Deep Learning, MSc in Data Science

Assignment 1: A Simple Classification Network

Handed out: 25 October 2018

Due: 15 November 2018

Description

In this assignment you will build a deep classification network, optimised for the dataset CIFAR-10[1]. You may build on/extend the example provided below, and in the classroom. Your solution must be a simple feedforward network. Any work you may choose to do with CNNs is extra and will not itself earn you extra marks – however it will lead to better accuracy.

In optimising your network, make sure you address the following aspects:

1. Data pre-processing
2. Depth and width of your network
3. Choice of activation and loss functions
4. Regularisation
5. Choice of learning optimisers
6. Hyper-parameters, such as batch size, epochs, learning rate, etc., depending on the chosen configuration.

Deliverable

Write a **2-page report**[2] describing your approach and findings. The report should justify your choices when designing your neural network. It should also include a brief presentation of your final/best classification performance.

You must also submit your source code, commented and structured as needed.

Make sure you write your name both on the report and in the source code.

Bundle and submit your report **in PDF** along with your source code, **in a single zip or gz file**.

1 https://www.cs.toronto.edu/~kriz/cifar.html

2 Indicatively, minimum font size: 11pt Times, minimum page margins: 1.5cm.

Template Code

```
import keras
import numpy as np
import keras
import tensorflow as tf
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout
from keras.optimizers import SGD

batch_size = 128
num_classes = 10
epochs = 40

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train = x_train.reshape(60000, 784)
x_test = x_test.reshape(10000, 784)
x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
x_test /= 255

y_train = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)

model = Sequential()
model.add(Dense(100, activation='relu', input_shape=(784,)))
model.add(Dense(100, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))

model.summary()

model.compile(loss='categorical_crossentropy',
              optimizer=SGD(),
              metrics=['accuracy'])

history = model.fit(x_train, y_train,
                    batch_size=batch_size,
                    epochs=epochs,
```

```
                    verbose=1,
                    validation_data=(x_test, y_test))
score = model.evaluate(x_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```