University of Echahid Hama Lakhdar, El-oued
institute of exact sciences
Department of computing
$1^{er} Master$: Artificial Intelligence and Data science
Semester : 1.2024
Mohammed Anouar Naoui

— Deep Learning 5 : Image Classification with the MNIST Dataset—

---

1 MNIST dataset

The MNIST dataset is a widely used benchmark dataset in the field of machine learning and computer vision. It consists of a collection of 28x28 pixel grayscale images of handwritten digits (0 through 9), along with their corresponding labels indicating the digit that each image represents.

The MNIST dataset is commonly used for training and testing various machine learning algorithms, particularly for tasks such as image classification and digit recognition. It has a training set of 60,000 examples and a test set of 10,000 examples.

2 Load Data and split into train,valid

```
from tensorflow.keras.datasets import mnist
# the data, split between train and validation sets
(X_train, y_train), (x_test, y_test) = mnist.load_data()
```

3 Exploring the MNIST Data

its images are represented as a collection of unsigned 8-bit integer values between 0 and 255, the values corresponding with a pixel's grayscale value where 0 is black, 255 is white, and all other values are in between.

```
x_train.dtype
x_train.min()
x_train.max()
```

— How mach of images.

— What is its dimension of pixels representation.

```
x_train.shape
```

— Return the values of Matrix pixel of index image 9.

— Dress this image.

```
x_train[9]
import matplotlib.pyplot as plt
image = x_train[9]
plt.imshow(image, cmap='gray'
```

4 Preparing the Data for Training

1. Flatten the image data, to simplify the image input into the model
2. Normalize the image data, to make the image input values easier to work with for the model
3. Categorize the labels, to make the label values easier to work with for the model.

4.1 Flatten the image data

It is the process of transforming the shape from (c_1, c_2, C_3) into $(c_1, c_2 \times c_3)$ with reshape function.

```
x_train =x_train.reshape(c1, c2×c3)
x_valid = x_valid.reshape(c1, c2×c3)
```

4.2 Normalizing

Normalizing :Converting integer values to floating point values between 0 and 1.

, keeping inputs within this range often leads to more stable training, better numerical properties, and compatibility with common activation functions and optimization techniques.

```
x_train = x_train / 255
x_valid = x_valid / 255
x_train.dtype
x_train.min()
x_train.max()
```

5 Categorically Encoding the Labels

The MNIST dataset involves classifying handwritten digits into one of 10 classes (digits 0 through 9). This is a typical example of a multiclass classification problem where each sample can belong to one of multiple classes. to_categorical encodes the target variable into a binary matrix representation suitable for multiclass classification tasks.

```
import tensorflow.keras as keras
num_categories = 10

y_train = keras.utils.to_categorical(y_train, num_categories)
y_test = keras.utils.to_categorical(y_test, num_categories)
```

6 Creating the model

- Input layer :512 units,activation function=Relu. input_shap=.
- 1 hidden layer 512 neurons, activation function = Relu.
- output layer neurons=. activation function=.

7 Compiling and train model

```
model.compile(loss='.', metrics=['accuracy'])
history = model.fit(
    x_train, y_train, epochs=5, verbose=1, validation_data=(x_valid, y_valid)
)
```

8 Plot accuracy/Epochs diagrame

9 Predict the image of index 9

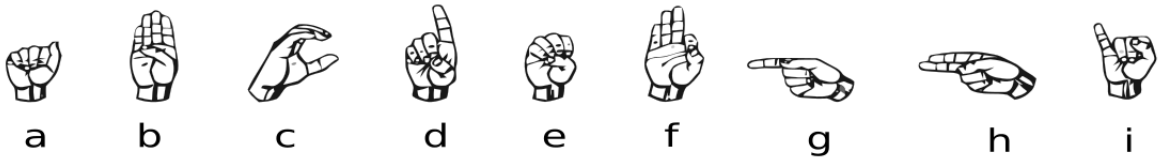
10 Clear the Memory

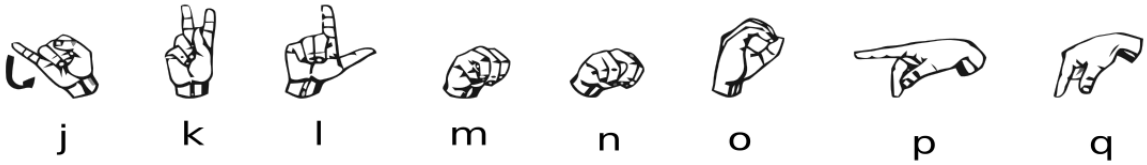
```
import IPython
app = IPython.Application.instance()
app.kernel.do_shutdown(True)
```

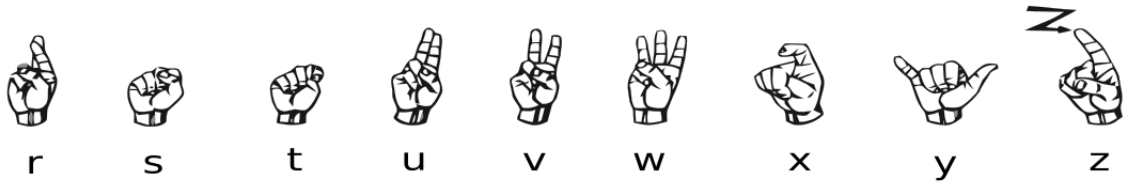
11 American Sign Language Dataset

The American Sign Language Dataset contains 26 letters. Two of those letters (j and z) require movement, so they are not included in the training dataset.

Write deep learning model for asl classification.


a
b
c
d
e
f
g
h
i


j
k
l
m
n
o
p
q


r
s
t
u
v
w
x
y
z