Rules of the game

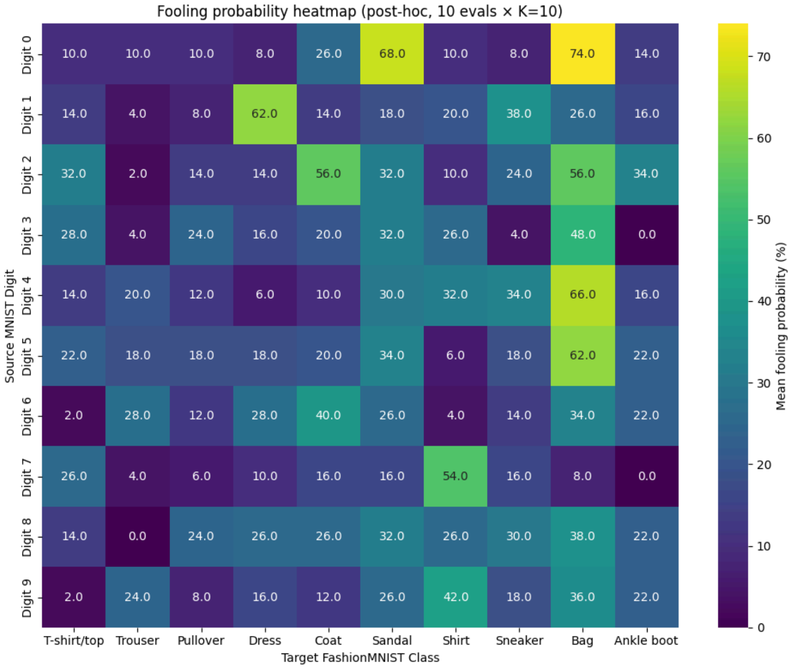
Explicación de lo que he hecho (Ale)

El codigo y los notebooks estan aqui
https://github.com/alejandro-arguelles/interclass-adversarial-attack.git
importante poner en la primera celda de general attack la ubicación de los modelos para que no tengais que entrenarlos.

La idea es demostrar que incluso los clasificadores generativos no son los suficientemente robustos a ataques adversarios y que los detectores marginales que se pueden obtener de estos tampoco lo son. Para ello hemos creado un ataque que altera muestras de un dataset (MNIST) para confundir a una victima entrenada con otro dataset (FMINST) y que ademas no sea detectado por el marginal detector. El fundamento de este ataque reposa sobre la likelihood paradox. Las dos ideas principales de este atque son las siguientes. 1: Las perturbación de la imagen no se hacen directamente sobre el espacio de las imagenes sino sobre el espacio latente de estas creado por el VAE. 2: Durante el entrenamiento se le fuerza al VAE ateacante a que no solo produzca imagenes parecidas a las de la entrada sino que ademas estas imagenes tengan una baja NLL según la victima (Negative Log Likelihood). Así al perturbar en el espacio latente nos aseguramos de que la imagen perturbada se parece a la original y ademas el marginal detector de la victima no la detecta.

Algunas observaciones: parece que al entrenarse con todas las clases le cuesta bajar a la NLL siendo casi constante desde el primera epoch oscilando lentamente. con k sampling para calcular la NLL observamos K = 5 una NLL de 221, con k = 100 una de 217 y con k = 1000 una de 216. Para observar este fenomeno mejor hacemos un “ataque” de cifra 1 a camiseta y entrenamos el modelo para ver mejor como varian la NLL y la elbo. En el notebook one to shirt si se aprecia muy bien como se mejora enc da epoch la NLL y al final si vemos que se le empieza a adaptar mas en forma que en textura las reconstrucciones para que se parezca un peliz ams a una camiseta y engañe al detector. Ademas en este notebook hay un plot de como la distribucion esta shifteada al acabar el entrenamiento.

En el notebook ataque general se implementa este ataque y se exponen estos resultados. El ataque se hace intentando maximizar la probabilidad de la clase a atacaralterando el vector latente generado por el VAE del atacante. Por cierto todo esto sigue un esquema GBY que era el mas sencilloq ue habia en el paper para experimentar. La probabilidad



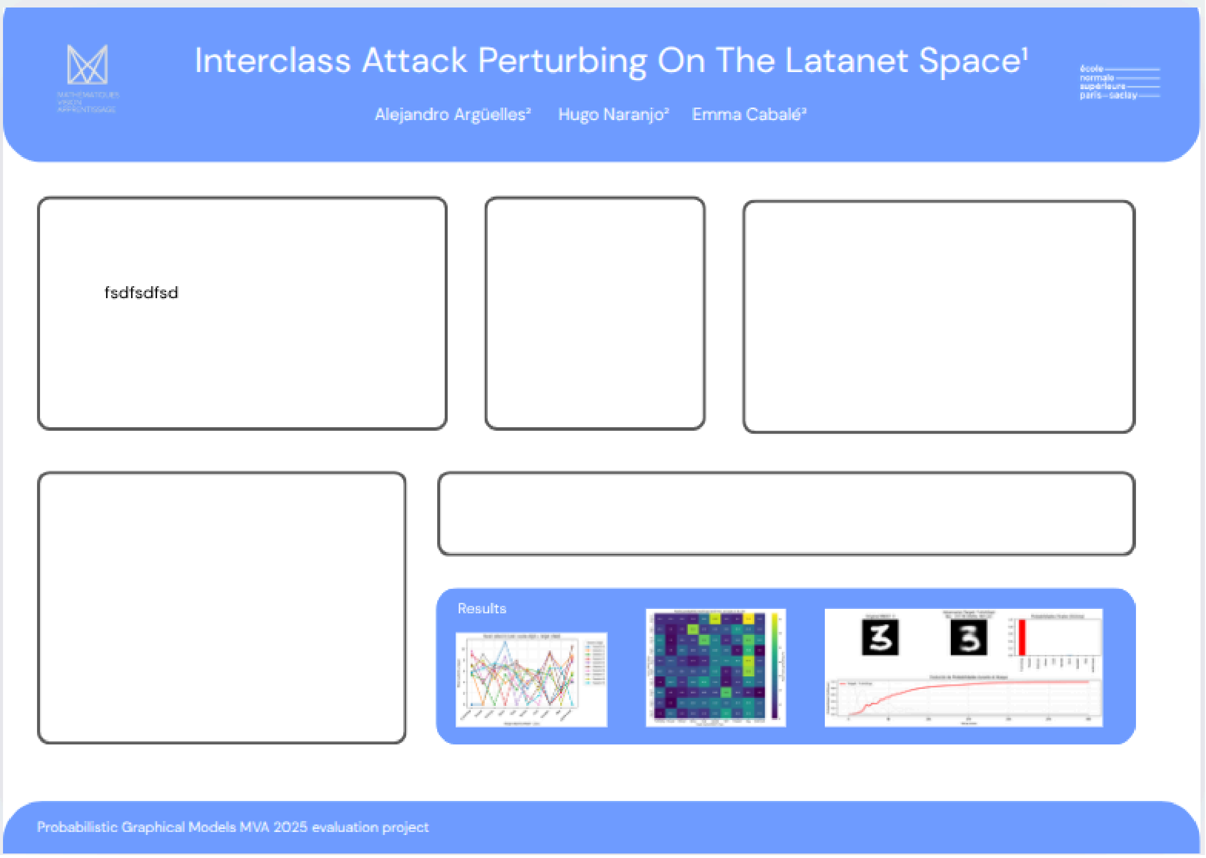
	T-shirt/top	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot
Digit 0	10.0	10.0	10.0	8.0	26.0	68.0	10.0	8.0	74.0	14.0
Digit 1	14.0	4.0	8.0	62.0	14.0	18.0	20.0	38.0	26.0	16.0
Digit 2	32.0	2.0	14.0	14.0	56.0	32.0	10.0	24.0	56.0	34.0
Digit 3	28.0	4.0	24.0	16.0	20.0	32.0	26.0	4.0	48.0	0.0
Digit 4	14.0	20.0	12.0	6.0	10.0	30.0	32.0	34.0	66.0	16.0
Digit 5	22.0	18.0	18.0	18.0	20.0	34.0	6.0	18.0	62.0	22.0
Digit 6	2.0	28.0	12.0	28.0	40.0	26.0	4.0	14.0	34.0	22.0
Digit 7	26.0	4.0	6.0	10.0	16.0	16.0	54.0	16.0	8.0	0.0
Digit 8	14.0	0.0	24.0	26.0	26.0	32.0	26.0	30.0	38.0	22.0
Digit 9	2.0	24.0	8.0	16.0	12.0	26.0	42.0	18.0	36.0	22.0

de engañar al clasificador y de pasar el detector esta ahi marcada. Todos pasan el detector y luego hay porcentajes notables de engaño como el de digit o a bag con un 74%. Recomiendo que mireis el codigo y que se lo paseis a chatgpt para que explique las partes y lo resuma. Una peculiaridad es que durante el ataque se fija el lote en el que se calculan las probabilidades de cada clase a k= 1000 ejemplos por clase para calcular las probabilidades de cada clase, esto de sfija para quitar ruido ya que cuando no se fija el gradiente es bastante aleatorio y le cuesta mas llegar, igualmente luego suele engañar aunque la clasificacion final se haga con k = 10 y otros ejemplos. EN eel experimento anterior una vez creado el ataque se hacen 10 clasificaciones con k = 10 sampling en cada una y se calcula el porcentaje de acierto.

En la carpeta Results estan los resultados en texto plano, hacer plots bonitos.

En plots hay plots ya, importantes son: distrib shift de como se shiftea la distribucion de los unos para que tengan mas Negative loglikelihood incluso aunque ya de por si los naturales no pasaban el threshold. la imagen one to shirt es interesante tambien,y exito prob.

He hecho el diseño de base del poster porque queria probar algunas ideas, es este



Interclass Attack Perturbing On The Latanet Space[1]

Alejandro Argüelles[2] Hugo Naranjo[2] Emma Cabalé[2]

fsdfsdfsd

Results

Probabilistic Graphical Models MVA 2025 evaluation project

se puede acceder y editar con el siguiente enlace https://www.canva.com/design/DAG67oM0jCc/WF9Fx8KCZf3JnnMzoQz5rQ/edit?utm_content=DAG67oM0jCc&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton para que lo acabeis. Puede estar bien comparar los resultados del GBY con los del paper.

Outline

1. Deep Bayes Classifier (DeepBC)
   1.1. Motivation
      1.1.1. Less work has investigated the robustness of generative classifiers against AA
         1.1.1.1. Generative classifiers should be robust to many recently proposed adversarial attacks if the "off-manifold" conjecture holds true
      1.1.2. Classical discriminative classifiers only model the conditional distribution of the label given the input
      1.1.3. Do not understand the underlying structure of the data, not being able to discern "natural" data from altered one.

1.1.4. Vulnerability when faced with attack techniques that fool a deep net with impoerceptible perturbations.

1.1.5. popular naive bayes and linear discriminant analysis perform poorly on image classification. (pixels are not independent on images!)

1.1.5.1. Serious security threats to ML systems

1.1.5.2. 2019 - Defense Examples:

1.1.5.2.1. Adversarial Training

1.1.5.2.1.1. Augments the training data with adversarially perturbation inputs.

1.1.5.2.1.2. moderate results

1.1.5.2.2. Bayes Neural Networks

1.1.5.2.2.1. Uncertainty estimates can be used to detect adversarial attacks

1.1.5.2.2.2. Denoisers

1.1.5.2.2.2.1. Rely on “off-maifold” conjecture

1.1.5.2.2.2.2. GANs, Auto encoders

1.1.5.2.2.2.3. Challenged by “sphere class” example.

1.2. How does it work?

1.2.1. Improvment of classical naive Bayes cassifier by applying conditional deep genrative models

1.2.2. Models the conditional distribution of an input by a deep latent variable model (LVM)

1.2.2.1. LVM learned with the variational auto-encoder algorithm

1.2.2.2. For Classification, Bayes Rules is approximated via Importance Sampling

1.2.3. joint probability Pd(x,y)

1.2.3.1. Pd(x|y_c) follows de manifold assumption

1.2.3.2. Pd(y|x) computed using bayes rule once you have a generative model p(x|y)

1.2.4. Deep Latent variable model is the key here!

1.2.4.1. p(x,y,**z**)

1.2.4.2. the way of defining this p(x,y,**z**) leads to different classifiers

1.2.4.3. LVM leads to a conditional distribution that is not factorised, which is much more powerful than naive Bayes

1.2.4.3.1. Discriminative

1.2.4.3.2. Generative

1.2.4.3.3. the impose different assumptions on the data generation process

1.2.4.4. Training method

1.2.4.4.1. amortized posterior q approximator

1.2.4.4.2. maximize ELBO

1.2.4.5.

1.3. Previous Work

1.3.1. Gu & Rigazio 2014 - Usage of generative models as defence technique

1.3.1.1. Auto-encoders to denoise the inputs before feeding them to the classifier

2. Detection methods for adversarial attackers

2.1. Critital Propoerties

2.1.1. Representations of the data manifold are the same for both the classifier and the detector

2.1.1.1. Share same generative model

2.1.2. logit values in generative classifiers have clear semantic meaning

2.1.3. marginal/logit detection aims to reject x, such that it has low log(p(x)) and or low log(p(x,y))

2.2. Three methods

2.2.1. First 2 use learned generative models as a proxy of the data manifold and reject inputs that are far away from it

2.2.2. Last model computes statistics for the classifiers output probability vector and rejects input that leads to under-confident predictions

2.2.3. METHOD 1: Marginal detection

2.2.3.1. rejects inputs that are far away from manifold based on a threshold

2.2.3.1.1. -log(p(x)) > delta

2.2.3.1.2. Threshold is based on E(-log(p(x))) + Var(-log(p(x)))

2.2.4. METHOD 2: Logit Detection

2.2.4.1. Logit in generateive classifiers has a well-defined meaning and can be used to detect attacks, even when the classifier is fooled

2.2.4.2. Rejection using joint density

2.2.4.3. y = F(x) a victim, one can reject x if -log(p(x, F(x)) > delta

2.2.4.4. Threshold chosen in the same way as Marginal detection

2.2.5. METHOD 3: Divergence detection

2.2.5.1. rejection of inputs with over- and under-confidence predictions

2.2.5.2. p(x) probability vector, collect the mean classification probability vector P(x) = E[p(x)], then compute E(-log(p(x))) + Var(-log(p(x)))

2.2.5.3. Select divergence D[p(x)][P(x)], and thus we reject the x such that D[p(x)][P(x)] > E(-log(p(x))) + Var(-log(p(x)))

2.2.5.3.1. Different divergences / distance measures can be applied such as the total variation TV

3. Types of attacks

3.1. White/black box

3.1.1. white box

3.1.1.1. you have acces to the network weights

3.1.2. black box

3.1.2.1. you dont have access to the network weights

3.2. Depending on the norm

3.2.1. l_inf attacks

3.2.1.1. Alteration is few but distributed pixel to pixel

3.2.1.2. “every pixel can be altered up to a certain (little) level

3.2.1.3. method 1: Fast gradient sign method

3.2.1.4. method 2: projected gradient descents

3.2.1.5. method 3: momentum iterative methods

3.2.2. l_2 attacks

3.2.2.1. Noise is limited by l2 norm

3.2.2.2. energy restricted

3.2.2.3. some pixels can have big changes as long as this is compensated with other pixels staying almost unaltered…

3.2.2.4. Clarini and Wagner CW L2 attack

3.2.3. Distillation Attacks!

3.2.3.1. Distills the victim classifier with a “student” CNN by forcing it to reproduce the results of the victim CNN and then apply white-box attacks.

3.2.3.1.1. grey-box

3.2.3.1.1.1. attacker has acces to both the training data and the output probability vectors of the classifiers on the training set

3.2.3.1.2. black-box

3.2.3.1.2.1. attacker only has access to queried labels on agiven input (una mierda vamos, solo ve la etiqueta no las probabilidades)

3.2.3.1.2.2. they use papernot to train a substitute CNN using Jacobian-Based dataset augmentation

3.2.4. SPSA (Evolutionary Strategies)

3.2.4.1. is a black box setting where you only have axes to the logit values of the prediction given an input.

3.2.4.2. they use SPSA l_inf attacks (Uesato et al)

3.2.4.2.1. basically similar to white box but gradients are computed numerically using the logit values form the victim classifier.

4. Experimental Results

4.1. The generative DeepBC is more robust than deep discriminative classifiers

4.1.1. Robustness is evaluated on MNIST and binary classification from CIFAR-10

4.1.1.1. For Deep NN we improve its robustness with CIFAR-10 multiclass by fusing discriminatively learned visual features witht he proposed generative classifiers

4.1.1.2. detection rates do not growth monotonically as c increases.

4.1.1.3. c = 10 is the sweet spot that achieves best success rates

4.2. Proposed detection methods are effective

4.3. Deep LVM-based generative classifiers generally outperform the randomised discriminative ones and the bottleneck is useful for defending agains l_inf attacks

4.4. Modelling unobserved variables are effective for defending afainst adversarial attacks

5. Overall Conclusions and Contributions

5.1. Their conclusions are nuanced: generative models are not a panacea; there are both theoretical and empirical limits to their robustness

6. Current state of Paper’s Proposal

NARRATIVE PROPOSAL:

1. Likelihood Paradox

One of the major assumptions in this paper relies on the “off-manifold” conjecture, which states that generative classifiers accurately capture the real geometry of the data. If true, the detector would see as outliers adversarial attacks, noise, and data form a different nature/dataset. The paper presents three detection methods, two of which are based on learning the generative model as a proxy of the data manifold, and reject inputs that are far away from it. As such, we want to challenge the validity of this assumption, and thus showcase the weaknesses of these detection methods. Upon literary review on the topic, we came across the notion of the likelihood paradox as well of the “inspiration”-parer: “A Geometric Explanation of the Likelihood OOD Detection Paradox (11 Jun 2024)”.

We want to showcase the weakness in the “off-manifold” assumption by training the proposed generative models on FMNIST, with the marginal detector and then run MNIST as a test set and observe whether the model firstly does not detect that it is an “off-manifold” data, and furthermore, test the confidence with which it provides us this classification.

Lastly, and if we have time, we would like to developpe an kind of “inter-dataset” attack, tailored to this problem, that takes a MNIST VAE and searches in the latent space the best adversarial exemple of a digit (e.g a certain image of a 5) that is viewed by the FMINST generative classifier as a normal data in FMinst (e.g a normal shirt). This kind of “attack” may appear bizarre or usless but the examples (if found and abundant) may conform very clear counterexamples to the off manifold paradox.

Email para Latouche:

We hope you are doing well. As we begin outlining our report for the project, we wanted to ask for your feedback on the angle we are considering.

The paper we selected is **“Are Generative Classifiers More Robust Against Adversarial Attacks?”**; as noted in the list of articles we could pick from, it’s quite dense! One of its central assumptions is the *off-manifold conjecture*: the idea is that adversarial examples, noise, or samples from a different dataset are far away from the data manifold, and thus should be detected as outliers. Two of the three detection methods in the paper explicitly rely on this premise.

Our goal would be to challenge the validity of this assumption and highlight potential weaknesses in these detection methods. In our literature review, we came across the *likelihood paradox* as well as the recent paper *"A Geometric Explanation of the Likelihood OOD Detection Paradox" (June 2024)*, which motivated us to explore this direction further.

Concretely, we would like to:

- Train the proposed generative models on FMNIST, use the marginal detector as described in the paper, and

- Evaluate MNIST as a test set to see whether the classifier (i) fails to detect MNIST as off-manifold and (ii) still provides high-confidence classifications.

If time allows, we would also like to explore a small extension: designing a simple "inter-dataset" attack. The idea would be to take a MNIST-trained VAE and search in its latent space for adversarial MNIST examples that the FMNIST generative classifier interprets as *normal* FMNIST samples (e.g., a digit '5' mapped to something classified as a shirt). While a bit unconventional, such examples, if found, could offer clear and extreme counter-examples to the off-manifold assumption and to the "robustness" of generative classifiers.

Would this focus be suitable for the report? We would be very happy to adjust or refine the direction based on your guidance.

Best regards,

Hugo, Alex and Emma.

DUDAS: el error de calculo de p(x) viene de que elbo es una lower bound y de que KL(p(z|x)||q(z|x)) puede ser muy grande???
Podria estar bien investigar priors mas flexibles para que genere y aproxime p(x) mejor? pero a lo mejor no quieres eso, no se puede aproximar KL2??? para hacer los detectores mas robustos??

Descripción de los experimentos e ideas exploradas (en español y con mis palabras):

La idea es demostrar que incluso los clasificadores generativos no son los suficientemente robustos a ataques adversarios y que los detectores marginales que se pueden obtener de estos tampoco lo son. Para ello hemos creado un ataque que altera muestras de un dataset (MNIST) para confundir a una victima entrenada con otro dataset (FMINST) y que ademas no sea detectado por el marginal detector. El fundamento de este ataque reposa sobre la likelihood paradox. Las dos ideas principales de este atque son las siguientes. 1: Las perturbación de la imagen no se hacen directamente sobre el espacio de las imagenes sino sobre el espacio latente de estas creado por el VAE. 2: Durante el entrenamiento se le fuerza al VAE ateacante a que no solo produzca imagenes parecidas a las de la entrada sino que ademas estas imagenes tengan una baja NLL según la victima (Negative Log Likelihood). Así al perturbar en el espacio latente nos aseguramos de que la imagen perturbada se parece a la original y ademas el marginal detector de la victima no la detecta.

Algunas observaciones: parece que al entrenarse con todas las clases le cuesta bajar a la NLL siendo casi constante desde el primera epoch oscilando lentamente. con k sampling para calcular la NLL observamos K = 5 una NLL de 221, con k = 100 una de 217 y con k = 1000 una de 216. Para observar este fenomeno mejor hacemos un “ataque” de cifra 5 a camiseta y entrenamos el modelo para ver mejor como varian la NLL y la elbo. En el notebook onte to shirt si se aprecia muy bien como se mejora enc da epoch la NLL y al final si vemos que se le empieza a adaptar mas en forma que en textura las reconstrucciones para que se parezca un peliz ams a una camiseta y engañe al detector.