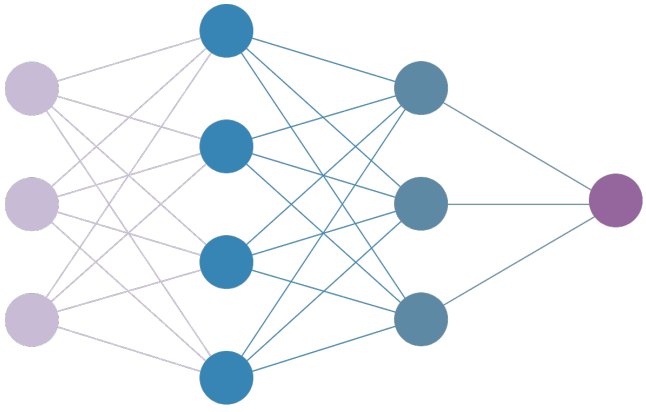
B0 - HUB

B-INN-000

A.I. Masterclass

Identify the genre of a song with Neural Networks



MASTERCLASS EXPLANATION:

The masterclass is a series of workshop / lessons that will help you to understand how to use neural networks.
The masterclass will last 8 hours and will be divided into 3 parts:

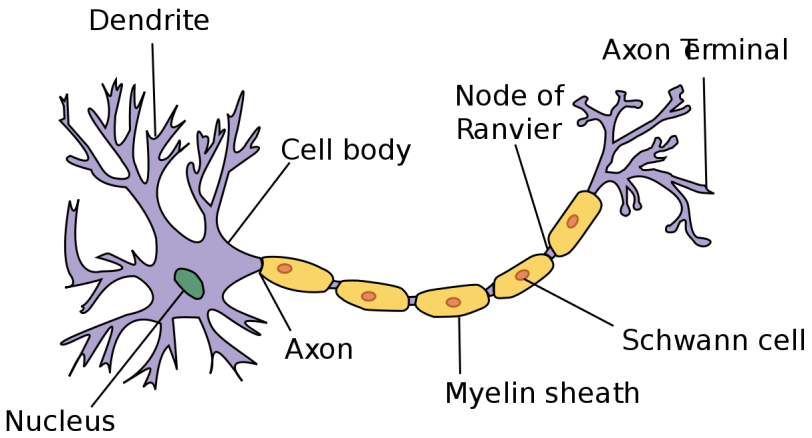
* Part 1: **Introduction to neural networks and the basics of the machine learning**
* Part 2: **Developing a neural network**
* Part 3: **Testing and Upgrading the neural network**

You need to be confortable with basic mathematics and have at least Docker installed on your computer

Part 1: Introduction to N.N.

Part 1.1: Where come from neural networks ?

The idea of neural networks began as a model of how neurons in the brain function.
The goal of neural networks was to mimic the brain. So we used the same model as the neurons in the brain.

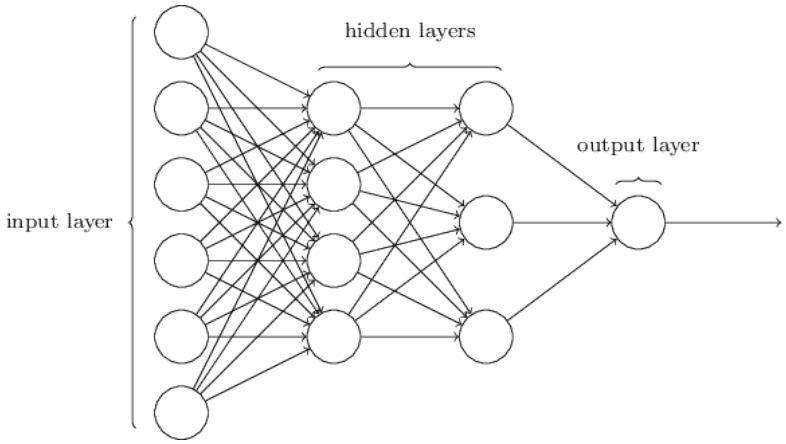


The information(s) that comes from the dendrites are the inputs.
The information(s) passes through the axon to end up in another neuron

Part 1.2: Describing the neural network

The neural network is a set of neurons that are connected to each other.
The NN has the particularity that it is a set of layers. 3 types of layers exactly:

- The input layer is the layer that contains the input data.
- The hidden layer is the layer that contains the neurons that process the input data.
- The output layer is the layer that contains the output data.

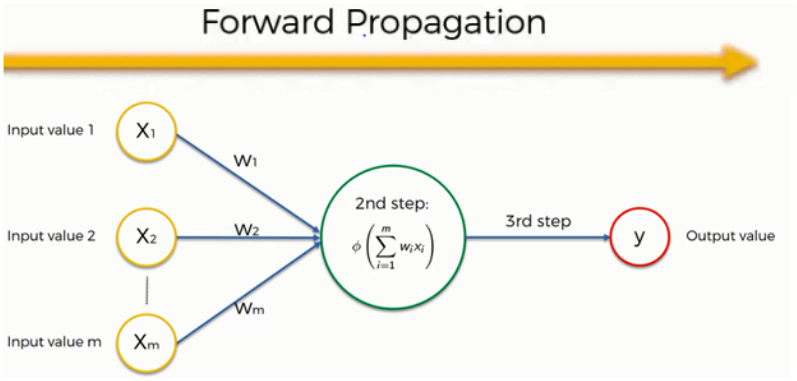


Each layer has his own functionnality and has a number of neurons.

Each neuron has his own weights and biases. We'll see how to use them in the next part.
We'll see that each neuron in the hidden layer has a function called activation function.
The NN can have several units in the output layer in the case of **multiclass classification**.
A **multiclass classification problem** is a problem that has multiple existing solutions.
We can take the example of the object recognition problem. We have a set of images of objects and we want to know which one of them is the object we are looking for. So, the possible solutions are maybe a chair, a table, a sofa, a lamp etc.

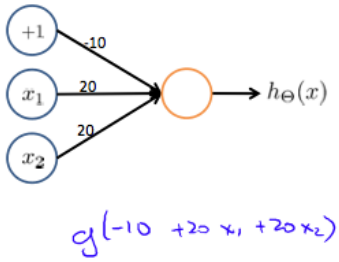
PART 1.3: HOW THE NEURAL NETWORK LEARNS ?

First, the neural network uses the input data and the weights of the hiden layers' units to compute the output data.
This technique is called the **forward propagation**.



To illustrate the forward propagation, we will use the following example:
Let's say we want to compute the OR binary function. We'll use 2 inputs units, 1 hidden layer with 1 unit and 1 output unit. We already have the weights of the hidden layer.

Example: OR function



x_1	x_2	$h_\Theta(x)$
0	0	$g(-10) \approx 0$
0	1	$g(10) \approx 1$
1	0	≈ 1
1	1	≈ 1

The activation function g will take the following form:
g(x) with g being the sigmoid function and x equal to :
x = (-1 * 10) + 20 * x1 + 20 * x2

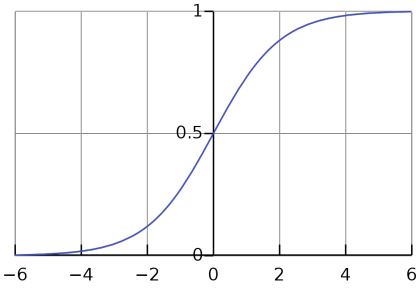
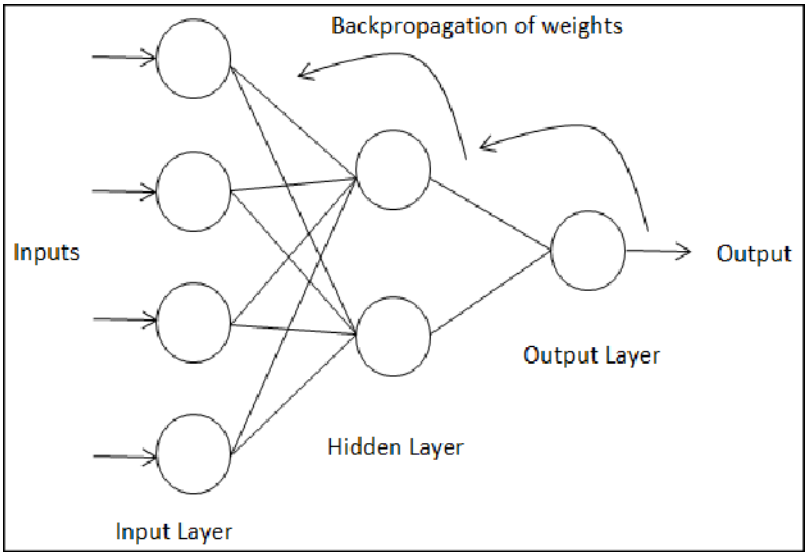


Table of calculation of the activation function:

x1	x2	g(x)
0	0	g(-10) = 0
0	1	g(10) = 1
1	0	g(10) = 1
1	1	g(30) = 1

You can try to find the weights for the AND or XOR function

Then, the neural network tries to improve the output data by changing the weights.
So, the neural network learns the relationship between the input data and the output data.
This technique is called the **back propagation**.



Behind this simple reprensentation, the Neural Network computes the following calculus:

Summary: the equations of backpropagation

$$\delta^L = \nabla_a C \odot \sigma'(z^L) \qquad \text{(BP1)}$$

$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l) \qquad \text{(BP2)}$$

$$\frac{\partial C}{\partial b^l_j} = \delta^l_j \qquad \text{(BP3)}$$

$$\frac{\partial C}{\partial w^l_{jk}} = a^{l-1}_k \delta^l_j \qquad \text{(BP4)}$$

Part 2: Developing a neural network

Configuration

For this part of the masterclass, we will develop a neural network that can identify the genre of a song.
We will use the **multiclass classification** technique.

First, we will import a database of songs and their genre (already computed).
Extract the data.zip file and put it in the root of the repository.
Now, we have database of songs and theirs genres !

Secondly, to have the perfect environment for the development of the neural network, we will use a docker container. Use the following command to start the container:

```
Terminal
~/B-INN-000>
> sudo docker run -v $PWD:/mnt -w /mnt -it miseyu/docker-ubuntu16-python3.6:latest
$> apt-get update && apt-get install -y ffmpeg
$> pip install -upgrade pip
$> python3.6 -m pip install -r requirements.txt
```

If you exit the container, you can use the following command to restart it:

```
Terminal
~/B-INN-000> docker start [container_id]
docker exec -it -w /mnt [container_id] bash
```

So, let's deep in the database. We have 1000 songs per genre and 8 genres. So, by simple calculus, we can determine that we have 8 * 1000 = 8000 songs.
But, instead of computing 1000 songs per genre (which would be too much -> take a long time), we can start by computing 5,10,50 songs per genre.

Development

Ok, now we're setup and ready to create our Neural Network.
For that, we will give you the following variables `nb_features`, `nb_hidden_layer_sizes`, `Xtrain`, `Ytrain`, `Xtest`, `Ytest`.
Now, your job is to create your own Neural Networks !

https://scikit-learn.org/stable/modules/neural_networks_supervised.html#classification

Write your code in the app.py file between commentaries.

Juan and Rafik are available to answer your questions

Part 3: Testing and Upgrading the neural network

There are a lot of things that you can / need to do to improve you Neural Network:

- Testing and plotting the differents results are very importants things to do ! Try to compute your NN with differents variables (`nb_features`, `nb_music_by_genre`, `nb_hidden_layer_sizes`, etc.) and compare the results by plotting the results with graphs
- Creating another set named Cross-Validation set which will be used to test NN when modifying variables
- You can reduce the number of features (reducing the time of computing NN) with the PCA algorithm (feel free to search by yourself)
- Plot learning curves (Cross-Validation set and Validation set) to establish if there is high bias or high variance