FRPN: Fully Recursive Perceptron Network

Speaker: Alberto Rossi

About FRPN

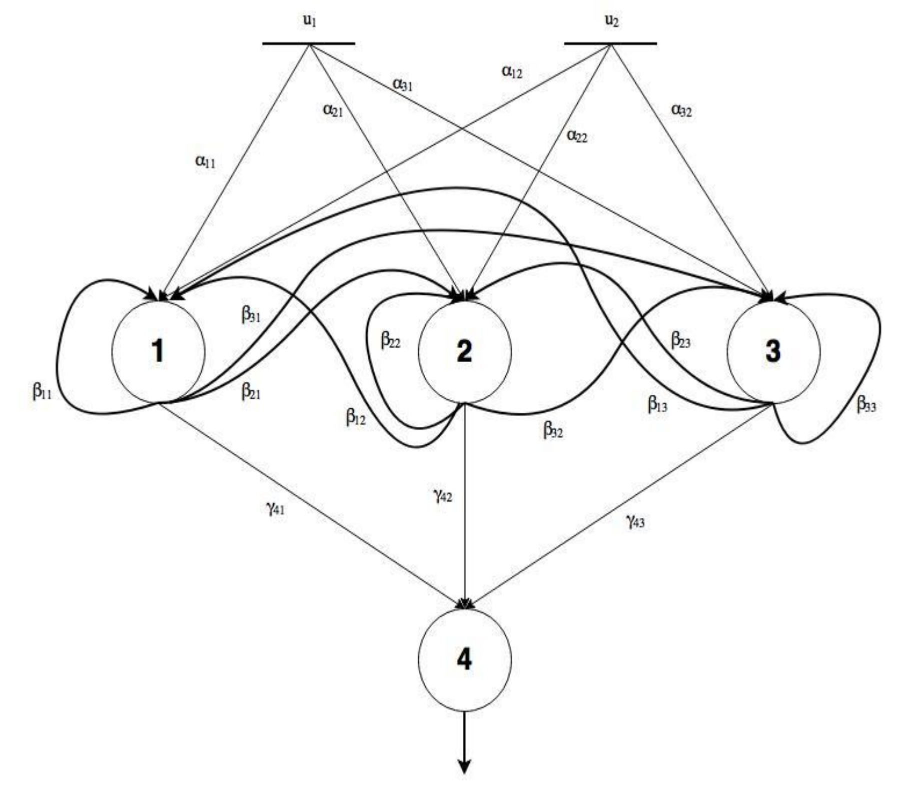
- Developed by M. Hagenbuchner, A. Tsoi, F. Scarselli and S. Zhang
- Implemented in CUDA
- It is an alternative to MLP
- Composed by only three layers: input, hidden, output
- Hidden layer neurons are interconnected, with instantaneous weighted connection
- No need to choose the number of hidden layer
- Higher computational capabilities than MLP

Fundamental Background

The graph neural network model
F. Scarselli, M. Gori, A. Tsoi, M. Hagenbuchner, G. Monfardini 2009

Generalization of backpropagation to recurrent neural network
F. Pineda1987

Architecture



- Three layers only (input, hidden, output)
- Hidden neuron *i* accept input from inputs (*u*) and all hidden neurons
- State

  $$x_i = f(\sum_{j=1}^{m} \alpha_{ij} u_j + \sum_{k=1}^{n} \beta_{ik} x_k + b_i)$$

- Output

  $$y_i = g(\sum_{j=1} \gamma_{ij} x_j + c_i)$$

- *f* non-linear activation function
- Params=n(n+m+p)+n+p

m input
n hidden neuron,
p output

Training

Backward step
Using common gradient descent algorithm

Forward step
Iteratively use state equation until neuron output converge to a stable point or for a predefined number of iteration.
Then compute the output

Properties

- When weights are small, the solution of the state equation exist and is unique thanks to Banach theorem (i.e the computed function is a contraction)
- The solution could be computed with iterative algorithm like Jacobi
- If, during learning, the weights increase we could have no solution or multiple solution to the system, and the recursion is stopped at a maximum number of iteration.

Result on MNIST

Test aim is to compare MLP and FRPN having the same number of parameter.

It could be seen how FRPN beat always MLP, meaning that can do better with the same resource.

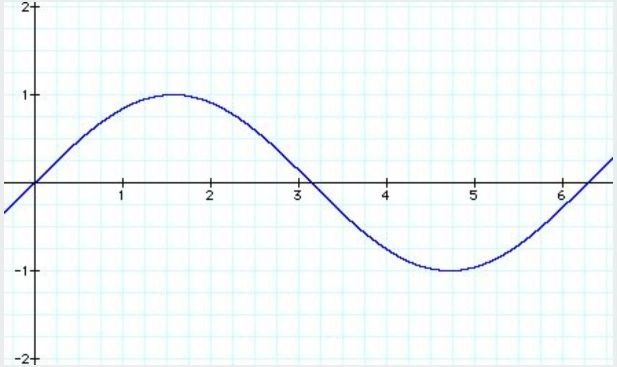
Architecture	Number of neurons in the hidden layer	Number of parameter	BTE [%]
MLP	10	8530	9.05
FRPN	10	8630	**7.38**
MLP	36:18	31168	2.52
FRPN	35	31055	**1.62**
MLP	162:81	150427	0.76
FRPN	150	150310	**0.45**
MLP	695:348	830888	0.5
FRPN	580	830570	**0.32**

Our research question:

Can FRPN learn an iterative method/function?

Sine function

Theorem: MLP with n hidden neurons could learn function with 2(n-1) minimum and maximum



Our test reveal that FRPN could learn more than 2(n-1) minimum and maximum beat again MLP.

For sine function means that FRPN with n hidden neurons could estimate sine in a range range bigger than [0,2n*pi]

Could we learn sine everywhere? How?