EARL 2016

EFFECTIVE APPLICATIONS OF THE R LANGUAGE

London 13th - 15th September

Modelling Boiler Breakdown using Deep Learning Algorithms

Timothy Wong

Data Scientist, Centrica plc

Who we are?

'Centrica is an international energy and services company. Everything we do is focused on satisfying the changing needs of our customers.'

Bord Gáis Energy

British Gas

Scottish Gas

centrica

centrica storage

Direct Energy

DYNO

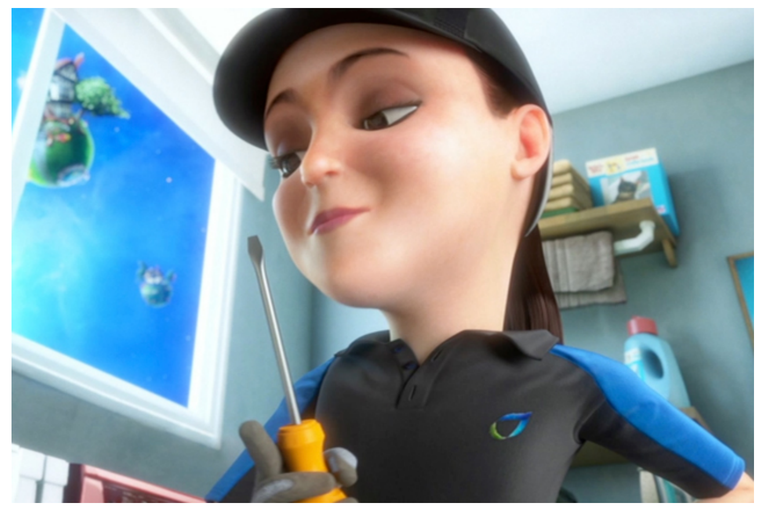
hive

Timothy WONG – Data Scientist
https://uk.linkedin.com/in/timothy-wong-7824ba30

Problem

Scottish Gas British Gas

- We know how many boiler breaks down on each day when customers decide to inform us
  - Can we model historical trends?
  - If so, can we make inference to the past/future?



Timothy WONG – Data Scientist
https://uk.linkedin.com/in/timothy-wong-7824ba30

Why Neural Network?

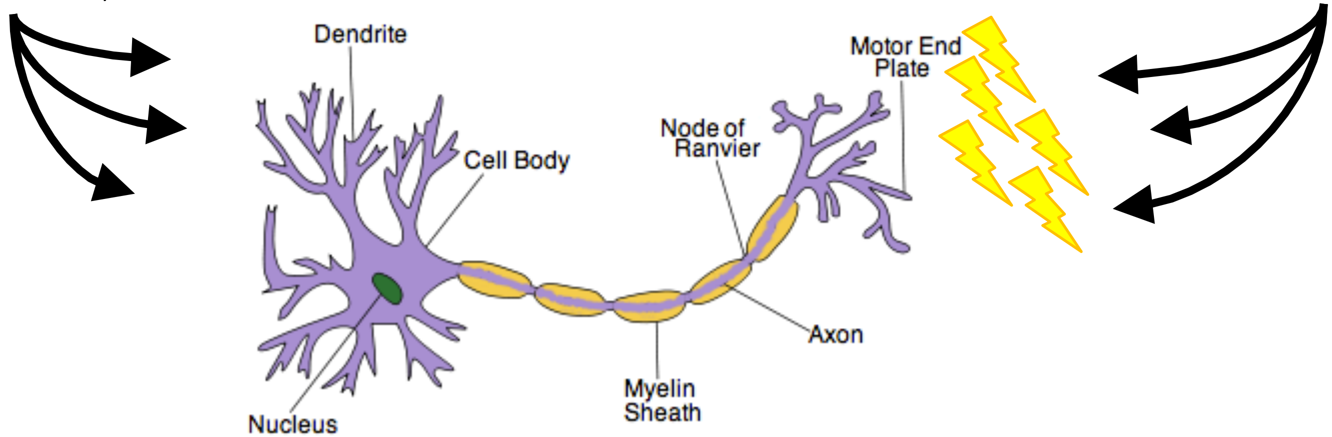
- Non-parametric method
- Do not assume underlying model structure
- Learns complex behaviour

Neural Network

Inputs
(from other neurons)

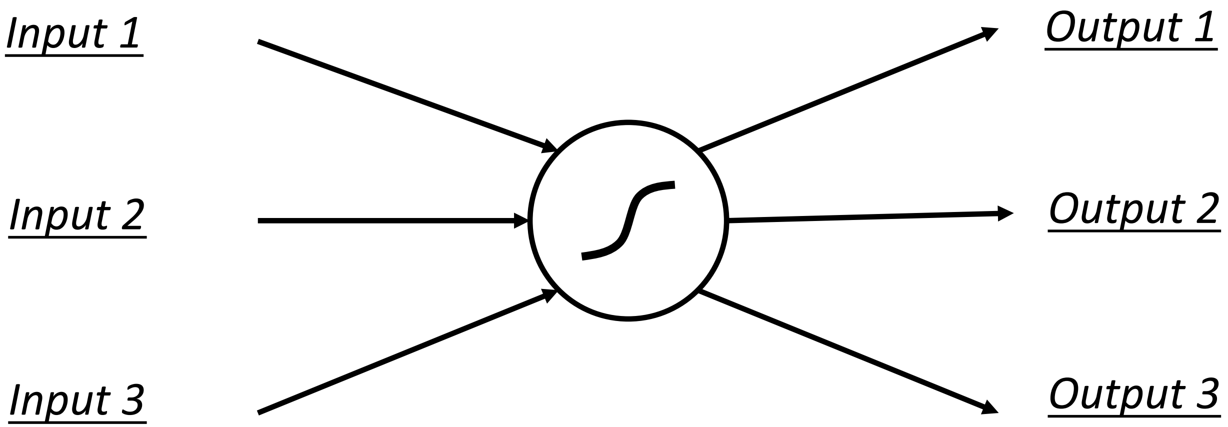
Process→→→

Outputs
(to other neurons)



Artificial Neural Network (1)

- ANN is a mathematical model based on biological brains
  - Each neuron has multiple inputs and multiple outputs
  - Non-linear activation function
  - ANN models normally contains thousands / millions of parameters



Artificial Neural Network (2)

$$y = f\left(\sum_{i=1}^{M} w_i x_i\right)$$

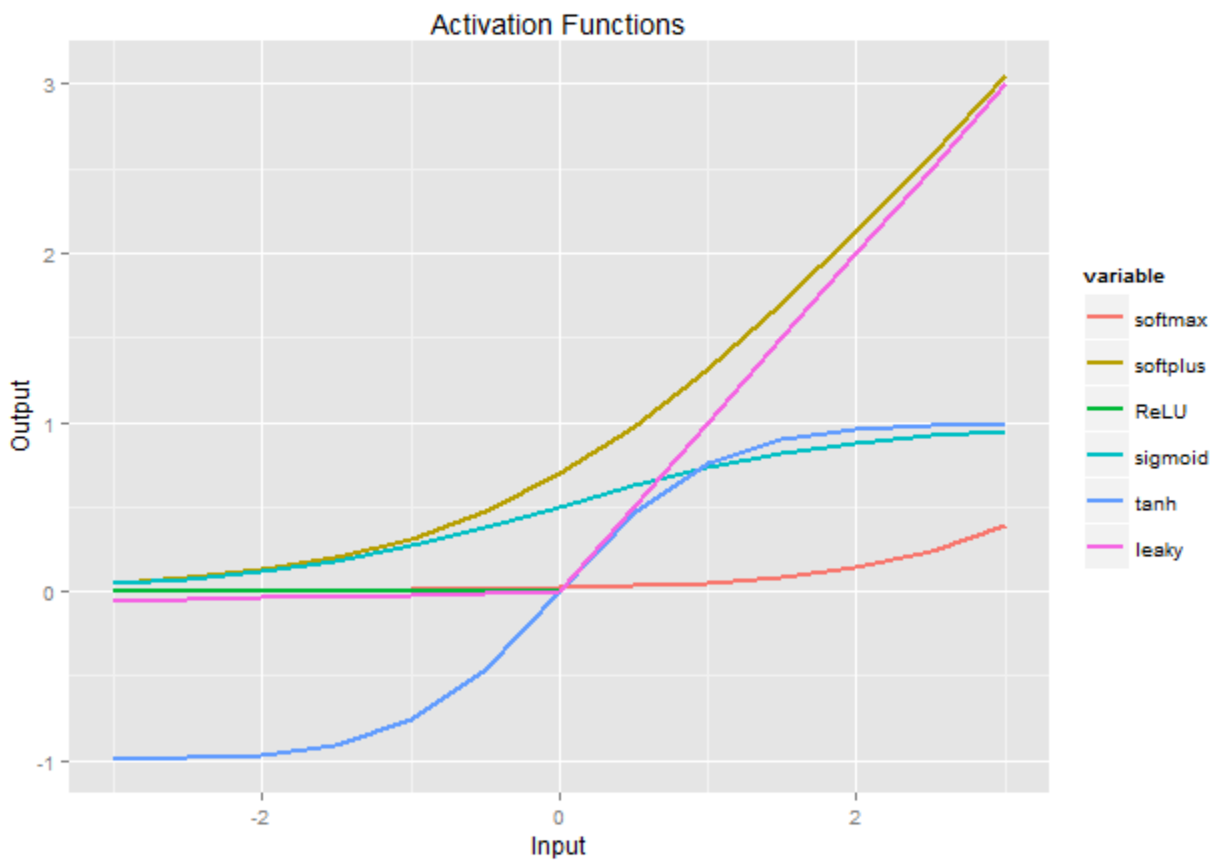
… where y is one of the neuron outputs

Pick an non-linear activation function…:

Softmax	$f(x) = \frac{e^x}{\sum_{k=1}^{K} e^{x_k}}$	$f(x) = \frac{1}{1+e^{-x}}$	Sigmoid
Softplus	$f(x) = \ln(1+e^x)$	$f(x) = \tanh x$	Hyperbolic Tangent
ReLU (Rectifier)	$f(x) = \max(x, 0)$	$f(x) = \max(x, ax)$	Leaky ReLU

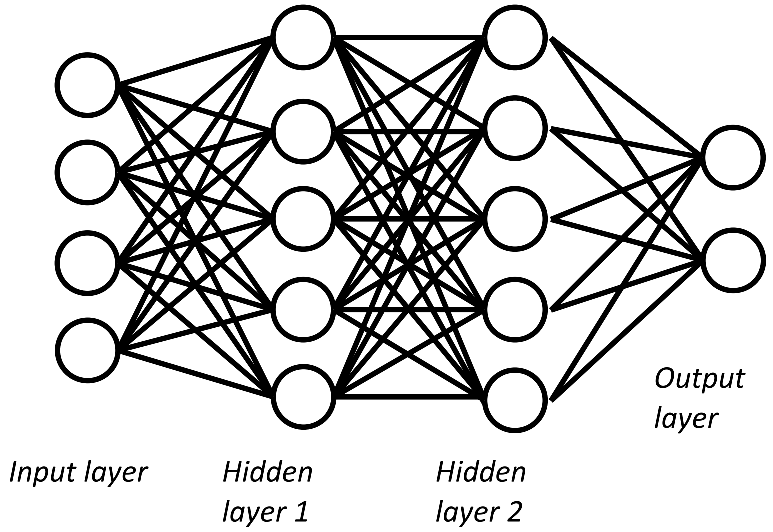
Timothy WONG – Data Scientist
https://uk.linkedin.com/in/timothy-wong-7824ba30

Artificial Neural Network (3)



Artificial Neural Network (4)

- A ‘classic’ neural net:
  - Has multiple input neurons
  - Has multiple layers of *fully-connected* hidden neurons
  - Has at least one output neuron
  - Always forward-feeding (also called MLP: *Multi-Layer Perceptron*)

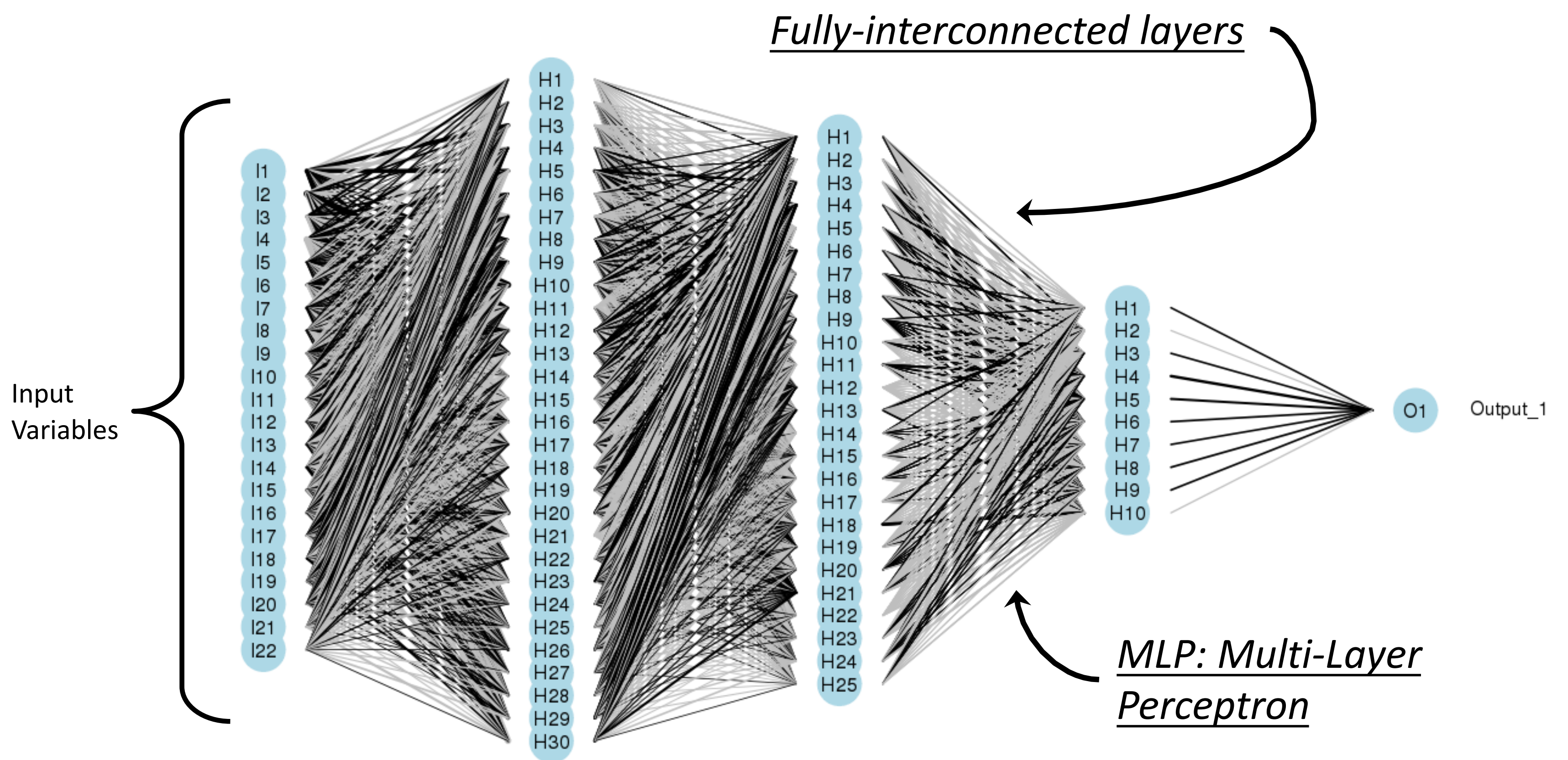


Artificial Neural Network (5)

- It is “deep learning”
  - Defined as multiple processing layers of non-linear relationships
- Neural nets can process regression or classification problems
- It’s very common in handling complex non-linear problems.

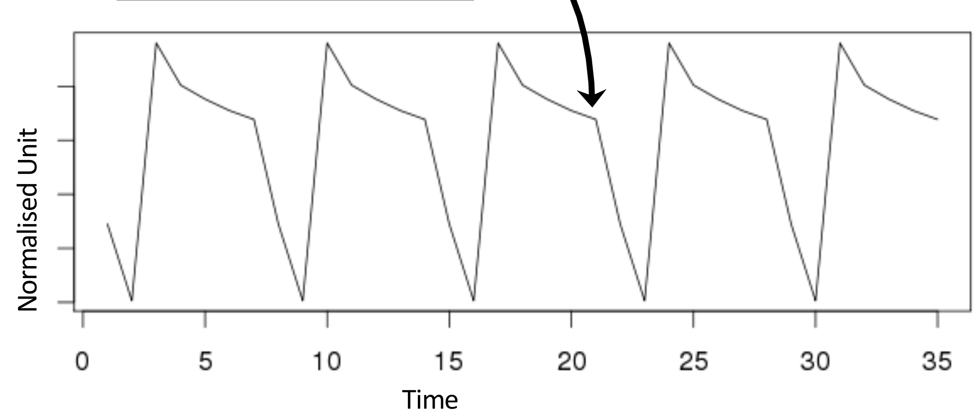
Timothy WONG – Data Scientist
https://uk.linkedin.com/in/timothy-wong-7824ba30

Artificial Neural Network (6)

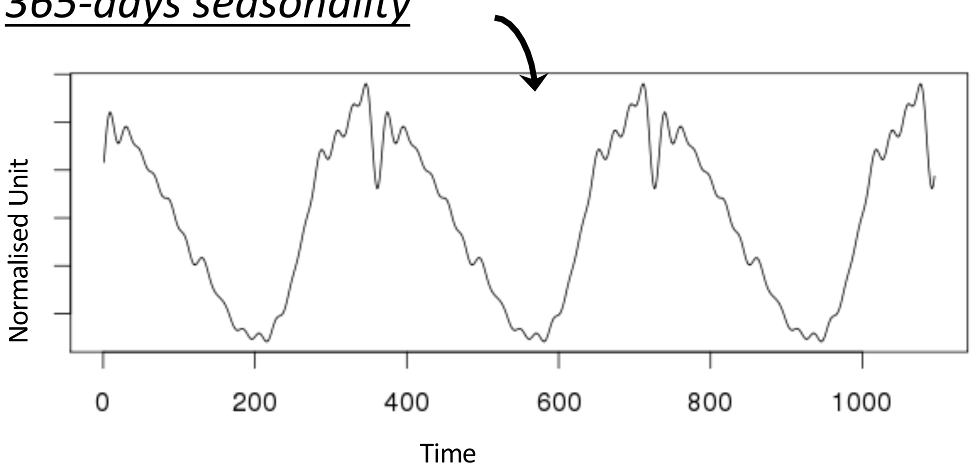


Time Series Decomposition
(Multiple Seasonality)

7-days seasonality

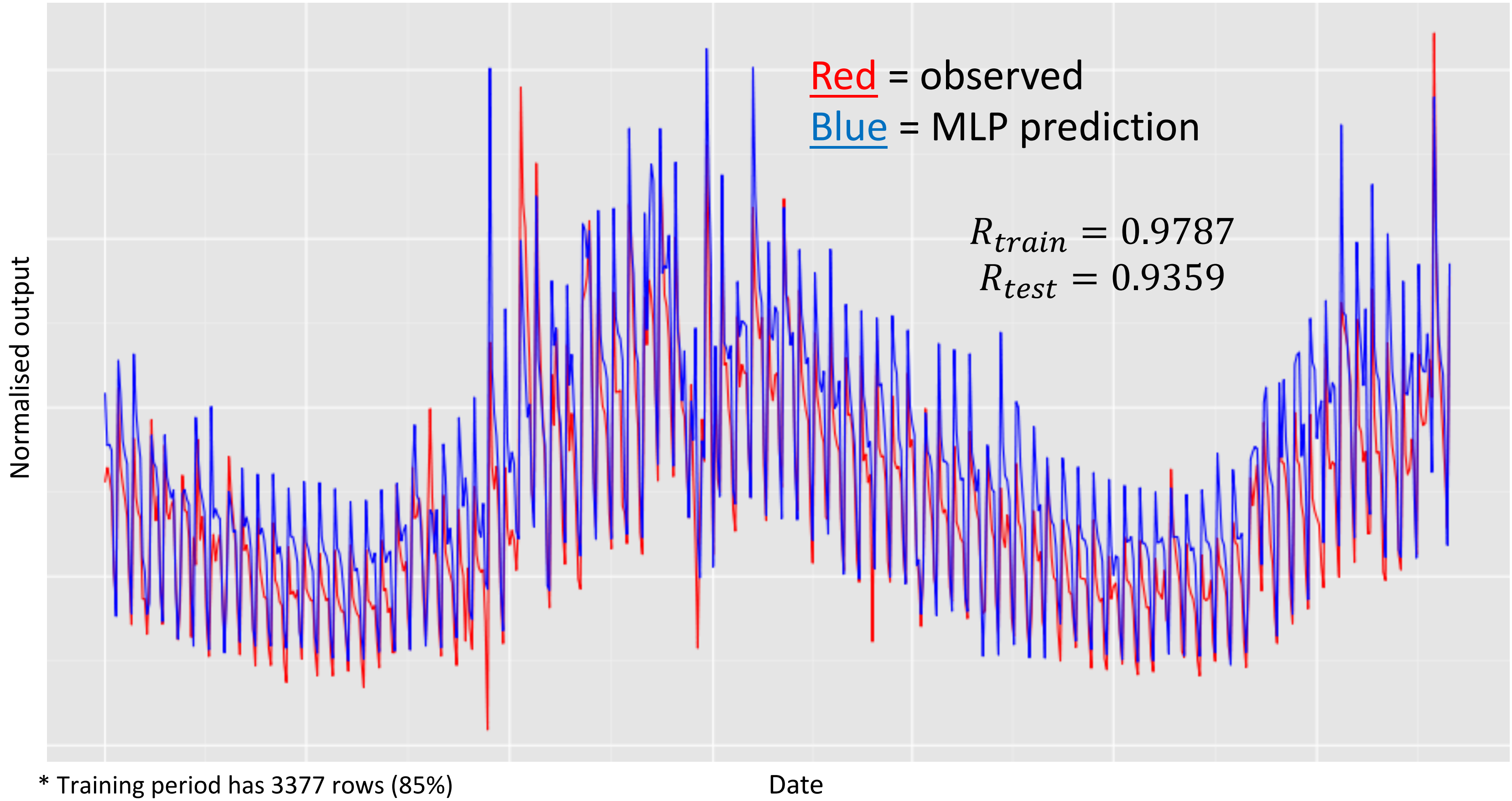


365-days seasonality



Timothy WONG – Data Scientist
https://uk.linkedin.com/in/timothy-wong-7824ba30

Artificial Neural Network (7)



* Training period has 3377 rows (85%)
Test period has 609 rows (15%)

Artificial Neural Network (8)

- Always starts with random seed
  - Gradient descent = locally optimal
  - Therefore results are not always reproducible

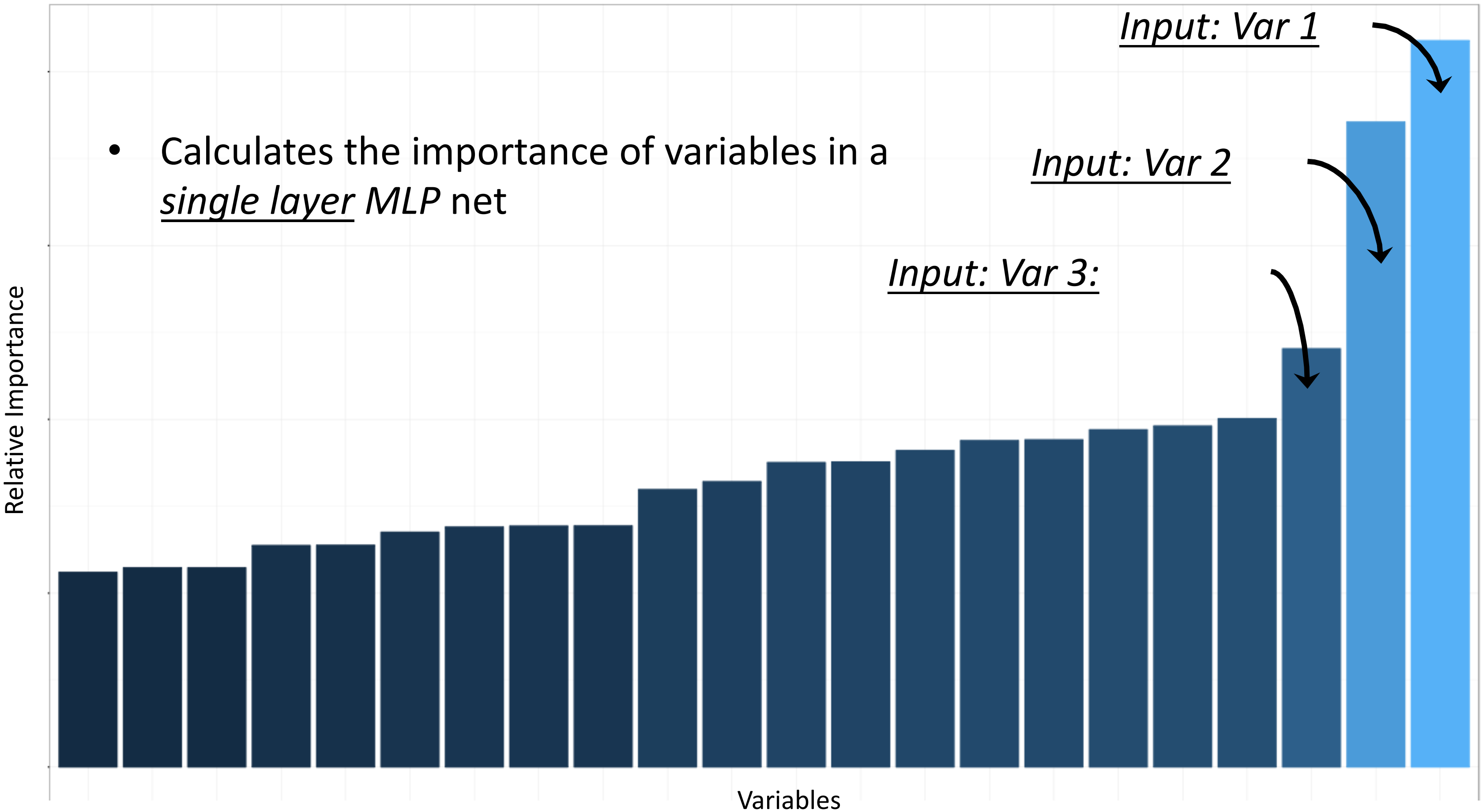
- ANN is a black box system
  - Regression coefficient & p-values do not exist !
  - Difficult to extract useful insight from model weights

- Long training time

Timothy WONG – Data Scientist
https://uk.linkedin.com/in/timothy-wong-7824ba30

Garson's Algorithm

- Calculates the importance of variables in a *single layer* *MLP* net



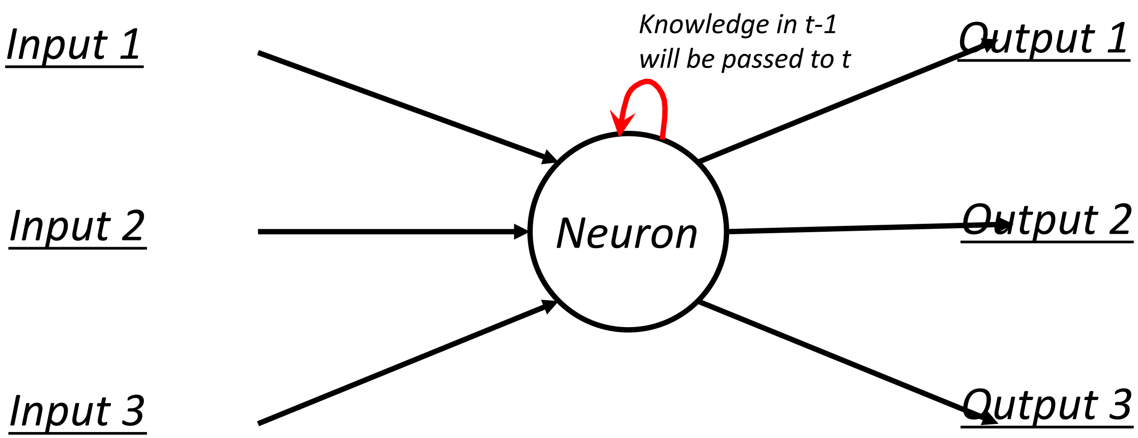
Recurrent Neural Network (1)

- Neural nets can be further adapted to deal with time series
  - Time series implies data must be presented in the correct sequence (i.e. not possible to flip observations)
- So far the concept of time is not yet present in Neural Nets
  - i.e. in MLP you can flip the sequence of observations and still achieve the same model theoretically
  - Solution = Recurrent Neural Network (RNN)
- Elman/Jordan type recurrent networks
  - Good for short term memory (i.e. state at time t depends on $t-1$)
  - LSTM network for longer term lag effect

Timothy WONG – Data Scientist
https://uk.linkedin.com/in/timothy-wong-7824ba30

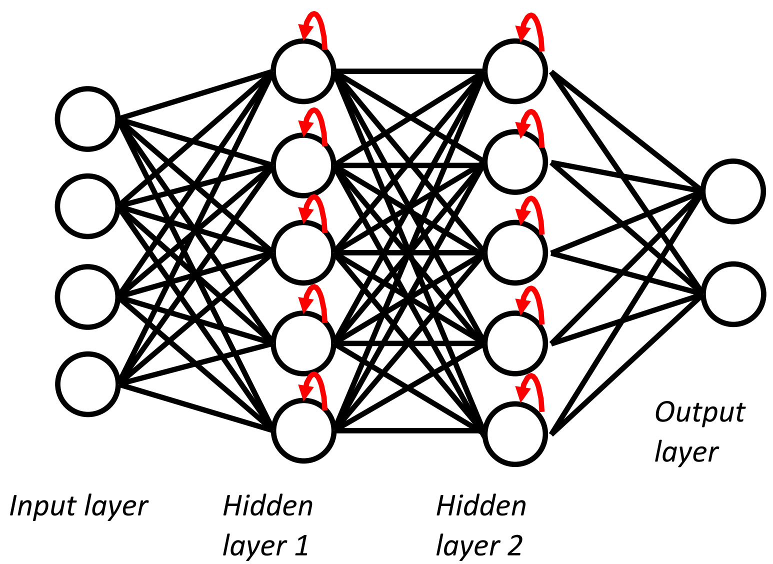
Recurrent Neural Network (2)

- RNN is a special variant in neural net
  - It knows how to remember what happened at previous time step.
  - To rephrase: it has feedback loops
    (Step t is dependent on $t-1$)



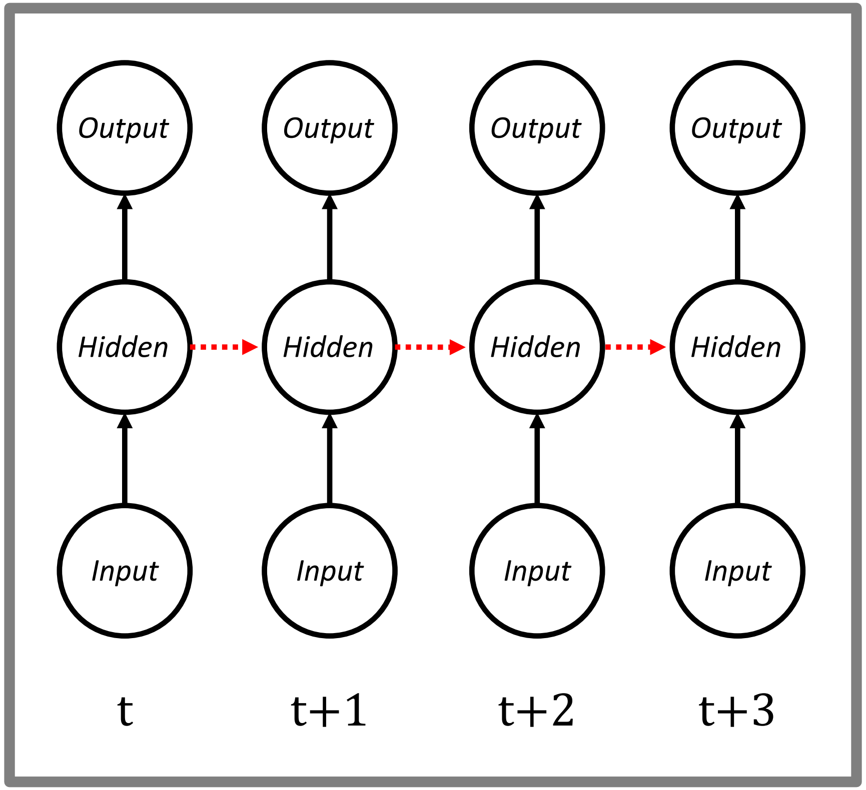
Recurrent Neural Network (3)
Elman Network

- *Hidden* neurons relate to the previous state



Recurrent Neural Network (4)

Unfolding RNN: "Back-Propagation Through Time (BPTT)"



Elman Network

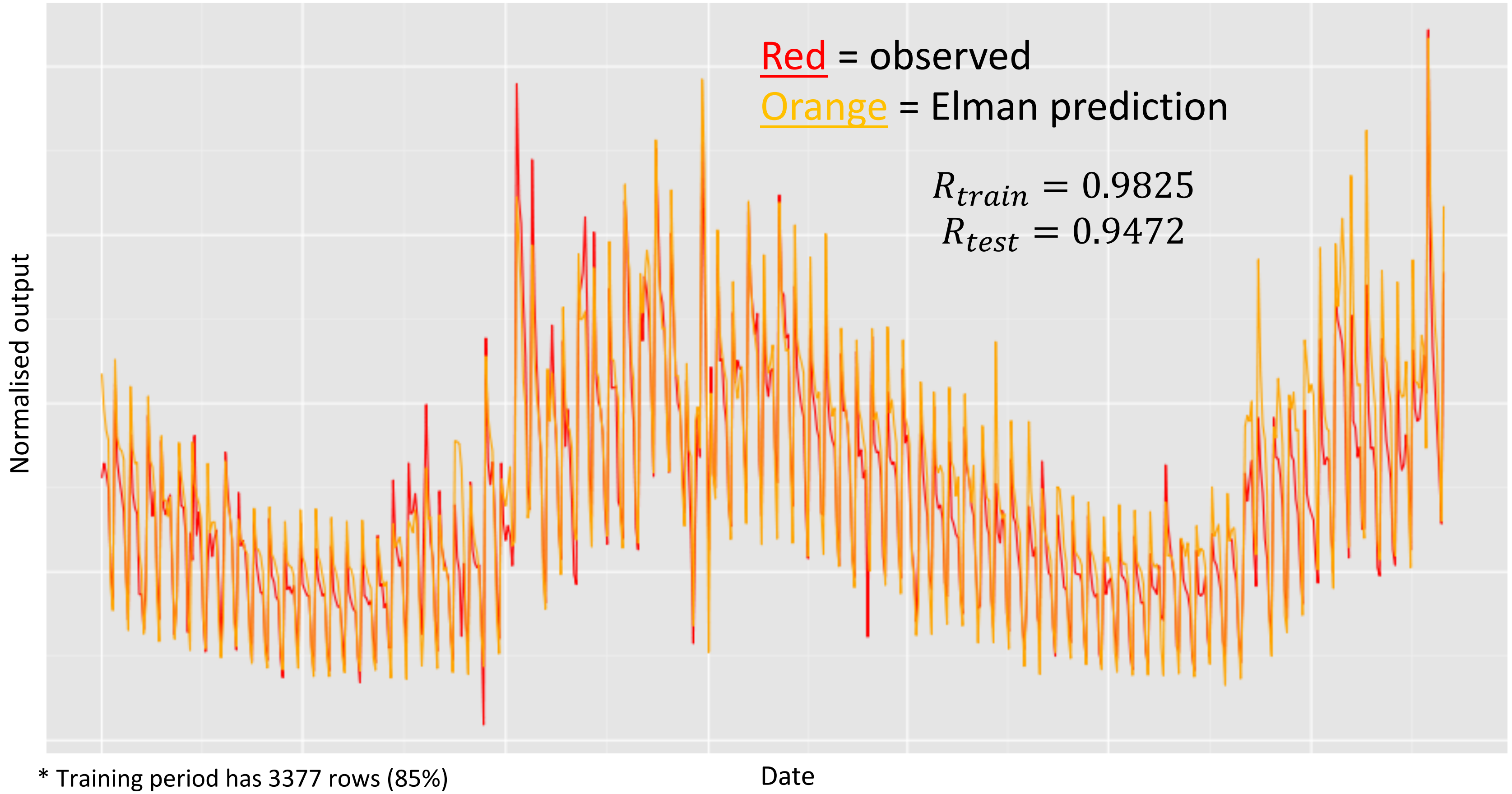
Elman network:

Turning temporal relationship into forward-feeding network using BPTT

Timothy WONG – Data Scientist
https://uk.linkedin.com/in/timothy-wong-7824ba30

Recurrent Neural Network (5)

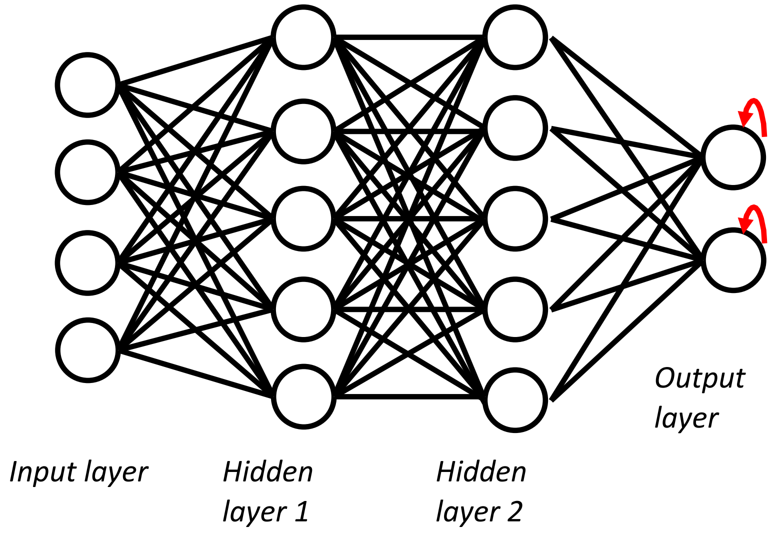
Elman Network



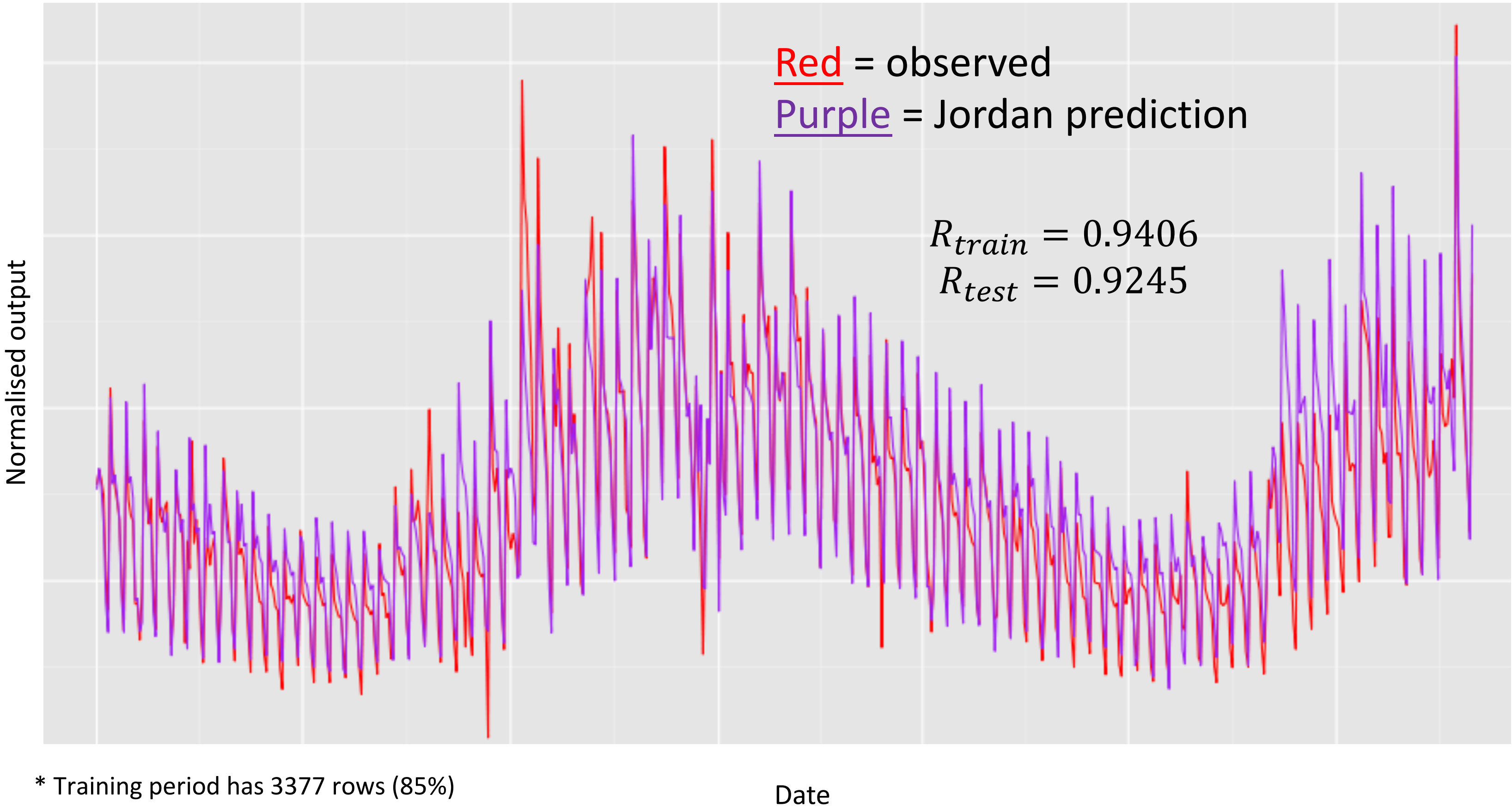
* Training period has 3377 rows (85%)
Test period has 609 rows (15%)

Recurrent Neural Network (6)
Jordan Network

- *Output* neurons relate to the previous state



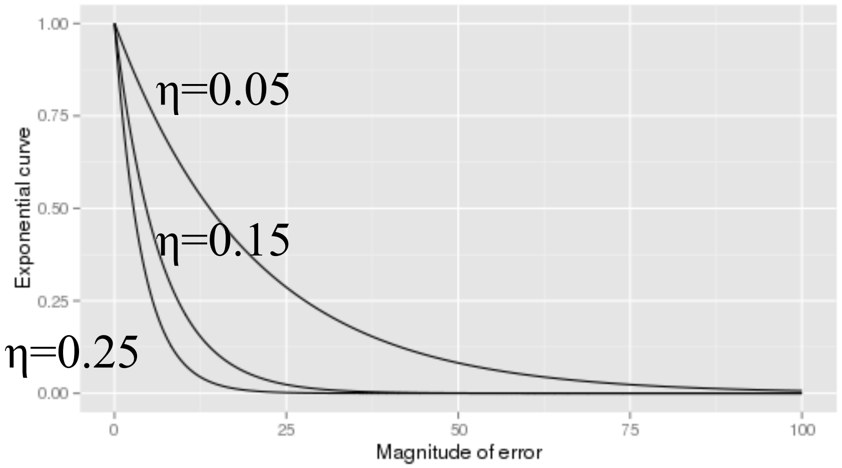
Recurrent Neural Network (7)
Jordan Network



* Training period has 3377 rows (85%)
Test period has 609 rows (15%)

Merging Predictions (1)

- Models can be merged using weighted average
- Weight of each model is penalised according to the amount of error it makes at each step
  - Exponential decay rate "η"
- This implies you can deploy as many models as you like!
  - Aggregating algorithm will select the best for you.



Merging Predictions (2)

- Determine learning rate η
- Initialise equal weights for all models $n = 1,2,3 \dots N$ at time $t = 1$
- FOR $t = 1,2,3 \dots T$
  - FOR model $n = 1,2,3 \dots N$
    - Observe model n's prediction (γ_t^n) at time t
    - Compare γ_t^n with actual outturn (ω_t)
    - Model n suffers error $\lambda(\gamma_t^n, \omega_t)$ at time t (e.g. squared error)
    - Apply exponential factor to the weight at previous time step $t - 1$

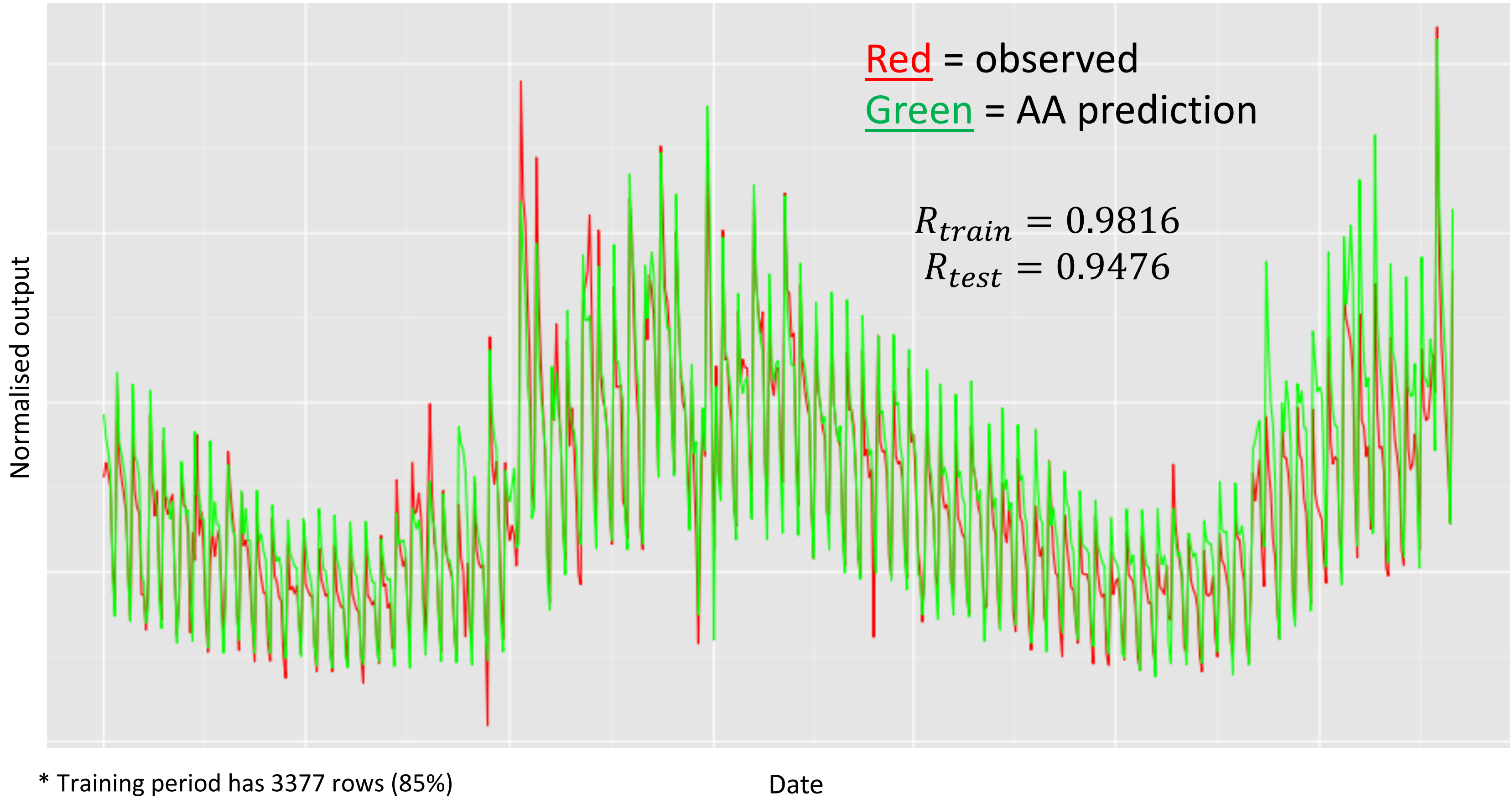
$$w_n^t = w_n^{t-1} \times e^{-\eta\lambda(\gamma_t^n, \omega)}$$

    - Normalise all weights to avoid loss of computer precision

$$\sum_{n=1}^{N} w_n^t = 1$$

  - END FOR
- END FOR

Merging Predictions (3)

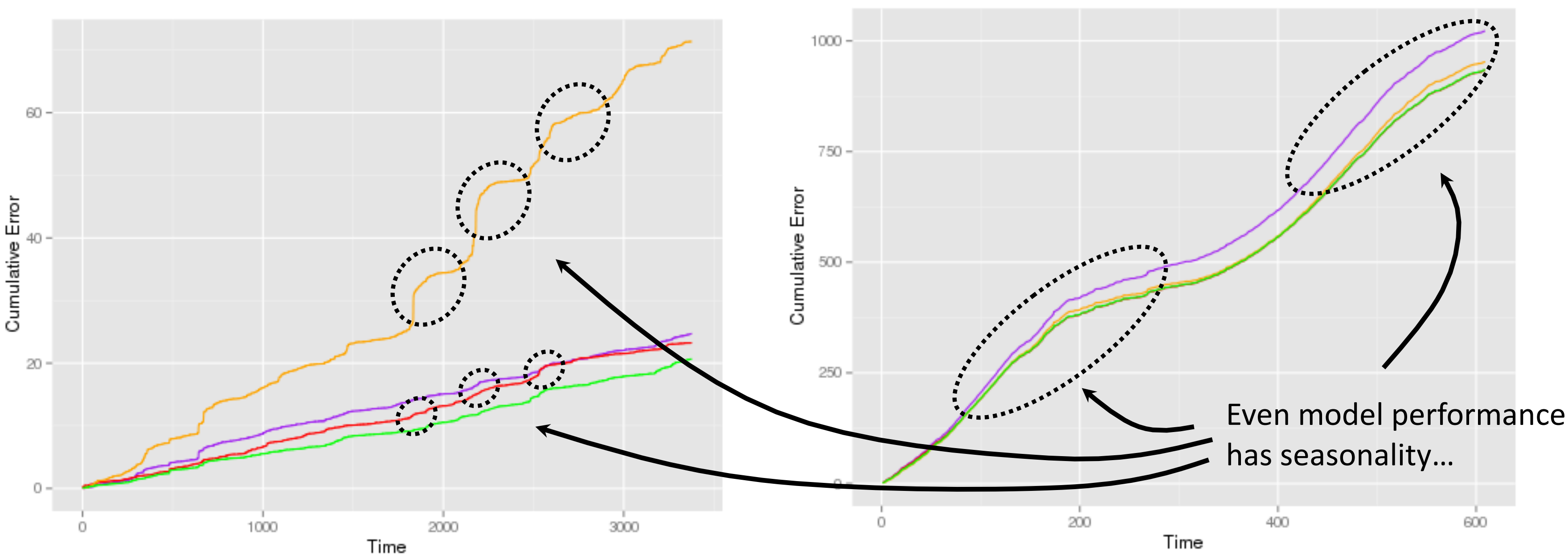


* Training period has 3377 rows (85%)
Test period has 609 rows (15%)

Merging Predictions (4)

Assess models by measuring cumulative error

Training set | *Test set*



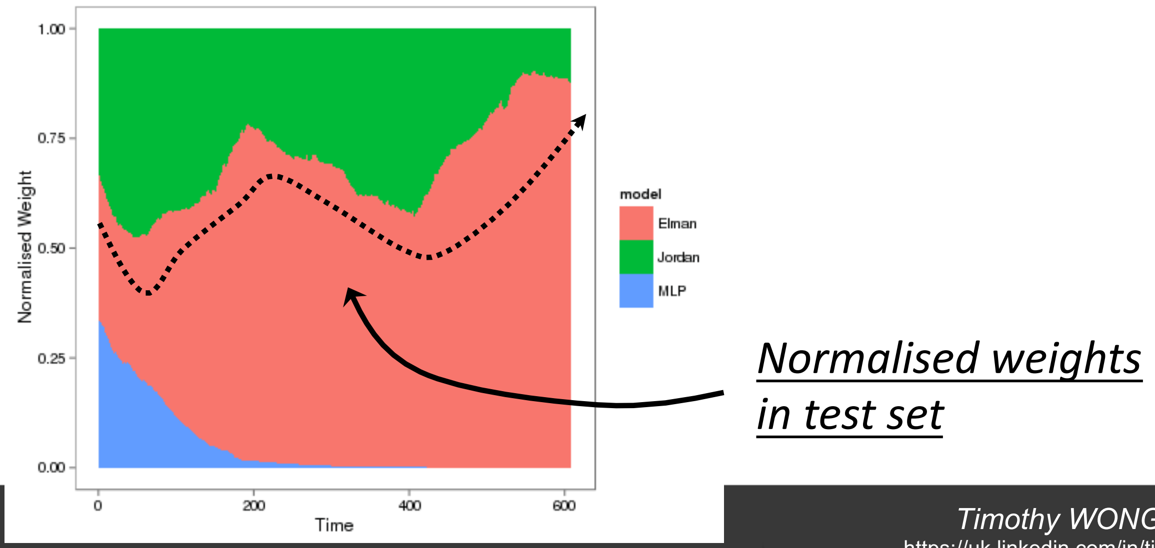
MLP	Elman	Jordan	AA

* Training period has 3377 rows (85%)
Test period has 609 rows (15%)

Merging Predictions (5)

Weights analysis

- No model is always the best model
  - Some model might perform better in particular conditions
    (i.e. Switching is required)
  - Aggregating algorithm does it automatically by weighting all available models based on recent performance



Overall Summary

	Mean Squared Error (MSE)		Pearson's correlation coefficient	
	Training	Test	Training	Test
MLP (30,25,10)	0.0073133	1.67828	0.9787	0.9359
Elman (30,25,10)	0.0068782	1.53291	0.9825	0.9477
Jordan (40)	0.0211193	1.56392	0.9439	0.9245
AA	0.0051921	1.54887	0.9852	0.9466

* Training period has 3377 rows (85%)
Test period has 609 rows (15%)

Possible Improvements

- Minor tweaks
  - More neuron / more iteration
  - Sliding time window

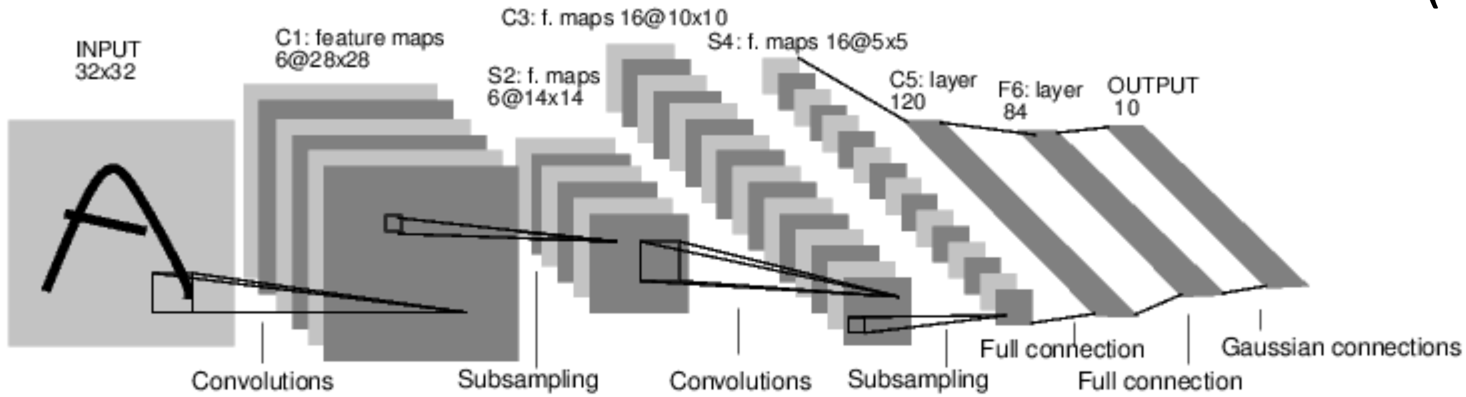
- Use different neural net variants
  - e.g. LSTM

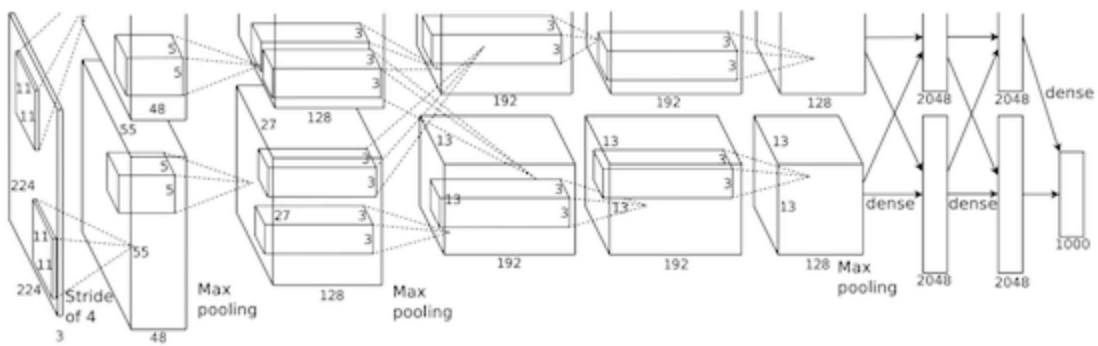
- Deeper model on CUDA-capable GPU device

- Customise topology

Customised Topology (1)

LeNet-5 (1998)

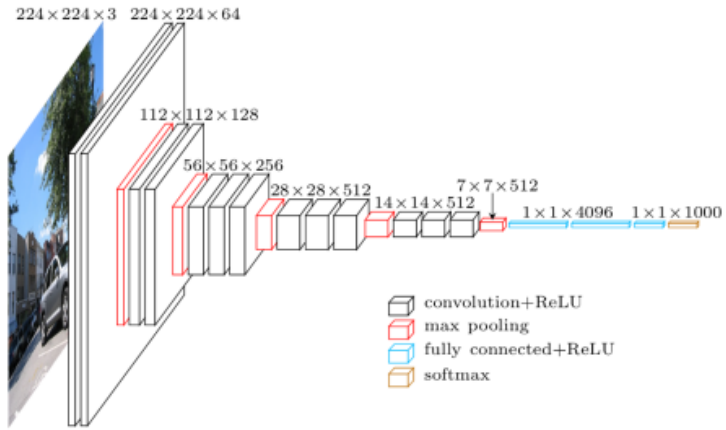




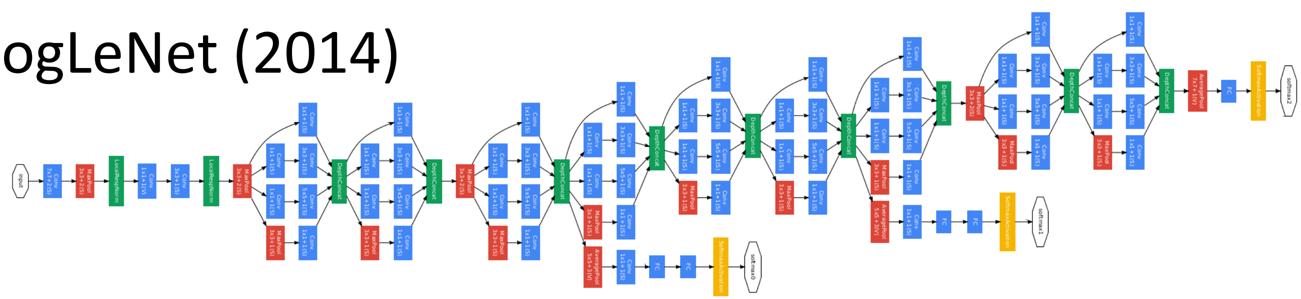
AlexNet (2012)

Customised Topology (2)

VGG-16 (2014)



GoogLeNet (2014)



Alternative Uses

- Predictive modelling
  - Multi-seasonal time series data:
    - Insurance claims
    - Engineer demand
    - Energy consumption
    - Call agent scheduling
    - … etc.

- Classification problem

Useful Resources (1)

- R package: RSNNS
  https://github.com/cbergmeir/RSNNS/tree/master/demo

- Live neural net demo
  http://playground.tensorflow.org

The End - Thanks!

British Gas
Looking after your world

Timothy WONG – Data Scientist
https://uk.linkedin.com/in/timothy-wong-7824ba30