# Deep speech

Vincent Rébiscoul, Stéphane Pouget and Florent Guépin

**This document present our project in machine learning. We have implemented a voice recognition system i.e. our program is able to recognize spoken language and translate into text by computers. We use python3, Keras and Tensorflow.**

## Introduction

We have implemented this artcile (*1*) with python3, Keras and Tensorflow. The neuron network used is not common. So we created our own neuron network model.

## Model

We have seven layers of neuron. The three first layers are computed by :

$$h_t^{(l)} = g(W^{(l)}h_t^{(l-1} + b^{(l)}$$

where $g(z) = \min\{\max\{0, z\}, 20\}$ and $W^{(l)}, b^l$ are the weight matrix and bias parameters for layers $l$.

The fourth layer is a bi-directional reccurent layer. This layer includes two sets of hidden units : a set with forward reccurence, $h^f$, and a set with backward recurrence $h^b$ :

$$h_t^{(f)} = g(W^{(4)}h_t^{(3)} + W_r^{(f)}h_{t-1}^{(f)} + b^{(4)}$$

$$h_t^{(b)} = g(W^{(4)} h_t^{(3)} + W_r^{(b)} h_{t+1}^{(b)} + b^{(4)}$$

The fifth (non-recurrent) layer takes both the forward and backward units as inputs $h_t^{(5)} = g(W^{(5)} h_t^{(4)} + b^{(5)}$ where $h_t^{(4)} = h_t^{(f)} + h_t^{(b)}$. The output layer is a standar softmax function that yields the predicted character probabilities for each time slice $t$ and character $k$ in the alphabet :

$$h_{t,k}^{(6)} \equiv \mathbb{P}(c_t = k|x) = \frac{exp(W_k^{(6)} h_t^{(5)} + b_k^{(6)}}{\sum_j exp(W_j^{(6)} h_t^{(5)} + b_j^{(6)}}$$

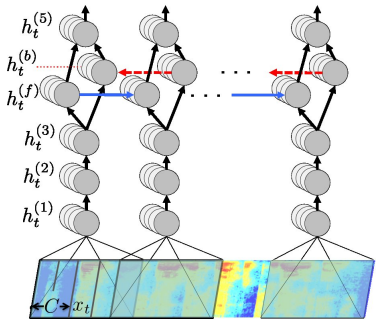


Figure 1: Structur of our RNN model

# Our work

This article (*1*) create a new model and use a ctc loss function. So to create this model, we have customized our model so that it is like on the article. For that we had to work a lot on the documentation of keras and tensorflow. However our main problem was the ctc loss function but finally everything is good.

# References and Notes

1. A. Y. Hannun, *et al.*, *CoRR* **abs/1412.5567** (2014).