DIRECT
Data Intensive Research Enabling Clean Technologies

WindChaser: a Deep Learning Framework for Wind Power prediction and Energy Use Recommender

Yiwen Wu[1], You Chen[1], Xiaoxiao Jia[1] and Yize Chen[2]
Department of Materials Science and Engineering[1] and Department of Electrical Engineering[2]
University of Washington, Seattle, WA

Introduction

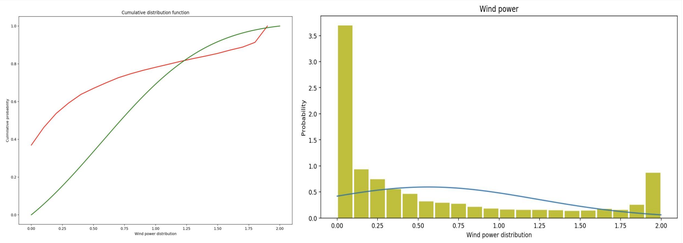
Backgrounds: Renewables are having growing penetration in modern grids
Challenges: a). The stochastic process of renewable generation is hard to forecast using traditional models
b). Renewable prices are fast changing. From a customer method, how to find an algorithm to cut down electricity bill
Proposed Methods: a). Deep Long Short-Term Memory networks for forecasting future wind power generation
b). Q-Learning framework for end-user electrical appliances scheduling.

Data Description

Datasets: *Wind Integration* by National Renewable Energy Laboratory (*NREL*). They recorded the wind power generation data nationwide for past 7 years.
Original datasets include power generation recorded every 5 minutes over the years from wind farms and large-scale solar panels (288 data points per day).

- Select wind farms in Seattle, WA
- One-year of data, 80% as training, 20% as testing

Data Processing and Analysis

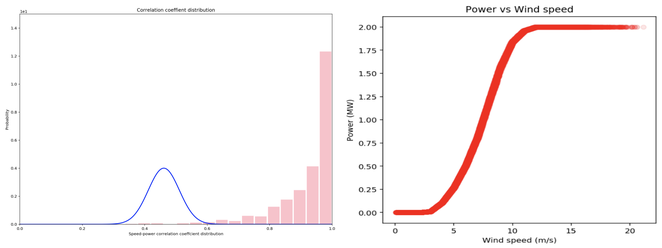


The CDF and PDF wind power in comparison to a fitted normal distribution.

Dataset Inspection

- Wind power is mainly concentrated around 0 MWh(low power), and the fitted cumulative distribution is not matching well.
- By simply calculating the power distribution and visualizing it can not help us understand and further predict the wind power generation accirately.

The T-test for the *wind power* and *wind speed*, p-value is much more than 5%. Solely focusing on the average value in our data is inadequate.



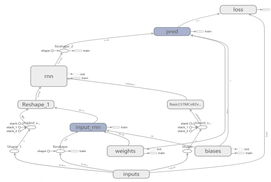
Feature Representation

We choose each 122 data (about 10 hr) to calculate the correlation coefficient between power and speed. We see the correlation coefficient is mainly concentrate around '1'. Which means the wind speed may be the most influential factor of power generation.

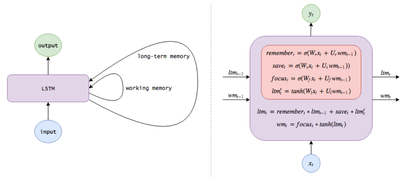
Deep Learning for Power Forecasts

Long Short-Term Memory

The Long Short-Term Memory network (or LSTM), is a specific designed deep recurrent neural network. It is trained using Back-propagation through time, and overcomes the vanishing gradient and the long-term dependency problem.



Problem Formulation



Learning objective:

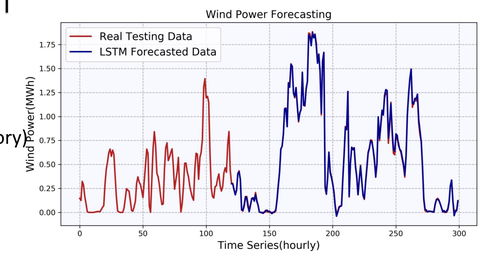
$$\theta_{\{W_r, W_s, W_f, W_i, U_r, U_s, U_f, U_i\}} = \arg\min\Big(\sum_{i=1}^{batch\ size} \| \hat{y}(t) - y(t) \|_2\Big)$$

Network Structure:

- Forget Gate Layer(deals with long/short memory)
- Input Gate Layer(Input representation)
- Output Gate Layer(Output Loss optimization)

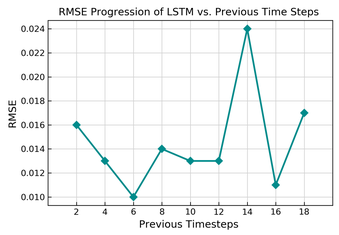
Simulation Setup

LSTM memory: 6 hours with interval of 1hour
Activation function: Sigmoid/Tanh
Training epochs: 100 (with early stop conditions)
Batch size: 2/14/60
Hidden Layer size: 5/10
Project prerequisites: Tensorflow, Python 3.5



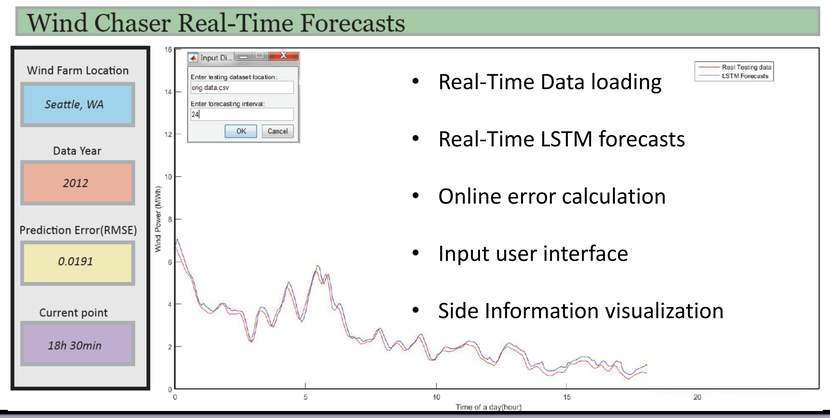
Simulation Results

	RNN unit = 10 Batch size = 14	RNN unit = 5 Batch size = 14	RNN unit = 10 Batch size = 2	RNN unit = 10 Batch size = 60
Without Wind Power Input	RMSE = 0.010 Wall Time: 44.2s	RMSE = 0.016 Wall Time: 43.8s	RMSE = 0.004 Wall Time: 4min31s	RMSE = 0.046 Wall Time: 14.8s
With Wind Power Input	RMSE = 0.012 Wall Time: 41s	RMSE = 0.026 Wall Time: 36.5s	RMSE =0.004 Wall Time: 4min 2s	RMSE = 0.057 Wall Time: 17.3s



- Two wind power forecasting model (with or without history wind power values)
- Results on different time window/batch size
- Model trained till loss convergence
- 5 input features (e.g., wind speed, wind direction, temperature)
- RMSE reported on stand alone test sets
- Accurate regression results reported

Graphical User Interface

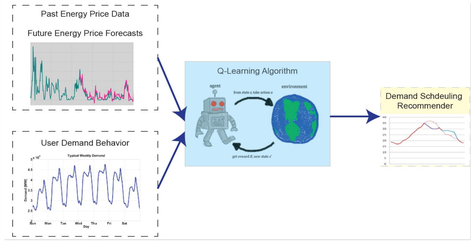


- Real-Time Data loading
- Real-Time LSTM forecasts
- Online error calculation
- Input user interface
- Side Information visualization

Q-Learning for Energy Use Scheduling

Motivation:

a) Renewable energy price is **highly volatile**
b) For a user, objective is to **minimize power costs** by changing electrical appliances **working schedule**



Learning Framework:

Input data:

- Past price data;
- Future price forecasts;
- User demand profile

Learn to make decision:

- Balance short/long term profits(e.g., energy costs)
- Find the optimal action

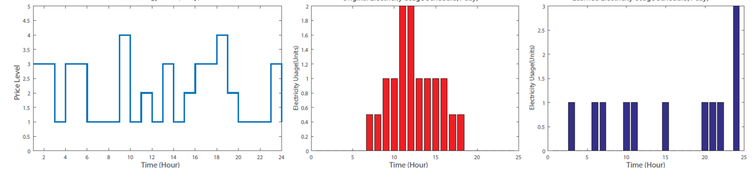
Problem Formulation:

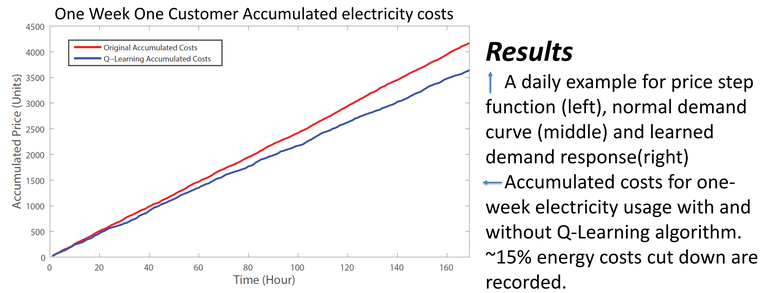
Q-leaning updates: $r(s_t, a_t) = U(s_t, a_t) + f(p_t, a_t)$

$$Q_{k+1}(s_t, a_t) = Q_k(s_t, a_t) + \alpha[r(s_t, a_t) + \gamma \max_a Q(s_{t+1}, a_t) - Q(s_t, a_t)]$$

Optimality condition holds based on Bellman's equation

$r(s_t, a_t)$: Reward function $f(s_t, a_t)$: Energy cost function
$Q(s_t, a_t)$: Update till convergence $U(s_t, a_t)$: User utility satisfication function
$s(t)$: Daily electricity usage profile $p(t)$: Electricity Step Price $a(t)$: Buying or hold option





Results

A daily example for price step function (left), normal demand curve (middle) and learned demand response(right)
Accumulated costs for one-week electricity usage with and without Q-Learning algorithm. ~15% energy costs cut down are recorded.

Conclusions & Future Work

- Derived the loss functions scalable for variable length forecasting problems
- Simulated and validated model performance on a set of forecasts test sets.
- Brought out a following up reinforcement learning algorithm for scheduling electrical appliances
- The integrated algorithm would benefit both renewable generations and end-use customers; real-time tests and validation

References:
[1] Gensler, André, et al. "Deep Learning for solar power forecasting—An approach using AutoEncoder and LSTM Neural Networks." *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*. IEEE, 2016.
[2] O'Neill, Daniel, et al. "Residential demand response using reinforcement learning." *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*. IEEE, 2010.
[3] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016, November). TensorFlow: A System for Large-Scale Machine Learning. In *OSDI* (Vol. 16, pp. 265-283).
*We would like to thank Prof. Baosen Zhang, Prof. David Beck and Moke Mao for precious advice and discussions.