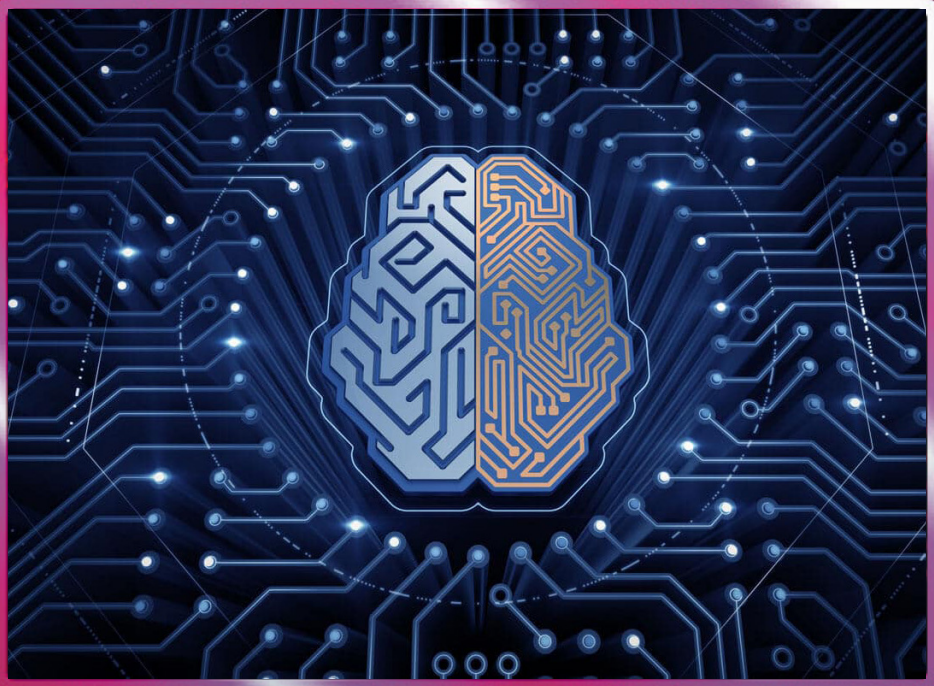


INTELLIGENZA ARTIFICIALE
PROSPETTIVE PER LA SICUREZZA E LA DIGITAL FORENSICS

Si sente molto parlare di intelligenza artificiale (AI o IA), machine learning e deep learning negli ultimi tempi, in realtà è un argomento che affonda le sue radici a parecchi decenni fa, infatti l'IA fu teorizzata e poi sviluppata con modelli matematici sin dal 1956 ad opera di nomi eccelsi come John McCarthy, Marvin Minsky, Claude Shannon e Nathaniel Rochester, fino a giungere ai giorni nostri che la stanno valorizzando molto dal punto di vista di sviluppo, grazie all'evoluzione dell'hardware più potente e di molte librerie software gratuite ed open source, sviluppate dai numerosi ricercatori sul web.

Dott. Nanni BASSETTI, laureato in Scienze dell'Informazione a Bari, è libero professionista specializzato in informatica forense. Iscritto all'albo dei C.T.U. presso il Tribunale di Bari. Project manager di CAINE Linux Live Distro forense, Fondatore di CFI – Computer Forensics Italy, una grande community di computer forensics italiana.

Facciamo subito una distinzione tra le varie definizioni che si intersecano nel vasto mondo dell'Intelligenza Artificiale.
L'intelligenza artificiale è un campo di sviluppo dell'informatica, che tende ad emulare l'intelligenza umana e riprodurre gli schemi di ragionamento tramite algoritmi. Gli organismi biologici, come gli esseri umani, ma anche gli animali, imparano dall'esperienza e dall'osservazione del mondo, su questo concetto si sviluppano gli algoritmi che dovrebbero imparare analizzando i dati che ricevono in input.

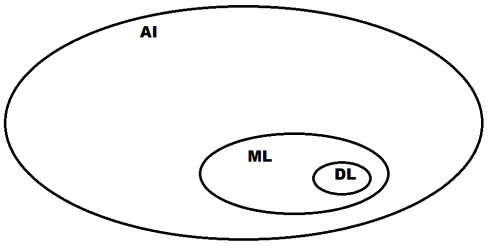


Figura 1 - Differenza tra l'AI (Artificial Intelligence), il Machine Learning (ML) ed il Deep Learning (DL)

L'AI (Artificial Intelligence) è il macro insieme che comprende il Machine Learning (ML) ed il Deep Learning (DL), che poi sono le implementazioni di metodi di intelligenza artificiale per risolvere specifici problemi, infatti non dobbiamo pensare all'AI come una "*general purpose*", ossia tipo quelle dei film di fantascienza, bensì dobbiamo pensare ad applicazioni disegnate per degli scopi ben precisi, come la classificazione, il riconoscimento, la previsione di qualcosa, ma non di tutto lo scibile umano. Pertanto, potremmo avere dei sistemi in grado di classificare fotografie, audio, video, testi, prevedere i prezzi di un mercato, l'andamento di malattie, la ricerca di molecole specifiche, la guida autonoma, il riconoscimento vocale e testuale, l'individuazione di anomalie, ecc.. Ma non avremo, almeno per ora il famoso HAL 9000 o Terminator.

Il Machine Learning è il sottoinsieme dell'IA, che basa le proprie capacità di classificazione, riconoscimento e predizione, sulle "*features*" che gli vengono date in pasto, quindi necessita di un dataset composto da caratteristiche di input (features) e di "*label*" (etichette) in certi casi come l'apprendimento supervisionato. Il Deep Learning differisce dal Machine Learning dal fatto che l'implementazione di questi sistemi si ispira alla natura del cervello, quindi si compone di input che vanno in elaborazione in una rete neurale, che emette degli output, ma la differenza principale dal ML è che le features vengono estratte dagli input grezzi, quindi non vi è bisogno di inserire un dataset numerico di caratteristiche che descrivano qualcosa, ma si può utilizzare l'input così com'è e la rete neurale troverà in autonomia le caratteristiche ed imparerà.

Esempio per capire la differenza tra l'input per un sistema di ML e per un sistema di DL.

Dataset di gatti e cani nel ML

Baffi	Orecchie	
8 cm	5 cm	**gatto**
6 cm	20 cm	**cane**

Dataset di gatti e cani nel DL

Figura 2 - Differenza tra l'input per un sistema di ML e per un sistema di DL

Come si intuisce, nel DL ci basta dare l'input di tante fotografie di cani e gatti e l'algoritmo imparerà ad estrarre le caratteristiche che li distinguono, così da creare un dataset numerico che gli servirà a classificare, questo necessità di molta potenza di calcolo rispetto al ML.

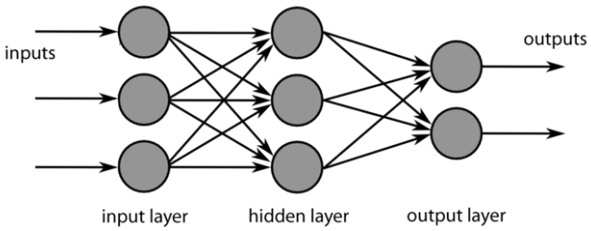
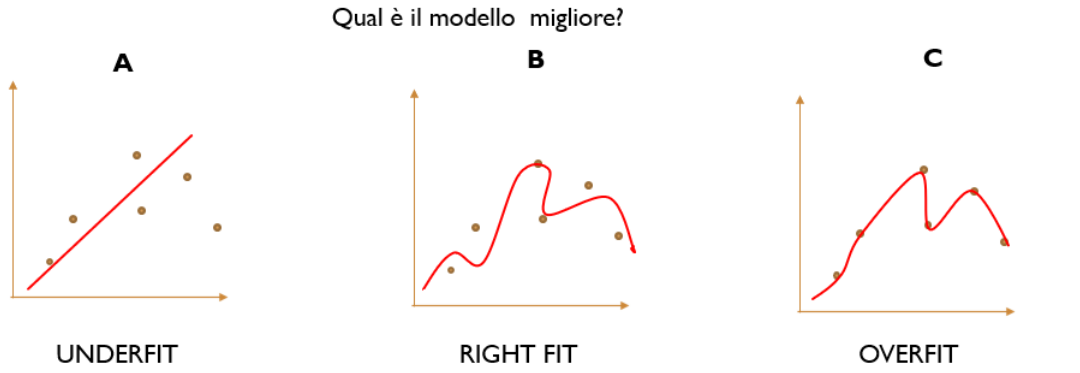


Figura 3 - Schema di una rete neurale per il Deep Learning (fonte https://commons.wikimedia.org/wiki/File:MultiLayerNeuralNetworkBigger_english.png)

Quindi una rete DL è formata da dei nodi di Input che vanno in elaborazione negli strati nascosti (hidden layers), i cosidetti neuroni, per poi generare degli output.



L'**underfit** generalizza male, troppo grandi gli errori
L'**overfit** è troppo preciso e generalizza male, di fronte ad un dato non presente nel dataset, non sa generalizzarlo non sa come classificarlo o prevederlo

Figura 4 -Fenomeno dell'overfitting e dell'underfitting

Il Machine Learning può essere di varie tipologie, le principali sono:

▶ **Supervisionato,** ossia il dataset sul quale deve addestrarsi è composto da features e label, quindi come nel seguente esempio:

Prezzo	Tipologia oggetto
120	Occhiali
40	Camicia

dove le features sono i prezzi e le label sono la tipologia di oggetto, se avessimo un dataset grande, con tante caratteristiche per gli occhiali e le camicie, alla fine se inseriamo un nuovo oggetto con certe caratteristiche di prezzo, misure, ecc.. l'algoritmo avrà imparato a classificarlo come un qualcosa presente nel dataset, quindi il nostro oggetto ignoto, sarà classificato come qualcosa di noto.

▶ **NON-Supervisionato,** in questo caso nel dataset mancano le label, quindi l'algoritmo dovrà lavorare sulla previsione o sulla classificazione tramite clustering, basato solo sulle caratteristiche numeriche, che servono a far predire all'algoritmo quale sarà il prossimo numero o come aggregare dei valori in gruppi (cluster). Esempi: andamento di un titolo di borsa o prezzi di qualcosa, ecc.

▶ **Rinforzato,** l'algoritmo ha degli agenti che provano varie strategie, se ricevono dei «premi», è una misurazione sulla qualità delle azioni intraprese dal sistema, che così facendo impara a trovare le strategie migliori.

Una caratteristica dei vari algoritmi è quella di saper generalizzare l'output, nel senso che non devono saper predire o classificare in maniera perfetta, altrimenti non riuscirebbero a comprendere un input nuovo che non fa parte di ciò che hanno imparato dai dati del dataset d'apprendimento, infatti come diceva Albert Einstein:"*L'intelligenza non è la conoscenza ma è la capacità d'immaginazione*".

Se il dataset fosse fatto solo dai colori classici, quando l'algoritmo riceve un rosso scuro o un rosso chiaro, non sarebbe in grado di classificarlo come rosso, perché lui è troppo preciso, quindi si basa su quello che ha imparato e non è in grado di generalizzare e comprendere che anche il rosso chiaro è un rosso.
Questo fenomeno si chiama *overfitting*, viceversa se non è in grado nemmeno di riconoscere il rosso normale, siamo nell'*underfitting*, ossia non ha imparato per niente bene e commette parecchi errori.

Ma com'è fatto un neurone? Senza addentrarci nella matematica, il neurone funziona a grandi linee in questa maniera: riceve l'input, poi crea una funzione d'attivazione, che è composta da una combinazione lineare tra i valori degli input e dei pesi (valori numerici) arbitrari, infine calcola la differenza tra il valore della previsione ed il valore della funzione d'attivazione, tramite una funzione di costo; se è alto, allora ricalcola i pesi e ci riprova fino a che il valore della funzione di costo non si minimizza, così la differenza tra l'output atteso e la previsione è minima e quindi ha imparato a

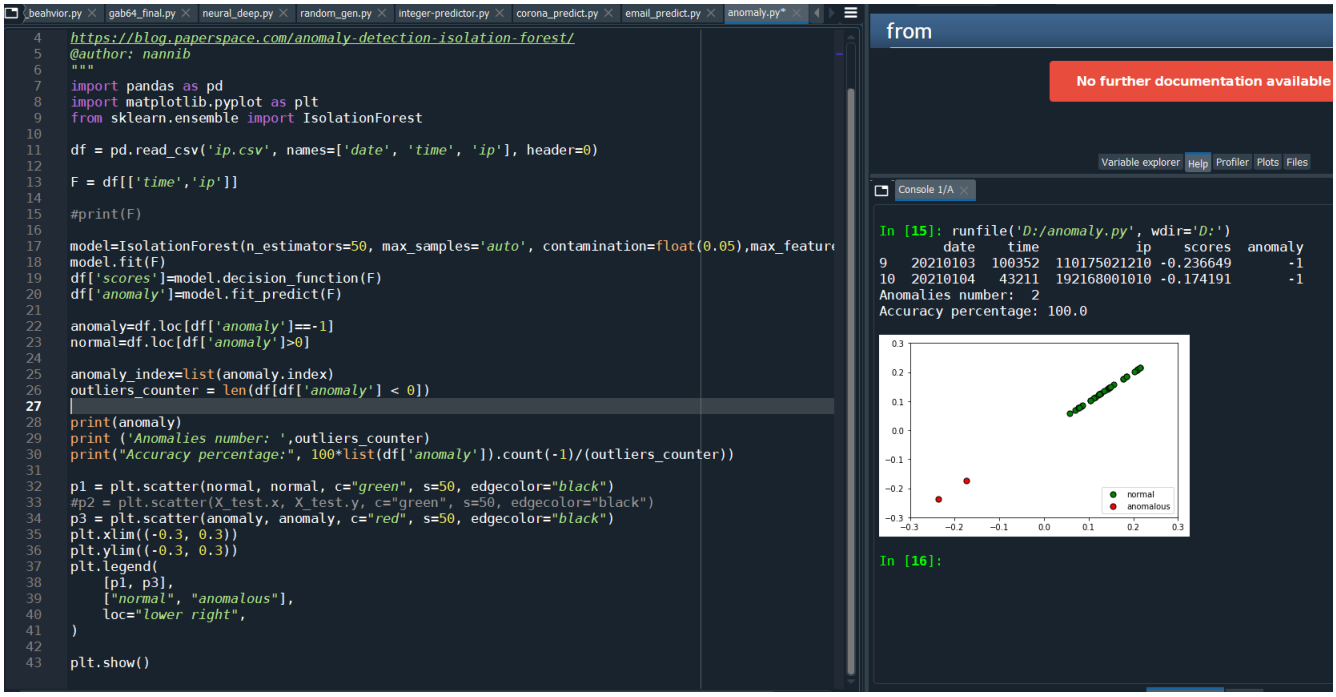


Figura 5 - Machine learning per individuare le connessioni

riconoscere bene l'input.
Con questo sistema, la rete neurale è in grado di imparare e classificare al meglio un oggetto in input.

In sintesi, quindi, gli input vanno nel neurone che calcola la previsione *forward propagation*, poi si ricalcolano, i parametri w1,w2 e b, (*back propagation*), per minimizzare la funzione di costo e quindi avvicinarsi alla previsione alla fine questo avanti ed indietro trova la miglior previsione e la rete impara.
Ma per avere il valore corretto dei pesi, bisogna conoscere l'output reale, quello che corrisponde all'immissione di quell'input, per far questo si addestra la rete dividendo il dataset in due insiemi, un training set ed un testing set, quindi per esempio, su 100 elementi del dataset 70 sono di training, quindi li utilizzeremo come output noti e 30 di testing, ossia faranno finta di essere degli input nuovi e sconosciuti, ma in realtà si sa quali sono perché sono nel dataset.

Quando l'algoritmo si addestra può capire quante volte indovina bene e quante no, quando viene raggiunto un risultato soddisfacente, allora si possono conservare i pesi ed il modello in un file, così la rete non dovrà ogni volta ri-addestrarsi, ma partirà già da un modello pre-addestrato che funzionava bene.
Chiaramente il risultato soddisfacente si basa sulla grandezza del dataset, da come è fatto, dagli iperparametri dell'algoritmo, ossia tanti settaggi che si possono fissare in fase di programmazione, come il numero di iterazioni, ecc.

Applicazioni nella cybersecurity e nella digital forensics

È chiaro che sistemi del genere possono esser utilizzati nella cybersecurity e nella digital forensics per automatizzare alcuni processi, che possono andare dal riconoscimento dell'autore di un testo, al riconoscimento di un'anomalia di un sistema o di qualche connessione, come nel riconoscere volti, oggetti, ecc.
Inoltre, grazie alle innumerevoli librerie e dataset sviluppate dalla community di riferimento, con poche righe di codice si possono realizzare dei modelli anche solo a fini educativi.

La connessione anomala

Pensiamo ad un dataset di connessioni formato da indirizzi IP ed orari di connessione, quindi un tabulato, potremmo addestrare l'algoritmo a capire qual è la connessione "anomala" e basterebbero poche righe di codice (**figura 5**).
L'algoritmo sfrutta l'Isolation Forest, un algoritmo non supervisionato e quindi non necessita di etichette per identificare il valore anomalo / anomalia. Il fondamento dell'algoritmo è quello di "isolare" le anomalie creando alberi decisionali su attributi casuali, questo partizionamento casuale produce percorsi più brevi per le anomalie quindi, se ci sono poche anomalie ci saranno partizioni più piccole di dati, perché i valori delle anomalie hanno maggiori probabilità di essere separati nel partizionamento iniziale
Quindi, quando una foresta di alberi casuali produce lunghezze di percorso più brevi per alcuni punti particolari, è molto probabile che siano anomalie.

Ricerca dell'autore di un testo

In questo caso avendo un dataset di testi scritti da vari autori, si può addestrare la macchina a capire chi ha scritto un testo nuovo che forniamo in input (**figura 6**).
Qui si tratta di convertire le parole in numeri e poi calcolare dei valori, quindi un testo nuovo avrà un valore numerico, se la distanza tra il valore del nuovo testo ed il valore di altri testi di un certo autore, allora quel testo probabilmente apparterrà a quell'autore.

Gli algoritmi genetici

Per completezza è giusto anche citare gli algoritmi genetici, che voglio inserire nel grande mondo dell'AI, essi sono degli algoritmi che imitano l'evoluzione della specie per giungere ad un obbiettivo, quindi partendo da genitori casuali, essi genereranno figli, che tramite cross-over e mutazioni giungeranno a produrre l'oggetto target.

Conclusioni

In sostanza tutto si riduce ad analisi statistiche di oggetti ridotti a vettori o matrici di numeri, tutto è saper calcolare la vicinanza di questi numeri tramite complesse operazioni matematiche e generare un responso, guardare sotto il cofano dell'AI magari toglie un po' di quel senso "magico" che si immaginava, ma rende meglio la realtà e ci fa capire il motivo per cui non esiste ancora un "cervellone" in grado di risolvere qualsiasi problema, di avere autocoscienza, evolversi in più direzioni, siamo ancora lontani da quello.
L'argomento è vasto, complicato ma affascinante, per questo ci vengono in aiuto tantissimi strumenti open source, distribuiti apposta in questa modalità proprio perché la comunità scientifica vuole il massimo della collaborazione e condivisione della conoscenza per fare passi avanti nello sviluppo dell'AI.
Abbiamo strumenti e librerie come Sklearn, TensorFlow, Keras ma anche innumerevoli programmi su piattaforme come GitHub, oltre a framework gratuiti di sviluppo come Anaconda 3, Spyder, PyCharm, solo per parlare del linguaggio Python, ma c'è tanto anche i C++ e Java. Non mancano, inoltre, numerosi tutorial in varie lingue sia sul web testuale sia su YouTube. Ed infine molti dataset già pronti all'uso. Quindi non ci sono scuse, se la cosa vi interessa, tutto ciò che vi serve è a portata di clic, almeno per iniziare. ©

APPROFONDIMENTI

- https://nannibassetti.blogspot.com/2021/01/machine-learning-costruiamo-un.html
- https://nannibassetti.blogspot.com/2020/05/come-funziona-un-neurone-di-una-rete.html
- https://www.cybersecurity360.it/soluzioni-aziendali/algoritmi-genetici-e-machine-learning-nella-digital-forensics-applicazioni-pratiche/
- https://nannibassetti.blogspot.com/2021/02/machine-learning-attribuire-la.html
- https://nannibassetti.blogspot.com/2021/03/machine-learning-per-il-rilevamento.html.

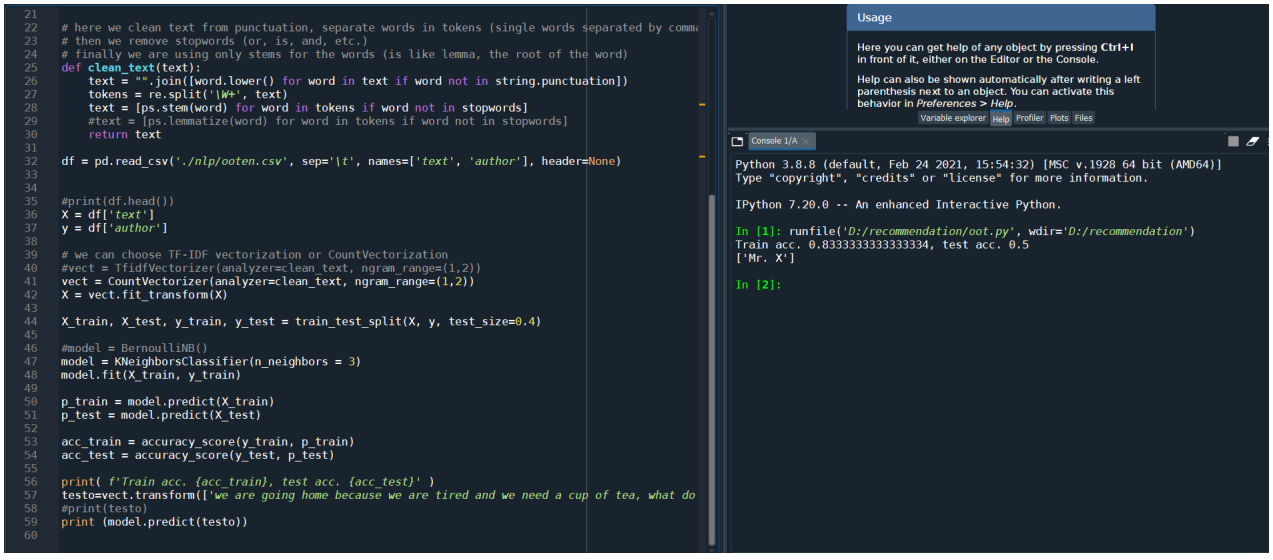


Figura 6 - Machine Learning per identificare l'autore di un testo