MA3832-Practical 3: Multi-layer perceptron neural network

Dr. Kelly Trinh

Sofwares

1. Install `Tensorflow` (version 2 or >) and `Keras` on your computer.

Goals

1. Build, train and deploy MLP using Python
2. Fine tuning parameters in MLP

Installing TensorFlow 2

Assuming you installed Jupyter and Scikit-Learn by following the installation instructions in Chapter 2, you can simply use pip to install TensorFlow. If you created an isolated environment using virtualenv, you first need to activate it:

```
$ cd $ML_PATH               # Your ML working directory (e.g., $HOME/ml)
$ source env/bin/activate   # on Linux or MacOSX
$ .\env\Scripts\activate    # on Windows
```

Next, install TensorFlow 2 (if you are not using a virtualenv, you will need administrator rights, or to add the `--user` option):

```
$ python3 -m pip install --upgrade tensorflow
```

For GPU support, you need to install `tensorflow-gpu` instead of `tensorflow`, and there are other libraries to install. See *https://tensorflow.org/install/gpu* for more details.

To test your installation, open a Python shell or a Jupyter notebook, then import TensorFlow and tf.keras, and print their versions:

```
>>> import tensorflow as tf
>>> from tensorflow import keras
>>> tf.__version__
'2.0.0'
```

Workflow of implementing a neural network

- Shape: ensuring the dataset have the right shape for TensorFlow

- Normalisation: normalising the data, i.e. tranform [0-255] pixels values to [0-1] value
- One-Hot Encode Class Labels: convert category labels using one-hot encoding
- Building the model: build a sequential model in Keras
- Training: train the model
- Test: test the model and make predictions

Data Description

In this section, we work with the Fashion MNIST dataset, which contains 60000 small suare 28×28 pixel grayscale images of 10 types of clothing 0: T-shirt/top
1: Trouser
2: Pullover
3: Dress
4: Coat
5: Sandal
6: Shirt
7: Sneaker
8: Bag
9: Ankle boot

Our task is to construct a MLP to classify the types of clothing.

1. Call `x_train, y_train, x_val,y_val` from `fashion_mnist.load_data()`
2. Describe data type of `x_train, y_train`
3. Pre-processing data before implementing MLP using `Keras`
4. Build a neural network with the following structure:
    - Number of Hidden layers = 2: The first hidden layer contains 64 neurons while the second layer includes 32 neurons.
    - Activation function: Both hidden layers will use the Relu activation function
    - Optimizer: Stochastic Gradient Descent
5. Check the initial weights and biases.
6. Plot metrics measures (e.g. loss, accuracy) against number of epochs.Comment on the performance of the model.
7. Use the trained MLP to predict the first 5 instances of the test data.
8. Save and restore the model
9. Tune the number of hidden layers and the number of neurons.