# DL LAB 2: An application with RNN

Senne Deproost and Diego Saby

November 2019

## 1 Introduction

In the field of machine learning we try create computational models to find solutions for a wide set of tasks. One of the more popular problems to solve are the classification of instances and function approximation with regression. Solutions like *Support Vector Machines* and *Decision Trees* have already proven their worth in the past. Techniques from the deep learning (DL) sub-field of machine learning have shown their usability when tackling on data with high dimensionality. Deep learning's ability to scale better with these kinds of data makes it useful in many application domains like machine control and decision support systems. Variations in the model's architecture could insure better performance on a domain specific task.

One DL architecture we'll focus on in this lab report is the *Recurrent Neural Network* architecture. Whereas classical *Feedforward Neural Network* (FNN) only allows the passage of activation through the neuron layers, a RNN network will remember the neuron activations from a previous step in time. This is implemented with the inclusion of memory nodes that will hold on to previous encountered activation values. An advantage of this kind of networks is the usage of *Temporal Sequences*, allowing the model to train on data like text sentences, music recordings and streams of data.

Automated content creation using DL models has known a significant rise in popularity over the past years. With *Generative Adverserial Networks* (GAN), we can use two competing networks to train a generator for images (Goodfellow et al., 2014). In this architecture, one network, the has to mimic given image data input distribution and another adversarial network that has to estimate the loss of the predicted regression with given training data. This prediction can then be used to optimize the generative network.

- talk about RNN and LSTM

- Talk about how RNN is useful for music
- Introduce our project -¿ RNN on Chopin Nocturnes
- Something else?...

## 2 Preprocess the data

One of the first questions we were opposed to was how to encode music in order to train the Neural Network. Our source files are midi files. They are an useful way to encode the music where we can extract the notes and their duration from the file.
To simplify the problem we decided to abstract the notes duration and just use the notes pitch.

### 2.1 Enumerating the notes

Our first attempt was done inspired by some code found on the internet to produce video games music. **add reference**
To extract the notes the notes from midi file we used a library called *Music21*. It provided us with the notes names or chord names that were played. From there we have a sequence of notes names. Then we just assign the notes a number. For the prediction we used a vector of all the notes and a with a 0 for the next note that follows the sequence.

## References

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative Adversarial Nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 27* (pp. 2672–2680). Curran Associates, Inc. Retrieved from `http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf`