

LINUX DAY 2021

REINFORCEMENT LEARNING: PREMIAMI E IMPARERÒ!

ORNELLA FANAIS



DATI A CHI?

WWW.LINUXDAY.IT
23 OTTOBRE 2021



DI COSA PARLEREMO DURANTE IL TALK

SOMMARIO

- 01 ALPHAGO
- 02 MACHINE LEARNING
- 03 REINFORCEMENT LEARNING
- 04 AMBIENTE OPENAI GYM
- 05 AGENTE TF-AGENTS
- 06 ADDESTRAMENTO ATTRAVERSO DEEP-LEARNING
- 07 ELABORAZIONE DEI DATI
- 08 COLAB E GPU



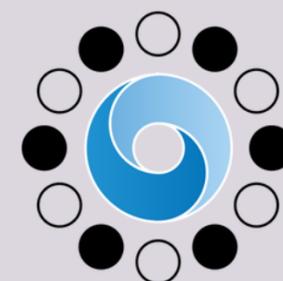
"I THOUGHT ALPHAGO WAS BASED ON PROBABILITY CALCULATION AND THAT IT WAS MERELY A MACHINE. BUT WHEN I SAW THIS MOVE, I CHANGED MY MIND. SURELY, ALPHAGO IS CREATIVE."

LEE SEDOL
WINNER OF 18 WORLD GO TITLES



DEEP MIND: IL CASO ALPHAGO

IL GIOCO DI GO È STATO A LUNGO CONSIDERATO IL PIÙ IMPEGNATIVO TRA I GIOCHI STRATEGICI A CAUSA DELLA DIFFICOLTÀ DI VALUTARE LE POSIZIONI E GLI SPOSTAMENTI.



AlphaGo

SUPERVISED LEARNING POLICY NETWORK

ROLLOUT POLICY CHE CONSISTE NEL GIOCARE LA STESSA POSIZIONE E REGISTRARE IL RISULTATO. UNA RETE SI VIENE ADDESTRATA CON DELLE MOSSE PROVENIENTI DA UN DATASET DI PARTITE.

12 LAYER CONVOLUTIONAL NEURAL NETWORK
4 WEEKS ON 50 GPUS

REINFORCEMENT LEARNING POLICY NETWORK

L'APPRENDIMENTO VIENE MIGLIORATO DALLA VALUTAZIONE DEL NUMERO DI PARTITE VINTE. SI CREA UN DATASET CON UNA POLICY NETWORK CHE SCEGLIE I MOVIMENTI.

12 LAYER CONVOLUTIONAL NEURAL NETWORK
1 WEEKS ON 50 GPUS

REINFORCEMENT LEARNING VALUE NETWORK

ATTRAVERSO UNA REGRESSIONE VIENE PRONOSTICATO IL RISULTATO UTILIZZANDO UNA VALUE NETWORK PER DETERMINARE LE POSIZIONI SUL GOBAN.

12 LAYER CONVOLUTIONAL NEURAL NETWORK
1 WEEKS ON 50 GPUS



DATI A CHI?

WWW.LINUXDAY.IT
23 OTTOBRE 2021



TIMELINE DEL REINFORCEMENT LEARNING

IL REINFORCEMENT LEARNING (RL) È UNO DEI CAMPI PIÙ INTERESSANTI NELL'AMBITO DEL MACHINE LEARNING OLTRE AD ESSERE UNO DEI PIÙ ANTICHI.



1992
TD-GAMMON

APPRENDE IL GIOCO DEL BACKGAMMON CON UNA MINIMA CONOSCENZA DELLE REGOLE AVVICINANDOSI AL LIVELLO DEI MAESTRI PIÙ FORTI DEL MONDO.



2013
ATARI 2600

IMPARA A GIOCARE DA ZERO A QUALSIASI GIOCO DELL'ATARI 2600 SENZA ALCUNA CONOSCENZA PRELIMINARE DELLE REGOLE.



2016
ALPHAGO

ALPHAGO SCONFIGGE IL LEGGENDARIO LEE SEDOL, VINCITORE DI 18 TITOLI MONDIALI, FAMOSO PER LA SUA CREATIVITÀ E CONSIDERATO IL PIÙ GRANDE GIOCATORE DELL'ULTIMO DECENNIO.



2050
ROBOCUP

INIZIATIVA IDEATA NEL 1993 E AVVIATA NEL 1997 CON L'OBIETTIVO DI REALIZZARE, ENTRO IL 2050, UNA SQUADRA DI ROBOT UMANOIDI AUTONOMI.

MACHINE LEARNING

IL MACHINE LEARNING È OGGI RITENUTO UNO DEGLI APPROCCI PIÙ IMPORTANTI DELL'INTELLIGENZA ARTIFICIALE.

UN SISTEMA DI MACHINE LEARNING DURANTE LA FASE DI TRAINING APPRENDE A PARTIRE DA ESEMPI (IN MODO PIÙ O MENO SUPERVISIONATO) E SUCCESSIVAMENTE SARÀ IN GRADO DI GENERALIZZARE E GESTIRE NUOVI DATI NELLO STESSO DOMINIO APPLICATIVO.

SUPERVISED LEARNING

CONSISTE NEL TRARRE UN MODELLO A PARTIRE DAI DATI DI ADDESTRAMENTO ETICHETTATI (TRAINING SET) I QUALI CONSENTONO DI EFFETTUARE PREVISIONI RELATIVE A DATI NON DISPONIBILI. UN ESEMPIO È L'UTILIZZO DI UN ALGORITMO DI CLASSIFICAZIONE CHE PERMETTA IL FILTRAGGIO DEI MESSAGGI SPAM DI POSTA ELETTRONICA,

UNSUPERVISED LEARNING

II DATI DI ADDESTRAMENTO NON SONO ETICHETTATI. IL SISTEMA CERCA DI IMPARARE SENZA POTER CONTARE SULLA GUIDA DI UNA VARIABILE NOTA. UN TIPOICO ESEMPIO È IL CLUSTERING CHE CONSENTE DI ORGANIZZARE UNA SERIE DI INFORMAZIONE ALL'INTERNO DI GRUPPI SIGNIFICATIVI COME ACCADE PER LE INDAGINI DI MARKETING.

SEMISUPERVISED LEARNING

POSSONO GESTIRE DATI PARZIALMENTE ETICHETTATI. ALCUNI SERVIZI DI HOSTING DI FOTO PERMETTONO IL RICONOSCIMENTO PER ESEMPIO DEI VOLTI ADOTTANDO IL CLUSTERING ED È SUFFICIENTE ETICHETTARE UN SOLO VOLTO PER CLASSIFICARE GLI ALTRI.

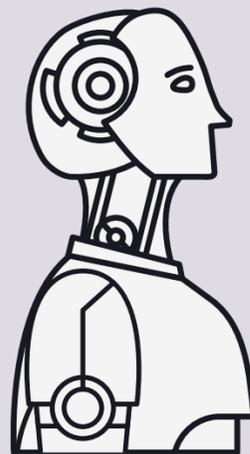
REINFORCEMENT LEARNING

IL SISTEMA DI APPRENDIMENTO CHIAMATO AGENTE OSSERVA L'AMBIENTE, LO AIUTA A SELEZIONARE ED ESEGUIRE AZIONI OTTENENDO IN CAMBIO UNA RICOMPENSA. CASSIE IL ROBOT BIPEDE DELL'AGILITY ROBOTICS ADOTTA QUESTA TIPOLOGIA DI ALGORITMI PER IMPARARE A CAMMINARE.



REINFORCEMENT LEARNING

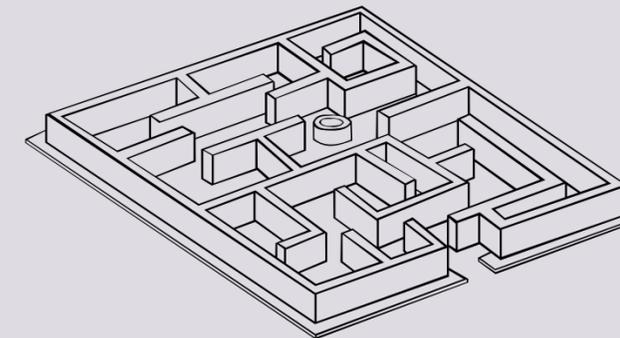
L'APPRENDIMENTO DELL'AGENTE È DATO DALLA CONTINUA INTERAZIONE CON L'AMBIENTE AL FINE DI RAGGIUNGERE IL CORRETTO ADDESTRAMENTO CHE DEVE CULMINARE CON LA MASSIMIZZAZIONE DELLA RICOMPENSA.



AGENTE

AD OGNI STEP:

- ESEGUE L'AZIONE
- RICEVE INFORMAZIONI SULLO STATO
- RICEVE LA RICOMPENSA



AMBIENTE

AD OGNI STEP:

- RICEVE L'AZIONE
- EMETTE UNO STATO (MODIFICA IL SUO STATO)
- EMETTE LA RICOMPENSA



DATI A CHI?

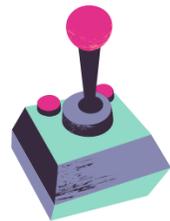
WWW.LINUXDAY.IT
23 OTTOBRE 2021



AMBIENTE RL

È L'UNIVERSO IN CUI ESISTE L'AGENTE.

L'AMBIENTE SI TROVA SEMPRE IN UNO STATO SPECIFICO CHE VIENE MODIFICATO DALLE AZIONI CHE L'AGENTE INTRAPRENDE.



AZIONI

AZIONI ESEGUITE DALL'AGENTE PER
MODIFICARE LO STATO DELL'AMBIENTE



STATO

INFORMAZIONI CHE L'AMBIENTE FORNISCE
ALL'AGENTE.



RICOMPENSA

FORNISCE ALL'AGENTE UN FEEDBACK
RELATIVO ALL'AZIONE INTRAPRESA.

AMBIENTE OPENAI GYM

È UN TOOLKIT OPEN-SOURCE CHE FORNISCE UN'AMPIA VARIETÀ DI AMBIENTI SIMULATI RILASCIATO NEL 2016 DA ELON MUSK E SAM ALTMAN, CON LO SCOPO DI FORNIRE UN BENCHMARK DI INTELLIGENZA ARTIFICIALE FACILE DA CONFIGURARE E UTILIZZABILE CON PYTHON.

OpenAI

01

action space

INSIEME DI AZIONI CHE POSSONO ESSERE ESEGUITE E SONO SIA DISCRETE CHE CONTINUE

02

observation space

INSIEME DI OSSERVAZIONI FORNITE ALL'AGENTE CHE IDENTIFICANO LO STATO DELL'AMBIENTE NELLE FASI TEMPORALI

03

step

ESEGUIRE UN'AZIONE CHE RESTITUISCE LA CORRENTE OSSERVAZIONE, RICOMPENSA E L'INDICAZIONE SULLA FINE DELL'EPISODIO

04

reset

RIPORTA L'AMBIENTE AL SUO STATO INIZIALE OTTENENDO LA PRIMA OSSERVAZIONE DI INIZIALIZZAZIONE

05

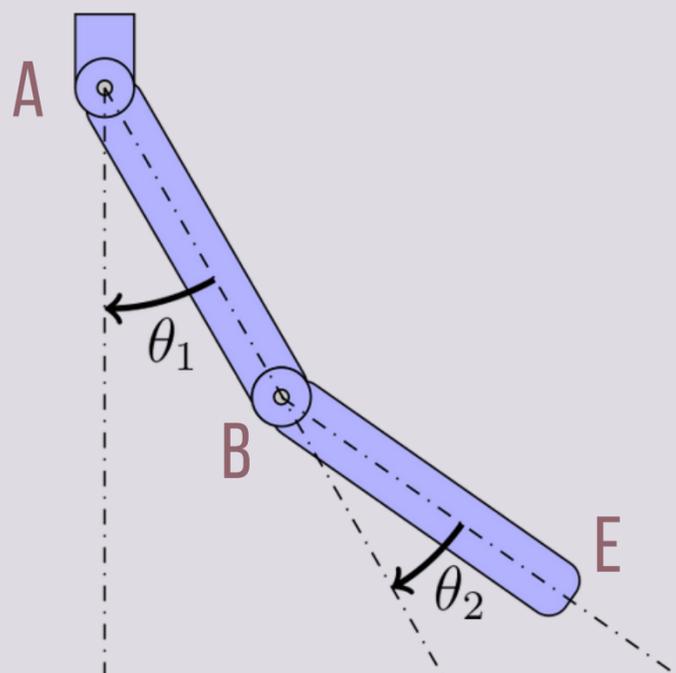
render

RAPPRESENTA IN FORMA GRAFICA L'AMBIENTE



SISTEMA ROBOTICO: ACROBOT

OBIETTIVO: FAR OSCILLARE L'END-EFFECTOR SOPRA IL PRIMO GIUNTO DI UNA QUANTITÀ PARI A UNO DEI COLLEGAMENTI IN UN TEMPO MINIMO.



VARIABILI DI STATO

- 4 POSIZIONI DEI GIUNTI
- 2 VELOCITÀ ANGOLARI DEI GIUNTI

COPPIA APPLICATA AL GIUNTO B

- (+1) UNA COPPIA POSITIVA
- (-1) UNA COPPIA NEGATIVA
- (0) NESSUNA COPPIA APPLICATA.

RICOMPENSA

- (-1) PER OGNI STEP ESEGUITO,
- (0) ALLA FINE DELL'EPISODIO.

 **DATI A CHI?**

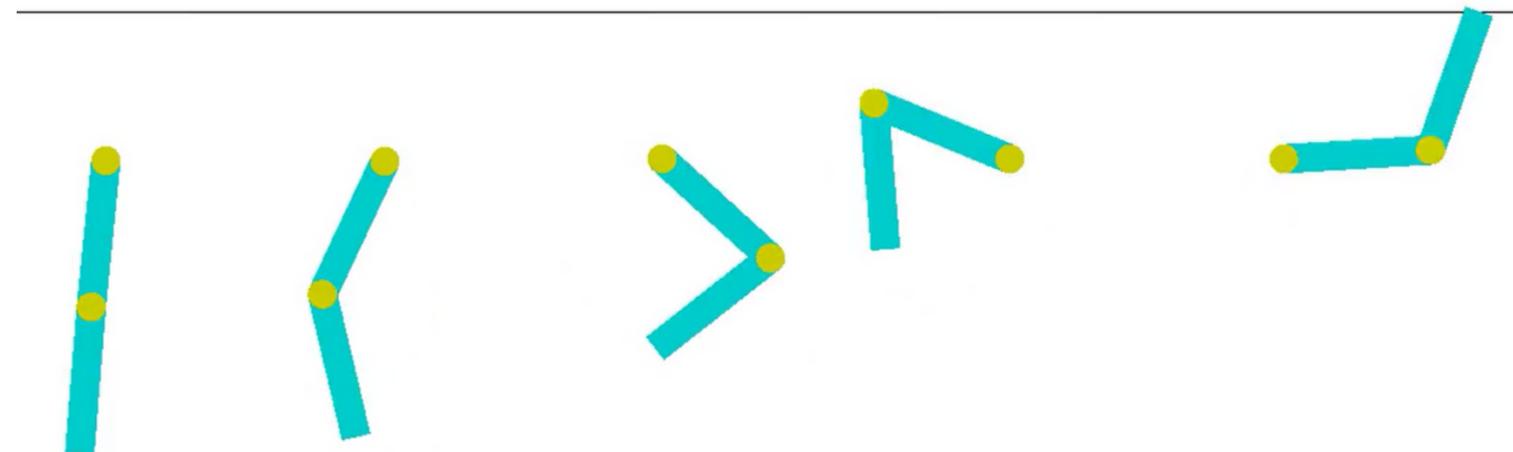
WWW.LINUXDAY.IT
23 OTTOBRE 2021



ENVIRONMENT OPENAI GYM: ACROBOT-V1

CARATTERISTICHE DELL'ESPERIMENTO:

- CONFIGURAZIONE INIZIALE: COLLEGAMENTI RIVOLTI VERSO IL BASSO
- L'OBIETTIVO È RAGGIUNTO PER UNA RICOMPENSA MEDIA PARI A -100 PER EPISODIO
- L'EPISODIO TERMINA AL RAGGIUNGIMENTO DELL'OBIETTIVO OPPURE VIENE "FORZATO" L'ARRESTO AL RAGGIUNGIMENTO DEI 500 STEP.



`__init__`

- OBSERVATION_SPACE (POSIZIONE DEI GIUNTI
VELOCITÀ ANGOLARE DEI GIUNTI)
- ACTION_SPACE (0 = POSITIVA, 1 = NEGATIVA, 2 =
NESSUNA)

`Step()`

- SCELTA DELL'AZIONE RANDOMICA
- DINAMICA DEL SISTEMA E IL METODO NUMERICO RK
PER RICAVARNE LA SOLUZIONE
- ATTRIBUZIONE DEL REWARD

`Reset()`:

- RIPORTA IL SISTEMA ALLA
CONFIGURAZIONE INIZIALE

WORKING WITH OPENAI GYM

```
import gym
gym.make("Acrobot-v1")
```

MODULO E ENVIRONMENT

```
env.observation_space
env.action_space
```

BOX([-28.274333953857422, 28.274333953857422, (6,), FLOAT32])

DISCRETE(3)

```
env.reset()
```

ARRAY([0.99987675, -0.01570013, 0.99984668, 0.01751027, 0.00964072, 0.08657703]).

```
env.step(action=0)
```

ARRAY([0.99999847, 0.001749 , 0.99999471, -0.0032539 , 0.16044516, -0.28752225]), -1.0, FALSE, {})

```
for episode in range(10):
    obs = env.reset()
    for step in range(50):
        action = env.action_space.sample()
        nobs, reward, done, info = env.step(action)
        print(f"Step {step}: State={nobs}, Reward={reward}")
```



AGENTE RL

È L'ELEMENTO CHE INTRAPRENDE L'AZIONE CHE INFLUENZA LO STATO DELL'AMBIENTE AL FINE DI RICEVERE LA RICOMPENSA



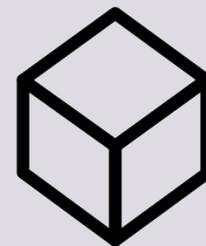
POLICY

RAPPRESENTA IL MODO IN CUI L'AGENTE SCEGLIE DI COMPIERE UN'AZIONE PER RAGGIUNGERE L'OBIETTIVO

$f(x)$

FUNZIONE VALORE

È LA MISURA DELLA BONTÀ DELL'AZIONE SCELTA



MODELLO MATEMATICO MARKOV DECISION

PROCESS

STRUMENTO DI CALCOLO



DATI A CHI?

WWW.LINUXDAY.IT
23 OTTOBRE 2021



DEEP LEARNING

NEL 2013 DEEPMIND DIMOSTRÒ CHE L'UTILIZZO DI RETI NEURALI PROFONDE (DEEP NEURAL NETWORKS DNN) PUÒ FUNZIONARE MOLTO MEGLIO, SOPRATTUTTO PER PROBLEMI COMPLESSI.



NEURONI ARTIFICIALI

È UTILIZZATO ALL'INTERNO DI UNA CLASSICA RETE NEURALE. AD OGNI NEURONE APPARTENENTE AD UNA RETE È ATTRIBUITO UN PESO.



ARTIFICIAL NEURAL NETWORK

I NEURONI ARTIFICIALI SONO DISPOSTI SU STRATI E OGNUNO È CONNESSO IN MODO TALE CHE LE INFORMAZIONI POSSANO PROPAGARSI DA UN LIVELLO ALL'ALTRO.
I LIVELLI PRESENTI SONO QUELLO DI INPUT/OUTPUT E HIDDEN.



GRADIENT DESCENT

È UN ALGORITMO DI OTTIMIZZAZIONE GENERICO CON L'OBIETTIVO DI MINIMIZZARE LA FUNZIONE DI PERDITA. LA RETE VIENE PERCORSO IN FORWARD/BACK PROPAGATION IN MODO DA AGGIORNARE I PESI.

DEEP Q LEARNING

È UN METODO DI RL DI APPROSSIMAZIONE DI FUNZIONE CHE RAPPRESENTA UN'EVOLUZIONE DEL METODO Q-LEARNING IN CUI UNA TABELLA CONTENENTE LE FUNZIONI VALORE (S, A) VIENE SOSTITUITA DA UNA RETE NEURALE.

01

EQUAZIONE DI BELLMAN

CONTIENE LE INFORMAZIONI RELATIVE ALLO STATO PRECEDENTE, QUELLO ATTUALE, LA RICOMPENSA E GLI IPERPARAMETRI.

02

PREDICT NETWORK

UTILIZZATA PER PREVEDERE I VALORI Q E APPRENDE I PESI W CORRETTI UTILIZZANDO IL GRADIENT DESCENT.

03

TARGET NETWORK

VIENE CONGELATA PER DIVERSI PASSAGGI DI TEMPO MENTRE VENGONO AGGIORNATI E COPIATI I PESI DALLA PREDICT NETWORK.

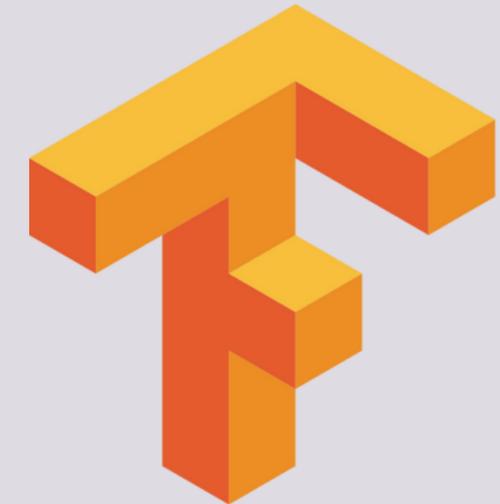
04

EXPERIENCE REPLAY

LE INFORMAZIONI DI TRANSIZIONE < S; A; R; S₀ > DEVONO ESSERE SALVATE ALL'INTERNO DI UN BUFFER

TENSORFLOW E TF-AGENTS

È UNA LIBRERIA OPEN SOURCE DI REINFORCEMENT LEARNING, DEFINITA LO STATO DELL'ARTE DI TENSORFLOW. È MODULARE E HA COME OBIETTIVO QUELLO DI RENDERE LA PROGRAMMAZIONE AI FRUIBILE AD UN NUMERO SEMPRE MAGGIORE DI UTENTI.



OPENAI GYM VS TF-AGENTS

METODI:

- `RESET()`: RESTITUISCE UN OGGETTO `TIMESTEP` CHE CONTIENE L'OSSERVAZIONE INIZIALE;
- `STEP()` APPLICA UN'AZIONE E RESTITUISCE UNA TUPLA CONTENENTE LO STATO, LA RICOMPENSA, IL `DONE` E LE EVENTUALI `INFO`;

DA ARRAY A TENSORI:

- UNA PRE-ELABORAZIONE DELL'AMBIENTE GYM E LA SUA SUCCESSIVA CONVERSIONE IN AMBIENTE TENSORFLOW

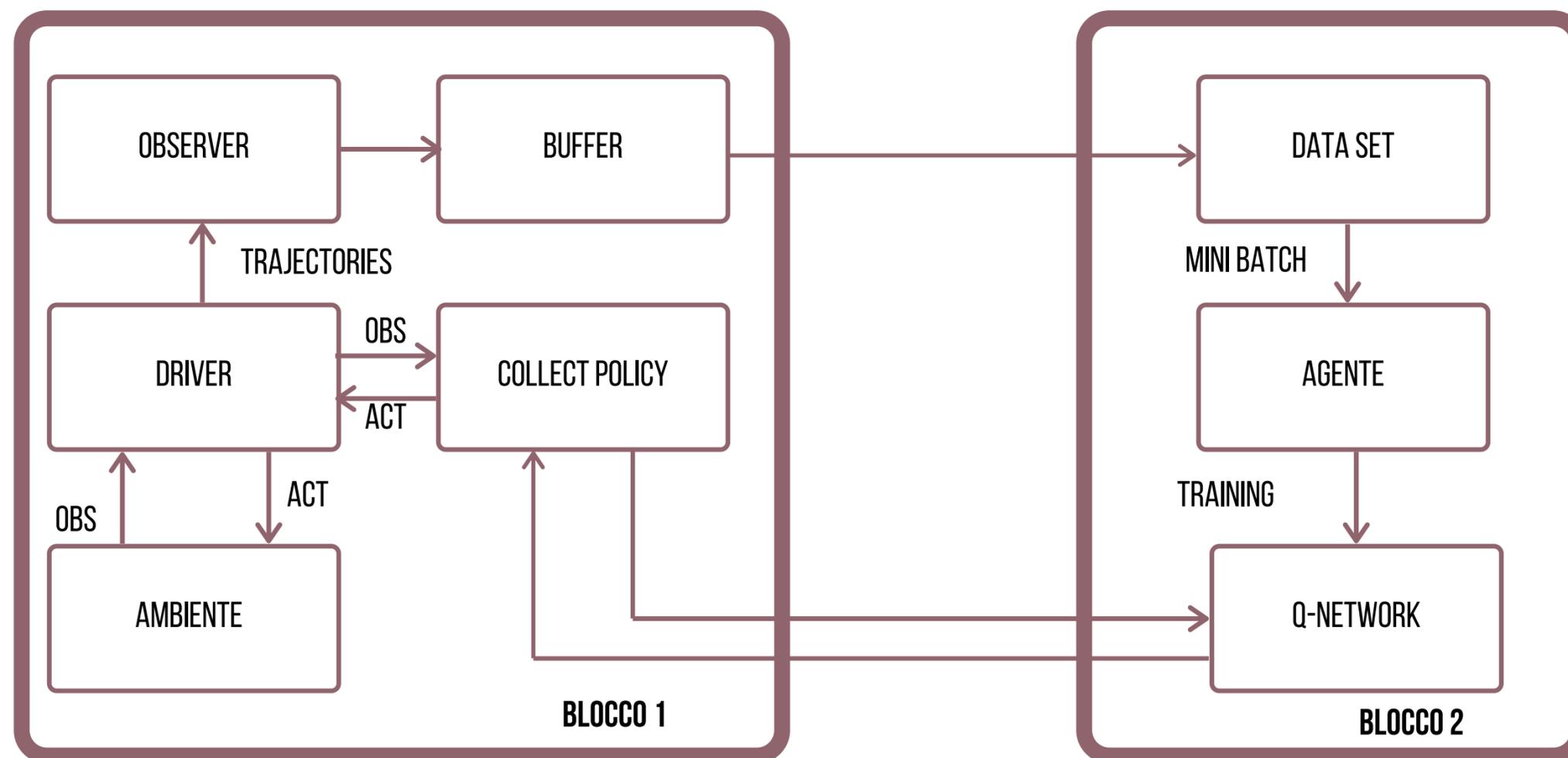


DATI A CHI?

WWW.LINUXDAY.IT
23 OTTOBRE 2021



STRUTTURA DI UN AGENTE TF-AGENT



- **BLOCCO 1**
ESPLORA L'AMBIENTE E RACCOGLIE LE ESPERIENZE
- **BLOCCO 2**
APPRENDE E AGGIORNA LA POLICY

MODULO: ENVIRONMENT

QUESTO PRIMO MODULO PREVEDE IL CARICAMENTO E LA CONVERSIONE DI DUE AMBIENTI:

- EVAL_ENV VERRÀ UTILIZZATO PER LA VALUTAZIONE;
- TRAIN_ENV VERRÀ UTILIZZATO PER IL TRAINING.

```
import tensorflow as tf
from tf_agents . environments import suite_gym
from tf_agents . environments import tf_py_environment
```

```
train_py_env = suite_gym.load(env_name)
eval_py_env = suite_gym.load(env_name)
train_env = tf_py_environment.TFPyEnvironment(train_py_env)
eval_env = tf_py_environment.TFPyEnvironment(eval_py_env)
```

`suite_gym.load`

RAPPRESENTA IL CARICAMENTO
DELL'AMBIENTE GYM;

`tf_py_environment.TFPyEnvironment`

PERMETTE LA CONVERSIONE TRA UN AMBIENTE OPENAI GYM
SVILUPPATO MEDIANTE NUMPY (ARRAY) IN UN
TFPYENVIRONMENT (TENSORI).



MODULO: Q-NETWORK

```
from tf_agents . networks import q_network

fc_layer_params = (100 ,32 ,32 ,100)
q_net = q_network.QNetwork(
train_env.observation_spec(),
train_env.action_spec(),
fc_layer_params = fc_layer_params)
```

`fc_layer_params`

RAPPRESENTA IL NUMERO E LA
DIMENSIONE DEI LIVELLI NASCOSTI CHE IN
QUESTO CASO SONO DI TIPO DENSE CON
FUNZIONE DI ATTIVAZIONE RELU.

`train_env.observation_spec()`

RAPPRESENTA LO SPAZIO DELLE OSSERVAZIONI

`train_env.action_spec()`

RAPPRESENTA LO SPAZIO DELLE AZIONI

 **DATI A CHI?**

WWW.LINUXDAY.IT
23 OTTOBRE 2021 

MODULO: DEEP Q NETWORK

```
from tf_agents . agents . dqn import dqn_agent

optimizer = tf.compat.v1.train.AdamOptimizer (
    learning_rate = learning_rate)
train_step_counter = tf.Variable (0)
agent = dqn_agent.DqnAgent(
    train_env.time_step_spec(),
    train_env.action_spec(),
    q_network =q_net,
    optimizer = optimizer,
    td_errors_loss_fn = common.element_wise_squared_loss,
    train_step_counter = train_step_counter)
```

`time_step_spec()`
`action_spec`

- CONTIENE LE INFORMAZIONI RELATIVE ALL'OSSERVAZIONE, ALLA RICOMPENSA, AL VALORE DISCOUNT E AL TIME STEP;
- ACTION_SPEC FORNISCE LE INFORMAZIONI RELATIVE ALL'AZIONE;

`optimizer`

RICHIÈDE UN OTTIMIZZATORE IN QUESTO CASO ADAM OPTIMIZER.

`element_wise_squared_loss`

È LA FUNZIONE DI PERDITA

MODULI: POLICY E REPLAY BUFFER

```
eval_policy = agent.policy  
collect_policy = agent.collect_policy
```

```
replay_buffer =  
tf_uniform_replay_buffer.TFUniformReplayBuffer (  
data_spec = agent.collect_data_spec ,  
batch_size = train_env.batch_size ,  
max_length = replay_buffer_max_length)
```

POLICY

- AGENT.POLICY: LA POLICY PRINCIPALE UTILIZZATA PER LA VALUTAZIONE E LA DISTRIBUZIONE.
- AGENT.COLLECT_POLICY: UN SECONDO CRITERIO UTILIZZATO PER LA RACCOLTA DEI DATI

REPLAY BUFFER

- TIPOLOGIA DI BUFFER: LINEARE
- RAPPRESENTA LE SPECIFICHE RELATIVE ALLE TRAIETTORIE CHE IL BUFFER MEMORIZZERÀ;
- RAPPRESENTA LA DIMENSIONE DEI BATCHES;
- LUNGHEZZA MASSIMA DEL BUFFER.

CREAZIONE DI UN DATA SET

IL BUFFER DI RIPRODUZIONE DEVE ESSERE IN GRADO DI RACCOGLIERE LE TRAIETTORIE IN MANIERA TALE DA CREARE UN DATA SET DI ESPERIENZE.



COLLECT_STEP

INIZIALIZZA L'AMBIENTE AL CURRENT_TIME_STEP,
GENERA L'AZIONE ATTRAVERSO LA POLICY E RICAVA
LA TUPLA DELLE TRAIETTORIE

```
dataset = replay_buffer . as_dataset (  
num_parallel_calls =3,  
sample_batch_size = batch_size ,  
num_steps =2).prefetch (3)
```



COLLECT_DATA

LA FUNZIONE DI ITERARE LA FUNZIONE DEF
COLLECT_STEP PER UN CERTO NUMERO DI STEP,
GENERANDO UNA TUPLA COLLECT_STEP

DATA SET

- NUMERO DI STEP DI CAMPIONAMENTO
- MISURA DEL BATCH
- NUMERO DI ELEMENTI IN PARALLELO CHE ELABORA



DATI A CHI?

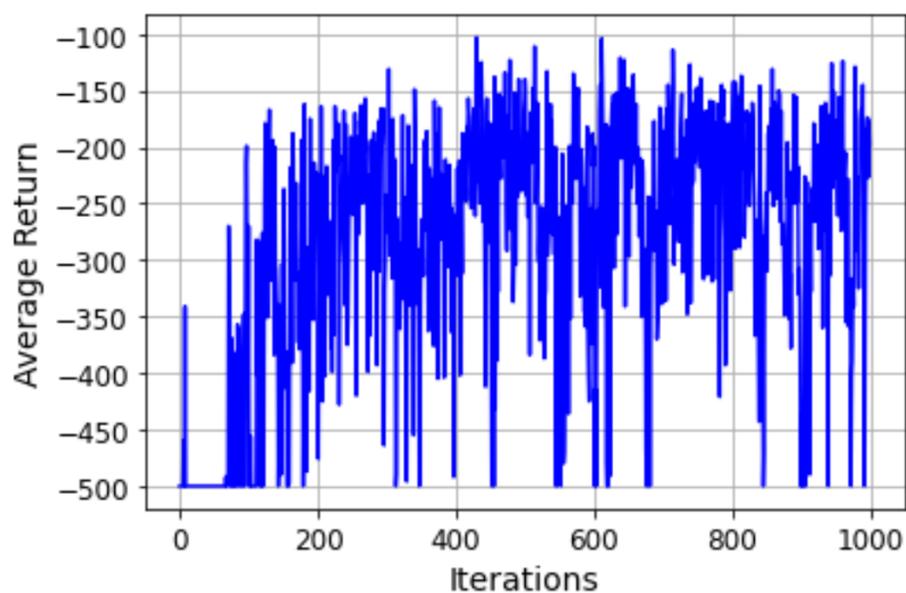
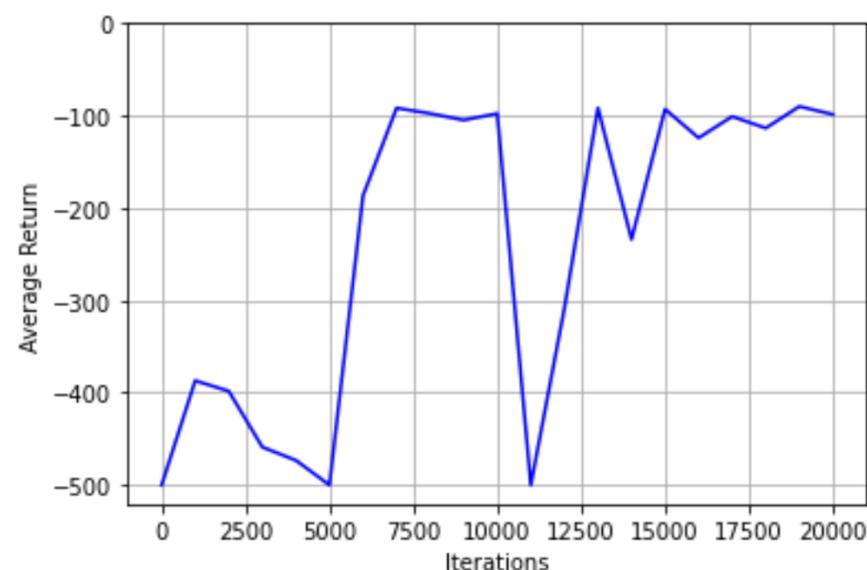
WWW.LINUXDAY.IT
23 OTTOBRE 2021



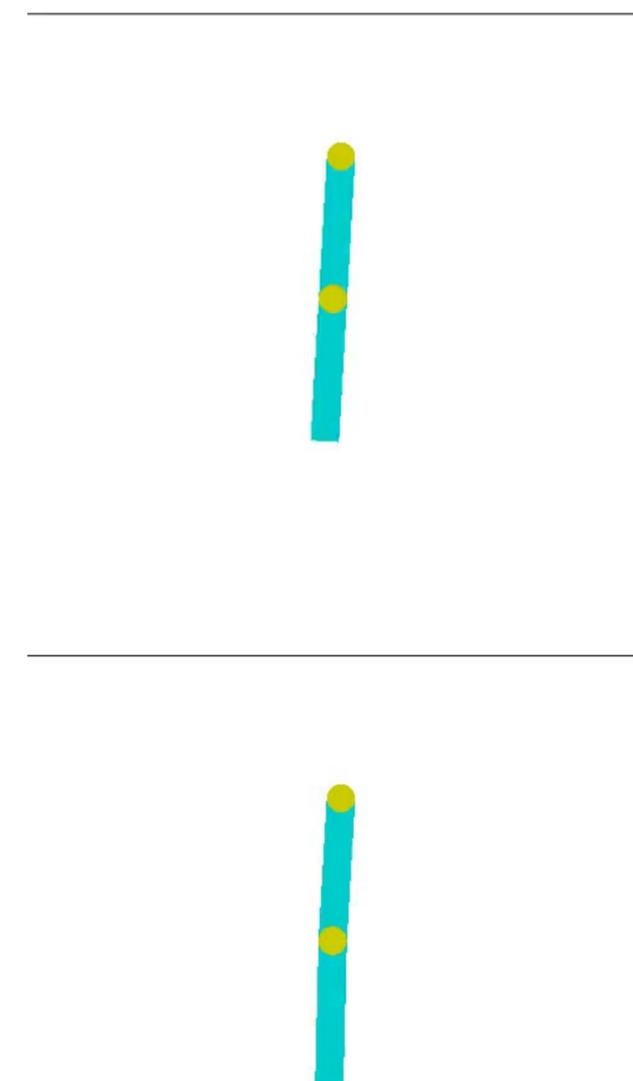
RISULTATI DEL TRAINING

DATI E IPERPARAMETRI IMPOSTATI PER L'ESPERIMENTO:

- N. LIVELLI 3
- NODI HIDDEN (100,32,100)
- N. ITERAZIONI 20000
- N. INITIAL COLLECT STEPS 10000
- N. DATI REPLAY BUFFER 100000
- BATCH SIZE 64
- LEARNING RATE 1E-3
- N. AGGIORNAMENTI 500



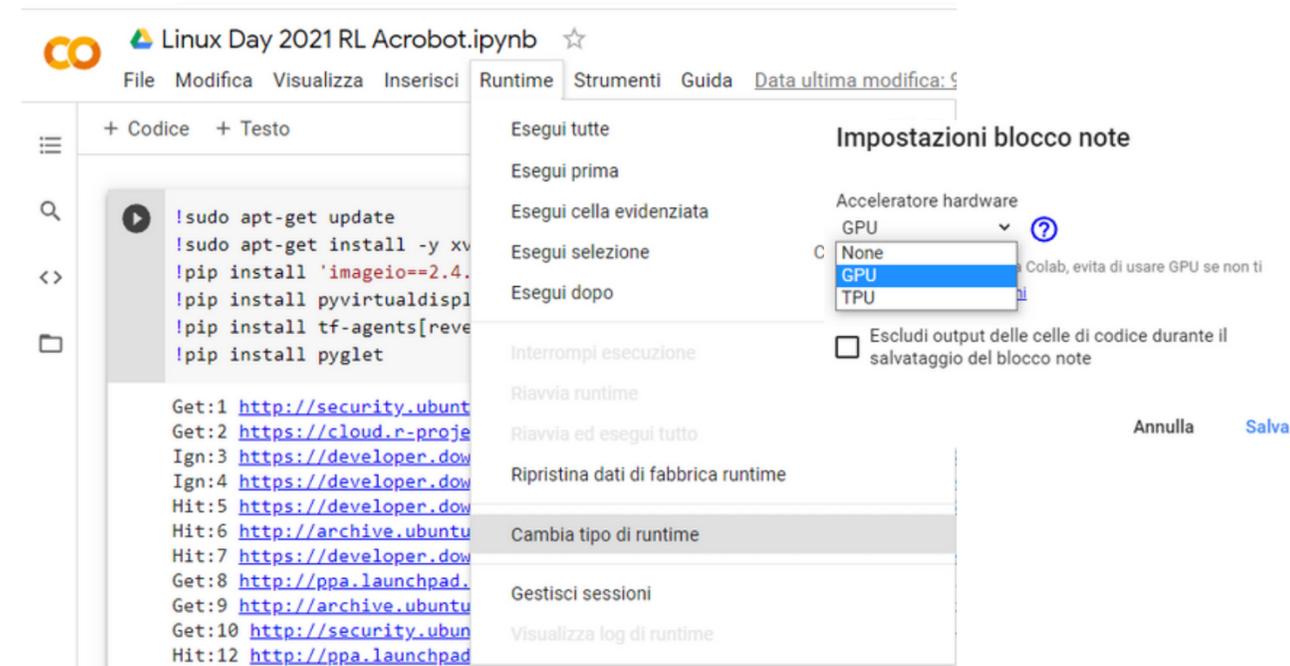
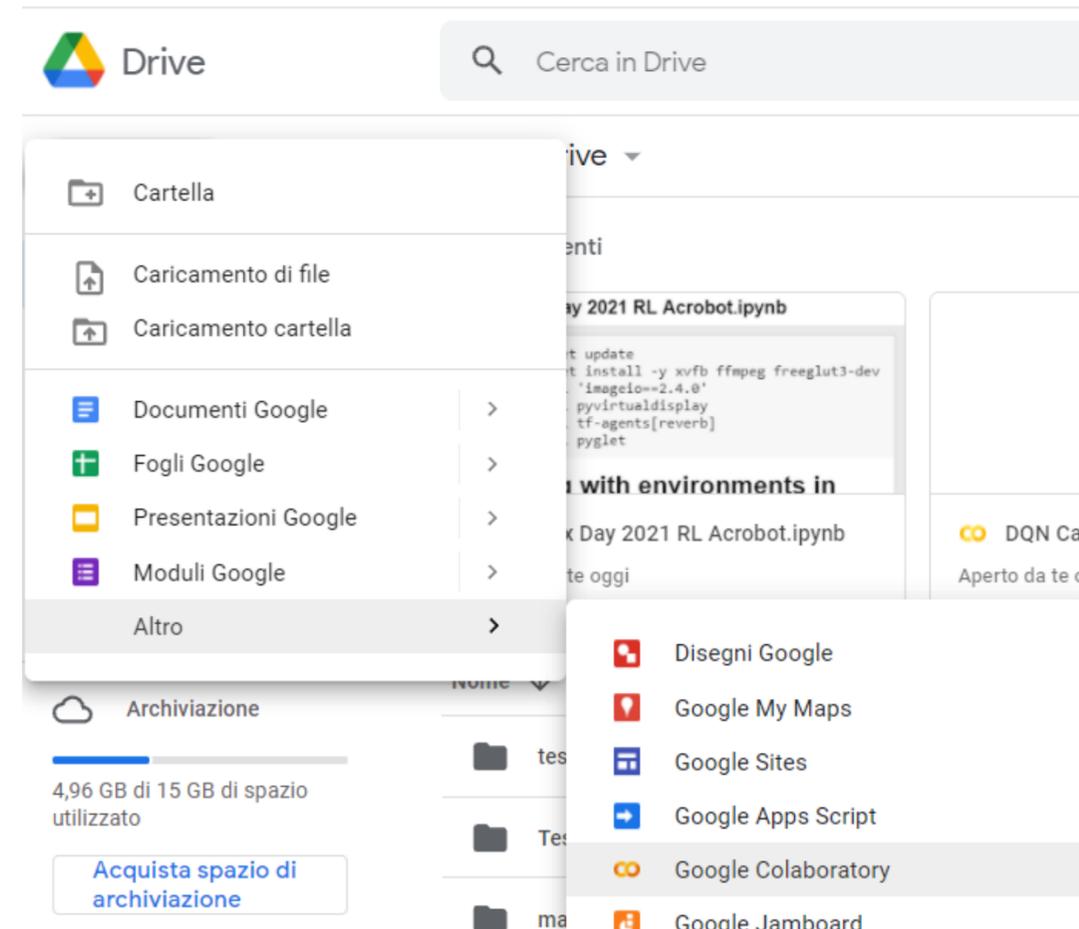
DQL VS Q-LEARNING



COLABORATORY E GPU

"COLAB" PERMETTE DI SCRIVERE ED ESEGUIRE CODICE PYTHON DIRETTAMENTE DAL BROWSER CON I SEGUENTI VANTAGGI:

- NESSUNA CONFIGURAZIONE NECESSARIA
- ACCESSO GRATUITO ALLE GPU



COLAB METTE A DISPOSIZIONE QUATTRO DIFFERENTI SCHEDE:

- TESLA K80,
- TESLA T4S,
- TESLA P4S,
- TESLA P100S.



01 [HTTPS://DEEPMIND.COM/RESEARCH/CASE-STUDIES/ALPHAGO-THE-STORY-SO-FAR](https://deepmind.com/research/case-studies/alphago-the-story-so-far)

02 [HTTPS://GYM.OPENAI.COM/ENVS/ACROBOT-V1/](https://gym.openai.com/envs/acrobot-v1/)

03 [HTTPS://DEEPMIND.COM/RESEARCH/PUBLICATIONS/2019/PLAYING-ATARI-DEEP-REINFORCEMENT-LEARNING](https://deepmind.com/research/publications/2019/playing-atari-deep-reinforcement-learning)

04 [HTTPS://WWW.TENSORFLOW.ORG/](https://www.tensorflow.org/)

05 [HTTPS://COLAB.RESEARCH.GOOGLE.COM/](https://colab.research.google.com/)

06 [HTTPS://GITHUB.COM/ORNELLAFANAIS/LINUXDAY2021.GIT](https://github.com/OrnellaFanaIS/linuxday2021.git)

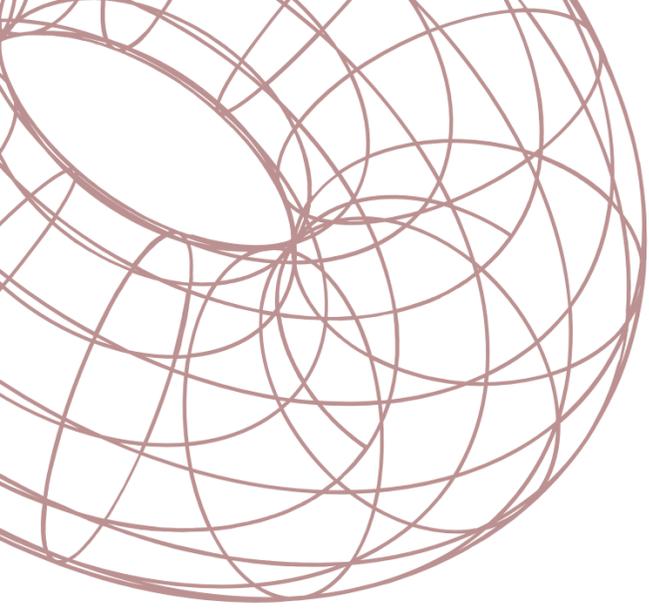
REFERENCES



DATI A CHI?

WWW.LINUXDAY.IT
23 OTTOBRE 2021





GRAZIE PER L'ATTENZIONE !

"CHE SIANO ESPERIENZE SIMULATE O SOGNI, LE INFORMAZIONI SONO AL TEMPO STESSO REALTÀ E FANTASIA. E, IN OGNI CASO, TUTTI I DATI CHE UNA PERSONA ACCUMULA DURANTE IL CORSO DELLA PROPRIA ESISTENZA NON SONO CHE UNA GOCCIA NEL MARE."



BATOU, GHOST IN THE SHELL

LinkedIn: www.linkedin.com/in/ornella-fanais

