Project

Normal Autoregressive Neural Network

Financial Engineering

Consider the GEFCom2017 dataset on New England households' consumption. It contains the data published by the overseer of New England bulk electric power system, ISO New England (ISONE). Work with data for the whole New England area. Use 9 calendar years of the GEFCom2017 dataset (as in [1]), from January 2008 up to December 2016 (the link to the dataset is available in the project folder). The year 2011 is used to validate the calibrated model and select the best hyper-parameters, the year 2012 for testing and the years from 2013 to 2016 to verify the robustness and reliability of the results achieved with the selected model.

In the analysis aggregate the power consumption values (in GWh) and average the weather conditions, i.e. consider the daily consumption and the average temperature for every day. In the models consider the natural logarithm of power consumption.

For the analysis:	standardize all variables such that they are all included in the interval [0, 1] in the training set as in [2] p.1, min max normalization.
For presenting results:	you should plot time-series, provide results and calculate errors after you have de-standardized the power consumption.

1. Describe the features of the time-series (see also Tab.1 and Fig.1 in [1]).

2. Fit a GLM model on the natural logarithm of power consumption w.r.t. the calendar effect (use the same independent variable of equation (1) in [1]).

3. Build an Auto-regressive Neural Network that fits the residual of the GLM (the Neural Network should have as input the weather conditions and the calendar effects). The Neural Network structure should be the one of equation (2) in [1]. To fit the Neural Network use the likelihood function for a residual R_t as in p.8 of [1]

$$L(\mu_t, \sigma_t | R_t) = \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp -\frac{(R_t - \mu_t)^2}{2\sigma_t^2} \quad .$$

<u>Hint</u>: As loss function consider $-\frac{10^3}{N}\sum_{t=1}^{N} \log L(\mu_t, \sigma_t | R_t)$, where N is the number of days in the training set

4. Train on the 2008-2010 time-window and validate on the 2011. Select the best model (in term of root mean squared error) w.r.t. the grid of hyper-parameters in table 1.
   Hint: Use an early stopping criterium (e.g. from keras.callbacks import EarlyStopping), in particular fix min_delta and patience parameters in order to obtain accurate results in a "reasonable" time.
   During the estimation of the Neural Network parameters (in the training) use an **l1** regularization on the output of the hidden layer (activity regularizer). Your loss should include an additional term with the **l1** norm of all outputs of the hidden layer (on the training set) multiplied by the regularization parameter λ
$$Loss = Loss_{output} + \lambda \sum_{t=1}^{N} \sum_{i} |y_{i,t}|$$
   $y_{i,t}$ is the i-th output of the hidden layer for the day t.
   <u>Hint</u>: In Keras we suggest to use keras.regularizers.l1 (activity regularizer) on the hidden layer. The **l1** regularizer takes as input the regularization parameter λ.

Hyper-parameter	*Values*
Number of neurons (hidden layer)	3, 4, 5, 6
Activation function	softmax and sigmoid
Initial learning rate (for Keras ADAM)	0.1, 0.01, 0.003, 0.001.
Batch size	50, no batch
Regularization parameter λ	0.001, 0.0001, 0

Table 1: Set of hyper-parameters that should be considered in the validation step. The initial learning rate is a parameter of ADAM (the reported values are the ones of the Keras ADAM solver, in Keras it is a parameter of the fit method), while batch size is the size of the batch.

5. Calibrate the selected model (with the selected hyperparameters) on the 2009-2011 time-window. Plot the selected models' forecast on the test set 2012. Plot also a 95% confidence interval.
   <u>Hint</u>: Each density forecast at time t is a simple Gaussian with mean $T_t + S_t + \mu_t$ and variance σ_t.

6. Plot pinball loss (Fig. 8 of [1]) and the backtested confidence interval for the test set (Fig.8 and 9) (cf. [1] section 4). Test the performance of the model on the years 2013 to 2016 each time calibrating on the three-year time window and testing on the following year (e.g. calibrate on 2009-2011 and test on 2012). Compare with simple benchmarks as GLM and ARX in [1].

7. Discuss the ex-post forecast technique. Advantages and limits of this technique.

8. FACULTATIVE. Sensitivity analysis. Repeat point 4, 5 and 6 after having (considerably) enlarged the hyper-parameters grid. More neurons, more activation functions, more possible initial learning rates, batch size and regularization parameters (possibly even more Neural Network layers). You can also try to insert the stopping criterion parameters in the grid search procedure.

Library: Python. Keras with tensorflow back-end.
In Python use the ADAM Keras optimizer, from keras.callbacks import EarlyStopping.
Whenever possible, fix a seed for the random simulation: this allows to repeat end reproduce your results.
Optional: Matlab.

[1] M. Azzone and R. Baviera (2020), Neural Network Middle-Term Probabilistic Forecasting of Daily Power Consumption, Mimeo.
[2] Patro, S., and Kishore Kumar Sahu (2015). "Normalization: A preprocessing stage." arXiv preprint arXiv:1503.06462.