

An Integrated Artificial Neural Network Multi-Agent System for Energy Demand Management

Full Unit Project: Project Plan

Benedict Wilkins

Supervisor: Zhiyuan Lou

2016

Abstract

Energy demand management (EDM) has become an increasingly pressing issue in our technological age. EDM, specifically demand-side management (DSM) is concerned with planning and implementing models that allow consumers to alter their habits. This leads to reduction in energy demand peaks and overall energy consumption.¹ Energy providers have a number of ways of managing fluctuations in energy consumption that include buying energy, using peaking power plants and energy storage techniques.

This project aims to provide an alternative solution by integrating two related sub-fields in computer science; machine learning and multi-agent systems. This has become possible with the advent of Smart Meters that are now being offered by providers such as British Gas.² Modelling these Smart Meters as agents and placing them in a distributed multi-layered environment allows automated data collection and pre-processing. The processed data can then be given to a top level agent with specialised prediction capabilities.

Artificial neural networks (ANNs) will provide these prediction capabilities. ANNs have proven to be an effective method of prediction and classification. They have been used in time series forecasting to model complex systems such as financial markets and foreign exchange markets and so will be suited to this problem.³ ANNs have the ability to approximate highly non-linear functions and so have been used instead of more traditional statistical models. Using ANNs in this case will further explore their forecasting capabilities.

¹ (Davito, Humayun, & Robert, 2009)

² (British Gas, 2016)

³ (Oancea & Ciucu, 2014)

Final Deliverables

The final system will consist of an ANN, a multi-agent system, a data generator, and a system interface. It will collect and process data automatically and use this data to make forecasts on future energy demand. The system will be flexible and scalable, a user may want to forecast for a certain region or for more than one region at a time. The agent environment should be structured in a way that will allow this. The system should scale up, with 25 million homes in the UK⁴ all using utilities and potentially in the future all with smart meters. It is unrealistic to test such a large number of agents but some effort will be made to test the scalability of the system. The system should be distributed (see Agents Framework below).

ANN

The ANN will be responsible for creating forecasts based on the data collected by the agents. The exact implementation of the ANN will be discussed in the Artificial Neural Network for Time series Prediction report. A specialised agent will house the ANN - the prediction agent. See Agent Framework below.

Data Generator

The data generator will generate realistic data to be used in the system (see Proof of Concept Programs). The data will be used to train the ANN and as a way of simulation data transfer in the multi-agent system. It will essentially provide a test bed for the system as a whole. If a suitable smart meter reading dataset cannot be found this will act as the sole data provider to the system. There are however various places to retrieve this kind of data so this is unlikely.⁵

System Interface

The system interface will be used by the user(s) to communicate/configure the system. The user(s) will be able to configure the ANN, the agents, the agent environment, the data generator. A user will also be able to see system information such as work-loads of agents, data being collected, statistics, and most importantly the forecasts made by the ANN.

⁴ (BBC News, n.d.)

⁵ Two examples of potential data acquisition. Also see Data risk.

<http://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households>

<http://www.ucd.ie/issda/data/commissionforenergyregulationcer/> (data needs to be requested)

Agent Framework

The agents will be placed in a multi-layered environment, each layer will contain one or more agents and each agent in a layer may be a manager of some agents in the layer below illustrated in figure 1. The system will be developed to have arbitrary layers in this fashion with the top layer agent(s) as prediction agents, the bottom layer as data collection agents and intermediate data pre-processing agents. Depending on the amount of data pre-processing to be done it may be preferable to distribute the system in terms of layers or agents. The agents will have the capability to communicate via an internet protocol of choice. This can be done using the Starlite framework⁶ as it allows sensor/actuator modules to be attached to the agents, these modules can be set up to accommodate any protocol. The full details of the Starlite framework will be discussed in the Introduction to Multi-agent Systems report mentioned below.

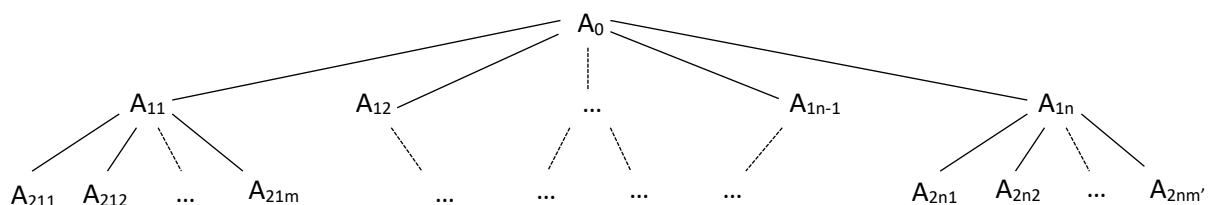


Figure 1. A_L is an agent at layer L . A_0 will typically be the prediction agent – in this case the one that uses the ANN. Agents directly below A_0 will be supplying it with clean formatted data. The bottom layer of the hierarchy will be reserved for data collection – in this case the smart meter agents. The lines represent communication channels between the agents. These channels will be used to send the collected data up the hierarchy. The channels will be bi-directional as an agent may want to send control data to an agent it is managing. See figure 2 for an illustration of communication channels.

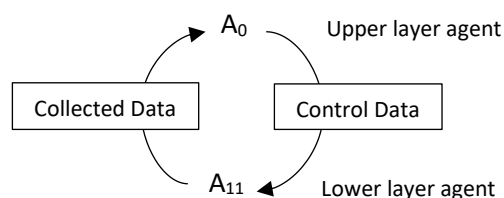


Figure 2. Control data is sent down the hierarchy and the collected data – in this case the smart meter data is sent up the hierarchy. Control data may be any data and will depend on the systems use. In this system it may be instructions for data pre-processing.

Possible extensions

- Method of altering the data generation system in order to take into account any predictions made – see how this affects future predictions made.
- A system visualizer showing useful graphs and statistics

⁶ //Agent and environment – GOLEM etc find a proper reference!

Reports

The reports described below will provide the base theory of the project. They will talk about design choices in this project, including - why use ANNs and multi-agent framework and how to implement both? The multi-agent system will be developed in java (as Starlite is a java framework), there is still some exploration to be done on the implementation details of the ANN – TensorFlow is a good candidate. They will also go into detail on the integration of the two systems to create the complete system.

Smart Meters and Energy Demand Management

Discussing EDM and Smart meters in this project is important. Smart meters will be the sole data collectors and so some study will be done on ones currently available. This will be to find out information such as the format of the data collected and time intervals of collection. There will also be a discussion on EDM; the alternative solutions to the problem and motivations for this solution in particular.

Introduction to Multi-agent Systems

This report will explore multi-agent systems in general and more specifically GOLEM⁷ - more recently Starlite. It will lay the theoretical foundation for the final system to be developed and include how the agents will be modelled, how they will communicate and how the environment they act in will be set up. It will also contain a brief section of distributed agent environments as in a real setting the system the smart meter agents will be external – separated by location.

Introduction to Artificial Neural Networks

Artificial neural networks are the core of this project, this report will provide the base for later reports by talking about the fundamentals of ANNs. These include base concepts; activation functions, weights, nodes, layers and the methods that drive ANNs including the famous Backpropagation training method. This will include a mathematical derivation of Backpropagation.

Artificial Neural Networks for Time Series Prediction

Moving forward from the introduction to ANNs, this report will explore ANNs for time series forecasting/prediction in detail. Including an exploration of the different ANN models including RNNs, LSTMs⁸ and the sliding window input approach. There will be some discussion on the properties of time series data e.g. seasonality, and what makes it different from other kinds of data.

⁷ (Alrayes, Kafali, & Stathis, 2016)

⁸ (MachineLearner, 2016)

Proof of Concept Programs

The proof of concept programs below will demonstrate the different components of the final system. The Data Generation System and Data Pre-processor will be used in the final system. A lot of thought will go into their design to make sure they are suitable. This will cut down the work load in the second term. The other programs will probably be used as templates for the final system, they will explore the possibilities of the final system and act as experiments for making later design decisions.

A Basic Multi-Agent System

A small multi-agent system that is hierarchical, agents will communicate with each other and have data processing capabilities; illustrated in figure 1.0. The GOLEM – more recently Starlite framework will be used to develop the agents and their environment in java. Some thought will be given to the integration of java with the machine learning library that will be used to build the ANN.

A Basic Artificial Neural Network

Implement a basic ANN and train it. The ANN will try to model 3 logic functions AND, OR and XOR.

ANN model (LSTM, RNN, sliding window)

Implement the chosen ANN model, this will depend on the resources available to me (LSTMs require a lot of processing power!) The model to be used in the final system will be decided here. A list of requirements should be made including – scalability, resources availability etc. The most suitable model will depend on these requirements.

Data Generation System

The data generation system will generate *useful* and realistic data that can be used in a multi-agent system. This program will consider how to provide the data to the agents in a realistic way – allowing a simulation of the final system later on.

Time Series Data Pre-Processor

An implementation of a data pre-processor specifically for time series data. This will be one example of a module that can be attached to an intermediate data processing agent. Some research will be done into pre-processing for ANNs.

Timeline

(P = Program, R = Report)

Key Dates			
First Term		Start Date	Delivery Date
Project Plan		-	23:59 Friday 30 th September 2016
Smart Meters and Energy Demand Management: This is the problem definition, it is best to define the problem before starting. It will also lead on to the multi-agent system/data generation program – how to collect/simulate data from smart meters	R	Friday 30 th September 2016	Friday 7 th October 2016
Introduction to Multi-agent Systems: This report should be written at the same time as the basic multi-agent system as they complement each other, I will likely get some insight into multi-agent systems while implementing one.	R	Friday 7 th October 2016	Wednesday 19 th October 2016
A Basic Multi-Agent System: See above	P	Friday 7 th October 2016	Wednesday 19 th October 2016
Introduction to Artificial Neural Networks: This report should be written at the same time as program below for the same reason as the above agent report/program pair. This ANN report/program pair will leading nicely on to the more complex topic of Artificial neural networks for time series prediction.	R	Wednesday 19 th October 2016	Wednesday 2 nd November 2016
A Basic Artificial Neural Network: See above	P	Wednesday 19 th October 2016	Wednesday 2 nd November 2016
Data Generation System: The data generated here can be used for testing the programs below.	P	Wednesday 2 nd November 2016	Monday 7 th November 2016
Time Series Data Pre-processor: The pre-processor leads on to implementing the ANN model, the model will need clean data for testing.	P	Monday 7 th November 2016	Sunday 13 th November 2016
Artificial Neural Networks for Time Series Prediction: This report should be written at the same time as the program below for the same reason as before. At this point all of the component/templates are ready for the final system.	R	Sunday 13 th November 2016	Monday 28 th November 2016
ANN model (LSTM, RNN, sliding window) : See above	P	Sunday 13 th November 2016	Monday 28 th November 2016
Interim Programs and Reports: Submission of programs and reports		-	23:59 Thursday, 1 st December 2016
Prepare for Interim Review Viva: Create a presentation and prepare for below.		Thursday 1 st December 2016	Sunday 4 th December 2016
Interim Review Viva: Presentation about the project.		-	Monday 5 – Friday 9 December 2016
Christmas Holidays		Start Date	Delivery Date

Review First Term Programs/Reports: How successful was the Data Generation and Time Series Data Pre-processor, they will likely be used in the final project so they should work well.	Friday 9 th December 2016	Friday 16 th December 2016
Fully Functioning Multi-agent System: Build the multi-agent system. This should include the environment, the communication sensors/actuators for the agents and specialised sensors for the smart meter agents – these should be compatible with the Data Generation System assuming it has been completed properly	Friday 16 th December 2016	Monday 9 th January 2017
Second Term: Milestones	Start Date	Delivery Date
Implement the Final ANN: Build the ANN, the model will have been chosen in the Artificial Neural Networks for Time Series Prediction report and implementation details will have been well thought out by this point including the libraries to use, and what machine I will be using to run it.	Monday 9 th January 2017	Tuesday 31 st January 2017
Integrate ANN with the Multi-agent System: Add the ANN to the top level prediction agent – the prediction agents should have the capability but if not it should be implemented here. (note that some time will be spent on the final report from here on)	Tuesday 31 st January 2017	Monday 20 th February 2017
Start writing the final report: This may be started during the above milestones, it will be good to record things (the project diary) and use this.	Wednesday 1 st February 2017	2pm Thursday 30 th March 2017
Full Unit Draft report ready	-	Friday 17 th February 2017
Create the interface: The interface will be partly/completely developed by this point as it will be required for testing the system. It should be finalised and completed here.	-	Monday 6 th March 2017
Integrate everything: The project is essentially complete at this point, this time should be spent finalising the project and testing.	Monday 6 th March 2017	2pm Wednesday 30 th March 2017
Full Unit Final Programs and Report: Final deadline.	-	2pm Wednesday 30 th March 2017

Risk Assessment

Computing Power: Risk: Low, **Severity:** High, **Impact:** Potentially not being able to continue with ANNs at worst.

A major risk is not having the enough processing power to handle an ANN algorithm. ANNs are notorious for requiring heavy processing as the algorithms typically have thousands of iterations and involve transformations on large matrices. A powerful computer will be required in order to use (particularly) an LSTM for this project. If this is not available the project may continue using an RNN or using the sliding window model (A relatively powerful machine will still be required but it will reduce the load). The severity can be further reduced by limiting the number of agents in the system – reducing the amount of data and so reducing the amount of computation required.

Data: Risk: Low, **Severity:** Medium, **Impact:** Not being able to continue with ANNs (if Timing risk is also an issue)

ANNs are a data driven approach to prediction, they require large amounts of data to work well.⁹ There may not be sufficient real world data to train the ANN and so data will be generated using a data generation system. This system will supply data to the smart meter agents and be used to show that the system as a whole can work in a real world setting.

Timing: Risk: Low, **Severity:** Medium (if Data risk fails then High), **Impact:** Removing the data generation system

The project is large, with two main sections – the ANN and multi-agent system. These two should be the main focus of the project. Some effort should be made on the interface but it can be quite basic. The data generation system is not essential provided there is real world data (see Data risk). Removing this aspect should free up time to work on the rest of the project.

Integration: Risk: Low, **Severity:** High, **Impact:** Re-writing the ANN in the worst case

If there is no way to effectively integrate the multi-agent system with the ANN. This should be one of the requirements when selecting libraries/languages to use to implement the ANN and so is unlikely to become an issue. If it is realised, the ANN will have to be written using a different language/library as the multi-agent system is set to be written in java.

⁹ (Allende, Moraga, & Salas, 2002)

Bibliography

- Allende, H., Moraga, C., & Salas, R. (2002). Artificial neural networks in time series forecasting: A comparative analysis. *Kybernetika*, 38, 685 - 707.
- Alrayes, B., Kafali, O., & Stathis, K. (2016). RECON: A Robust Multi-agent Environment for Simulating COncurrent Negotiations. *Advances in Agent-base Complex Automated Negotiation*, 157-174.
- BBC News. (n.d.). Retrieved from BBC:
<http://news.bbc.co.uk/1/shared/spl/hi/guides/456900/456991/html/>
- British Gas. (2016). *Smart Meters*. Retrieved from British Gas: <https://www.britishgas.co.uk/smarter-living/control-energy/smart-meters.html>
- Davito, B., Humayun, T., & Robert, U. (2009). *The smart grid and the promise of demand-side management*. McKinsey and Company.
- MachineLearner. (2016, June 14). *CS231n Lecture 10 - Recurrent Neural Networks, Image Captioning, LSTM*. Retrieved from Youtube: <https://www.youtube.com/watch?v=iX5V1WpxxkY>
- Miller, S. (2015, August 10). *Mind: How to Build a Neural Network (Part One)*. Retrieved from Github: <http://stevenmiller888.github.io/mind-how-to-build-a-neural-network/>
- Mitchell, T. M. (1997). *Machine Learning*. MIT Press, The McGraw-Hill Companies Inc.
- Oancea, B., & Ciucu, S. C. (2014). Time series forecasting using neural networks. *arXiv*, 1.
- PyData. (2015, December 4). *Alex Rubinsteyn: Python Libraries for Deep Learning with Sequences*. Retrieved from Youtube: <https://www.youtube.com/watch?v=E92jDCmJNek>

Bibliography Discussion

(Allende, Moraga, & Salas, 2002), This paper compares traditional methods of time series forecasting such as ARMA with ANNs, the mathematical theory goes quite in depth, but the bulk of the paper is quite readable. The project was going to use neural networks from the start but this paper demonstrated that ANNs can be suitable for time series forecasting.

(Alrayes, Kafali, & Stathis, 2016), This paper gives an example of an agent framework – GOLEM and its application. This paper encouraged the use of multi-agent systems for a project such as this.

(BBC News, n.d.), gives an indication of how the system may scale.

(British Gas, 2016), Gives some information on smart meters, this information will be requires when building the smart meter agents.

(Davito, Humayun, & Robert, 2009), A good discussion of the problem, this paper shows that the problem is still ongoing.

(MachineLearner, 2016), Gave an introduction to ANNs for time series forecasting including RNNs and LSTMs. Suggested using numpy machine learning library for implementation.

(Miller, 2015), Gives a very basic example of an ANN. Good understanding of how ANNs work is required for the project, this example is a start to that understanding.

(Mitchell, 1997), is a standard place to start for machine learning, it introduces the theory and required concepts. The most relevant chapter was Artificial Neural Networks, it gave a good introduction to ANNs and goes into more detail later in the chapter.

(Oancea & Ciucu, 2014), an experimental comparison between different ANN models for time series forecasting. Specifically a classic feed forward network and an RNN, the RNN performed better. This demonstrates that a more specialised ANN is better used for forecasting.

(PyData, 2015), an introduction to LSTMs, and RNNs. It give some examples of machine learning libraries including Keras. In this case they are used for sequential data analysis – but they can be used for time series forecasting.