Bachelor of Science
Thesis Defence

Formal Verification of Deep Neural Networks for Sentiment Classification

Paulius Skaisgiris
Supervisor: Dr. Pieter Collins

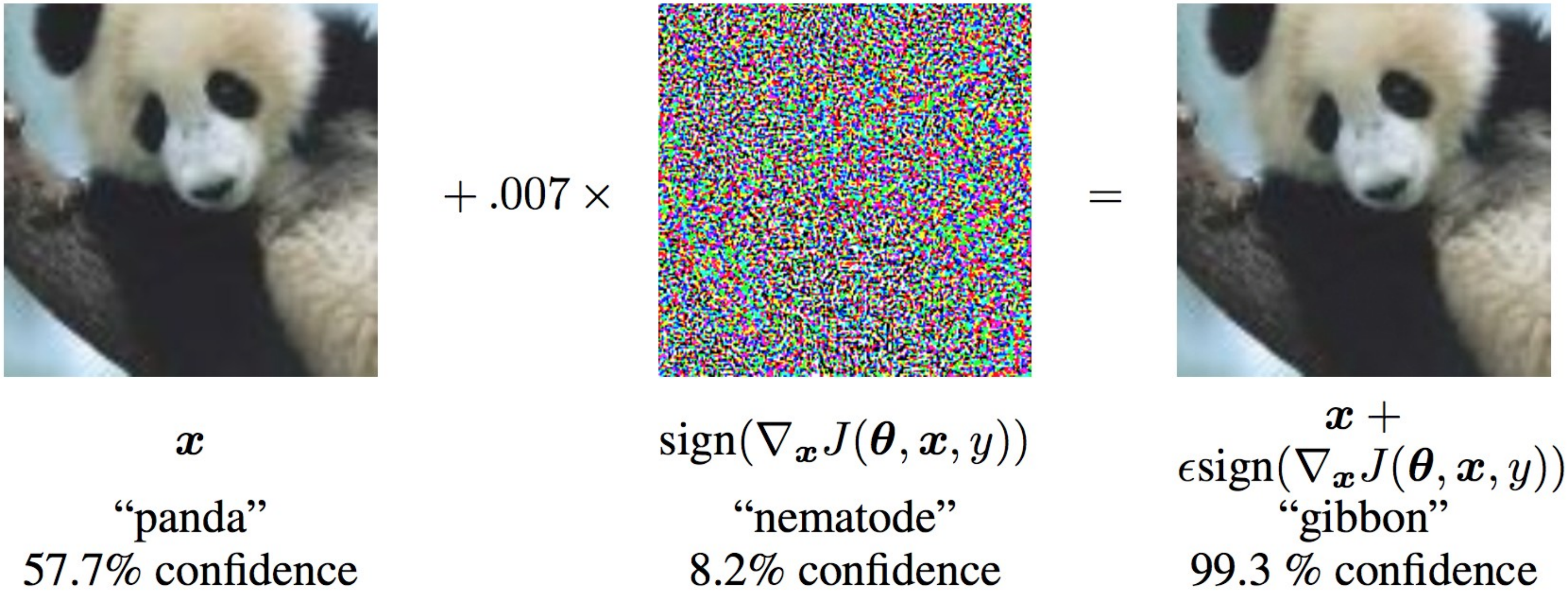
Department of Data Science and Knowledge Engineering

Maastricht University

What is the problem?

- Deep neural networks (DNNs) are a strong method of modelling data
- Reported to achieve superhuman performance across a wide variety of tasks

What is the problem?



[1] Szegedy et al. 2014 "Intriguing properties of Neural Networks"

What is the problem?

Prediction	SST word-level examples (by exhaustive verification, not by adversarial attack)
+	it ’ s the kind of pigeonhole-resisting romp that hollywood too rarely provides .
-	it ’ s the kind of pigeonhole-resisting romp that hollywood too rarely **gives** .
-	sets up a nice concept for its fiftysomething leading ladies , but fails loudly in execution .
+	sets up a nice concept for its fiftysomething leading ladies , but fails **aloud** in execution .
Prediction	**SST character level examples (by exhaustive verification, not by adversarial attack)**
-	you ’ ve seen them a million times .
+	you ’ ve se**r**n them a million times .
+	choose your reaction : a.) that sure is funny !
-	choose **t**our reaction : a.) that sure is funny !

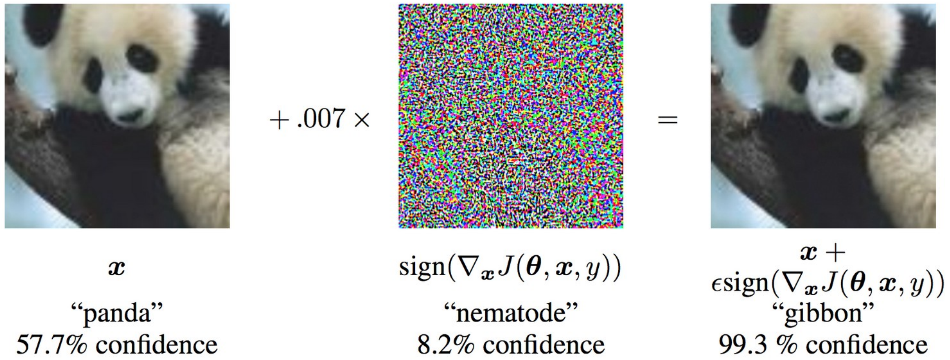
[2] Huang et al. 2019 “Achieving verified robustness to symbol substitution”

How can we solve this?

- Prove that the deep neural network is **robust** against adversarial examples **using formal verification**
- I aim to explore perturbations in the latent/encoding space

Background: Robustness

- For slight perturbations/alterations of the input *that are indistinguishable to a human*, will the model’s output stay correct?



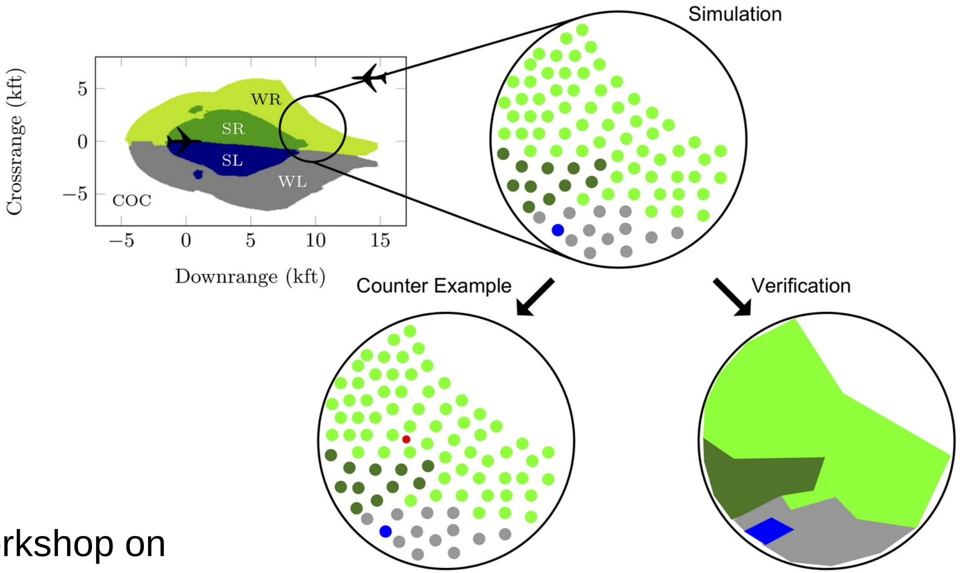
[1] Szegedy et al. 2014 “Intriguing properties of Neural Networks”

Background: Verification

- Given a program *P* and property *φ*, does *P* satisfy *φ*?
  - Option 1: *prove* that property *φ* holds
  - Option 2: provide a *counter-example* showing that it does not

Background: Verification

- Stronger guarantees than testing: holds for *any* possible input
  - Not just a finite set that was tested



[3] Liu et al. 2019 CARS Workshop on NeuralVerification.jl

Research questions

Research questions

- How can we formulate the robustness property for the sentiment classification problem in a form which is amenable for formal verification analysis?

Research questions

- How can we formulate the robustness property for the sentiment classification problem in a form which is amenable for formal verification analysis?
- What complexity of feedforward neural networks for the sentiment classification problem can be verified by existing verification tools?

Research questions

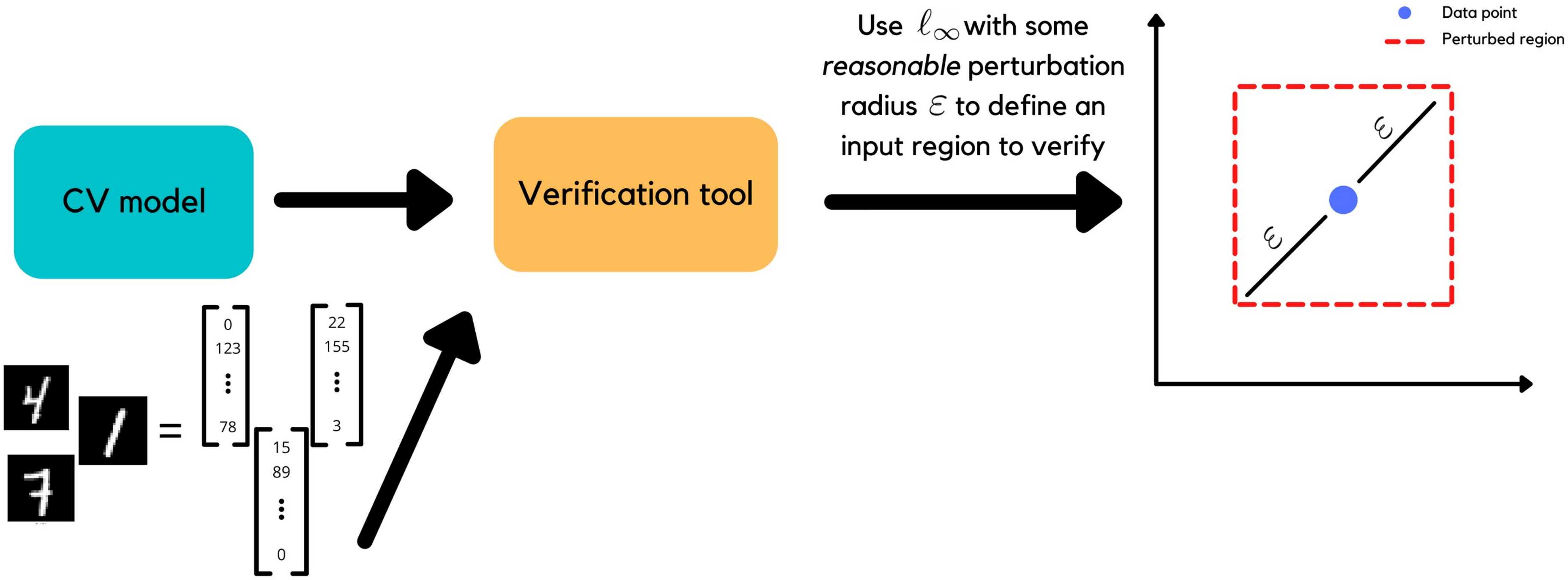
- How can we formulate the robustness property for the sentiment classification problem in a form which is amenable for formal verification analysis?
- What complexity of feedforward neural networks for the sentiment classification problem can be verified by existing verification tools?
- Do neural networks with piecewise-linearly approximated activation functions perform just as well and is training them equally efficient as with smooth activation functions?

Research questions

- How can we formulate the robustness property for the sentiment classification problem in a form which is amenable for formal verification analysis?

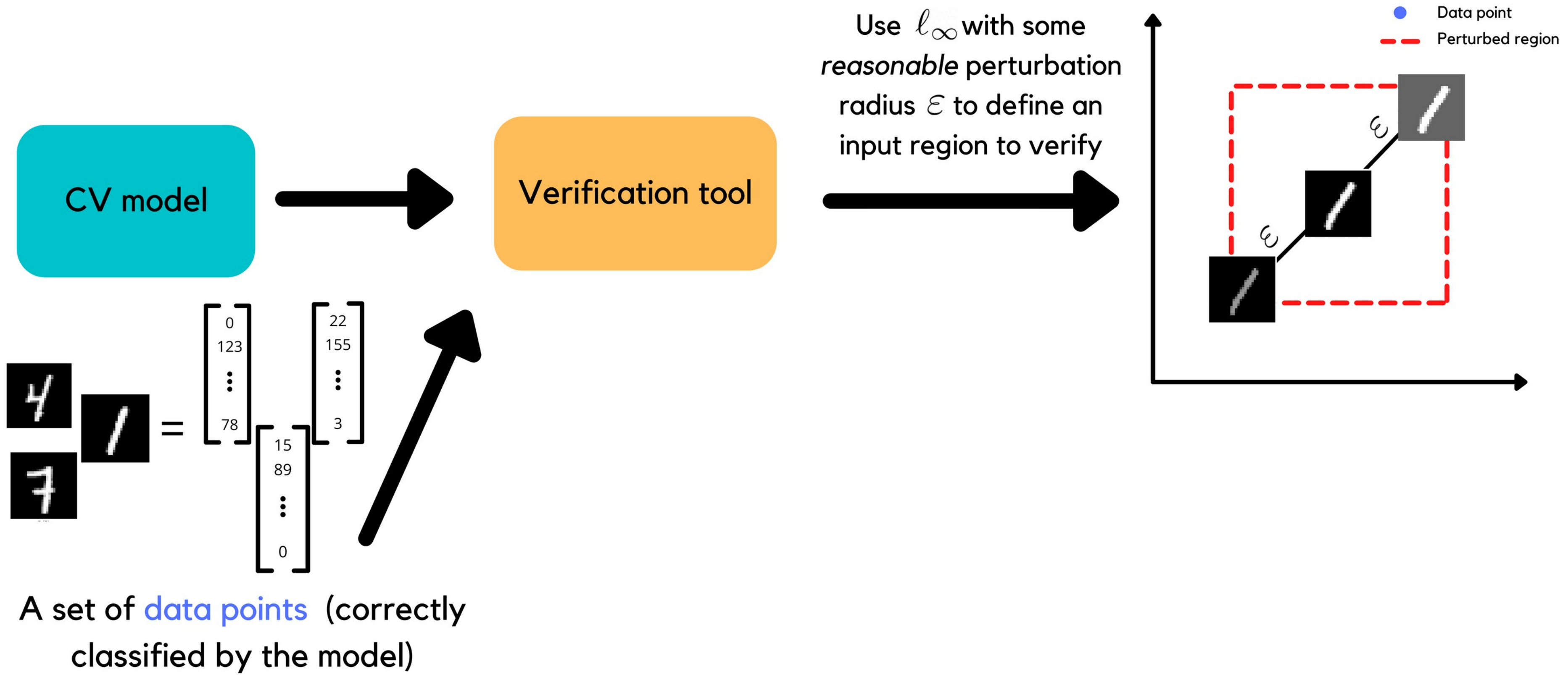
Methods

Robustness verification for CV



A set of data points (correctly classified by the model)

Robustness verification for CV

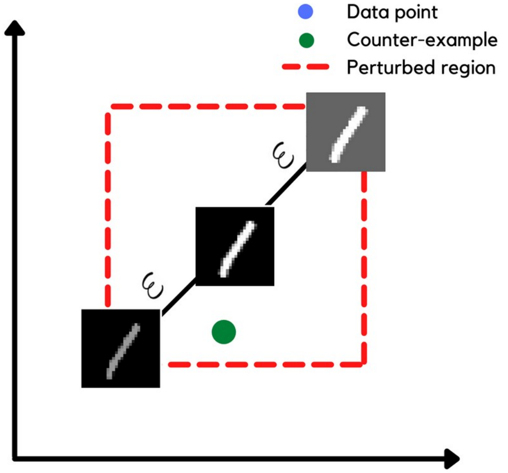


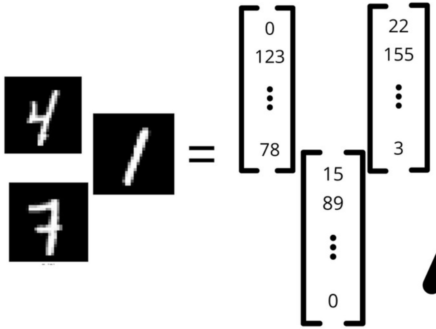
Robustness verification for CV

CV model

Verification tool

Use ℓ_∞ with some *reasonable* perturbation radius ε to define an input region to verify



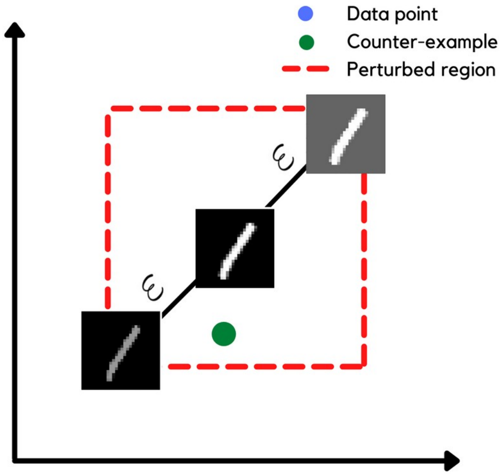


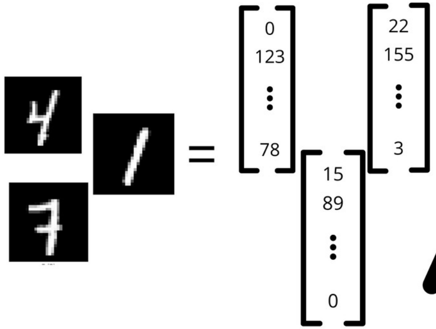
A set of data points (correctly classified by the model)

Robustness verification for CV

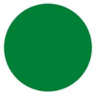
CV model → Verification tool

Use ℓ_∞ with some *reasonable* perturbation radius ε to define an input region to verify

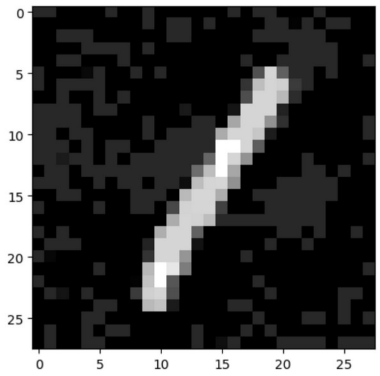




A set of data points (correctly classified by the model)

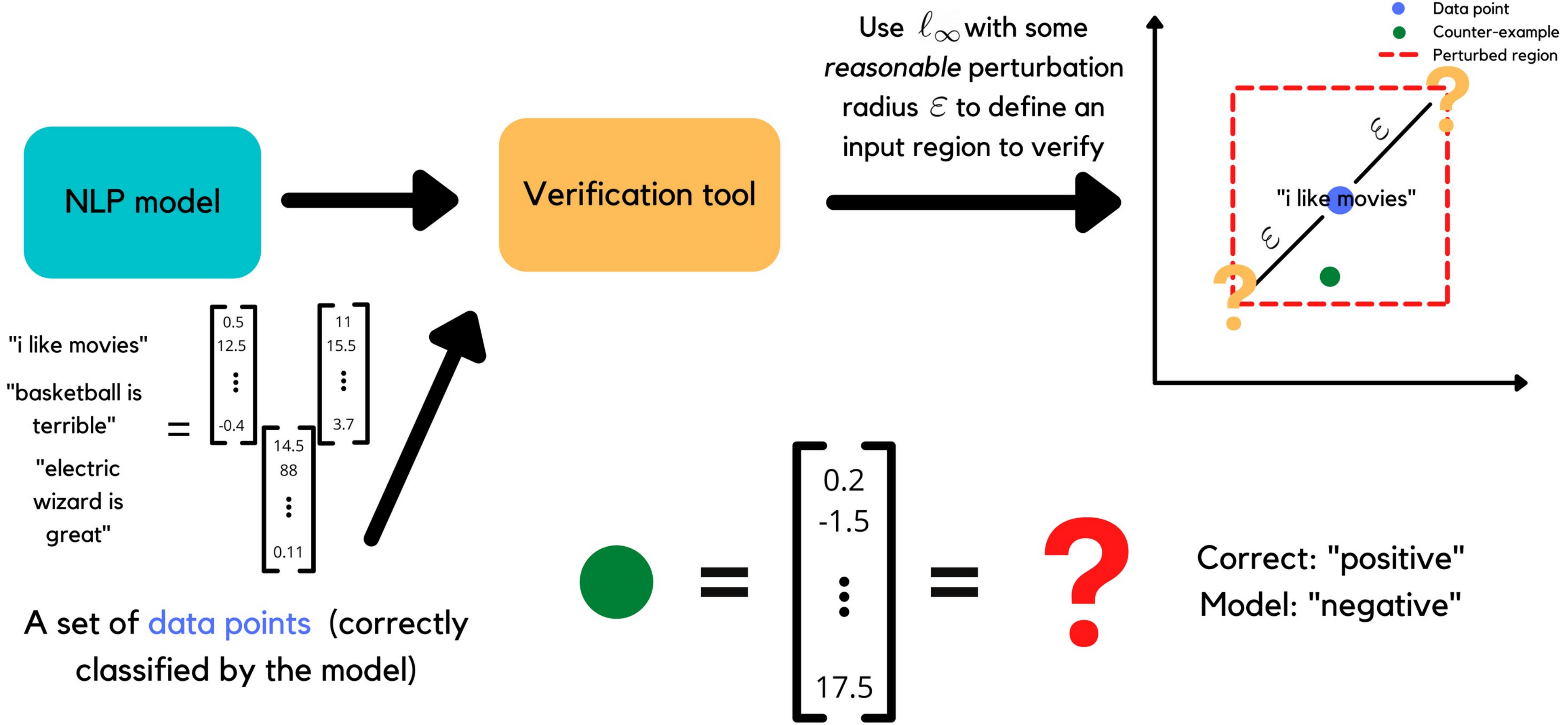


$$= \begin{bmatrix} 122 \\ 42 \\ \vdots \\ 255 \end{bmatrix} =$$



Correct: "1"
Model: "8"

Problems for NLP robustness interpretation



Methods for NLP robustness interpretation

- Are semantically similar sentences in the latent space nearby each other by Chebyshev distance?

Methods for NLP robustness interpretation

- Are semantically similar sentences in the latent space nearby each other by Chebyshev distance?
  - Try different text encoding methods: GloVe, FastText, Doc2Vec, USE, InferSent, DistilRoBERTa. Even better: autoencoders?

Methods for NLP robustness interpretation

- Are semantically similar sentences in the latent space nearby each other by Chebyshev distance?
- The perturbation radius ε is no longer interpretable

Methods for NLP robustness interpretation

- Are semantically similar sentences in the latent space nearby each other by Chebyshev distance?
- The perturbation radius ε is no longer interpretable
  - Try different values, try nearest neighbours

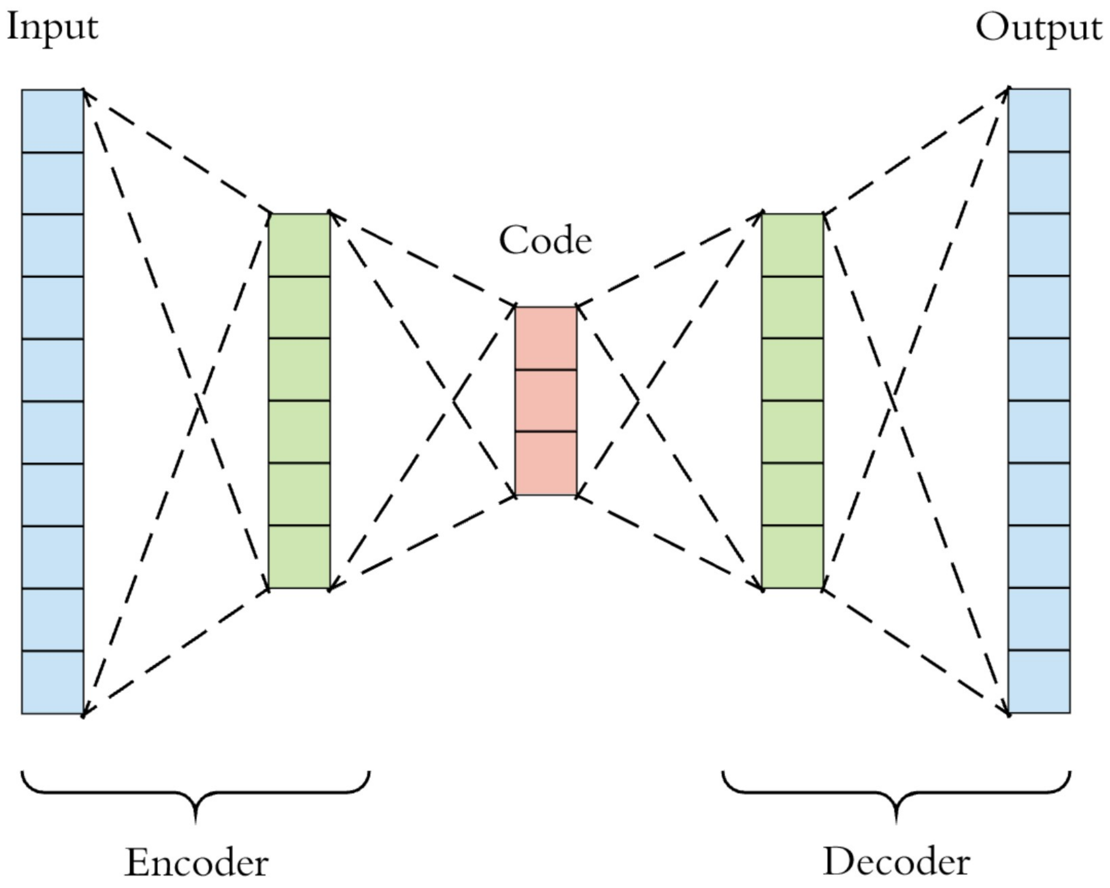
Methods for NLP robustness interpretation

- Are semantically similar sentences in the latent space nearby each other by Chebyshev distance?
- The perturbation radius ε is no longer interpretable
- Counter-examples cannot be directly converted to discrete input space (text)

Methods for NLP robustness interpretation

- Are semantically similar sentences in the latent space nearby each other by Chebyshev distance?
- The perturbation radius ε is no longer interpretable
- Counter-examples cannot be directly converted to discrete input space (text)
  - K-nearest neighbours
  - Autoencoder

Autoencoder



[4] M. Stewart 2019 “Comprehensive Introduction to Autoencoders”

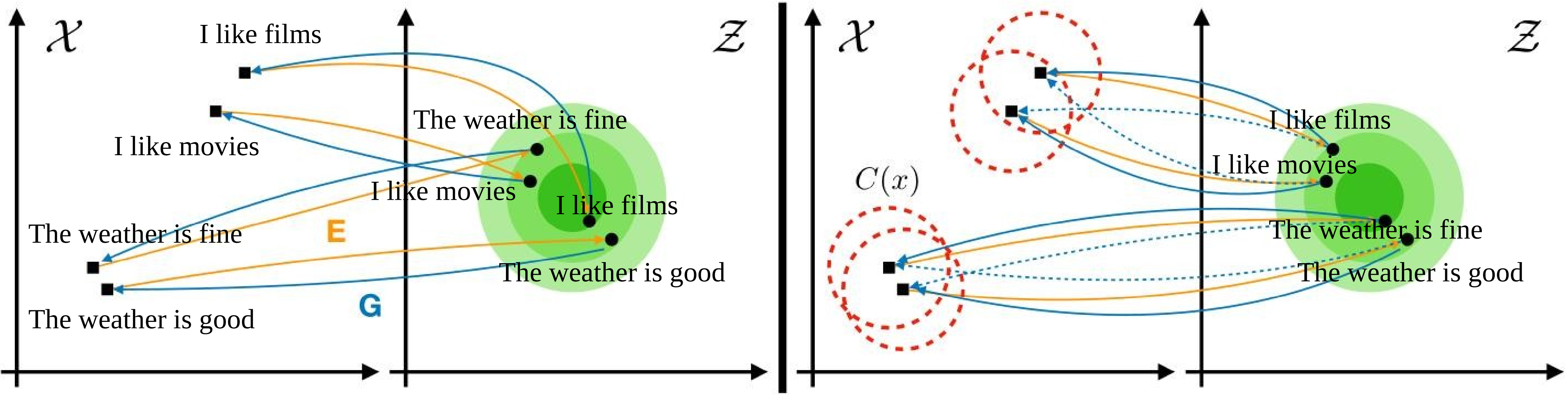
Denoising Adversarial Autoencoder (DAAE)

Educating Text Autoencoders: Latent Representation Guidance via Denoising

Tianxiao Shen [1] **Jonas Mueller** [2] **Regina Barzilay** [1] **Tommi Jaakkola** [1]

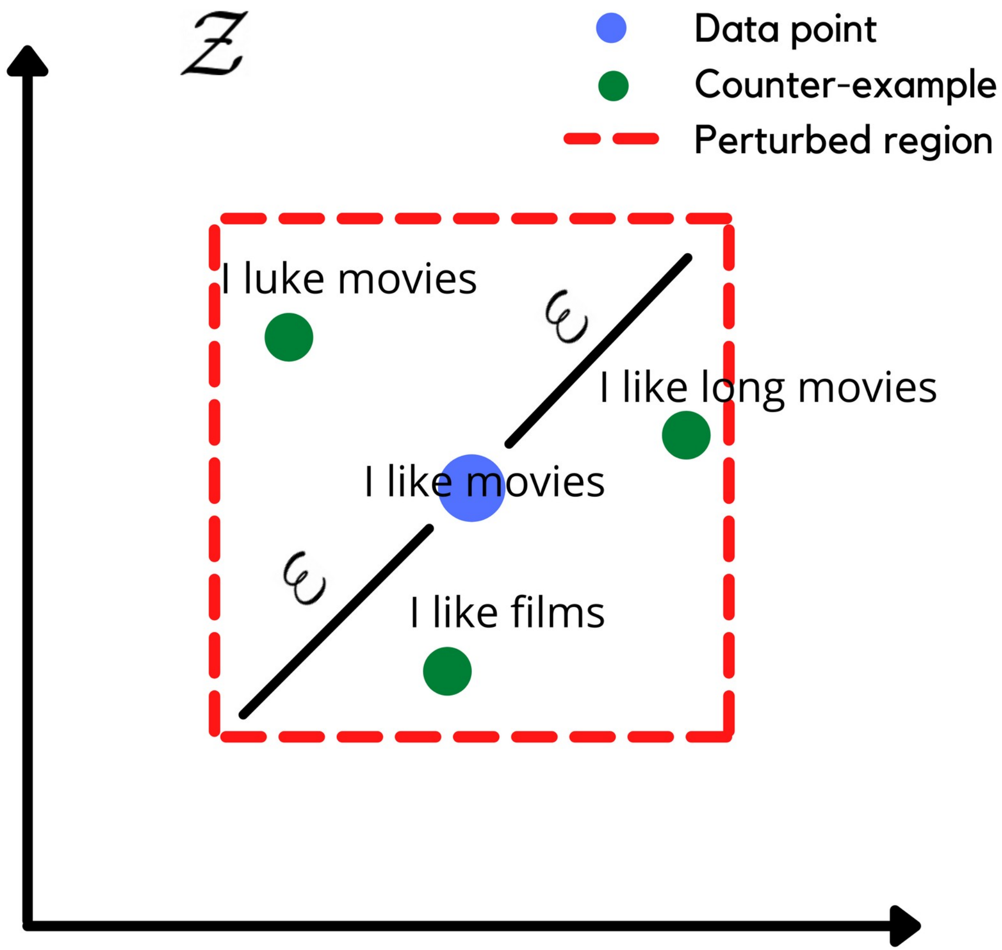
Denoising Adversarial Autoencoder (DAAE)

Educating Text Autoencoders: Latent Representation Guidance via Denoising



[5] Shen et al. 2020 “Educating Text Autoencoders: Latent Representation Guidance via Denoising”

Hypothesis for an ideal latent space



Results

Analysis of latent spaces

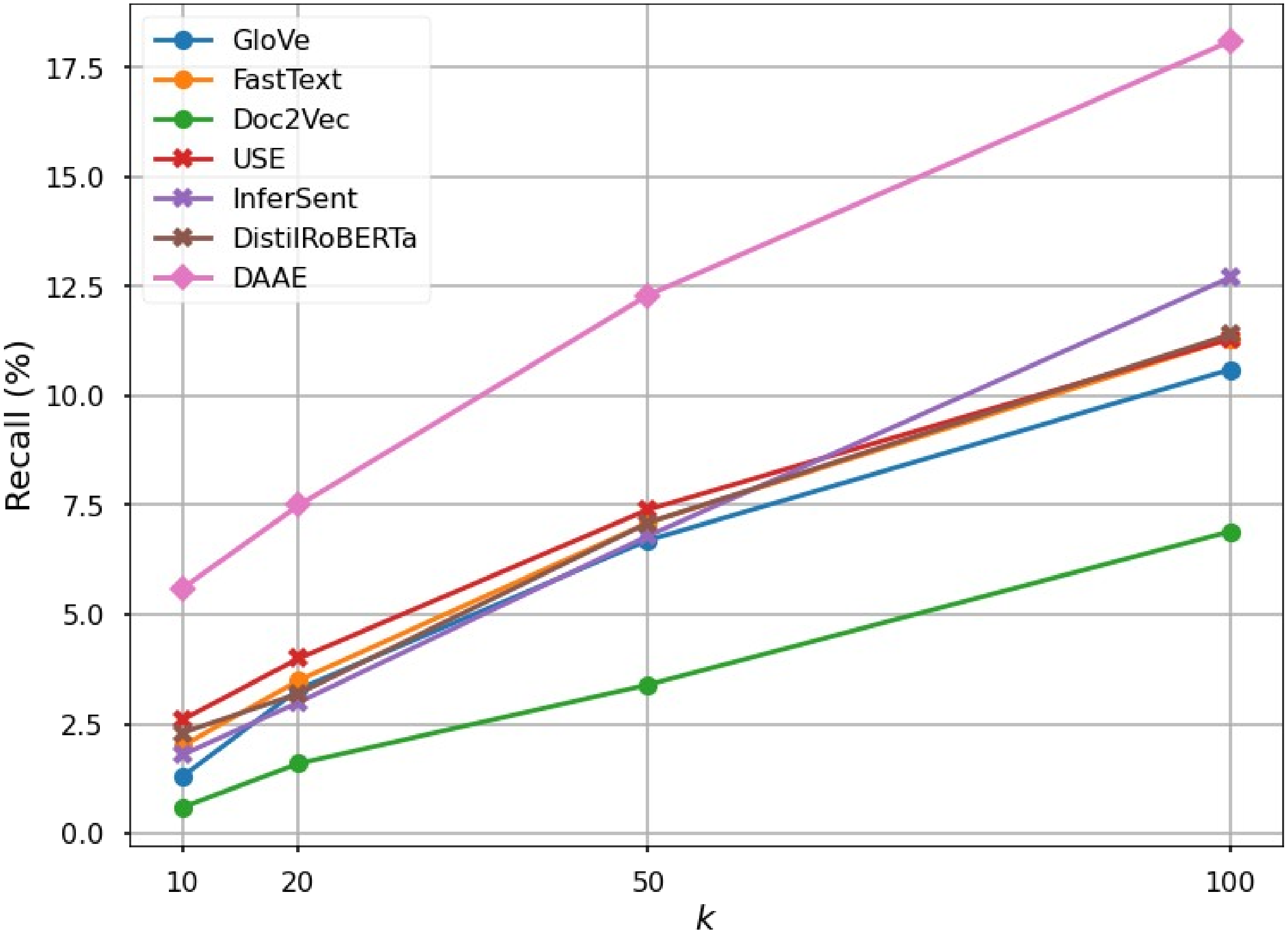
"How well is the neighborhood preserved?"

$$\frac{|\mathrm{NN}_x \cap \mathrm{NN}_z|}{|\mathrm{NN}_x|}$$

Normalized Levenshtein distance

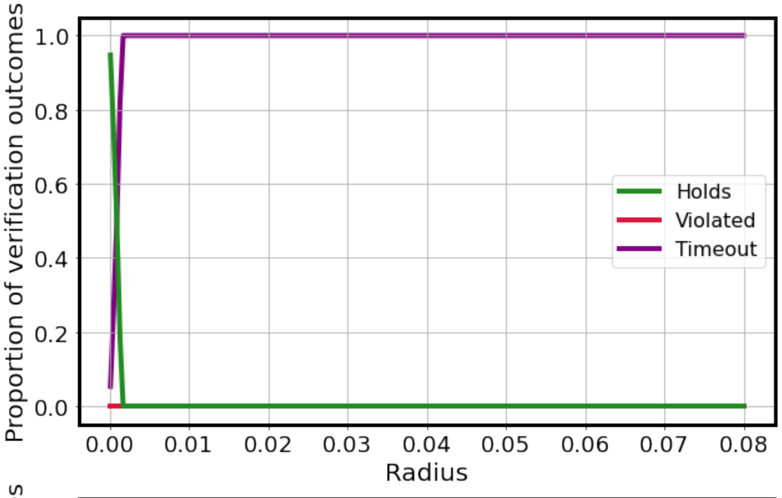
Chebyshev distance

IMDB dataset (split into sentences per review, rather than whole review)

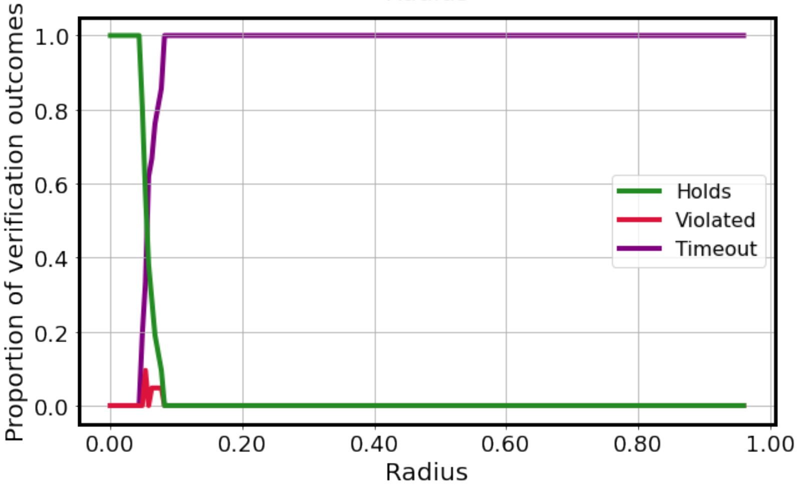


Robustness verification for NLP DNNs

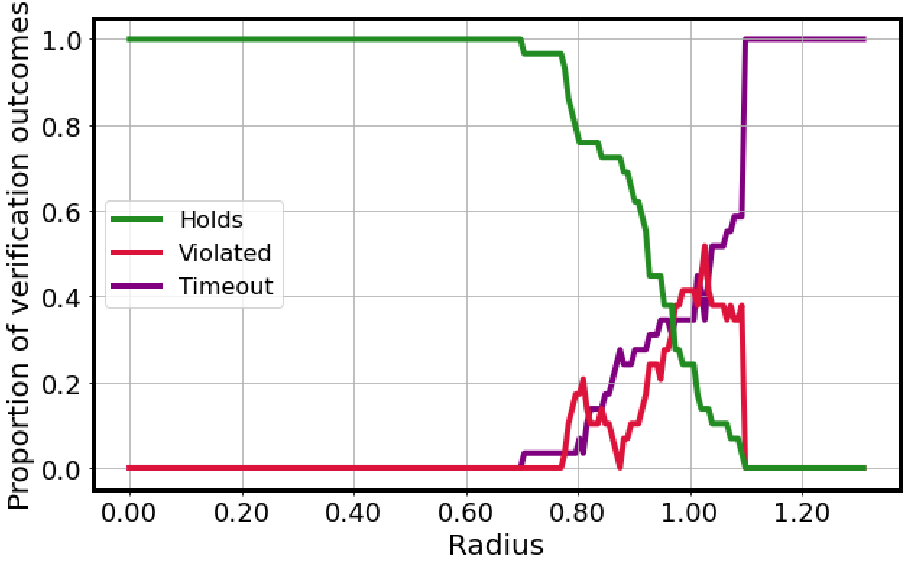
FastText



DistilRoBERTa



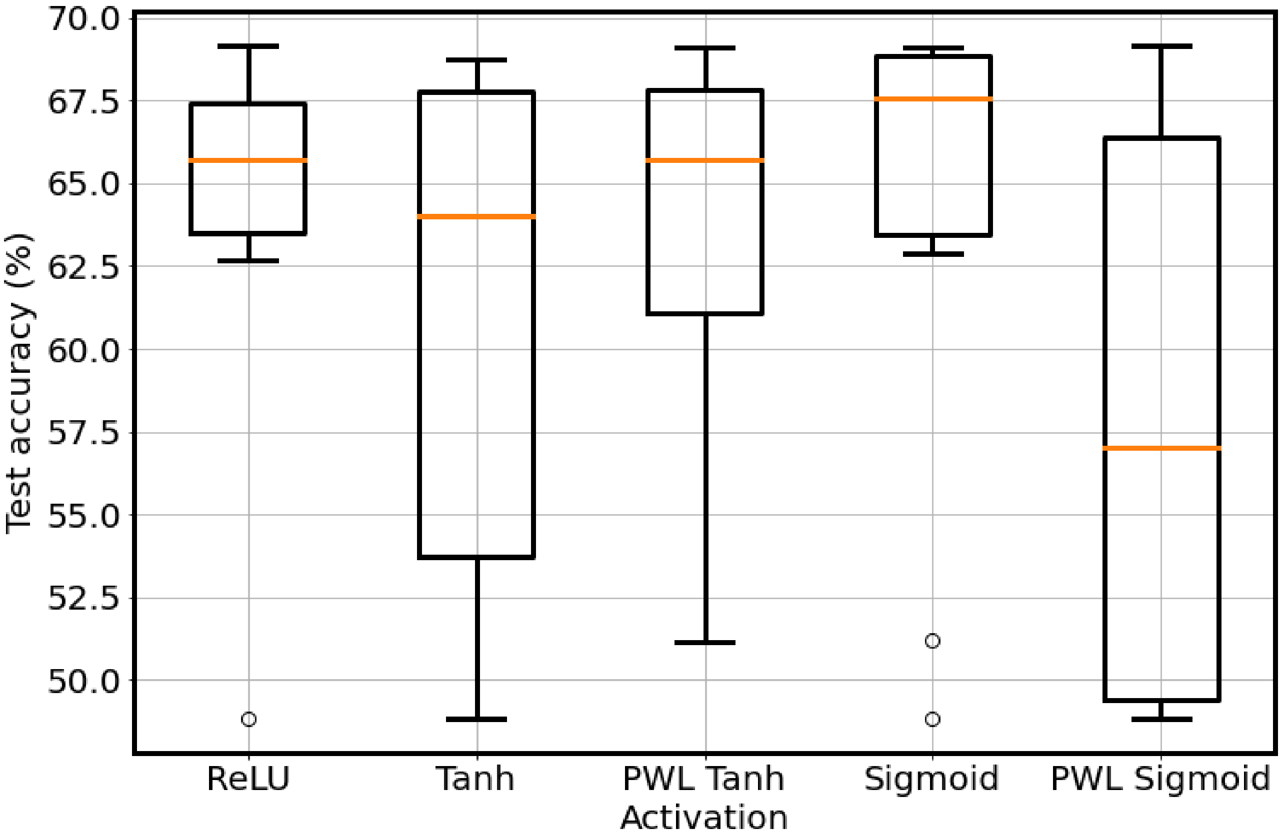
DAAE



Timeout: 10s / property
Size: 5 layers, 25 nodes / layer

PWL activation trade-offs

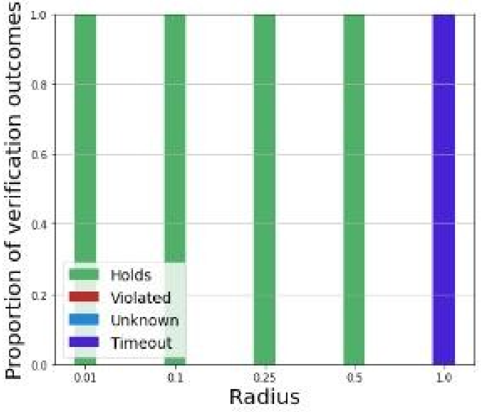
Only networks that performed well considered (smaller size, no InferSent or DAAE encodings)



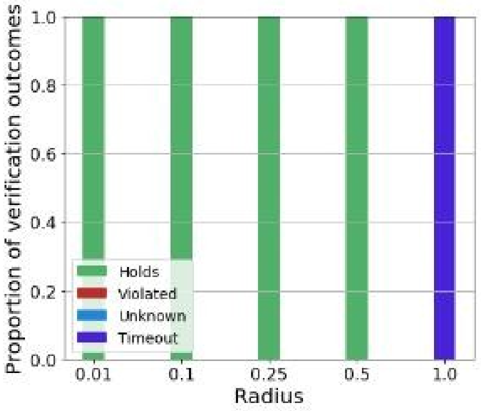
Timeout: 200s

PWL activation trade-offs

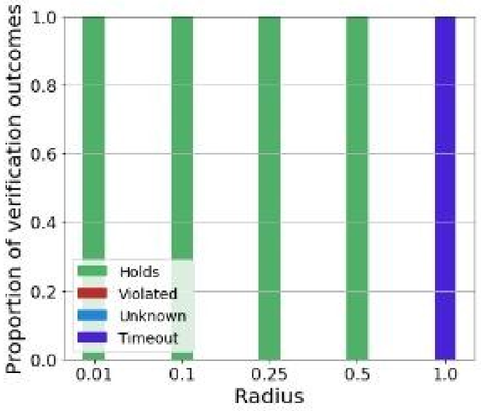
DAAE



(a) ReLU

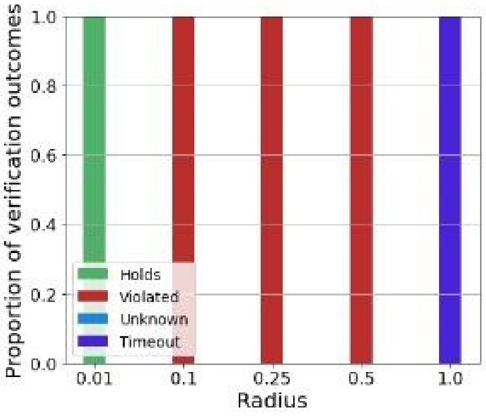


(b) Sigmoid

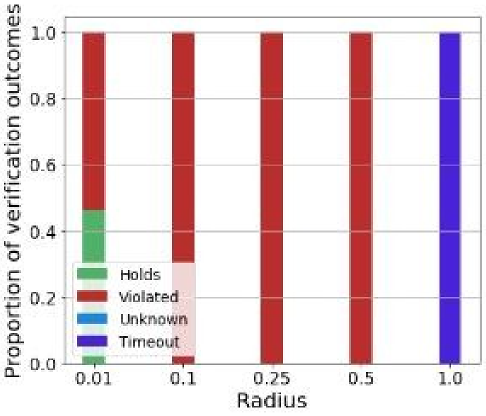


(c) PWL Sigmoid

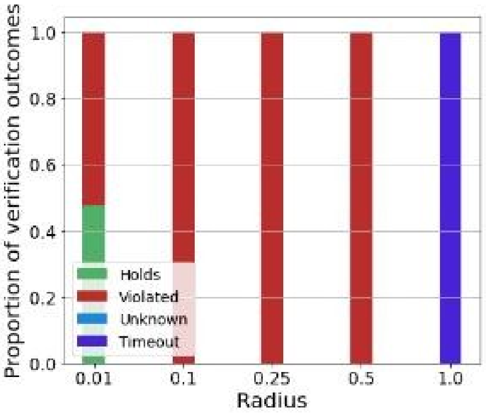
Distil RoBERTa



(a) ReLU



(b) Sigmoid



(c) PWL Sigmoid

Conclusion

- DAAE induced latent space is useful for robustness
  - Performs better than average word embeddings, sentence encoders, even BERT-based models
- KNN is a good way to interpret the perturbation radius, but autoencoders are more fine-grained and generate new samples

Conclusion

- DAAE induced latent space is useful for robustness
  - Performs better than average word embeddings, sentence encoders, even BERT-based models
- KNN is a good way to interpret the perturbation radius, but autoencoders are more fine-grained and generate new samples
- Network sizes and perturbation radius affects the efficiency of the solvers

Conclusion

- DAAE induced latent space is useful for robustness
  - Performs better than average word embeddings, sentence encoders, even BERT-based models
- KNN is a good way to interpret the perturbation radius, but autoencoders are more fine-grained and generate new samples
- Network sizes and perturbation radius affects the efficiency of the solvers
- PWL activation functions can be trained rather efficiently and produce comparable results to smooth versions
- Activation functions affect the verification results

Conclusion

- DAAE induced latent space is useful for robustness
  - Performs better than average word embeddings, sentence encoders, even BERT-based models
- KNN is a good way to interpret the perturbation radius, but autoencoders are more fine-grained to generate new samples
- Network sizes and perturbation radius affects the efficiency of the solvers
- PWL activation functions can be trained rather efficiently and produce comparable results to smooth versions
- Activation functions affect the verification results

Thank you for your attention.
Questions?

References

[1] Szegedy et al. 2014 “Intriguing properties of Neural Networks”
[2] Huang et al. 2019 “Achieving verified robustness to symbol substitution”
[3] Liu et al. 2019 CARS Workshop on NeuralVerification.jl. Accessed at: https://github.com/sisl/NeuralVerification-CARS-Workshop
[4] M. Stewart, 2019 “Comprehensive Introduction to Autoencoders”, Accessed at: https://towardsdatascience.com/generating-images-with-autoencoders-77fd3a8dd368
[5] Shen et al. 2020 “Educating Text Autoencoders: Latent Representation Guidance via Denoising”