Model Creation

AI Model Development

Time: 60 mins

Introduction

In this class, the student/s will learn to separate the training and testing data, create a model and plot the loss values on the graph..

New Commands Introduced

● train_test_split()	Splits the data into array for training and testing.
● Sequential()	Initializes the model for training..
● Conv2D()	Add the convolution layer
● flatten()	Converts the output array from Feature Learning Layers to 1 D array.
● fit()	Function used to train a model
● Droupout()	Removes extra nodes from the CNN.
● compile()	Compiles the model
● summary()	Provides the summary of the model

Vocabulary

- **Compiling** is a method of assembling a model using the provided data..
- **Matplotlib** is a library which helps to plot the points on a graph

Learning Objectives

Student/s should be able to:

- ***Recall*** how to process the data for the model
- ***Demonstrate*** how the create the model.
- ***Explain*** the steps required to create the model.

.

Activities

1. **Class Narrative:** *(2 mins)*
    - Brief the student/s that you can split the data to train and test the model, create the model and plots it's training and validation loss values on a graph.

2. **Concept Introduction Activity:** *(5 mins)*
    - Let the student/s play the explore-activity to see how teachable machine works.
    - Using the slides, explain that the student/s will learn:
        - to create training and testing data.
        - to add epochs
        - to accuracy measures of the model

3. **Activity 1: Create Training and Testing Data** *(14 mins)*

    Student Activity: *(7 mins)*

    - Guide the student/s to split the images and ages data for training and testing and print their lengths to check the number of images used for training.

4. **Activity 2: Add Epochs** *(12 mins)*

    Teacher Activity: *(6 mins)* .

    - Demonstrate how to create the convolution layer
    - Explain how to reduce the dimensions of the images after the convolution layer.
    - Explain how to convert the output array returned from the Feature Learning Layers to 1D using array Flatten(), and remove extra node in the CNN.

    Student Activity: *(6 mins)*

    - Guide the student/s to create the convolution layer, reduce the dimensions of the images after the convolution layer and convert the output array returned from the Feature Learning Layers to 1D using array Flatten(), and remove extra node in the CNN.

5. **Activity 3: Loss Measures of the Model** *(12 mins)*

    Teacher Activity: *(6 mins)*

    - Ask the students to observe how the prediction percentages change depending on the images.
    - Demonstrate how to plot the training loss at each epoch using history.

    Student Activity: *(6 mins)*

    - Guide the student/s to plot the validation loss at each epoch using history.

6. **Introduce the Post class project:** *(2 min)*
    - Use the concepts learned in the class to calculate the loss of the model given.

7. **Test and Summarize the class learnings:** *(5 mins)*
    - Check for understanding through quizzes and summarize learning after respective missions.

- Summarize the overall class learning towards the end of the class.

8. **Additional activities:**
   - Encourage the student/s to change the activation function to ‘tanh’.
   - Encourage the student/s to exchange the x-axis and y-axis.

9. **State the Next Class Objective:** *(1 min)*
   - In the next class, student/s will learn to to use the model to predict the age of detected faces in real-time.

U.S. Standards:

CSTA: 2-AP-11, 2-AP-12, 2-AP-13, 2-AP-14, 2-AP-19

Links Table		
Activity	**Activity Name**	**Link**
Class Presentation	Model Creation	https://s3-whjr-curriculum-uploads.whjr.online/32610ddd-2146-4440-a0f2-52d9bf3fad74.html
Explore Activity	Model Creation	https://github.com/Tynker-Computer-Vision/TNK-M10-PRO-C75-SAS-BP
Student Activity 1	Create Training and Testing Data	https://github.com/Tynker-Computer-Vision/TNK-M10-PRO-C75-SAS-BP
Teacher Reference: Student Activity 1 Solution	Create Training and Testing Data: Solution	https://github.com/Tynker-Computer-Vision/TNK-M10-PRO-C75-SAS
Teacher Activity 2	Add Epochs	https://github.com/Tynker-Computer-Vision/TNK-M10-PRO-C75-TAS-BP
Teacher Reference: Teacher Activity 2 Solution	Add Epochs: Solution	https://github.com/Tynker-Computer-Vision/TNK-M10-PRO-C75-TAS
Student Activity 2	Add Epochs	https://github.com/Tynker-Computer-Vision/TNK-M10-PRO-C75-SAS-BP
Teacher Reference: Student Activity 2 Solution	Add Epochs: Solution	https://github.com/Tynker-Computer-Vision/TNK-M10-PRO-C75-SAS
Teacher Activity 3	Validation of the Model	https://github.com/Tynker-Computer-Vision/TNK-M10-PRO-C75-TAS-BP
Teacher Reference: Teacher Activity 3 Solution	Validation of the Model: Solution	https://github.com/Tynker-Computer-Vision/TNK-M10-PRO-C75-TAS
Student Activity 3	Validation of the Model	https://github.com/Tynker-Computer-Vision/TNK-M10-PRO-C75-SAS-BP
Teacher Reference: Student Activity 3 Solution	Validation of the Model: Solution	https://github.com/Tynker-Computer-Vision/TNK-M10-PRO-C75-SAS
Student's Additional Activity 1	Change the Activation Function	https://github.com/Tynker-Computer-Vi

		sion/TNK-M10-PRO-C75-SAS-BP
Teacher Reference: Student's Additional Activity 1 Solution	Change the Activation Function: Solution	https://github.com/Tynker-Computer-Vision/TNK-M10-PRO-C75-SAS
Student's Additional Activity 2	Exchange the Axes	https://github.com/Tynker-Computer-Vision/TNK-M10-PRO-C75-SAS-BP
Teacher Reference: Student's Additional Activity 2 Solution	Exchange the Axes: Solution	https://github.com/Tynker-Computer-Vision/TNK-M10-PRO-C75-SAS
Post Class Project	Loss Measure of the Model	https://github.com/Tynker-Computer-Vision/TNK-M10-PRO-C75-PCP-BP
Teacher Reference: Post Class Project Solution	Loss Measure of the Model: Solution	https://github.com/Tynker-Computer-Vision/TNK-M10-PRO-C75-PCP