Classifying Electrocardiograms Using Deep Learning

Fredrik Kindström frekin@student.chalmers.se
Alexander Håkansson alehak@student.chalmers.se
Shruthi Dinakaran shruthi@student.chalmers.se
Giovanni Pagliarini guspaglgi@student.gu.se
Raya Altarabulsi gusaltara@student.gu.se

Abstract

In recent years, there has been increasing interest in the practical application of Artificial Intelligence to several fields. Astonishing results have already been observed in a wide range of different areas, and there is much ongoing research exploring the actual potential of Machine Learning. Tools such as artificial neural networks, coupled with learning algorithms, have proven to be versatile techniques for automating hard tasks, and in some cases, AI even succeeded in achieving super-human efficacy. This leaves room for discussion on the ethics and responsibility of AI deployment.

In this essay, we report the results attained in training neural networks for diagnosing a heart condition by processing a patient's electrocardiogram (ECG). In particular, we tune and assess the performances of two networks of different types, namely convolutional neural networks and recurrent neural networks. Both kinds have reasons to perform well on ECG signals, which consist of time-distributed measurements of the heartbeat. The models are evaluated in terms of accuracy, and the best model achieves an accuracy of 99.44%. We also present some of the methods used in the training process, give an overview of similar studies, and finally discuss the results, with emphasis on the big-picture meaning of such achievements.

1 Introduction

Automating tedious tasks have long been the very focus for computers, and modern artificial intelligence applications are in some sense an extension of this, in the same way simple number crunching done by computers has been a staple part of our businesses for over half a decade now. We want computers to do the tedious work, and in some cases even more than that, we want them to do everything for us. The last request has been historically difficult, simply because computers are not able to think for themselves.

Artificial intelligence is what we call it when an application attempts to cross the border of intelligent computing. To understand how this might work, it's meaningful to try and look at and draw inspiration from how we, humans, do it. When you open the lid and look inside some of the most powerful and *intelligent* modern applications, it turns out that most of the so called intelligence comes down to mathematical statistics and trial-and-error.

One branch in machine learning that has gained a lot of momentum in recent years is artificial neural networks, which is, simply put, a model that draws inspiration from how the human brain in its most basic form. Despite the initial primitive simplification used, this method has proven to be extraordinarily good at certain tasks. Specifically at classifying large amounts of high-dimensional data, a substance that the last ten years of information technology advancements has left us drowning in.

There are of course an infinite amount of valid applications for this technique, but in this project we have chosen to focus on the task of classifying electrocardiogram signals, often called *ECG signals*, of the human heart.

1.1 Motivation

The aim of this project is to investigate the possibilities of classifying ECG signals using artificial neural networks. Electrocardiogram is a method of measuring cardiac activity in terms of frequency and character of heartbeats. This has a number of useful future applications and ECG is a suitable medical data to work on since the human heart sits at the centre of much of our general well-being.

A challenge when attacking the problem is that the ECG-data provided online for training is often pre-processed and aligned. This does not give a good representation of how a live ECG signal might look like in the real world. In order to get more satisfying results, the project needs to take this fact into consideration and feed the networks data that better represents how real signals look like.

2 Artificial Neural Networks

As mentioned previously, an artificial neural network is a model inspired by biological neural networks [1]. The model revolves around the concept of neurons, which are, in their simplest form, computational units taking some input, mutating it according to some rule, and outputting the result. The output of a neuron can be passed on as input to other neurons, thus neurons can be chained and assembled together in large structures which overall are able to approximate any multivariable function. This is known as the universal approximation theorem.

For a given task, the correct values for a neural network's parameters are not known a priori. The common method used to *learn* them involves starting with a random initial configuration and iteratively nudging them to approach the desired behavior using the backpropagation algorithm [2]. In this process, the network structure plays a determinant role: typically, neurons are organized into layers, and network classifiers often tend to have a funneled shape, in the sense that each layer tries to represent the same data of previous layer with less neurons.

For this reason, bottle-necks are introduced between the input layer (usually the largest, since it needs to represent the data in its whole original form, e.g a node for each pixel in an image) and the output layer (simply expressing the classification outcome), and they serve as constraints forcing the network to learn ways to condense and represent the same data in a different form.

2.1 Convolutional Neural Networks (CNN)

Convolutional neural networks (CNNs) is a class of artificial neural networks specialized for pattern recognition, and it is mostly used for analysing or classifying images. A regular feed-forward neural network can be used for the same purpose of classifying data but, as it would need a single neuron for each data input, it does not scale well for large input sizes such as images. A CNN provide a solution for this problem by including, besides the usual hidden layers, a special type of hidden layer, called *convolutional*, which performs the input-output transformation as a convolution operation.

Convolutional layer. Unlike fully connected layers, where each neuron receives data from the entire previous layer, a neuron in a convolutional layer receives data from a fixed size subarea of the previous layer called a receptive field. Each convolutional layer has a number of *filters* that consists of many receptive fields and these filters make it able to detect different patterns. A filter can be thought of as a relatively small matrix that will convolve (or slide) across the input data.

Each filter is responsible to detect a pattern and in the case of image analysis, for example, filters can detect edges, circles, squares, etc. These filters can learn to detect complex patterns, such as a dog's ear, or eyes [3].

Pooling is a sample-based discretization process. The objective is to down-sample an input representation, reducing its dimensionality and allowing for assumptions to be made about features contained in the sub-regions binned.

Max pooling is popular among the various pooling methods: When convolutional layers interweave with max pooling layers, the output identifies whether a certain feature was present in any region of the previous layer, although not precisely which one. Max pooling layers act as a *zoom out* operation, summarizing data regions (think of an image being compressed), and overall makes the network invariant to minor transformations of the input.

2.2 Recurrent Neural Networks (RNN)

A recurrent neural network (RNN) is a type of network where the output from some nodes in a layer depends on previous context, unlike the regular neural network that only has one source of input. This differentiation is implemented as a feedback loop connected to the RNN's past decisions. So the output of a recurrent network depends also on time, which proceeds in discrete steps. RNNs are commonly used for tasks involving text processing, speech recognition, and times series data.

It can be thought of as adding a memory to the network, and the information circulates in the hidden states of the recurrent network to process sequences of input, similarly to how our human memory works.

This behaviour is captured in *Long Short-Term Memory* (or LSTM), a special type of unit commonly used in RNNs.

2.3 Dropout

Dropout is a method of mitigating the effect of overfitting [4]. Between two layers, for every iteration of training every edge between two nodes has a probability of p to be zeroed out. The probability p is often referred to as the dropout rate. The idea is that dropout forces the network to generalize, as it can no longer rely on a single node.

2.4 Early Stopping

Early stopping is another method, like dropout, which also mitigates the effect of overfitting [5]. During training of the model, both the performance against the training and validation data is evaluated. If the performance against the validation data starts to decrease, it is symptom that the model starts to overfit on the training data, so there is actually no point int training anymore. Some notion of patience can be incorporated, in order to allow temporary drops in performance.

2.5 k-fold Cross-validation

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation [6]. This approach involves randomly dividing the datasets into k groups or folds, of approximately equal size. The first fold is treated as a validation set and the method is fit on the remaining $k - 1$ folds.

It is used to estimate the skill of a machine learning model on unseen data. It uses a limited sample in order to estimate how the model is expected to perform when used to make predictions on data not used during the training of the model. It generally results in a less biased estimate of the model's abilities, rather than other methods like a simple train/test split. The results of a k-fold cross-validation run are often summarized with the mean of the model skill scores.

3 Methods

In this section, the dataset in use is presented, and the process of developing the classifiers described.

3.1 Dataset

The original dataset consists of 5000 signals representing the heartbeats of a patient with severe congestive heart failure [7]. Each signal is made up of 140 data points, and had been labeled as belonging to one out of five classes in a previous work [8].

In our study, we limited the classification to recognize two of these classes, namely *regular heartbeat* and *R-on-T Premature Ventricular Contraction*. The data was downloaded and stripped of the classes not of interest. Finally, the resulting set was randomly sampled, restricting the number of signals to 1805, which were randomly split into three subsets: 60% of the samples to be used as training data, 30% as validation data, and a set of 10% for final testing.

In order to simulate a real world scenario, in which the input is an (ideally infinite) stream of datapoints, a circular shift of random size was performed on each sample. In principle, this operation allows a well-trained network to recognize unsynchronized (i.e not aligned) signals, and can therefore be implemented as an online classifier.

3.2 Defining the model

Mainly two different types of artificial neural networks are explored as part of this project; convolutional neural networks and recurrent neural networks. The models were implemented using the Python machine learning framework *Keras* [9]. Common for both types of models in this project is that they use the raw data signals, without any pre-processing. The final layer, the output layer, has the same structure in both models: it is a fully connected layer with a single neuron, using the sigmoid function as activation. This means that the models will output a number between zero and one. If the number is larger than $\frac{1}{2}$ the

input signal will be classified as non-healthy, otherwise it will be classified as healthy (i.e representing a regular heartbeat). The Adam optimizer [10] is used in both model architectures for training, but the learning rate of it was left as a hyperparameter.

For the convolutional neural network model, one-dimensional convolutional layers, max pooling layers, dropout layers and fully connected layers were ultimately used. The architecture, along with the hyperparameters of the layers, were experimented with as the project progressed. The main idea of the final model, as shown in Figure 1a, is that the input signal is funneled down through a series of convolutional and max pooling layers to find patterns, which are then fed through a single fully connected layer before producing an output. For this convolutional model, the following hyperparameters were considered: filter size of convolutional layers, kernel size of the convolutional layers, pool size in the max pooling layers, dropout rate for the dropout layers, and the neuron count in the final fully connected layer. All of the layer, besides the final output layer, uses the ReLU [11] activation function.

The recurrent model initially comprised a single hidden LSTM layer with the identity function as activation; later in the project the use of a second fully connected layer was investigated. The following hyper-parameters were studied: the output dimension and activation function of the recurrent layer, the patience parameter for early stopping.

The dropout layers were added at the end of the project, in an attempt to squeeze out the last bit of performance of the models by trying to reduce overfitting [4]. This was especially successful in the convolutional model.

The models were trained using mini-batch training, which is commonly used for speeding up the training of artificial neural networks [12]. Binary cross entropy loss was used as the cost function when training the model. The training of the models also utilized Early Stopping [5], as a way of reducing overfitting. Since the dataset used for this project is fairly small, k-fold cross-validation is used to better assess how well the models performed — with k set to 10. In each iteration of the cross-validation, the model was trained on the the training data, and after each epoch of the training it was evaluated against the validation data. Since early stopping was used, the training would then stop when the model started overfitting to the training data. A setting in Keras made sure that the model was then rolled back to the epoch where it performed the best. For evaluating if one model performed better than another, the average accuracy of the 10-fold cross-validation for each model was compared.

4 Related work

The subject of classifying ECG signals is, not surprisingly, quite established as a good application of artificial intelligence using machine learning for science. There have been several publications on the subject.

In work done by Sahar H. El-Khafif and Mohamed A. El-Brawany [13], they showed that a classifier could be trained to detect specific conditions, such as ischemic heart diseases. The researchers found that they could correctly classify as much as 93 percent of the data they used for testing. While this result is not as high as some successful neural net models used in practice, or this groups results, it is still a good indication that there is potential in using these techniques to aid medical professionals and potentially save lives thanks to quicker processes in hospitals. Also for more involved analysis like a specific disorder.

Another recent study showed some even more promising results [14]. The scientists trained a complex neural net model on a massive dataset and found that their model could outperform human professionals in detecting several heart conditions when reviewing the same data. This is interesting since the study actually proved an actual benefit over humans at a particular task that is practiced everyday.

5 Result

The accuracies of the two best performing model configurations for the convolutional and recurrent models are shown in Table 1. The table shows that the recurrent model got an accuracy of 0.97777 and the convolutional model got an accuracy of 0.99444.

Model	Accuracy
Recurrent neural network model	0.97777
Convolutional neural network model	0.99444

Table 1: Accuracy of the convolutional and recurrent neural network models.

The structure of the convolutional model that achieved the accuracy shown in Table 1 is represented in Figure 1a. The first convolutional layer has 50 filters, with a kernel size of 5. After this layer there is a dropout layer with a 0.05 dropout rate. This dropout layer feeds into a max pooling layer of size 10, which in turn feeds into another convolutional layer with 25 filters of kernel size 3. After this second convolutional layer, there is another dropout with 0.05 dropout rate. Finally, the signals feed into a fully connected layer with 15 neurons.

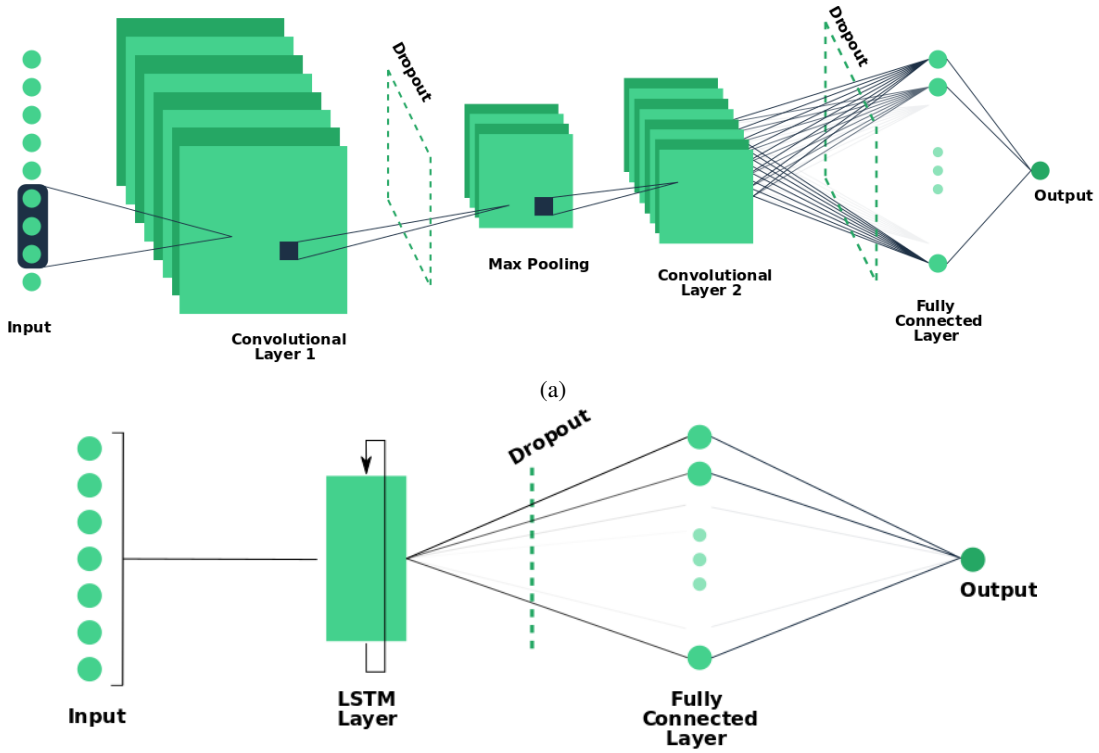

Figure 1: Depiction of the convolutional (a) and recurrent (b) models implemented in the project. The first includes two convolutional layers, one of which followed by a max pooling one. In the recurrent model, the input flows through a LSTM layer. Both models share a final fully connected layer and a single-neuron output representation; dropout is introduced after the convolutionals and LSTM layers.

The recurrent model already showed high accuracies with a single LSTM layer ($\sim 98\%$ on the validation data), and no improvements were measured with the introduction of a second, fully connected, layer. In particular, most of the tuning process was hindered by high fluctuations ($\pm 0.5\%$) between different runs, due to the randomized initial training configurations. This, together with the a small room for improvement, made it difficult to make direct assumptions on how hyper-parameters affects the outcome.

Ultimately, the above accuracy for the LSTM model was attained with 140 LSTM units on the recurrent layer followed by a 10% dropout rate, and a 5-neuron second, feed-forward, hidden layer. The most appropriate learning rate for this architecture seems to be 0.01. As regards of early-stopping, the trend suggest that good values for the patience lay between 4 and 16 epochs; in this context, this is the number of epochs to wait before aborting, while the accuracy keeps decreasing.

6 Discussion

The resulting accuracies on the test data are very high, surely above the group's initial expectations: considering that the techniques used for attaining such precision are quite standard, on the whole we suspect that the simplified task of classifying signals from only two classes might have been more trivial than expected.

One thing this experience seems to suggest is that ECG provides very suitable data for this particular technology. In fact, the outcome is even more impressive considering the reduced size of the dataset in use, quite limited compared to related papers.

6.1 Drawbacks and challenges

A re-occurring theme in human tech-history is that whenever a new and promising tech field is developing, people tend to put too much trust and overlook a lot of the challenges with implementing the tech to scale. One such challenge that is fortunately gaining more recognition is the realization that a system based on artificial intelligence can also become quite biased in it's interpretations. A common misconception is that a computer is always, by design, objective.

This mindset causes problems that are particularly devastating when it comes to artificial intelligence. Since AI is biased on learning on some data, the final model is highly dependant on the data we feed it. If we feed it tainted data, then the model will give tainted predictions.

As for ECG, for instance, a research team might train a network on data that comes from patients in a certain geographical area (most likely where the research team is located), and implicitly infer that the results generalize to people from all around the world. This might cause faulty results since heart conditions might be caused by multiple and diverse environmental factors. ECG signals have also been shown to differ between gender [15] and that might also lead to a biased AI when only training on mostly male signals, for example. AI does not remove any of the challenges that we have to face when doing research the traditional way.

7 Future work

A first improvement on our work could be to train the models on a much larger dataset. Especially given the general availability of data, this step seems essential in order develop solid classifiers that can actually be battle-tested in the real world.

The practical application of this study is, of course, the deployment of a trained network directly into medical contexts. Because of the out-of-phase capabilities that our convolutional model learned, a hardware implementation would not need special detectors dedicated to signal synchronization, but would be able to tap directly into live, raw ECG signal and analyze it in real time at a lower cost. This could then serve as an alarm system in a wearable electronic (smartwatch-like, for instance). If the system detects an episode of alarming heart activity, medical personnel could be notified immediately. The complex abilities of a neural net as opposed to simple value checks make it possible to also monitor more intricate conditions with such a system.

Distribution of work in project

Giovanni, Shruti, and Alexander wrote all of the code in the programming project. The artificial neural network models were primarily developed by Giovanni and Alexander. For the essay, all authors contributed equally as much.

References

[1] R. P. Lippmann, "An introduction to computing with neural nets,"" *IEEE Assp magazine*, vol. 4, no. 2, pp. 4–22, 1987.

[2] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception*, Elsevier, 1992, pp. 65–93.

[3] S. Haykin, *Neural networks*. Prentice hall New York, 1994, vol. 2.

[4] 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*, vol. 15, no. 1, pp. 1929–1958, 2014.

[5] L. Prechelt, "Early stopping-but when?" In *Neural Networks: Tricks of the trade*, Springer, 1998, pp. 55–69.

[6] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 112.

[7] E. K. Y. Chen, *Dataset: ECG5000 – Time Series Classification*, `http://www.timeseriesclassification.com/description.php?Dataset=ECG5000`, [Online; accessed 01-March-2019].

[8] Y. Chen, Y. Hao, T. Rakthanmanon, J. Zakaria, B. Hu, and E. Keogh, "A general framework for never-ending learning from time series streams," *Data Mining and Knowledge Discovery*, vol. 29, no. 6, pp. 1622–1664, Nov. 2015, ISSN: 1573-756X. DOI: `10.1007/s10618-014-0388-4`. [Online]. Available: `https://doi.org/10.1007/s10618-014-0388-4`.

[9] F. Chollet *et al.*, *Keras*, `https://keras.io`, 2015.

[10] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[11] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

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

[13] S. H. El-Khafif and M. A. El-Brawany, "Artificial neural network-based automated ecg signal classifier," *ISRN Biomedical Engineering*, vol. 2013, 2013.

[14] P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, C. Bourn, and A. Y. Ng, "Cardiologist-level arrhythmia detection with convolutional neural networks," *arXiv preprint arXiv:1707.01836*, 2017.

[15] H. Mieszczanska, G. Pietrasik, K. Piotrowicz, S. McNitt, A. J. Moss, and W. Zareba, "Gender-related differences in electrocardiographic parameters and their association with cardiac events in patients after myocardial infarction," *The American journal of cardiology*, vol. 101, no. 1, pp. 20–24, 2008.