Machine learning algorithms Classic formulations

Carlos Perales-González[1]

[1]Universidad Loyola Andalucía

2020-2021

Overview

Categories in Machine Learning

- **Supervised learning**: pattern recognition. Algorithms: ELM, SVM, Decision trees, ...
  - *Classification*
  - *Regression*
- **Unsupervised learning**: data structure. Algorithms: K-means, PCA, ...
- **Reinforcement learning**: rewards to behaviors.

Deep learning is a field of Supervised Machine Learning.

Supervised Learning

Generally speaking, a predictor f is a function with vector of features $\boldsymbol{x} \in \boldsymbol{X}$ as input and $y_n \in \boldsymbol{Y}$ as output / target.

$$f(\boldsymbol{x}_n; \theta) \approx y_n,$$

The training process is a minimization of the Error function f over dataset $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$ where optimal parameters $\theta \in \Theta$ are tuned.

$$\begin{aligned} \min_{\theta \in \Theta} \quad & \text{Error}((f(\boldsymbol{x}_1; \theta), y_1), \ldots, (f(\boldsymbol{x}_N; \theta), y_N)) \\ \text{subject to:} \quad & \text{Model restrictions} \end{aligned} \tag{1}$$

Example: Extreme Learning Machine

ELM is a type of Neural Network where the weights in the hidden layer are predefined [2].

- **Prediction**:

$$f(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x})^{\prime}\beta \tag{2}$$

with

$$\boldsymbol{h}(\boldsymbol{x}) = (\phi(\boldsymbol{x}; \boldsymbol{w}_d, b_d),\ d = 1, \ldots, D), \tag{3}$$

and ϕ is the activation function, for example sigmoid,

$$\phi(\boldsymbol{x}; \boldsymbol{w}_d, b_d) = \frac{1}{1 + \exp(-(\boldsymbol{w}_d^{\prime} \cdot \boldsymbol{x} + b_d))}. \tag{4}$$

- **Training**:

$$\min_{\beta \in \mathbb{R}^D} (\boldsymbol{H}\beta - \boldsymbol{Y})^{\prime}(\boldsymbol{H}\beta - \boldsymbol{Y}) + C\beta^{\prime}\beta \tag{5}$$

Example: Support Vector Machine (I)

A plane that divides one class for another. There are several versions of this algorithm.

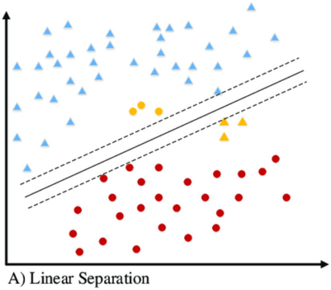

Figure: SVM linear and kernel for a binary classification problem

Support Vector Machine (II)

According to its linearity:

- Linear $\rightarrow f(\boldsymbol{x}) = sign(< v, \boldsymbol{x} > +b) \in \{-1, 1\}$.

$$< \boldsymbol{x}_n, \boldsymbol{x}_m >= \boldsymbol{x}_n^{'}\boldsymbol{x}_m \tag{6}$$

- Kernel $f(\boldsymbol{x}) = sign(< v, \phi(\boldsymbol{x}) > +b) \in \{-1, 1\}$. For example, gaussian kernel is

$$< \phi(\boldsymbol{x}_n), \phi(\boldsymbol{x}_m) >= k(\boldsymbol{x}_n, \boldsymbol{x}_m) = e^{-(\boldsymbol{x}_n - \boldsymbol{x}_m)^2} \tag{7}$$

Support Vector Machine (II)

Most popular versions

- Least Squares [4]

$$
\begin{aligned}
\min_{\boldsymbol{v}\in\mathbb{R}^K} \quad & \frac{1}{2} < \boldsymbol{v}, \boldsymbol{v} > + \frac{C}{2}\sum_{n=1}^{N} \epsilon_n^2 \\
\text{subject to:} \quad & y_n(< \boldsymbol{v}, \boldsymbol{x}_n > + b) = 1 - \epsilon_n, \ \forall n.
\end{aligned} \tag{8}
$$

- Quadratic Problem (classic) [1]

$$
\begin{aligned}
\min_{\boldsymbol{v}\in\mathbb{R}^K} \quad & \frac{1}{2} < \boldsymbol{v}, \boldsymbol{v} > + \frac{C}{2}\sum_{n=1}^{N} \epsilon_n \\
\text{subject to:} \quad & y_n(< \boldsymbol{v}, \boldsymbol{x}_n > + b) \geq 1 - \epsilon_n, \ \forall n. \\
& \epsilon_n \geq 0, \ \forall n.
\end{aligned} \tag{9}
$$

- The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code.
- It saves the results of the code already run.
- It allows to mix real text, equations and code.

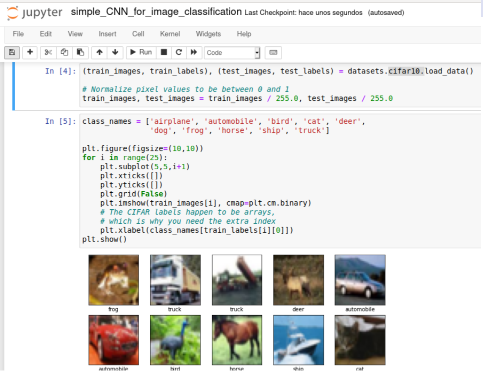

Figure: How Jupyter Notebook looks like.

Let's do some code!

Homework

- Create a virtual environment in Python 3.7.
- Install tensorflow, jupyter, ipython, numpy, pandas, sklearn.
- Explore example with `DecisionTreeClassifier` and `SVC` from `sklearn` library and *CIFAR10* dataset[3].

`https://github.com/cperales/machine-learning-lessons`

References I

Corinna Cortes and Vladimir Vapnik.
Support vector machine.
Machine learning, 20(3):273–297, 1995.

Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang.
Extreme learning machine for regression and multiclass classification.
IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 42(2):513–529, 2011.

References II

Alex Krizhevsky.
Learning Multiple Layers of Features from Tiny Images.
. . . Science Department, University of Toronto, Tech. . . ., 2009.

Johan AK Suykens and Joos Vandewalle.
Least squares support vector machine classifiers.
Neural processing letters, 9(3):293–300, 1999.