Introduction to (Deep) Neural Networks

Luis F. Lago Fernández

MIAX-13

Course presentation

Lecturers

Dr. Luis F. Lago Fernández

- e-mail: luis.lago@uam.es

Christian Oliva Moya

- e-mail: christian.oliva@uam.es

Learning outcomes

Upon completion of this course, you will be able to:

- Understand the fundamentals of DL within the ML learning context
- Train a DNN, choosing the most appropriate characteristics and optimizing the hyperparameters
- Implement DL algorithms using different tools, such as TensorFlow, Keras or PyTorch

Course contents

1. Introduction to Deep Learning
2. Machine learning fundamentals
3. Neural Network basics:
   - Shallow neural networks
   - Backpropagation
4. Deep Neural Networks:
   - Practical aspects of deep learning: activation functions, loss functions, weight initialization
   - Batch normalization.
   - Regularization techniques, dropout

Course contents

5. Optimization techniques:
   - Stochastic Gradient Descent
   - Adaptive methods
6. Hyper-parameter tuning
7. Deep learning architectures:
   - Convolutional neural networks
   - Recurrent neural networks
   - Autoencoders and GANs
8. Deep Learning programming and parallelization tools: TensorFlow, Keras, PyTorch

Bibliography

- Deep Learning. I. Goodfellow, Y. Bengio and A. Courville. MIT Press, 2016. http://www.deeplearningbook.org/
- Neural Networks and Deep Learning. M. Nielsen. http://neuralnetworksanddeeplearning.com/
- Hands-On Machine Learning with Scikit-Learn and TensorFlow. A. Geron. O'Reilly, 2017.
- Deep Learning with Python. F. Chollet. Manning, 2017.
- Dive into Deep Learning. Zhang, Lipton, Li & Smola. https://d2l.ai/

Schedule

Day	**Contents**
Day 1	Introduction and basic concepts · Machine Learning Fundamentals · Linear Regression · Gradient Descent
Day 2	Logistic Regression · Non-linear models · Introduction to Neural Networks · NN Implementation (forward pass)

Schedule

Day	Contents
Day 3	Backpropagation · NN Implementation (backward pass) · Introduction to TensorFlow · Automatic Differentiation in TF
Day 4	NN implementation in TF · Practical aspects of NN Training: SGD, loss function, activation function · Keras I

Schedule

Day	**Contents**
Day 5	Practical aspects of NN Training: regularization, weight initialization, batch normalization · Second order optimization techniques · Keras II
Day 6	Hyperparameter optimization · Introduction to PyTorch

Schedule

Day	Contents
Day 7	Deep Learning Architectures · Kohonen Networks · Practical Assignment

Additional resources

- Introduction to Deep Learning, MIT, http://introtodeeplearning.com/
- Convolutional Neural Networks for Visual Recognition, Stanford, http://cs231n.stanford.edu/
- TensorFlow tutorials, https://www.tensorflow.org/tutorials/
- TensorFlow Playground, http://playground.tensorflow.org
- PyTorch tutorials

Recommendations

The following skills are highly recommended:

- Calculus
- Linear algebra
- Statistics and probability theory
- Python programming

What if I need to review maths concepts?

Goodfellow's book:

- Chapter 2: Linear algebra
- Chapter 3: Probability and information theory

Zico Kolter, 2015: Linear Algebra Review and Reference

What if I need to review programming concepts?

Justin Johnson: Python Numpy Tutorial (with Jupyter and Colab)

Introduction to deep learning

- What is deep learning?
- Why now?

What is Deep Learning?

- A subfield of Machine Learning: learn without being explicitly programmed
- Make predictions on data using Neural Networks
- Deep neural networks: many layers

Why deep learning now?

- Lots of data
- Increase in computational power (parallelization, GPUs, . . .)
- New programming tools, algorithms and tricks

Machine Learning basics

Suggested reading:
https://www.deeplearningbook.org/contents/ml.html

What is Machine Learning?

- Field of study that gives computers the ability to learn without being explicitly programmed. (Attributed to A. Samuel, 1959)
- Subfield of AI that studies computer algorithms that improve automatically through experience. (Wikipedia, 2020)

Different learning tasks

- Supervised machine learning
  - Classification
  - Regression
- Unsupervised machine learning
- Semi-supervised machine learning
- Reinforcement learning

Supervised machine learning - definitions

The problem data is the set of patterns
$\{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), ..., (\mathbf{x}_n, t_n)\}$, where:

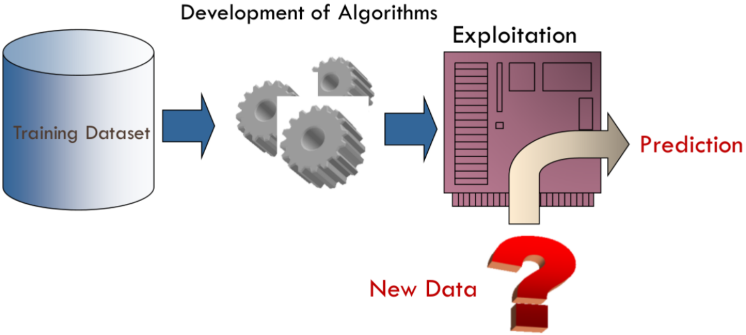
- $\mathbf{x}_i$ is the attribute vector for pattern i
- t_i is the target (variable to predict) for pattern i
- n is the total number of patterns

The goal is to predict the target t_i given the attribute vector $\mathbf{x}_i$

Parametric models for classification (regression)

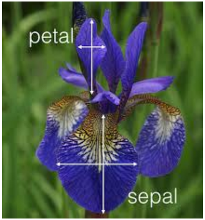
- A classifier (regressor) is a function $f(\mathbf{x}, \theta)$ that assigns each pattern $\mathbf{x}_i$ an estimation of its target value: $y_i = f(\mathbf{x}_i, \theta) \approx t_i$
- Model training: tune the model parameters θ in order to minimize a *loss function* $L(y_i, t_i)$
- Different function families define different types of models: Neural Networks, SVMs, etc.

Supervised machine learning - Overview



Supervised machine learning - an example

The Iris plant dataset (R.A. Fisher, 1936)



- Classify Iris plant samples into 3 subspecies: *Setosa*, *Virginica* and *Versicolor*
- Use the width and length of petal and sepal as attributes

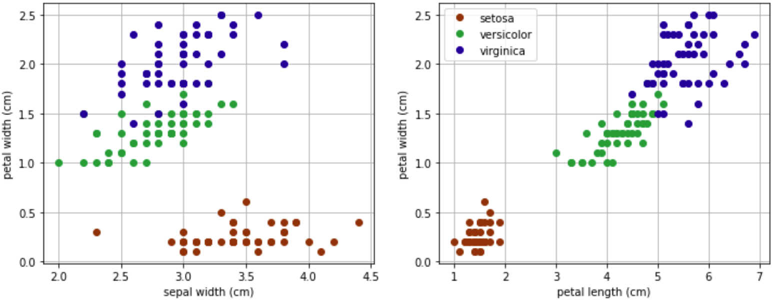
Supervised machine learning - an example

The Iris plant dataset (R.A. Fisher, 1936)

sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	target_num
5.1	3.5	1.4	0.2	setosa	0
5.4	3.7	1.5	0.2	setosa	0
5.4	3.4	1.7	0.2	setosa	0
4.8	3.1	1.6	0.2	setosa	0
5.0	3.5	1.3	0.3	setosa	0
7.0	3.2	4.7	1.4	versicolor	1
5.0	2.0	3.5	1.0	versicolor	1
5.9	3.2	4.8	1.8	versicolor	1
5.5	2.4	3.8	1.1	versicolor	1
5.5	2.6	4.4	1.2	versicolor	1
6.3	3.3	6.0	2.5	virginica	2
6.5	3.2	5.1	2.0	virginica	2
6.9	3.2	5.7	2.3	virginica	2
7.4	2.8	6.1	1.9	virginica	2
6.7	3.1	5.6	2.4	virginica	2

Supervised machine learning - an example

The Iris plant dataset (R.A. Fisher, 1936)



Classification or regression?

Supervised machine learning - second example

The Boston Housing problem

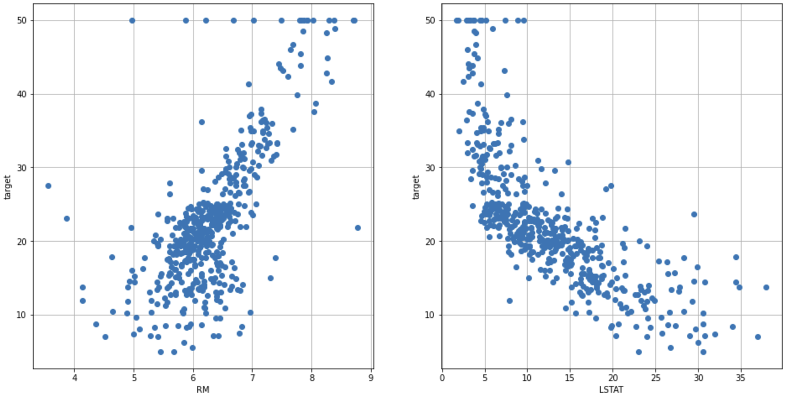
- Predict housing prices in the suburbs of Boston

The Boston Housing problem

CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	target
0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98	24.0
0.22489	12.5	7.87	0.0	0.524	6.377	94.3	6.3467	5.0	311.0	15.2	392.52	20.45	15.0
1.25179	0.0	8.14	0.0	0.538	5.570	98.1	3.7979	4.0	307.0	21.0	376.57	21.02	13.6
1.13081	0.0	8.14	0.0	0.538	5.713	94.1	4.2330	4.0	307.0	21.0	360.17	22.60	12.7
0.03359	75.0	2.95	0.0	0.428	7.024	15.8	5.4011	3.0	252.0	18.3	395.62	1.98	34.9
0.08873	21.0	5.64	0.0	0.439	5.963	45.7	6.8147	4.0	243.0	16.8	395.56	13.45	19.7
0.14932	25.0	5.13	0.0	0.453	5.741	66.2	7.2254	8.0	284.0	19.7	395.11	13.15	18.7
0.08826	0.0	10.81	0.0	0.413	6.417	6.6	5.2873	4.0	305.0	19.2	383.73	6.72	24.2
0.04113	25.0	4.86	0.0	0.426	6.727	33.5	5.4007	4.0	281.0	19.0	396.90	5.29	28.0
0.04684	0.0	3.41	0.0	0.489	6.417	66.1	3.0923	2.0	270.0	17.8	392.18	8.81	22.6
0.14866	0.0	8.56	0.0	0.520	6.727	79.9	2.7778	5.0	384.0	20.9	394.76	9.42	27.5
0.10793	0.0	8.56	0.0	0.520	6.195	54.4	2.7778	5.0	384.0	20.9	393.49	13.00	21.7
0.06899	0.0	25.65	0.0	0.581	5.870	69.7	2.2577	2.0	188.0	19.1	389.15	14.37	22.0
0.34006	0.0	21.89	0.0	0.624	6.458	98.9	2.1185	4.0	437.0	21.2	395.04	12.60	19.2
0.29090	0.0	21.89	0.0	0.624	6.174	93.6	1.6119	4.0	437.0	21.2	388.08	24.16	14.0

Supervised machine learning - second example

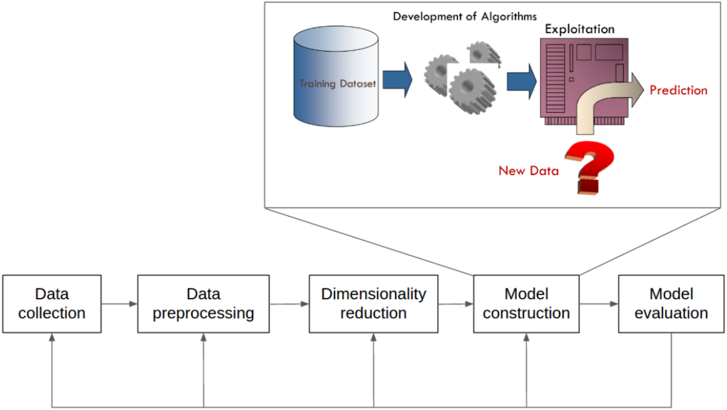
The Boston Housing problem



Classification or regression?

The ML design cycle

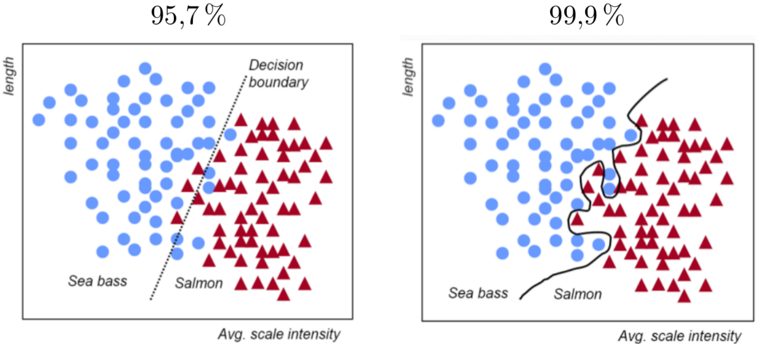
Model construction/training is just a single step in a much bigger process



Model evaluation and selection

- How to assess the quality of a trained model
- How to compare two different models
- Model validation and hyper-parameter tuning
- Different evaluation metrics: loss, accuracy, confusion matrix, ROC analysis, etc.

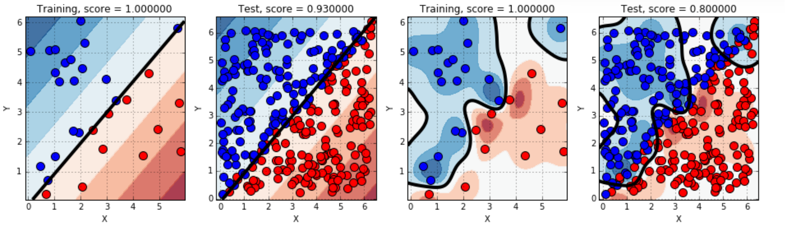
Model evaluation and selection - Example



(From Duda, Hart and Stork, *Pattern Classification*, 2001)

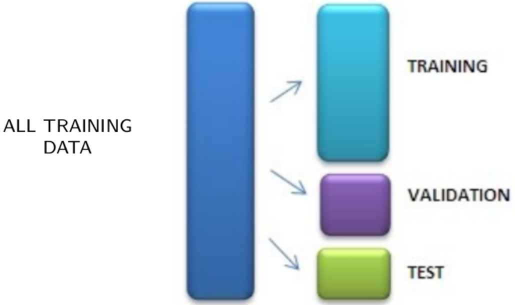
Which model is better?

Model complexity and generalization



- Complex models are able to better adapt to the training data
- *Overfitting*: too much adaptation to the training data may lead to a poor generalization

Training, validation, test



- Use different data for training and validating the model
- Early stoping

(Proper) Regularization

- Modify the loss function by introducing a term that penalizes model complexity
- LOSS $=$ ERROR $+$ COMPLEXITY

Review: Core topics

- DL is a subfield of ML that uses NNs to make predictions on data
- NNs are supervised, parametric models that learn from examples (classification, regression)
- Model training: tune the parameters in order to adapt to the training data
- Model validation: complexity, generalization, overfitting, regularization

Next

- Neural networks with a single neuron
- Linear regression
- Logistic regression (classification)