Lesson Plan

Lesson 1.1: Neural Networks
Lesson 1.2: Image Recognition Training
Lesson 2.1: Programming Fundamentals
Lesson 2.2: Conditional Programming
Lesson 2.3: Loops
Lesson 2.4: Functional Programming
Lesson 3.1: Basic and Advance Training
Lesson 3.2: AI Interfacing
Lesson 3.3: Autonomous Systems
Lesson 4 : Debugging your code

Lesson 3.2 AI Inferencing & Image recognition

Warm-up	(10 mins)
Main Activity	(20 mins)
Wrap up	(10 mins)
Extras	(10 mins)

Prerequisites: In our previous chapter we discussed training an AI model and how we'll be training a neural networks along with some important programming concepts. If you haven't already, go back to page 18.

Lesson Overview: Students will understand how to test an AI model. We'll play another game level 2.1 & 2.2 and thus teach AI Inferencing.

Lesson objective:

1. To make students understand artificial Intelligence concepts like testing a trained model or how to use one.
2. Also to make students familiar with image recognition and how AI is used to achieve it.

Requirements:

1. Images of famous personalities used in chapter 1.2 of this book.
2. CoderMindz game board
3. Decks of Coder Mindz Code cards

Getting started:

1. We'll be playing section 2.1 & .2.2 - Image Recognition & Adaptive Learning from the rule book. The aim should be to teach students more about AI model inferencing.
2. Remind students about how they learned about training a model in chapter 3.1 from this book. Ask few questions to refresh their concepts.
3. After the rule book is explained, arrange the cards and the board.

AI inferencing:

Testing your trained model is as important as training one. Model inferencing is the process of getting predictions from a trained model. Depending on the training data and other hyperparameters, an ML model could be overtrained or undertrained.

Overfitting:

Overfitting happens when our model generalizes well on the training dataset during the inferencing, but performs poorly on the new dataset.

Underfitting:

Underfitting on the other hand is the situation when the model predicts poorly on both the training dataset as well as new dataset.

How to detect overtrained and undertrained models?

We use validation metrics like loss and accuracy to determine if the model is overfitting or underfitting. When the model is about to become overtrained, these validation parameters stop improving further with respect to each epoch and gradually start decreasing. This is how we detect if the model is overfitting.

Undertrained models perform poorly on both the training and test datasets. While, overtrained models perform exceptionally good on the training dataset, but perform poorly on test dataset.

Image recognition:

Like we studied how to train an image classifier model previously in chapter 1.2, computers use these trained models to classify new images. Based on the quality and quantity of the data fed to the network, a neural network might perform poorly or exceptionally well depending upon its hyperparameters. Hyperparameters are a set of parameters which rule the training process of a model. They describe how a model is trained and how a network is structured.

Board Activity:

1. Following the rules we mentioned in the previous chapters, the first student to reach the end of the board game wins.
2. For image recognition, the player has to collect each piece of image on the board before reaching the end. The game cannot end until all the pieces are acquired.

3. Each player reviews their code cards. The sequence of our programming code. The player who reaches the destination first wins.

Extended Learning:

- Undertrained/Overtrained model:

Ask the students to pay attention and show them three images of some famous personalities like Steve Jobs, Warren Buffet and Jack Ma or any other famous person which you think will be difficult for your audience to recognize too easily.
You can carry three photos of your friends which will be completely new to the students.

Show them these images one by one and call out the names of each person in the photo. Ask the students to remember the photos and their respective names. Repeat this until students agree that they've remembered the images shown to them.

After you've done introducing the images, without announcing the name in the picture, raise one image and ask your class to recognize the person in the image.

Most of them will answer correctly. If they fail to recognize easily, remind them again about the person in the image and check if they recognize it later. This is a phenomenon in machine learning is called an undertrained model where the model doesn't recognize the inputs given to it accurately.

- Image Recognition:

Display the three images you used in chapter 1.2 of this book without announcing their names to the students. Raise one image and ask your class to recognize the person in the image.

Most of them will answer correctly.

Make them understand how their brains stored information about each photo shown to them a week or few days ago. How it stores the features of the image like the skin color, nose, ears, eyes and other facial features to recognize each individual uniquely. This is the same way AI extracts features from the images during the training and stores it.

