**Report Title: Music Generation**

Members: Yerzhanuly Azamat, Yerman Bekzat, Amanzholov Sanzhar

Group: IT-2203

Teacher: Yeleu Sultanmurat

**Introduction**

*Problem*

The creation of music has always been a deeply human activity, blending artistry, emotion, and technical skill. However, with advances in artificial intelligence, there's a growing interest in exploring how AI can contribute to the field of music composition. The challenge lies in developing an AI system that can generate melodious and harmonically pleasing music autonomously, mimicking the creativity and intuition of a human composer.

**Literature Review**

Several approaches have been explored in the realm of AI-driven music generation. One of the earliest techniques involves rule-based systems, where music composition rules are explicitly programmed into the AI. However, these systems often lack the ability to produce truly creative outputs. With the advent of deep learning, neural networks, especially Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, have become popular for their ability to learn and generate sequences, including music. LSTM, a type of RNN, is particularly suited for this task due to its ability to remember and utilize past information, making it ideal for sequential data like music.

RNNs for Music Generation: Magenta by Google is an open-source project exploring the role of machine learning in the process of creating art and music. They have developed several models that use RNNs for music generation.

LSTM-based Models: The LSTM model by Eck and Schmidhuber (2002) is one of the pioneering works in using neural networks for music composition, demonstrating the potential of LSTMs in generating melodies that can follow a musical structure.

*Current Work*

Our project focuses on developing a melody generation AI that leverages the capabilities of LSTM structures within RNNs to create new musical pieces. We aim to advance the state-of-the-art by refining the model's ability to generate melodies that are not only novel but also musically coherent and pleasing to the ear. This involves preprocessing musical data, training our LSTM model on a diverse dataset, and fine-tuning the generation process to produce high-quality music outputs.

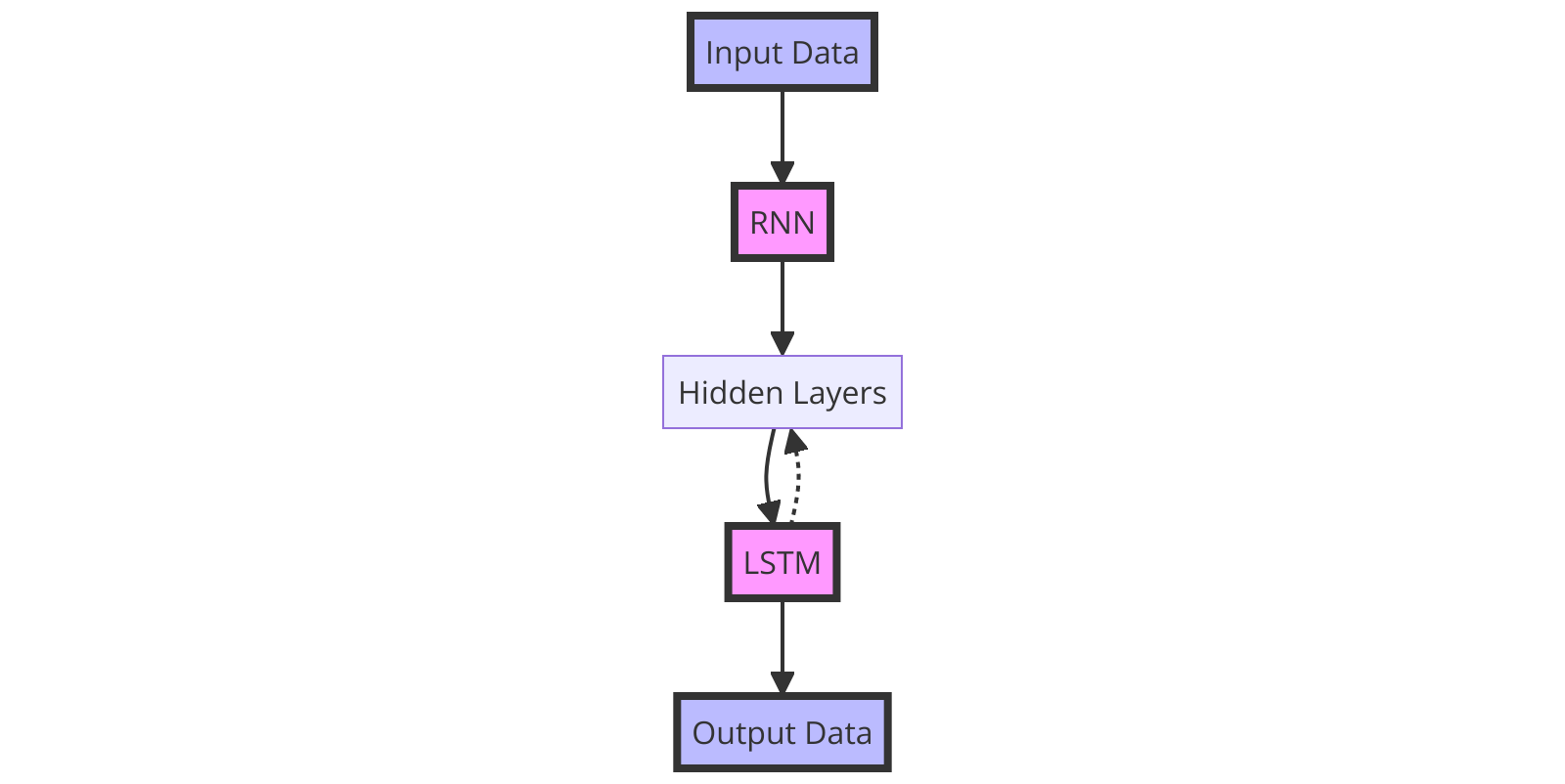
**Data and Methods**

*Information about the Data*

We utilized the Kern dataset, which consists of classical music pieces encoded in the kern notation format. This dataset was chosen for its rich collection of musical compositions, providing a wide variety of melodies and harmonic structures. Our preprocessing steps included filtering songs based on acceptable durations (e.g., quarter notes, half notes), transposing all pieces to a uniform key (C major/A minor), and encoding the musical notes and rests into a time-series format suitable for training our LSTM model.

*Description of the ML/DL Models*

Our model architecture is based on LSTM networks, known for their effectiveness in handling sequence prediction problems. LSTMs are a special kind of RNN (Image 1.1), capable of learning long-term dependencies. Unlike standard feedforward neural networks, LSTMs have feedback connections that make them powerful for learning from sequences of data. Our model includes LSTM layers followed by dropout layers to prevent overfitting, and a dense layer with a softmax activation function to predict the next note in a sequence. The model was compiled using the Adam optimizer and sparse categorical cross-entropy as the loss function, suitable for classification tasks with multiple classes.



*Image 1.1*

Image 1.1 illustrates the architecture of Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs), focusing on how they process input data through their structure. Here's a breakdown of the components and their interactions:

Input Data: This represents the initial data fed into the model, which could be a sequence of values or features relevant to the task at hand, such as time series data or text for language processing.

RNN: The Recurrent Neural Network layer receives the input data. RNNs are designed to handle sequences of data by maintaining a form of memory based on previous inputs. This allows them to exhibit temporal dynamic behavior. Unlike traditional neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition.

Hidden Layers: These layers can be thought of as the processing units of the neural network. For RNNs, hidden layers are particularly important because they store the state of the network. In this diagram, the hidden layers connect the RNN and LSTM components, indicating that the output of the RNN feeds into the LSTM and the LSTM can influence the hidden state.

LSTM: Long Short-Term Memory networks are a special kind of RNN, capable of learning long-term dependencies. They were introduced to avoid the long-term dependency problem in standard RNNs. The LSTM layer receives processed information from the hidden layers and can send information back to influence the hidden state, showcasing its ability to both influence and be influenced by the network's memory. This feedback loop allows LSTMs to make decisions based on both recent input and relevant information from earlier in the sequence.

Output Data: This is the final output produced by the model after processing the input data through the RNN and LSTM layers. The output could be a prediction, classification, or any other form of result depending on the task.

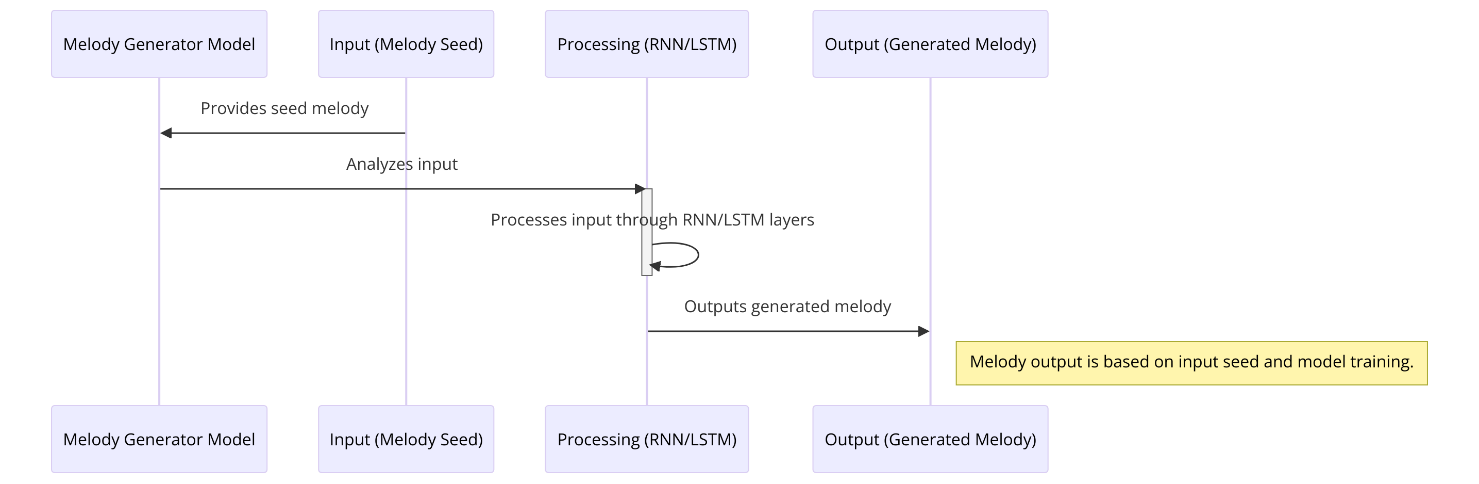
****  
Image 1.2

Diagram in image 1.2 represents a sequence diagram that details the process flow of a melody generation model using Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks. Here's the breakdown of the process as shown in the diagram:

* Melody Generator Model: This is likely the system or application that is designed to generate melodies.
* Input (Melody Seed): This represents the initial input to the model. It is a 'seed' melody, which means a starting set of notes or a musical idea from which the model will generate a full melody.
* Processing (RNN/LSTM): This is the core of the model where the actual processing happens. The RNN/LSTM refers to the type of neural network used, which is adept at processing sequences and time-series data. RNNs are known for their ability to remember previous inputs due to their internal memory. LSTM is a special kind of RNN that can learn long-term dependencies.
* Output (Generated Melody): This is the final output from the model, which is the generated melody based on the input seed.

The arrows indicate the flow of data and the sequence of actions:

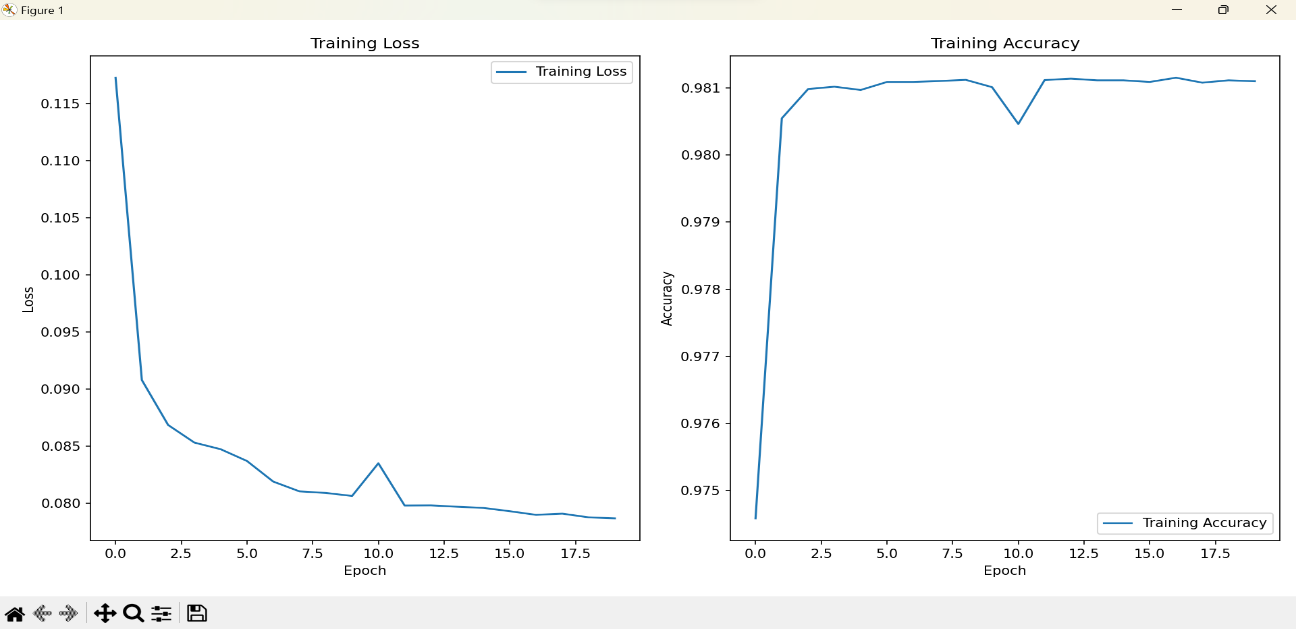
* "Provides seed melody" arrow indicates that the input melody seed is provided to the melody generator model.
* "Analyzes input" arrow shows that the model analyzes this input.
* "Processes input through RNN/LSTM layers" indicates a loop where the model, using its RNN/LSTM architecture, processes the input data repeatedly. This is typically where the model would learn the patterns and structures within the seed melody to generate new sequences of notes that form the output melody.
* "Outputs generated melody" arrow shows the direction of the flow from the processing stage to the output stage, resulting in the generated melody.

Finally, there's a note to the right of the "Output (Generated Melody)" indicating that the output is based on the input seed and the model training. This implies that the quality and structure of the generated melody will depend significantly on both the provided seed melody and how well the model has been trained on music data.

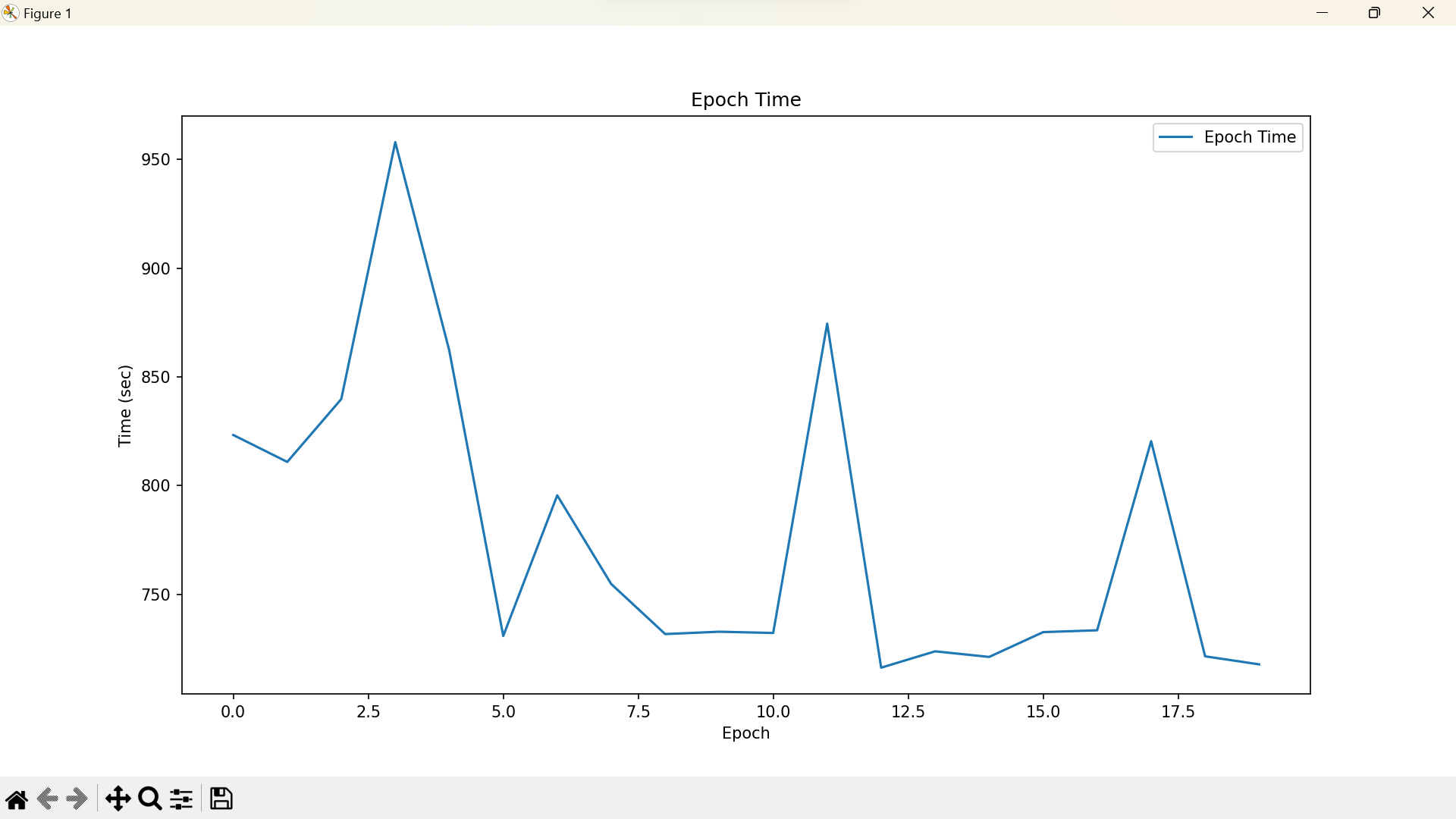
**Results**

Our LSTM model was trained on the preprocessed dataset, resulting in a model capable of generating new melodies based on input seed sequences. The results were evaluated based on the model's ability to produce musically coherent sequences that resemble the style of the training data.

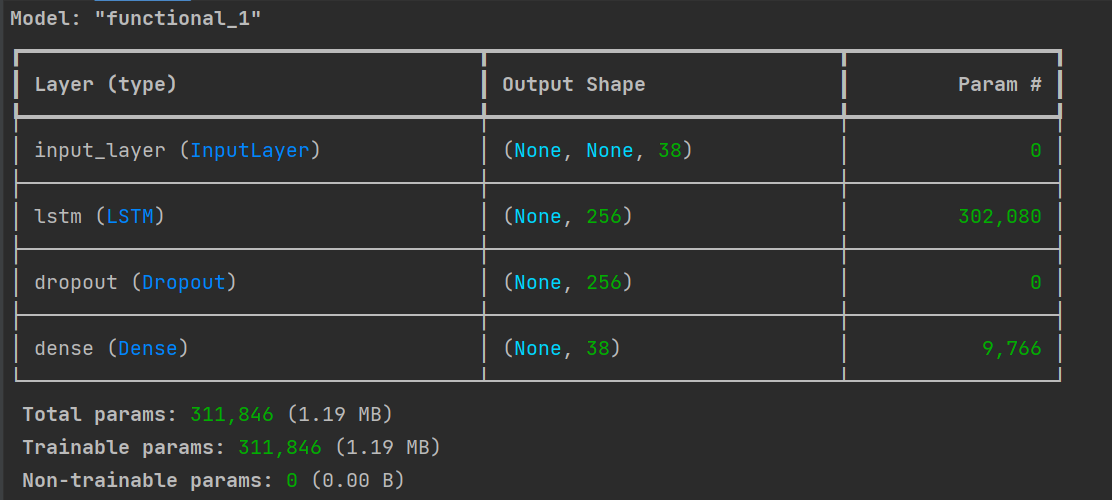
Generated Melodies: The model successfully generated melodies that were both unique and musically pleasing. Examples of generated melodies are presented in MIDI format for auditory evaluation.

Model Accuracy and Loss: Throughout the training process, the model demonstrated improvement in accuracy and a decrease in loss, indicating effective learning from the dataset. 

*image 2.1*



*image 2.2*



*Image 2.3*

**Discussion**

*Critical Review of Results*

The generated melodies, while musically coherent, sometimes lacked the complexity and variation found in human compositions. This suggests that while the model has learned basic musical structures, there's room for improvement in capturing the nuances and emotional depth of music.

*Next Steps*

To enhance the model's performance, future work could explore:

Increasing the diversity and size of the training dataset to encompass a broader range of musical styles.

Experimenting with more complex LSTM architectures or incorporating other types of neural networks, such as Transformer models, which have shown promise in sequence generation tasks.

Implementing techniques to control the creativity and variability of the generated music, such as varying the temperature parameter in the sampling process.

*Conclusion*

This report outlines the initial steps towards creating an AI capable of generating melodious music, highlighting the potential of LSTM networks in the field of creative AI. Further research and refinement are necessary to fully realize the capabilities of AI in music composition, pushing the boundaries of creativity and machine intelligence.

**Sources**

* Google Magenta Project: https://magenta.tensorflow.org/
* Eck, D., & Schmidhuber, J. (2002). A First Look at Music Composition using LSTM Recurrent Neural Networks. Istituto Dalle Molle Di Studi Sull'Intelligenza Artificiale.