

# 1. INTRODUCTION

OpenNN is a software library written in C++ for advanced analytics. It implements neural networks, the most successful machine learning method.

- The main advantage of OpenNN is its high performance.
- This library outstands in terms of execution speed and memory allocation. It is constantly optimized and parallelized in order to maximize its efficiency.
- Some typical applications of OpenNN are business intelligence (customer segmentation, churn prevention...), health care (early diagnosis, microarray analysis...) and engineering (performance optimization, predictive maintenance...).

The documentation is composed by tutorials and examples to offer a complete overview about the library.

- The documentation can be found at the official OpenNN site.
- CMakeLists.txt are build files for CMake, it is also used by the CLion IDE.
- The .pro files are project files for the Qt Creator IDE, which can be downloaded from its site. Note that OpenNN does not make use of the Qt library.

OpenNN is developed by Artelnic, a company specialized in artificial intelligence.

# 2. PROJECTS

## 2.1. Examples

### 2.1.1. simple function regression

type:

```
./simple_function_regression
```

### 2.1.2. simple pattern recognition

type:

```
./simple_pattern_recognition
```

### **2.1.3. airfoil self noise**

type:

`./airfoil_self_noise`

### **2.1.4. airline passengers**

type:

`./airline_passengers`

### **2.1.5. breast cancer**

type:

`./breast_cancer`

### **2.1.6. iris plant**

type:

`./iris_plant`

### **2.1.7. logical operations**

type:

`./logical_operations`

### **2.1.8. pima indians diabetes**

type:

`./pima_indians_diabetes`

### **2.1.9. urinary inflammations diagnosis**

type:

`./urinary_inflammations_diagnosis`

#### **2.1.10. yacht hydrodynamics design**

type:

`./yacht_hydrodynamics_design`

#### **2.1.11. yacht hydrodynamics production**

type:

`./yacht_hydrodynamics_production`

#### **2.1.12. leukemia**

type:

`./leukemia`

#### **2.1.13. pollution forecasting**

type:

`./pollution_forecasting`

#### **2.1.14. temperature forecasting**

type:

`./temperature_forecasting`

#### **2.1.15. mnist**

type:

`./mnist`

### **2.2. Neural Turing Machine**

#### **2.2.1. PU**

type:

`./PU`

### 3. WORKFLOW

type:

```
rm -rf build  
mkdir build  
cd build
```

```
cmake ..  
make
```

### 4. CONCLUSION