Generating Digits and Sounds with Artificial Neural Networks

1st Maroš Stredanský
Katedra kybernetiky a umelej inteligencie
Technická univerzita v Košiciach
Košice, Slovenská republika
maros.stredansky@student.tuke.sk

2nd Tomáš Vank
Katedra kybernetiky a umelej inteligencie
Technická univerzita v Košiciach
Košice, Slovenská republika
tomas.vank@student.tuke.sk

3nd Martin Števlík
Katedra kybernetiky
Technická univerzita v Košiciach
Košice, Slovenská republika
martin.stevlik@student.tuke.sk

***Abstract*—Neural networks belong among the special tools of machine learning. Recurrent neural networks have been designed for time sequence modelling, and unlike conventional forward neural networks, they include recurrent connections, that allow the information from the previous season to affect the activity of neurons in the current time. Recurrent networks have been successfully used in repetitive applications to solve tasks of prediction, management adaptation, system identification or information processing. An important factor limiting their wider use is the complexity of the algorithms used to train them.**

***Index Terms*—networks,neural networks,music,evolutionary computation.**

I. Introduction

Artificial neural networks are computational models inspired by biological neural networks, such as the human brain. The parallel with nature is in the way of realization of the calculation - the resulting complicated calculation is created by the interaction of many elements making the calculations easier. Many types of neural networks have been designed [1], though surely the most popular include multilayer perceptron neural networks that are trained by the error propagation algorithm.

From an engineering point of view, neural networks are popular and frequently used computational models, especially due to their ease of use and the ability to quickly achieve satisfactory results. A neural network consists of a plurality of computational elements, called neurons, which are interconnected via weight links of varying intensity. Training of the neural network consists of modifying the intensities of the weighted links so that the neural network responds accordingly to the inputs, to produce the desired outputs.

Classical forward neural networks are used for example, when dealing with classification tasks (determining whether an object has a given property) or regression (finding a dependency describing the relationship between variables). Recurrent neural networks have been designed to handle tasks with a time context such as prediction (predicting the following value or values in sequence).

Recurrent neural networks can be considered as a simple modification of the forward neural networks resulting from the addition of so-called ”neural networks”. recurrent links. Recurrent linkages associated neurons in such a way that neuronal activity from the previous time step through the recurrent linkage affects neuron activity at the current time step.

Forward links carrying information within the current calculation step are shown in solid line, and recurrent links carrying information from the previous to the current calculation step are shown in dashed line.

Recurrent neural networks have been successfully used in several practical tasks requiring time sequence modelling. Several architectures, training algorithms and their modifications have been designed, but recurrent neural networks are still not commonly used as machine learning tools. The reason is the significantly higher computational complexity of training algorithms compared to classical forward neural networks. Successful application of recurrent networks also usually requires a deeper knowledge of the problem task, the type of recurrent network used, as well as the chosen training algorithm.[1]

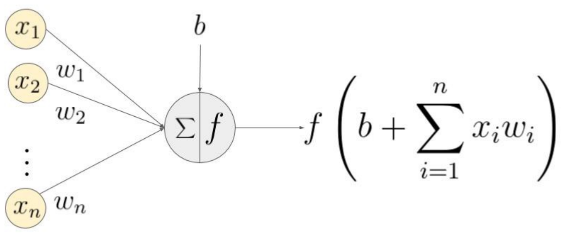
II. Introduction of terms

A. Forward neural networks

Multilayer Perceptron is the most commonly used type of neural network [17]. We use the feedforward attribute in the context of recurrent neural networks to emphasize the direction of information flow. In this subsection, we briefly summarize the basic information about forwarding neural networks and briefly describe the error propagation algorithm

B. Neuron

In the field of artificial neural networks, the term neuron denotes simple elements of which artificial neural networks are composed. Perceptron is a neuron that performs a simple mathematical calculation. It contains N inputs 1x to Nx, which are connected to it via weight connections with intensities w1 to wN. The intensity, or weight, of the link, determines how strongly a given input affects the resulting neuron activity.[2]The calculation performed by a neuron can thus be written as:



An example of a neuron showing the input (x_1 - x_n), their corresponding weights (w_1 - w_n), a bias (b) and the activation function f applied to the weighted sum of the inputs.

The calculation performed by a neuron

Where o indicates so-called the fundamental activity of the neuron, o indicates the output activity of the neuron and is the threshold of the neuron. It is advisable to consider the threshold of a neuron as a special weight w0 (= w0) coming from a special input 0 x always set constant to 1. The relation often referred to in the literature, in which the sum wi xi threshold is subtracted, is equivalent to equation (1) given that the threshold may also be negative. The function f is called activation or transition function. A frequently used activation function is a sigmoidal function.[2] Here is the relation to the calculation of the functional value and the calculation of the derivative:

$$\begin{aligned}\frac{ds(x)}{dx} &= \frac{1}{1+e^{-x}}\\ &= \left(\frac{1}{1+e^{-x}}\right)^2 \frac{d}{dx}\left(1+e^{-x}\right)\\ &= \left(\frac{1}{1+e^{-x}}\right)^2 e^{-x}(-1)\\ &= \left(\frac{1}{1+e^{-x}}\right)\left(\frac{-e^{-x}}{1+e^{-x}}\right)\\ &= s(x)(1-s(x))\end{aligned}$$

Derivative of the Sigmoid Function

III. Creative Neural Network Model for Automated Melody Generation

The first study that we have been observed was Deep Artificial Composer by Florian Colombo, Alexander Seeholzer and Wulfram Gerstner. Their study describes a creative Neural Network Model for Automated Melody Generation. They were presenting the Deep Artificial Composer (DAC), a recurrent neural network model of note transitions for the automated composition of melodies, their model can be trained to produce melodies with compositional structures extracted from large datasets (5-10 GB) of diverse styles of music. They asses the creativity of DAC – generated melodies by a new measure, the novelty of musical sequences, showing that melodies imagined by the DAC are as novel as melodies produced by human composers. [3]

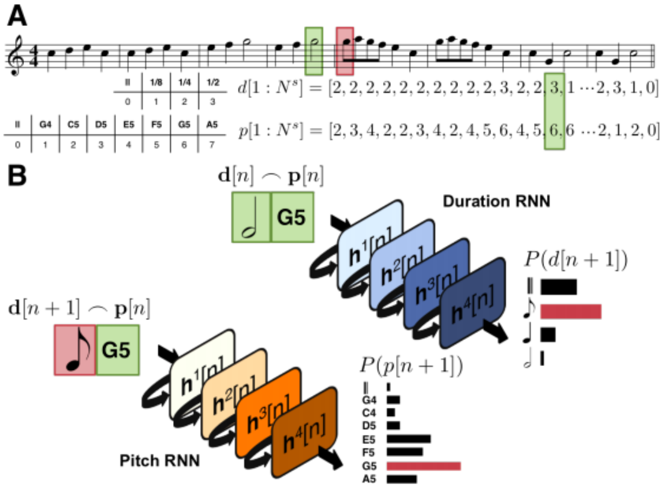
A. *General Statistical formalism for melody generation.*

They considered a monophonic melody as a sequence of N notes x[1 : N]. The DAC is built under the assumption that each note x[n] is drawn sequentially from a probability distribution (the "note transition distribution") T[n] over all possible notes. Generally, for a melody s, the transition distribution of note at time step n will depend on song dependent inputs s [n] (e.g. the history of notes up to note n) and a set of fixed model parameters . Then, the probability of a melody s could be expressed as the probability of all notes occurring in sequence:

$$P(x^s[1:N^s]) = \prod_{n=1}^{N^s} T\left(x^s[n] \mid \Phi^s[n], \Theta\right) . \tag{1}$$

All notes occurring in sequence

B. *Deep Artificial Composer*



Deep Artificial Composer networks

Deep Artificial Composer networks: (A) Representation of the nursery rhyme Brother John in the DAC by two sequences of integers d[1 : N s] and d[1 : N s]. Each integer in the top sequence corresponds to a single duration given by the duration alphabet to the left. Similarly, each integer in the bottom sequence corresponds to a single pitch given by the alphabet of pitches to the left. (B) The flow of information in DAC networks. The current duration d[n] and pitch p[n] are inputs to the duration network (bottom), while the pitch network (top) receives as inputs the current pitch p[n] and the upcoming duration p[n + 1]. Both RNNs learn to approximate the distribution over the next possible durations and pitches based on their inputs and their previous hidden state (recurrent connections). To generate melodies from the artificial composer, the upcoming duration and pitch are sampled from the estimated transition distributions (bars on the lower right). However, the inputs and parameters of this distribution have to be precisely defined – they could in principle encompass many different dimensions: the notes, the harmony, the tempo, the dynamic, and meta-parameters like for example the composer's musical and non musical experiences, or his general state of mind at the time of composition They tried to present and evaluate the Deep Artificial Composer model for the algorithmic composition of melodies. Using of trainable set allowed them automatically create monophonic music, their model can be trained on an corpus in the MIDI format.[3]

They Showed DAC network as network that is able to learn similarities and differences between different music styles what is hard problem to solve. They are showing results on the example of Irish and Klezmer music. Indeed, most of the melodies generated by the trained artificial composer are consistent in style – when the model starts generating a melody with a structure close to an Irish folk song, the entire generated melody is close to the structure of an Irish tune. In similar ways, provided that it has been sufficiently trained, the artificial composer is also consistent in scale and rhythm. [2]

C. Learning

Their model has to learn to correctly represent past events in order to be able to condition predictions on their internal representation of the history. By doing it, repetitions and more complex structures can be encoded and reused by the model during the generation of new melodies. The only way for the model to exactly repeat the same sequence twice is to produce Dirac distributions. They demonstrated that the DAC as a model of melody composition produces melodies that are as similar to melodies in the training set as real tunes of the same style are.

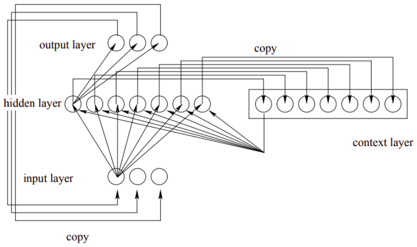
They introduced a measure of creativity of generated sequences, by means of computing the novelty of produced sequences with respect to the trained corpus. They used the novelty measurement to classify the style of melodies produced by the DAC. A much more sophisticated alternative would be to include such a classifier in the DAC itself by adding labeled style input and output units.

IV. Cybernetic Composer

The second study describes a music composition as a domain well-suited for evolutionary reinforcement learning. Instead of applying explicit composition rules, a neural network is used to generate melodies. An evolutionary algorithm is used to find a neural network that maximizes the chance of generating good melodies. Composition rules on tonality and rhythm are used as a fitness function for the evolution. We observe that the model learns to generate melodies according to these rules with interesting variations. This study written by Chun-Chi J. Chen and Risto Miikkulainen was quite interesting because they tried to create a melodies with Evolving Recurrent Neural Networks. Using neural networks on music composition is inherently hard because their behavior is not easy to predict nor control. However, the connectionist approach promised to give more flexibility and an ability to create novel situations, which makes it a suitable method for their experiment. [4]

A. Architecture

The most common neural network architecture is the three-layer feed-forward network. However, such a network does not contain a mechanism to remember past history. This makes it unsuitable for the music generation task because repetitive rhythmic patterns and pitches are important elements in composition. Instead, a recurrent network such as the SRN [6] is necessary. Our model is based on the SRN with the following input/output scheme (figure 1): The input layer represents a measure at time T, and the output layer represents the measure at time T 1. In other words, we feed the SRN network the current measure as input to get the next measure and this way compose the melody one measure at the time. [4]



This figure describes the melody generation network where the values of the output nodes at time t are copied to the input nodes at time T 1, and a copy of the hidden layer is saved in the context layer. This way, the network can generate output sequences from an initial starting point. The network is fully connected in the forward direction, and its forward weights are evolved [4].

V. DeepJ

Composition of music is a hard challenge that consist from many factors. It has been always about to express artists. They always wanted express themselves trought music and other kind of arts. Designing algorithms that produce humanlevel art and music, in the field of artificial intelligence, is a difficult yet rewarding challenge. Designing algorithms that produce humanlevel art and music, in the field of artificial intelligence, is a difficult yet rewarding challenge. Nowadays, human artics has more oportunity to compose music such as using methods of artificial intelligence, especially using algorithms of neural networks.[5]

On the begining , the first algorithms of neural networks has been limited by specific styles of music, such as jazz, Back chorales and also pop music. Genre-agnostic methods of composing music, such as the Biaxial LSTM [4] are able to train using a variety of musical styles, but are unable to be style-consistent in their outputs, pivoting to different styles within one composition. We have observed case study DeepJ: Style-Specific Music Generation“ by Huanru Henry Mao, Taylor Shin and Garrison Cottrell. In their case they tried introduce DeepJ as a reference to discjockeys or DJs. Their DeepJ represented a deep learning model which is capable of composing polyphonic music conditioned on a specific or a mixture of multiple composer styles. Their main contributions in their work was incorporation of enforcing output of musical style.[5]

Main different between first two methods and DeepJ is DeepJ‘s ability create polyphonic music instead of compose monophonic music (a single tune without harmony) such as CONCERT. The newes models of neural networks are able to compose music by doing simple prediction of next melodies by previous notes. One of those models of neural networks that can compose music by doing simple predicition is Long-Short Term Memory (LSTM) method.[5]

Basically the polyphonic music generation is more complex than monophonic music generation. It’s just because of that how they are processing input data, one is processing and sending by single way and second one is doing it by two flows. Every step of prediction has to consist of learned data that represents previous steps of melody, in this case we are talking about probability of any combination of notes to be played at the next time step. Early works in polyphonic composition used combination of RRNS and restricted Boltzmann machines (RBM).

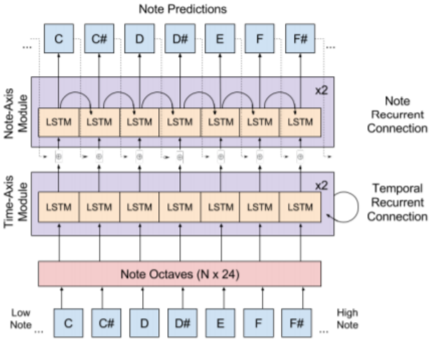
A. Biaxal LSTM

If we want to understand architecture of DeepJ at first we have to describe function and architecture of Biaxial LSTM that is used by previous studies described above. We have to notice that Biaxial LSTM is generating polyphonic music and data representation of Biaxial model uses a piano roll representations of notes. Piano roll is represented by notes that consist from binary vector. Basically we are talking about two attributes N and T that represents binary matrix where N is the number of playable notes and T represents number of time steps. If one value of this matrix represents value 1 than it means that given note has already been played by instrument. Analogically we are describing rule that define if note has been played or no, if given value represents 0 than we didn’t play given note yet.

$$t_{play} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

This matrix describes representation of four notes that each one consists from two values – value N and value T.

In this architecture exist system called replay system. Replay system represents matrix that consist of same values but it has different output, instead of doing decision-making is returning opposite value. This mechanism doing decision-making of repeating previous note without break, if play matrix plays note that consists from value 0 than replay matrix will play the same note as is represented in play matrix.



This figure describes Note predictions , that consists from three layers. As the input we have previous played notes and as the output we have notes that have been predicated by system. The time-axis section is inspired by convolution neural network, we do have also Note-Axis which consist of Long Short Term Memory (LSTM). Time-Axis input is quite similar to a convolution kernel. Kernel methods are a class of algorithms for pattern analysis, whose best known member is the support vector machine (SVM). The general task of pattern analysis is to find and study general types of relations (for example clusters, rankings, principal components, correlations, classifications) in datasets. Basically kernel function helps devide objects into few classes.

B. DeepJ method

Generally we can say that architecture of DeepJ method is just upgraded Biaxial LSTM architecture. Insted of two matrix DeepJ contains three matrixes (play,replay,dynamics). In this case play matrix and replay matrix are the same, but there is another one – added dynamics matrix. Dynamics matrix obtained values in interval ¡0,127¿ , those values have one function to define volume of tone. Dynamics in music is defined as the relative volume of a note. This matrix also consists of values N and T. Authors base their model architecture exactly on Biaxial LSTM’s design. The primary difference between those architectures is the way of use of style conditioning at every layer.

C. Training

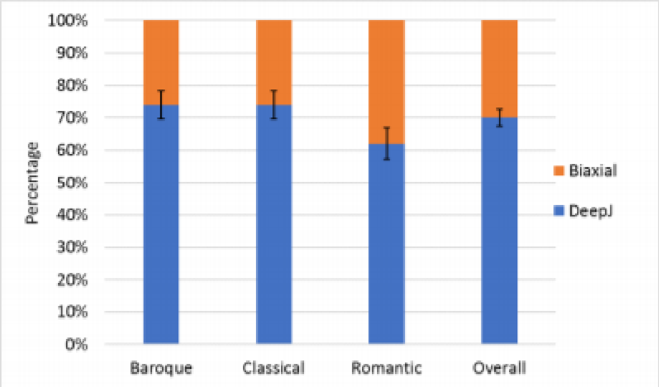
An RNN using LSTM units can be trained in a supervised fashion, on a set of training sequences, using an optimization algorithm, like gradient descent, combined with backpropagation through time to compute the gradients needed during the optimization process, in order to change each weight of the LSTM network in proportion to the derivative of the error (at the output layer of the LSTM network) with respect to corresponding weight.

D. *Activation function*

In their work they used ReLu activation function. ReLU activation function turns the value into zero immediately in the graph, which in turns affects the resulting graph by not mapping the negative values appropriately.

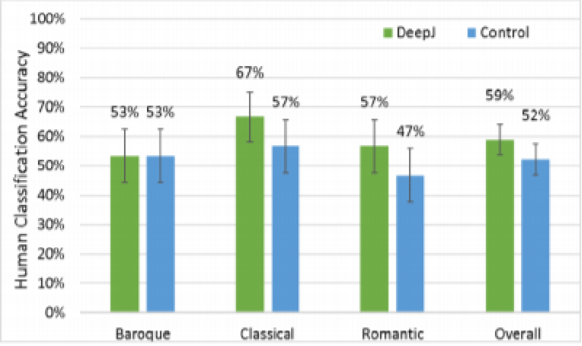
E. *Results of NN*

- Graph 1 describes that authors used sample of 300 people, where they were asking them on that which of those generated music they like more (generated by Biaxial method or generated by DeepJ method). Sample of 210 people liked music generated by DeepJ method, what is 70 percent of sample.



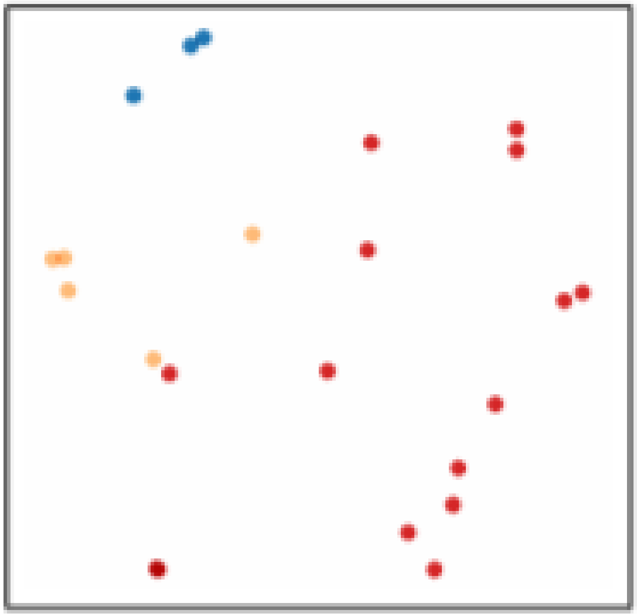
Graph 1

- Here authors compared sample of 90 people, in this case they wanted to know if used sample is able to recognize style of music they played to them. Results shows that people recognized kind of music more when music came from DeepJ method insted of control samples.



Graph 2

- Graph 3 decribes vizualisation of position each author and style of music, basically we can say picture describes visualization with perplexity of 10 and learning rate of 10 after 3000 iterations, where blue dots represents baroque, yellow dots represents classical music and red dots represents romantic style. They realized that some of dots were close to each other, reason was that those authors were close between age, that means those authors were on way to another style of music as for example Bethowen that they labeled as a classical composer (yellow dot next to the red dot) falls between the cluster centers of classical and romantic composers. Here we can see how difficult is describe each author into one single dot.



Graph 3

VI. Discusion

In our work we realized that neural networks become strong tool for artificial intelligence, they are often use in many ways to solve problems on different levels. The parallel with nature is in the way of realization of the calculation - the resulting complicated calculation is created by the interaction of many elements making the calculations easier. An understanding of the future of neural networks and their applications will help researchers to appreciate the importance and essentiality of their role in the development of a human-like artificial brain. Big purpose of using NN is in colonizing other planet in our solar system or galaxy by calculating probability of colonazition newer planet.

VII. Conclusion

Music composition is a challenging craft that has been a way for artists to express themselves ever since the dawn of civilization. Designing algorithms that produce human-level art and music, in the field of artificial intelligence, is a difficult yet rewarding challenge. Recently, advances in neural networks have helped us transition from writing music composition rules to developing probabilistic models that learn empirically-driven rules from a vast collection of existing music.

Neural network music generation algorithms have been limited to particular styles of music, such as jazz, Bach chorales, and pop music. These music generation methods have created interesting compositions, but their specialization to a particular style of music may be undesirable for practical applications. Genre-agnostic methods of composing music, such as the Biaxial LSTM [4] are able to train using a variety of musical styles, but are unable to be style-consistent in their outputs, pivoting to different styles within one composition.

References

[1] [1] Návrat, Pavol, and M. Bieliková. ”Umelá inteligencia. 1. vyd. Bratislava: Vydavateľstvo STU, 2002. 405 s.”

[2] HARMON, Leon D. Artificial neuron. Science, 1959, 129.3354: 962-963.

[3] Colombo, Florian Seeholzer, Alexander Gerstner, Wulfram. (2017). Deep Artificial Composer: A Creative Neural Network Model for Automated Melody Generation. 81-96. 10.1007/978-3-319-55750-26.

[4] Ames C. and Domino, M. (1994). Cybernetic Composer: An Overview. Technical Report, Kurzweil Foundation, Automated Composition Project.

[5] H. H. Mao, T. Shin and G. Cottrell, ”DeepJ: Style-Specific Music Generation,” 2018 IEEE 12th International Conference on Semantic Computing (ICSC), Laguna Hills, CA, 2018, pp. 377-382.