Deep Learning for Time Series Forecasting

M2 Data Science & Artificial Intelligence

Juliette Chevallier

January 28, 2021

Introduction

Making Predictions with Sequences

Sequence $\leftrightarrow$ Explicit order on the observations that must be preserved when training models and making predictions.

- **Sequence Prediction**: Weather forecasting, Stock market prediction, Product recommendation;
- **Sequence Classification**: DNA Sequence Classification, Anomaly Detection, Sentiment Analysis;
- **Sequence Generation**: Text Generation, Handwriting Prediction, Music Generation;
- **Sequence-to-Sequence Prediction**: Multi-Step Time Series Forecasting, Text Summarization, Program Execution.

From: Vinyals, Toshev, Bengio, Erhan. Show and tell: A neural image caption generator. *CVPR* 2015

Previously Studied “Tools”

Time Series Analysis (Lecture 1):

- Describing temporal dynamics in great detail;
- Specific interest: unemployment rate, stock market indices, *etc.*;
- Realization of a stochastic process;
- Decomposition: trend, seasonality and (stochastic) reminder;

Previously Studied “Tools”

Time Series Analysis (Lecture 1):

- Describing temporal dynamics in great detail;
- Specific interest: unemployment rate, stock market indices, *etc.*;
- Realization of a stochastic process;
- Decomposition: trend, seasonality and (stochastic) reminder;

Longidudinal Data Analysis $\leftrightarrow$ Mixed-Effect Models (Lecture 2):

- Make inferences about the population;
- Fairly general temporal processes: growth, disease monitoring, *etc.*;
- Low sample size;
- Highly structured data, grouping factors such as species, gender, *etc.*;
- Bayesian frameworks allows prediction;

Previously Studied “Tools”

Time Series Analysis (Lecture 1):

- Describing temporal dynamics in great detail;
- Specific interest: unemployment rate, stock market indices, *etc.*;
- Realization of a stochastic process;
- Decomposition: trend, seasonality and (stochastic) reminder;

Longidudinal Data Analysis ↔ Mixed-Effect Models (Lecture 2):

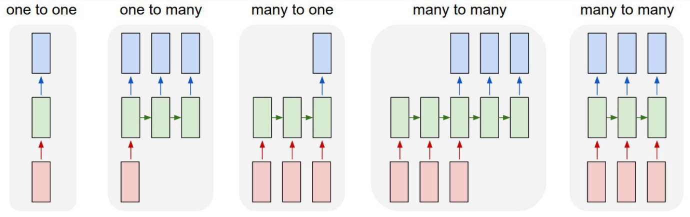
- Make inferences about the population;
- Fairly general temporal processes: growth, disease monitoring, *etc.*;
- Low sample size;
- Highly structured data, grouping factors such as species, gender, *etc.*;
- Bayesian frameworks allows prediction;

? Huge amount of data, but *does not fall under the “time series”* as defined in Lecture 1.
→ **Deep Learning**: MLP and CNN regression, RNN (LSTM), *etc.*

Neural Networks for Time Series

Why neural networks?

- Robust to noise, Support missing values;
- Nonlinear;
- Multivariate inputs and multi-step forecasts;
- Recurrent neural networks (RNN): Learned temporal dependence.



Choosing the right setting

Tasks	Possible techniques
Weather forecasting Stock market prediction Product recommendation	
DNA Sequence Classification Anomaly Detection Sentiment Analysis	
Text Generation Handwriting Prediction Music Generation	
Multi-Step Time Series Forecasting Text Summarization Program Execution	

AE: Autoencoder
CNN: Convolutional Neural Network
GAN: Generative Adversarial Networks
LME: Linear Mixed-Effect Models
MLP: Multilayer Perceptrons
NLME: Nonlinear Mixed-Effect Models
PDE: Partial Differential Equation
RNN: Recurrent Neural Network
TS: Time Series
VAE: Variational Autoencoders

Choosing the right setting

Tasks	Possible techniques
Weather forecasting	*PDE, RNN*
Stock market prediction	*TS*, RNN
Product recommendation	*RNN*
DNA Sequence Classification	
Anomaly Detection	
Sentiment Analysis	
Text Generation	
Handwriting Prediction	
Music Generation	
Multi-Step Time Series Forecasting	
Text Summarization	
Program Execution	

AE: Autoencoder
CNN: Convolutional Neural Network
GAN: Generative Adversarial Networks
LME: Linear Mixed-Effect Models
MLP: Multilayer Perceptrons
NLME: Nonlinear Mixed-Effect Models
PDE: Partial Differential Equation
RNN: Recurrent Neural Network
TS: Time Series
VAE: Variational Autoencoders

Choosing the right setting

Tasks	Possible techniques
Weather forecasting	*PDE, RNN*
Stock market prediction	*TS*, RNN
Product recommendation	*RNN*
DNA Sequence Classification	*CNN*
Anomaly Detection	*(N)LME, MLP, CNN, AE, GAN*
Sentiment Analysis	*(N)LME, MLP, CNN, AE, GAN*
Text Generation	
Handwriting Prediction	
Music Generation	
Multi-Step Time Series Forecasting	
Text Summarization	
Program Execution	

AE: Autoencoder
CNN: Convolutional Neural Network
GAN: Generative Adversarial Networks
LME: Linear Mixed-Effect Models
MLP: Multilayer Perceptrons
NLME: Nonlinear Mixed-Effect Models
PDE: Partial Differential Equation
RNN: Recurrent Neural Network
TS: Time Series
VAE: Variational Autoencoders

Choosing the right setting

Tasks	Possible techniques
Weather forecasting	*PDE, RNN*
Stock market prediction	*TS*, RNN
Product recommendation	*RNN*
DNA Sequence Classification	*CNN*
Anomaly Detection	*(N)LME, MLP, CNN, AE, GAN*
Sentiment Analysis	*(N)LME, MLP, CNN, AE, GAN*
Text Generation	*CNN, VAE, GAN*
Handwriting Prediction	*(N)LME, CNN, VAE, GAN*
Music Generation	*CNN, VAE, GAN*
Multi-Step Time Series Forecasting	
Text Summarization	
Program Execution	

AE: Autoencoder
CNN: Convolutional Neural Network
GAN: Generative Adversarial Networks
LME: Linear Mixed-Effect Models
MLP: Multilayer Perceptrons
NLME: Nonlinear Mixed-Effect Models
PDE: Partial Differential Equation
RNN: Recurrent Neural Network
TS: Time Series
VAE: Variational Autoencoders

Choosing the right setting

Tasks	Possible techniques
Weather forecasting	*PDE, RNN*
Stock market prediction	*TS*, RNN
Product recommendation	*RNN*
DNA Sequence Classification	*CNN*
Anomaly Detection	*(N)LME, MLP, CNN, AE, GAN*
Sentiment Analysis	*(N)LME, MLP, CNN, AE, GAN*
Text Generation	*CNN, VAE, GAN*
Handwriting Prediction	*(N)LME, CNN, VAE, GAN*
Music Generation	*CNN, VAE, GAN*
Multi-Step Time Series Forecasting	*TS*, RNN
Text Summarization	*RNN*
Program Execution	*CNN, AE*

AE: Autoencoder
CNN: Convolutional Neural Network
GAN: Generative Adversarial Networks
LME: Linear Mixed-Effect Models
MLP: Multilayer Perceptrons
NLME: Nonlinear Mixed-Effect Models
PDE: Partial Differential Equation
RNN: Recurrent Neural Network
TS: Time Series
VAE: Variational Autoencoders

Choosing the right setting

Tasks	Possible techniques
Weather forecasting	*PDE, RNN*
Stock market prediction	*TS*, RNN
Product recommendation	*RNN*
DNA Sequence Classification	*CNN*
Anomaly Detection	*(N)LME, MLP, CNN, AE, GAN*
Sentiment Analysis	*(N)LME, MLP, CNN, AE, GAN*
Text Generation	*CNN, VAE, GAN*
Handwriting Prediction	*(N)LME, CNN, VAE, GAN*
Music Generation	*CNN, VAE, GAN*
Multi-Step Time Series Forecasting	*TS*, RNN
Text Summarization	*RNN*
Program Execution	*CNN, AE*

AE: Autoencoder
CNN: Convolutional Neural Network
GAN: Generative Adversarial Networks
LME: Linear Mixed-Effect Models
MLP: Multilayer Perceptrons

NLME: Nonlinear Mixed-Effect Models
PDE: Partial Differential Equation
RNN: Recurrent Neural Network
TS: Time Series
VAE: Variational Autoencoders

Introduction

Multilayer Perceptron Regression

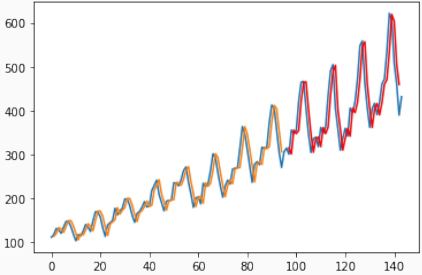
- Time series prediction $\longleftrightarrow$ Regression problem: x_{t+1} as a function of x_t
  $\rightsquigarrow$ *Multilayer Perceptron model*

Multilayer Perceptron Regression

- Time series prediction $\longleftrightarrow$ Regression problem: x_{t+1} as a function of x_t
  $\rightsquigarrow$ *Multilayer Perceptron model*

- Long training, hyperparameters to be tuned...

- No more efficient than an ARIMA model (or even less)

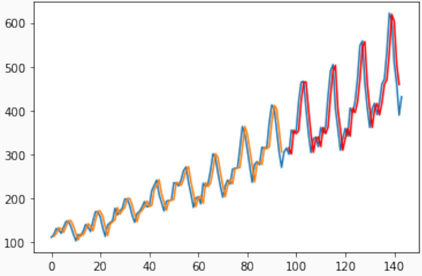


Airline passengers: Blue=Whole Dataset, Orange=Training, Red=Predictions

Multilayer Perceptron Regression

- Time series prediction $\longleftrightarrow$ Regression problem: x_{t+1} as a function of x_t
  $\rightsquigarrow$ *Multilayer Perceptron model*
- Long training, hyperparameters to be tuned...
- No more efficient than an ARIMA model (or even less)

Do not do that!



Airline passengers: Blue=Whole Dataset, Orange=Training, Red=Predictions

Long Short-Term Memory Networks for Time Series Forecasting

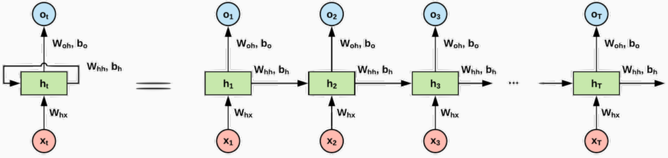
Recurrent Neural Networks (RNN) Reminders

- *Idea:* Make use of sequential information;
  "Memory" $\rightarrow$ capture information about what has been calculated so far

- *Different types of RNN's:*
  - One-to-one *e.g.* Image classification,
  - One-to-Many *e.g.* Image captioning,
  - Many-to-One *e.g.* Sentiment analysis,
  - Many-to-Many *e.g.* Machine Translation;

- O_t output state, h_t current time stamp, h_{t-1} previous time stamp, and x_t passed as input state
  W_{hh} weight at previous hidden state, W_{hx} weight at current input state, and W_{hy} weight at the output state

$$\boxed{h_t = \tanh\left(W_{hh}h_{t-1} + W_{hx}x_t\right)} \quad \text{and} \quad \boxed{y_t = W_{hy}h_t}$$



Recurrent Neural Networks (RNN) Reminders

While backpropogating you may get 2 types of issues:

- Vanishing Gradient,
- Exploding Gradient.

Recurrent Neural Networks (RNN) Reminders

While backpropogating you may get 2 types of issues:

- Vanishing Gradient
  $\rightarrow$ $Relu$ activation function,
  $\rightarrow$ LSTM, GRU.

- Exploding Gradient
  $\rightarrow$ Truncate or squash the gradients.

Recurrent Neural Networks (RNN) Reminders

While backpropogating you may get 2 types of issues:

- Vanishing Gradient
  $\rightarrow$ $Relu$ activation function,
  $\rightarrow$ LSTM, GRU.

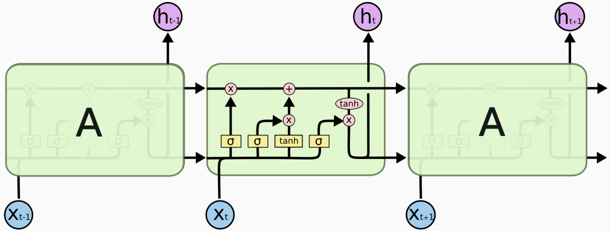
- Exploding Gradient
  $\rightarrow$ Truncate or squash the gradients.

Remarks:

- Training an RNN is a very difficult task,
- It cannot process very long sequences if using $\tanh$ or $Relu$ as an activation function.

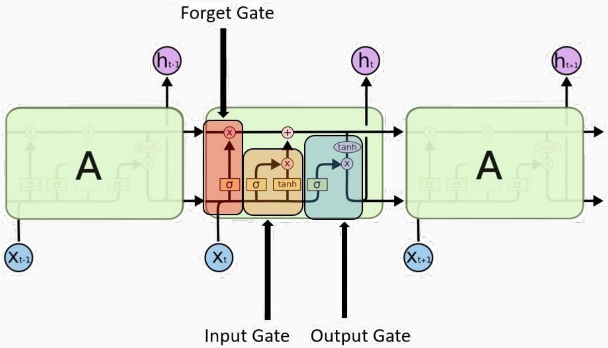
Long Short-Term Memory Networks for Time Series Forecasting

"Special kind of RNN's, capable of learning long-term dependencies."



Remark: A Blog on LSTM's with nice visualization:
https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714

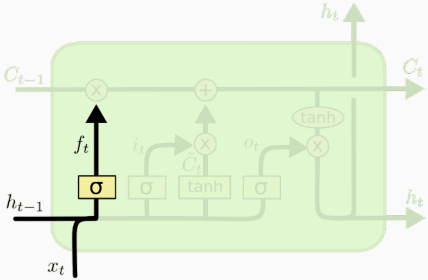
"Special kind of RNN's, capable of learning long-term dependencies."



LSTM had a three step process: Forget gate, Input gate, Output gate.

Remark: A Blog on LSTM's with nice visualization:
`https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714`

"Decides how much of the past you should remember."



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

Input:

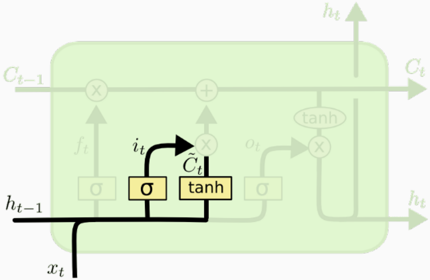
- Previous state h_{t-1},
- Content input x_t.

Output:

- A number between 0 (omit this) and 1 (keep this)for each number in the cell state C_{t-1}.

Update Gate *or* Input Gate:

"Decides how much of this unit is added to the current state."



$$\boxed{i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right)}$$

$$\boxed{\tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right)}$$

$$\boxed{C_t = f_t * C_{t-1} + i_t * \tilde{C}_t}$$

Input:

- Previous state h_{t-1},
- Content input x_t.

Output:

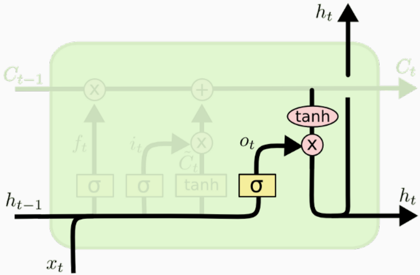
- Cell state C_t.

Remarks

- Sigmoid decides which values to let through 0,1;
- tanh gives weightage to the values which are passed deciding their level of importance ranging from -1 to 1.

Output Gate

"Decides which part of the current cell makes it to the output."



$$\boxed{o_t \;=\; \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right)}$$

$$\boxed{h_t \;=\; o_t * \tanh(C_t)}$$

Input:

- Previous state h_{t-1},
- Content input x_t,
- Cell stateC_t,

Output:

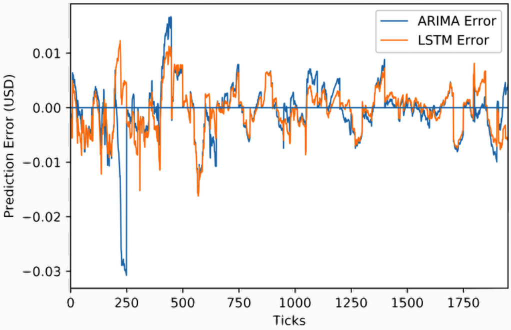
- Current state h_t.

Long Short-Term Memory Networks for Time Series Forecasting

Time Series Forecasting Using LSTM Networks

See "***Practical Work 5*** – *Time Series Forecasting Using LSTM Networks*"

- Strength of LSTM for time series forecasting,
- a non-exhaustive list of different variants of the vanilla LSTM.



Baughman, Haas, Wolski, Foster, Chard. *Predicting Amazon Spot Prices with LSTM Networks*. 2018.

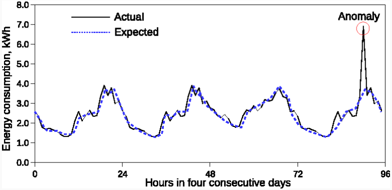
Deep Learning for Anomaly Detection

Anomaly detection: Outlier detection *or* Novelty detection.

Broad domains of applications: risk management, compliance, security, financial surveillance, health and medical risk, AI safety, *etc.*

Major Problem Complexities:

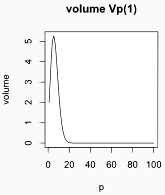
- *Unknownness*,
- *Heterogeneous* anomaly classes,
- *Rarity* and class imbalance,
- *Diverse types* of anomaly: *(i)* Point anomalies, *(ii)* Conditional anomalies, *(iii)* Group anomalies.

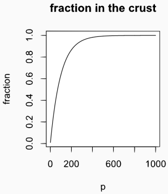


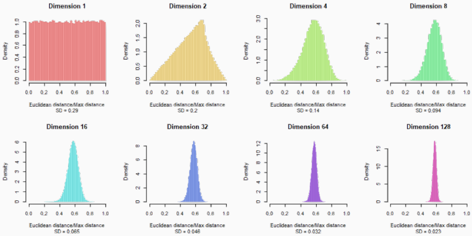
Pang, Shen, Cao and van den Hengel. *Deep learning for anomaly detection: A review*. 2020.

Largely Unsolved Challenges in Anomaly Detection

1. Low anomaly detection recall rate;
2. Anomaly detection in high-dimensional and/or not-independent data;
3. Data-efficient learning of normality/abnormality;
4. Noise-resilient anomaly detection;
5. Detection of complex anomalies;
6. Anomaly explanation.







Largely Unsolved Challenges in Anomaly Detection

1. **Low anomaly detection recall rate**:
   A still high number of false positives on real-world data;

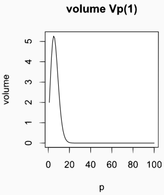
2. Anomaly detection in high-dimensional and/or not-independent data;

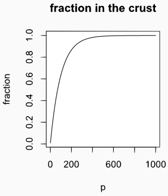
3. Data-efficient learning of normality/abnormality;

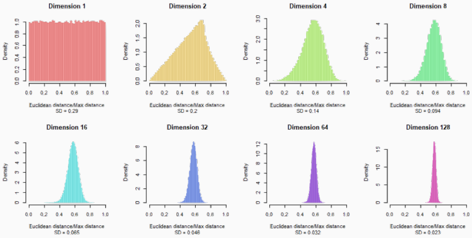
4. Noise-resilient anomaly detection;

5. Detection of complex anomalies;

6. Anomaly explanation.







Largely Unsolved Challenges in Anomaly Detection

1. Low anomaly detection recall rate:
   A still high number of false positives on real-world data;

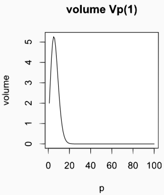
2. **Anomaly detection in high-dimensional and/or not-independent data**:
   → Subspace/feature selection-based methods
   ⚠ Intricate feature + proper information preserved?
   Detect anomalies from dependent instances (temporal, spatial,. . .);

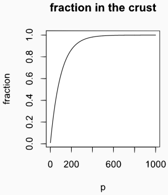
3. Data-efficient learning of normality/abnormality;

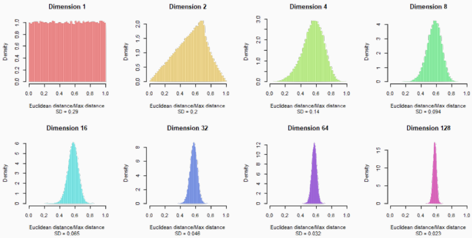
4. Noise-resilient anomaly detection;

5. Detection of complex anomalies;

6. Anomaly explanation.







Largely Unsolved Challenges in Anomaly Detection

1. Low anomaly detection recall rate:
   A still high number of false positives on real-world data;

2. Anomaly detection in high-dimensional and/or not-independent data:
   → Subspace/feature selection-based methods
   ⚠ Intricate feature + proper information preserved?
   Detect anomalies from dependent instances (temporal, spatial,. . .);

3. **Data-efficient learning of normality/abnormality**:
   Fully supervised anomaly detection requires labeled anomaly data
   Unsupervised anomaly detection requires prior knowledge of true anomalies
   → Semi-supervised or weakly-supervised anomaly detection;

4. Noise-resilient anomaly detection;

5. Detection of complex anomalies;

6. Anomaly explanation.

Largely Unsolved Challenges in Anomaly Detection

1. Low anomaly detection recall rate*:*
   A still high number of false positives on real-world data;

2. Anomaly detection in high-dimensional and/or not-independent data*:*
   $\rightarrow$ Subspace/feature selection-based methods
   ⚠ Intricate feature + proper information preserved?
   Detect anomalies from dependent instances (temporal, spatial,. . .);

3. Data-efficient learning of normality/abnormality*:*
   Fully supervised anomaly detection requires labeled anomaly data
   Unsupervised anomaly detection requires prior knowledge of true anomalies
   $\rightarrow$ Semi-supervised or weakly-supervised anomaly detection;

4. **Noise-resilient anomaly detection***:*
   Weakly/semi-supervised anomaly detection assume clean labeled data;

5. Detection of complex anomalies*;*

6. Anomaly explanation*.*

Largely Unsolved Challenges in Anomaly Detection

1. Low anomaly detection recall rate*:*
   A still high number of false positives on real-world data;

2. Anomaly detection in high-dimensional and/or not-independent data*:*
   $\rightarrow$ Subspace/feature selection-based methods
   ⚠ Intricate feature + proper information preserved?
   Detect anomalies from dependent instances (temporal, spatial,. . .);

3. Data-efficient learning of normality/abnormality*:*
   Fully supervised anomaly detection requires labeled anomaly data
   Unsupervised anomaly detection requires prior knowledge of true anomalies
   $\rightarrow$ Semi-supervised or weakly-supervised anomaly detection;

4. Noise-resilient anomaly detection*:*
   Weakly/semi-supervised anomaly detection assume clean labeled data;

5. **Detection of complex anomalies***:*
   Most methods are for point anomalies
   and focus on detect anomalies from single data sources;

6. Anomaly explanation.

Largely Unsolved Challenges in Anomaly Detection

1. Low anomaly detection recall rate*:*
   A still high number of false positives on real-world data;

2. Anomaly detection in high-dimensional and/or not-independent data*:*
   → Subspace/feature selection-based methods
   ⚠ Intricate feature + proper information preserved?
   Detect anomalies from dependent instances (temporal, spatial,. . .);

3. Data-efficient learning of normality/abnormality*:*
   Fully supervised anomaly detection requires labeled anomaly data
   Unsupervised anomaly detection requires prior knowledge of true anomalies
   → Semi-supervised or weakly-supervised anomaly detection;

4. Noise-resilient anomaly detection*:*
   Weakly/semi-supervised anomaly detection assume clean labeled data;

5. Detection of complex anomalies*:*
   Most methods are for point anomalies
   and focus on detect anomalies from single data sources;

6. **Anomaly explanation***:*
   Risks if anomaly detection models directly used as black-box models
   → Explanation + Human expert.

Deep Learning for Anomaly Detection

Deep methods:

- Aims: learning feature representations or anomaly scores via neural networks
- End-to-end optimization;
- Learning of representations specifically tailored for anomaly detection;
- Learning intricate structures and relations from diverse types of data;
  High-dimensional data, image data, video data, graph data, *etc.*
- Many effective and easy-to-use network architectures;

	Traditional	**Deep**
End-to-end Optimization	$\times$	✓
Tailored Representation Learning	$\times$	✓
Intricate Relation Learning	Weak	Strong
Heterogeneity Handling	Weak	Strong

Deep Anomaly Detection

Dataset: Let $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ with $x_i \in \mathbb{R}^d$.
Let $\mathcal{Z} \in \mathbb{R}^k$, $k \ll n$, be a representation space.

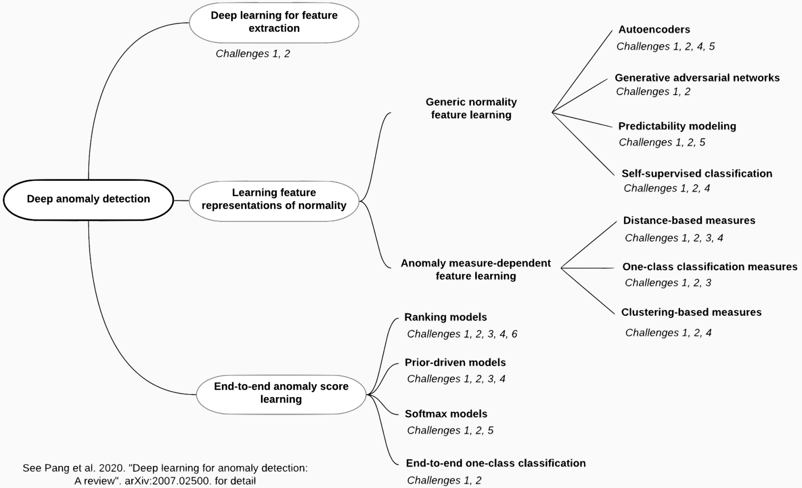
Deep anomaly detection aims at:

- learning a feature representation mapping function $\phi\colon \mathcal{X} \to \mathcal{Z}$
- ***OR*** an anomaly score learning function $\tau\colon \mathcal{X} \to \mathbb{R}$

so that:

- anomalies easily differentiated from normal data instances in the space induced by ϕ or τ,
- where ϕ and τ are neural networks with $h \in \mathbb{N}$ hidden layers,
- weight matrices: $\Theta = \{M^1, M^2, \ldots, M^h\}$.

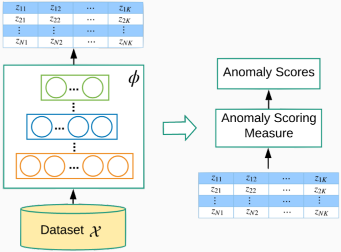
Hierarchical Taxonomy of Deep Anomaly Detection Methods



Proposed by *Pang, Shen, Cao, van den Hengel (2020)*

Taxonomy of current deep anomaly detection techniques

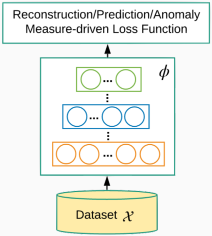
Deep Learning For Feature Extraction



- *Goal*: Extract low-dimensional feature representations from high-dimensional and/or non-linearly separable data;
- The deep learning components work purely as dimensionality reduction only;

- f: unrelated to ϕ scoring method, applied onto the new space;
- Compared to PCA or random projection, better capability in extracting semantic-rich features and non-linear feature relations;

Learning Feature Representations of Normality



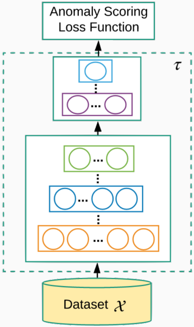
- *Goal:* Couple feature learning ***with*** anomaly scoringin some ways;

- *Two groups:* generic feature learning and anomaly measure-dependent feature learning.

- *Generic Normality Feature Learning:* Learn representations through generic methods not primarily designed for anomaly detection, but by forcing them to capture some key underlying data regularities;
  Autoencoders, Generative adversarial networks, Predictability modeling, Self-supervised classification

- *Anomaly Measure-dependent Feature Learning:* Learning feature representations specifically optimized for one particular anomaly measure.
  Distance-based measures, One-class classification measures, Clustering-based measures

End-to-end Anomaly Score Learning

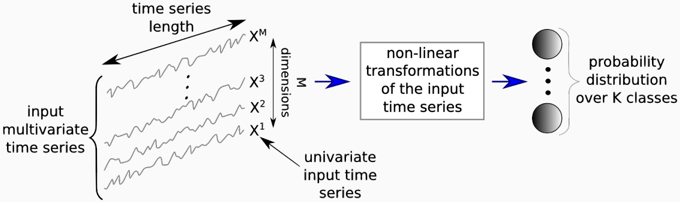


- *Goal:* Learning scalar anomaly scores in an end-to-end fashion;
- The anomaly scoring is not dependent on existing anomaly measures; it has a neural network that directly learns the anomaly scores;
- Novel loss functions are often required to drive the anomaly scoring network.

- Ranking models, Prior-driven models, Softmax models, End-to-end one-class classification

Suggested list of tools & datasets for anomaly detection on time-series data:
`https://github.com/rob-med/awesome-TS-anomaly-detection`

- ***Deep learning for time series classification**: A review*
  Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar & Pierre-Alain Muller
- Why dedicated algorithms for time series?



Conclusion

- Neural Networks and especially Recurrent Neural Networks have proven their efficiency;
- Be careful to choose the right method;
- A burgeoning and rapidly growing field of research.