Audio Event Classification using Deep Learning in an End-to-End Approach

Master thesis

Jose Luis Diez Antich

Aalborg University Copenhagen
A. C. Meyers Vænge 15
2450 Copenhagen SV
Denmark

AALBORG UNIVERSITY

STUDENT REPORT

Title:
Audio Event Classification using Deep Learning in an End-to-End Approach

Participant(s):
Jose Luis Diez Antich

Supervisor(s):
Hendrik Purwins

Page Numbers: 38

Date of Completion:
June 16, 2017

Abstract:

The goal of the master thesis is to study the task of Sound Event Classification using Deep Neural Networks in an end-to-end approach. Sound Event Classification it is a multi-label classification problem of sound sources originated from everyday environments. An automatic system for it would many applications, for example, it could help users of hearing devices to understand their surroundings or enhance robot navigation systems. The end-to-end approach consists in systems that learn directly from data, not from features, and it has been recently applied to audio and its results are remarkable. Even though the results do not show an improvement over standard approaches, the contribution of this thesis is an exploration of deep learning architectures which can be useful to understand how networks process audio.

Contents

1 Introduction **1**
1.1 Scope of this work . . . 2

2 Deep Learning **3**
2.1 Overview . . . 3
2.2 Multilayer Perceptron . . . 4
2.2.1 Activation functions . . . 5
2.3 Convolutional Neural Networks . . . 6
2.3.1 Convolutional layer . . . 6
2.3.2 Pooling layer . . . 7
2.4 Regularization . . . 8
2.4.1 Dropout . . . 8
2.4.2 Batch Normalization . . . 8
2.4.3 Data augmentation . . . 8
2.4.4 Early Stopping . . . 9

3 State of the art: Everyday listening **10**
3.1 Overview . . . 10
3.2 Common approaches . . . 11
3.3 Raw audio . . . 12
3.3.1 SoundNet . . . 13
3.4 Datasets . . . 13

4 End-to-End learning for audio events **15**
4.1 UrbanSound8k data set . . . 15
4.2 System Design . . . 17
4.2.1 Dieleman2014 . . . 18

5 Experimental Results **21**
5.1 Final system description . . . 21
5.2 Analysis of the Results . . . 22
5.3 Visualization of the network . . . 23

6 Conclusion **25**
6.1 Future works 25

Bibliography **27**

A UrbanSound8K taxonomy **33**

B Keras summary of RawCNN **35**

C Filters of the strided convolutional layer **37**

Chapter 1

Introduction

Speech and music are the most popular auditory information in acoustic research. Speech has been extensively studied in, for example, the fields of automatic speech recognition [45; 49], and speech generation [18]. In the same way, research in music is also vast. The topics in music research include, among others, music transcription [30], genre classification [58] and beat tracking [23]. Nevertheless, speech and music are just two of the many types of sound sources that can be heard.

The other types of sounds, such as sounds originated from traffic, machinery, walking, or impacts, can be included in the category of *environmental* or *everyday* sounds [56]. This category of sounds carries important information which humans use to be aware and also to interact with their surroundings.

As with speech and music, there has also been research addressing the problem of identifying automatically environmental sounds scenes or the sources, or events, that compose it. This research field can be considered under the umbrella of Computational Auditory Scene Analysis [9]. This field has got significant interest recently, but it is still an open problem and, for this reason, the purpose of this thesis is to tackle it with recent approaches.

As reviewed in Chapter 3, the recent approaches are based on Neural Networks, or Deep Learning. Neural Networks (NNs) and, more especially, Convolutional Neural Networks (CNNs) have been applied successfully to a different range of problems in image processing, such as object recognition [32], object detection [47] and image generation [60]. Deep Learning has also been applied to fields such as natural language processing [15]. And it has also been applied to the audio domain, in, for example, speech generation [18], music tagging [17] or source separation [26].

Although Deep Learning has already been used in the field of everyday listening, the approach taken in this thesis explores a less common approach in which the audio signal is directly input to the Deep Learning system, instead of being first transformed into a feature representation. This approach is usually referred as end-to-end learning.

The motivation for this thesis arises from the increasing exposure to different environments of devices with the ability to listen.

Applications

An automatic system for acoustic scene or events classification could have many potential usages. It could be applied to robot navigation systems [59; 13], to surveillance systems, or noise monitoring systems [53]. Wearable devices could adjust themselves according to the context, hearing aids, for example, could change the equalization settings automatically. In the same way, it could be used to detect and classify speech [7], bird-song [8], musical sounds [21], pornographic sounds [28] or dangerous events [39]. This could be applied to improve the accessibility of audio archives [46] and information retrieval. And it could be used by smart home systems that log the events and notify the users when an event happens. Finally, a system that is aware of its context and understands it could also be used to predict patterns of new events [59].

1.1 Scope of this work

Overview

The thesis is an exploration of neural networks using and end-to-end learning approach in audio. It first evaluates several approaches to audio event classification, and presents a prototype system based on this evaluation. The work gives a general overview of everyday listening and, also, Deep Learning. Special focus is put on exploring how the network architecture represents the audio signal.

The thesis attempts to answer how deep learning architectures are capable of processing audio signals directly.

Structure

The thesis is organized as follows: Chapter 2 deals with the topic of Deep Learning, it introduces the necessary concepts later used in, first, Chapter 3 which gives a general overview of the everyday listening field, explaining the common approaches and the data sets that stimulate its research. And, second, Chapter 4 which reports the exploration of different deep learning architectures and gives a detailed description of the final architecture. Chapter 5 presents the results of the final architecture in classifying audio events. It compares is to different state of the art approaches, as well. Finally, Chapter 6 concludes the thesis with suggestions for future work and a summary of the contributions made by this thesis.

Chapter 2

Deep Learning

This chapter gives an brief overview on the field of Deep Learning. The focus will be on explaining the concepts that will be later used in the development of the proposed system. For a comprehensive description of the field, the reader is referred to Goodfellow et. al. [22]. This chapter follows closely this reference as well as the Stanford CS class CS231n[1].

It is worth mentioning the meaning of the term *Deep Learning*. The purpose of the earliest artificial intelligence algorithms was to model how learning happened in the brain, thus these algorithms were called artificial neural networks (ANNs). Deep Learning is the term applied to the algorithms not necessarily inspired by the biological neural networks [22] and contain many more processing layers than a traditional neural network.

2.1 Overview

Most definitions of Deep Learning highlight the use of models with multiple layers of nonlinear processing units that process and transform the input data. These sequential transformations create a feature representation hierarchy, which can be supervised or unsupervised [16].

These models can be seen as an infinitely flexible function, which can, for example, translate a language to another or recognize cats in pictures. To allow this function to undertake its task, fitting its parameters is required, which is achieved by the technique called *backpropagation*.

Finally, and the reason why deep learning is so popular today is the recent availability of, first, devices that allow to fit these parameters quickly, Graphical Processing Units (GPUs), and, second, large data bases that allow to scale the algorithms.

The following sections give an general overview on the building blocks of Deep Learning.

[1] Convolutional Neural Networks for Visual Recognition: `https://cs231n.github.io/`

2.2 Multilayer Perceptron

The multilayer perceptron (MLP), also called Feed Forward Network, is the most typical neural network model. Its goal is to approximate some function f . Given, for example, a classifier $y = f\ (\mathbf{x})$ that maps an input x to an output class y, the MLP find the best approximation to that classifier by defining a mapping, $y = f(x;\theta)$ and learning the best parameters, θ, for it.

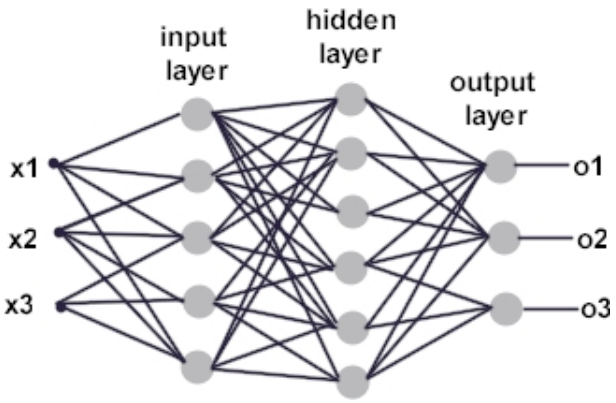
The MLP networks are composed of many functions that are, for instance, chained together. A network with three functions or layers would form $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$.

Each of these layers are composed of units that perform an affine transformation of a linear sum of inputs. Each layer is represented as $y = f(\mathbf{W}\mathbf{x}^T + \mathbf{b})$. Where f is the *activation function* (covered below), $\mathbf{W}$ is the set of parameter, or weights, in the layer, $\mathbf{x}$ is the input vector, which can also be the output of the previous layer, and $\mathbf{b}$ is the bias vector.

The layers of a MLP consists of several fully connected layers because each unit in a layer is connected to all the units in the previous layer. In a fully connnected layer, the parameters of each unit are independent from the rest of units in the layer, that means each unit possess a unique set of weights.

Figure 2.1 shows a diagram of a multilayer perceptron network of three layers: the input layer with five units, a hidden layer with six units and an output layer. It has three inputs and outputs.

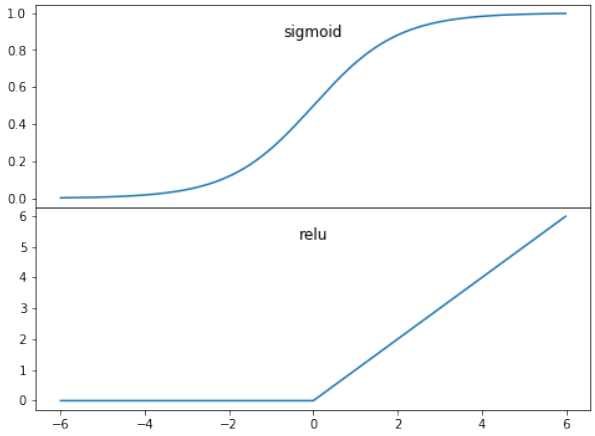
Figure 2.1: Multilayer perceptron



In a supervised classifier system, each input vector is associated to a label, or ground truth, defining its class. The output of the network gives a class score, or prediction, for each input. To measure the performance of the classifier, the *loss function* is defined. The loss will be high if the predicted class does not correspond to the true class, it will be low otherwise.

In order to train the network, an optimization procedure is required. This procedure will find the values for the set of weights, $\mathbf{W}$ that minimize the loss function.

Figure 2.2: Activation functions



A popular strategy is to initialize the weights to random values and refine them iteratively to get lower loss. This refinement is achieved by moving on the direction defined by the gradient of the loss function. And it is important to set a *learning rate* defining the amount in which the algorithm is moving in every iteration.

The gradient of the weights is computed efficiently using *backpropagation*, which computes gradients through recursive application of chain rule.

2.2.1 Activation functions

Non-linear activation functions describe the input-output relations in a non-linear way. Thus gives the model power to be more flexible in describing arbitrary relations.

Here, two of the most popular activation functions are described.

Sigmoid function

The sigmoid function takes the form:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

It takes a real number, x, as input and returns a value between 0 and 1, which is a particular case of the logistic function and can be interpreted as a probability value. Its shape is shown in figure 2.2

An output layer composed of sigmoid units is usually used for problems in which an input vector can belong simultaneously to several classes.

Rectified Linear Unit (ReLU)

The Rectified Linear Unit is the most popular activation function at the moment and it computes the function $f(x) = max(0, x)$, as shown in figure 2.2.

Softmax activation

The softmax function is the generalization of the logistic function to multiple classes. Therefore, its output can represent a categorical distribution over multiple classes. The softmax is given by the formula

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

for $j = 1, \ldots, K$. Where K is the number of classes.

2.3 Convolutional Neural Networks

Convolutional neural networks (CNNs) [33] are networks that use a mathematical operation called convolution instead of a matrix multiplication in at least one of their layers. This kind of networks was inspired by the visual cortex and have been very successful in practical applications, usually in image recognition.

The CNN architectures are usually built with three main types of layers: Convolutional layer, Pooling layer and Fully Connected layer. In this section the first two are explained.

2.3.1 Convolutional layer

The convolution is an operation between two functions or signals. It is denoted with an asterisk and its one-dimensional version is written as:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a)$$

The first signal, x, is called the **input** and the second signal, w, is the **filter**. In this one-dimensional case, the term t is the time index and a is a time shift value. In the convolutional layers, the output of the convolution is referred as the **feature map**.

For images, it is more common to use a two-dimensional convolution that accepts as input a three dimensional matrix (width, height and colour channels) and outputs a three dimensional matrix as well.

The parameters of the convolutional layer are composed of a set of learnable filters. Each of those filters will be convolved across the width and height of the input. This will produce a two dimensional feature map. The network will learn filters that are activated when a certain feature is detected. In the case of images, this feature can be an edge and, in the case of sound, can be a frequency component.

Each filter in the layer will produce a separate feature map. Stacking these maps will produce the output tensor with dimensions (output width, output height, number of filters).

As said before, each unit in a fully connected layer (FC) is connected to all the previous units. One difference between FCs and CNNs is that each filter of a convolutional layer is connected to a limited number of input values. This property of the convolutional layers is also referred as **sparse interaction** and the hyperparameter controlling it is the *filter size*. Secondly, each set of parameters in a FC unit is independent from the rest. In CNNs, each filter applies the same weights at each local region of the input signal. This property is called **parameter sharing** and it causes each filter to *find* the same feature across the input. Each output feature map will describe a different detected feature. The third property of CNNs is **equivariance**. In short, it means that the changes in the input result in the same changes in the output. (Goodfellow et.al (2016) [22]: Section 9.1, p. 331)

These properties, in particular parameter sharing, allow the networks to be suitable to analyze signals while reducing the number of parameters.

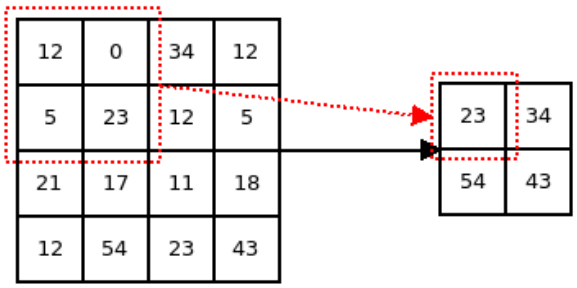
2.3.2 Pooling layer

After the convolutional stage, a typical CNN will use an activation function to produce a non-linear representation. To modify this output further, a pooling layer is used.

The goal of the pooling layer is to reduce the spatial size of the representation to reduce the number of parameters and, thus, computation in the network. To do that, the output of the network at a certain position is replaced with a statistical opeartion on the neighborhood values.

A Pooling layer can apply several operations to the neighborhood of a location, such as the *maximum*, the *average*, the *weighted average*, and the L^2 *norm*. MAX Pooling is currently the most popular operation in practical applications and it returns the maximum value within a neighborhood of values. Figure 2.3 shows an example of max pooling, one can see this action reduces the dimension of the input.

Figure 2.3: Max Pooling



Apart from reducing the computational cost of training the network, pooling layers help to make the network invariant to translations. That is, if the input is translated by a small amount, the pooling will result in the same values. This is an interesting property when it is more important to detect if the feature is present

rather than detecting its location.

2.4 Regularization

A successful machine learning system needs to achieve a low *training error* while minimizing the difference between training and *test error*. Training error is the error measure defined over the data that the system used for training, in the same way, test error is defined over unseen data. The performance of the system is measured using the test error.

The situation when the training error is low while the test error is high can occur when the model is too complex for its task. Overfitting is the term used to define this situation.

In order to avoid overfitting, a deep learning system can be modified to include some *regularization* techniques. In this section, some of them are briefly described.

2.4.1 Dropout

Dropout is one simple technique to avoid overfitting. It consists in *dropping* units from the network during training [54]. Dropping a unit means to temporarily remove it with all its connections, both input and output, with a given probability.

This technique addresses two regularization methods: the first one is *model combination*, also referred to as bagging, which involves training a model with all possible setting of its parameters; the second method is related to the fact that deep learning methods require large amounts of data.

Dropout requires a hyperparameter which sets a value for the probability that a unit will be disconnected.

2.4.2 Batch Normalization

During training a network, the input distribution of a layer changes as the parameters of the preceding layer change. In deep networks, this is a problem known as *covariance shift* because the layer needs to adapt to this change continuously. Covariance shift causes the training to be slower and requires a careful initialization of the parameters.

Batch Normalization [27] is an optimization technique to overcome covariance shift by normalizing each input batch with its mean and variance.

The usage of a batch normalization layer is placed after the convolution or the fully connected layer, but before the activation layer. It allows to use higher learning rates and be less careful about initialization while requiring fewer training steps.

2.4.3 Data augmentation

Another way to make a machine learning system avoid overfitting is to use more data to train it. Since the amount of data is limited, it is also possible to create new fake

data by modifying the training data. This procedure is known as *data augmentation* and it is domain-specific.

In object recognition, techniques of data augmentation include, among others, rotating the images, translating them or perturbating the color.

In the audio domain, data augmentation techniques include pitch shifting, time stretching or dynamic range compression.

2.4.4 Early Stopping

Early Stopping is a fruitful but simple strategy consisting in stopping the training of the model when a monitored measure, for example the validation error, has not improved for some amount of time.

The parameters of Early Stopping are: (1) the quantity to be monitored, (2) the number of epochs with no improvement after which training will be stopped, called *patience* and (3) the minimum change in the monitored quantity to qualified as an improvement, called *minimum delta*.

Chapter 3

State of the art: Everyday listening

The main topics of machine listening research have been speech and music. Even though these are just two of the many sound sources that can be heard in most environments, the analysis of environmental sounds has been limited until recently. The lack of, first, public annotated data and, second, a common vocabulary for it have been causes for the scarce research in this field [53].

In this chapter, the field of everyday listening is given an overview which cover its terminology, some data sets that stimulate its research and the common methods that tackle it.

3.1 Overview

As said in the introduction, the Everyday listening field can be considered under the umbrella of Computational Auditory Scene Analysis (CASA). The two challenges of CASA are, first, to classify the acoustic environment, or scene, and, second, to recognize the distinct sound events in the scene.

Acoustic scene classification (ASC) is the first challenge of CASA and its goal is to recognize the environment in which an audio signal was recorded [41; 55]. This environment can be defined based on a physical or a social context (park, office, meeting, ...) [35]. It is a single-label classification problem similar to music genre recognition or speaker recognition.

The analysis of the events can be separated in two problems: the detection or the recognition. Sound Event Detection (SED) aims to identify the start and the end time stamps of each event in an audio signal. And it can be further divided into monophonic or polyphonic detection. The output in the first approach will be the most dominant sound at each time instance. In polyphonic detection, it is required to detect all the overlapping events. Sound Event Recognition (SER) aims to classify each event into different classes and the location of the event is considered a different problem.

This thesis is focused in exploring the second approach to sound event analysis: Sound Event Recognition.

3.2 Common approaches

Historically, many works on Sound Event Detection (SED) or Environmental Sound Classification (ESC) have relied on speech recognition techniques. Thus, the most common features used were the Mel Frequency Ceptrum Coefficients (MFCCs) [1], VQT Spectrograms [61; 31] the Mel-spectrogram [53], or the mel-band energy features [11]. These were used in combination with classifiers such as Gaussian Mixture Models [14], Hidden Markov Models [1], Non Negative Matrix Factorization [20], Support Vector Machines [57; 64; 43] or Random Forest Classifier [53; 43]. Recent approaches use feature learning with, for example, the scattering transoform [50] or the log-mel-spectrogram [51]. Finally, the state of the art is based on of Deep Neural Networks (DNNs). These state of the art approaches include Feed Forward Neural Networks [11; 10], Deep Convolutional Neural Networks (DCNN) [42; 52], or Recurrent Neural Networks (RNN) [24; 3; 65].

As this master thesis is focused in Deep Convolutional Neural Networks applied to the sound event classification task, it is interesting to go more into detail in similar approaches.

In [42], Piczak used a DCNN to classify the sound events in the UrbanSound8K dataset. As the input features, the the Log-scaled Mel-spectrograms were used. The DCNN architecture consisted in 2 convolutional layers followed by 3 fully connected (dense) layers. A MaxPooling layer followed the first convolutional layer and a Dropout layer followed the first dense layer. With this architecture, the reported categorical accuracy was 73%.

Using the same dataset and input features, the work of Salamon and Bello [52] addressed the problem with a different architecture and a data augmentation procedure. The DCNN architecture consisted in 3 convolutional layers, the first two interleaved with max pooling operations, followed by 2 dense layers. As activation functions they used Rectified Linear Units (ReLUs) for all the layers except for the output layer which used Softmax. The data augmentation stage included the following deformations: time stretching, pitch shifting, dynamic range compression and background noise. This method reported 79% categorical accuracy.

In [10], the purpose is Sound Event Detection (SED), which involves to determine the onset and offset time of each event as well as to identify its class. To address it, a combination of convolutional and recurrent networks (CRNN) is used. The architecture is chosen due the capability of the convolutional networks to learn local translation invariant filters and the capability of the recurrent networks to model temporal dependencies. In particular, the architecture consists of four parts: a convolutional block which takes as input the log mel band energies, a recurrent block on top of it, a single dense layer which estimates event activity probabilities for each frame which are binarized into predictions in the final part. Their proposed

method is compared across several data sets to two previous approaches, a Feedforward Neural Network model and a Gaussian Mixture Model. The CRNN shows a clear improvement over them, however, it is dependent on large amounts of data.

3.3 Raw audio

As said before, these approaches have relied on using hand-crafted features designed to be practical for the classification task. Using these features requires significant prior knowledge in the task [17], and it is possible these features are not the best representation of the data to be used by the classification method [48].

In computer vision, feature learning methods which do not require any preprocessing of the images represent the state of the art. In the audio domain, this is a recently opened trend and it uses the raw waveform as the direct input to a deep learning system [17; 48; 6; 40; 18; 5; 62; 4].

The work of Dieleman and Schrauwen in [17] is the first attempt of an *end-to-end learning* approach to automatic music tagging. End-to-end learning is referred to processing architectures where the stack that connects the input to the desired output is learned from data. Thus, the construction of features and the classification task are not two separate problems, but one.

The deep learning architecture presented in [17] expects as input 3 seconds of audio and it consists of a one-dimensional strided convolution layer, two filter stages and a classification stage. The filter stages are composed of a one-dimensional convolution followed by a max pooling layer and a RELU activation. The classification stage contains two dense layers.

Even though, the task they faced was performed better by an approach based on the spectrogram, they prove that networks are capable of discovering frequency decompositions and, when incorporating a feature pooling layer, phase translation-invariant features.

After the work of Dieleman and Schrauwen, a similar approach has been used in speech recognition. Either with a similar convolutional architecture [40] or with adding Recurrent layers, such as long short term memory (LSTM), after a convolution stage [48]. The analysis of the proposed neural networks agree with [17], since, first, [40] reports that the first convolution layer learns matching filters which are combined linearly after the max pooling operation. And, second, in [48] the network learns auditory-like filterbanks of bandpass filters.

In the same way, the end-to-end approach has been applied to audio generation [18; 4; 5]. Wavenet[1], released by Google's DeepMind in September 2016, is the most significant example of it. Wavenet addresses the problem of Text-to-Speech synthesis by modelling waveforms sample by sample. Its architecture is based on dilated causal convolutions. These special kind of convolutions allow the network to increase its receptive field, i.e. the number of input samples, and to model each sample given the previous samples only.

[1] https://deepmind.com/blog/wavenet-generative-model-raw-audio/

Finally, this strategy has also been applied to the problem of this thesis, environmental sounds analysis. According to the author's knowledge, only two publications address this problem with this strategy, these are [6] and [62]. However, it is not the focus of the second publication, which is the reason why only the first publication, SoundNet, is reviewed here.

3.3.1 SoundNet

Soundnet [6] is an inspiration for this thesis because, as said before, is a deep convolutional network that inputs raw audio waveforms as directly and it is applied to the tasks of audio scene and event recognition.

The available data sets for environmental sound are more much smaller than the data sets available in computer vision. These small data sets are not suitable to train deep neural networks. Aytar et. al. overcome this obstacle by using a data set of 2 million videos and training the audio network by transfer learning with a vision network as its supervision.

In particular, state of the art vision neural networks, ImageNet CNN [32] and Places CNN [63], are used to *teach* SoundNet to recognize scenes and objects. This training is based on minimizing the KL-divergence between the predictions of the teacher vision network, given the video, and the output distribution of SoundNet, given sound.

Two architectures are presented in SoundNet's publication. The first architecture has 8 layers and the second 5 layers and both are composed of 1-dimensional convolution layers, max pooling layers and RELU activation functions.

Once SoundNet is trained, its performance is compared to several state-of-the-art methods in different audio scene and events classification data sets. SoundNet reports an improvement over the other methods.

3.4 Datasets

As said before, the lack of public annotated data has been an obstacle for the field of everyday listening. To overcome this obstacle and stimulate this research, the datasets UrbanSound, UrbanSound8K [53] and Audio Set [19] have been released. In the same way, the IEEE DCASE Challenge [36] addresses the same objective. A comprehensive list of environmental sound data bases is maintained by Toni Heittola[2].

The focus of the UrbanSound and UrbanSound8K datasets[3], released in 2014, is the urban environmental sounds and it is build around the most frequent sound sources in noise complaints filed in New York City. The 10 classes in the dataset are: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot,

[2] https://www.cs.tut.fi/~heittolt/datasets
[3] https://serv.cusp.nyu.edu/projects/urbansounddataset/

jackhammer, siren and street music. The source of the recordings is Freesound[4].

On the one hand, UrbanSound contains 27 hours of audio with 18.5 hours of manually labelled sound events. It can be used for sound event detection as it contains the original 1302 recordings along with the annotations for the start and end times of each event. On the other hand, UrbanSound8k is a subset of the previous designed for sound event classification as it contains 8732 audio snippets of 4 seconds across each event class aranged in 10 folds.

In 2017, Google released Audio Set [19], an ontology[5] of 632 audio event classes and a collection of 2.1 million human-labelled 10-second sound clips drawn from Youtube videos. In contrast to previous data sets, Audio Set is not limited to any context, but considers all sound events. The six top level layers in the Audio Set hierarchy graph are: Human sounds, Sounds of things, Animal Sounds, Source-ambiguous sounds, Natural sounds, Music and Channel, Environment and Background.

In the same way as the mentioned datasets, The Detection and classification of acoustic scenes and events (DCASE) Challenge holds the same objective: to stimulate the development of computational scene and event analysis. The challenge has been held in 2013, 2016 and will be held in 2017 as well. It provides different data sets depending on the task.

The development data set for the Sound event detection in real life audio task, TUT Sound Events 2017 [36], consists of recording of street acoustic scenes, which were selected as representing human activities and hazard situations. There are 24 audio recordings, 1.54 hours in total, and 6 classes: brakes squeaking, car children, large vehicle, people speaking, people walking.

[4] freesound.org

[5] It can be explored interactively in https://research.google.com/audioset/dataset/index.html

Chapter 4

End-to-End learning for audio events

A system which trains its function directly from data and, thus, connects its input to the desired output is known as an *end-to-end learning* system [38; 17]. The main advantage of such systems is avoiding the need of using hand-crafted heuristics, which rely in prior knowledge about the specific problem and require significant engineering effort.

In addition, learning the features from the data can outperform hand-crafted engineered features because the features are created for the particular task.

In the previously cited references, this approach was used for music audio tagging and vehicle navigation. Both of these references used deep learning as it is well suited for the end-to-end approach because the same objective function is used by several layers of processing.

This chapter covers the exploration of deep learning architectures applied to the task of sound event recognition in an end-to-end approach. In particular for audio, the input is the waveform and not a hand-crafted feature such as the Mel Frequency Cepstral Components (MFCCs).

For this purpose, first, the employed data set is described in detail, second, the proposed system architecture is detailed as well as, third, the decision process that lead to it.

4.1 UrbanSound8k data set

The UrbanSound8k data set has been briefly described in the previous chapter. As it is employed for the development of the end-to-end architecture, the different classes it contains are described here. This can help identify what the network should be able to distinguish.

In [53], Salamon et. al. describe a taxonomy[1] for urban sounds and the data

[1] See Appendix A to see an image of the taxonomy

set they release includes 10 low-level classes from that taxonomy, which are: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren and street music. There are up to 1000 examples per class and each of them has a maximum duration of 4 seconds. The data set is given divided into 10 folds to allow cross validation.

The research Salamon et. al. addresses the sound pollution in cities, thus noise complaints were used to identify the most common sound sources. Given this, 7 of the classes in the data set are of mechanical sources. Three of them belong to the *road* category within the *motorized transport* category, two belong to *construction*, the other two are gun shot and air conditioner.

The goal of the European project CLOSED [56] is to provide a useful measurement tool for sound designers. One of the steps to deliver that is to categorize everyday sounds. They review and unify several everyday listening categories (See Figure 2.8 of part 2 the deliverable number 4.1 for the cited reference). These categories group sound events given their acoustic similarity.

The three main categories in this taxonomy are related to the pitch of the sound, its rhythm and its sequence. If the sound is pitched, this can be continuous or changing. The timbre is analyzed with both its spectral and time properties. Finally, the sound sequence can be composed of one or more elements.

Describing the diferent classes of the UrbanSound8K data set according to the categories from [56] can be of help when designing the deep learning architecture employed to distinguish them [44]. Thus, the following part of the section does this.

The `air conditioner` class belongs to the mechanical upper level category in the urban sound taxonomy and in the ventilation sub category. In general, it is an unpitched noisy class with most of the energy in the lower part of the frequency spectrum. Its timbre is continuous over time and its sequence is also regular.

The `children playing` class, of the high level category human and sub category voice, is composed of samples very different between them in sound sequence and also timbre. The sound sequence has several components with a very irregular pattern. The timbre is also very different depending in if the children are talking or screaming. These segments have also an important component of background noise. However, the voice of the children is significantly higher in pitch that the background.

The segments composing the `dog bark` class, of the nature high level category and animals subcategory, can be recognized because of the short impulsive sounds. The timbre is usually stable over time, but the pitch, although typically high, can vary. In the same way, the sound sequence has one component which can be repeated.

The `drilling` class, of the high level category mechanical and sub category construction, can be described as with a short sound sequence composed of several noises with a regular pattern. Its pitch is stable in high frequencies and its timbre is continuous in time and rich in spectral properties.

The `engine idling` class belongs to the road low-level category of the motorized transport category in the mechanical high-level category. It is characterized for a sound sequence composed of several noises with a regular pattern. There is no pitch

and the timbre is continuous with high energy in the low frequencies.

The `gun shot` class, of the mechanical high level category and social/signals sub category, is characterized of short impulsive unpitched noises. Its sound sequence consists of one element.

The `jackhammer` belongs to the construction sub category within the mechanical high-level category. Its sound sequence consists of several noises presenting a pattern which changes its speed. It is a high pitched class and its timbre is noisy and continuous over time.

The `car horn` class is shared across all the leaves in the road low-level category of the motorized transport category in the mechanical high-level category. It is characterized for a sound sequence composed of unique component which can be repeated. It is a high-pitched class and the timbre is continuous over time with an important high frequency component.

The position in the urban taxonomy of the `siren` class is similar as the `car horn`. Its sound sequence is composed of one element; its pitch is high but changing; its timbre is continuous.

Although the categories of [56] do not describe music, if the `street music` class is described in its terms, is generally rich in the

The `street music` cannot be thoroughly described with the categories in [56]. The sequences in this class are of a very varied nature as they contain different instruments and different rhythms. In general, is the only class which presents harmonic pitches.

One of the challenges of the system will be to distinguish the classes `air conditioner`, `drilling`, `jackhammer` and `engine idling` as they are the most similar. Previous work in this data set [53; 42] has shown these classes to be problematic. These approaches, however, relied in hand-crafted features. Therefore, it will be interesting to see if an end-to-end learning approach can overcome this difficulties.

4.2 System Design

In the previous chapter, the SoundNet architecture is described in detail because serves as an inspiration for this thesis. It is also an inspiration the architecture presented in the work of Dieleman and Schrauwen [17] given that is the first attempt of an end-to-end learning approach to music.

This section describes in detail the steps that lead to the proposed end-to-end deep learning architecture, which is described in Chapter 5. These steps start by modifying the architectures mentioned above. The first experiments are based on the Dieleman's architecture, which is next described along with the modifications.

The system is implemented using the Python programming language. Specifically, the package Librosa [34] was used for the audio files handling, and the package Keras [12] was used for the deep learning architecture building and training. As the Keras backend, the package Tensorflow [2] was used to enable GPU acceleration with

multiple devices. In particular, three NVIDIA TITAN X GPU devices were used for the training.

The sampling rate of the audio is 44100Hz and, before feeding it to the networks, it was normalized between -1 and 1.

The networks were trained using Adam optimizer [29] with fixed learning rate of 0.001; Early Stopping monitoring the validation categorical accuracy with patience of 10 epochs, and minimum delta of 0.001; batch size of 256.

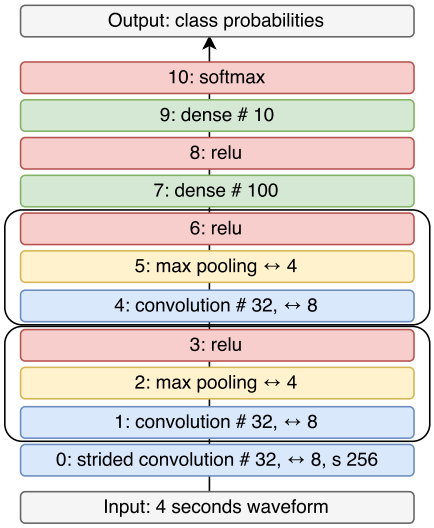
4.2.1 Dieleman2014

The architecture from [17], which will be called `dieleman2014` in this thesis, it is simply built up of two convolutional blocks and a MLP block.

The input to the network is 3 seconds of audio and it is first feed in to a *strided convolution* which can be seen as a downsample operation. Each of the two subsequent convolutional blocks consist of a convolution of 32 filters and filter size of 8 followed by a max pooling operation and a ReLU activation function. The MLP block consists of a fully connected layer of 100 units and ReLU activation function an output layer of 50 units and sigmoid activation function.

In order to work with the UrbanSound8k data set, the network is modified to allow an input of 4 seconds and the final sigmoid activation is replaced by a softmax function. See a representation in figure 4.1.

Figure 4.1: Architecture of dieleman2014. The filter sizes and pooling sizes are indicated with ↔, the number of filters is indicated with # and s indicates stride.



Hereafter, further experimentation with the network is described and the performance results are reported. The performance measured by validation accuracy in classifying correctly the different classes in the data set. The model is trained with 9 folds and the last fold is used as validation. The reported value is the mean accuracy over all validation folds. To verify for significant results, first, the Shapiro-Wilk normality test is performed given it is suitable for small data sets. Second, an ANOVA test is performed to verify if the improvements are significant or not. In general, the data passed the Shapiro-Wilk test, but only one set of the results passed the ANOVA.

The experiments on the network modify its building blocks: (1) strided convolution; (2) number of filters in the convolutional blocks; (3) number of convolutional blocks. And also add regularization to the network: (4) Dropout; (5) Batch normalization.

Training each modified network took at least 2 hours; time and resources limited the number of models that could be trained.

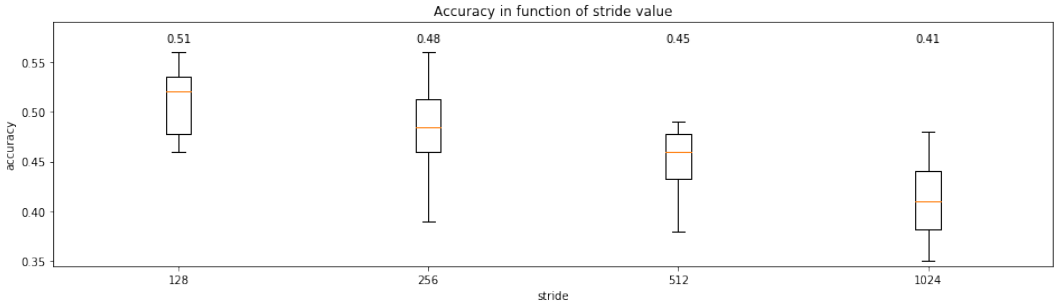
4.2.1.1 - Strided convolution

The first modification affects the strided convolution. In the original paper, the value of its stride is set to match the spectrogram representation which serves as baseline. It ranges from 256 to 1024, being 256 the value which achieves best performance in their task.

The strided convolution summarizes the input signal, one can think this can remove important characteristics of the signal. The opposite, however, can overload the network with too much information. Therefore a balance between these options must be found.

For the task in this thesis, several stride values have been evaluated and the best performance has been achieved with a value of 128. However, the ANOVA test does not report a significant improvement from a stride of 256 (f-value=2.43, p-value=0.14). Figure 4.2 shows a box plot of the accuracy values reported by the stride experiment.

Figure 4.2: Box plot of the accuracy values reported in the stride experiments. The value on top of each box plot is the mean validation accuracy.



Taking the model with 256 stride as the original `dieleman2014`, one can see the starting accuracy value for this experiment is 48%.

4.2.1.2 - Filter size of strided convolution

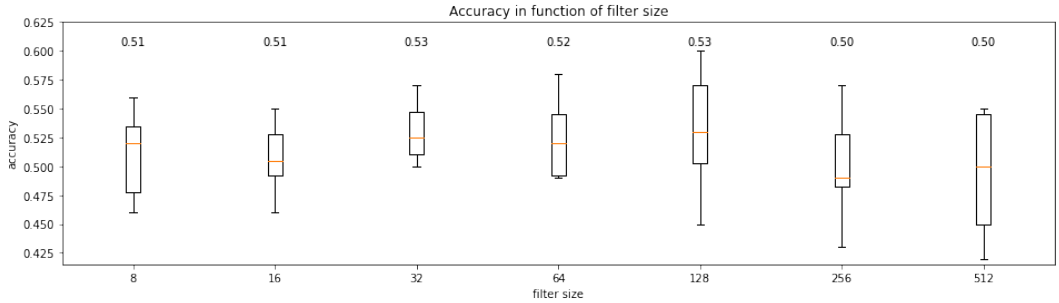
In the original implementation of `dieleman2014`, the three convolutional layers have the same number of filters and filter size. It is common to see architectures in which each subsequent convolutional layer adds more filters and those are getting smaller, see SoundNet [6] for an example.

For this reason, experiments in changing the filter size of the first strided convolution have been performed. The original value for the filter size is 8 to which the values 16, 32, 64, 128, 256 and 512 have been added. The accuracy, however, does not change significantly.

Figure 4.3 shows the results of the experiment. As one can see, the best values are achieved with two filter sizes: 32 and 128. The accuracy value reached is 53%. However, the significance test does not report difference (f-value=0.92, p-value=0.41).

The following experiments will be performed using filter size of 128 because a bigger feature map will help visualize better what the network is learning.

Figure 4.3: Box plot of the accuracy values reported in the filter size experiments. The value on top of each box plot is the mean validation accuracy.



4.2.1.3 - Number of convolutional blocks

In the same way as the filter size, SoundNet also includes several convolutional blocks. Likewise, a third block has been added to the network. In this setup, the number of filters is $64, 128, 256$ for each block and the corresponding filter sizes are $32, 16, 8$.

The results show an improvement reaching accuracy values of 55%. But, again, the ANOVA test does not report a significant difference between the results (f-value=0.66, p-value=0.42).

4.2.1.4 - Regularization: Dropout

In the previous experiments, the training accuracy is not reported, but it is notably higher than the validation accuracy, i.e. the models overfit the data. For the experiment of different strides, the training accuracy is between 10 to 16 points higher than the validation accuracy. For the experiment of the filter sizes, it is between 16 and 29 points higher. For this reason, it is crucial to add some kind of regularization.

A Dropout layer has been added before and after the hidden fully connected layer with a probability of dropping a unit of 50%. The results improve significantly and the accuracy value is 62%. The ANOVA test confirms this (f-value=6.64, p-value=0.02).

Other values of dropout probability and different combinations in the two dropout layers have been also analyzed, but the best results are when both dropout are at 50% rate.

4.2.1.5 - Regularization: Batch Normalization

A batch normalization layer has been added to each convolutional block just before the activation layer. The accuracy achieved with this version of the model are actually significantly lower than in other architectures. The value reached with it is 45%. This regularization procedure is found across the state-of-the-art models and it is proven to be an effective tool to avoid overfitting. A possible cause for the poor performance in this task is the limited amount of training data.

Chapter 5

Experimental Results

The previous chapter described the process of building the end-to-end architecture. In this chapter, the results on the data set attained with the final model are described. Moreover the model filters are inspected in order to understand what the network has learned.

5.1 Final system description

The only hyperparameter that affected significantly the performance of the network in a positive way has been the dropout. Nevertheless, the final network architecture includes also the modifications that improved slightly the mean accuracy. These modifications are the stride of value 128, an additional convolutional block and the dropout layers.

The final architecture can be visualized in figure 5.1. As one can see, it is composed of three parts: (1) a summary stage; (2) a convolutional stage and; (3) a dense stage:

1. The **summary stage** consists of a strided convolution of 32 filters of size 128 and stride 128.

2. The **convolutional stage** is composed of three blocks each containing a convolutional layer, followed by a max pooling layer and a rectified linear unit as the activation function. The convolutional layers have 64, 128 and 256 filters respectively of size 32, 16 and 8. The max pooling is for the three block of size 8.

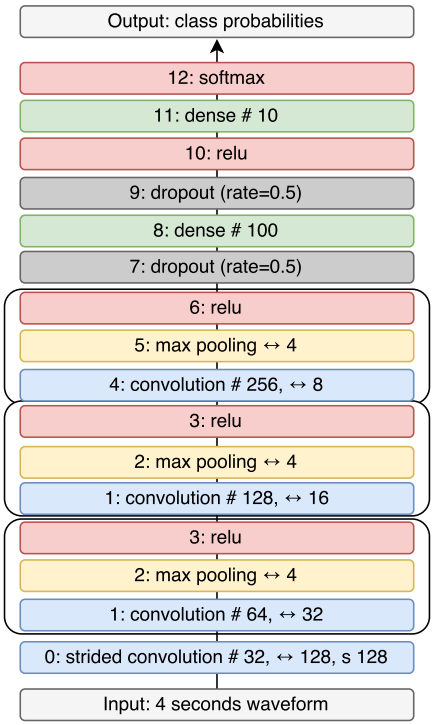


Figure 5.1: Final system architecture. The filter sizes and pooling sizes are indicated with ↔, the number of filters is indicated with # and s indicates stride.

3. The **dense stage** consists of two fully connected layers with 100 and 10 units, respectively. The first of them is preceded and followed by dropout layers with 50% probability of dropping. The activation function after the second dropout is the Rectified Linear Unit. And a softmax function is used to compute the output distribution.

Appendix B contains the representation of the model given by the `summary` function in Keras.

5.2 Analysis of the Results

As reported in the previous chapter, the proposed architecture, which is named RawCNN, achieves 62% of accuracy in classifying the sounds of the UrbanSound8K dataset in their respective classes. This value, however, is notably lower than the previous published attempts at this data sets. Table 5.1 summarizes the results of these attempts.

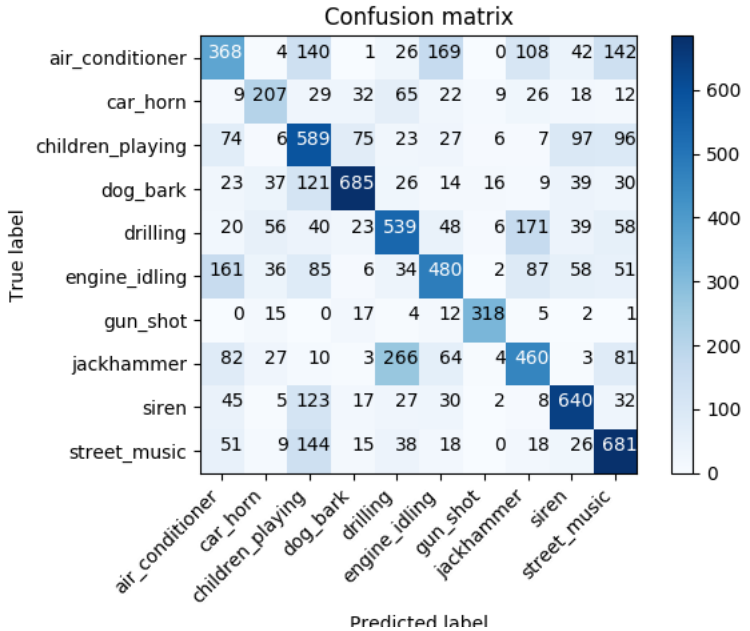
Table 5.1: Comparison between the accuracy reported by several systems on the UrbanSound8K. SVM is support vector machine, SKM is Spherical K-means, Piczak CNN is a Convolutional Neural Network, SB-CNN is different Convolutional Neural Network which uses Data augmentation.

System	Features	Accuracy
SVM (baseline) [53]	MFCC	68.0%
SKM [51]	Log Mel-Spectrogram	73.6%
PiczakCNN [42]	Log Mel-Spectrogram	73.7%
SB-CNN [52]	Log Mel-Spectrogram	**79.0**%
RawCNN	Raw waveform	62.0%

As said before, one of the advantages of using an end-to-end approach is letting the network create the features tailored to the problem. This would solve or diminish the problem of confusing classes, as it is reported in the literature. In the cited references, it is described how three pairs of classes are the most confused given its timbre similarities: air conditioner with idling engines; jackhammers with drills and; children playing with street music. As the confusion matrix in figure 5.2 presents, this same issue is found with the RawCNN model. The mentioned pair of classes are the most confused with RawCNN, and the confusion matrix is very similar to the matrices presented in the literature. In addition, in this case, since the model is not capable of find good patterns, there is more confusion among classes.

The reasons for this poor performance are not completely clear given that in [17] a similar model is applied with good results to a problem of audio tagging. In the same way as the experiments with batch normalization, it is possible that one of these reasons is the limited amount of training data in the data set used to train RawCNN.

Figure 5.2: Confusion matrix for the proposed RawCNN model evaluated on the UrbanSound8K data set.



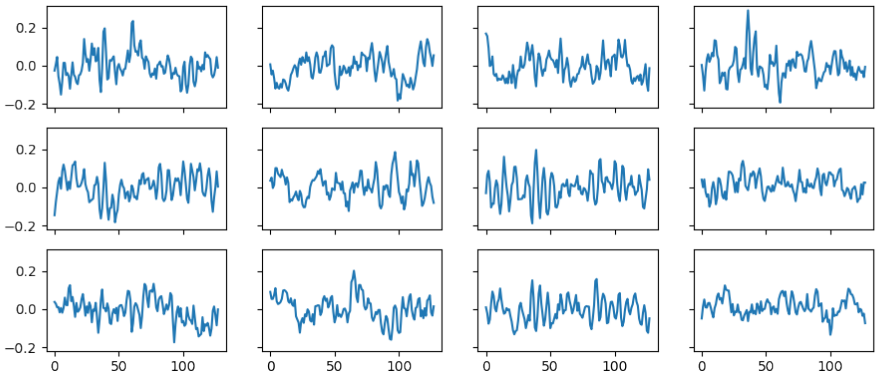
The requirement for a large amount of training data is a well known problem in deep learning, given the high dimensionality of raw waveform data [50], this requirement might be more strict in end-to-end approaches to audio.

5.3 Visualization of the network

The two references which inspired this thesis [6; 17], the networks were able to discover frequency decompositions from the raw waveforms. In contradistinction to these models, RawCNN did not attain good results in classifying the sound events. For this reason, the filters of this network do not present clear shapes as found in the cited references. One of the reason for this important difference is the amount of data used in each case, SoundNet, in particular, used a dataset of 2 million videos.

In fact, as one can see in figure 5.3, the filters are very noisy, which motivates the study of different weight initialization, as opposite to random initialization. In [48], gammatone impulse responses are used to initialize the weights, which leads to better results for their task.

Figure 5.3: Subset of filters of the strided convolutional layer. See Appendix C for the rest of the filters



Chapter 6

Conclusion

The problem of Sound Event Classification has been studied with the aim of developing a deep learning architecture for it in an end-to-end approach. This approach simplifies the problem by using the raw waveform directly, which avoids the need for hand-crafted features and allows the network find the representation that best suits the task.

An overview of the field of Deep Learning has been covered, describing typical architectures such as the multi-layer perceptron and the convolutional neural networks. Some regularization techniques have been detailed, as well.

The field of everyday listening has also been reviewd along with its tasks, common approaches to them and the available data sets encouraging its research. The data set UrbanSound8K has been described in detail as an attempt to find clues to build a dedicated architecture for it.

To find a suitable architecture to classify the sound events of UrbanSound8K, several neural network architectures have been evaluated. The starting point of the architecture was the network presented in [17] because it is the simplest network using raw waveform as input. From then, the hyperparameters evaluated were (1) the stride size, (2) the size of the convolutional filters, (3) the number of convolutional blocks, (4) regularization with dropout, (5) regularization with batch normalization.

After the evaluation of several blocks, an architecture was proposed, RawCNN, which was based on a summary stage, a convolutional stage and a fully connected stage. Unfortunately, RawCNN performed notably worse than the baseline method. In spite of that, the confusion matrix obtained by the proposed model evaluated on the UrbanSound8K data set shows a similar pattern as the previous approaches. Finally, as opposite to networks relying on the raw waveform as input, RawCNN was not able to discover frequency decompositions from the audio.

6.1 Future works

The results of the experiments in this thesis point to several directions and options for further experimentation with the network architecture:

Recurrent networks Several deep learning architectures relying in hand-crafted features have also used *Recurrent* layers [37] after several convolutional blocks [48; 62; 10]. In the same way as these, it would be interesting to also add to the proposed architecture layers such as Long Short Term Memory [25].

Gated activation units Very recent architectures [18; 60] present the so-called Gated activation units which combine two different set of weights into two activation functions to obtain complex interactions with the input. A further modification of the network could replace the ReLU activation with these.

Combine raw waveform with other features Given the failure of the proposed end-to-end approach, using additional features to the waveform should be considered. These features could rely in stereo information [62], which could be easily extracted from the waveform and would not require significant expertise to compute, or in spectral features [48].

Data augmentation In Chapter 2, data augmentation was described as a regularization method consisting in creating new data by modifying the existing data with different procedures. In [52], data augmentation was applied to the UrbanSound8K data set by using the audio deformations Time Stretching, Pitch Shifting, Dynamic Range compression and adding background noise. The reported results show an improvement in accuracy from 73% with the standard data set to 79% with the augmented data set.

In the case of the architecture presented in this thesis, the accuracy values achieved with the standard data set were significantly lower than the accuracy values reported in [52]. In order to compare machine learning algorithms, it is required that the same data set is used by all compared algorithms ([22] Chapter 7, p 241). Thus, before augmenting the data set, the architecture must be first enhanced. However, it is still an interesting further option.

Bibliography

[1] *Acoustic event detection in real life recordings*. Zenodo, Aug. 2010.

[2] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[3] S. Adavanne, G. Parascandolo, P. Pertilä, T. Heittola, and T. Virtanen. Sound event detection in multichannel audio using spatial and harmonic features. In *IEEE Detection and Classification of Acoustic Scenes and Events workshop*, 2016.

[4] D. Ardila, C. Resnick, A. Roberts, and D. Eck. Audio deepdream: Optimizing raw audio with convolutional networks.

[5] S. O. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, J. Raiman, S. Sengupta, et al. Deep voice: Real-time neural text-to-speech. *arXiv preprint arXiv:1702.07825*, 2017.

[6] Y. Aytar, C. Vondrick, and A. Torralba. Soundnet: Learning sound representations from unlabeled video. In *Advances in Neural Information Processing Systems*, pages 892–900, 2016.

[7] J. Barker, M. Cooke, and D. Ellis. Decoding speech in the presence of other sources. *Speech Communication*, 45(1):5–25, jan 2005.

[8] F. Briggs, B. Lakshminarayanan, L. Neal, X. Z. Fern, R. Raich, S. J. K. Hadley, A. S. Hadley, and M. G. Betts. Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach. *The Journal of the Acoustical Society of America*, 131(6):4640–4650, jun 2012.

[9] G. J. Brown and M. Cooke. Computational auditory scene analysis. *Computer Speech & Language*, 8(4):297–336, 1994.

[10] E. Cakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen. Convolutional recurrent neural networks for polyphonic sound event detection. *arXiv preprint arXiv:1702.06286*, 2017.

[11] I. Choi, K. Kwon, S. H. Bae, and N. S. Kim. Dnn-based sound event detection with exemplar-based approach for noise reduction.

[12] F. Chollet et al. Keras. https://github.com/fchollet/keras, 2015.

[13] S. Chu, S. Narayanan, C. c. Kuo, and M. Mataric. Where am I? Scene Recognition for Mobile Robots using Audio Features. In *2006 IEEE International Conference on Multimedia and Expo*. Institute of Electrical and Electronics Engineers (IEEE), jul 2006.

[14] C. Clavel, T. Ehrette, and G. Richard. Events detection for an audio-based surveillance system. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 1306–1309. IEEE, 2005.

[15] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[16] L. Deng, D. Yu, et al. Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387, 2014.

[17] S. Dieleman and B. Schrauwen. End-to-end learning for music audio. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 6964–6968. IEEE, 2014.

[18] S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, et al. Wavenet: A generative model for raw audio. 2016.

[19] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[20] J. F. Gemmeke, L. Vuegen, P. Karsmakers, B. Vanrumste, et al. An exemplar-based nmf approach to audio event detection. In *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*, pages 1–4. IEEE, 2013.

[21] D. Giannoulis, A. Klapuri, and M. D. Plumbley. Recognition of harmonic sounds in polyphonic audio using a missing feature approach. In *2013 IEEE International Conference on Acoustics Speech and Signal Processing*. Institute of Electrical and Electronics Engineers (IEEE), may 2013.

[22] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[23] M. Goto. An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, 30(2):159–171, 2001.

[24] T. Hayashi, S. Watanabe, T. Toda, T. Hori, J. Le Roux, and K. Takeda. Bidirectional lstm-hmm hybrid system for polyphonic sound event detection. In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, pages 35–39, 2016.

[25] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[26] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis. Joint optimization of masks and deep recurrent neural networks for monaural source separation. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 23(12):2136–2147, 2015.

[27] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[28] M. J. Kim and H. Kim. Automatic extraction of pornographic contents using radon transform based audio features. In *2011 9th International Workshop on Content-Based Multimedia Indexing (CBMI)*. Institute of Electrical and Electronics Engineers (IEEE), jun 2011.

[29] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[30] A. Klapuri and M. Davy. *Signal processing methods for music transcription*. Springer Science & Business Media, 2007.

[31] T. Komatsu, T. Toizumi, R. Kondo, and Y. Senda. Acoustic event detection method using semi-supervised non-negative matrix factorization with a mixture of local dictionaries. *Detection and Classification of Acoustic Scenes and Events 2016*, 2016.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[33] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[34] B. McFee, M. McVicar, O. Nieto, S. Balke, C. Thome, D. Liang, E. Battenberg, J. Moore, R. Bittner, R. Yamamoto, D. Ellis, F.-R. Stoter, D. Repetto, S. Waloschek, C. Carr, S. Kranzler, K. Choi, P. Viktorin, J. F. Santos, A. Holovaty, W. Pimenta, and H. Lee. librosa 0.5.0, Feb. 2017.

[35] A. Mesaros, T. Heittola, and T. Virtanen. Tut database for acoustic scene classification and sound event detection. In *Signal Processing Conference (EUSIPCO), 2016 24th European*, pages 1128–1132. IEEE, 2016.

[36] A. Mesaros, T. Heittola, and T. Virtanen. Tut sound events 2017, development dataset, Mar. 2017.

[37] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3, 2010.

[38] U. Muller, J. Ben, E. Cosatto, B. Flepp, and Y. L. Cun. Off-road obstacle avoidance through end-to-end learning. In *Advances in neural information processing systems*, pages 739–746, 2006.

[39] S. Ntalampiras, I. Potamitis, and N. Fakotakis. An Adaptive Framework for Acoustic Monitoring of Potential Hazards. *EURASIP Journal on Audio Speech, and Music Processing*, 2009:1–15, 2009.

[40] D. Palaz, R. Collobert, et al. Analysis of cnn-based speech recognition system using raw speech as input. Technical report, Idiap, 2015.

[41] V. Peltonen, J. Tuomi, A. Klapuri, J. Huopaniemi, and T. Sorsa. Computational auditory scene recognition. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 2, pages II–1941. IEEE, 2002.

[42] K. J. Piczak. Environmental sound classification with convolutional neural networks. In *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*, pages 1–6. IEEE, 2015.

[43] K. J. Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1015–1018. ACM, 2015.

[44] J. Pons, T. Lidy, and X. Serra. Experimenting with musically motivated convolutional neural networks. In *Content-Based Multimedia Indexing (CBMI), 2016 14th International Workshop on*, pages 1–6. IEEE, 2016.

[45] L. R. Rabiner and B.-H. Juang. Fundamentals of speech recognition. 1993.

[46] R. Ranft. Natural sound archives: past present and future. *Anais da Academia Brasileira de Ciências*, 76(2):456–460, jun 2004.

[47] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[48] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals. Learning the speech front-end with raw waveform cldnns. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[49] H. Sak, A. Senior, K. Rao, O. Irsoy, A. Graves, F. Beaufays, and J. Schalkwyk. Learning acoustic frame labeling for speech recognition with recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 4280–4284. IEEE, 2015.

[50] J. Salamon and J. P. Bello. Feature learning with deep scattering for urban sound analysis. In *Signal Processing Conference (EUSIPCO), 2015 23rd European*, pages 724–728. IEEE, 2015.

[51] J. Salamon and J. P. Bello. Unsupervised feature learning for urban sound classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 171–175. IEEE, 2015.

[52] J. Salamon and J. P. Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283, 2017.

[53] J. Salamon, C. Jacoby, and J. P. Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1041–1044. ACM, 2014.

[54] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[55] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley. Detection and classification of acoustic scenes and events. *IEEE Transactions on Multimedia*, 17(10):1733–1746, 2015.

[56] P. Susini, N. Misdariis, G. Lemaitre, O. Houix, D. Rocchesso, P. Polotti, K. Franinovic, Y. Visell, K. Obermayer, H. Purwins, et al. Closing the loop of sound evaluation and design. *Perceptual Quality of Systems*, 2(4), 2006.

[57] A. Temko, R. Malkin, C. Zieger, D. Macho, C. Nadeu, and M. Omologo. Clear evaluation of acoustic event detection and classification systems. In *International Evaluation Workshop on Classification of Events, Activities and Relationships*, pages 311–322. Springer, 2006.

[58] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302, 2002.

[59] J.-M. Valin, F. Michaud, B. Hadjou, and J. Rouat. Localization of simultaneous moving sound sources for mobile robot using a frequency-domain steered beamformer approach. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, pages 1033–1038. IEEE, 2004.

[60] A. van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798, 2016.

[61] T. Virtanen, A. Mesaros, T. Heittola, M. Plumbley, P. Foster, E. Benetos, and M. Lagrange. *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*. Tampere University of Technology. Department of Signal Processing, 2016.

[62] Y. Xu, Q. Kong, Q. Huang, W. Wang, and M. D. Plumbley. Convolutional gated recurrent neural network incorporating spatial features for audio tagging. *arXiv preprint arXiv:1702.07787*, 2017.

[63] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.

[64] X. Zhuang, X. Zhou, M. A. Hasegawa-Johnson, and T. S. Huang. Real-world acoustic event detection. *Pattern Recognition Letters*, 31(12):1543–1551, 2010.

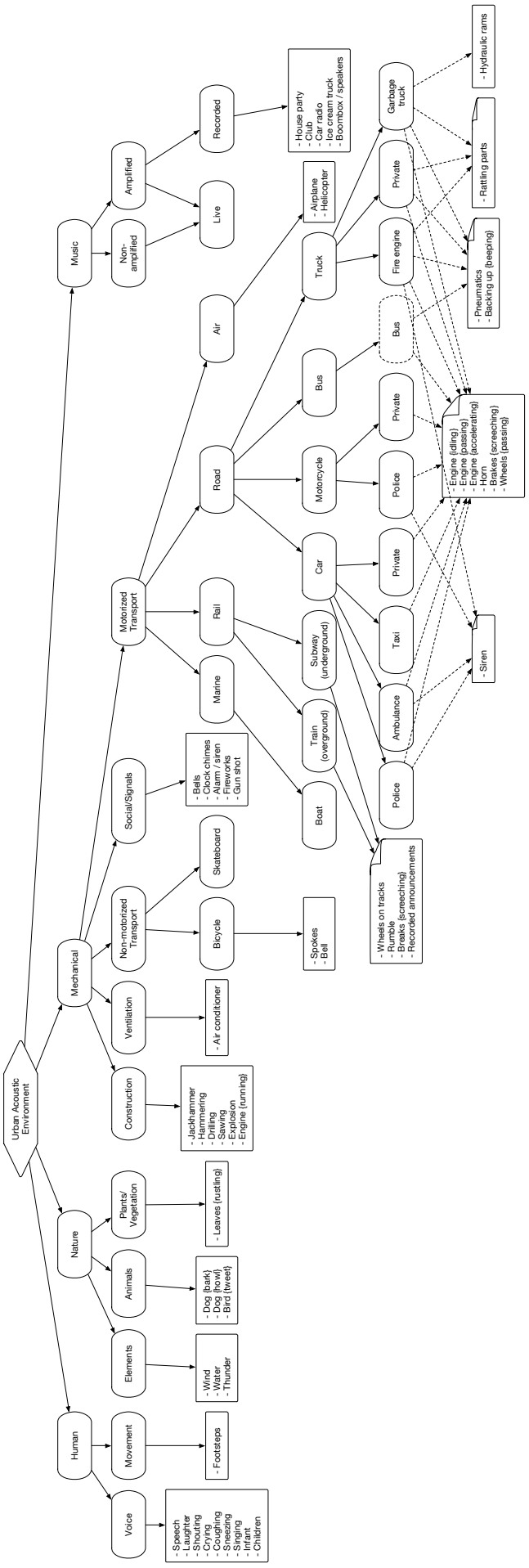
[65] M. Zöhrer and F. Pernkopf. Gated recurrent networks applied to acoustic scene classification and acoustic event detection. *Detection and Classification of Acoustic Scenes and Events*, 2016, 2016.

Appendix A

UrbanSound8K taxonomy

This appendix contains the taxonomy presented by Salamon et. al. [53] which is used to create the UrbanSound and UrbanSound8K datasets.

Figure A.1: UrbanSound taxonomy



Appendix B

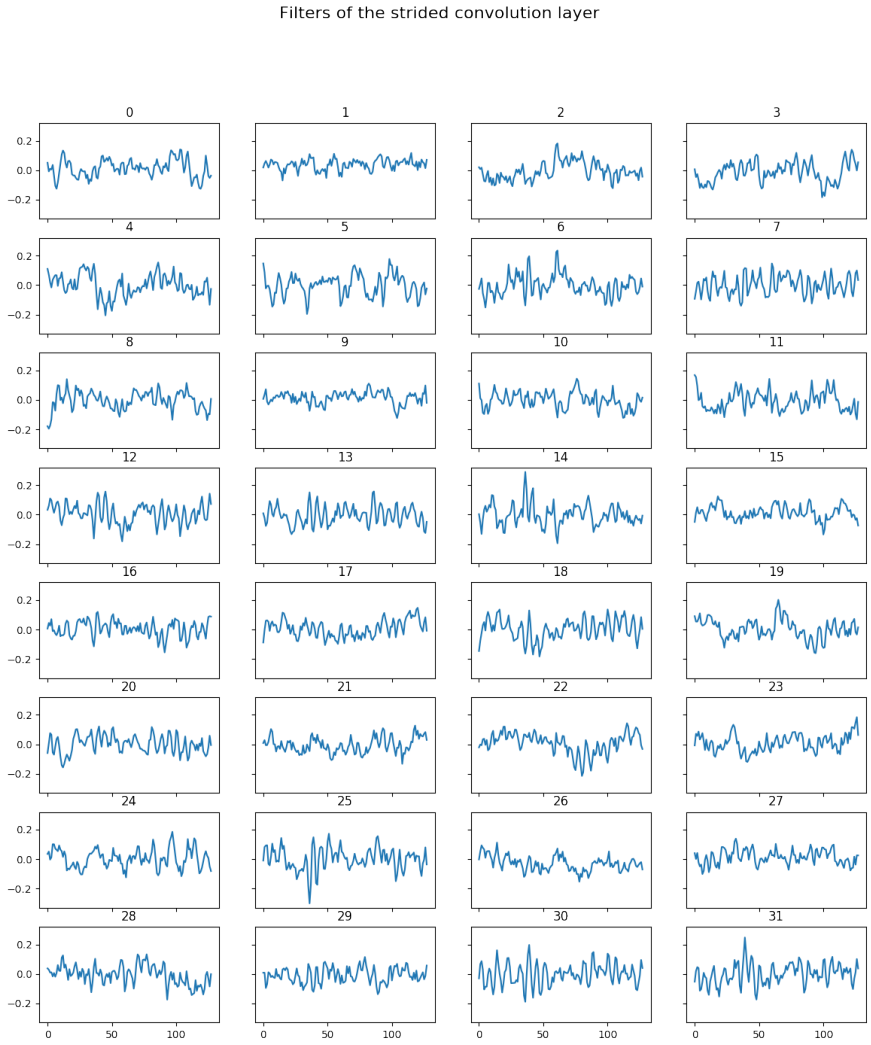
Keras summary of RawCNN

Layer (type)	Output Shape	Param #
input_9 (InputLayer)	(None, 180809, 1)	0
conv1d_9 (Conv1D)	(None, 1412, 32)	4128
block_1_conv (Conv1D)	(None, 1412, 64)	65600
block_1_act (Activation)	(None, 1412, 64)	0
block_1_pool (MaxPooling1D)	(None, 353, 64)	0
block_2_conv (Conv1D)	(None, 353, 128)	131200
block_2_act (Activation)	(None, 353, 128)	0
block_2_pool (MaxPooling1D)	(None, 88, 128)	0
block_3_conv (Conv1D)	(None, 88, 256)	262400
block_3_act (Activation)	(None, 88, 256)	0
block_3_pool (MaxPooling1D)	(None, 22, 256)	0
flatten_9 (Flatten)	(None, 5632)	0
dropout_17 (Dropout)	(None, 5632)	0

```
_________________________________________________________________
dense_17 (Dense)             (None, 100)               563300
_________________________________________________________________
relu (Activation)            (None, 100)               0
_________________________________________________________________
dropout_18 (Dropout)         (None, 100)               0
_________________________________________________________________
dense_18 (Dense)             (None, 10)                1010
=================================================================
Total params: 1,027,638
Trainable params: 1,027,638
Non-trainable params: 0
_________________________________________________________________
```

Appendix C

Filters of the strided convolutional layer


Filters of the strided convolution layer
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
0.2
0.0
−0.2
0
50
100