Final Project

Due 5/04/2021

The objective of this project is for you to become familiar with the operation of Convolutional Neural Networks (CNN). The recommended platform is jupyter lab.

To refresh your understanding of neural networks, refer to this link:

https://towardsdatascience.com/understanding-neural-networks-19020b758230

For backpropagation explanation, refer to this link:

https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd

Use the notebook at:
https://cse.sc.edu/~yiannisr/590/2021/Final-Project.ipynb

In it you will find a fully functional CNN. The implemented CNN loads a dataset, provides a template for Data Augmentation, defines a network, trains the CNN, and plots loss and accuracy together with the confusion matrix. In particular you will find the following parts:

Loading data:
(x_train, y_train), (x_test, y_test) = mnist.load_data()

For data augmentation:
Uncomment:

```
'''datagen = ImageDataGenerator(
        featurewise_center=False,  # set input mean to 0 over the dataset
        samplewise_center=False,  # set each sample mean to 0
        featurewise_std_normalization=True,  # divide inputs by std of the dataset
        samplewise_std_normalization=False,  # divide each input by its std
        zca_whitening=False,  # apply ZCA whitening
        rotation_range=10,  # randomly rotate images in the range (degrees, 0 to 180)
        zoom_range = 0.1, # Randomly zoom image
        width_shift_range=0.1,  # randomly shift images horizontally (fraction of total width)
        height_shift_range=0.1,  # randomly shift images vertically (fraction of total height)
        horizontal_flip=False,  # randomly flip images
        vertical_flip=False) ''' # randomly flip images
```

Comment out:

```
datagen = ImageDataGenerator()
```

The proposed network:

```
model=Sequential()

model.add(Conv2D(filters=64, kernel_size = (3,3), activation="relu",
input_shape=(28,28,1)))
```

```
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Conv2D(filters=128, kernel_size = (3,3), activation="relu"))

model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Conv2D(filters=256, kernel_size = (3,3), activation="relu"))

model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Flatten())
model.add(BatchNormalization())
model.add(Dense(512,activation="relu"))

model.add(Dense(10,activation="softmax"))
```

You should explore the following options:

1. Data Augmentation: Using the commented out **ImageDataGenerator** method augment the training dataset and discuss the change in performance and the augmentation selected (why).
2. Training Iterations: Change the number of epochs during training.
3. Improve the provided CNN in:
    a. Depth: by adding additional layers
    b. Width: by increasing the number of filters per layer.

In your report include the plots of accuracy and loss for the different modifications and discuss their impact. In addition, report on the speed of each epoch for each modification in order to understand the impact in performance. Use model.summary() to see the number of total and trained parameters.

3. (Grad Credit)

Uncomment:
#(x_train, y_train), (x_test, y_test) = fashion_mnist.load_data()
Comment out:
(x_train, y_train), (x_test, y_test) = mnist.load_data()
To run your CNN on a different dataset.

What to Submit

Please together with your code in the form of a jupyter notebook provide a report discussing your implementation together with some illustrative examples (screen shots). Ensure that is well written and formatted. Document your work in images and use captions to describe what each image displays. Use of LaTeX is encouraged.