Homework 5

ECSE 552 - Winter 2021

Due date: 23:59 at 26 March, 2021

Weather Forecasting using Deep Learning

In this homework, you will attempt to create a time-series forecasting model. Specifically, you will be using the weather time-series dataset [1] recorded by the Max Planck Institute for Biogeochemistry from 2009 to 2016. This dataset was prepared by François Chollet and originally has 10-minute intervals. We are only going to do hourly predictions so we removed some of the rows in the dataset attached to this homework.

Given a history of weather attributes from time $t-k$ to $t-1$, you have to predict[2] the following values for time t:

- `p(mbar)`, atmospheric pressure
- `T (degC)`, air temperature
- `rh (%)`, relative humidity
- `wv(m/s)`, wind velocity.

An illustration of the task is shown in Figure 1.

Your input features are the weather attributes of the previous k time steps. You are required to use `p(mbar)`, `T (degC)`, `rh (%)`, and `wv(m/s)` as features. You may include other attributes provided that these are values from the previous time steps.

To reduce search space, you may select a fixed values of $k \in [4, 8]$. You may also preprocess[3] the attributes in any way you want as long as there is no data leakage. For example, you may use the hour in the `Date Time` column since the time of day can affect the temperature. You may also predict other attributes if you think this is useful in terms of learning. However, we will only look the the 4 attributes above for model comparisons.

[1] https://www.bgc-jena.mpg.de/wetter/
[2] multi-task learning
[3] hint: time and angles have different notions of similarity compared to temperature

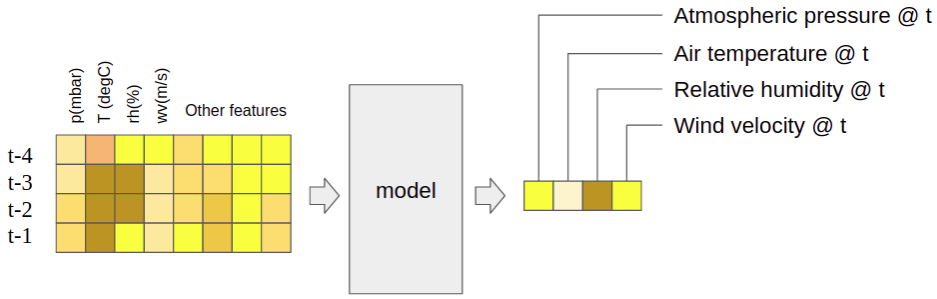

Figure 1: Problem overview with $k = 4$

You have to implement 3 models:

- a baseline model consisting of only dense layers
- a baseline model that uses at least one layer of LSTM
- your own model (may use any neural network, including but not limited to CNNs and GRUs)

For the LSTM model, features must be fed to the network sequentially (i.e. from $t - k$ to $t - 1$).

The first 2 models should have a justifiable "default" loss function (e.g. MSE, Cross Entropy, KL Divergence). Your own model may use custom loss functions. There is no limitation on the number of layers, nodes, and regularizations that you are allowed to use provided that they are justified in your report.

To ensure fairness of comparison, the first 2 models should have the same number of layers as your own model. You do not need to perform an extensive hyperparameter search as long as they are reasonable (e.g. fixed batch size, learning rate, etc.). Note that we do not expect you to create a state-of-the-art model.

Report

The following details are expected in your report:

- Method (40 pts)
  - features
  - preprocessing steps (if applicable)
  - specific model architecture (of the 3 models)
  - loss function/s
  - your model (advantages, special properties, difference from the other two, etc.)
  - hyperparameters and how you selected them
  - training procedure
- Results (30 pts)
  - metrics used for comparison
  - model comparisons in terms of the 4 specified attributes (separately)

  - overall performance comparison

- Additional Analysis for your model
  - Error propagation (i.e. attempting to predict values at time t to $t+k$ when you are only given data from time steps $t-k$ to $t-1$). This means that you will have to use the "predicted values" at time t as input[4] to your model to predict the attributes at time $t+1$. (15 pts)
  - other analysis (be creative) (15 pts)

You need to submit your code as a proof but **only your report (in pdf) will be graded.**

NOTE: Do not include code or references to your code in your report.

[4]If your model only predicts the 4 attributes but uses other features, assume that the other features are given