Utilizing Attribution Maps for Regularization of Deep Neural Networks

Content

Introduction

Current Situation & Problem

Current situation

Deep learning very successful in the recent years.

- AlexNet[1] 2012
- AlphaGo[2] 2016
- ChatGPT[3] 2022

Research often more advanced than industry applications.

Problem

- Large amounts of data needed, otherwise overfitting occurs leading to low generalization.
- Regularization to counteract this exists but does not cover all domains well.

Approach and Contributions

Analyze and refine a domain independent regularization technique, attribution based augmentation (ABA).
Scope: Deep Learning, especially classification

Contributions

01

Framework

Framework , terminology and pipeline.

02

Theory

Identify mechanisms for different pipeline parts.

03

Methods

MendABA, FastABA & benchmarks.

Background

Problems in DL

Problems

- Overfitting and generalization
- Interpretability

Problems in DL

Problems

- Overfitting and generalization
  => Regularization
- Interpretability

Problems in DL

Problems

- Overfitting and generalization
  => Regularization
- Interpretability
  => Attribution methods

Attribution Methods

Importance of input features for model prediction are determined and can be displayed in an interpretable fashion.

Path attribution:

- Layer-wise Relevance Propagation[4]
- Integrated Gradients[5]

Gradient-only:

- Vanilla Gradient[6]
- Gradient x Input[7]
- CAM variants[8]

Occlusion based:

- LIME[9]
- SHAP[10]

Regularization

Regularization classes[11] :

- Data
- Architecture
- Optimization

=> Can be combined

Regularization via data:

- Noise Injection[12]
- Adversarial Training[13]
- Hand-crafted data augmentation[14]
- Attribution based augmentation[15]

Attribution Based Augmentation

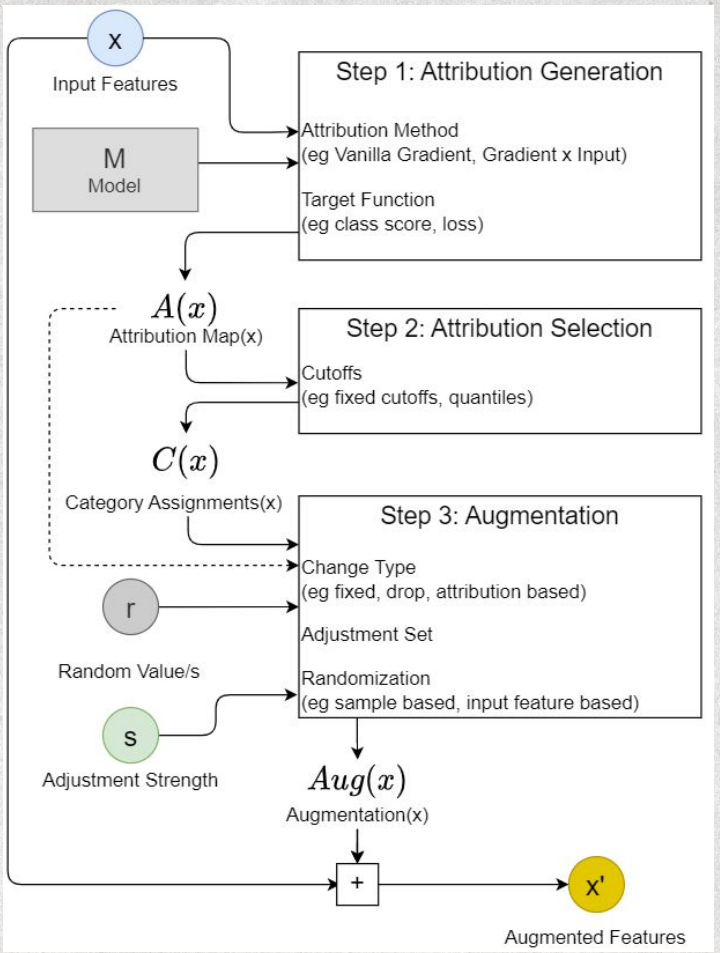
In ABA regions of input are augmented in different ways based on the attribution map.

Starting point for thesis is the Challenger[15] method:

1. Compute attribution map.
2. Determine most important features for one class based on percentile of corresponding attributions.
3. Randomly increase or decrease the highest or lowest attribution features. Some original samples are preserved.
4. Training continues with augmented samples.

Framework

Pipeline

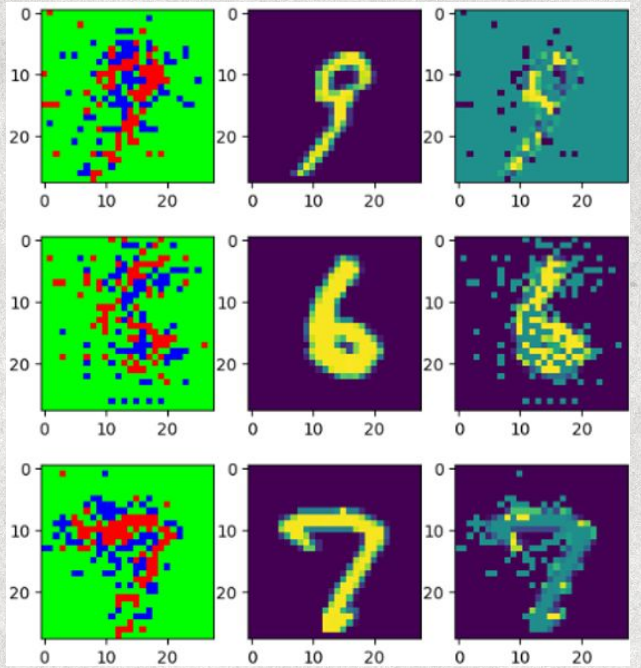

x
Input Features
M
Model
Step 1: Attribution Generation
Attribution Method
(eg Vanilla Gradient, Gradient x Input)
Target Function
(eg class score, loss)
$A(x)$
Attribution Map(x)
Step 2: Attribution Selection
Cutoffs
(eg fixed cutoffs, quantiles)
$C(x)$
Category Assignments(x)
Step 3: Augmentation
Change Type
(eg fixed, drop, attribution based)
r
Random Value/s
Adjustment Set
Randomization
(eg sample based, input feature based)
s
Adjustment Strength
$Aug(x)$
Augmentation(x)
+
x'
Augmented Features

Adjustments

Adjustment:
Defines changes applied on inputs based on category.

$$AugBase_i(x) = \sum_{n=1}^{n_c} C_{ni} * a_n$$

$$Aug'(x) = AugBase(x) * s * |\mathcal{N}(0, 1)|$$



[0,0,-1]

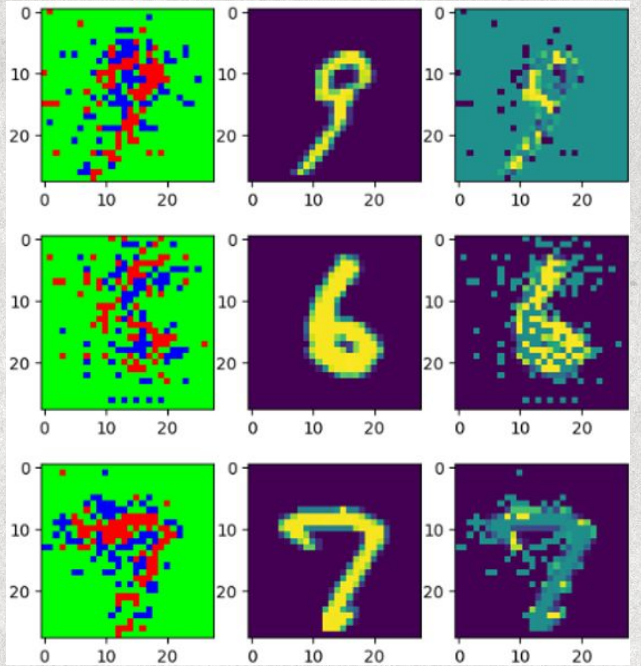
[0,-1,0]

[1,0,0]

Adjustment Sets

Set containing possible adjustments:

- One adjustment in the set is applied on each sample.
- Adjustments from the set are selected randomly.



Analysis

Augmentation Step

How to select good adjustment sets?

Phenomenon: Adjustment sets often outperform constituents by a lot.

Adjustment Groups

Low	Middle	High	Type	Index
0	0	0	Neutral	0
0	1	0	Neutral	1
0	-1	0	Neutral	2
0	0	1	Epsilon	3
0	0	-1	Delta	4
1	0	0	Delta	5
-1	0	0	Epsilon	6

With cross entropy loss:

High attribution inputs when increased should increase loss

⇒ Epsilon

High attribution inputs when decreased should decrease loss

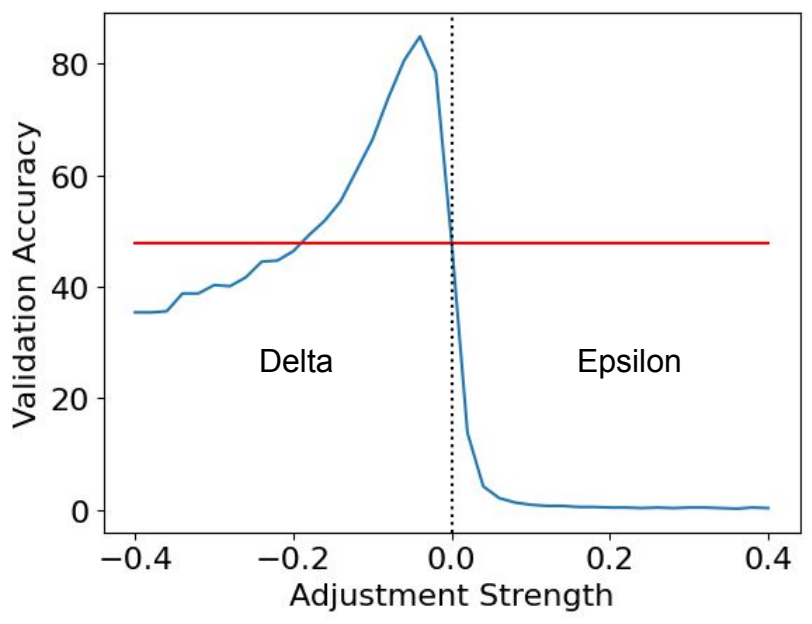
⇒ Delta

Medium attribution inputs should keep loss relatively stable

⇒ Neutral

Adjustment Groups

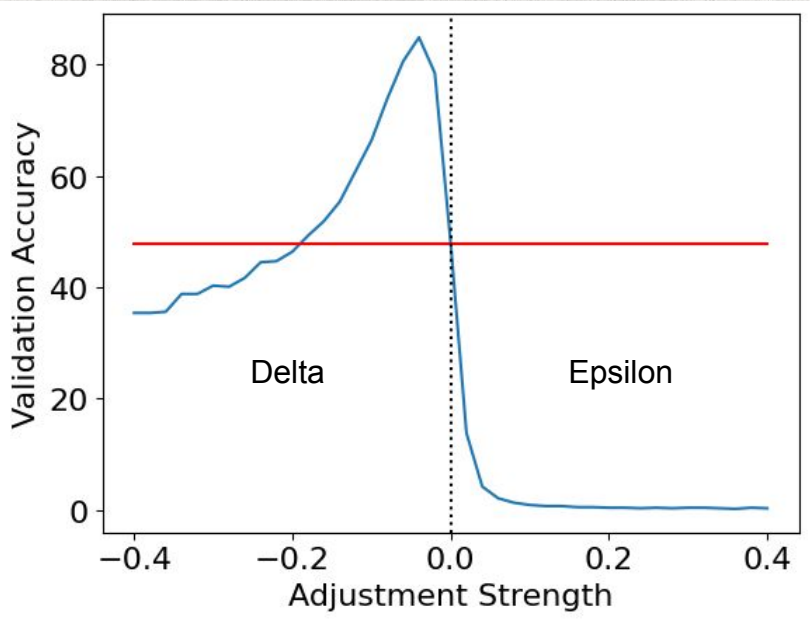
Adjustments applied in evaluation



20Newsgroups[16] Tiny, small CNN

Adjustment Groups

Adjustments applied in evaluation



Adjustments applied in training

Low	Middle	High	Type	Index	MAX 5 runs	AVG 5 runs	Rating
0	0	0	Neutral	0	49.1	**47.2**	Baseline
0	1	0	Neutral	1	47.6	**46.44**	Medium
0	-1	0	Neutral	2	48.9	**47.02**	Medium
0	0	1	Epsilon	3	55.3	**53.12**	Good
0	0	-1	Delta	4	42.6	**41.88**	Bad
1	0	0	Delta	5	45.1	**41.28**	Bad
-1	0	0	Epsilon	6	57.5	**53.94**	Good
0	1	1	Epsilon	7	57.5	**55.86**	Top
0	-1	1	Epsilon	8	56.3	**53.82**	Good
0	1	-1	Delta	9	44.1	**42.06**	Bad
0	-1	-1	Delta	10	43.6	**42.38**	Bad
1	1	0	Delta	11	44.6	**41.46**	Bad
1	-1	0	Delta	12	46.3	**43.08**	Bad
-1	1	0	Epsilon	13	59.5	**57.98**	Top
-1	-1	0	Epsilon	14	57.6	**54.34**	Good
1	0	1	Neutral	15	49.4	**47**	Medium
1	0	-1	Delta	16	48.9	**44.68**	Bad
-1	0	1	Epsilon	17	57.6	**55.3**	Top
-1	0	-1	Neutral	18	47.9	**45.7**	Medium
1	1	1	Neutral	19	50.6	**48.14**	Medium
1	-1	1	Neutral	20	52.3	**46.72**	Medium
1	1	-1	Delta	21	47.7	**42.88**	Bad
1	-1	-1	Delta	22	45.6	**42.82**	Bad
-1	1	1	Epsilon	23	57.3	**53.08**	Good
-1	-1	1	Epsilon	24	58.3	**55.86**	Top
-1	1	-1	Neutral	25	51.4	**48.7**	Medium
-1	-1	-1	Neutral	26	49.2	**46.56**	Medium

20Newsgroups Tiny, small CNN

Augmentation Step

How to select good adjustment sets?

Phenomenon: Adjustment sets often outperform constituents by a lot.

Two Hypotheses

Variety

More adjustments in an adjustment set tends to improve generalization.

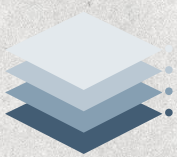
Adjustments from different adjustment groups in the same adjustment set tend to improve generalization. Especially opposed ones.

Tension

Hypothesis: Experiment

A={[1,0,0]}

A={[1,0,-1]}



A={[0,0,-1]}

Models trained with one adjustment each from an adjustment group.

A={[1,0,0],[0,0,-1],[1,0,-1]}

Model trained with all adjustments from an adjustment group.

Compare validation accuracies

Hypothesis: Results

Variety

=> Some support

Tension

=> Good support

Tiny CIFAR10[17] with VGG16[18]

Adj. group	All of groups	AVG of groups	Difference
Neutral	37.43	36.35	1.08
Delta	34.98	34.50	0.48
Epsilon	39.90	36.23	3.66
Eps. & Neu.	40.40	36.29	4.11
Eps. & Del.	40.27	35.37	4.90
All	40.26	35.69	4.56

20Newsgroups with small CNN

Adj. group	All of groups	AVG of groups	Difference
Neutral	46.62	47.05	-0.43
Delta	45.02	42.50	2.52
Epsilon	53.84	54.81	-0.97
Eps. & Neu.	56.50	50.93	5.57
Eps. & Del.	57.53	48.66	8.88
All	58.94	48.12	10.82

Augmentation Generation Step

Occlusion based:

- LIME
- SHAP

→ General but very compute intensive.

Path attribution:

- Layer-wise Relevance Propagation → Architecture depencency.
- Integrated Gradients → Compute intensive but good attribution map quality.

Gradient-only:

- Vanilla Gradient
- Gradient x Input
- CAM variants

→

- Fast, general and easy to implement.
- Fast, general, easy to implement, LRP equivalence.
- High architecture dependency and some compute intensive.

Recommendation

For settings where compute is limited:
Use Gradient x Input or Vanilla Gradient.

For settings with high compute:
Use Smoothgrad[19] in combination with Gradients x Input or Vanilla Gradient, or use Integrated Gradients.

Additional Findings

- Class score as attribution target seems slightly better than loss.
- Using randomization on fixed values in the augmentation step seems to work best.
- Opposite adjustments seems to work well in the adjustment sets.
- Including medium adjustments seems to be helpful.

Method Creation

MendABA

- Vanilla Gradient
- 3 category split with percentiles at 10% and 90%.
- Randomization per sample and adjustment set: A={1,2,3,4,5,6}

FastABA

- Vanilla Gradient
- 2 category split with cutoff at 0.
- Randomization per sample and adjustment set: A={0,0,3,4,5,6,16,17}

Benchmark

Datasets

Dataset	Train-Full	T-Med	T-Tiny	Valid	Test	Input Dim	Classes
MNIST	50000	1000	100	10000	10000	28x28x1	10
CIFAR	40000	-	1000	10000	10000	32x32x3	10
20Newsgroups	14866	-	2000	1000	4000	1000x100	20
Tiny ImageNet	100000	-	-	10000	10000	64x64x3	200
PTB-XL	17441	-	-	2193	2203	1000x12	71

MNIST[20], Tiny ImageNet[21], PTB-XL[22]

Results: Analysis Datasets

Dataset	Subset	Epochs	Vanilla	Challenger	MendABA	FastABA
MNIST	Tiny	1000	63.1	83.1	82.8	84.7
	Med	100	91.6	96.0	95.9	96.2
	Full	10	99.1	99.2	99.2	99.2
CIFAR	Tiny	500	33.5	38.4	41.0	38.5
	Full	30	73.8	74.5	74.8	74.7
NEWS	Tiny	100	46.2	57.2	60.5	58.1
	Full	30	66.7	72.8	72.9	73.0

Results: Tiny ImageNet

Model	RandAug	Epochs	Vanilla	Challenger	MendABA	FastABA
ResNet50	No	100	31.31	32.74	32.86	33.13
ResNet50	Yes	100	46.93	45.75	47.98	46.71
EfficientB3	No	300	41.33	40.94	42.11	42.18
EfficientB3	Yes	300	53.59	53.29	53.66	53.63
VIT Base 16	No	50	28.20	29.95	31.77	33.01

ResNet50[23], EfficientNetB3[24], ViT[25]

Results: PTB-XL

Dataset	Vanilla	Challenger	MendABA	FastABA
PTB-XL	91.40	91.50	91.50	91.40

Conclusion

Conclusion

Framework

- The seperation of ABA into 3 steps and the introduction of the concepts of adjustments, adjustment groups and adjustment sets seems useful.
- Culminated in theory of adjustment tension, which should help select useful adjustment sets.

Benchmark

- ABA seems to be performant on small datasets and a good regularization option for domains other than vision.
- In vision it seems difficult to outperform hand-crafted data augmentation.

Thanks!

Sources

1. A. Krizhevsky, I. Sutskever, and G. E. Hinton. “Imagenet classification with deep convolutional neural networks.” In: Communications of the ACM 60.6 (2017), pp. 84–90.
2. D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. “Mastering the game of Go with deep neural networks and tree search.” In: *nature* 529.7587 (2016), pp. 484–489.
3. A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. “Improving language understanding by generative pre-training.” In: (2018).
4. S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation.” In: PloS one 10.7 (2015), e0130140.
5. M. Sundararajan, A. Taly, and Q. Yan. “Axiomatic attribution for deep networks.” In: International conference on machine learning. PMLR. 2017, pp. 3319–3328.
6. K. Simonyan, A. Vedaldi, and A. Zisserman. “Deep inside convolutional networks: Visualising image classification models and saliency maps.” In: arXiv preprint arXiv:1312.6034 (2013).
7. A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje. “Not just a black box: Learning important features through propagating activation differences.” In: arXiv preprint arXiv:1605.01713 (2016).
8. B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. “Learning deep features for discriminative localization.” In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, pp. 2921–29Lime29.
9. M. T. Ribeiro, S. Singh, and C. Guestrin. “" Why should i trust you?" Explaining the predictions of any classifier.” In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016, pp. 1135-1144.
10. S. M. Lundberg and S.-I. Lee. “A unified approach to interpreting model predictions.” In: Advances in neural information processing systems 30 (2017).

Sources

11. J. Kuka˘cka, V. Golkov, and D. Cremers. “Regularization for deep learning: Ataxonomy.” In: arXiv preprint arXiv:1710.10686 (2017).
12. C. M. Bishop et al. Neural networks for pattern recognition. Oxford university press, 1995.
13. I. J. Goodfellow, J. Shlens, and C. Szegedy. “Explaining and harnessing adversarial examples.” In: arXiv preprint arXiv:1412.6572 (2014).
14. C. Shorten and T. M. Khoshgoftaar. “A survey on image data augmentation for deep learning.” In: Journal of big data 6.1 (2019), pp. 1–48.
15. C. Tomani and D. Cremers. “CHALLENGER: Training with Attribution Maps.” In: arXiv preprint arXiv:2205.15094 (2022).
16. D. Dua and C. Graff. UCI Machine Learning Repository. 2017.
17. A. Krizhevsky, G. Hinton, et al. “Learning multiple layers of features from tiny images.” In: (2009).
18. K. Simonyan and A. Zisserman. “Very deep convolutional networks for largescale image recognition.” In: arXiv preprint arXiv:1409.1556 (2014).
19. D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. “Smoothgrad: removing noise by adding noise.” In: arXiv preprint arXiv:1706.03825 (2017)
20. L. Deng. “The mnist database of handwritten digit images for machine learning research.” In: IEEE Signal Processing Magazine 29.6 (2012), pp. 141–142.
21. Y. Le and X. Yang. “Tiny imagenet visual recognition challenge.” In: CS 231N 7.7 (2015), p. 3.
22. P. Wagner, N. Strodthoff, R.-D. Bousseljot, D. Kreiseler, F. I. Lunze, W. Samek, and T. Schaeffter. “PTB-XL, a large publicly available electrocardiography dataset.” In: Scientific data 7.1 (2020), p. 154.
23. K. He, X. Zhang, S. Ren, and J. Sun. “Deep residual learning for image recognition.” In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, pp. 770–778.
24. M. Tan and Q. Le. “Efficientnet: Rethinking model scaling for convolutional neural networks.” In: International conference on machine learning. PMLR. 2019, pp. 6105–6114.
25. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. “Attention is all you need.” In: Advances in neural information processing systems 30 (2017).

Additional Resources

Attribution Methods

- Completeness: non-zero attribution for differing predictions
- Implementation invariance: equal attributions for networks with equal outputs
- Linearity: Combine model weights => combine attributions

LRP: Backward pass has special operations.
Integrated Gradients: Gradients computed in between baseline and real input
⇒ attributions averaged over the steps
LIME: Local surrogate model (interpretable architecture) trained with samples containing a fraction of non-zero input elements each.

$$\xi(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

SHAP: Assign contribution of each input feature, mean marginal contribution of each input factor.
Regularly very hard to compute => techniques like Kernel SHAP that is pretty similar to LIME.
CAM: Constructs CNN without FC layers to help in explaining model.
Variants work with gradients but are still CNN only.

DeconvNet, Guided Backpropagation same as vanilla with different backward pass (similar to LRP)

Regularization Techniques

Data: Noise Injection, Adversarial Training, Hand-crafted data augmentation, attribution based augmentation, label smoothing.
Architecture: Batch normalization, ensembles, dropout.
Optimization: Early stopping, transfer learning, weight decay. (Term: feature diversity, L1, L2...)

Other Foundations

- **Overfitting and generalization**: Overfitting on data is the model learning relations in the data, that do not generalize, which can be understood as overfitting.
  Problems with low generalization can occur without overfitting though,
  if the density of the data distribution is not dense enough in that area for example.
- **Domain definition**: Broad areas encompassing data types and tasks.
  Eg. Computer Vision, NLP, Time Series.

Framework

Example challenger adjustment set and indexing:
A = {[1, 0, 0], [−1, 0, 0], [0, 0, 1], [0, 0, −1], [0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0]} or
A = {[[1, 0, 0], 1/8], [[−1, 0, 0], 1/8], [[0, 0, 1], 1/8], [[0, 0, −1], 1/8], [[0, 0, 0], 4/8]}
A = {3, 4, 5, 6, 0, 0, 0, 0}

Definition Delta, Neutral and Epsilon:
Definite predictions on loss, class scores and accuracy are impossible for most settings, based on adjustment group.
Reason: Attribution target and method often do not relate directly to these.
The result is that the definition of the groups is only based on tendencies.
Adjustment groups are based on effect:
=> Adjustment group reverses between cross entropy and class score as attribution target.

Relation to other Regularization

Noise Injection: One category case with fixed values would be similar to noise injection on inputs, but not sensible (attribution map calculated for nothing). Concept included in randomization though.

Adversarial Training: FGSM for adversarial training is entailed in ABA framework.
Special case of vanilla gradient targeting CE loss, 2 category split with cutoff at 0
and adjustment set only containing the epsilon adjustment A = {[-1,1]}

Dropout: Setting in augmentation step where categories are set to 0,
can be interpreted as attribution guided droupout

Future Work

- Best attribution method? Gradient x Input particularly promising.
- Augment not only input features.
- Check attribution guided dropout.
- Exact mechanism of tension.
- Non-classification.
- More analysis on non-vision datasets.
- Check on adversarial defense of ABA trained models.

Adjustments in Isolation

Tiny MNIST

Low	Middle	High	Type	Offset	Low	High	Index	MAX 5 runs	AVG 5 runs	Rating
0	0	0	Neutral	0	0	0	0	68.39	**66.94**	Baseline
0	1	0	Neutral	1	-1	-1	1	82.09	**79.52**	Top
0	-1	0	Neutral	-1	1	1	2	80.59	**77.24**	Good
0	0	1	Epsilon	0	0	1	3	82.92	**78.04**	Good
0	0	-1	Delta	0	0	-1	4	77.87	**74.84**	Good
1	0	0	Delta	0	1	0	5	73.21	**67.29**	Medium
-1	0	0	Epsilon	0	-1	0	6	85.91	**79.50**	Top
0	1	1	Epsilon	1	-1	0	7	83.96	**81.19**	Top
0	-1	1	Epsilon	-1	1	2	8	83.22	**80.49**	Top
0	1	-1	Delta	1	-1	-2	9	78.65	**72.50**	Good
0	-1	-1	Delta	-1	1	0	10	76.57	**73.48**	Good
1	1	0	Delta	1	0	-1	11	79.42	**70.79**	Medium
1	-1	0	Delta	-1	2	1	12	79.45	**76.84**	Good
-1	1	0	Epsilon	1	-2	-1	13	83.25	**80.18**	Top
-1	-1	0	Epsilon	-1	0	1	14	82.00	**81.08**	Top
1	0	1	Neutral	0	1	1	15	77.31	**74.98**	Good
1	0	-1	Delta	0	1	-1	16	82.64	**77.38**	Good
-1	0	1	Epsilon	0	-1	1	17	85.08	**83.94**	Top
-1	0	-1	Neutral	0	-1	-1	18	81.01	**77.47**	Good
1	1	1	Neutral	1	0	0	19	76.34	**69.56**	Medium
1	-1	1	Neutral	-1	2	2	20	80.90	**77.10**	Good
1	1	-1	Delta	1	0	-2	21	77.31	**74.52**	Good
1	-1	-1	Delta	-1	2	0	22	79.47	**73.26**	Good
-1	1	1	Epsilon	1	-2	0	23	82.95	**80.99**	Top
-1	-1	1	Epsilon	-1	0	2	24	85.10	**82.03**	Top
-1	1	-1	Neutral	1	-2	-2	25	82.71	**78.20**	Good
-1	-1	-1	Neutral	-1	0	0	26	77.40	**70.28**	Medium

Tiny CIFAR10

Low	Middle	High	Type	Index	MAX 5 runs	AVG 5 runs	Rating
0	0	0	Neutral	0	36.16	**34.73**	Baseline
0	1	0	Neutral	1	39.91	**38.73**	Top
0	-1	0	Neutral	2	37.09	**36.49**	Good
0	0	1	Epsilon	3	36.49	**35.01**	Medium
0	0	-1	Delta	4	36.13	**33.77**	Medium
1	0	0	Delta	5	36.00	**34.73**	Medium
-1	0	0	Epsilon	6	38.25	**34.53**	Medium
0	1	1	Epsilon	7	39.39	**38.62**	Top
0	-1	1	Epsilon	8	36.17	**35.03**	Medium
0	1	-1	Delta	9	36.21	**34.21**	Medium
0	-1	-1	Delta	10	34.73	**33.21**	Bad
1	1	0	Delta	11	37.84	**34.23**	Medium
1	-1	0	Delta	12	34.63	**33.89**	Medium
-1	1	0	Epsilon	13	39.19	**37.90**	Top
-1	-1	0	Epsilon	14	36.78	**36.03**	Good
1	0	1	Neutral	15	37.35	**36.76**	Good
1	0	-1	Delta	16	39.30	**35.96**	Good
-1	0	1	Epsilon	17	37.44	**36.11**	Good
-1	0	-1	Neutral	18	37.11	**35.86**	Good
1	1	1	Neutral	19	37.95	**36.14**	Good
1	-1	1	Neutral	20	36.93	**36.74**	Good
1	1	-1	Delta	21	37.20	**36.10**	Good
1	-1	-1	Delta	22	37.29	**34.38**	Medium
-1	1	1	Epsilon	23	39.30	**37.53**	Top
-1	-1	1	Epsilon	24	36.17	**35.33**	Medium
-1	1	-1	Neutral	25	38.65	**37.89**	Top
-1	-1	-1	Neutral	26	35.78	**33.78**	Bad

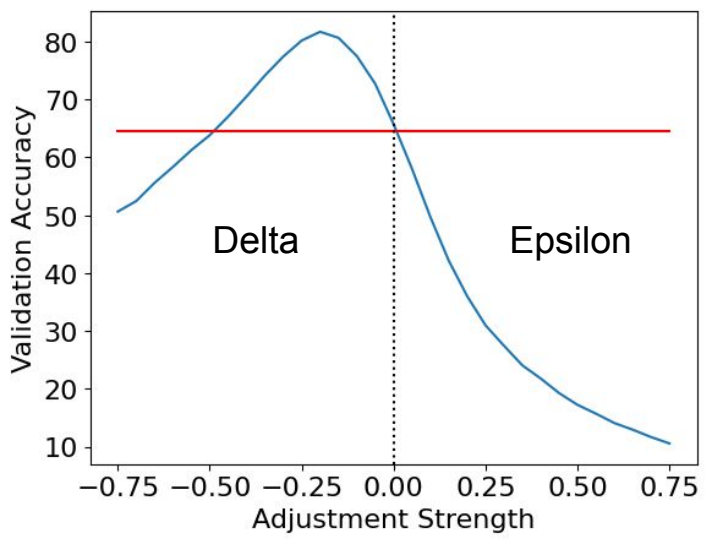
MNIST Hypothesis Experiment

Tiny MNIST theory test

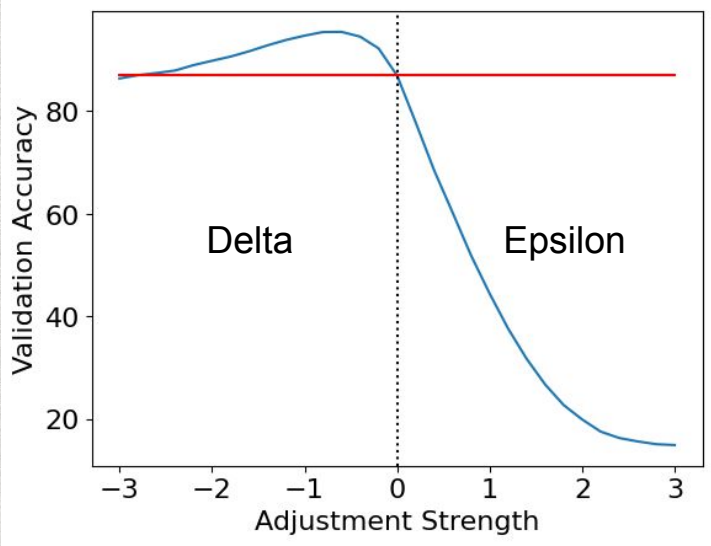
Adj. group	All of group	AVG of groups	Difference
Neutral	76.82	74.59	2.23
Delta	76.64	73.43	3.21
Epsilon	83.51	80.83	2.68
Eps. & Neu.	82.14	77.71	4.43
Eps. & Del.	84.55	77.13	7.42
All	83.72	76.28	7.44

MNIST Frozen and Adversarial Defense

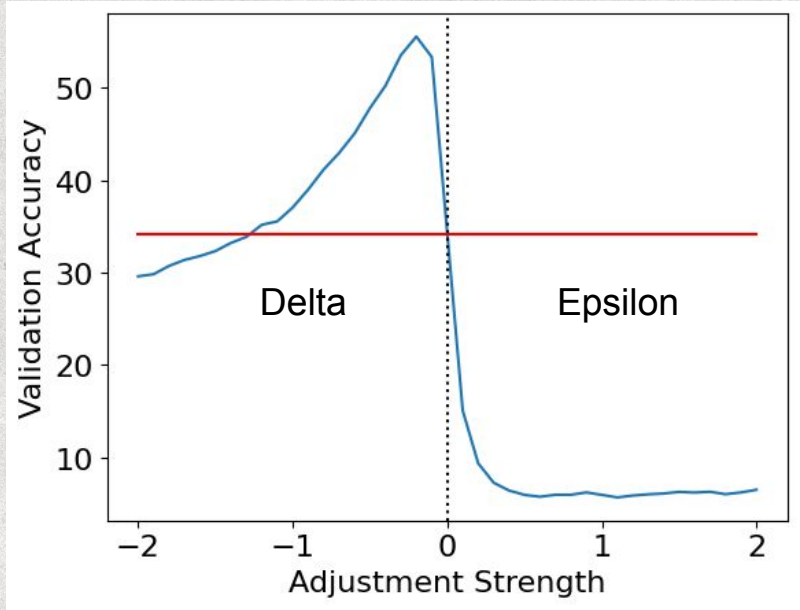
Frozen VGG11 on tiny MNIST



Frozen VGG11 on tiny MNIST, trained with ABA



CIFAR10 frozen



Bench Full

Dataset			Vanilla		Challenger			MendABA			FastABA			FastABA-CE		
No BN, val	Subset	Epochs	median	std	median	std	strength	median	std	strength	median	std	strength	median	std	strength
MNIST	Tiny (100)	1000	63.19	9.27	85.00	1.02	2.00	84.75	1.33	1.00	84.54	2.32	1.00	85.04	1.71	1.00
k=4	Med (1000)	100	92.38	0.90	96.14	0.43	1.00	95.97	0.45	1.00	96.44	0.29	1.00	95.26	0.45	1.00
	Full	10	99.19	0.06	99.25	0.08	0.50	99.24	0.08	0.50	99.25	0.07	0.50	99.29	0.07	0.20
CIFAR	Tiny (1000)	500	34.73	1.57	38.76	2.72	0.20	41.22	1.22	0.50	39.60	0.58	1.00	39.91	0.74	0.50
k=4	Full	30	74.93	0.70	75.45	0.73	0.20	75.56	0.65	0.20	75.01	0.55	0.10	74.83	0.47	0.05
NEWS	Tiny (2000)	100	47.20	2.03	58.40	1.77	0.10	60.30	1.44	0.10	59.10	0.53	0.10	58.60	1.12	0.10
k=6	Full	30	67.90	1.88	74.00	0.81	0.05	74.00	1.09	0.05	73.70	1.01	0.05	73.70	1.00	0.02
No BN, test	Subset	Epochs	median	std	median	std	strength	median	std	strength	median	std	strength	median	std	strength
MNIST	Tiny (100)	1000	63.09	8.91	83.12	1.12	2.00	82.84	1.28	1.00	84.65	1.87	1.00	82.23	2.22	1.00
k=4	Med (1000)	100	91.62	1.08	96.03	0.41	1.00	95.88	0.58	1.00	96.20	0.21	1.00	94.94	0.37	1.00
	Full	10	99.07	0.07	99.24	0.06	0.50	99.23	0.06	0.50	99.21	0.06	0.50	99.32	0.08	0.20
CIFAR	Tiny (1000)	500	33.53	0.94	38.39	1.80	0.20	41.01	1.34	0.50	38.51	0.90	1.00	39.69	0.74	0.50
k=4	Full	30	73.83	0.82	74.53	0.62	0.20	74.77	0.50	0.20	74.71	0.73	0.10	74.22	0.36	0.05
NEWS	Tiny (2000)	100	46.23	1.46	57.25	2.02	0.10	60.54	1.40	0.10	58.09	0.86	0.10	57.98	0.97	0.10
k=6	Full	30	66.73	1.29	72.79	1.03	0.05	72.93	1.25	0.05	73.04	1.15	0.05	72.91	1.30	0.02
BN, val	Subset	Epochs	median	std	median	std	strength	median	std	strength	median	std	strength	median	std	strength
MNIST	Tiny (100)	500	84.63	1.52	85.48	1.55	0.10	85.74	0.95	0.10	86.28	0.81	0.50	86.20	0.79	0.50
k=4	Med (1000)	50	96.63	1.24	96.61	0.44	0.05	96.97	0.17	0.05	96.84	0.36	0.10	96.76	0.40	0.10
	Full	10	99.22	0.06	99.28	0.07	0.02	99.29	0.07	0.02	99.35	0.09	0.10	99.27	0.09	0.10
CIFAR	Tiny (1000)	200	46.18	0.89	46.80	0.64	0.01	46.32	0.88	0.01	46.50	0.90	0.01	46.69	0.70	0.01
k=4	Full	30	83.07	0.75	84.31	0.29	0.03	83.67	0.65	0.02	83.92	0.69	0.01	83.83	0.62	0.01
NEWS	Tiny (2000)	50	47.42	1.79	59.40	0.57	0.10	60.20	0.95	0.10	58.60	1.60	0.10	58.30	0.99	0.05
k=6	Full	30	67.00	1.45	75.30	1.09	0.05	76.00	0.38	0.05	75.70	0.61	0.05	75.20	0.69	0.02
BN, test	Subset	Epochs	median	std	median	std	strength	median	std	strength	median	std	strength	median	std	strength
MNIST	Tiny (100)	500	83.25	1.79	83.43	1.56	0.10	83.59	1.23	0.10	85.4	1.44	0.50	84.43	1.13	0.50
k=4	Med (1000)	50	96.72	1.55	96.63	0.38	0.05	96.91	0.19	0.05	96.93	0.41	0.10	96.68	0.34	0.10
	Full	10	99.22	0.05	99.30	0.08	0.02	99.33	0.07	0.02	99.31	0.12	0.10	99.29	0.07	0.10
CIFAR	Tiny (1000)	200	45.55	0.95	45.93	0.43	0.01	45.47	0.69	0.01	45.40	0.89	0.01	45.79	0.74	0.01
k=4	Full	30	82.49	0.66	83.73	0.49	0.03	83.35	0.38	0.02	83.46	0.50	0.01	83.32	0.41	0.01
NEWS	Tiny (2000)	50	46.89	2.60	57.86	0.89	0.10	58.97	0.90	0.10	57.45	1.20	0.10	57.53	1.55	0.05
k=6	Full	30	65.75	1.35	74.24	1.36	0.05	74.81	0.43	0.05	74.43	0.76	0.05	73.95	0.67	0.02

Bench Full

			Vanilla		Challenger			MendABA			FastABA		
Tiny ImageNet	RandAug	Epochs	median	std	median	std	strength	median	std	strength	median	std	strength
ResNet50	No	100	31.31	0.81	32.74	0.28	0.100	32.86	0.90	0.100	33.13	0.49	0.030
ResNet50	Yes	100	46.93	1.44	45.75	0.80	0.100	47.98	2.81	0.100	46.71	1.11	0.030
EfficientB3	No	300	41.33	0.60	40.94	0.92	0.030	42.11	0.83	0.030	42.18	0.64	0.030
EfficientB3	Yes	300	53.59	0.79	53.29	0.55	0.003	53.66	0.61	0.003	53.63	0.47	0.003
ViT Base 16	No	50	28.20	0.72	29.95	0.66	0.100	31.77	0.74	0.100	33.01	0.42	0.100

	Epochs	median	std	median	std	strength	median	std	strength	median	std	strength
PTB-XL	50	91.40	0.26	91.50	0.36	0.100	91.50	0.40	0.100	91.40	0.28	0.030

Tension Predicted Mechanism of Action

Interaction with optimization landscape