Neural Networks: Spring 2019
Kharazmi University
Homework 1
Due Date: Ordibehesht, 28, 1398

To implement the task in this problem you can utilize Keras library. You can check some examples on Keras repository[1]. In this assignment, you will learn the five key steps in using Keras to create a neural network:

1. How to load data.
2. How to define neural network in Keras.
3. How to compile a Keras model using the efficient numerical backend.
4. How to train a model on data.
5. How to evaluate a model on data.

You will also learn how to visualize data.

Problem 1

In this problem, we are going to use the Pima Indians onset of diabetes dataset.. It describes patient medical record data for Pima Indians and whether they had an onset of diabetes within five years. As such, it is a binary classification problem (onset of diabetes as 1 or not as 0). All of the input variables that describe each patient are numerical. This makes it easy to use directly with neural networks that expect numerical input and output values. Dataset records and description are available in pima directory.

1. Load the data and split it to training and validation sets.
2. Build a multilayer perceptron two hidden layers of size 12 and 8 respectively and output layer of size one. Use relu activation for hidden layers and sigmoid for output.
3. Compile the model with binary cross-entropy loss and adam optimizer.
4. Train the model for 100 epochs. Save the model at the end of each epoch.
5. Draw loss and accuracy at each epoch for both train and validation data.
6. Choose the best model and load it from trained model file. Using this model make prediction for all the training and test data and provide the overall accuracy.

[1] https://github.com/keras-team/keras/tree/master/examples

Problem 2

In this problem you are expected to implement a convolutional neural network and train it on the given data according to the following steps:

1. Data preprocessing:
   1.1. Check the ReadMe.txt file in digits directory. Then, load training and test data from train.csv and test.csv files respectively using pandas package in python.
   1.2. Normalize pixel values to the range [0, 1].
   1.3. Reshape each set to the shape (#samples, width, height, channel).
   1.4. Encode all labels to one-hot.
   1.5. Split training data into training and validation sets.
2. Build CNN architecture using keras:

Layer type	Description
Convolution	filters: 64, kernel size: (3,3), strides: (1,1), padding: true, activation: relu
Max Pooling	kernel size: 2
Convolution	filters: 64, kernel size: (3,3), strides: (1,1), padding: true, activation: relu
Max Pooling	kernel size: 2
Convolution	filters: 32, kernel size: (3,3), strides: (1,1), padding: true, activation: relu
Max Pooling	kernel size: 2
Dense	size: 64, activation: relu
Dropout	probability: 0.25
Dense	size: 32, activation: relu
Dropout	probability: 0.25
Dense	size: 10, activation: softmax

3. Augment image data using ImageDataGenerator class.
4. Set SGD optimizer with learning rate equal to 0.01 and momentum to 0.5.
5. Train the model with original data. At the end of each epoch save the trained model. Plot the loss function and accuracy for training and validation data over all epochs.
6. Train the model with augmented data. At the end of each epoch save the trained model. Plot the loss function and accuracy for training and validation data over all epochs.
7. Feed one test sample into the network and draw
   7.1. Output (64) images of the first convolution layer.
   7.2. Output (64) images of the second convolution layer.
   7.3. Output (64) images of the third convolution layer.
8. Draw the confusion matrix for validation data.

9. Train the model with original data again but use a learning rate scheduler. Compare the training process (speed, convergence …) and accuracy of the both training schemes.

Problem 3

In this problem you are expected to implement a convolutional neural network and train it on the given amazon data for sentiment prediction according to the following steps:

1. Data preprocessing:
    1.1. Check the ReadMe.txt file in amazon directory. Then, load the data from amazon.csv file using pandas package in python. Use cleaned version of text data.
    1.2. Encode all labels to one-hot.
    1.3. Make a dictionary with word as value and its index as key.
    1.4. Save the dictionary into a json file.
    1.5. Load the dictionary from save json and convert words in dataset into their equivalent index.
    1.6. Split training data into training and validation sets.
2. Building CNN architecture using keras:

Layer Type	Specs
Embedding	Length:100
1D convolution	kernel size: 3, activation function: relu, number of kernels: 64
1D convolution	kernel size: 3, activation function: relu, number of kernels: 32
1D convolution	kernel size: 3, activation function: relu, number of kernels: 16
Pooling	size: global, type: maximum
Fully Connected	Size: 12, activation function: relu
Fully Connected	size: 1, activation function: sigmoid

3. Compile the model with SGD optimizer with learning rate equal to 0.01 and momentum to 0.5.
4. Train the model with a learning rate scheduler.
5. Do you see any over-fitting problem? If yes, propose and implement at least two solutions to learn the sentiment prediction task with a higher accuracy.