Projet 2

Only pytorch !

No numpy !
No sklearn !

Only math library no autograd
or neural network module

Code should work with autograd off:
torch.set_grad_enabled(False)

Autograd ?

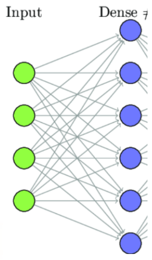
Autograd is a PyTorch package for the differentiation for all operations on Tensors.
It performs the backpropagation starting from a variable.
In deep learning, this variable often holds the value of the cost function.
backward executes the backward pass and computes all the backpropagation gradients automatically.
This signals to autograd that every operation should be tracked.

=> "Implementation" of derivative to do the backpropagation
=> Minimization of loss

Objective:

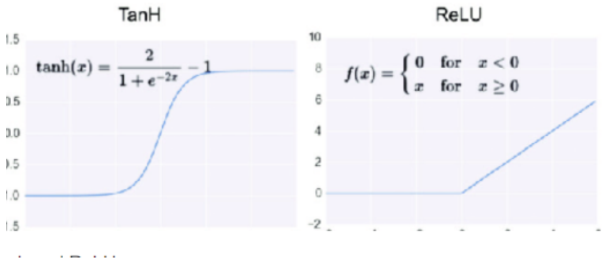
Provide tools to build **fully connected layers (FCL), tanH,ReLU**, **forward+backward pass**, optimize parameter with **SGD** or **MSE**

Remainder:



FCL: all the inputs from one layer are connected to every activation unit of the next layer.

Activation function: ReLU,tanH

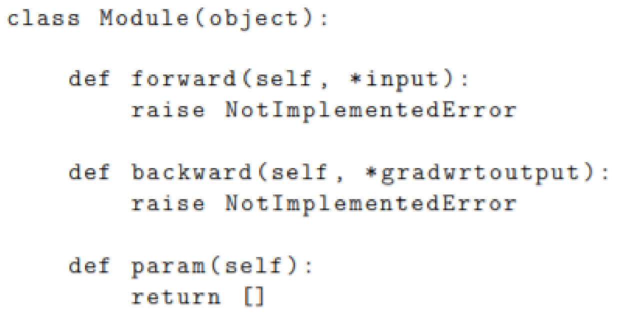


SGD: Stochastic gradient descent => update of weigths which minimize loss

MSE: the mean square error between y and f(x)

Try it by building a network with it !

Structure:



```
class Module(object):

    def forward(self, *input):
        raise NotImplementedError

    def backward(self, *gradwrtoutput):
        raise NotImplementedError

    def param(self):
        return []
```

forward, tensor -> tensor: just apply model.

- backward should get as input the gradient of the loss of last FF. Minimization + update weight ?
return new weight ?

- to define: return pair gradient/parameters ?

A big class with corresponding methods

Some modules may requires additional methods, and some modules may keep track of information from the forward pass to be used in the backward.

We should implement at least the modules Linear (fully connected layer), ReLU, Tanh, Sequential, LossMSE to compute the MSE loss

Sequantial? : Model where we have a layer by layer structure (common structure) opposite to functional: where layers connect to more than just the previous and next layers