Air-quality forecasting in Belgium using Deep Neural Networks, Neuroevolution and distributed Island Transpeciation

Thesis Presentation

K. Theodorakos[1]

[1]Master of Artificial Intelligence: Engineering and Computer Science

6th August 2019

KU LEUVEN

Background

Ozone

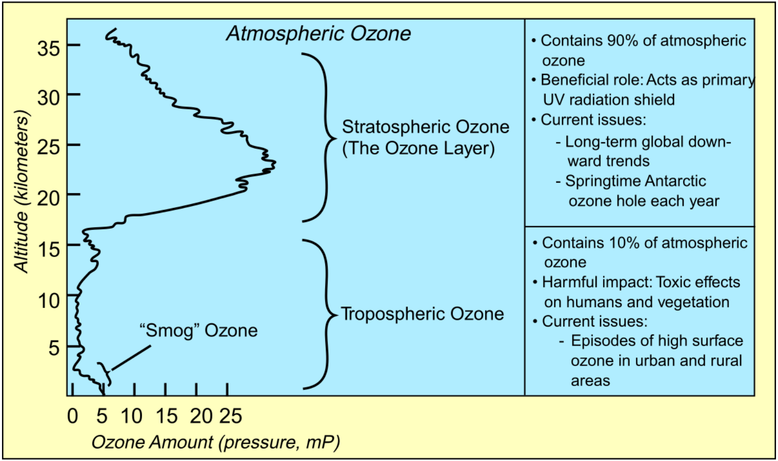


Figure 1: Atmospheric ozone distribution [Pro16]. Concentration levels (BE): **Background**: 60 $\mu g/m^3$, **healthy limit**: 120 $\mu g/m^3$, **public informing (>1 station)**: **180** $\mu g/m^3$, **alert**: **240** $\mu g/m^3$ [Eur02].

Particulate Matter

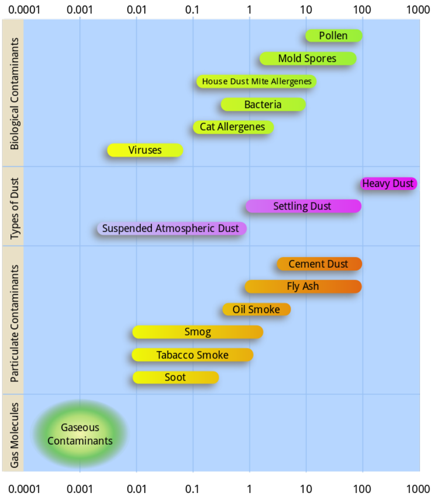
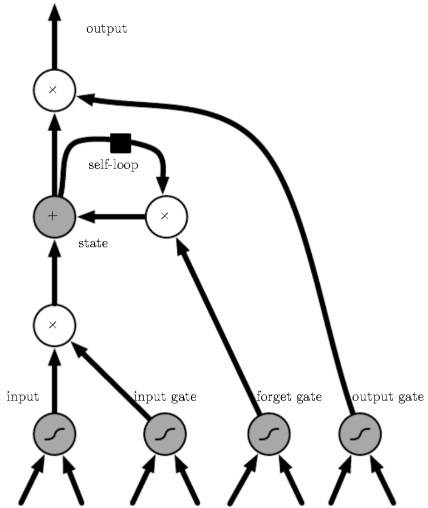


Figure 2: Diameter-size distribution of airborne particles in μm [Nie09].

Long Short Term Memory (LSTM) cell [HS97, IBC16]

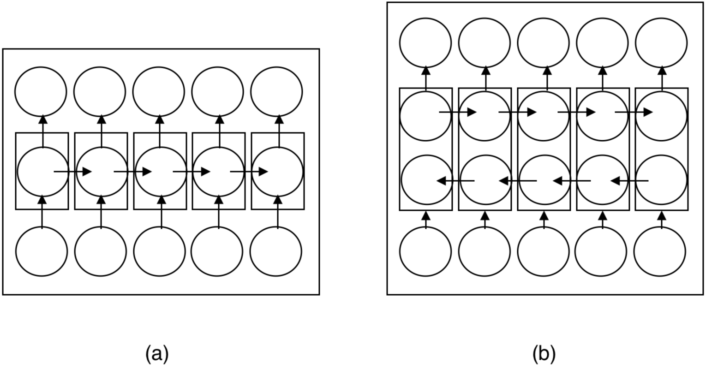
Input unit: any squashing nonlinearity. Gated units: sigmoid nonlinearity.



- **Input**: Output of regular neuron (from input feature).
- **Input gate**: Regulates *input* accumulation into *state*.
- **State**: Has linear *self-loop*.
- **Self-loop**: Single time-step delay.
- **Forget gate**: Controls *self-loop* weight.
- **Output gate**: Regulates *output*.

Bidirectional Recurrent Neural Networks (BRNN) [SP97]

Causality from forward and backward time (state) directions



Structure overview

(a) unidirectional RNN
(b) bidirectional RNN

Naive LSTM model

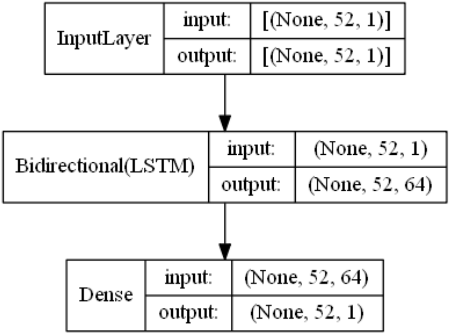
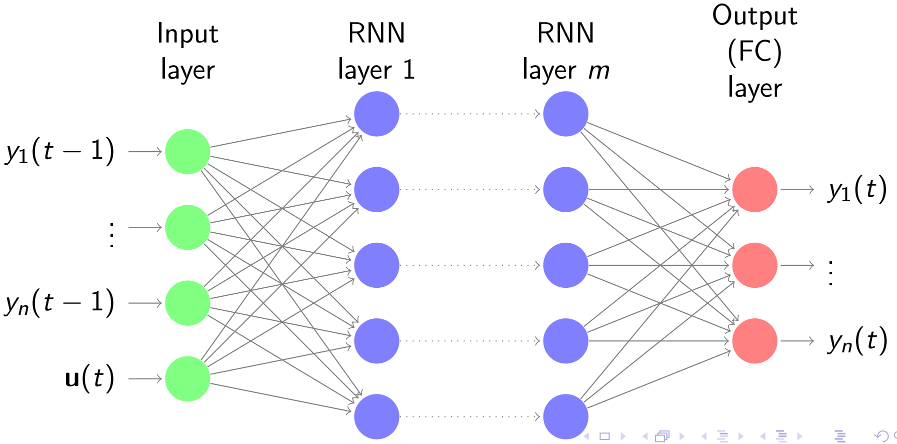


Figure 3: A naive LSTM model of 52 input variables: 1x bidirectional LSTM layer with 64 units and 1x Fully Connected (dense) layer giving 1 output.

Multiple-Input Multiple-Output (MIMO)

MIMO Nonlinear AutoRegressive eXogenous (NARX) model:

$y_1(t), ..., y_n(t) = f(y_1(t-1), ..., y_n(t-1), \mathbf{u}(t))$
where $\mathbf{t}$: time-step, $\mathbf{u}$: exogenous variables vector, $\mathbf{y_1}, ..., \mathbf{y_n}$: time-series



Evolutionary Algorithms (EA): Components

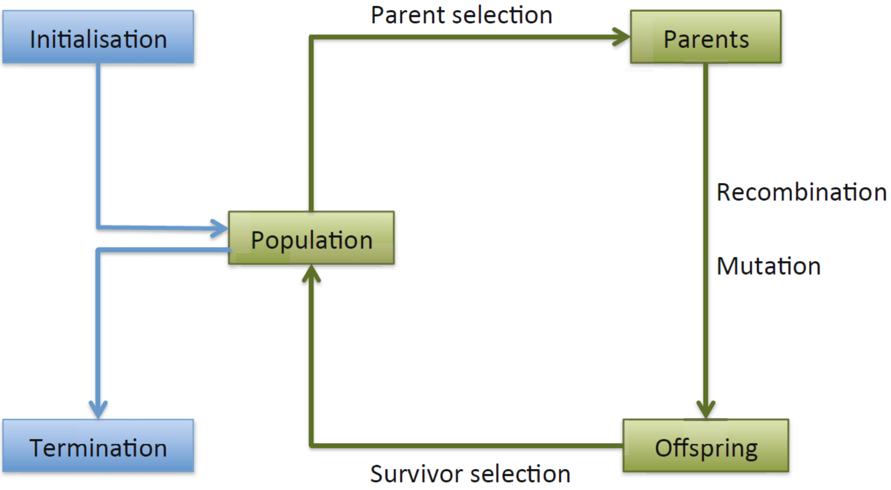


Figure 4: Evolutionary algorithm (usually stochastic) components [ES15].

Evolutionary Algorithms (EA): Representation spaces

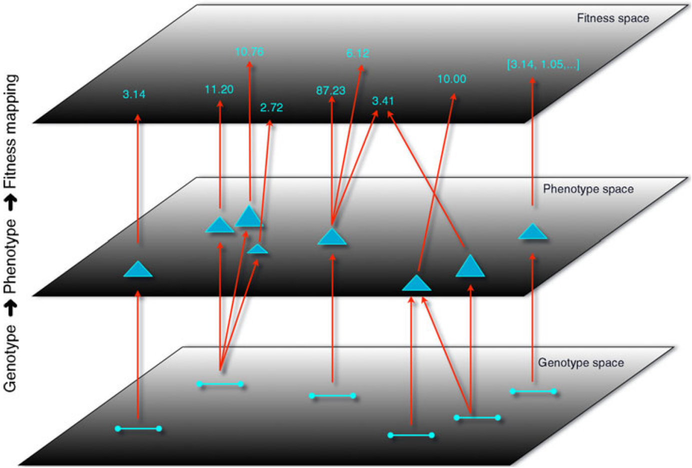
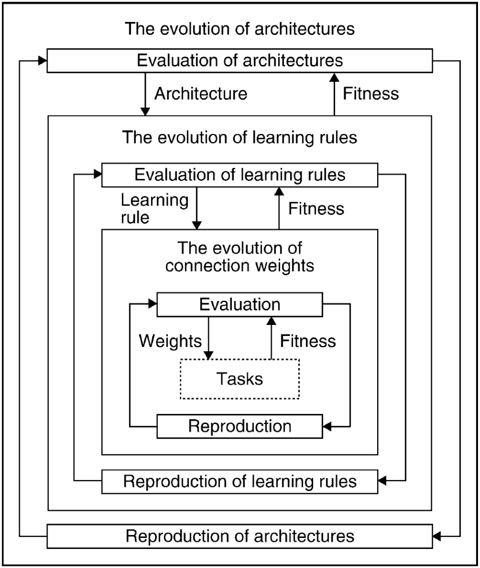


Figure 5: EA individual representation spaces: genotype, phenotype and fitness [RBK12].

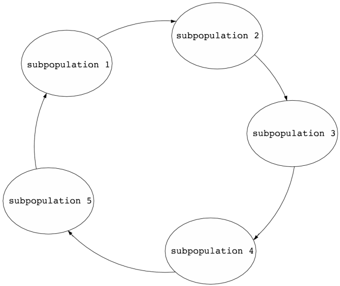
Neural Architecture Search

Neuroevolution [Yao99, Kov12, RMS+17, MLM+19]

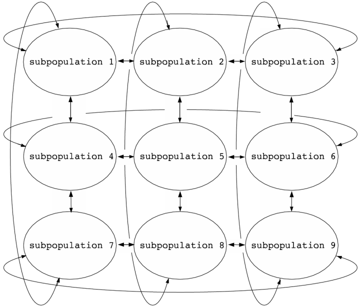


Method

Island model



(a) Ring (1 dimension)



(b) Toroidal 2D grid (Von Neumann neighborhood)

Figure 6: Island model topologies [Tom05] for periodic migration of individual solutions between subpopulations.

Neuroevolution: Island species

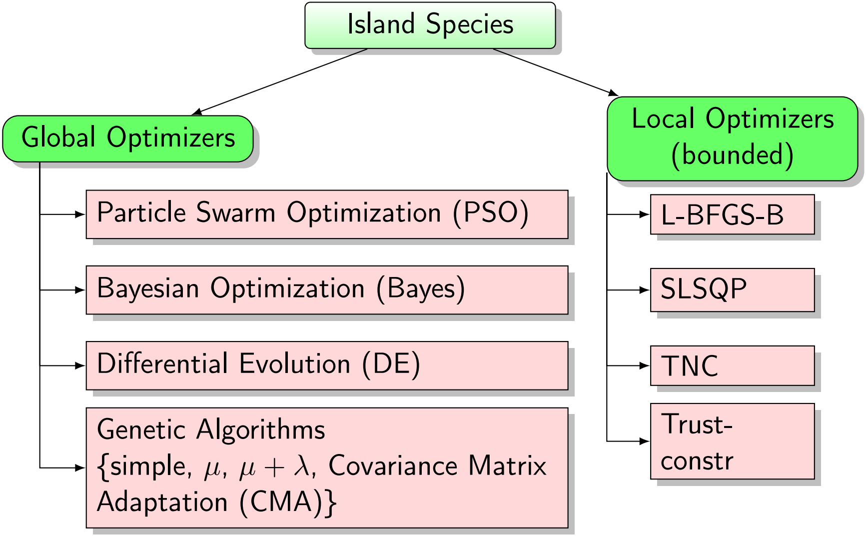
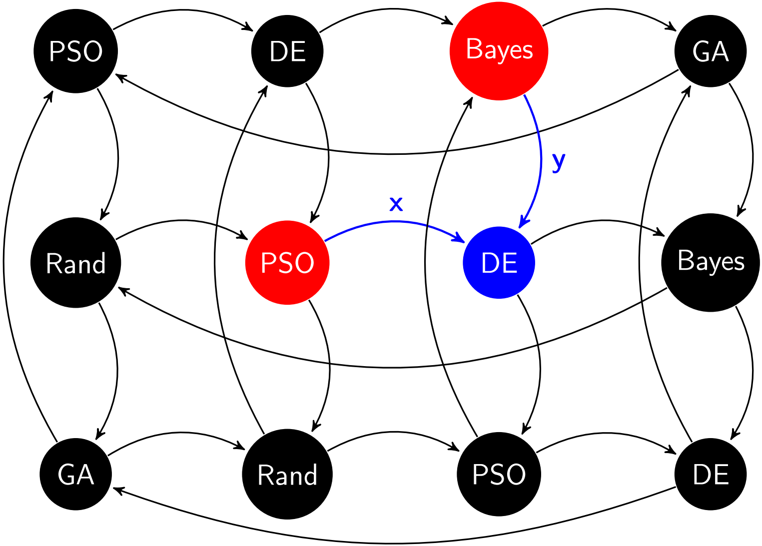


Figure 7: Island species for global and local search.

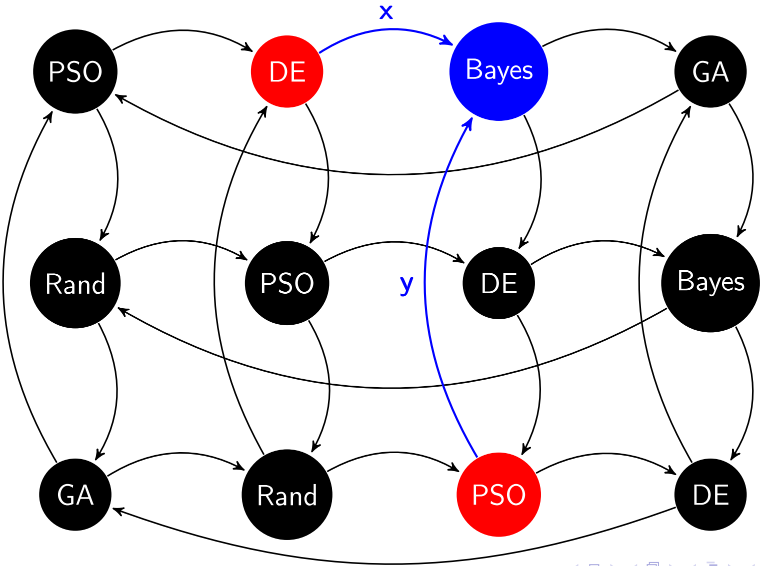
nD Cellular Automata (2D example)

Single-direction Von Neumann neighborhood (radius 1)



nD Cellular Automata (2D example)

Single-direction Von Neumann neighborhood (radius 1)



nD Cellular Automata (4D example)

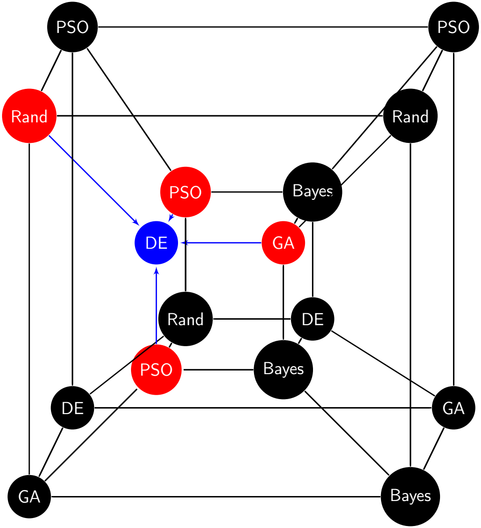


Figure 8: 4D toroidal grid mesh island neighborhood. Inspiration: torus interconnection networks [ABC+10].

Ranking based neighbor selection

nD grid yields μ neighbors:

- Which migrating candidate to choose?

Linear Ranking (LR) selection [BH91, Whi89]:

1. Rank μ neighbors by fitness (worst rank: $i = 0$).
2. Adjust selection pressure $s \in (1, 2]$:
   - Small s: $\approx$ uniform random.
   - Large s: Best ranks have higher selection probability.
3. Random choice given probabilities: $P_{LinearRank}(i) = \frac{(2-s)}{\mu} + \frac{2i(s-1)}{\mu(\mu-1)}$

Finally: Replace island's worst candidate (within "internal representation" level).

Neuroevolution: Island Transpeciation

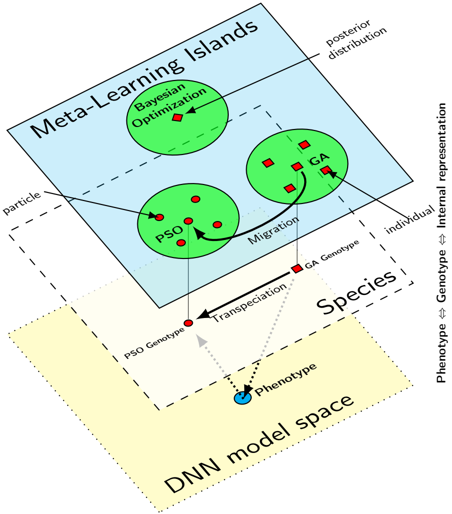


Figure 9: *Genotype* ⇔ *internal representation* (lossy): parallelism and co-evolution (cooperative & competitive) of diverse global optimizers.

Neuroevolution: Architecture genes

Sequential model search of 4 base + auxiliary layers

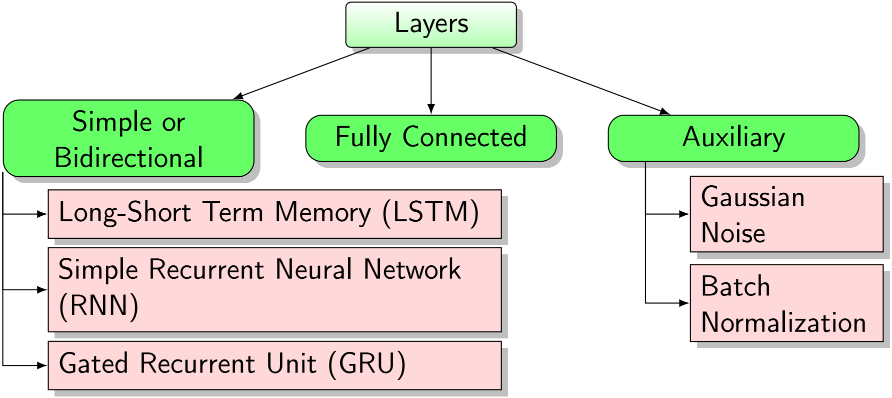


Figure 10: Architecture search [IS15, KUMH17].

Neuroevolution: Hyperparameter genes

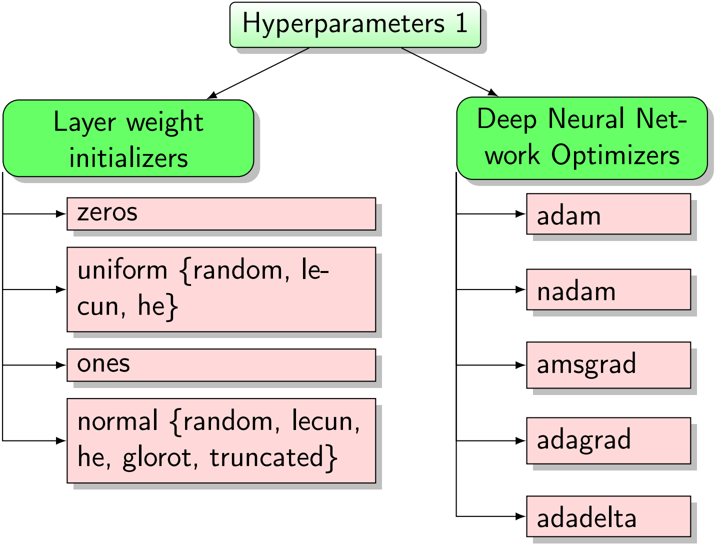


Figure 11: Optimizers [DJS11, Zei12, KB14, RKK18, SMDH13].

Neuroevolution: Hyperparameter genes 2

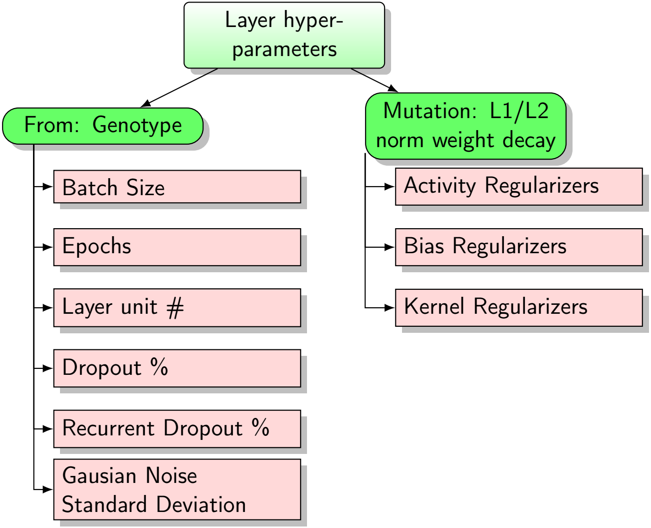


Figure 12: Hyperparameters 2 [SHKS14].

Data & Training I

Data

- Particulate Matter 10 μm: Hourly concentrations $\Rightarrow$ Daily mean.
- Ozone: Hourly concentrations $\Rightarrow$ Daily max of 8-hour rolling mean.

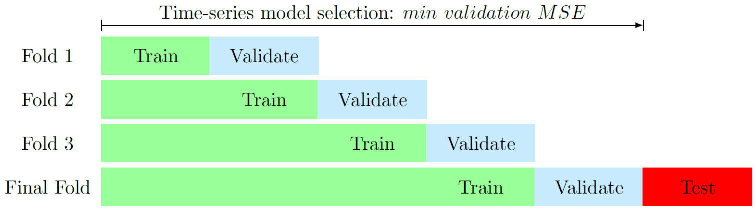
Processing:

1. Missing data: Fill from the most correlated stations.
2. Linear interpolation (forward/backward fill).
3. Post-processing: Feature scaling (standardization).

Data & Training II

Training

- Time-series Cross-Validation [Ber15].
- Max validation fold length: 365 days.
- Learning Rate reduction on plateau.
- Early Stopping (over-fitting avoidance).



Implementation: CPU and GPU Parallelism

CPU

- **Blocking Master-Slave** pattern (agent migration).
- Message Passing Interface (MPI): One island per process.
- Centralized island-to-island individual migrations.

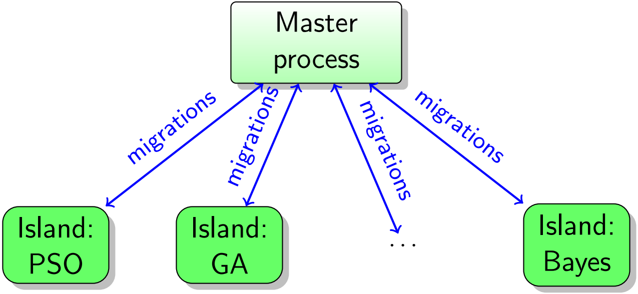
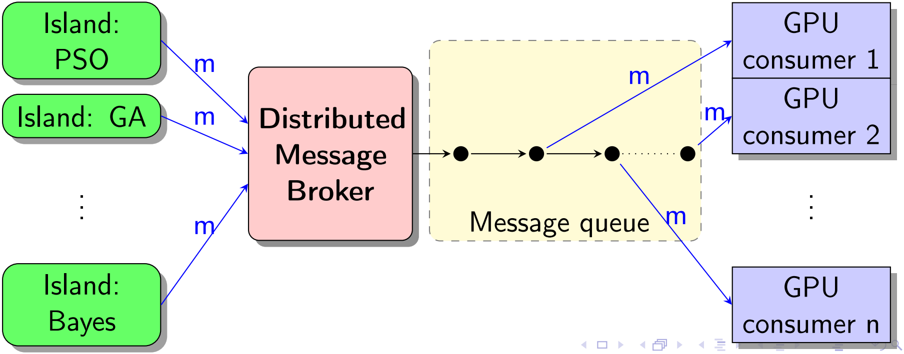


Figure 13: Master-slave pattern.

Implementation: CPU and GPU Parallelism

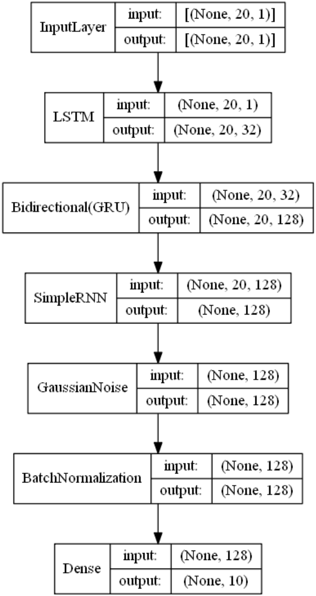
GPU

- **Asynchronous** Competing Consumers (model training).
- Distributed Message Broker (Rabbit MQ).
- "Hot-plug", fault-tolerant queue.
- Networked GPU workers, of hybrid architecture & performance.



Results

Island Transpeciation evolved MIMO DNN example:



Island Transpeciation vs Random Search

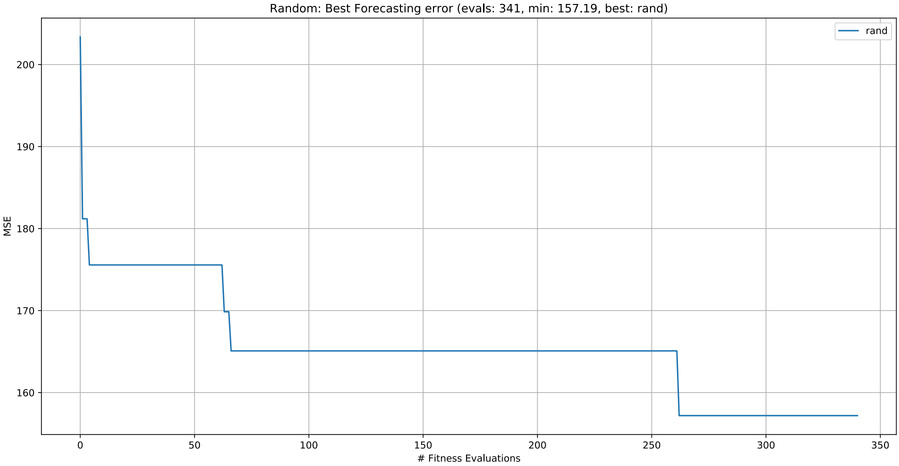


Figure 15: Neural architecture search: **Random uniform.**

Island Transpeciation vs Random Search

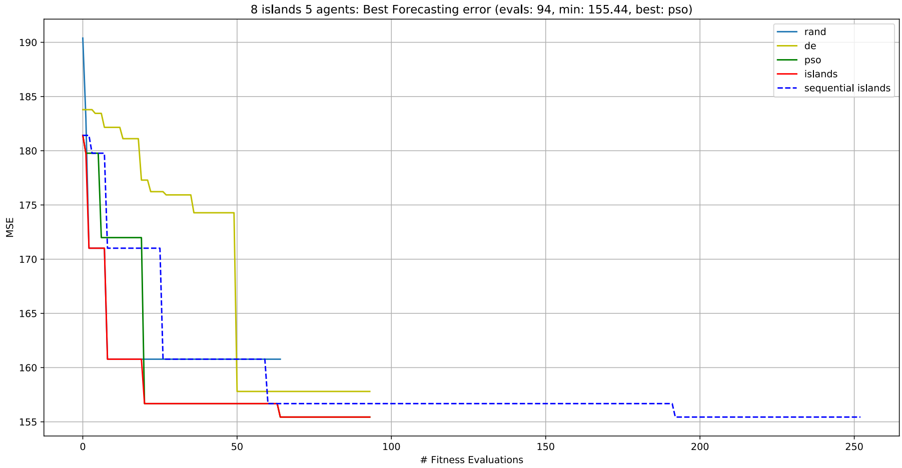
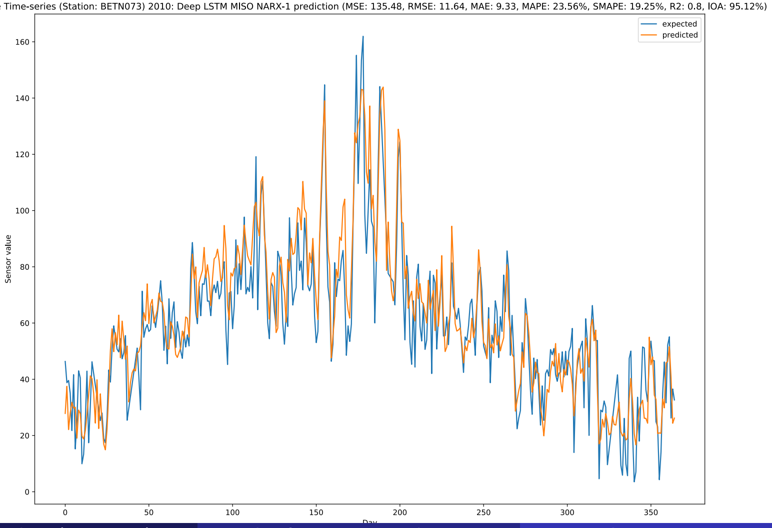


Figure 16: Neural architecture search: **Island transpeciation** (3x DE, 3x PSO and 2x Rand search islands, with 5 individuals each). **Error reduction: up to** $\approx$ **3.82%** vs naive/random LSTM models [The19].

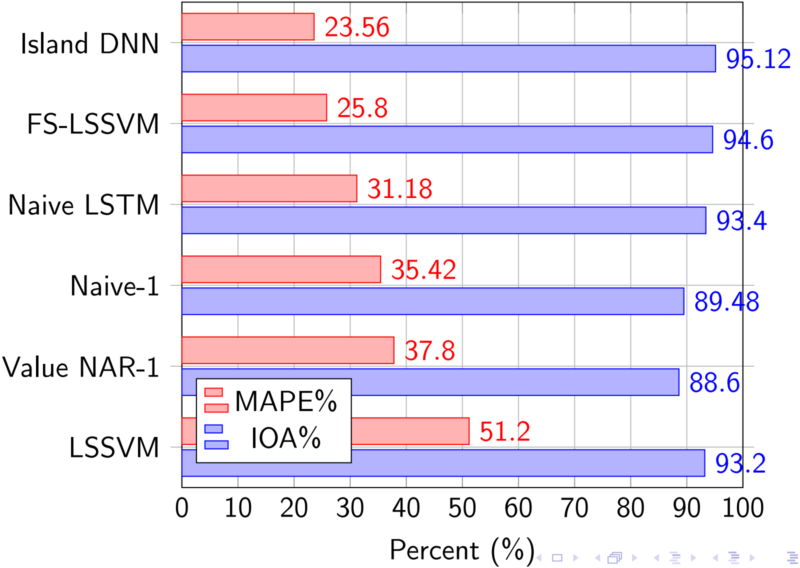
Ozone: Multiple-Input Single-Output (MISO) NARX

train: 2000-2009, test: 2010, data: 51x weather variables [DUS+11]



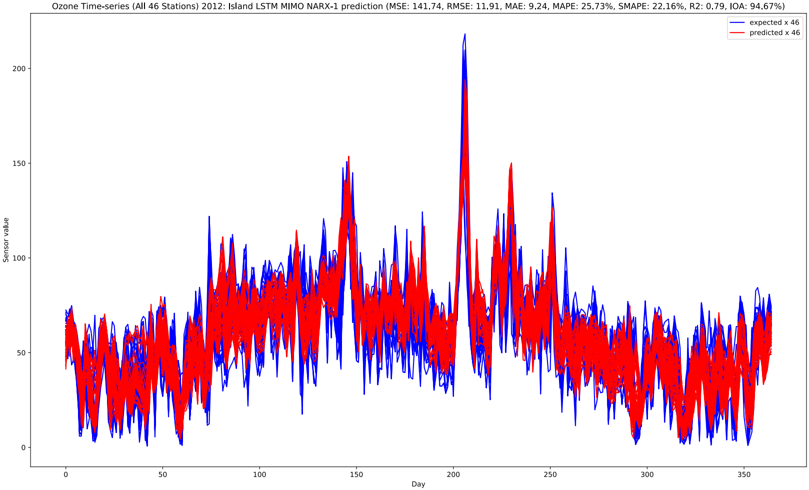
Ozone: MISO NARX (1 station: BETN073, 1 day lag)

train: 2000-2009, test: 2010, missing data: linear interpolation, models: [Alv13, Dum18]



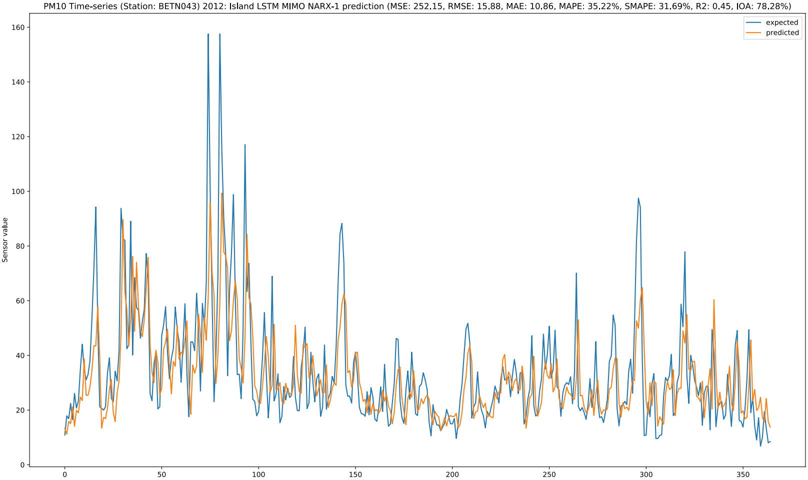
Ozone: At day ≈210, predicted **"inform public" level**

train: 1990-2011, test: 2012



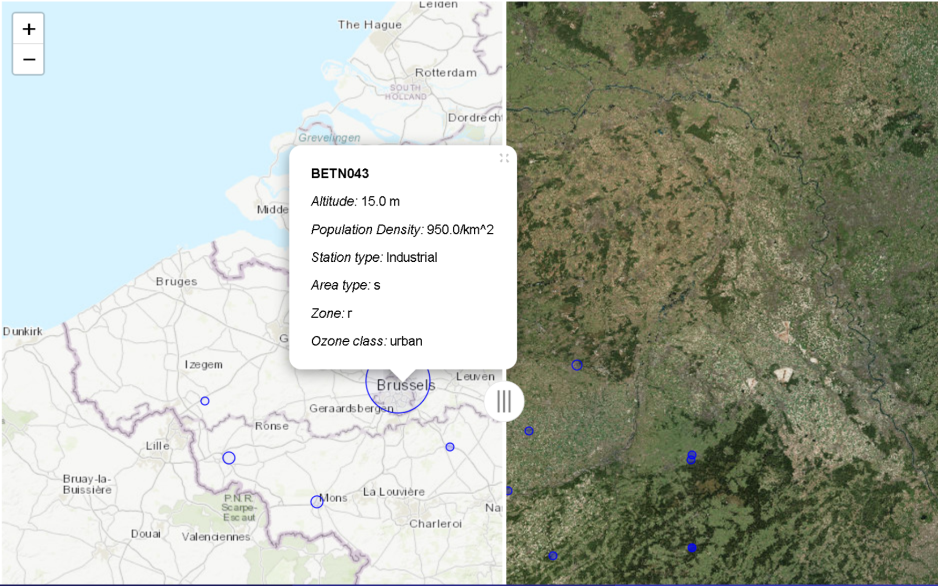
Particulate Matter $10\mu m$: Worst station prediction

BETN043 Brussels, Mean Absolute Error (MAE): 10.86



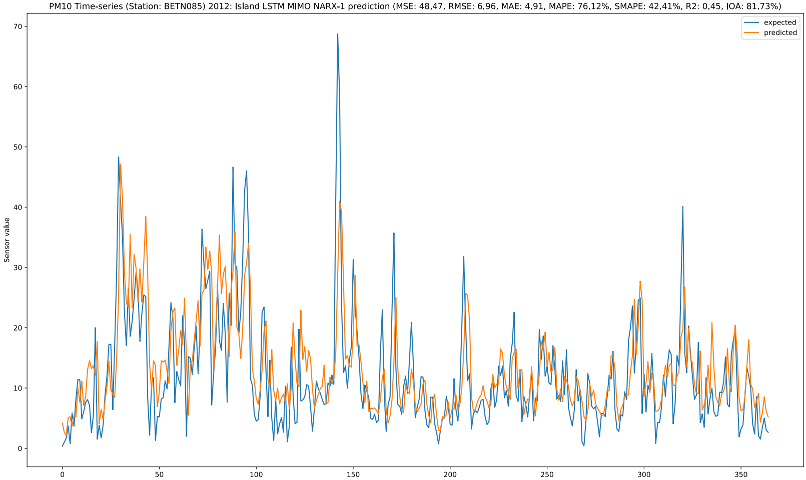
Particulate Matter $10\mu m$: Worst station prediction

BETN043 Brussels, MAE: 10.86



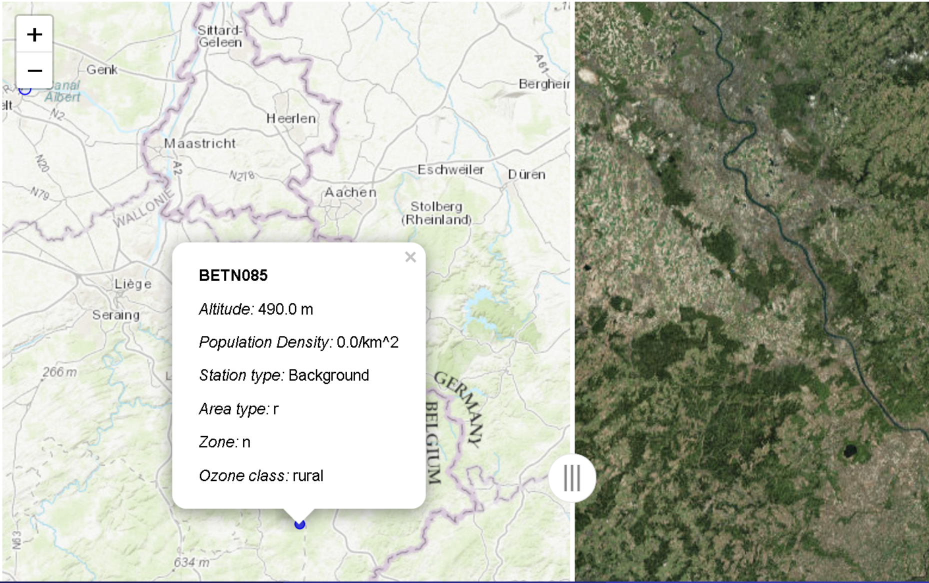
Particulate Matter $10\mu m$: Best station prediction

BETN085 Ardennes MAE: 4.91

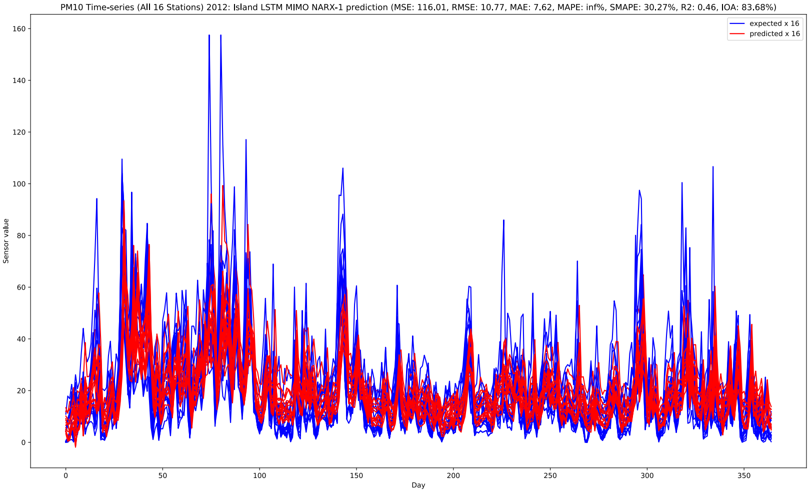


Particulate Matter $10\mu m$: Best station prediction

BETN085 Ardennes MAE: 4.91

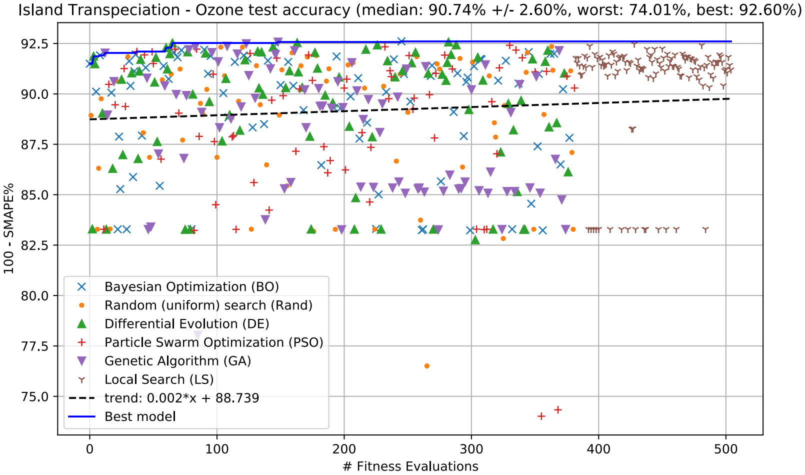


Particulate Matter $10\mu m$: BETN stations in BE (1-day lag)



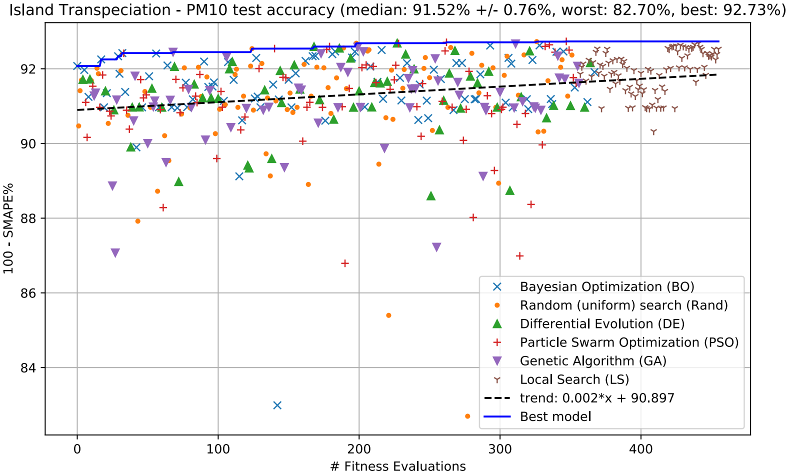
Island Transpeciation: Ozone 2018 (46 stations)

18 islands (1D grid: 1x18), 5 individuals per island, calendar cyclical features

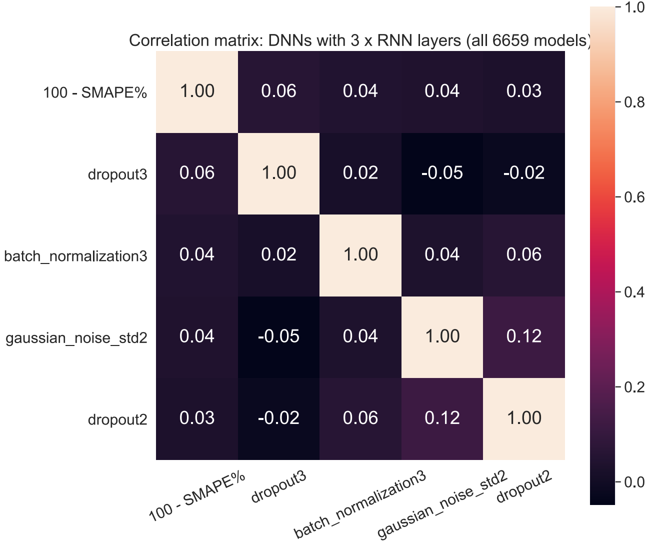


Island Transpeciation: PM 10μm 2018 (16 stations)

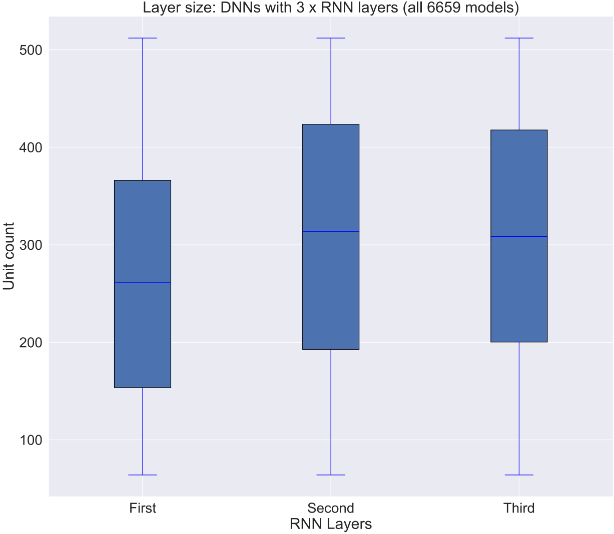
18 islands (CA 3D grid: 3x3x3), 5 individuals per island, calendar cyclical features



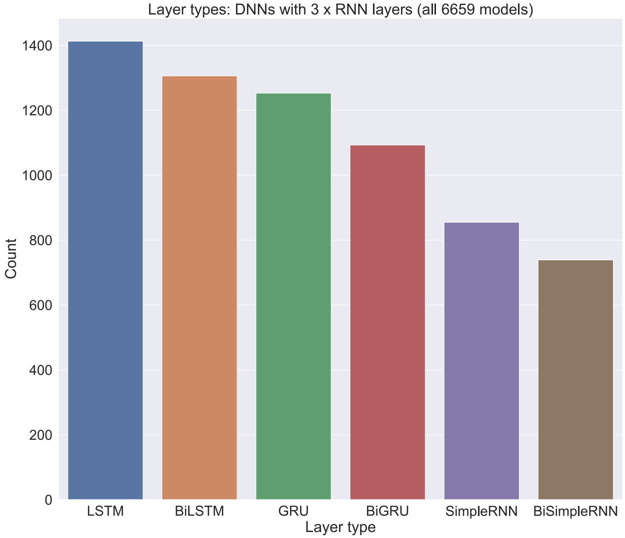
Island Transpeciation: DNN hyperparameters



Island Transpeciation: DNN layer sizes



Island Transpeciation: DNN layer types



Conclusions

Conclusions I

Deep Neural Networks for air-quality forecasting:

- *All-in-one* architecture.
- Successfully predicted ozone alert level in BE, 1 day before (for 2012).
- Particulate Matter 10μm stations with *high population density* (i.e Brussels), harder to predict.
- Not only exogenous variables, but *nearby station time-series* augment predictions.

Conclusions II

Island Transpeciation: Pros

- Can generate better models than *naive or random search*.
- *Highly diverse global optimizers* can co-evolve via cooperation and competition.
- Iterative hyper-parameter optimizers can be *parallelized*.
- *Hybrid GPU workers* can be combined with *fault tolerance*: local workstation, embedded, cloud Virtual Machines, supercomputer nodes.
- "Survival of the fattest" auto-regulated asynchronously (via model *size* vs *training speed* trade-off):
  - Fatter models: more trainable parameters BUT slower training.
  - Smaller models: less trainable parameters BUT train faster $\Rightarrow$ MORE evolutionary iteration improvements.

Conclusions III

Island Transpeciation: Cons

- Additional configuration versus off-the-self hyperparameter optimizers.
- On *internal representation* ⇔ *genotype* encoding/decoding:
  - May be lossy.
  - Requires altering optimizer source code internals.
  - Encoding/decoding not always two-way i.e random search: OUTSIDE migrations only.
- Adding islands: increases parallelism needs. Islands should iterate enough, to perform optimally.

APPENDIX

Development Environment

Hardware

CPU, RAM & storage

- Intel i7 6800K @ 3.4GHz with 6 physical cores.
- Memory 64GB DDR4 2800MHz.
- Samsung 960 Pro 512GB M.2 drive.

Hybrid GPU workers (CUDA capable)

- MSI *GTX 1070 Ti 8GB* (Pascal architecture).
- Gigabyte *GTX 970 4GB* (Maxwell architecture).
- *Nvidia Jetson TX2 4GB* (embedded ARM 64 device with Pascal GPU).
- Vlaams Supercomputer Centrum (VSC): up to 4x *Nvidia Tesla P100 16GB* (Pascal architecture) per node.
- Elastic Compute Cloud (EC2) Amazon Machine Instances (AMI): 4x p3.8xlarge with *Nvidia Tesla V100 16GB* (Volta architecture).

Environment

- OS: Windows 10 Education Version 1803 (Build 17134.885).
- IDE: Pycharm 2018.2.4 Professional (https://www.jetbrains.com/pycharm/download).
- Package Manager: Anaconda Navigator 1.9.7 (https://www.anaconda.com/download/).

Programming & Frameworks

- Language: Python 3.6.8.
- Libraries & frameworks:
  - Tensorflow 1.14 (https://www.tensorflow.org/).
  - Pandas 0.24.2, SciPy 1.3.0, Numpy 1.16.4, mpi4py 2.0.0, deap 1.2.2
  - Cuda 10 v10.1.105 (https://developer.nvidia.com/cuda-downloads).

Thesis Info

Jupyter Notebooks

- **Ozone**: `https://nbviewer.jupyter.org/github/temp3rr0r/Ozone-Narx-DNN/blob/master/models/NarxModelSearch/Ozone%20Time-Series%20Forecasting%20Models%20(All%20Belgium).ipynb`.
- **Particulate Matter**: `https://nbviewer.jupyter.org/github/temp3rr0r/Ozone-Narx-DNN/blob/master/models/NarxModelSearch/Particulate%20Matter%20Time-Series%20Forecasting%20Models.ipynb`
- **Map** of PM10 stations: `https://nbviewer.jupyter.org/github/temp3rr0r/Ozone-Narx-DNN/blob/master/data/Map%20of%20stations.ipynb`

Project Repository

`https://github.com/temp3rr0r/Ozone-Narx-DNN`

References I

N. R. Adiga, M. A. Blumrich, D. Chen, P. Coteus, A. Gara, M. E. Giampapa, P. Heidelberger, S. Singh, B. D. Steinmacher-Burow, T. Takken, M. Tsao, and P. Vranas.
Blue Gene/L torus interconnection network.
IBM Journal of Research and Development, 2010.

Johanna Orellana Alvear.
Application of LS-SVMs to Ozone Forecasting in Belgium.
2013.

Christoph Bergmeir.
A Note on the Validity of Cross-Validation for Evaluating Time Series Prediction A Note on the Validity of Cross-Validation for Evaluating Time Series Prediction.
(April), 2015.

References II

Thomas Bäck and Frank Hoffmeister.
Extended Selection Mechanisms in Genetic Algorithms.
In *Proceedings of the 4th International Conference on Genetic Algorithms*, 1991.

John Duchi JDUCHI and Yoram Singer.
Adaptive Subgradient Methods for Online Learning and Stochastic Optimization * Elad Hazan.
Technical report, 2011.

Denis Dumoulin.
Fixed-Size Multi-Output LSSVM for Nonlinear System Identification.
2018.

References III

D. P. Dee, S. M. Uppala, A. J. Simmons, P. Berrisford, P. Poli, S. Kobayashi, U. Andrae, M. A. Balmaseda, G. Balsamo, P. Bauer, P. Bechtold, A. C.M. Beljaars, L. van de Berg, J. Bidlot, N. Bormann, C. Delsol, R. Dragani, M. Fuentes, A. J. Geer, L. Haimberger, S. B. Healy, H. Hersbach, E. V. Hólm, L. Isaksen, P. Kållberg, M. Köhler, M. Matricardi, A. P. Mcnally, B. M. Monge-Sanz, J. J. Morcrette, B. K. Park, C. Peubey, P. de Rosnay, C. Tavolato, J. N. Thépaut, and F. Vitart.
The ERA-Interim reanalysis: Configuration and performance of the data assimilation system.
Quarterly Journal of the Royal Meteorological Society, 2011.

A. E. Eiben and J. E. Smith.
Introduction to Evolutionary Computing Genetic Algorithms.
Springer, 2015.

References IV

European Commission.
Directive 2002/3/EC of the European Parliament and of the council of 12 February 2002 relating to ozone in ambient air.
Official Journal of the European Union, 2002.

Sepp Hochreiter and Jürgen Schmidhuber.
Long Short-Term Memory.
Neural Computation, 9(8):1735–1780, 1997.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville.
Deep Learning.
MIT Press, 2016.

Sergey Ioffe and Christian Szegedy.
Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.
2 2015.

References V

Diederik P. Kingma and Jimmy Ba.
Adam: A Method for Stochastic Optimization.
12 2014.

Tim Kovacs.
Genetics-Based Machine Learning.
In *Handbook of Natural Computing*, pages 937–986. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter.
Self-Normalizing Neural Networks.
6 2017.

References VI

Risto Miikkulainen, Jason Liang, Elliot Meyerson, Aditya Rawal, Daniel Fink, Olivier Francon, Bala Raju, Hormoz Shahrzad, Arshak Navruzyan, Nigel Duffy, and Babak Hodjat.
Evolving Deep Neural Networks.
Artificial Intelligence in the Age of Neural Networks and Brain Computing, pages 293–312, 1 2019.

Mark J. Nieuwenhuijsen.
Exposure assessment in occupational and environmental epidemiology.
2009.

Project Science Office.
NASA's Earth Observing System, 2016.

Grzegorz Rozenberg, Thomas Back, and Joost N. Kok.
Handbook of Natural Computing.
2012.

S. J. Reddi, S. Kale, and S. Kumar.
On the Convergence of Adam and Beyond.
In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc Le, and Alex Kurakin.
Large-Scale Evolution of Image Classifiers.
2017.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, and Ruslan Salakhutdinov.
Dropout: A Simple Way to Prevent Neural Networks from Overfitting.
Technical report, 2014.

References VIII

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton.
On the importance of initialization and momentum in deep learning.
Technical report, 2013.

Mike Schuster and Kuldip K. Paliwal.
Bidirectional recurrent neural networks.
IEEE Transactions on Signal Processing, 1997.

Konstantinos Theodorakos.
Genetic Algorithms and Evolutionary Computing project : "Forecasting air-quality in Belgium with Neuroevolution and Island Transpeciation".
2019.

Marco Tomassini.
Spatially Structured Evolutionary Algorithms.
Springer, 2005.

References IX

D Whitley.
The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best.
In *Proceedings of the Third International Conference on Genetic Algorithms*, 1989.

Xin Yao.
Evolving artificial neural networks.
Proceedings of the IEEE, 1999.

Matthew D. Zeiler.
ADADELTA: An Adaptive Learning Rate Method.
12 2012.