Abstract

Computers have become increasingly present in our daily lives over the past years, thanks to several converging factors such as the development of highly efficient hardware in terms of storage and computational power capabilities, or the trivialisation of keeping portable computing devices - such as smartphones - on oneself at all times. This trend - already predicted decades ago by the founder of *ubiquitous computing* Mark Weiser - has led to an increase in unobtrusive and invisible wearable sensors, facilitating the process of acquiring data, and especially time-series ones. At the same time, remarkable progress in the development of even more effective machine learning techniques has been observed over the past decades, notably regarding approaches to learn relevant *features* to provide an abstracted representation of the data. The most notable phenomenon illustrating this trend is the fast rise in popularity of *deep learning*, which refers to machine learning using Deep Neural Networks (DNNs), and is reputed to yield remarkable performances when large amounts of data are available. The conjunction of those two trends has opened up various new possibilities of application as evidenced by the growing research community of ubiquitous computing.

Despite this favourable context, the application of machine learning approaches - and more specifically deep learning - still faces obstacles. The success of obtaining a proper machine learning model strongly hinges on the quality and quantity of data used to train it. On the one hand, most breakthroughs using DNNs over the past years have been made in application fields using images thanks to the availability of very large benchmark image datasets such as *ImageNet*. On the other hand, time-series data - which represent the most common type of data in ubiquitous computing application - remain scarce: data acquisition campaigns with wearable sensors are costly in terms of time and resources in practice, and the large diversity in types of time-series data acquired by different sensors makes the building of an equivalent of ImageNet for time-series difficult. While deep learning using time-series data has already been successfully applied in various contexts in past studies, their scope is usually limited to their specific application field, and not always rigorously compared with other state-of-the-art machine learning approaches.

In this context, this thesis proposes to explore the topic of using deep learning for time-series classification, which is one of the most commonly encountered machine learning problems in ubiquitous computing applications. It attempts in particular to provide elements of answer to two questions which have not been fully addressed in the literature: 1. Is it beneficial to use deep learning for time-series classification

over other traditional machine learning approaches when data are limited or scarce? 2. Is there a way to bypass data scarcity issues to enhance the performances of DNNs for time-series classification? The thesis is structured as follows: after an introduction presenting the context with more details and providing examples illustrating the difficulties of applying deep learning in practice, each question is investigated more closely in one dedicated Chapter.

The first part of the thesis (Chapter 2) explores the first of the two aforementioned questions in the context of sensor-based Human Activity Recognition (HAR). Sensor-based HAR has established itself as the most popular application of ubiquitous computing due to its numerous potential applications in real-life, the affordability and the pervasiveness of motion-based sensors, and the simplicity of data annotation which has led to relatively higher amounts of available data. The Chapter focuses more specifically on the question of how to obtain proper feature representations of the data for HAR. While abundant, the sensor-based HAR literature on this topic has so far been mostly scattered, with little efforts to compare the performances of the different approaches on a fair basis. In this Chapter, a comparative study between various state-of-the-art feature extraction approaches - supervised or unsupervised, centred around feature engineering or feature learning with DNNs - is detailed. An evaluation framework allowing a strict comparison is firstly defined and used to carry out extensive experiments on two sensor-based HAR benchmark datasets: *OPPORTUNITY* and *UniMiB-SHAR*. The results highlighted three main phenomena: the superiority of supervised feature learning approaches over unsupervised ones, the dominance of methods using deep learning to learn features over those relying on manual feature engineering in all tested configurations, and the effectiveness of hybrid DNN models combining convolutional and recurrent layers. The study therefore validates the effectiveness of deep learning for time-series classification in the context of sensor-based HAR.

The second part of this thesis (Chapter 3) focuses on time-series *transfer learning* to bypass data scarcity issues and enhance the performances of DNNs for time-series classification. Transfer learning has become state-of-the-art when image modalities are involved due to its proven and repeated ability to let DNNs yield better classification performances than without transfer. Time-series transfer learning however still remains relatively unexplored due to lower quantities of available time-series data, and large differences in data formats depending on which sensors were used to acquire them. Several transfer methods have been proposed in the literature, but their scope is usually limited either to specific application fields, to strict conditions of similarity between the *source* and *target domains*, or to single-channel time-series data. A new time-series transfer learning approach addressing those issues was developed in the frame of this thesis. It proposes to use sensor modality classification as an auxiliary task to learn general transferable time-series features. Time-series datasets related to various applications of ubiquitous computing are firstly aggregated to build a source domain. A DNN processing single-channel data (sDNN) is trained to recognise sensor modalities, and its weights are then transferred to a DNN architecture processing multichannel time-series (mDNN)

on the target domain. The mDNN is finally fine-tuned on the target domain to solve the target problem. This process can be applied to any target domain using multichannel time-series data no matter how many channels were used, thus addressing the limitations of existing approaches. Experiments carried out for two distinct target applications of ubiquitous computing - sensor based HAR and Emotion Recognition - showed that the proposed transfer approach yields classification performance improvements compared to not using any transfer. Such results indicate its strong potential for applications of ubiquitous computing relying on time-series classification.

Zusammenfassung

Computer sind in den letzten Jahren in unserem täglichen Leben, dank mehrerer konvergierender Faktoren wie der Entwicklung hocheffizienter Hardware in Bezug auf Speicher- und Rechenleistung oder der Möglichkeit, tragbare Computergeräte - wie Smartphones - ständig bei sich zu haben, immer präsenter geworden. Dieser Trend - bereits vor Jahrzehnten von Mark Weiser, dem Begründer des Ubiquitous Computing, vorausgesagt - hat zu einer Zunahme von unauffälligen und unsichtbaren tragbaren Sensoren geführt, die die Erfassung von Daten, insbesondere von Zeitreihen, erleichtern. Gleichzeitig wurden in den letzten Jahrzehnten bemerkenswerte Fortschritte bei der Entwicklung noch effektiverer maschineller Lerntechniken beobachtet, insbesondere in Bezug auf Ansätze zum Erlernen relevanter Merkmale, um eine abstrahierte Darstellung der Daten zu erhalten. Besonders hervorgehoben wird dies durch den aktuellen Trend und schnellen Anstieg der Popularität von Deep Learning, das auf maschinelles Lernen unter Verwendung von Deep Neural Networks (DNNs) beruht und dem nachgesagt wird bemerkenswerte Leistungen zu erzielen, falls große Datenmengen verfügbar sind. Die Verbindung dieser beiden Trends hat verschiedene neue Anwendungsmöglichkeiten eröffnet, wie die wachsende Forschungsgemeinschaft des Ubiquitous Computing beweist.

Trotz dieser günstigen Ausgangssituation stößt die Anwendung von Ansätzen des maschinellen Lernens - und insbesondere des Deep Learnings - immer noch auf Hindernisse. Der Erfolg eines geeigneten Modells hängt stark von der Qualität und Quantität der zum Training verwendeten Daten ab. Einerseits wurden die meisten Durchbrüche mit DNNs in den letzten Jahren in Anwendungsbereichen erzielt, die Bilder verwenden, dank der Verfügbarkeit von sehr großen Benchmark-Bilddatensätzen wie *ImageNet*. Auf der anderen Seite sind Zeitreihendaten - die die häufigste Art von Daten in Ubiquitous-Computing-Anwendungen darstellen - nach wie vor rar: Datenerfassungskampagnen mit tragbaren Sensoren sind in der Praxis zeitaufwendig- und ressourcenintensiv, und die große Vielfalt an Arten von Zeitreihendaten, die von verschiedenen Sensoren erfasst werden, macht den Aufbau eines Äquivalents zu ImageNet für Zeitserien schwierig. Während Deep Learning unter Verwendung von Zeitreihendaten in vergangenen Studien bereits erfolgreich in verschiedenen Kontexten angewendet wurde, ist ihr Umfang in der Regel auf ihr spezifisches Anwendungsgebiet beschränkt und wird nicht immer rigoros mit anderen State-of-the-Art-Maschinenlernansätzen verglichen.

In diesem Zusammenhang schlägt die vorliegende Arbeit vor, das Thema der Verwendung von Deep Learning für die Klassifizierung von Zeitreihen zu erforschen,

welches eines der am häufigsten auftretenden maschinellen Lernprobleme in Ubiquitous-Computing-Anwendungen darstellt. Es wird insbesondere versucht, Antworten auf die foldgenden zwei Fragen zu liefern, die in der Literatur noch nicht vollständig behandelt wurden: 1. Ist es vorteilhaft, Deep Learning für die Zeitreihenklassifizierung gegenüber anderen traditionellen maschinellen Lernansätzen zu verwenden, wenn die Daten begrenzt oder knapp sind? 2. Gibt es eine Möglichkeit, das Problem der Datenknappheit zu umgehen, um die Leistung von DNNs für die Zeitreihenklassifikation zu verbessern? Die Arbeit ist wie folgt strukturiert: Nach einer Einleitung, die den Kontext detaillierter darstellt und Beispiele liefert, die die Schwierigkeiten bei der Anwendung von Deep Learning in der Praxis illustrieren, werden die zwei Kernfragen in jeweils eigenen Kapiteln genauer untersucht.

Der erste Teil der Arbeit (Kapitel 2) untersucht die erste der beiden oben genannten Fragen im Kontext der sensorbasierten Human Activity Recognition (HAR). Sensorbasierte HAR hat sich aufgrund der zahlreichen Anwendungsmöglichkeiten im realen Leben, der Erschwinglichkeit und der weiten Verbreitung von bewegungsbasierten Sensoren sowie der Einfachheit der Datenannotation, die zu relativ großen Datenmengen geführt hat, als die populärste Anwendung des Ubiquitous Computing etabliert. Dieses Kapitel konzentriert sich speziell auf die Frage, wie man geeignete Merkmalsrepräsentationen der Daten für HAR erhält. Die sensorgestützte HAR-Literatur zu diesem Thema ist zwar reichlich vorhanden, aber bisher nur verstreut aufzufinden und es gibt kaum Bemühungen, die Leistungen der verschiedenen Ansätze auf einer fairen Basis zu vergleichen. In diesem Kapitel wird eine vergleichende Studie zwischen verschiedenen State-of-the-Art-Merkmalsextraktionsansätzen - überwacht oder unüberwacht, mit Schwerpunkt auf Feature-Engineering oder Feature Learning mit DNNs - detailliert beschrieben. Ein Evaluierungsrahmen, der einen strengen Vergleich ermöglicht, wird zunächst definiert und zur Durchführung umfangreicher Experimente an zwei sensorbasierten HAR-Benchmark-Datensätzen verwendet: *OPPORTUNITY* und *UniMiB-SHAR*. Die Ergebnisse heben drei Hauptphänomene hervor: die überlegenheit von überwachten Feature-Learning-Ansätzen gegenüber unbeaufsichtigten, die Dominanz von Methoden, die Deep Learning zum Lernen von Features verwenden, gegenüber solchen, die sich auf manuelles Feature-Engineering in allen getesteten Konfigurationen verlassen, und die Effektivität von hybriden DNN-Modellen, die convolutional und recurrent Schichten kombinieren. Die Studie validiert daher die Effektivität von Deep Learning für die Zeitreihenklassifikation im Kontext von sensorbasiertem HAR.

Der zweite Teil dieser Arbeit (Kapitel 3) konzentriert sich auf *transfer learning* für Zeitreihendaten, um die Probleme der Datenknappheit zu umgehen und die Leistung von DNNs für die Zeitreihenklassifikation zu verbessern. Transfer Lerning hat sich als State-of-the-Art Ansatz etabliert, insbesondere wenn es um Bildmodalitäten geht, da es bewiesen wurde, dass DNNs mit der Methodik bessere Klassifizierungsergebnisse erzielen können als ohne. Zeitreihen Transfer Learning ist jedoch noch relativ unerforscht, da weniger Zeitseriendaten zur Verfügung stehen und es große Unterschiede in den Datenformaten gibt, je nachdem, welche Sensoren

für die Datenerfassung verwendet wurden. In der Literatur wurden mehrere Transfermethoden vorgeschlagen, aber ihr Umfang ist in der Regel entweder auf bestimmte Anwendungsbereiche, auf strenge ähnlichkeitsbedingungen zwischen dem Quell- und dem Zielgebiet oder auf einkanalige Zeitreihendaten beschränkt. Im Rahmen dieser Arbeit wurde ein neuer Zeitreihen Transfer Learning Ansatz entwickelt, der diese Probleme adressiert. Es wird vorgeschlagen, die Klassifizierung der Sensormodalität als Hilfsaufgabe zu verwenden, um allgemeine übertragbare Zeitserienmerkmale zu lernen. Zeitseriendatensätze, die sich auf verschiedene Anwendungen des Ubiquitous Computing beziehen, werden zunächst aggregiert, um eine Quelldomäne zu bilden. Ein DNN, das einkanalige Daten verarbeitet (sDNN), wird trainiert, um Sensormodalitäten zu erkennen, und seine Gewichte werden dann auf eine DNN-Architektur übertragen, die mehrkanalige Zeitreihen (mDNN) in der Zieldomäne verarbeitet. Das mDNN wird schließlich auf der Zieldomäne feinabgestimmt, um das Zielproblem zu lösen. Dieser Prozess kann auf jede Zieldomäne mit Mehrkanal-Zeitreihendaten angewendet werden, unabhängig davon, wie viele Kanäle verwendet wurden, und behebt so die Einschränkungen bestehender Ansätze. Experimente, die für zwei verschiedene Zielanwendungen des Ubiquitous Computing durchgeführt wurden - sensorbasierte HAR und Emotionserkennung - haben gezeigt, dass der vorgeschlagene Transfer-Ansatz eine Verbesserung der Klassifikationsleistung im Vergleich zum Verzicht auf einen Transfer liefert. Diese Ergebnisse weisen auf das große Potenzial des Ansatzes für Anwendungen des Ubiquitous Computing hin, die auf der Klassifizierung von Zeitreihen beruhen.

Chapter 1

Introduction

The past years have seen a tremendous rise in machine learning applications due to multiple converging factors such as the development of computer hardware with increased processing power and data storage capacities, or societal factors like the democratisation of electronic devices - such as smartphones - taking an increasing place in everyone's daily life. Those trends have received an increasing attention from researchers studying how to take leverage of the increasing availability of such devices, leading to the emergence of the research field of *ubiquitous computing* - term firstly coined by Mark Weiser who predicted those trends three decades ago [1].

Ubiquitous computing is a field centred around the study of *wearable devices*, which refers to electronic devices that can easily be carried by its user. It encompasses a large variety of topics such as the design of practical, unobtrusive and powerful wearable devices, the establishment of communication protocols to retrieve their data in optimal conditions, or the processing of the aforementioned data for complex applications requiring some highly-abstract reasoning. It consequently has a significant overlap with various other research fields. One of them is *machine learning* which refers to an ensemble of techniques to train computers using sensor data to perform complex tasks requiring high-level reasoning, and is widely used to create systems able to provide meaningful applications using the collected data by the wearable devices.

Machine learning heavily relies on the availability of data which should be in quantities as large as possible to train models in optimal conditions. For this reason, it has especially benefitted from the development of ubiquitous computing, and more specifically from the explosion in popularity of wearable sensors of all types such as RGB cameras embedded in smartphones or other devices providing physiological or behavioural data from the wearer. The growing availability of data and the increasing simplicity of sharing them all over the world has led to the emergence of more powerful machine learning approaches over time. The most widely known example of those are Artificial Neural Networks (ANNs), which are the cornerstone of the now famous and pervasive *deep learning*. Despite having been discovered more than 50 years ago, ANNs truly started to explode in popularity only a few years ago following the appearance of increasingly powerful computational units which simplified their training process, and after ANNs significantly outperformed the state-of-the-art for RGB image classification in the *ImageNet Large Scale Visual Recognition Challenge 2012* (ILSVRC 2012) [2]. After their initial success, the machine learning research community was swept away by the deep learning hype, in particular in the field of image processing where data are abundant thanks to past efforts for large-scale data collection and annotation such as the *ImageNet* dataset [3]. After a growing number of studies reported performances never attained previously for several key image processing problems like image classification [2, 4], image segmentation [4], object detection in images [5, 6, 7], etc., ANNs have established themselves as the main state-of-the-art approach for machine learning in the field over the past years.

In the light of the achievements obtained by deep learning on image modalities, the machine learning community has naturally drawn its attention to ANNs for the processing of other types of data. The most important one of those are *time-series data* which designate one-dimensional series of data readings acquired successively in time. Time-series data have a paramount importance in ubiquitous computing because of how sensors measuring one-dimensional series of values are commonly used in ubiquitous computing applications. Numerous past works of the literature have shown promising results for ANNs in several application fields of ubiquitous computing such as sensor-based Human Activity Recognition [8, 9, 10]. But the relative scarcity of time-series datasets (compared to image ones) coupled to the large data requirements to properly train an ANN have limited the application of deep learning on time-series in practice. The objective of this thesis is to provide an additional contribution to the scientific field of time-series deep learning by rigorously comparing its performances to other state-of-the-art machine learning methods and exploring approaches to make its practical application easier.

The rest of the Section is structured as follows: Section 1.1 firstly provides more details about the fundamental concepts surrounding the context of this thesis such as ubiquitous computing (Section 1.1.1), time-series classification (Section 1.1.2) and deep learning (Section 1.1.3). Section 1.2 then expands on the motivation behind this thesis and provides two illustrative examples taken from the experience of the author of this thesis. Section 1.3 explicitly states the contribution of this thesis. Finally, Section 1.4 presents the overall layout of the thesis.

1.1 Fundamental concepts

This Section presents a general overview of the fundamental concepts associated with the context in which this thesis takes place. Section 1.1.1 defines the concept of ubiquitous computing in a more detailed way. Section 1.1.2 explains how machine learning can be applied to ubiquitous computing applications. Finally, Section 1.1.3 provides more details on deep learning - i.e. machine learning with deep ANNs - which supplanted traditional machine learning in many application fields.

1.1.1 Ubiquitous computing

Past years have seen notable technological progress with the explosion in numbers of increasingly powerful, small computers and other smart devices. A prime example illustrating this phenomenon is the constantly increasing popularity and pervasiveness of smartphones. Invented in 1992 by IBM (USA) and popularised to the general public with the release of the first iPhone (Apple Inc., USA) in 2007, smartphones have continuously evolved to provide not only an increasing amount of services, but also of portable sensors. It is now estimated that 3.5 billion people in the world own a smartphone as of 2020, representing a reach of 45.04% of the world population less than three decades after their invention[1].

The remarkably fast spread of increasingly complex and powerful devices is often seen as a consequence of *Moore's law*, which refers to an observation made in 1965 by the then CEO of Intel Gordon Moore who predicted that the number of transistors in an integrated circuit would double every two years, leading to an exponential increase in computational power [11]. Initially estimated to last for only one decade, this trend has remained true until today. While Moore's law validity will necessarily come to an end at some point, experts have not been able to accurately predict when it would happen, with the current estimations placing it somewhere in the upcoming decade[2].

The implications of Moore's law on technological advances and their associated societal changes were understood fairly early on, with the American computer scientist Mark Weiser introducing for the first time the term ubiquitous computing to refer to computers becoming pervasive in everyone's daily life [1] in 1988. Later on, Weiser specified his vision by introducing four fundamental principles defining goals for ubiquitous-computing-related applications [12]:

1. Computers should help their users.
2. While providing their assistance, computers should be quiet and invisible.
3. Computers should "extend the unconscious" of their users, by helping them to make decisions intuitively.
4. Technology should create calm.

Over time, a community aggregated to work on the development of systems following Weiser's four founding principles. The *ubiquitous computing* research area nowadays includes a very wide array of fields such as hardware development to obtain increasingly small and performing sensors, science of design to develop unobtrusive and user-friendly devices, communication and network analysis to develop protocols and methods allowing better and more reliable data transmission

[1]Statistics taken from `https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/` (last accessed on 24/05/2021)

[2]Information taken from `https://www.androidauthority.com/moores-law-smartphones-1088760/` (last accessed on 24/05/2021)

or data analysis and machine learning to detect patterns in the data obtained from the wearable devices.

Ubiquitous computing has in particular very close connections with artificial intelligence, which refers to the science of training computers to perform complex tasks, and of which data science and machine learning are sub-fields. Weiser's first and third principles both suggest that computers should assist their users in their decision making, which most of the time involves highly-abstract reasoning. The development of an ubiquitous computing system therefore requires machines to be able to understand and extend such reasoning, which can be provided by machine learning techniques. Since the latter rely on large quantities of data to train the mathematical models required to achieve this objective, the recent expansion of ubiquitous computing has been seen as an opportunity by the machine learning community to develop new means of getting data via wearable devices, leading to an explosion in the number of intelligent systems for many applications such as health monitoring [13], interactive learning systems [14], and assisted living systems for vulnerable people [15].

1.1.2 Machine learning and time-series classification

Machine learning refers to the study of mathematical algorithms which can be used to teach computers to perform complex tasks usually requiring highly-abstract reasoning to be solved. The fundamental way machine learning approaches this goal is by analysing data related to the task considered to train a mathematical model. Once learned from the data, such model is then given to the machine to be re-used on "unseen" real-life data, i.e. data different from the training one used previously to train the model (also referred to as *training data*). It should be noted that machine learning is notably reliant on large quantities of data to work properly, since more of them increases the likelihood that the trained model will learn proper assumptions while generalising well on "unseen" data (which is often referred to as the bias-variance trade-off [16]). Initially evoked for the first time by the American computer scientist A. L. Samuel in 1959 [17], machine learning has become increasingly prevalent over the past decades due to its important overlap with other rising fields of study such as algorithmic, statistics, computer science, data analysis or mathematical optimisation, and due to the large range of applications it could solve. It is for instance widely used in ubiquitous-computing-related studies as a means to provide computers with the "intelligence" required to assist their users in making decisions, thus fulfilling Weiser's first and fourth principles.

Machine learning approaches can be classified into several categories depending on which type of data they require to properly train their models, and how they use them. Three main families of machine learning approaches are usually distinguished:

- **Supervised learning** which relies on using data annotated with *labels* indicating the desired model output for each example of the dataset.

- **Unsupervised learning** which attempts to detect patterns in the data without using any labelling information.
- **Semi-supervised learning** which lies in-between supervised and unsupervised learning and attempts to train models using incompletely labelled datasets.

The task of acquiring labels to annotate data - while appearing simple - can prove to be very complicated in real-life due to multiple reasons ranging from labels simply being difficult to obtain (e.g. wearable-emotion recognition described in Section 1.2.2) to the quantity of data being too large to annotate in a reasonable amount of time or resources (e.g. the 14 million images of the *ImageNet* dataset [3]). While labels are not needed in an unsupervised learning context, supervised learning has remained the most widely used category of machine learning approaches until now because of the notably better performances it has obtained compared to semi-supervised or unsupervised learning techniques in most application domains.

When a dataset and its associated labels are available, supervised learning defines and uses a mathematical framework to solve a specific problem for a particular application. Most supervised learning methods do this by translating the problem either into a *regression* or *classification* problem, where regression approaches attempt to train a model to associate its inputs with a specific output value, while classification ones aim at obtaining models able to associate their input data with categorical outputs. Regression and classification approaches are fairly similar in terms of solving procedures or available algorithms. Both aim at approximating a function $\phi : x \to y$ which maps input data samples x to continuous and discrete labels y, respectively. This is done by computing *features* which refer to values computed on the input data that can be used to properly represent them in an abstract way for the considered problem to solve. Features allow to associate each data sample x with a *feature vector* $\mathbf{f}(\mathbf{x}) = \{f_1(x), f_2(x), ..., f_n(x)\} \in \mathcal{R}^n$ where $n \in \mathcal{N}^*$ is the number of computed features. A mathematical model is then trained in the *feature space* $\mathcal{R}^n$ to match each feature vector $\mathbf{f}(\mathbf{x})$ with its associated label y.

In practice, regression and classification problems however differ in terms of difficulty, with classification models usually being easier to train than regression ones. While any machine learning approach requires large quantities of training data and associated labels, regression approaches tend to need more of them than classification in order to precisely approximate their continuous targets. Classification has therefore become the preponderant framework in most machine learning applications.

The reliance of machine learning on large quantities of available data had the consequence of splitting the research community depending on which sensor modalities they use to provide input data for their models. Machine learning approaches remain the same independently of what type of data is used in the training set. In practice however, it has been observed that the relevance of the trained models increases substantially the more data is available. As a consequence, a divide has effectively appeared over the past decade between researchers working with image modalities - where the quantity of available data is plentiful due to the pervasiveness of cameras and some initiatives to build very large scale datasets like *ImageNet* [3] - and those working with other sensor modalities. Ubiquitous computing researchers tend to fall into the second category as the intrusiveness (and sometimes obtrusiveness) of cameras could be considered as a contradiction of Weiser's second principle of "computers being invisible to the user". As a result, ubiquitous computing applications have favoured the use of wearable sensors providing data values sequentially in time, also commonly referred to as *time-series data*.

In this context, time-series classification has become an important topic for sensor-based applications, with abundant works in the past literature trying to propose high performing classification approaches in many application fields such as Human Activity Recognition, sensor-based emotion and pain recognition (respectively discussed in Sections 2, 1.2.2 and 1.2.3). Mirroring the trends in the image processing research community, time-series classification has in particular been strongly impacted by the rise of deep learning, which has established itself as the new baseline approach over the past years.

1.1.3 Deep learning

Deep learning is a term introduced by the American artificial learning professor Rina Dechter in 1986 [18] to refer to machine learning using deep *Artificial Neural Networks* (ANNs).

ANNs are a class of mathematical models whose principles are loosely based on how biological neurons in the human brain work. They consist of a set of interconnected *artificial neurons*, where each artificial neuron is a very simple non-linear computational unit as shown in Figure 1.1. Each neuron takes a multidimensional input $\mathbf{x} = \{x_1, x_2, ..., x_n\} \in \mathcal{R}^n$ with $n \in \mathcal{N}^*$ and outputs a single value $y \in \mathcal{R}$ using the following equation:

$$y = \sigma(\sum_{k=1}^{n} w_k x_k) + b$$

where σ is a non-linear function referred to as *activation function*, $\forall k \in \{1, 2, ..., n\}, w_k \in \mathcal{R}$ are internal parameters referred to as *neural weights* and $b \in \mathcal{R}$ is an offset parameter called *neural bias.*

In an ANN, artificial neurons are organised in a layer-wise structure with the outputs of the neurons of one layer being used as inputs of the neurons of the next layer as shown in Figure 1.2. A layer of an ANN with $n_{in} \in \mathcal{N}^*$ inputs and $n_{out} \in \mathcal{N}^*$ outputs can therefore be mathematically represented by the formula:

$$\mathbf{y} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$$

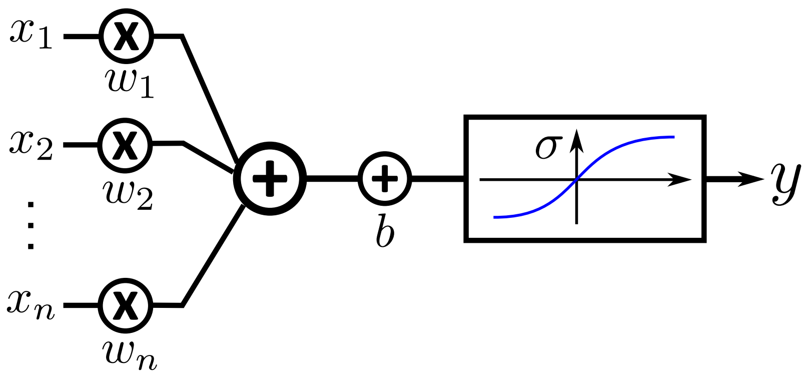

Figure 1.1: Graph of an artificial neuron: $n \in \mathcal{N}^*$ inputs $\{x_1, x_2, ..., x_n\} \in \mathcal{R}^n$ are multiplied by n weights $\{w_1, w_2, ..., w_n\} \in \mathcal{R}^n$, summed together, added to a bias $b \in \mathcal{R}$ and then sent throught a non-linear function σ to provide an output value $y \in \mathcal{R}$.

where $\mathbf{y} \in \mathcal{R}^{n_{out}}$ is the vector of the layer neural outputs, $\mathbf{x} \in \mathcal{R}^{n_{in}}$ is the vector of the layer inputs, $\mathbf{W} \in \mathcal{R}^{n_{out} \times n_{in}}$ is the matrix of weights connecting the neurons to the ones of the previous layer and $\mathbf{b} \in \mathcal{R}^{n_{out}}$ is the vector containing the biases of the neurons of the layer. Similarly, an ANN stacking $N \in \mathcal{N}^*$ layers containing $n^{(l)} \in \mathcal{N}^*$ neurons each (for $1 \leq l \leq N$) can be mathematically represented as a series of composite activation functions:

$$\mathbf{y}^{(l)} = \begin{cases} \sigma(\mathbf{W^{(l)}x} + \mathbf{b^{(l)}}) & \text{if } l = 1 \\ \sigma(\mathbf{W^{(l)}y}^{(l-1)} + \mathbf{b^{(l)}}) & \text{if } 2 \leq l \leq N \end{cases}$$

where $\mathbf{y}^{(l)} \in \mathcal{R}^{n^{(l)}}$, $\mathbf{W^{(l)}} \in \mathcal{R}^{n^{(l)} \times n^{(l-1)}}$, $\mathbf{b^{(l)}} \in \mathcal{R}^{n^{(l)}}$ respectively refer to the output, weight matrix and biases of layer $l \in \{1, 2, ..., N\}$, and x to the input of the network.

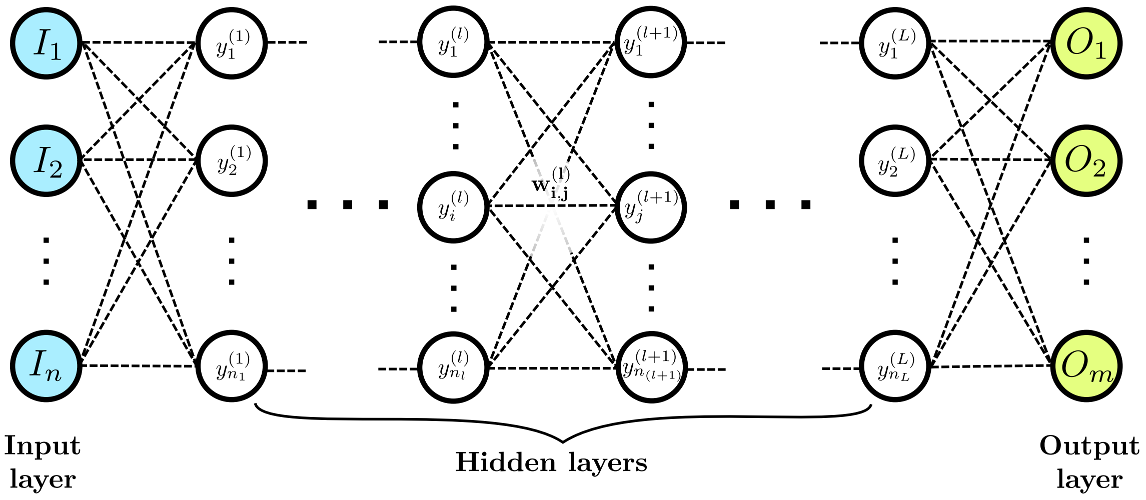

Figure 1.2: Example of an artificial neural network with $n \in \mathcal{N}^*$ inputs, $m \in \mathcal{N}^*$ outputs and $L \in \mathcal{N}^*$ layers. Each layer l has $n_l \in \mathcal{N}^*$ neurons (for $1 \leqslant l \leqslant L$). $y_i^{(l)}$ and $w_{i,j}^{(l)}$ respectively designate the output of the i^{th} neuron in the l^{th} layer and the weight connecting the i^{th} neuron on layer l to the j^{th} neuron of layer $l+1$.

The first and last layers of an ANN are respectively called *input* and *output layers*, while intermediate ones are referred to as *hidden layers*. An ANN is considered to be *deep* as long as it has at least two hidden layers, and is in that case referred to as *Deep Neural Network* (DNN). ANNs can be used for regression or classification purposes by adapting the size and activation functions of their output layer. For classification in particular, the common practice consists of setting the number of output neurons to the number of classes and using a *softmax* activation function [19]. Each neuron of a softmax output layer yields a value between 0 and 1 and the sum of all softmax outputs is equal to 1. The output of one softmax neuron can therefore be assimilated to a probability score of the ANN input data being associated to its corresponding class.

The *training of an ANN* (or DNN) refers to the process of fine-tuning the internal parameters of the model - weights and biases of the neurons of all layers - so that it can produce the desired outputs on its output layer. Neural weights and biases are usually initialised to values following an initialisation scheme that proved its effectiveness in practice, such as the *Glorot* initialisation strategy [20] which picks values at random in an interval whose boundaries depend on the sizes of the current and next layers for each neuron. Weights and biases are then iteratively updated during the training process. This is performed by firstly defining a *loss function $\mathcal{L}$* which computes the errors between the expected and neural outputs, and then minimising the loss function using a mathematical optimisation approach. For ANNs, the most commonly used method is the *gradient descent approach* which consists in iteratively updating each internal parameter $\theta \in \mathcal{R}$ of a given ANN (i.e. weight or bias) in the opposite direction of the derivative of the loss function with respect to θ, i.e.

$$\forall \theta \, , \, \theta \leftarrow \theta - \lambda \frac{\partial \mathcal{L}}{\partial \theta}(x)$$

where $\lambda \in \mathcal{R}^{+*}$ is a parameter to control the speed of updates called *learning rate* and $x \in \mathcal{R}^n$ an input example of the training set. The updating process is repeated for all samples x of the training set to complete one *epoch*. For ANNs, $\frac{\partial \mathcal{L}}{\partial \theta}(x)$ can only be directly computed if θ is a parameter belonging to the output layer by using the chain rule of derivation. The updates to the parameters belonging to the other layers are performed by using the *backpropagation algorithm* [21] which back-propagates the loss to the neurons of the previous layers using the chain rule of derivation. More in details, using the following notations:

- $N \in \mathcal{N}^*$ number of layers of the ANN
- $n^{(l)} \in \mathcal{N}^*$ number of neurons of layer l (with $1 \leq l \leq N$)
- $\mathcal{L}$ loss function
- σ activation function (assumed to be the same for all neurons of the ANN)
- $w_{ij}^{(l)} \in \mathcal{R}$ weight between the i^{th} neuron of layer $l-1$ and the j^{th} neuron of layer l (with $2 \leq l \leq N$)

- $y_j^{(l)} \in \mathcal{R}$ output of the j^{th} neuron of layer l
- $b_j^{(l)} \in \mathcal{R}$ bias of the j^{th} neuron of layer l
- $s_j^{(l)} \in \mathcal{R}$ weighted sum of inputs of the j^{th} neuron of layer l, i.e. $y_j^{(l)} = \sigma(s_j^{(l)} + b_j^{(l)})$

it can be shown that

$$\frac{\partial \mathcal{L}}{\partial w_{ij}^{(l)}}(x) = s_i^{(l-1)} \delta_j^{(l)}$$

with

$$\delta_j^{(l)} = \begin{cases} \frac{\partial \sigma(s_j^{(l)})}{\partial s_j^{(l)}} \frac{\partial \mathcal{L}}{\partial y_j^{(l)}}(x) & \text{if } l = N \\ \frac{\partial \sigma(s_j^{(l)})}{\partial s_j^{(l)}} \sum_{k=1}^{n^{(l)}} w_{jk} \delta_k^{(l+1)} & \text{otherwise.} \end{cases}$$

Neural biases are updated in a similar way as weights by the backpropagation algorithm by considering them as a neural weight associated with a constant input of 1.

ANNs are fairly old models whose origins are usually reported to go back as far as 1943, when the American scientists Warren McCulloch and Walter Pitts proposed a simplified version of the artificial neuron called Threshold Logic Unit [22]. This idea was later on expanded by various researchers such as the American psychologist F. Rosenblatt who introduced the *perceptron* model in 1958 [23] which served as a basis for the current modelling of artificial neurons, or the Ukrainian mathematician A. Ivakhnenko who was the first to propose a network structure featuring several layers in 1967 [24]. DNNs were however dismissed for a long time because they were considered as too complex to train properly. Methods to train DNNs based on gradient descent approaches had been found as early as 1974 [25] but could not be applied in practice due to high complexity caused by the high number of parameters in an ANN, associated with hardware-related limitations. It is only during the last decade that the training of ANNs - and more specifically DNNs - has become practical with researches highlighting the possibility to take leverage of high computational power units such as Graphics Processing Units (GPUs) to properly apply the backpropagation algorithm to large DNNs [26]. In the wake of those findings, deep learning started to yield state-of-the-art performances largely outperforming other traditional machine learning approaches, in particular in the field of image processing. The beginning of the deep learning hype is considered to coincide with the DNN *AlexNet* winning the ImageNet Large Scale Visual Recognition Challenge in 2012 (ILSVRC2012) by a very large margin over other machine learning methods [2]. Since then, DNNs have progressively established themselves as the main machine learning approach when image modalities are involved, and their use is now generalising to other types of modalities too, including time-series.

In the wake of the popularisation of deep learning, the reasons for the success and remarkable performances of DNNs have been thoroughly investigated in past researches. Because of the complexity of the mathematical concepts surrounding such models (analysis of complex composite functions), deep learning research has been notably lacking a solid mathematical framework until now. As a result, most works in the literature have remained fairly application-oriented so far. They have, however, highlighted a few key aspects of DNNs:

- **Strong representative power:** in 1989, the *Universal Approximation Theorem* was demonstrated [27], which states that any basic ANN with a single hidden layer containing an arbitrarily large (but finite) number of neurons can approximate any continuous function taking its input values on compact subsets of $\mathcal{R}^n$, under the assumption that the activation function is not polynomial. The implications of such findings are very impactful as they suggest that ANNs could be theoretically used to solve a very large variety of problems. In practice, it was shown that the number of neurons required for a single-hidden layer ANN to work properly might be too large to be practical depending on the complexity of the function to approximate [28]. Past works have nevertheless shown that using DNNs instead of shallow models could still yield good performances, largely outperforming traditional machine learning approaches in several application domains - in particular related to image processing - such as image classification [2, 7] and object detection [5, 4, 7].

- **Flexibility and modularity:** past researches showed that the traditional architecture of DNNs could be altered to better fit some specific application problems without fundamentally changing the mathematical background behind deep learning (gradient-descent-based training procedure especially). This has in particular led to the emergence of Convolutional Neural Networks (CNNs) pioneered by Kunihiko Fukushima [29] and Yann LeCun [30] for image-based applications, or of Recurrent Neural Networks (RNNs) whose first successful practical application was reported for speech recognition by Sepp Hochreiter [31]. More details about both architectures will be reported in Section 2.3.2. Alterations to the traditional DNN structure explored in the past literature have also included combinations of different types of DNNs [2, 8].

- **Feature learning:** a study carried out by Matthew Zeiler in 2013 [32] highlighted an interesting property of CNNs by showing that each neuron in the convolutional layers of the *AlexNet* model trained for image classification on *ImageNet* [2] was detecting a specific visual pattern whose complexity increased the deeper the layer was (e.g. straight lines or edges for neurons in the first layers, specific object parts for neurons of the deeper layers). Such findings suggest that each neuron learns a specific feature on the input data whose level of abstraction is higher the deeper the layer this neuron belongs to is. Other studies corroborated such behaviour, showing in particular that it was also observed for other types of DNNs for applications other than image classification such as speech recognition from audio data [33]. As a result, deep learning has progressively taken the place of old traditional approaches relying on asking domain experts to suggest features for a specific classification problem in such application fields.

- **Scalability with big data:** DNNs have shown to benefit from a large number of training examples more than other traditional machine learning models for regression or classification. Traditional approaches such as Support Vector Machines (SVM) [34] or Random Forests (RF) [35] for instance both have a training complexity at least quadratic with their number of training examples, and therefore cannot be trained under reasonable amounts of time on large datasets in practice. The compatibility of deep learning to the current era of big data - where large amounts of data are increasingly easily shared and stored - has contributed to the rise of its popularity over the past years.

Despite the current overwhelming popularity of DNNs in the machine learning community, the application of deep learning still remains confronted with some difficulties in practice, like the need for powerful computational resources to train complex models on large datasets in a reasonable amount of time (e.g. the DNN *VGG-Net* trained for ILSVRC2014 took between two to three consecutive weeks to be trained on four NVIDIA Titan Black GPUs [36]). The lack of rigorous mathematical framework surrounding DNNs also causes some deep learning aspects to still remain obscure as of today. Some topics such as how to optimise DNNs hyper-parameters (e.g. number of layers, number of neurons per layer, choice of the optimiser and learning rate, etc.) [37, 38, 39] or to interpret the decisions process of a trained DNN [40, 41] still remain active nowadays, especially in the image processing research field where data are more abundant.

1.2 Motivation

This section provides more details on the motivation of the thesis with regards to the context described in Section 1.1. Section 1.2.1 defines the problems that the thesis attempts to address. Sections 1.2.2 and 1.2.3 provide two practical examples - respectively about sensor-based emotion recognition and sensor-based pain recognition - to illustrate the difficulties raised in the previous Section. Section 1.3 lists the main contributions of this thesis and explains how they address the problems. Finally, Section 1.4 provides more details about the structure of the thesis.

1.2.1 Context

Despite promising results obtained in many application fields of ubiquitous computing, deep learning methods remain difficult to apply to time-series data properly. Two reasons mainly explain this phenomenon.

The first reason is that in order to be trained properly, DNNs tend to require an even larger amount of data than other machine learning models. DNNs are models which typically contain a high number of internal trainable parameters - in that case, neural weights and biases. For instance, the matrix of weights between two