

Assignment 3 - Recurrent Neural Networks

GSN Team

May 13, 2020

1 Dataset

In this assignment you are going to work with sequences of events. Your data are time series representing random walks of a particle on nodes in a square lattice. One of the four classes assigned to each time series represents a different type of random behavior of the particle. Your task is to build a classifier that can assign a type of behavior based on the time series.

Zip file with data contains four files:

- **train_x.csv** - training data in CSV format. Each row contains one time series delimited with ",", and each element contains a node coordinates separated with "-"
- **test_x.csv** - test data in the same format
- **train_y.csv** - training data class for each row from **train_x.csv** file,
- **test_y.csv** - test data class for each row from **test_x.csv** file.

2 Framework

You should only use PyTorch tensor operations and layers from **torch.nn** module like **Linear**, **Dropout**, **LSTM**, etc. for building your network.

3 Task

3.1 Data analysis - 1 point

Parse data files and load data into an array. Take a look at the dataset, visualise paths of some of the time series. See if you can create a hypothesis about what is the difference between classes.

3.2 Model on raw data - 1 point

Load data into a tensor. Because the lattice is two dimensional your time series will contain 5 steps of two numbers representing x_1 and x_2 coordinates. Create a recurrent architecture of your choice (LSTM, GRU, etc.) which can be trained on data with variable length of each time series. As a test truncate some random time series in train and test data to length 4 and train a network for one epoch. Then train a network on original (not truncated) training data for as many epochs as you like and check accuracy of your classifier on test data.

You should have at least 65% of accuracy on a test set.

3.3 Model with embedding - 1 point

After you trained a model on raw data create a model with embedding layer. At the beginning assign a distinct random number from 0 to 99 to each node in the lattice and change your input data to reflect this change. You should obtain a time series containing 5 integers each.

After that add an embedding layer to your network with embedding of size 2. Change your model to work with it and train your model on new dataset. If you need some inspiration for the architecture you should take a look on some simple models for text classification. In this task you should also change your network to work with batches longer than one (if you didn't do it in the previous model).

Your network should have similar accuracy than the previous network, so you should obtain at least 65% of accuracy on the test data.

3.4 Embedding analysis - 1 point

Now take a look what embedding you have trained. Plot the structure of the dataset in the original space (2D lattice) and in embedding 2D space. Find out and visualise similarities and differences between this two structures. Create a visualisation/animation how embeddings change during the training of a network.

4 Deadline

You should submit your solution by email by 23:59 on 02.06.2020 to your lab teacher with email title "Assignment 3 - Deep neural networks". Your code will be inspected during the lab session following the deadline. Note that even if you are one minute late after the deadline, your solution will not be inspected. We have no mercy whatsoever so you better not count on that.