Deep learning: Theory and Practice

Andrea Nigri, PhD

Dec 16, 2022

University of Foggia,
andrea.nigri@unifg.it

Outline

Outline

- Data analysis and Statistical Modeling
- What's deep learning
- Case study
- Conclusions

Data analysis and Statistical modeling

The bricks, i.e. the individual elements build a model, representative of a reality.

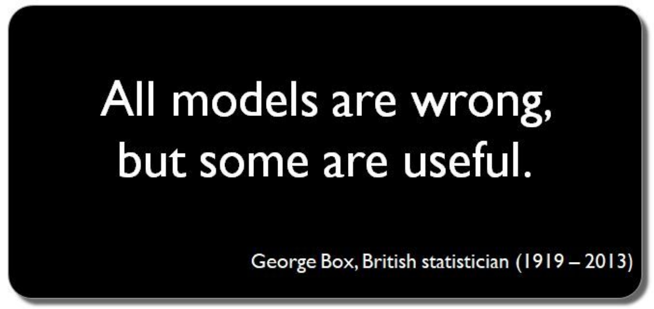
- In the **data analysis**, these bricks are the data
- Data that must be treated according to a given patter - our statistical model - to recreate/represent an observed reality that we want to approximate.

Why....? to understand an explicit or hidden pattern inherent in the data ... **why try to understand a pattern?**

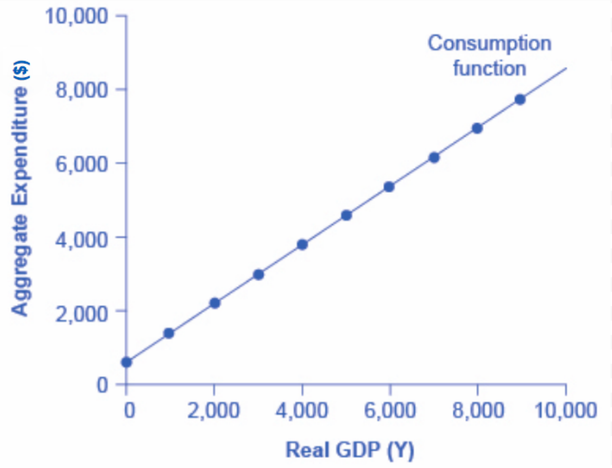
Interpret or predict

- To make decisions under conditions of uncertainty or even to anticipate upcoming scenarios and events.
- To this end, statistics uses probability to develop models with an underlying probabilistic nature to explain a given reality.

but...



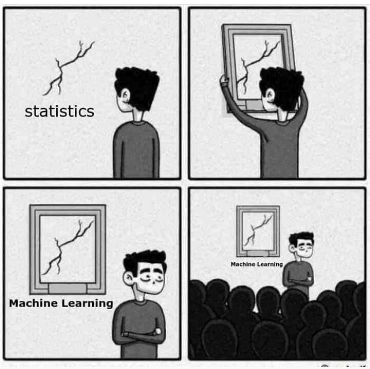
We will **never** know the so-called data generating model, i.e. the real model designed by nature that has allowed the generation of data.



Aggregate Expenditure ($)
10,000
8,000
6,000
4,000
2,000
0
Consumption function
0
2,000
4,000
6,000
8,000
10,000
Real GDP (Y)

What's deep learning

Statistical learning is an evolution of statistical modeling



- for a given set of training data $\{(x_1, y_1) \dots (x_n, y_n)\}$ sampled according to an unknown probability distribution $P(x, y)$, we find a function $f(\cdot)$ that minimises the expected error on a new test set of data:

$$\int L(y, f(x))P(x, y)dxdy$$

- where $L(y, f(x))$ is the loss function that measures the prediction error for a given x against the actual value y.

Statistics vs. Statistical Learning up to Machine Leaning

Statistical Learning: New term...old concepts.

- At the beginning of the nineteenth century - least squares
- 1940s - logistic regression
- early 1970s, generalized linear model

... they were almost linear methods because fitting non-linear relationships was computationally difficult.

Non-linear methods

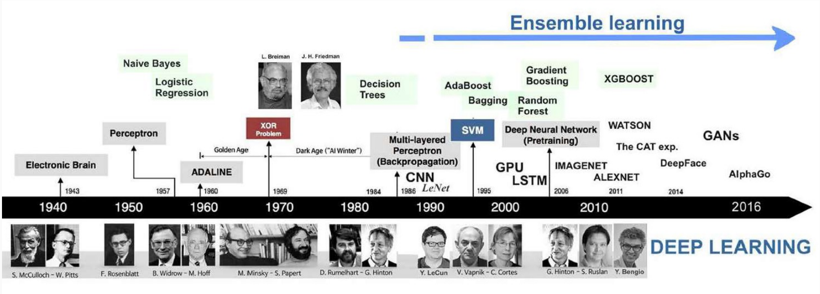
...By the 1980s, computing technology had finally improved sufficiently, that non-linear methods were no longer computationally prohibitive.

- mid 1980s, regression trees and generalized additive models ... Neural networks gained popularity
- 1990 support vector machines
- Machine Learning: modern evolution of statistical learning

Machine Leaning vs. Deep Learning

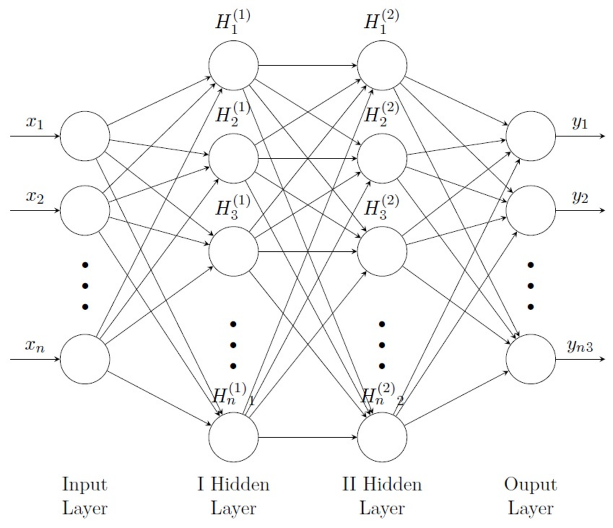
- While all deep learning is machine learning, not all machine learning is deep learning
- Machine Learning help the computer learn how to recognize things. This training requires a significant amount of human effort.
- Deep learning algorithms: hierarchical models (multi-layered neural network) that does not require preprocessing.

Time line



Ensemble learning
Naive Bayes
Logistic Regression
L. Breiman
J. H. Friedman
Decision Trees
AdaBoost
Bagging
Gradient Boosting
Random Forest
XGBOOST
Perceptron
XOR Problem
SVM
Deep Neural Network (Pretraining)
WATSON
GANs
The CAT exp.
Electronic Brain
Golden Age
Dark Age ("AI Winter")
Multi-layered Perceptron (Backpropagation)
ADALINE
CNN
LeNet
GPU
LSTM
IMAGENET
ALEXNET
DeepFace
AlphaGo
1943
1957
1960
1969
1984
1986
1995
2006
2011
2014
1940
1950
1960
1970
1980
1990
2000
2010
2016
S. McCulloch - W. Pitts
F. Rosenblatt
B. Widrow - M. Hoff
M. Minsky - S. Papert
D. Rumelhart - G. Hinton
Y. LeCun
V. Vapnik - C. Cortes
G. Hinton - S. Ruslan
Y. Bengio
DEEP LEARNING

Deep Learning: Deep Neural network



NN training involves an unconstrained optimization problem where the aim is to minimize a function in high dimensional space the so-called loss function, that measures the difference between the predicted values and observed ones. The back-propagation is the most used algorithm for the training of NNs. The algorithm compares the predicted values against the desired ones (objective) and modifies the synaptic weights by back-propagating the gradient of the loss function. Schematically, the procedure alternates forward and backward propagation steps:

- in the forward step, the prediction is computed fixing the synaptic weights,
- in the backward step, the weights are adjusted in order to reduce the error of the network.

The NN iteratively performs forward and backward propagation and modifies the weights to find the combination that minimizes the loss function

$$\mathcal{L}(y, \hat{y}) = -\sum_{i=1}^{n} y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i)$$

To minimize the loss function, we use the Gradient Descent optimization algorithm, that proceeds by minimizing $\mathcal{L}$ differentiating the loss function with respect to the weights ($\mathbf{W}$).

The algorithm proceeds using the chain derivation rule described in the following equation:

$$\frac{\partial \mathcal{L}(y, \hat{y})}{\partial w_{n,n}^{(k)}} = \frac{\partial \mathcal{L}(y, \hat{y})}{\partial H_n^{(k)}} \frac{\partial H_n^{(k)}}{\partial z_n^{(k)}} \frac{\partial z_n^{(k)}}{\partial w_{n,n}^{(k)}} \tag{1}$$

where $z_n^{(k)} = w_n^{(k)} H_n^{(k-1)} + b_n^{(k)}$. To update the weights ($\tilde{\mathbf{W}}$), the gradient of the loss function, $\nabla \mathcal{L}_t(y, \hat{y})$, is multiplied by a scalar, η, often called learning rate, according to the following scheme:

$$\tilde{\mathbf{W}} = \mathbf{W} - \eta \nabla \mathcal{L}_t(y, \hat{y}) \tag{2}$$

- NN, like other machine learning techniques, requires the splitting of the dataset into a training and a testing set. The training set stands for supervised learning, while the testing set is used to validate the model. After the training phase, the network has learned the input–output functional relationship and it should be able to predict future values using only the input.
- The search for the optimal parameters is then carried out through an optimization process where the NN initial weights are selected in an arbitrary (random) way so they are not optimal parameters. The iterations of the algorithm lead to the optimization of the weights and minimization of the error. The choices concerning the type of architecture (e.g., the number of hidden layers, units for each layer) and the hyperparameter (e.g., learning rate, activation functions, and loss function), remains a heuristic problem for NN users: the choice often depends on the type of data and it is not an easy step.

Case study

Case Study

Chinese consumers' attitude towards ready to eat salads by comparing traditional Logit with Machine Learning methods

Motivation:

- In the last decade, the ready to eat (RTE) food market has experienced substantial growth in China...but...
- ...No study offers an attitudinal and behavioral analysis of the topic
- A survey among Chinese respondents to understand the factors associated with RTE salad consumption

Contribution:

- **First,** we aim at profiling the typical consumer of RTE salads
- **Second,** we test different machine learning classification algorithms on primary consumer data.

Results: consumption is more common among

- young respondents,
- females,
- healthy-oriented individuals at the beginning of their career,
- the role of the subjective norm (other people influence on our choices) is positively associated with increased consumption of RTE salads.

Results: Predictive models

- RT
- RF
- SVM
- DNN

Variable	Frequency	Variable	Frequency
Gender		**Regular consumption of RTE**	
Female	*52%*	salads	
Male	*48%*	*Yes*	*25%*
Age		*No*	*75%*
21-25	*23%*	**Fitness**	
26-30	*34%*	*Yes*	*48%*
31-35	*12%*	*No*	*52%*
36-40	*17%*	**Learning-advertising**	
>40	*14%*	*Yes*	*76%*
Income		*No*	*24%*
Below 15,000rmb	*28%*	**Learning-social media**	
Between 15 and 20,000rmb	*25%*	*Yes*	*31%*
Between 20 and 25,000rmb	*21%*	*No*	*69%*
More than 25,000rmb	*27%*	**Purchasing-supermarket**	
Job seniority level		*Yes*	*75%*
Entry	*40%*	*No*	*25%*
Middle	*45%*	**Purchasing-convenience store**	
Managerial	*15%*	*Yes*	*49%*
Household size		*No*	*51%*
One or two people	*18%*	**Consumption for snacking**	
Three people	*50%*	*Yes*	*60%*
More than three people	*32%*	*No*	*40%*
Knowledge of RTE		**Consumption for lunch**	
Low	*25%*	*Yes*	*60%*
Medium	*39%*	*No*	*40%*
High	*36%*		
Subjective norm			
Low	*10.79%*		*60%*
Medium	*51.76%*		*40%*
High	*37.44%*		

Accuracy prediction

$\hat{y}/y$	1	0
1	TP	FP
0	FN	TN
	Sens: $\frac{TP}{TP+FN}$	Spec: $\frac{TN}{TN+FP}$

For binary target variables, we evaluate the level of accuracy and its 95% confidence interval (CI), true positive rate (Sensitivity), true negative rate (Specificity), and Cohen's Kappa. We define $y = 1$ for a regular consuption of RTE and 0 otherwise. Then a 2×2 confusion matrix has elements $a_{\text{row,column}}$ with predicted conditions $\widehat{y} = \{1, 0\}$ on rows and true conditions $y = \{1, 0\}$ on columns. The statistics are defined by,

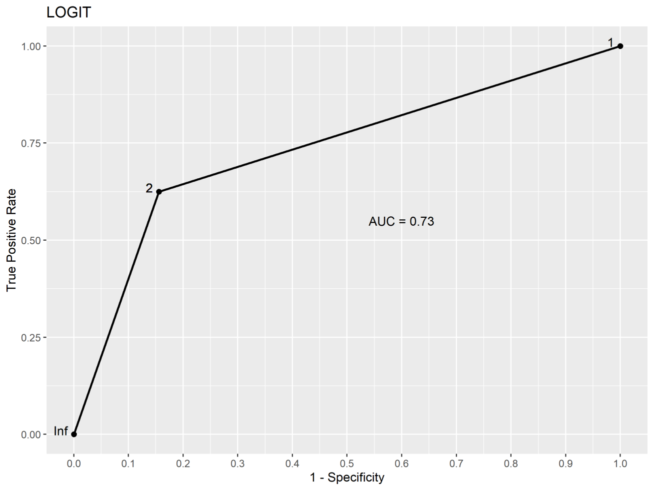
$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$$

Accuracy prediction

We calculate accuracy that refers to the portion of customers correctly classified with respect to RTE regular consumption. We calculate the 95% confidence interval using accuracy's standard deviation generated through iterations. Sensitivity (true positive rate) refers to the proportion of regular RTE consumers correctly identified as such. Poor sensitivity implies a large number of inclusion errors, i.e. identifying RTE consumers when in fact they are not. Specificity (true negative rate) refers to the proportion of customers correctly predicted to be occasional RTE consumers. Poor specificity implies a large number of exclusion errors.

The reported output also provides the McNemar test and the numerical and graphical representation of the ROC curve and the relative area under the curve (AUC) values. The area under the (ROC) curve, summarizes the classifier performance. The larger area under the curve the better the classifier.

Accuracy prediction



Let's go to practice.