stable lighting environments to have self-driving cars navigate. Besides self-driving cars, rapid adaptation is useful in many technologies: personalized medical imaging of tumors, analyzing market trends of specific stocks, and conversion of handwriting to digital text (OCR).

For these adaptations, the concept of "Continual Test Time Adaptation" (CoTTA) proves to be useful, as it allows models to update their parameters in real time and adapt to new data on the fly (Wang (2022)). We aim to assess the robustness of CoTTA. *Does the quality of CoTTA hold up on simple and complex models alike? How does model architecture affect the performance of CoTTA? Are different methods other than CoTTA superior under different architectures?*

2 Methods

We propose to test the adaptive abilities of CoTTA against several other variations of TTA.

These different variations are as follows:

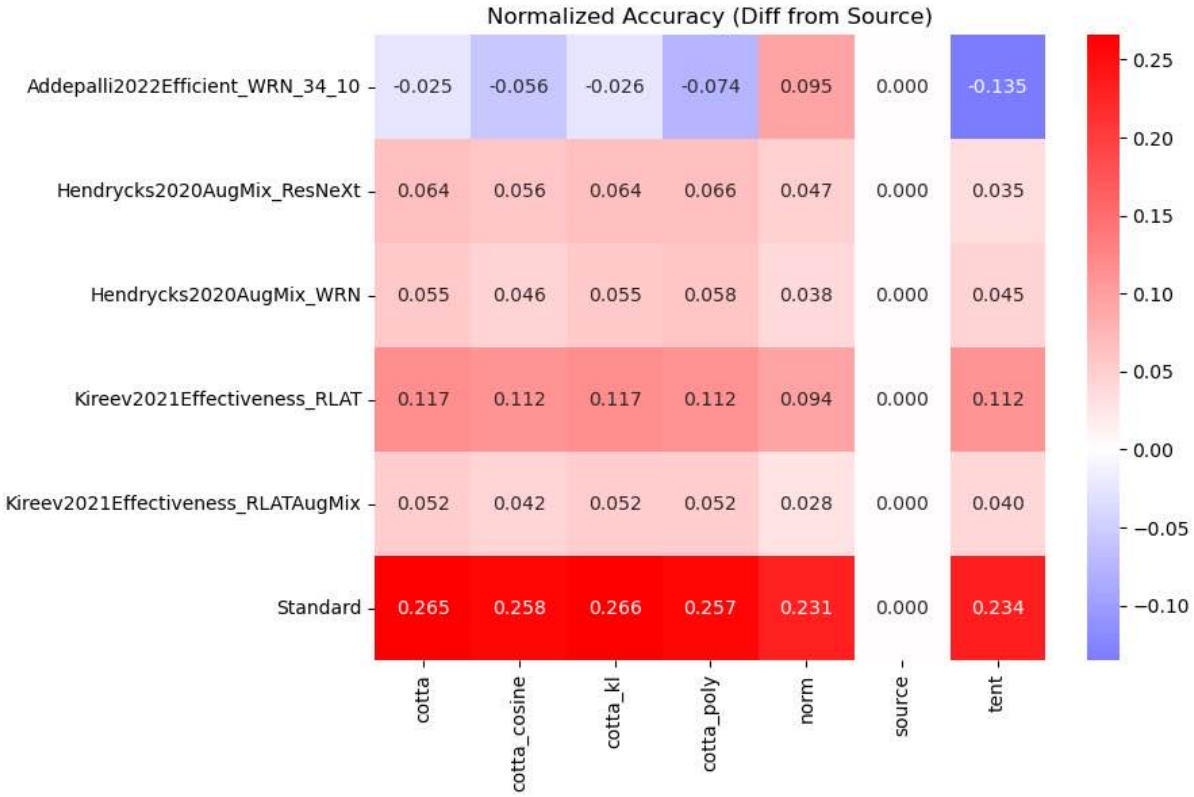
- No test-time adaptation.
- Vanilla test-time adaptation.
- TENT - fully test-time adaptation by entropy minimization
- CoTTA employing the cross-entropy loss
- COTTA employing the cosine similarity
- CoTTA employing Polyloss
- COTTA employing KL Divergence
- COTTA employing a self-training version of the cross-entropy loss

For each of these variations, we will adapt a list of neural network classifiers and replicate the test-time adaptation benchmark CIFAR10-to-CIFAR10-C for multiclass classification. Each classifier in our classifier list will have a different architecture. The end result will be a table where:

- Each row: represents a different neural network architecture.
- Each column: represents a different TTA method.
- Each entry in the table describes the accuracy, precision, recall, and F1 score of the

combination of architecture and TTA variation.

3 Results



When model architecture is held equal, we see that the improvement of accuracy, precision, recall, and F1 of CoTTA predictions relative to source, with any form of consistency loss, provides similar quality (robustness within ± 0.01% of accuracy for majority of models). This is shown by the ”horizontal stripes” lined across the columns with CoTTA variations. When TTA loss is held equal, we see that model size has a clear effect (though non-directional) on the improvement of accuracy, precision, recall, and F1 of CoTTA predictions relative to source. It is difficult to articulate a trend between size of model architecture and relative improvement of TTA method to source. The model with the best improvement by TTA to source was Standard (28 Layers, Wide), while the model with the worse improvement by TTA relative to source was Addepalli2022Efficient_WRN_34_10 (TODO insert model size).

$$\boldsymbol{L}(\hat{t},\hat{s}) = \underbrace{-\sum_{c} \hat{t}_c \log \hat{s}_c}_{\text{Cross-Entropy Loss}}$$

$$\theta^* = \arg\min_{\theta} \mathscr{L}_{\text{TTA}}(\mathscr{X}; \theta)$$

PolyLoss:

$$\boldsymbol{L}_{\text{poly}}(h'_\theta(x), y) = f(g^{-1}(h'_\theta(x))) - y^T h'_\theta(x)$$

Cosine Sim:

KL:

$$D_{\text{KL}}(P\|Q) = \sum_c P(x) \log \frac{P(x)}{Q(x)}$$

4 Discussion

It is clear that a valid substitution to the cross-entropy consistency loss in the original implementation of CoTTA will yield similar TTA improvements as CoTTA with cross-entropy consistency loss. This is true for many common metrics related to classification quality: accuracy, precision, recall, F1. This implies that altering the consistency loss of CoTTA does not tweak the relative weights of true positive, true negative, false positive, and false negative results, meaning that current biases or limitations of a model architecture will likely not significantly worsen or improve as a result of the introduction of a CoTTA framework. Different architectures pose a significant threat to the robustness of CoTTA as a method. For example, using the model Addepalli2022Efficient_WRN_34_10 had CoTTA-based TTA methods produce a negative effect on prediction quality across every metric. Future work is needed to determine the efficacy of using CoTTA-based methods on architectures greater than 40 layers in size, and even beyond, to use with larger LLM structures.

Limitations of the experimental approach include:

- A small sample size of select neural network architectures accessible on the RobustBench github repository. Different choices of model architectures might influence the underlying improvement produced by TTA. These changes are mentioned in previous literature, though as a mention to TTA in general and not CoTTA TODO need to verify and cite.
- Metrics were evaluated on the CIFAR10 to CIFAR10C benchmark continual task for TTA evaluation, which involves low resolution data souces. Other benchmarks for CoTTA methods, such as CIFAR100 to CIFAR100C or ImageNet to ImageNetC, were not performed in the interest of expediting the experiment. Results may be different on higher resolution data, as corruptions or distribution shifts fundamentally change more cumulative data when applied to a higher resolution image.

- Not all possible variations or hyperparameters were explored in the creation of this experiment. For example, the epsilon paramater for polyloss has a significant effect on the behavior of polyloss, so the TTA method of CoTTA with incorporated polyloss may need further examination of behavior with different values of epsilon.
- A non-gradual ordering of corruption types with only Severity 5 Corruptions were examined in this experiment. Lower severities of corruptions or the introduction of corruptions to a model gradually may influence the effectiveness of TTA methods involving CoTTA.

TODO update once methods and intro complete.

5 Conclusion

I'll write this once we pair everything else down.

6 Appendix

Adapted from the Quarter 2 Proposal

As machine learning becomes more widely used in today's world, there are various issues that arise with it, one of them being the presence of distribution shifts within data. Distribution shifts occur when data the model was trained on vastly differs from the data performed on at test time. These shifts can lead to miscalculations in models, making them perform worse and reducing their efficiency. For example, self-driving cars need to be able to adapt in real time to different lighting conditions. Otherwise, common environments with lots of rapid lighting changes, such as driving through a short tunnel or on a curving road during sunset, would be significantly more unsafe than stable lighting environments to have self-driving cars navigate.

Of course, rapid adaptation is useful in many technologies other than self-driving cars: personalized medical imaging of tumors, analyzing market trends of specific stocks, and conversion of handwriting to digital text (OCR) are all excellent examples. For these adaptations, the concept of continual test time adaptation (CoTTA) proves to be extremely useful, as it allows models to update their parameters in real time and adapt to new data (Wang (2022)). We aim to answer this question with our quarter two project: *How can we train neural networks to make these adaptations occur faster?*

One promising framework that allows neural networks underlying these machines to rapidly adapt to data is called Continual Test-Time Adaptation (CoTTA). CoTTA involves a complex system involving several interactions between two models: A "student" model (f_Θ) and a "teacher" model (f'_Θ). In default CoTTA, the student model at time t is trained via a cross-entropy loss, which is dependent on both the teacher and student models: $\ell_{\Theta_t} = -\sum_c {y'}_{tc} \log(\hat{y}_{tc})$; where $\hat{y}_t$ is the direct student prediction at time t and ${y'}_t$ is the teacher prediction with the option of being augmentation-averaged at time t (depending

on the specific version of CoTTA). But if we substitute the student's loss with a different loss function, such as a self-training loss function for cross-entropy dictated by conjugate pseudo-labels (mentioned in Goyal (2022)), or with a cosine similarity, what would happen? Would we see an increase or decrease in CoTTA's ability to adapt? Previous work on modifying CoTTA has not been substantially explored, and, combined with the fact that the specific loss functions we wish to investigate are either ubiquitously used in machine learning or satisfy other optimality conditions from machine learning adaptation frameworks, there is significant motivation to explore variations of CoTTA. We propose to test the adaptive abilities of several "variations" of CoTTA, each of which is the same as the vanilla version of CoTTA with the replacement of the student's loss function with a new loss function:

- Self-training loss function for cross-entropy loss.
- Cosine similarity between student and teacher predictions.
- A custom self-training loss function satisfying Goyal et al.'s expanded conjugate loss form.

For each variation, we will adapt a source model and replicate test-time adaptation benchmarks CIFAR10-to-CIFAR10-C, CIFAR100-to-CIFAR100-C, and ImageNet-to-ImageNet-C (inspired by Wang et. al), noting test error of each variation. Our end deliverables will be:

- A repository with reproducible code of our experiment.
- A paper detailing our methods and results.

7 Contributions

- Ansh Mujral: Implemented two different models from the integrated repo as well as validating the data.
- Ifunanya Okoroma: Implemented two different models from the integrated repo as well as validating the data.
- Keenan Serrao: Restructured the original CoTTA repository to keep relevant files related to architecture.
- Nicholas Swetlin: Fixed architecture and consolidated those files in a Google folder. Implemented two different models from the integrated repo as well as validating the data.

References

Goyal, Raghunathan Kolter, Sun. 2022. "Test-Time Adaptation via Conjugate Pseudo-labels." In *Neural Information Processing Systems (NeurIPS)*. [Link]

Wang, Gool Dai, Fink. 2022. "Continual Test-Time Domain Adaptation." [Link]