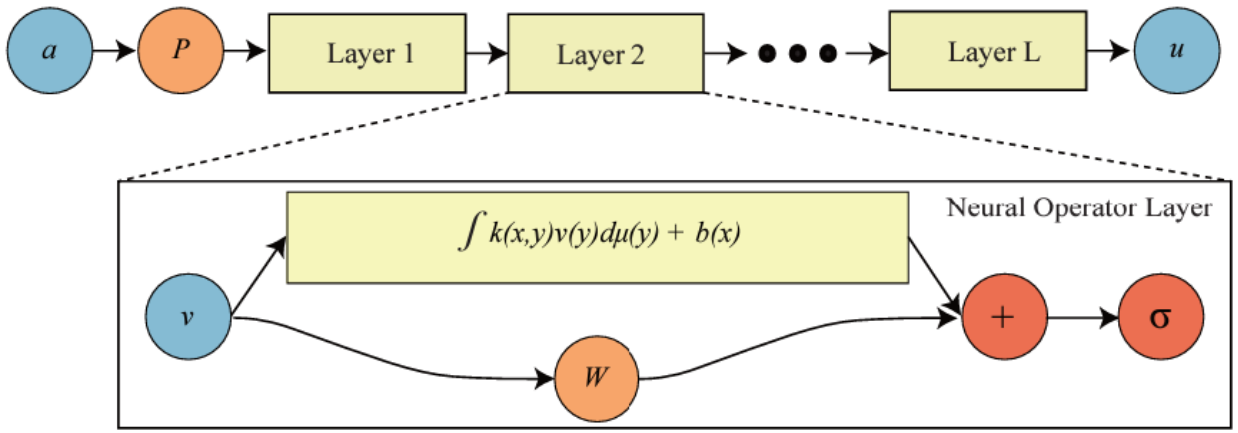
Neural Operator Network

Neural Operator

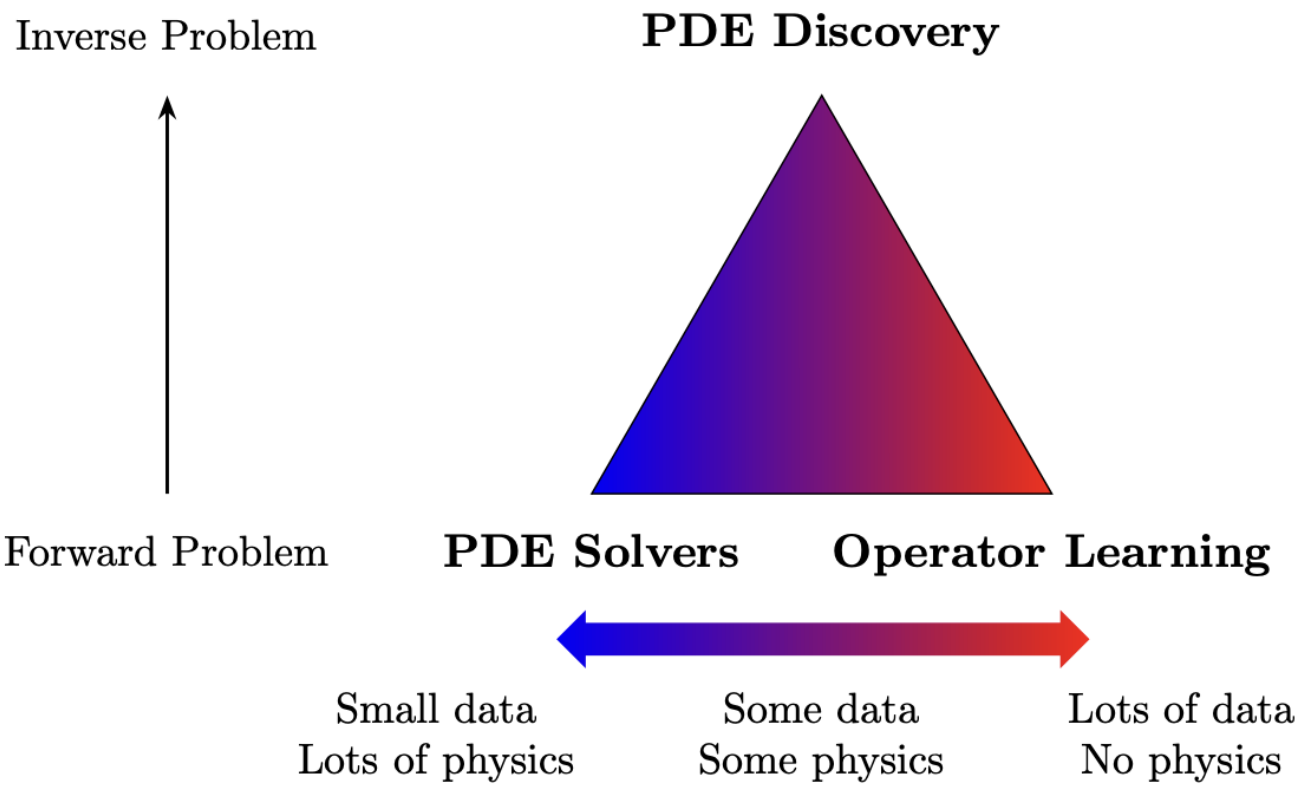
- **Neural operators** are a class of neural networks with infinite-dimensional inputs
- Neural operators generalize standard deep learning techniques to learn mappings between function spaces instead of between discrete vector spaces



Neural Operator

- Speeds up numerical PDE solvers: operator learning to build reduced-order models of complex systems
- Parameter optimization: mesh-free and parameterized by a neural network that can be evaluated at any point
- Benchmarking new techniques: preserve quantities in PDEs as symmetries, conservation laws, discretization independence
- Discovering unknown physics: solution operator of a PDE is often unknown

Operator Learning

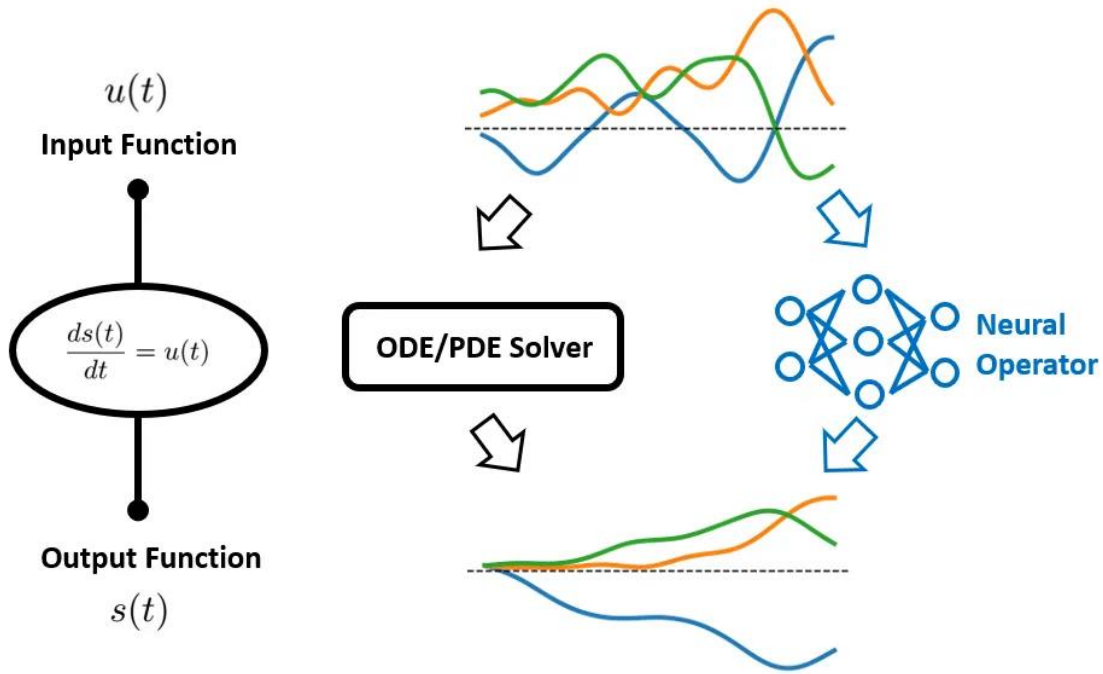


神經運算子網路（Neural Operator Network）是一種用於學習泛函映射的深度學習模型。與傳統神經網路不同，神經運算子網路專注於處理函數空間中的映射問題，例如偏微分方程（PDE）的解、參數化模型的逼近等。這類模型的主要特點是能夠高效地學習輸入函數到輸出函數之間的關係。

1. 神經運算子網路的目標

神經運算子網路的目標是學習一個運算子 $\mathcal{G}: \mathcal{X} \to \mathcal{Y}$，其中 $\mathcal{X}$ 和 $\mathcal{Y}$ 分別為輸入和輸出的函數空間。例如，給定某偏微分方程的初始條件 $u_0(x)$，學習解的運算子 $\mathcal{G}$ 使得：

其中 $u(x)$ 是 PDE 的解。



2. 核心架構

神經運算子網路的架構通常包含以下步驟：

(a) 升維層（Lifting Layer）

將輸入函數 $u_0(x)$ 嵌入到高維特徵空間中：

$$v(x) = P(u_0(x)),$$

其中 P 是升維運算（例如全連接層）。

(b) 運算子層（Operator Layers）

應用核積運算來建模全局依賴性。以傅立葉神經運算子（FNO）為例，使用傅立葉變換 F 提取全局特徵：

$$v'(k) = \sigma\left(W(k)F(v)(k)\right),$$

然後逆傅立葉變換回輸出空間。

(c) 降維層（Projection Layer）

將高維特徵空間映射回輸出函數空間：

$$u(x) = Q(v(x)),$$

其中 Q 是降維運算（例如全連接層）。

3. 數學形式化

神經運算子網路可以被表示為：

$$\mathcal{G} = Q \circ \mathcal{K} \circ P,$$

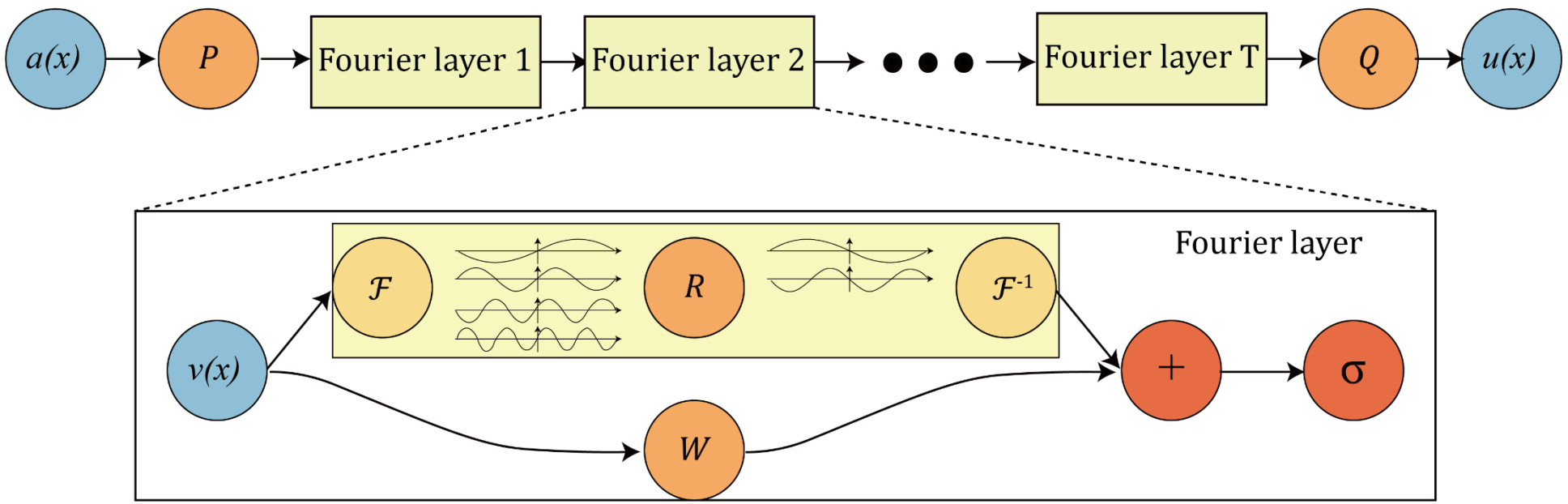
- P: 升維運算子。
- $\mathcal{K}$: 運算子層（核運算的堆疊）。
- Q: 降維運算子。

對於傅立葉神經運算子（FNO），具體的運算子層為：

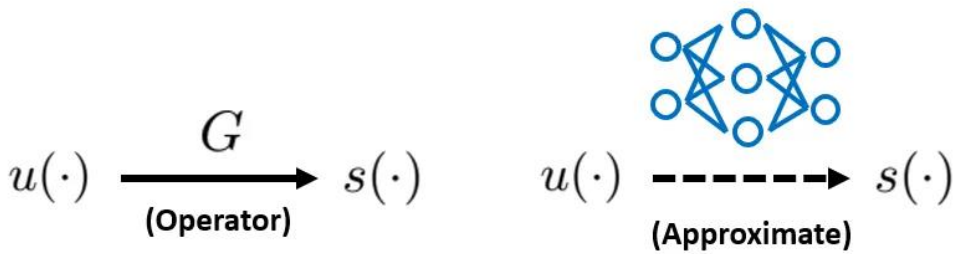
$$\mathcal{K}(v)(x) = \mathcal{F}^{-1}\left(W \cdot \mathcal{F}(v)\right)(x) + W_{\text{local}}(v)(x),$$

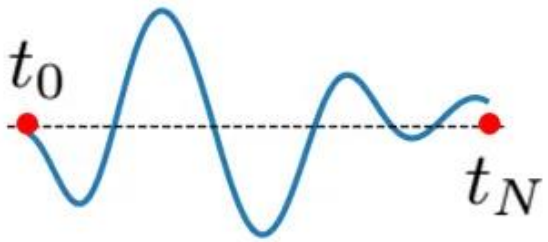
其中 $\mathcal{F}$ 和 $\mathcal{F}^{-1}$ 分別為傅立葉變換與逆變換，W 為學習的權重。

- $\mathcal{F}$ 和 $\mathcal{F}^{-1}$：傅立葉變換與逆變換。
- W：頻域中的學習參數，用於操作高頻與低頻特徵。

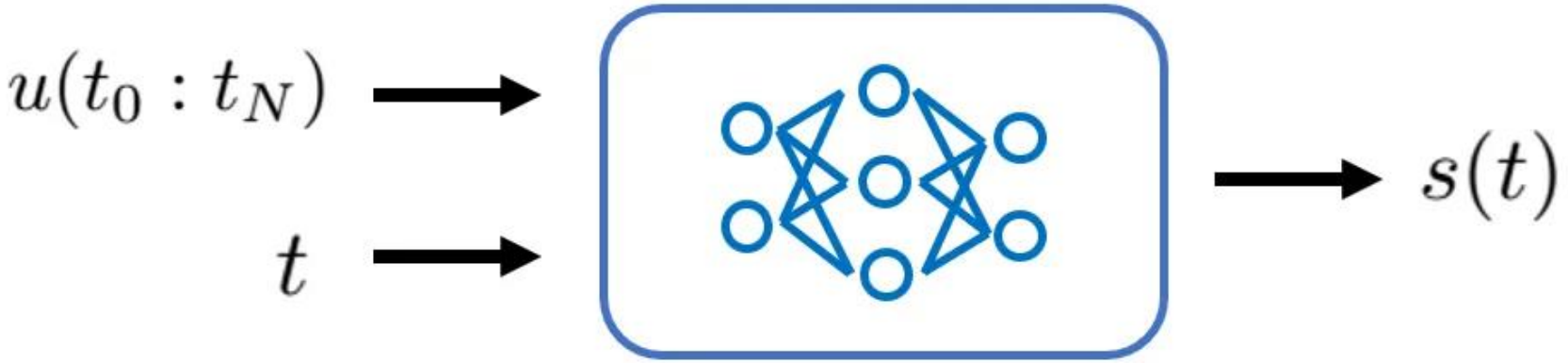


Neural Operator Network

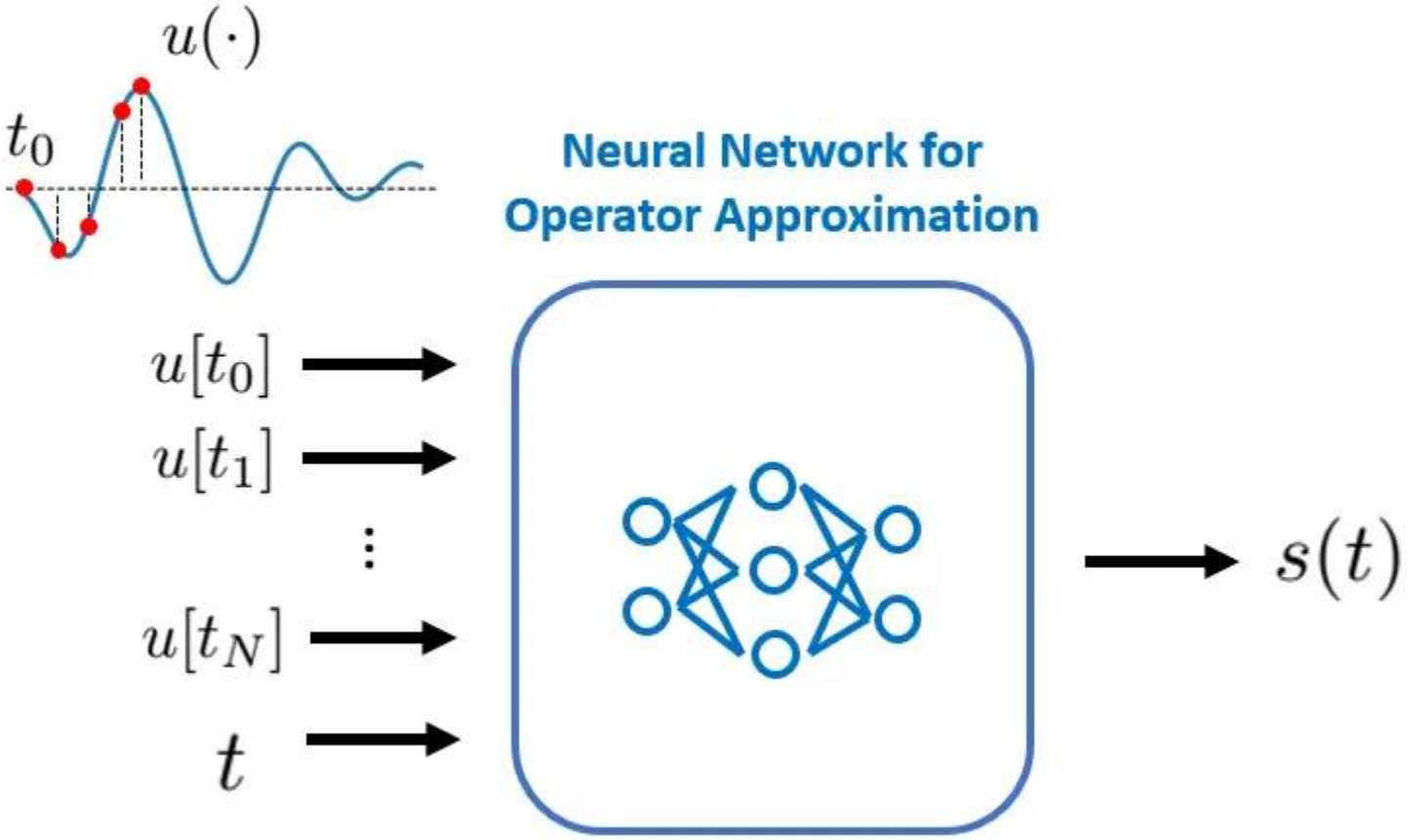




Neural Network for Operator Approximation



Neural Operator Network



深度運算子網路（Deep Operator Network，DeepONet）是一種神經網路架構，設計用於學習函數之間的運算子映射。與傳統深度學習模型不同，DeepONet 不僅可以處理輸入和輸出在歐幾里得空間中的關係，還可以學習函數空間中的映射，使其非常適合解決偏微分方程（PDE）等涉及泛函映射的問題。

1. 核心概念

DeepONet 的目標是學習一個運算子 $\mathcal{G}$，其將輸入函數 $u(x)$ 映射到輸出函數 $v(x)$，即：

$$\mathcal{G}(u)(y) = v(y),$$

其中 u 是輸入函數，v 是輸出函數，y 是輸出函數的變量。

2. 架構設計

DeepONet 的核心架構基於「分支網路（Branch Network）」和「樹幹網路（Trunk Network）」的結合：

(a) 分支網路（Branch Network）

分支網路用於處理輸入函數 $u(x)$，提取函數空間中的特徵。具體步驟如下：

1. $u(x)$ 通過一組離散點 $\{u(x_1), u(x_2), \ldots, u(x_m)\}$ 表示。
2. 分支網路接受這些離散值作為輸入，並輸出一個特徵向量 $b = [b_1, b_2, \ldots, b_p]$。

(b) 樹幹網路（Trunk Network）

樹幹網路用於處理輸出變量 y，並生成與 y 相關的特徵表示：

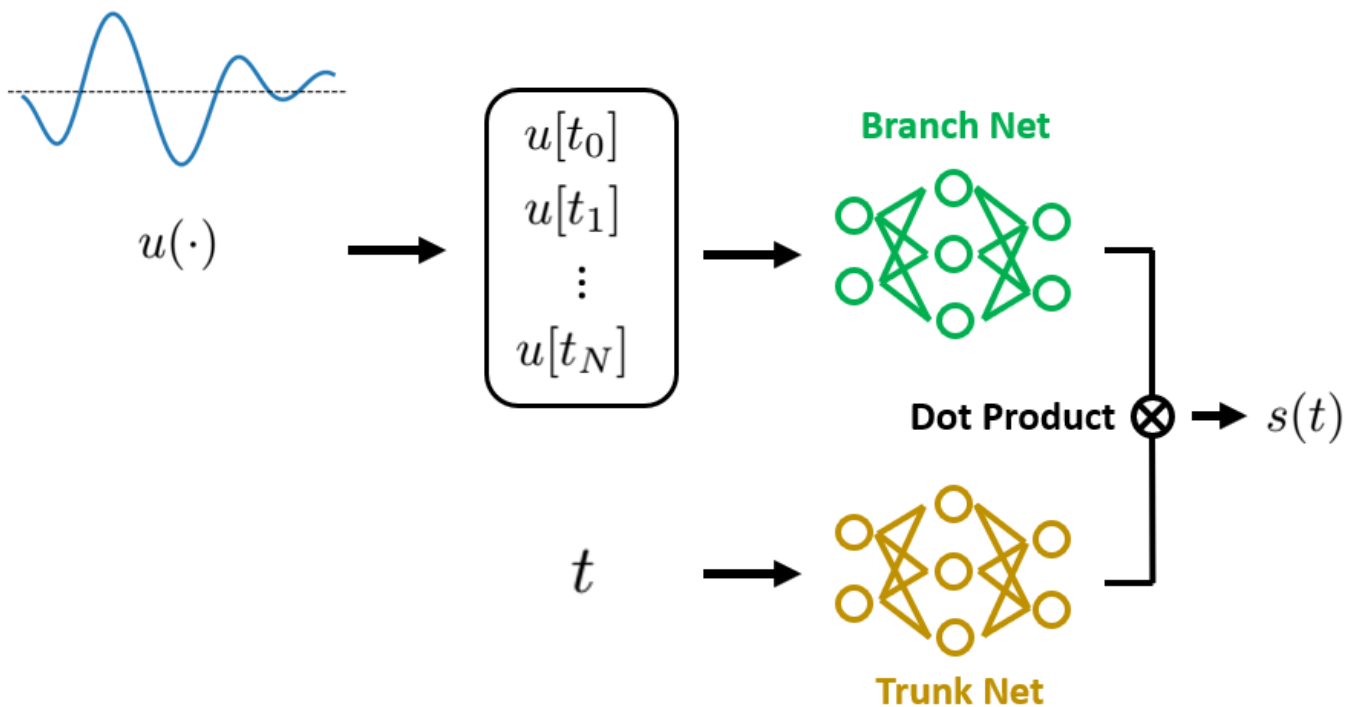
1. y 作為輸入，經過多層神經網路生成特徵向量 $t = [t_1, t_2, \ldots, t_p]$。

(c) 結合層

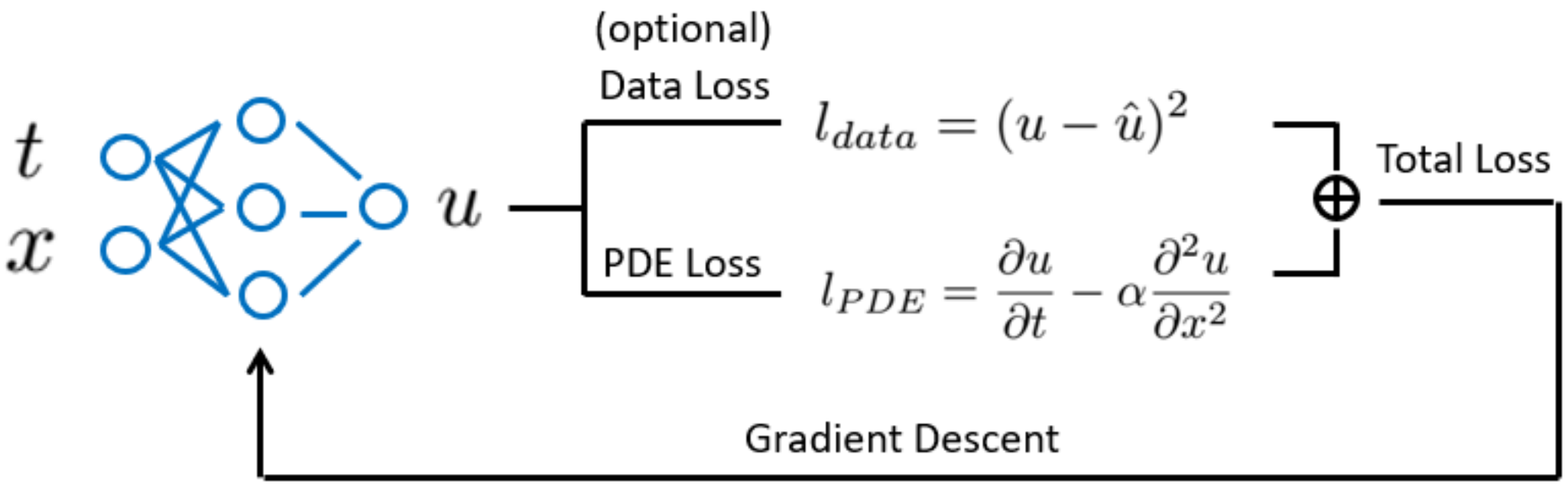
分支網路與樹幹網路的輸出通過內積結合，生成最終的運算子輸出：

$$\mathcal{G}(u)(y) = \sum_{i=1}^{p} b_i t_i,$$

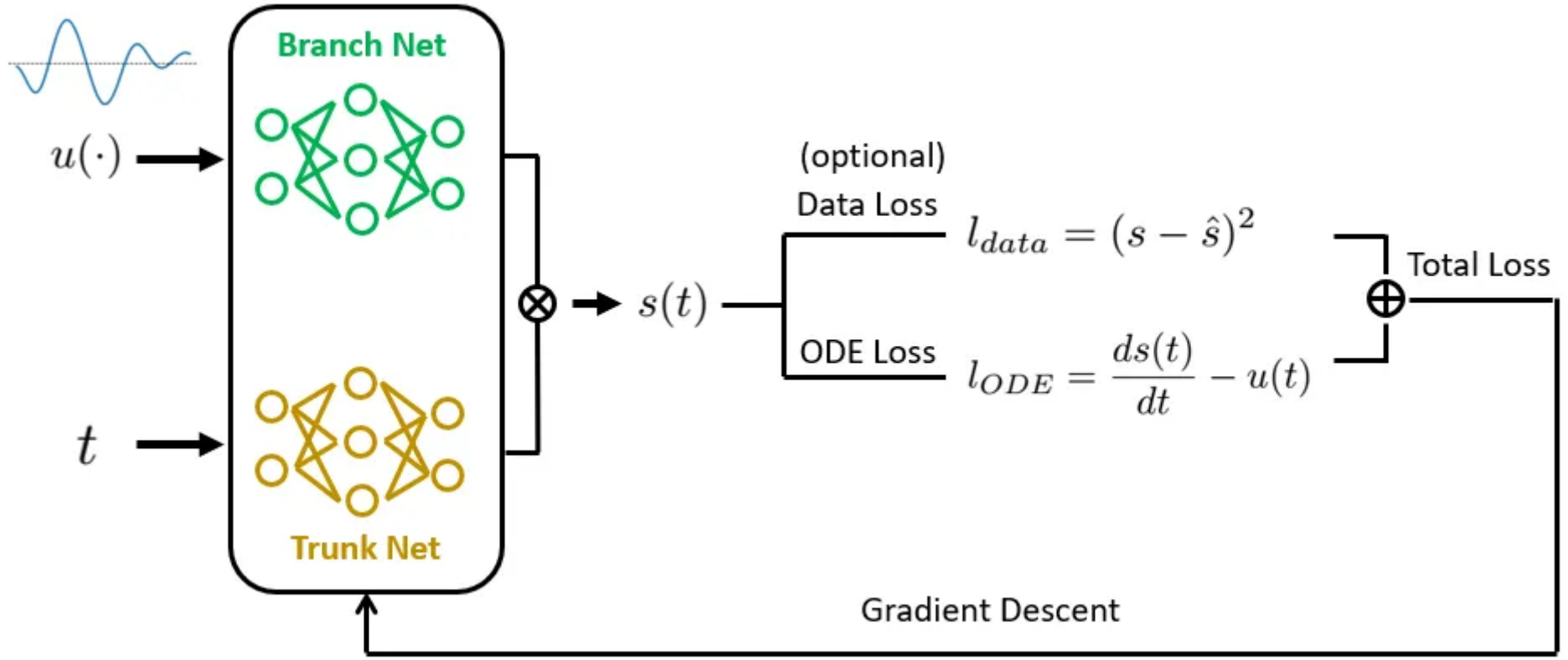
其中 b_i 來自分支網路，t_i 來自樹幹網路。



Physics-Informed Neural Network



Physics-Informed DeepONet



Physics-Informed DeepONet (PI-DeepONet) 是結合物理知識與深度運算子網路（DeepONet）的強大工具，用於解決受物理規律約束的問題，特別是涉及偏微分方程（PDE）或科學計算的應用。PI-DeepONet 的核心思想是在學習運算子的過程中顯式地融入物理約束，從而提高模型的準確性與物理一致性。

1. 核心概念

PI-DeepONet 在原始 DeepONet 的基礎上，透過物理信息（如 PDE 或邊界條件）的損失函數對網路進行監督，從而學習符合物理規律的運算子映射。其目標為：

$$\mathcal{G}(u)(y) \approx v(y),$$

同時確保滿足物理約束（如 PDE 方程和邊界條件）。

2. 架構設計

PI-DeepONet 的基本架構仍包括分支網路與樹幹網路，但在訓練過程中融入了物理知識：

(a) 分支網路與樹幹網路

- **分支網路（Branch Network）**：處理輸入函數 $u(x)$ 的離散表示，輸出特徵 $b_i(u)$。
- **樹幹網路（Trunk Network）**：處理輸出位置 y 的信息，輸出特徵 $t_i(y)$。
- **結合**：通過內積計算運算子輸出：

$$\mathcal{G}(u)(y) = \sum_{i=1}^{p} b_i(u) t_i(y).$$

(b) 物理信息嵌入

物理約束以顯式方式嵌入損失函數中，包括：

1. **PDE 損失**：確保運算子輸出的函數 $v(y)$ 滿足 PDE，例如：

$$\mathcal{L}[v(y)] - f(y) = 0,$$

   其中 $\mathcal{L}$ 是 PDE 的運算子，$f(y)$ 是源項。

2. **邊界條件損失**：確保 $v(y)$ 滿足給定的邊界條件，例如：

$$v(y) = g(y), \quad \text{當 } y \in \partial\Omega.$$

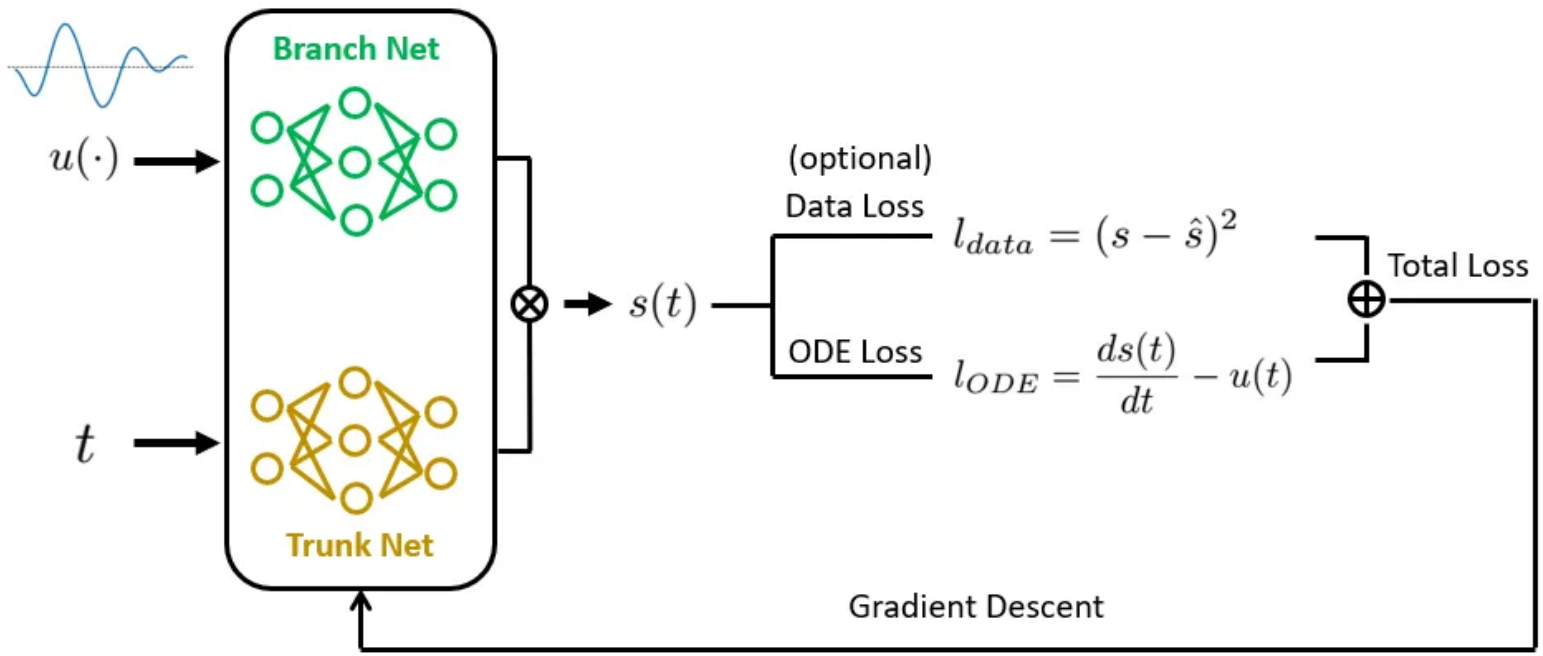
3. **數據驅動損失（可選）**：若有數據樣本，還可包含數據與模型輸出之間的誤差。

(c) 總損失函數

總損失函數為上述部分的加權和：

$$\mathcal{L}_{\text{total}} = \lambda_{\text{data}}\mathcal{L}_{\text{data}} + \lambda_{\text{PDE}}\mathcal{L}_{\text{PDE}} + \lambda_{\text{BC}}\mathcal{L}_{\text{BC}},$$

其中 $\lambda_{\text{data}}, \lambda_{\text{PDE}}, \lambda_{\text{BC}}$ 是權重超參數。



PyTorch