This activity assumes that you have successfully completed all previous activities. It also requires some focus. Learning curves are a key to debug and diagnose a model's performance. The goal in this activity is to plot learning curves and to interpret various learning curves. For a dataset of your choice, the first step is to shuffle the dataset. The next step is to split the dataset into the four arrays: XTRAIN, YTRAIN, XVALID, and YVALID. The next step is to train a neural network model using model.fit(). However, this time, XVALID and YVALID will also be passed as arguments to the model.fit() method. This is so when we call the method, it can evaluate the model on the validation set at the end of each epoch (see code block below). It is crucial to understand that the model.fit() method does NOT use the validation dataset to perform the learning, it is only to evaluate the model after each epoch. When calling the model.fit() method we can also save its output in a variable, usually named history. This variable can be used to plot learning curves (see code block below). The task in this activity is to plot many learning curves in various scenarios. In particular, it is of interest to observe and analyze how the learning plots look like in various settings. The following article discusses learning curves in more detail.

- **Reference Video:**
- Some insights on learning curves
- (Links to an external site.)

-
- **Articles:** Learning curves for diagnosing machine learning model performance

------ Steps ---------

 # Do the training (specify the validation set as well)

```
history = model.fit(XTRAIN, YTRAIN, validation_data = (XVALID, YVALID), verbose = 1)
```

 # Check what's in the history

```
print(history.params)
```

 # Plot the learning curves (loss/accuracy/MAE)

```
plt.plot(history.history['accuracy']) # replace with accuracy/MAE
```

```
plt.plot(history.history['val_accuracy']) # replace with val_accuracy, etc.
plt.ylabel('Accuracy')
plt.xlabel('epoch')
plt.legend(['training data', 'validation data'], loc='lower right')
plt.show()
```

Produce learning curves that represent the following cases:

1. the validation set is too small relative to the training set - for example, only 1% or 2% of the total rows of data.
2. the training set is too small compared to the validation sat - for example, only 1% or 2% of the total rows of data.
3. a good learning curve (practically good)
4. an overfitting model
5. a model that shows that further training is required
6. an underfit model that does not have sufficient capacity (also may imply that the data itself is difficult)