```python
import numpy,layers,nodes

class ReLU:

    def forward(self,Z): self.Z = Z; return numpy.maximum(0,Z)
    def backward(self,DA): return DA*(self.Z>0)

class Sum(nodes.Node):

    def __init__(self,I):
        self.I = I
        for i in self.I: i.set_output(self)
        self.reset()

    def reset(self): self.o = None

    def set_output(self,O): self.O = O

    def evaluate(self):
        if self.o is None: self.o = sum(i.evaluate() for i in self.I)
        return self.o

    def grad(self): return self.O.grad()

class BranchOut(nodes.Node):

    def __init__(self,I):
        self.I = I; I.set_output(self)
        self.O = []
        self.reset()

    def reset(self): self.di = None

    def set_output(self,O): self.O += [O]

    def evaluate(self): return self.I.evaluate()

    def grad(self):
        if self.di is None: self.di = sum(o.grad() for o in self.O)
        return self.di
```