Using Reed-Muller Codes for Classification with Rejection and Recovery

International Symposium on Foundations & Practice of Security - 2023

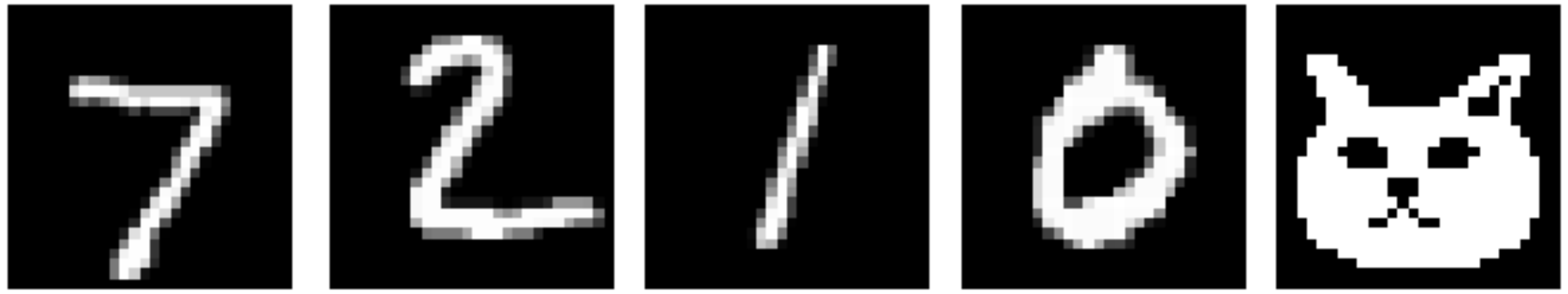
Daniel Fentham, Dave Parker, Mark Ryan

UNIVERSITY OF BIRMINGHAM

UNIVERSITY OF OXFORD

Classification Problems

- Real-world classifiers are expected to respond intelligently to any user input.
- However, there are many instances where this is not the case.

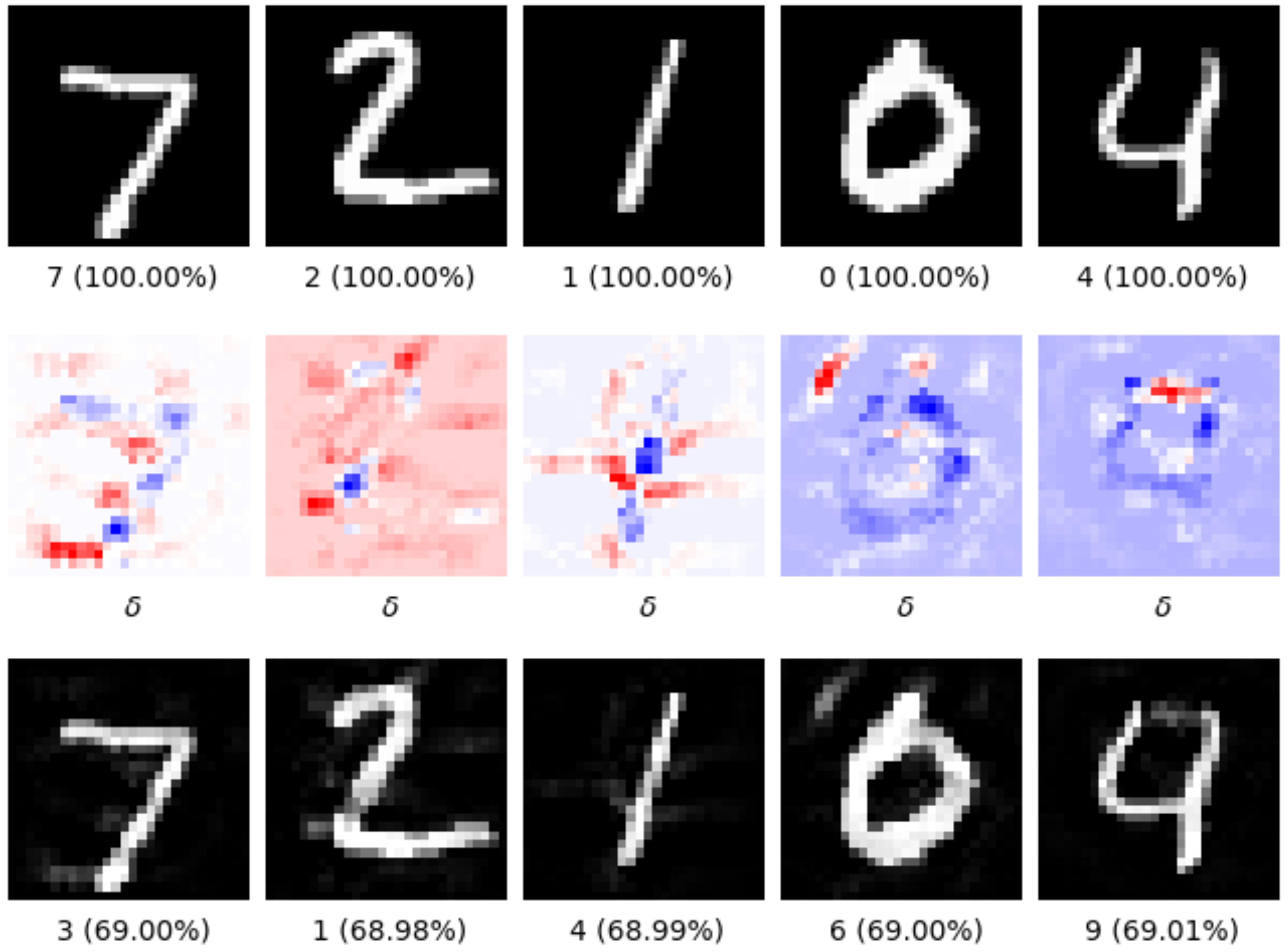


7 (100%) 2 (100%) 1 (100%) 0 (100%) 8 (98.58%)

- Neural Networks (NNs) are often forced to give a response from the limited set of class labels for Out-of-Distribution (OoD) inputs.

Adversarial Examples

- Adversarial examples are a related phenomenon.
- We apply calculated perturbations to clean inputs to push them out of the *'natural distribution'*.



- This can lead to incorrect and potentially dangerous results.

A Short Aside on Adversaries

- Adversarial attacks come in two primary flavours:
  - **Open-box** - The adversary has full access to the model (architecture and weights).
  - **Closed-box** - The adversary only has access to the final class.
- Open-box attacks use gradient-based approaches to generate adversaries.
- Closed-box attacks usually involve creating a surrogate model, which we use to generate adversarial examples. Due to *transferability*, these adversaries are effective at fooling the target model.
- Adversaries can be generated using different norms:
  - L_∞ - Encourage perturbations which focus on a single feature.
  - L_2 - Encourage perturbations across multiple features.

The Common Thread

- It is important to make a distinction between the types of OoD data:
    - **Artificially** OoD data (adversarial examples)
    - **Naturally** OoD data (an input the NN has not been trained to classify)
- Both artificial and natural OoD data lie off the distribution the NN was trained on.
- Within these regions, the NN must make unconstrained extrapolations to classify the data.
- Ideally, a NN would never be asked to make the classifications.
- How can we improve?

Classification with Rejection

- **Classification with Rejection** (CwR) aims to solve this problem by giving the NN the option to refuse to classify an input it believes is OoD.
- This approach works well for naturally OoD data.
- This approach works less well for artificially OoD data:
    - Adversaries tend to be rejected.
    - In some applications we may want to classify the rejected inputs.

Reed-Muller Aggregation Networks

- The **Reed-Muller Aggregation Network** (RMAggNet) aims to solve the problem by giving the network the option to refuse to classify an input it believes is OoD *after trying to correct it*.

- This method:
  - Splits classification over n networks, generating n independent sub-tasks.
  - Aggregates the results.
  - Identifies and corrects inconsistencies in the classification using Reed-Muller codes.

- This is an approach that approximates distances to a learned manifold.

- Is similar to previous work but *focuses on correction*.

Reed-Muller Codes

- Are defined by two hyperparameters m and r (where $m \geq r$)
- Using m and r we can determine:
  - $n = 2^m$ — The codeword length/number of networks we aggregate over.
  - $d = 2^{m-r}$ — The (guaranteed) minimum Hamming distance between any two codes.
  - $k = \Sigma_{i=0}^{r} \binom{m}{i}$ — The message length we can encode.
- Reed-Muller code notation is often in the form $[n, k, d]_2$

Selecting m and r

- There are a number of related factors to consider when deciding on appropriate values for m and r:
  - We need to have enough class codewords for the dataset, such that $|C| \leq 2^k$, and the probability of assigning a valid class to random noise is low (a low value for $\frac{|C|}{2^n}$).
  - We have appropriate error correction $t = \left\lfloor \frac{d-1}{2} \right\rfloor$.
  - We can feasibly train $n = 2^m$ networks and use them for inference.

Generating Reed-Muller codes

For a codeword length of n we generate codes with 2^{m-1} 0s followed by 2^{m-1} 1s, then 2^{m-2} etc., stopping at $2^{m-x} = 1$.

For example, with $m = 3$:

x_0 : 00000000	$x_1 \oplus x_2$: 11110000	$x_3 \oplus x_4$: 01100110	$x_1 \oplus x_2 \oplus x_3 \oplus x_4$: 10010110
x_1 : 11111111	$x_1 \oplus x_3$: 11001100	$x_1 \oplus x_2 \oplus x_3$: 11000011	
x_2 : 00001111	$x_1 \oplus x_4$: 10101010	$x_1 \oplus x_2 \oplus x_4$: 10100101	
x_3 : 00110011	$x_2 \oplus x_3$: 00111100	$x_1 \oplus x_3 \oplus x_4$: 10011001	
x_4 : 01010101	$x_2 \oplus x_4$: 01011010	$x_2 \oplus x_3 \oplus x_4$: 01101001	

This defines a closed set, which allows us to make guarantees about the Hamming distance between codewords.

Example

- If we have a dataset with $|C| = 10$ classes, we can define a Reed-Muller code with $m = 3$, $r = 1$ giving us an $[8,4,4]_2$ code.
- We can generate the Reed-Muller codes with this specification and take 10 of the class codewords.

	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8
0	0	0	0	0	1	1	1	1
1	0	0	1	1	0	0	1	1
2	0	1	0	1	0	1	0	1
3	1	1	1	1	0	0	0	0
4	1	1	0	0	1	1	0	0
5	1	0	1	0	1	0	1	0
6	0	0	1	1	1	1	0	0
7	0	1	0	1	1	0	1	0
8	0	1	1	0	0	1	1	0
9	1	1	0	0	0	0	1	1

- We train each network N_x (column) to return a 0 or 1 for each class as the rows indicate.
- Because this is an $[8,4,4]_2$ code, we can correct up to 1 bit.

Evaluation

- We compare RMAggNet to two other methods:
  - **CCAT** - A popular CwR method which uses adversarial training to return a uniform distribution for all OoD data.
  - **Ensemble** - Similar to RMAggNet except the individual networks learn the entire classification process, and there is no correction.
- We analyse the performance of these methods on the EMNIST and CIFAR-10 datasets consisting of clean in-distribution, and adversarial data.

EMNIST

Results on the clean EMNIST dataset (higher correctness is better).

CCAT			
τ	**Correct**	**Rejected**	**Incorrect**
0	**88.68**	0.00	11.32
0.10	**88.60**	0.15	11.24
0.20	**87.54**	2.41	10.05
0.30	**85.46**	6.45	8.09
0.40	**83.16**	10.29	6.55
0.50	**80.70**	14.22	5.08
0.60	**77.95**	18.03	4.02
0.70	**74.23**	22.71	3.06
0.80	**69.48**	28.46	2.06
0.90	**60.74**	38.05	1.20
1.0	**0.00**	100.00	0.00

Ensemble			
σ	**Correct**	**Rejected**	**Incorrect**
0	**89.78**	0.00	10.22
0.10	**89.78**	0.00	10.22
0.20	**89.78**	0.01	10.21
0.30	**89.76**	0.05	10.19
0.40	**89.70**	0.28	10.03
0.50	**89.20**	1.41	9.39
0.60	**87.76**	4.45	7.79
0.70	**85.94**	7.68	6.38
0.80	**83.89**	11.05	5.06
0.90	**80.60**	15.60	3.80
1.0	**68.34**	30.15	1.51

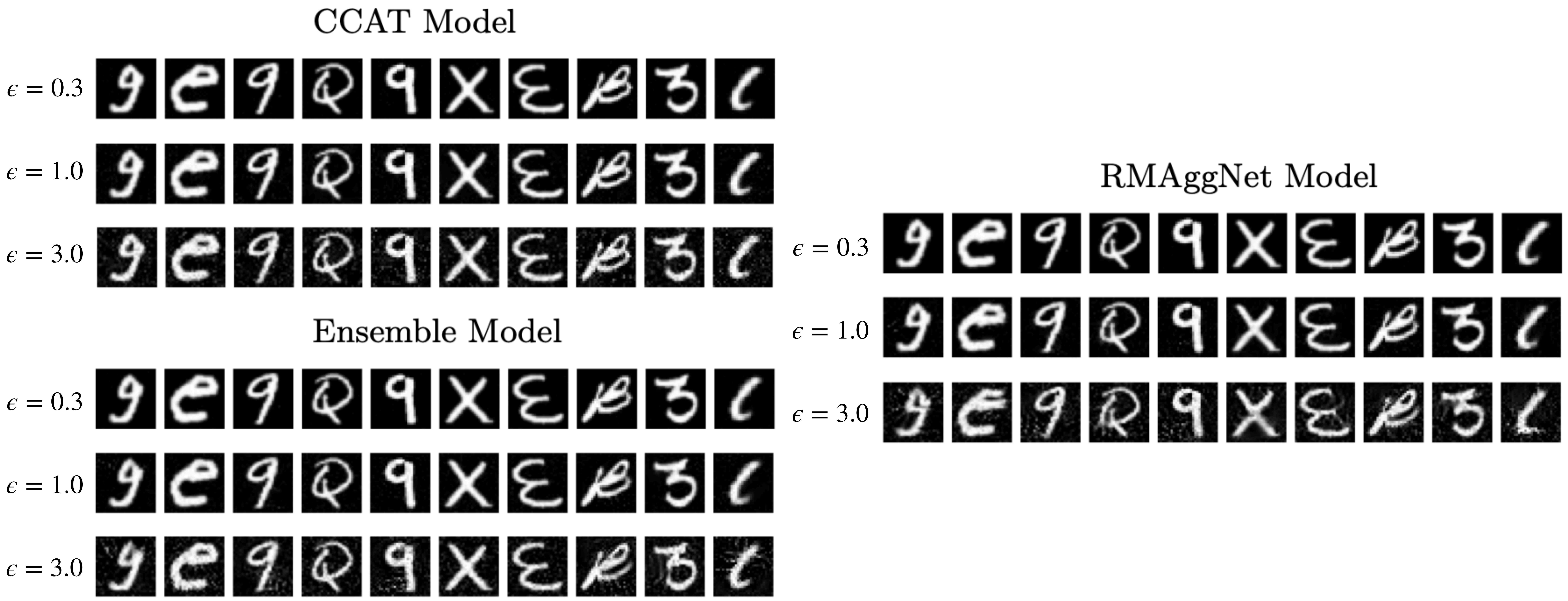
RMAggNet $[32, 6, 16]_2$			
EC	**Correct**	**Rejected**	**Incorrect**
7	**89.16**	2.14	8.70
6	**87.93**	4.59	7.48
5	**86.54**	6.98	6.48
4	**84.99**	9.49	5.52
3	**83.29**	12.03	4.68
2	**80.85**	15.15	4.00
1	**77.19**	19.59	3.23
0	**70.02**	27.72	2.26

EMNIST (Adversarial)

Results on an adversarial EMNIST dataset generated using a PGD open-box attack using the L_2 metric (lower incorrectness is better).

PGD(L_2)												
CCAT												
τ	ϵ	**Correct**	**Rejected**	**Incorrect**	ϵ	**Correct**	**Rejected**	**Incorrect**	ϵ	**Correct**	**Rejected**	**Incorrect**
0.00	0.30	0.20	0.00	**99.80**	1.0	0.00	0.00	**100.00**	3.0	0.00	0.00	**100.00**
0.30		0.00	100.00	**0.00**		0.00	100.00	**0.00**		0.00	100.00	**0.00**
0.70		0.00	100.00	**0.00**		0.00	100.00	**0.00**		0.00	100.00	**0.00**
0.90		0.00	100.00	**0.00**		0.00	100.00	**0.00**		0.00	100.00	**0.00**
Ensemble												
σ	ϵ	**Correct**	**Rejected**	**Incorrect**	ϵ	**Correct**	**Rejected**	**Incorrect**	ϵ	**Correct**	**Rejected**	**Incorrect**
0.00	0.30	84.80	0.00	**15.20**	1.0	62.40	0.00	**37.60**	3.0	19.60	0.00	**80.40**
0.30		84.80	0.00	**15.20**		62.40	0.00	**37.60**		19.60	0.00	**80.40**
0.70		79.50	9.10	**11.40**		57.10	13.00	**29.90**		19.50	0.20	**80.30**
1.00		60.60	35.90	**3.50**		47.40	39.80	**12.80**		18.50	9.80	**71.70**
RMAggNet												
EC	ϵ	**Correct**	**Rejected**	**Incorrect**	ϵ	**Correct**	**Rejected**	**Incorrect**	ϵ	**Correct**	**Rejected**	**Incorrect**
7	0.30	86.00	3.10	**10.90**	1.0	70.40	6.60	**23.00**	3.0	9.20	25.70	**65.10**
6		84.40	6.20	**9.40**		67.60	12.20	**20.20**		6.40	37.60	**56.00**
5		82.70	9.10	**8.20**		65.20	17.40	**17.40**		5.00	45.60	**49.40**
4		80.70	12.40	**6.90**		61.70	23.40	**14.90**		3.10	55.60	**41.30**
3		77.70	16.40	**5.90**		57.40	29.30	**13.30**		2.00	65.00	**33.00**
2		74.10	20.90	**5.00**		50.30	38.10	**11.60**		1.00	75.60	**23.40**
1		68.00	28.20	**3.80**		39.10	51.50	**9.40**		0.70	86.10	**13.20**
0		56.00	41.30	**2.70**		15.10	78.60	**6.30**		0.00	94.60	**5.40**

EMNIST (Adversarial)



CIFAR-10

Results on the clean CIFAR-10 dataset (higher correctness is better).

CCAT			
τ	**Correct**	**Rejected**	**Incorrect**
0	**76.41**	0.00	23.59
0.10	**76.41**	0.00	23.59
0.20	**76.21**	0.68	23.11
0.30	**75.10**	3.76	21.14
0.40	**72.62**	9.38	18.00
0.50	**68.59**	17.41	14.00
0.60	**64.04**	25.55	10.41
0.70	**58.81**	33.72	7.47
0.80	**52.81**	42.39	4.80
0.90	**44.64**	52.54	2.82
1.0	**1.09**	98.91	0.00

Ensemble			
σ	**Correct**	**Rejected**	**Incorrect**
0	**79.34**	0.00	20.66
0.10	**79.54**	0.00	20.46
0.20	**79.35**	0.00	20.65
0.30	**79.53**	0.02	20.45
0.40	**79.24**	1.05	19.71
0.50	**77.14**	6.73	16.13
0.60	**75.31**	11.23	13.46
0.70	**68.79**	22.62	8.59
0.80	**65.38**	28.00	6.62
0.90	**56.57**	40.23	3.20
1.0	**39.21**	59.83	0.96

RMAggNet $[16, 5, 8]_2$			
EC	**Correct**	**Rejected**	**Incorrect**
3	**77.11**	12.76	10.13
2	**68.32**	27.23	4.45
1	**57.90**	40.09	2.01
0	**42.46**	59.96	0.58

CIFAR-10 (Adversarial)

Results on an adversarial CIFAR-10 dataset generated using a PGD open-box attack using the L_2 metric (lower incorrectness is better).

PGD(L_2)												
CCAT												
τ	ϵ	**Correct**	**Rejected**	**Incorrect**	ϵ	**Correct**	**Rejected**	**Incorrect**	ϵ	**Correct**	**Rejected**	**Incorrect**
0.00	0.30	29.20	0.00	**70.80**	0.75	12.50	0.00	**87.50**	2.5	10.20	0.00	**89.80**
0.30		0.50	92.10	**7.40**		0.00	97.50	**2.50**		0.00	99.50	**0.50**
0.70		0.00	96.00	**4.00**		0.00	99.20	**0.80**		0.00	99.80	**0.20**
0.90		0.00	97.40	**2.60**		0.00	99.70	**0.30**		0.00	99.80	**0.20**
Ensemble												
σ	ϵ	**Correct**	**Rejected**	**Incorrect**	ϵ	**Correct**	**Rejected**	**Incorrect**	ϵ	**Correct**	**Rejected**	**Incorrect**
0.00	0.30	54.30	0.00	**45.70**	0.75	26.70	0.00	**73.30**	2.5	13.40	0.00	**86.60**
0.30		53.30	0.00	**46.70**		26.50	0.00	**73.50**		13.30	0.00	**86.70**
0.70		42.00	28.60	**29.40**		23.30	12.90	**63.80**		13.00	0.90	**86.10**
1.00		23.10	70.20	**6.70**		16.20	54.70	**29.10**		12.20	6.30	**81.50**
RMAggNet												
EC	ϵ	**Correct**	**Rejected**	**Incorrect**	ϵ	**Correct**	**Rejected**	**Incorrect**	ϵ	**Correct**	**Rejected**	**Incorrect**
3	0.30	64.00	15.30	**20.70**	0.75	39.80	17.00	**43.20**	2.5	11.10	10.80	**78.10**
2		52.40	36.30	**11.30**		32.90	37.90	**29.20**		9.60	23.20	**67.20**
1		41.20	52.10	**6.70**		25.60	54.90	**19.50**		8.20	41.80	**50.00**
0		28.70	69.10	**2.20**		19.40	71.70	**8.90**		5.60	63.00	**31.40**

CIFAR-10 (Adversarial)

CCAT Model

Ensemble Model

RMAggNet Model

Summary

- RMAggNet can effectively recover correct classifications from adversarial images for small ϵ, reducing the number of rejected images compared to CCAT.
- Experimentally, it has shown better performance than Ensemble methods in many situations.
- RMAggNet has applications in CwR systems where we aim to reduce the reliance on downstream error handling.
- This error correction framework provides an interesting approach to measuring distances from the learned data manifold.
- Next up: Expand to larger datasets and check model independence!

Thank you!

Code available at: https://github.com/dfenth/RMAggNet