

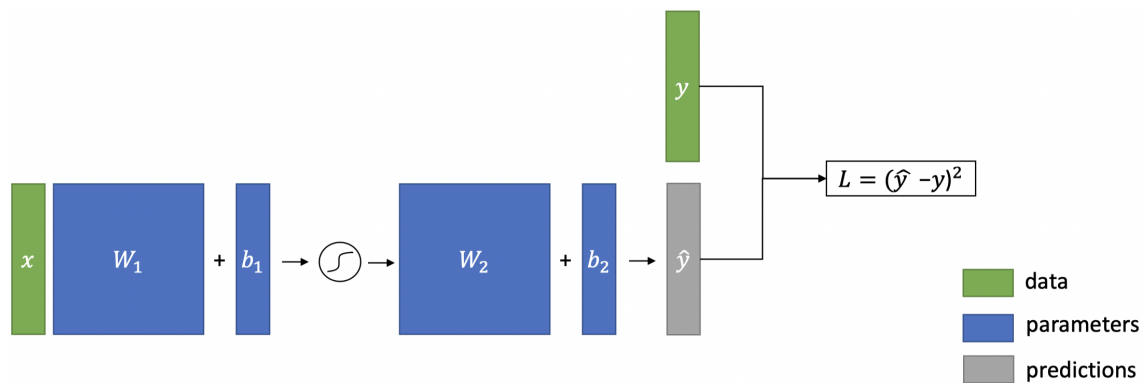
IntroML - Lecture Notes Week 8

Ruben Schenk, ruben.schenk@inf.ethz.ch

July 17, 2022

1 PyTorch Tutorial

Autodiff is an automatic implementation in PyTorch which calculates the gradients automatically given the *forward pass* (i.e. the chain of operations from input to output).



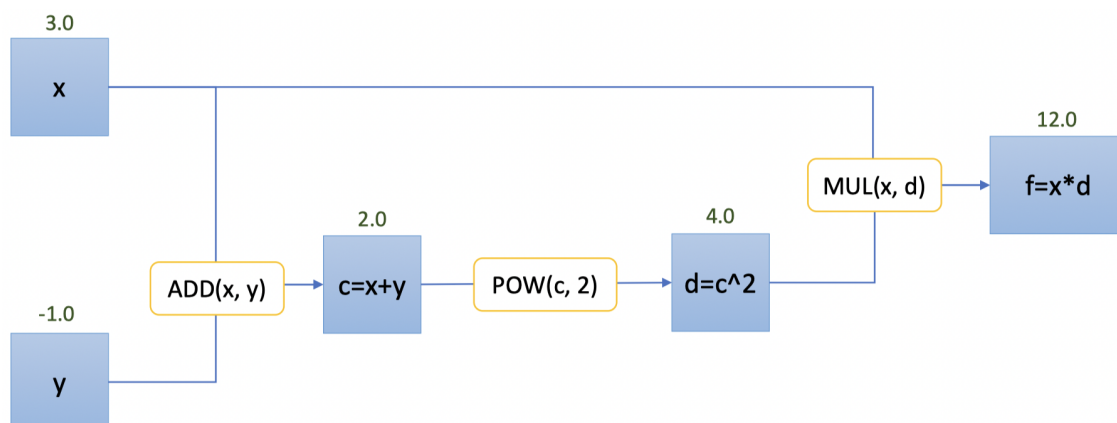
Given this forward pass, PyTorch automatically computes the derivatives (*backward pass*) for us!

How does Autodiff work? First, as already mentioned, we need to provide the forward pass. Given that, PyTorch constructs the computation graph (DAG) and stores the intermediate computed values at each node of the graph.

Example: Consider the following simple function:

```
def f(x, y):  
    c = x + y  
    d = c**2  
    return x * d  
  
x = torch.tensor([3.0], requires_grad=True)  
y = torch.tensor([-1.0], requires_grad=True)  
f_val = f(x, y)
```

For this function, we get the following forward pass:

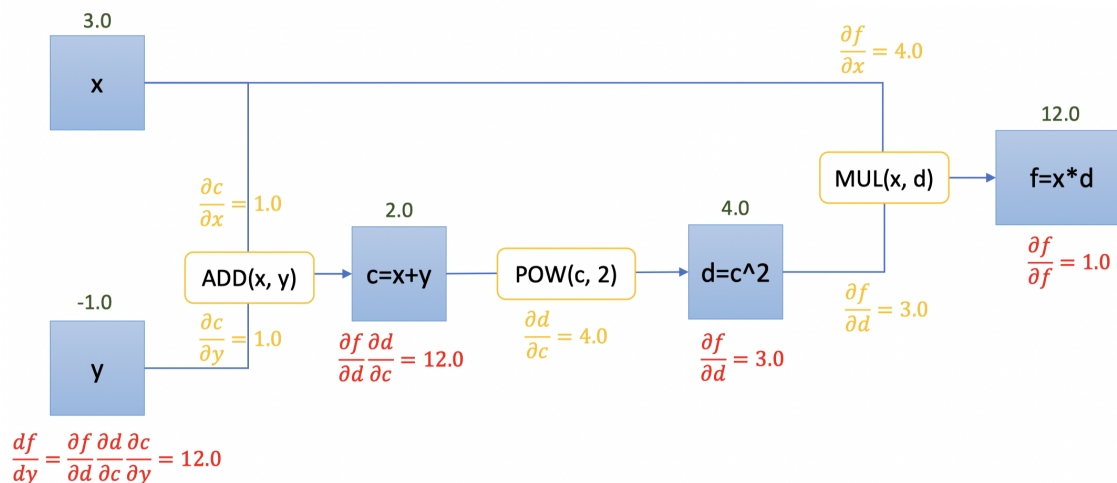


It is important to note here that every intermediate result is itself a tensor. Even if we override a variable, the previous tensor is kept alive.

Finally, Autodiff calculates the backward pass the following way:

- Go back from the root (i.e. the loss) to the leaves (i.e. the parameters)
- Each elementary operation has a "backward function" that computes its derivatives
- We apply the chain rule along each path
- Total derivative is the sum of all path derivatives from the root to the leaf

$$\frac{df}{dx} = \frac{\partial f}{\partial d} \frac{\partial d}{\partial c} \frac{\partial c}{\partial x} + \frac{\partial f}{\partial x} = 16.0$$



2 Unsupervised Learning: Clustering