Electrical load curve prediction for non residential customers using Bayesian neural networks

Honorine Royer,
Anne Philippe, Philippe Charpentier, Laurent Bozzi

EDF Lab Saclay & Université de Nantes

AppliBUGS - 17 Décembre 2020

Introduction

- Economic context : Opening of the French electricity market $\Longrightarrow$ Provide **new offers**
- Goal : predicting **full electrical load curves**, at half hourly period, over a year, for **non residential customers**

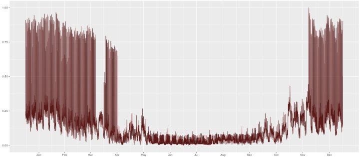


FIGURE – Individual load curve of a non residential customer

- Industrial stakes :
  - Predicted load curve taken from a **catalog** of **existing customers**
  - Correctly predict consumption during **hours of sunlight** $\Longrightarrow$ sizing of the photovoltaic installations.

Description of the dataset

- Two categories : labelled by **contract power**
  - $C4$: between $37kVA$ and $250kVA$
  - $C2$: over $250kVA$
- Data :
  - Individual load curves : consumption time series, over **one year** at **half hourly** period (17472 datapoints)
  - Billing information : mix of **continuous** and **categorical** variables (241 features after transformation)
    e.g. the **peak hours/off-peak hours** consumption **ratios**, the **business activity** (NAF)

Goal of the study

- Goal : predicting **load curves** of $C4$ consumers using only **billing information** (no historical consumption)

- Issue : **Small** $C4$ subset
  - $C2$: 93%
  - $C4$: 7%

- Idea :
  - Benefit from the **similarities on the load profiles** of $C2$ and $C4 \Longrightarrow$ use the $C2$ to **predict** the $C4$
  - All the variables are **standardized** : **Load curves** and **billing information**

Context of the study and notations

- Notations :
  - **Load curves** of length 17472 : X
  - **Customers' information** in dimension 241 : V
- Simple strategy : Multitarget nonlinear regression problem

$$\mathbb{E}(X|V) = g(V),$$

Context of the study and notations

- Issues
  - **Estimation**
    - high dimension
    - multitarget
  - **Prediction** $\hat{\mathbf{X}}_{new} = \hat{g}(\mathbf{V}_{new})$ does not belong to the **catalog** of **existing curves**
- Possible solutions :
  - **Estimation** in high dimension $\Longrightarrow$ **Dimensionality reduction**
  - **Multitarget** regression $\Longrightarrow$ **Deep learning**
  - **Prediction** $\Longrightarrow$ **Search** in a **catalog** of **observed curves**

Distance in the space of the curves

Industrial stake : Predict **accurately** consumption during **hours of sunlight**

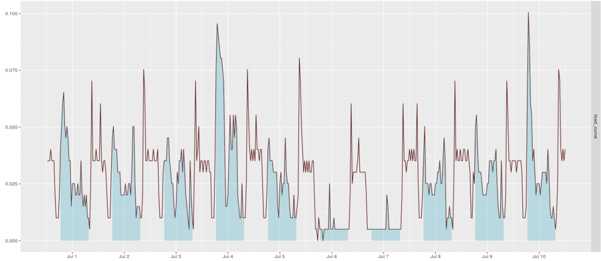


FIGURE – Load curve for one client zoomed over ten days in July, areas highlighted in blue relate to hours of sunlight

Solar power plant production : **power generation** over one year at **half hourly** period **aggregated** and **scaled** => set of **weights** : $(w_i^{sol})_{1 \leq i \leq n}$

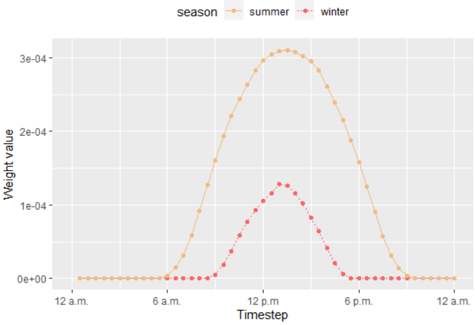


Figure – Solar weights for January 1st (red) versus July 1st (yellow)

Distance in the space of the curves

- Weighted MAE :

  Mean Absolute Error : **adapted** to give more **importance** to **periods** of **higher solar intensity weights** into account

$$\mathcal{E}_{sol}(Y, \hat{Y}) = \sum_{i=1}^{n} |Y_i - \hat{Y}_i| \times w_i^{sol}, \tag{1}$$

  where Y and $\hat{Y}$ are respectively a **load curve** and its **prediction**.

- Applicability of $\mathcal{E}_{sol}$:
  - **Loss function** for **training** models
  - **Optimization of the predictions** : **construction** of the **prediction** and **evaluation error**

Neural Networks

Neural networks : successive **layers** made of **nonlinear transformations**

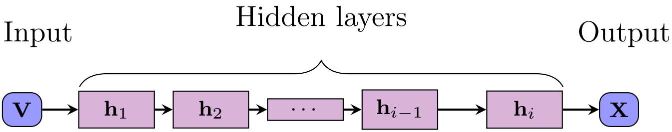


FIGURE – Diagram of a feedforward neural network with i hidden layers

Neural Networks

- The i **hidden layers feedforward neural network** :

$$\begin{aligned}\mathbf{h}_1 &= \sigma_1(\mathbf{W}_1^T \cdot V + \mathbf{b}_1),\\ &\vdots\\ \mathbf{h}_i &= \sigma_i(\mathbf{W}_i^T \cdot \mathbf{h}_{i-1} + \mathbf{b}_i),\\ X &= \sigma_{i+1}(\mathbf{W}_{i+1}^T \cdot \mathbf{h}_i + \mathbf{b}_{i+1}),\end{aligned}$$

Parameters : **weights** $\mathbf{W}_k$, **biaises** $\mathbf{b}_k,\ 1 \leq k \leq i+1,$

- Activation functions σ_k :**Rectified Linear Unit** (ReLU)

$$\sigma_k(x) = \max(0, x),\ \forall\ x \in \mathbb{R},\ 1 \leq k \leq i+1$$

Neural Networks

More sophisticated neural networks : **residual connections** designed to avoid the **vanishing gradient** problem.

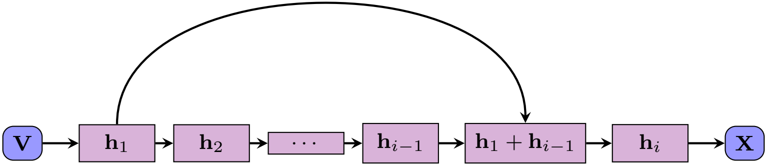


FIGURE – Diagram of a neural network with a **residual connection** between the outputs of the first layer and of the $i-1$eth hidden layer

Neural networks : Goodfellow et al. [2016], Krizhevsky et al. [2012], Graves et al. [2013]
Residual connections : He [2017]

Autoencoders : a particular case of neural networks
Input and ouput : X

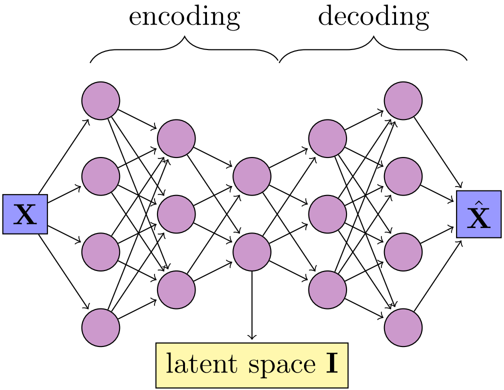


FIGURE – $\mathbf{X}$ input, $\hat{\mathbf{X}}$ reconstruction, $\mathbf{I}$ reduced dimension

Example of application : **Image compression**

Autoencoders : a particular case of neural networks

- **Inputs** and **Outputs** : X
  Reduced representation : I

$$
\begin{aligned}
I &= e(X), \\
X &= d(I)
\end{aligned}
$$

$$X \xrightarrow{e} I \xrightarrow{d} X$$

- Two parts :
  - encoding e : **dimensionality reduction**
  - decoding d : **reconstruction** of the input

Bayesian deep learning : Bayesian analysis applied to deep neural networks

- **Inputs** : V
  Outputs : I

- Prior distribution on the parameters :
  - Difficulty to incorporate **prior knowledge**
  - **Weights W** i.i.d. $\mathcal{N}(0, 1)$

- Posterior distribution :

$$p(\mathbf{W}|I, V) = \frac{\exp(-\frac{1}{2}\mathbf{W}^T\mathbf{W})p(I|V, \mathbf{W})}{p(I|V)}$$

- Issue :
  - $p(I|V)$ not **explicit**
  - **scalability** of **MCMC**

Bayesian deep learning : Inference

Variational Inference : **Approximation** of the **posterior**

- $\mathcal{Q}$ a **family of distribution** e.g. **Gaussian** or a product of **Gaussian distributions**

- Criterion : find $q^\star \in \mathcal{Q}$ an **approximation** of the **posterior**

$$q^\star(\mathbf{W}) = \underset{q \in \mathcal{Q}}{\operatorname{argmin}} \, K(q(\mathbf{W}), \; p(\mathbf{W}|I, V)).$$

- Equivalent to maximizing the Evidence Lower Bound :

$$L(\mathbf{W}) = \int q(\mathbf{W}) \log(p(I|V, \mathbf{W})) \, d\mathbf{W} - \int q(\mathbf{W}) \log(\frac{q(\mathbf{W})}{p(\mathbf{W})}) \, d\mathbf{W}.$$

Bayesian neural networks : Neal [1996], Kingma and Welling [2014], Gal [2016], Wen et al. [2018]
Variational inference : Blei et al. [2017]

Prediction of a new customer's load curve : first method

Three strategies for modelling : **two ways** to **predict** the load curve

First method :

- Forecasting : $\hat{\mathbf{X}}_{new}$, a **predicted load curve** of $\mathbf{X}_{new}$ is available
- Search of the nearest neighbor : in the **catalog** of **existing** curves $\mathcal{X} = (\mathbf{X}_k)_{1 \leq k \leq m}$

$$\hat{k}_X = \underset{1 \leq k \leq m}{\operatorname{argmin}} \mathcal{E}_{sol}(\mathbf{X}_k, \hat{\mathbf{X}}_{new}). \tag{2}$$

- Correction of the prediction : $\mathbf{X}_{\hat{k}_X}$ **predicted load curve** of $\mathbf{X}_{new}$

Prediction of a new customer's load curve : second method

Second method :

- Forecasting : from **dimensionality reduction**, $\hat{\mathbf{I}}_{new}$ a **reduced representation** of $\mathbf{X}_{new}$ is available

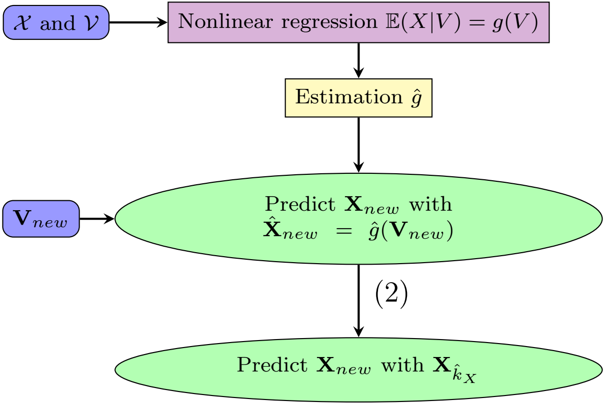
- Construction of the catalog of reduced curves : $\mathcal{I} = (\mathbf{I}_k)_{1 \leq k \leq m}$ from **reducing dimension** on $\mathcal{X} = (\mathbf{X}_k)_{1 \leq k \leq m}$

- Search of the nearest neighbor :

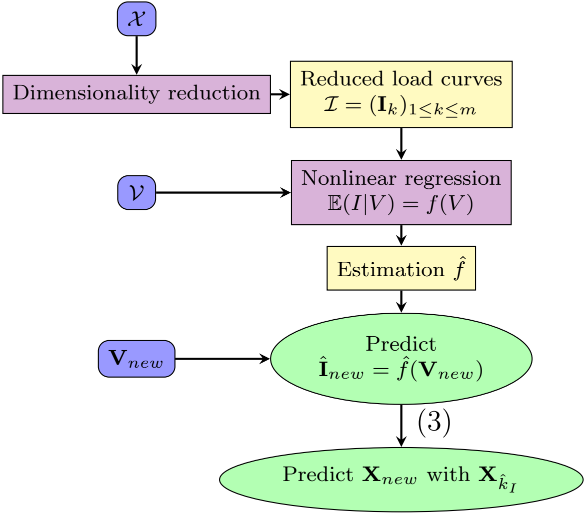
$$\hat{k}_I = \underset{1 \leq k \leq m}{\operatorname{argmin}} \, \mathcal{E}_{MAE}(\mathbf{I}_k, \hat{\mathbf{I}}_{new}). \tag{3}$$

- Correction of the prediction : $\mathbf{X}_{\hat{k}_I}$ **predicted load curve** of $\mathbf{X}_{new}$

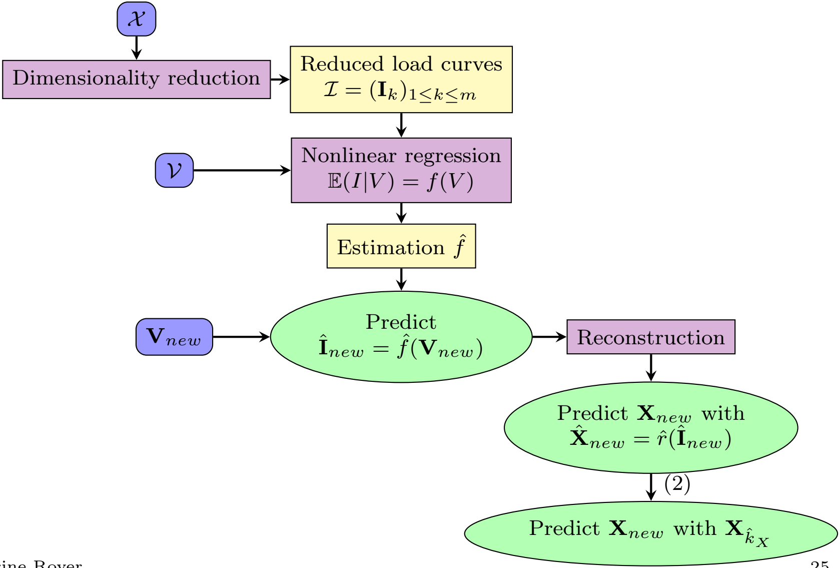
Multitarget nonlinear regression in high dimension (MNR)



Encoding and nonlinear regression (ENR)



Encoding, nonlinear regression and reconstruction (ENR-R)



Fine Tuning - A special case of Transfer learning

- Transfer Learning : using **knowledge** from one task and **exploiting** it to **solve another task**

- Fine tuning : special case of **transfer learning**
  - Two dataset and tasks : sharing some **similarities**
  - Pre-training a model : on the **first dataset** to learn the **first task**
  - Fine tune the model : **re-train** the pred-trained model (or some parts) on the **second dataset** to learn the **second task**

Fine Tuning - A special case of Transfer learning

Why use fine tuning ?

- **Lack of $C4$ observations**
- **Similarities** between the $C2$ and $C4$ customers
- **Improve performances** of the model on the **second task**

Transfer learning : Pan and Yang [2009], Torrey and Shavlik [2010]
Fine Tuning : Hinton and Salakhutdinov [2006]

Dimensionality reduction

TABLE – $\mathcal{E}_{sol}(\mathbf{X}_{new}, \hat{\mathbf{X}}_{new})$ various autoencoders and the discrete wavelet transform on the $C4$ testing subset.

$C4$ testing subset		
	Median	Mean
Autoencoder trained with $\mathcal{E}_{sol}$, **without fine tuning**	0.239	0.258
Autoencoder trained with $\mathcal{E}_{sol}$, **with fine tuning**	0.223*	0.248*
Autoencoder trained with $\mathcal{E}_{MAE}$	0.282	0.304
Wavelets	0.534	0.631

Dimensionality reduction - Reconstruction using the autoencoder

Figure – [Top] Original load curve of a $C4$ customer. [Bottom] Reconstruction with the autoencoder.

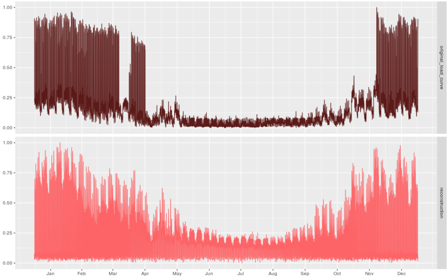
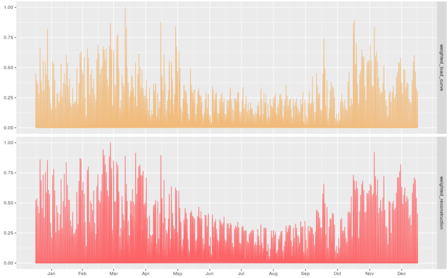


Figure – [Top] Weighted load curve of the customer. [Bottom] Weighted reconstruction with the autoencoder.



Multitarget non linear regression - MNR

Estimation of g : $\mathbf{NN}_i\mathbf{MNR}$, $i \in \{1, 2, 4, 6\}$ (hidden layers)

FIGURE – Simplified outline of the MNR framework.

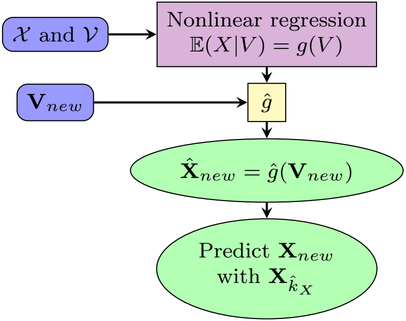
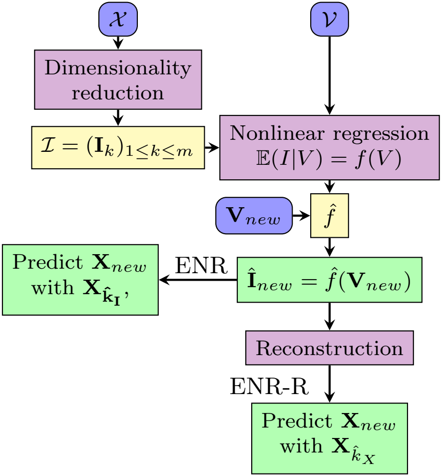


TABLE – $\mathcal{E}_{sol}(\mathbf{X}_{new}, \mathbf{X}_{\hat{k}_X})$ for the MNR scheme on the $C4$ testing subset

Without fine tuning		
	Median	Mean
$\mathbf{NN}_1\mathbf{MNR}$	1.642	1.644
$\mathbf{NN}_2\mathbf{MNR}$	1.700	1.654
$\mathbf{NN}_4\mathbf{MNR}$	1.532*	1.465*
$\mathbf{NN}_6\mathbf{MNR}$	1.542	1.476
With fine tuning		
	Median	Mean
$\mathbf{NN}_1\mathbf{MNR}$	0.553	0.722
$\mathbf{NN}_2\mathbf{MNR}$	0.609	0.702
$\mathbf{NN}_4\mathbf{MNR}$	0.668	1.101
$\mathbf{NN}_6\mathbf{MNR}$	0.547*	0.648*

Comparison of the two methods of prediction in the ENR-R and ENR strategies



Estimation of f :

- Neural Networks :
  - $\mathbf{NN}_1$ and $\mathbf{RN}_1$
- Bayesian models :
  - **Bayesian neural networks** : **BayesNN**$_1$ and **BayesRN**$_1$
  - **Deep Gaussian processes** : **DGP**$_2$

Encoding, nonlinear regression and reconstruction - ENR-R

TABLE – $\mathcal{E}_{sol}(\mathbf{X}_{new}, \mathbf{X}_{\hat{k}_X})$ obtained with the models tested for the ENR-R scheme on the $C4$ testing subset.

Dimensionality reduction method : Autoencoder				
	Without fine tuning		With fine tuning	
	Median	Mean	Median	Mean
$\mathbf{NN}_1$	0.755	0.847	0.570*	0.623*
$\mathbf{RN}_1$	0.792	0.896	0.575	0.754
$\mathbf{BayesNN}_1$	1.997	1.656	0.633	0.682
$\mathbf{BayesRN}_1$	0.751	0.943	0.611	0.652
$\mathbf{DGP}_2$	0.685*	0.842*	0.894	0.915

Encoding and nonlinear regression - ENR

TABLE – $\mathcal{E}_{sol}(\mathbf{X}_{new}, \mathbf{X}_{\hat{k}_I})$ obtained with the models tested for the ENR scheme on the $C4$ testing subset.

Dimensionality reduction method : Autoencoder				
	Without fine tuning		With fine tuning	
	Median	Mean	Median	Mean
$\mathbf{NN}_1$	0.422	0.456	0.491	0.539
$\mathbf{RN}_1$	0.427	0.465	0.502	0.560
$\mathbf{BayesNN}_1$	0.431	0.462	0.466*	0.499*
$\mathbf{BayesRN}_1$	0.409*	0.451*	0.503	0.559
$\mathbf{DGP}_2$	0.451	0.490	0.984	1.004

Reconstruction error $\mathcal{E}_{sol}(\mathbf{X}_{new}, \hat{\mathbf{X}}_{new})$

TABLE – Solar MAE $\mathcal{E}_{sol}(\mathbf{X}_{new}, \hat{\mathbf{X}}_{new})$ obtained with the models tested for the ENR-R scheme on the $C4$ testing subset.

Dimensionality reduction method : Autoencoder				
	Without fine tuning		With fine tuning	
	Median	Mean	Median	Mean
NN$_1$	0.624	0.687	0.504	0.547
RN$_1$	0.670	0.723	0.467	0.527
BayesNN$_1$	0.519*	0.583*	0.493	0.552
BayesRN$_1$	0.572	0.639	0.464*	0.526*
DGP$_2$	0.540	0.590	0.760	0.812

Comparison of the ENR-R and ENR strategies

- Fine tuning :
  - ENR-R strategy : all models are **improved** with **fine tuning**
  - ENR strategy : **fine tuning deteriorates** all the performances
  - Possible explanation : **reconstruction** with the **autoencoder with fine tuning** $\Longrightarrow$ **offsets** the **deterioration** of the performances
- Deep Gaussian Processes : **longer to train** than the other models $\Longrightarrow$ complicates potential **production phase**

Comparison of the ENR-R and ENR strategies

Prediction :

- ENR-R : overall **high errors** $\Longrightarrow$ not the **best prediction strategy** to search for the **nearest neighbor** over the **entire curve**
- ENR : **lower errors** obtained with the **autoencoder** for **dimensionality reduction**
- Reconstruction : **not real** load curves $\Longrightarrow$ **lower error** rates

Comparison of the ENR-R and ENR strategies

Bayesian neural networks :

- Lowest errors alternatively with $\mathbf{BayesRN_1}$ or $\mathbf{BayesNN_1}$ depending on the **strategy**
- $\mathbf{BayesRN_1}$: **ENR strategy** $\Longrightarrow$ **lowest error** without fine tuning : 0.409
- Posterior predictive distribution : possibility of obtaining **prediction intervals**

Prediction intervals using the Bayesian posterior predictive distribution

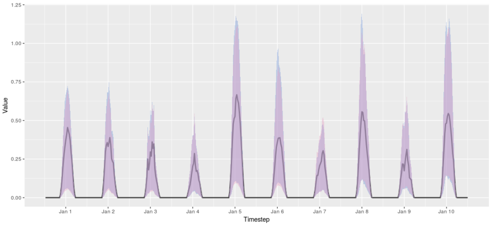
Two possibilities :

Searching for nearest neighbor for each sample :

$$\hat{k}_{I_j} = \underset{1 \leq k \leq m}{\operatorname{argmin}} \, \mathcal{E}_{MAE}(\mathbf{I}_k, \hat{\mathbf{I}}^{pos}_{new_j}), \ \forall j \in 1, \ldots, J$$

Discrete distribution on the curves $\Longrightarrow$ **Quantiles** for each time step of the curve

FIGURE – Weighted load curve of one $C4$ customer (black) for ten days and prediction intervals



Prediction intervals obtained by decoding samples

Decoding samples :

$$\hat{\mathbf{X}}^{pos}_{new_j} = \hat{d}(\hat{\mathbf{I}}^{pos}_{new_j}),\ \forall j \in 1, \ldots, J,$$

Ensemble of **reconstructed** load curves $\Longrightarrow$ **Quantiles** for each time step of the curve

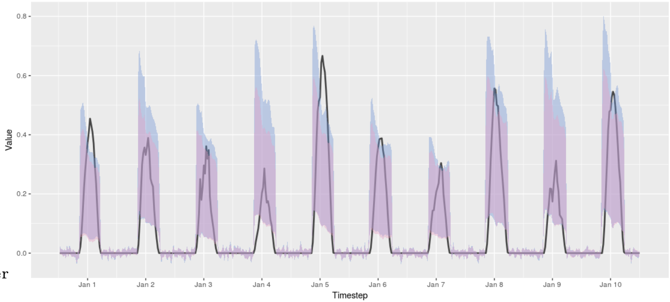


FIGURE – Weighted load curve of one $C4$ customer (black) for ten days and prediction intervals

Conclusion

- Transfer learning :
  - NN_i models' performances in the MNR strategy are **improved** $\Longrightarrow$ **not evenly**
  - **improves** the performances in the **ENR-R strategy** but **deteriorates** them in the **ENR strategy**
- Prediction intervals :
  - **Two possibilities** to obtain intervals from the **posterior predictive distribution**
  - **First possibility** : intervals follow the **shape of the curve** better, but are **larger** and **sometimes imprecise**

References

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, 2016. http://www.deeplearningbook.org.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In 2013 IEEE international conference on acoustics, speech and signal processing, pages 6645–6649. IEEE, 2013.

Wan He. Load forecasting via deep neural networks. Procedia Computer Science, 122 :308 – 314, 2017. ISSN 1877-0509. doi : https://doi.org/10.1016/j.procs.2017.11.374. URL http://www.sciencedirect.com/science/article/pii/S1877050917326170. 5th International Conference on Information Technology and Quantitative Management, ITQM 2017.

References

Radford M Neal. Bayesian learning for neural networks. PhD thesis, 1996.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In 2nd International Conference on Learning Representations, 2014.

Yarin Gal. Uncertainty in deep learning. University of Cambridge, 1(3), 2016.

Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger B. Grosse. Flipout : Efficient pseudo-independent weight perturbations on mini-batches. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018. URL https://openreview.net/forum?id=rJNpifWAb.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference : A review for statisticians. Journal of the American statistical Association, 112(518) :859–877, 2017.

References

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10) :1345–1359, 2009.

Lisa Torrey and Jude Shavlik. Transfer learning. In Handbook of research on machine learning applications and trends : algorithms, methods, and techniques, pages 242–264. IGI Global, 2010.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. science, 313(5786) : 504–507, 2006.