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

NICCOLÓ PIAZZESI

Relatore: *Andrea Ceccarelli*
Correlatore: *Correlatore*

Anno Accademico 2019-2020

Niccoló Piazzesi: *Attacchi verso sistemi di apprendimento in ambito autonomous driving: studio e implementazione in ambienti simulati,* Corso di Laurea in Informatica, © Anno Accademico 2019-2020

INDICE

ELENCO DELLE FIGURE

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 cosidette *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 cosi suddiviso:

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

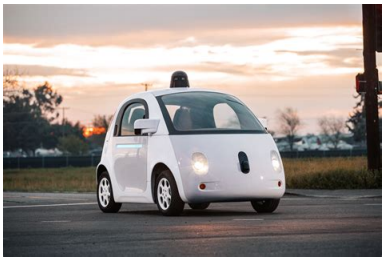

Figura 1: self driving car di Google

Figura 2: self driving car di Tesla

- **Capitolo 2:** descrizione degli strumenti utilizzati: in particolare 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

1

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 ne troviamo addirittura otto , tutte valide, ma forse 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].

1.1.1 *Agenti e ambienti*

Il concetto di macchina intelligente può essere ulteriormente astratto dall'idea di **agente computazionale intelligente**. Esaminiamo adesso questa definizone. Un **agente** è un'entità che compie azioni e percepisce informazioni da un ambiente.

Un'agente si comporta in modo **intelligente** quando:

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

Figura 3: Possibili definizioni di Intelligenza artificiale[7]

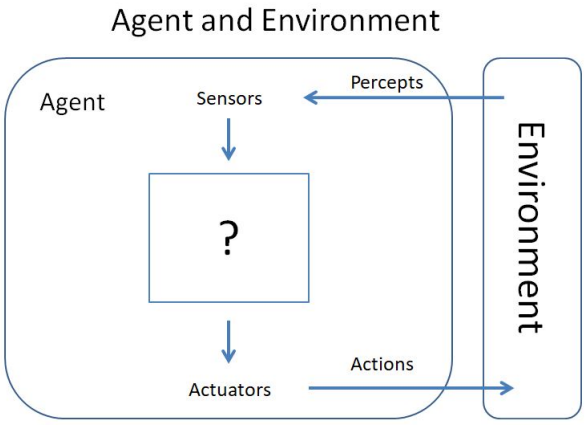

Figura 4: Agente computazionale

- le sue azioni sono appropriate alle circostanze e ai suoi scopi
- È flessibile per ambienti e scopi dinamici
- impara dall'esperienza
- fa le scelte **giuste** date le sue limitazioni percettive e computazionali. Un agente tipicamente non può osservare l'ambiente direttamente; ha una memoria e un tempo per agire molto limitati.

Un agente **computazionale** è un agente le cui decisioni possono essere descritte in termini di una computazione. Questo significa che, una decisione può essere scomposta in una serie di operazioni primitive implementabili in un dispositivo fisico.[6].

1.1.2 *Intelligenza e performance*

Abbiamo detto che un agente è intelligente quando compie le *scelte giuste*. Ma cosa significa fare la scelta giusta? Rispondiamo a questa domanda in un modo molto banale: considerando le *consequenze* del comportamento di un agente. Un agente si muove all'interno di un ambiente e compie una sequenza di azioni in base alle informazioni che riceve. Queste azioni causano variazioni allo stato dell'ambiente. Se la variazioni è desiderabile, l'agente ha fatto le scelte giuste. La nozione di desiderabilità è catturata da una **misura di prestazione(performance measure)** che valuta una qualunque sequenza di azioni. Ovviamente non esiste una performance measure globale; solitamente viene co-

PEAS description of the task environment for an automated taxi

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi Driver	safe, fast, legal, comfortable, maximum profit, getting to correct destination	roads, other traffic pedestrians customers	steering, accelarator, brake, signal, horn, display	cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard

Figura 5: descrizione PEAS dell'ambiente di lavoro di un taxi automatico

struita appositamente per uno specifico problema. Le specifiche dell'agente,dell'ambiente e della misura di prestazioni vengono raggruppate nell'**ambiente di lavoro(task environment)**, attraverso la descrizione PEAS(Performance,Environment,Actuators,Sensors).[7] La figura 5 riassume un esempio di descrizione PEAS dell'ambiente di lavoro di un taxi.

1.1.3 *Razionalità*

Ciò che abbiamo descritto fin'ora ci permette di verificare in modo quantitativo l'intelligenza di un agente. Espandiamo questa idea introducendo la nozione di **razionalità**. Cosa sia razionale in qualsiasi momento dipenda da quattro elementi:

- la misura di prestazioni
- la conoscenza a priori dell'agente
- le azioni che l'agente può compiere

- la sequenza di percezioni fino a quel momento

Questo ci porta alla definizione di **Agente Razionale**:

per ciascuna sequenza di percezioni, un agente razionale seleziona un'azione che massimizza il valore atteso della performance measure, data l'evidenza fornita dalla sequenza di percezioni e dalla conoscenza a priori dell'agente

La razionalità non significa **onniscienza**. Un agente deve essere in grado di prendere decisioni anche quando non ha a disposizione tutte le informazioni necessarie a compiere l'azione perfetta. Un agente razionale massimizza *il vallore atteso* della performance basandosi sulla sua conoscenza pregressa dell'ambiente, ma non sempre un ambiente è completamente osservabile. Per questo motivo sono fondamentali la fase di raccolta delle informazioni(**information gathering**) e di apprendimento(**learning**). Per poter migliorare le prestazioni, un agente deve essere in grado di raccogliere la massima quantità possibile di informazioni e di aggiungere tali informazioni alla propria base di conoscenza. Questo permette una scelta sempre migliore delle azioni da compiere.[7]

1.1.4 *Scopo dell'IA*

Lo scopo scientifico dell'IA è quello di studiare i principi e i meccanismi che rendono il comportamento intelligente possibile sia nei sistemi naturali che in quelli artificiali. Lo scopo ingegneristico di questa disciplina è la costruzione di dispositivi fisici in grado di comportarsi in modo intelligente. Da queste basi si sono sviluppati un innumerevole quantità di branche, tra cui le più importanti sono: sono:[3]

- **logical AI**
- **search**
- **knowledge and reasoning**
- **planning**
- **learning from experience**
- **genetic programming**

In figura 6 vengono riportate alcune delle applicazioni di maggior successo dei *Sistemi Intelligenti*. Il settore delle Self driving Car si basa in particolare su **Machine Learning** e **Computer Vision**

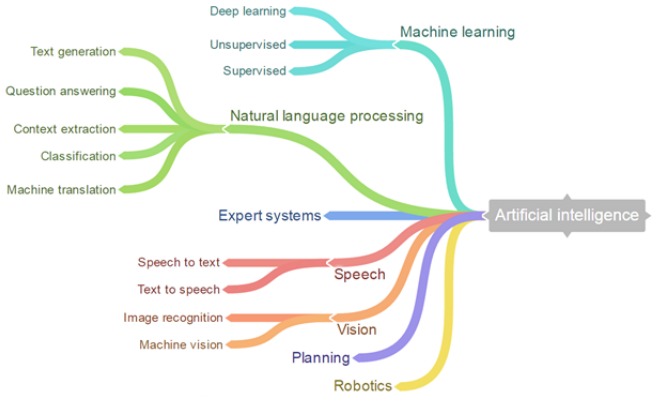

Figura 6: applicazioni dell'IA

1.2 MACHINE LEARNING

Il **Machine Learning(ML)** è una branca dell'intelligenza artificiale dedicata allo studio e allo sviluppo di algoritmi che migliorano le prestazioni attraverso l'esperienza[9]. L'abilità di un algoritmo di migliorare dall'esperienza automaticamente è fondamentale, perchè è impossibile scrivere programmi che sappiano a priori tutte le possibili situazioni in cui un agente potrebbe ritrovarsi. Inoltre l'ambiente in cui l'agente è inserito può mutare nel tempo[7]. Consideriamo il caso dei mercati finanziari. Un programma creato per prevedere i prezzi delle azioni di domani deve essere in grado di adattarsi alle variazioni improvvise causate ad esempio da una crisi globale.

1.2.1 *Problemi di apprendimento ben definiti*

Iniziamo ad approfondire i concetti di apprendimento automatico considerando alcuni task di apprendimento. Più precisamente:

Definizione 1.1. Si dice che un programma **impara** dall'esperienza E rispetto a una classe di tasks T e una misura di prestazioni P, se la sua prestazione nei tasks in T, misurata da P, migliora con l'esperienza E

Dalla definizione 1.1 possiamo specificare diversi problemi di apprendimento, ad esempio[4]:

- **Task T**: giocare a scacchi
- **Misura di prestazioni P**: percentuale di partite vinte contro gli avversari

- **Esperienza E**: giocare partite di prova contro se stesso

Ogni problema di learning consiste nel prendere la conoscenza a priori e i dati ricevuti(l'esperienza E) e traformarli in una rappresentazione interna utilizzata da un agente per prendere decisioni. La rappresentazione può corrispondere con i dati stessi ricevuti ma di solito è una sintesi compatta e significativa. In definitiva, per specificare una determinata tecnica di apprendimento è quindi necessario affrontare le seguenti questioni:

- **Task** Un task di apprendimento è una qualsiasi attività che può essere appresa da un agente
- **Feedback** Durante l'apprendimento a un agente viene fornito un riscontro in base alla correttezza delle azioni svolte. Il riscontro può essere un premio o una punizione. In base ad esso un agente modifica le proprie azioni migliorando coì la propria esecuzione su un determinato task.
- **Rappresentazione** Come detto in precedenza l'esperienza deve influenzare la rappresentazione interna di un agente. Gran parte del Machine Learning è focalizzato nel contesto di una specifica rappresentazione(es. reti neurali)
- **Online e Offline** Nel learning offline, tutti i training examples sono disponibili prima dell'azione di un agente. Nel learning online, gli esempi vengono ricevuti durante l'esecuzione.
- **Misura di Successo** Per sapere se un agente ha effettivamente imparato, è necessaria una misura di successo. La misura NON riguarda le prestazioni sui dati di addestramento, bensì le prestazioni su nuove esperienze.
- **Bias** Con il termine *bias* si intende la tendenza a preferire un'ipotesi rispetto a un'altra, concetto fondamentale nel processo di scelta
- **Noise**Una delle proprietà più importanti per un algoritmo di apprendimento è la capacità di gestione di dati condizionati. Infatti nella pratica i dati possono spesso risultare imperfetti o in alcuni casi incompleti .
- **Interpolazione e Estrapolazione** L'interpolazione riguarda le

2

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 cosidetti "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:

$$\mathbf{x'} = \mathbf{x} + \epsilon_x$$
$$\{\mathbf{x'} \in \mathbb{R}^{m \times n \times 3} | \operatorname*{argmax}_j f(x') \neq \operatorname*{argmax}_i f(x)\}$$

dove $\epsilon_x \in \mathbb{R}^{m \times n \times 3}$ è la perturbazione aggiunta all'input [8].

Lista di Attacchi			
Nome	Descrizione attacco	Applicabilità	Implementazione in Carla
Threshold Attack [8]	Viene imposta una soglia massima th alla perturbazione. Nello specifico l'attacco ottimizza il vincolo $\|\epsilon_x\|_\infty \leqslant th$ 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 [8]	È una variazione del threshold attack usando la norma 0 al posto della norma infinito.	Poichè l'attacco è concettualmente identico al threshold attack anch'esso verra iniettato nell'unità di elaborazione	Si accede alla struttura interna dei veicoli con le python API
Decision Tree Attack [5]	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 19.)

[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 19.)

[3] John Mccarthy. What is artificial intelligence. 1998. (Cited on pages 11 and 14.)

[4] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997. (Cited on page 15.)

[5] 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 21.)

[6] David Poole and Alan Mackworth. *Artificial Intelligence: Foundations of Computational Agents*. Cambridge University Press, Cambridge, UK, 2017. (Cited on page 12.)

[7] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (3rd Edition)*. Prentice Hall, 2009. (Cited on pages 3, 11, 13, 14, and 15.)

[8] 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 20 and 21.)

[9] Wikipedia. Machine learning, 2020. (Cited on page 15.)