Implicit Layers

Complex Functions as Layers

Layer 概念在深度学习中被滥用，因此需要区分两个概念：

- Operator: 计算图中的“原子”操作，以及它的显式梯度传递。即矩阵乘法、卷积、elementwise 的非线性函数等操作。
- Module: Operator 构成的集合，它的 backward pass 是根据计算图的构建隐式形成的，不需要显式定义。

将卷积等操作设计为原子算符的优势在于：

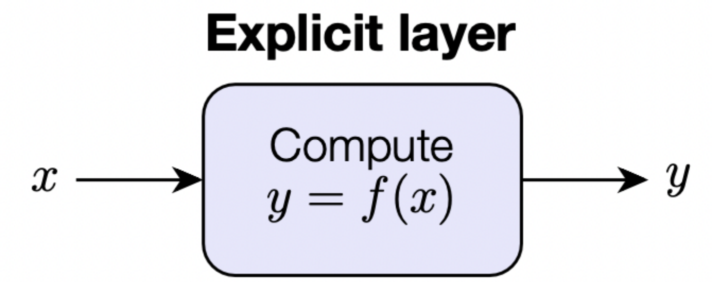
- 可以更为模块化地优化 forward pass
- 不需要在内存中存储所有的中间结果，节省内存资源。

如果想要在网络中定义一个执行更复杂操作的层，例如解决一个优化问题，求不动点，解 ODE 等操作，可以直接实现为一个 module, 从而直接植入到更大的计算图中。

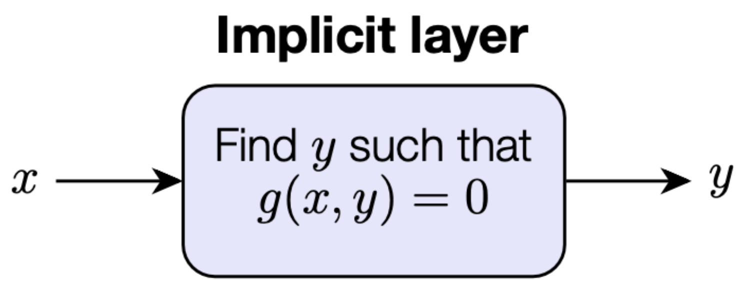
但是，将这些函数实现为 operator 的优势更大。

Explicit vs. Implicit Layers

事实上，到目前为止的所有层都是显式的，有从输入到输出的固定转换过程。

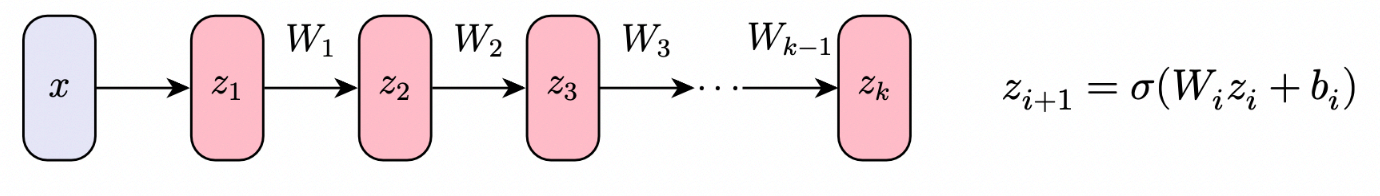


隐式层定义了更为复杂的操作，要求输入和输出满足某些联合条件。例如微分方程、不动点迭代、优化解等。

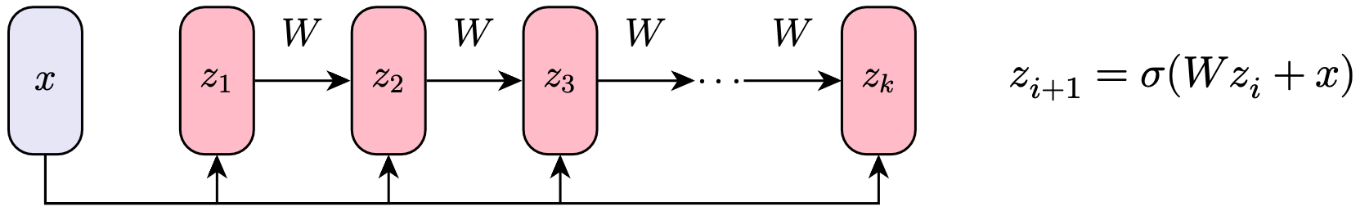


Example: Deep Equilibrium Models

考虑一个传统 MLP 模型：

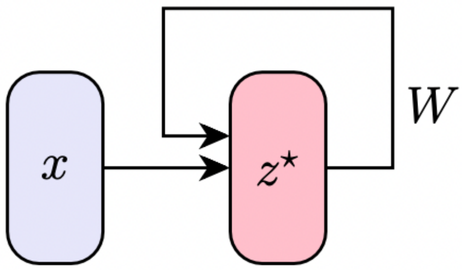


对该模型进行修改，在每一步中重新输入 x, 并且将统一权重参数矩阵为 W



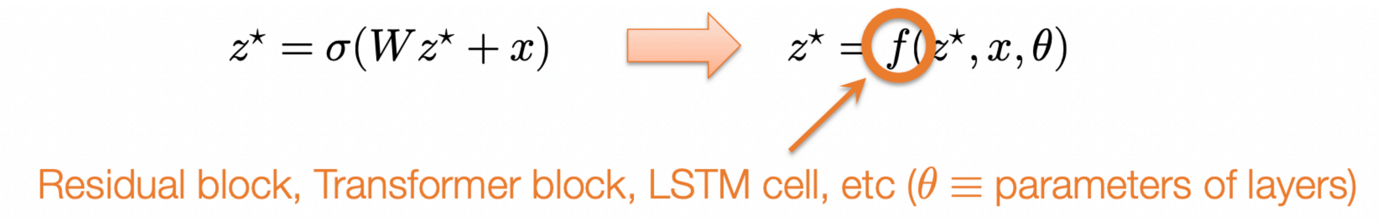
那么每次调用的函数都是相同的，于是可以设计网络使得最终迭代到不动点 (fixed/equilibrium point)

$$z^* = \sigma(W z^* + x) \tag{2}$$



因此可以定义一个 layer 使得能直接找到不动点（通过迭代或其他方法）

在实践中，我们希望找到一个更复杂的“单元”的平衡点，并将其用作我们的整个模型，即



如果将这个操作实现为一个 operator, 那么就不需要关心如何计算不动点，也不需要存储中间状态。

但是要实现为 operator 就必须实现相应的微分算法，即梯度计算部分。

Differentiation of Implicit Layers

定义一个求不动点的 implicit layer:

$$z^* = f(z^*, x) \tag{3}$$

其中 f 为非线性函数，求得的解为 $z^*(x)$

梯度计算如下：

$$
\begin{aligned}
&z^\star(x) = f(z^\star(x), x) \\
&\frac{\partial z^\star(x)}{\partial x} = \frac{\partial f(z^\star(x), x)}{\partial x} \\
&\frac{\partial z^\star(x)}{\partial x} = \frac{\partial f(z^\star(x), x)}{\partial z^\star(x)} \frac{\partial z^\star(x)}{\partial x} + \frac{\partial f(z^\star(x), x)}{\partial x} \\
&\frac{\partial z^\star(x)}{\partial x} = \left(I - \frac{\partial f(z^\star(x), x)}{\partial z^\star(x)}\right)^{-1} \frac{\partial f(z^\star(x), x)}{\partial x}
\end{aligned}
$$

在自微分框架中，我们需要计算 adjoint 与 layer gradient 的乘积：

$$
\bar{v}\frac{\partial z^\star(x)}{\partial x} = \underbrace{\bar{v}\left(I - \frac{\partial f(z^\star(x), x)}{\partial z^\star(x)}\right)^{-1}}_{\text{“Special” adjoint computation for implicit layer}} \frac{\partial f(z^\star(x), x)}{\partial x} = \underbrace{\bar{v}' \frac{\partial f(z^\star(x), x)}{\partial x}}_{\text{“Normal” AD adjoint}}
$$

也就是说，为了实现 implicit layer 的反向传播，只需要将这个特殊的 adjoint 计算加入 backward pass 中。

细节上，

$$
\begin{aligned}
\bar{v}' &= \bar{v}(I - \frac{\partial f(z^*(x), x)}{\partial z^*(x)})^{-1} \\
\bar{v}' &= \bar{v} + \bar{v}' \frac{\partial f(z^*(x), x)}{\partial z^*(x)}
\end{aligned}
\tag{4}
$$

实际上类似于解不动点问题：

$$
z_{i+1} = \bar{v} + z_i \frac{\partial f(z^*(x), x)}{\partial z^*(x)} \tag{5}
$$

也就是说，不动点算符的 backward pass 可以实现为另一个含有 AD adjoint 的不动点算符。