



UNIVERSITÀ
DEGLI STUDI
FIRENZE

Scuola di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea in Informatica

Tesi di Laurea

ATTACCHI VERSO SISTEMI DI
APPRENDIMENTO IN AMBITO
AUTONOMOUS DRIVING: STUDIO E
IMPLEMENTAZIONE IN AMBIENTI
SIMULATI

TITOLO INGLESE

NOME CANDIDATO

Relatore: *Andrea Ceccarelli*

Correlatore: *Correlatore*

Anno Accademico 2014-2015

Nome candidato: *Attacchi verso sistemi di apprendimento in ambito autonomous driving: studio e implementazione in ambienti simulati*, Corso di Laurea in Informatica, © Anno Accademico 2014-2015

INDICE

1	Fondamenti	11
1.1	Intelligenza Artificiale	11
2	L'Adversarial Robustness Toolbox	13
2.1	Trasformazioni Spaziali	14
2.2	Perturbazione dell'immagine	16

ELENCO DELLE FIGURE

Figura 1	self driving car di Google	9
Figura 2	self driving car di Tesla	9
Figura 3	Possibili definizioni di Intelligenza artificiale	11

ELENCO DELLE TABELLE

"Inserire citazione"
— *Inserire autore citazione*

INTRODUZIONE

Il mondo odierno è ormai pervaso dall'Intelligenza Artificiale. Uno dei settori di maggior interesse per la ricerca in questo ambito è indubbiamente quello delle cosiddette *Self Driving Car*, ovvero le macchine a guida autonoma. Grandi aziende quali *Google* e *Tesla* hanno già sviluppato dei propri modelli (Figura 1 e 2) e l'interesse per questo tipo di veicoli è sempre in maggior crescita. Per poter funzionare correttamente questi veicoli acquisiscono informazioni dall'ambiente circostante sotto forma di immagini. Queste immagini vengono classificate da un sistema interno che, in base alle informazioni ricevute, decide l'azione da compiere (sterzare, accelerare, frenare ecc.). Anni ed anni di sviluppo e ricerca hanno reso sempre più affidabili e sicuri questi sistemi ma restano comunque presenti delle vulnerabilità. Una vulnerabilità molto importante sono i cosiddetti *Adversarial attacks*. Il meccanismo di questi attacchi è molto semplice: alle immagini raccolte dal sistema viene applicata una modifica impercettibile a occhio umano ma in grado di causare un errore di classificazione che può portare, ad esempio, una macchina ad accelerare quando dovrebbe frenare. La ricerca su questi tipi di attacchi quindi è fondamentale per garantire l'affidabilità dei veicoli a guida autonoma. Lo scopo di questa tesi è l'implementazione di *Adversarial Attacks* contro modelli di guida autonoma in ambiente simulato. Il lavoro è così suddiviso:

- **Capitolo 1:** descrizione dei fondamenti teorici alla base dei sistemi di guida autonoma e più in generale dei sistemi intelligenti
- **Capitolo 2:** descrizione degli strumenti utilizzati: in particolare



Figura 1: self driving car di Google



Figura 2: self driving car di Tesla

vengono presentati *l'Adversarial Robustness Toolbox* e il simulatore *Carla*

- **Capitolo 3:** descrizione degli attacchi scelti per l'implementazione e motivazioni per le scelte effettuate
- **Capitolo 4:** Implementazione e risultati
- **Capitolo 5:** Conclusioni e possibili sviluppi futuri

FONDAMENTI

1.1 INTELLIGENZA ARTIFICIALE

Dare un'unica definizione di intelligenza artificiale risulta estremamente difficile a causa della vastità e dell'interdisciplinarietà dell'argomento. Soltanto nella figura 3 troviamo otto possibili definizioni, ciascuna valida, ma la descrizione migliore per i nostri scopi è quella data dal padre di questa disciplina John McCarthy: L'Intelligenza Artificiale è la scienza volta alla creazione di macchine *intelligenti*, più nello specifico di *programmi intelligenti*. [3]

<p>“The exciting new effort to make computers think ... <i>machines with minds</i>, in the full and literal sense" (Haugeland, 1985)</p> <p>“The automation of activities that we associate with human thinking, activities such as decision-making, problem solving, learning ..." (Bellman, 1978)</p>	<p>“The study of mental faculties through the use of computational models" (Charniak and McDermott, 1985)</p> <p>“The study of the computations that make it possible to perceive, reason, and act" (Winston, 1992)</p>
<p>“The art of creating machines that perform functions that require intelligence when performed by people" (Kurzweil, 1990)</p> <p>“The study of how to make computers do things at which, at the moment, people are better" (Rich and Knight, 1991)</p>	<p>“A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes" (Schalkoff, 1990)</p> <p>“The branch of computer science that is concerned with the automation of intelligent behavior" (Luger and Stubblefield, 1993)</p>

Figura 3: Possibili definizioni di Intelligenza artificiale

L'ADVERSARIAL ROBUSTNESS TOOLBOX

L'Adversarial Robustness ToolBox (ART) è una libreria python che permette lo sviluppo di difese per i modelli ad apprendimento automatico, rendendoli più sicuri ed affidabili. Questi modelli sono infatti vulnerabili ai cosiddetti "esempi antagonisti": dati in input (immagini, testo ecc.) creati specificatamente per produrre una determinata risposta dal modello. L'ART include questi attacchi e fornisce gli strumenti per sviluppare sistemi di difesa contro di essi. In questa sezione ci concentreremo sugli attacchi, descrivendone il funzionamento e una possibile implementazione su modelli ADAS realizzati all'interno del simulatore Carla.

Gli attacchi presenti nella libreria sono suddivisi nel seguente modo:

- Evasion Attacks, dove i dati in input vengono modificati fino ad avere un errore di classificazione
- Poisoning Attacks. In questo caso l'obiettivo è iniettare dati costruiti in modo specifico per compromettere la fase di apprendimento. Sono particolarmente efficaci nei casi in cui il riaddestramento è frequente.
- Extraction Attacks. Nei casi in cui il modello di apprendimento non sia direttamente accessibile, questi tipi di attacchi vengono sviluppati per addestrare un modello sostituto che sia funzionalmente equivalente al modello target.

Nel contesto della guida autonoma, ha senso concentrarsi sugli Evasion Attacks. Possiamo infatti assumere che il riaddestramento non avvenga con una tale frequenza da giustificare lo sviluppo intensivo di un attacco poisoning. Per quanto riguarda gli **Extraction Attacks**, muovendoci in un contesto open source e accademico dove i modelli sono liberamente accessibili, hanno un'utilità limitata per i nostri obiettivi. Passiamo ora a descrivere alcuni degli esempi presenti nel toolbox.

2.1 TRASFORMAZIONI SPAZIALI

I primi attacchi che consideriamo sono quelli che applicano semplici trasformazioni spaziali alle immagini senza modificare i pixel in modo diretto. Si tratta di attacchi molto interessanti in quanto facilmente implementabili. Si tratta infatti di ruotare, spostare o sovrapporre più immagini.

Lista di Attacchi			
Nome	Descrizione attacco	Applicabilità	Implementazione in Carla
Adversarial Patch [1]	L'attacco consiste nell'attaccare su un qualsiasi oggetto una specifica patch. Questa patch è creata in modo da causare errori di classificazione. Può prendere qualsiasi forma e viene addestrata su un'insieme di immagini, nelle quali verrà applicata con dimensioni e rotazione casuali	Un attaccante potrebbe semplicemente stampare la patch e attaccarla ad esempio su un cartello stradale, mandando in confusione i sistemi ADAS. Se si vuole utilizzare l'attacco su delle immagini si può semplicemente porre la patch direttamente sull'immagine.	È necessario modificare gli oggetti della simulazione (segnali, veicoli). Nello specifico, dobbiamo intervenire sul motore UE4, responsabile della costruzione di tali oggetti
Spatial Transformation Attack [2]	Le immagini che arrivano al classificatore vengono sottoposte a rotazione e traslazione fino a causare un errore di classificazione	L'attacco potrebbe essere iniettato a livello dei sensori di un sistema ADAS, in modo da modificare direttamente i dati in input causando problemi difficilmente rintracciabili. Questo richiede un tampering della telecamera, per imporre la trasformazione voluta alle immagini.	Si utilizzano le API fornite dal simulatore per acquisire le immagini create

2.2 PERTURBAZIONE DELL'IMMAGINE

In questo caso l'immagine in input viene sottoposta a una perturbazione, ovvero una modifica di un certo numero di pixel in modo da non essere visibile all'occhio umano ma in grado di causare misclassificazione. Questi attacchi risultano essere molto simili tra loro, ma spesso una piccola modifica può causare risultati estremamente diversi. Nello specifico l'adversarial sample \mathbf{x}' viene definito come:

$$\begin{aligned} \mathbf{x}' &= \mathbf{x} + \epsilon_{\mathbf{x}} \\ \{\mathbf{x}' \in \mathbb{R}^{m \times n \times 3} \mid \underset{j}{\operatorname{argmax}} f(\mathbf{x}') &\neq \underset{i}{\operatorname{argmax}} f(\mathbf{x})\} \end{aligned}$$

dove $\epsilon_{\mathbf{x}} \in \mathbb{R}^{m \times n \times 3}$ è la perturbazione aggiunta all'input [5].

Lista di Attacchi			
Nome	Descrizione attacco	Applicabilità	Implementazione in Carla
Threshold Attack [5]	Viene imposta una soglia massima th alla perturbazione. Nello specifico l'attacco ottimizza il vincolo $\ \epsilon_x\ _\infty \leq th$ dove th è uno dei seguenti valori $\{1, 3, 5, 10\}$	Dobbiamo poter perturbare le immagini e successivamente passarle al classificatore. Per questo l'attacco verrà implementato a livello dell'unità di elaborazione.	Si utilizzano le python API fornite dal simulatore per avere accesso alla struttura interna dei veicoli, dove risiede il classificatore
Low Pixel Attack [5]	È una variazione del threshold attack usando la norma o al posto della norma infinito.	Poichè l'attacco è concettualmente identico al threshold attack anch'esso verrà iniettato nell'unità di elaborazione	Si accede alla struttura interna dei veicoli con le python API
Decision Tree Attack [4]	Si sfrutta la struttura ad albero del classificatore, cercando un cammino dalla foglia originale ad una foglia che corrisponde a una classe diversa. Infine si applica la perturbazione all'esempio	L'attacco viene iniettato all'interno dell'unità di elaborazione del sistema di guida autonoma.	Si accede alla struttura interna dei veicoli con le python API

BIBLIOGRAFIA

- [1] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *CoRR*, abs/1712.09665, 2017. (Cited on page 15.)
- [2] Logan Engstrom, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. A rotation and a translation suffice: Fooling cnns with simple transformations. *CoRR*, abs/1712.02779, 2017. (Cited on page 15.)
- [3] John Mccarthy. What is artificial intelligence. 1998. (Cited on page 11.)
- [4] Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, abs/1605.07277, 2016. (Cited on page 17.)
- [5] Danilo Vasconcellos Vargas and Shashank Kotyan. Model agnostic dual quality assessment for adversarial machine learning and an analysis of current neural networks and defenses. *CoRR*, abs/1906.06026, 2019. (Cited on pages 16 and 17.)