Neural Operator

Liyao Lyu

Problem Setting

Problem Setting

Our goal is to learn a map between two infinite dimensional spaces.

Problem Setting

Our goal is to learn a map between two infinite dimensional spaces.

$$\mathcal{G}_\theta : \mathcal{A} \to \mathcal{U}, \theta \in \mathbb{R}^p$$

Problem Setting

Our goal is to learn a map between two infinite dimensional spaces.

$$\mathcal{G}_\theta : \mathcal{A} \to \mathcal{U}, \theta \in \mathbb{R}^p$$

Here $\mathcal{A}$ and $\mathcal{U}$ are two function space.

Problem Setting

Our goal is to learn a map between two infinite dimensional spaces.

$$\mathcal{G}_\theta : \mathcal{A} \rightarrow \mathcal{U}, \theta \in \mathbb{R}^p$$

Here $\mathcal{A}$ and $\mathcal{U}$ are two function space.

For Equation $\mathcal{L}u = f \quad u = 0$ on $\partial\Omega$, we want to find $\mathcal{G} = \mathcal{L}^{-1}$ such that $u = \mathcal{G}[f]$, which directly solve the PDE.

Problem Setting

Our goal is to learn a map between two infinite dimensional spaces.

$$\mathcal{G}_\theta : \mathcal{A} \to \mathcal{U}, \theta \in \mathbb{R}^p$$

Here $\mathcal{A}$ and $\mathcal{U}$ are two function space.

For Equation $\mathcal{L}u = f \quad u = 0$ on $\partial\Omega$, we want to find $\mathcal{G} = \mathcal{L}^{-1}$ such that $u = \mathcal{G}[f]$, which directly solve the PDE.

Problem Setting

Example

Problem Setting

Example

Consider an ODE first:

Problem Setting

Example

Consider an ODE first:

$$\frac{du}{dt} = f(t), \quad u(0) = u_0$$

Problem Setting

Example

Consider an ODE first:

$$\frac{du}{dt} = f(t), \quad u(0) = u_0$$

Given $\{f_j, u_j\}_{j=1}^{N}$ pairs of functions, we want the learn the inverse operator.

Problem Setting

Example

Consider an ODE first:

$$\frac{du}{dt} = f(t), \quad u(0) = u_0$$

Given $\{f_j, u_j\}_{j=1}^{N}$ pairs of functions, we want the learn the inverse operator.

the inverse operator $u(t) = \mathcal{G}[f](t) = u_0 + \int_0^t f(\tau)d\tau$

Problem Setting

Example

Consider an ODE first:

$$\frac{du}{dt} = f(t), \quad u(0) = u_0$$

Given $\{f_j, u_j\}_{j=1}^{N}$ pairs of functions, we want the learn the inverse operator.

the inverse operator $u(t) = \mathcal{G}[f](t) = u_0 + \int_0^t f(\tau)d\tau$

Problem Setting

Example

Problem Setting

Example

Consider a second order elliptic equation:

Problem Setting

Example

Consider a second order elliptic equation:

$$
\begin{aligned}
&-\nabla(a(x)\nabla u(x)) = f(x) && x \in \Omega \\
&u(x) = 0 && x \in \partial\Omega
\end{aligned}
$$

Problem Setting

Example

Consider a second order elliptic equation:

$$
\begin{aligned}
-\nabla(a(x)\nabla u(x)) &= f(x) \qquad && x \in \Omega \\
u(x) &= 0 && x \in \partial\Omega
\end{aligned}
$$

Given $\{a_j, u_j\}$ pairs of functions, want to learn the inverse operator.

Problem Setting

Example

Consider a second order elliptic equation:

$$
\begin{aligned}
-\nabla(a(x)\nabla u(x)) &= f(x) && x \in \Omega \\
u(x) &= 0 && x \in \partial\Omega
\end{aligned}
$$

Given $\{a_j, u_j\}$ pairs of functions, want to learn the inverse operator.

the inverse operator $u(x) = \mathcal{G}[a](x) = \int_{\Omega} G_a(x, y) f(y) dy$, where G_a is the green's function

Problem Setting

Example

Consider a second order elliptic equation:

$$
\begin{aligned}
-\nabla(a(x)\nabla u(x)) &= f(x) \qquad && x \in \Omega \\
u(x) &= 0 && x \in \partial\Omega
\end{aligned}
$$

Given $\{a_j, u_j\}$ pairs of functions, want to learn the inverse operator.

the inverse operator $u(x) = \mathcal{G}[a](x) = \int_\Omega G_a(x,y)f(y)dy$, where G_a is the green's function

Deep ONet

Network Structure

Deep ONet

Network Structure

$$\mathcal{G}[a](y) = \sum_{k=1}^{p} \sum_{i=1}^{n} c_i^k \sigma(\sum_{j=1}^{m} w_{ij}^k u(x_j) + b_i^k) \sigma(\omega_k y + \beta)$$

Deep ONet

Network Structure

$$\mathcal{G}[a](y) = \sum_{k=1}^{p} \sum_{i=1}^{n} c_i^k \sigma(\sum_{j=1}^{m} w_{ij}^k u(x_j) + b_i^k) \sigma(\omega_k y + \beta)$$

where $j = 0, \cdots, m$, and x_j are the uniform mesh points on the space, for example $x_j = \frac{j}{m} \in [0,1]$ in one dimensional case.

Deep ONet

Network Structure

$$\mathcal{G}[a](y) = \sum_{k=1}^{p} \sum_{i=1}^{n} c_i^k \sigma(\sum_{j=1}^{m} w_{ij}^k u(x_j) + b_i^k) \sigma(\omega_k y + \beta)$$

where $j = 0, \cdots, m$, and x_j are the uniform mesh points on the space, for example $x_j = \dfrac{j}{m} \in [0,1]$ in one dimensional case.

Banch block: $b_k(u(x_1), \cdots, u(x_m)) = \sum_{i=1}^{n} c_i^k \sigma(\sum_{j=1}^{m} w_{ij}^k u(x_j) + b_i^k)$

Deep ONet

Network Structure

$$\mathcal{G}[a](y) = \sum_{k=1}^{p} \sum_{i=1}^{n} c_i^k \sigma(\sum_{j=1}^{m} w_{ij}^k u(x_j) + b_i^k) \sigma(\omega_k y + \beta)$$

where $j = 0, \cdots, m$, and x_j are the uniform mesh points on the space, for example $x_j = \dfrac{j}{m} \in [0,1]$ in one dimensional case.

Banch block: $b_k(u(x_1), \cdots, u(x_m)) = \sum_{i=1}^{n} c_i^k \sigma(\sum_{j=1}^{m} w_{ij}^k u(x_j) + b_i^k)$

Trunk block: $t_k(y) = \sigma(\omega_k y + \beta_k)$

Deep ONet

Network Structure

$$\mathcal{G}[a](y) = \sum_{k=1}^{p} \sum_{i=1}^{n} c_i^k \sigma(\sum_{j=1}^{m} w_{ij}^k u(x_j) + b_i^k) \sigma(\omega_k y + \beta)$$

where $j = 0, \cdots, m$, and x_j are the uniform mesh points on the space, for example $x_j = \dfrac{j}{m} \in [0,1]$ in one dimensional case.

Banch block: $b_k(u(x_1), \cdots, u(x_m)) = \sum_{i=1}^{n} c_i^k \sigma(\sum_{j=1}^{m} w_{ij}^k u(x_j) + b_i^k)$

Trunk block: $t_k(y) = \sigma(\omega_k y + \beta_k)$

The full network is done as $\mathcal{G}[u](y) = \sum_{k=1}^{p} b_k t_k(y)$

Neural Operators

Network Structure

Neural Operators

Network Structure

Lifting $\mathscr{P}$: Using a pointwise function $\mathbb{R}^{d_a} \to \mathbb{R}^{d_{v_0}}$, map the input function

Neural Operators

Network Structure

Lifting$\mathscr{P}$: Using a pointwise function $\mathbb{R}^{d_a} \to \mathbb{R}^{d_{v_0}}$, map the input function

$$\{a(x) : \Omega \to \mathbb{R}^a\} \to \{v_0(x) : \Omega \to \mathbb{R}^{d_{v_0}}\}. \; (\, d_{v_0} > d_a)$$

Neural Operators

Network Structure

Lifting $\mathscr{P}$: Using a pointwise function $\mathbb{R}^{d_a} \to \mathbb{R}^{d_{v_0}}$, map the input function

$$\{a(x) : \Omega \to \mathbb{R}^{a}\} \to \{v_0(x) : \Omega \to \mathbb{R}^{d_{v_0}}\}.\ (\ d_{v_0} > d_a)$$

Iterative Kernel Integration: for $l = 0, \cdots, L-1$, map each hidden representation to next

Neural Operators

Network Structure

Lifting$\mathscr{P}$: Using a pointwise function $\mathbb{R}^{d_a} \to \mathbb{R}^{d_{v0}}$, map the input function

$$\{a(x) : \Omega \to \mathbb{R}^a\} \to \{v_0(x) : \Omega \to \mathbb{R}^{d_{v0}}\}. \; (d_{v_0} > d_a)$$

Iterative Kernel Integration: for $l = 0, \cdots, L-1$, map each hidden representation to next

$$\{v_l(x) : \Omega \to \mathbb{R}^{d_{v_l}}\} \to \{v_{l+1}(x) : \Omega \to \mathbb{R}^{d_{v_{l+1}}}\}$$

Neural Operators

Network Structure

Lifting$\mathscr{P}$: Using a pointwise function $\mathbb{R}^{d_a} \to \mathbb{R}^{d_{v_0}}$, map the input function

$$\{a(x) : \Omega \to \mathbb{R}^{a}\} \to \{v_0(x) : \Omega \to \mathbb{R}^{d_{v_0}}\}. \ (\, d_{v_0} > d_a)$$

Iterative Kernel Integration: for $l = 0, \cdots, L-1$, map each hidden representation to next

$$\{v_l(x) : \Omega \to \mathbb{R}^{d_{v_l}}\} \to \{v_{l+1}(x) : \Omega \to \mathbb{R}^{d_{v_{l+1}}}\}$$

Projection: using a pointwise function $\mathscr{Q} : \mathbb{R}^{d_{v_T}} \to \mathbb{R}^{d_u}$

Neural Operators

Network Structure

Lifting$\mathscr{P}$: Using a pointwise function $\mathbb{R}^{d_a} \to \mathbb{R}^{d_{v0}}$, map the input function

$$\{a(x) : \Omega \to \mathbb{R}^a\} \to \{v_0(x) : \Omega \to \mathbb{R}^{d_{v0}}\}. \; (\, d_{v_0} > d_a)$$

Iterative Kernel Integration: for $l = 0,\cdots, L-1$, map each hidden representation to next

$$\{v_l(x) : \Omega \to \mathbb{R}^{d_{v_l}}\} \to \{v_{l+1}(x) : \Omega \to \mathbb{R}^{d_{v_{l+1}}}\}$$

Projection: using a pointwise function $\mathscr{Q} :\mathbb{R}^{d_{v_T}} \to \mathbb{R}^{d_u}$

$$\{v_L(x) : \Omega \to \mathbb{R}^{d_{v_L}}\} \to \{u(x) : \Omega \to \mathbb{R}^{d_u}\}$$

Neural Operators

Integral Kernel Operators

Neural Operators

Integral Kernel Operators

V1:

Neural Operators

Integral Kernel Operators

V1:

$$(\mathscr{K}_l[v_l])(x) = \int_\Omega \kappa^{(l)}(x, y) v_l(y)\, dy \qquad \forall x \in \Omega\,.$$

Neural Operators

Integral Kernel Operators

V1:

$$(\mathcal{K}_l[v_l])(x) = \int_\Omega \kappa^{(l)}(x, y) v_l(y)\, dy \qquad \forall x \in \Omega\,.$$

where $k^{(l)} : \Omega \times \Omega \to \mathcal{R}^{d_{v_{l+1}} \times d_{v_l}}$.

Neural Operators

Integral Kernel Operators

V1:

$$(\mathcal{K}_l[v_l])(x) = \int_\Omega \kappa^{(l)}(x, y) v_l(y)\, dy \qquad \forall x \in \Omega\,.$$

where $k^{(l)} : \Omega \times \Omega \to \mathcal{R}^{d_{v_{l+1}} \times d_{v_l}}$.

V2:

Neural Operators

Integral Kernel Operators

V1:

$$(\mathcal{K}_l[v_l])(x) = \int_\Omega \kappa^{(l)}(x, y) v_l(y)\, dy \qquad \forall x \in \Omega\,.$$

where $k^{(l)} : \Omega \times \Omega \to \mathcal{R}^{d_{v_{l+1}} \times d_{v_l}}$.

V2:

$$(\mathcal{K}_l[v_l])(x) = \int_\Omega \kappa^{(l)}(x, y, a(x), a(y)) v_l(y)\, dy$$

Neural Operators

Integral Kernel Operators

V1:

$$(\mathcal{K}_l[v_l])(x) = \int_\Omega \kappa^{(l)}(x,y)v_l(y)\,dy \qquad \forall x \in \Omega\,.$$

where $k^{(l)} : \Omega \times \Omega \to \mathcal{R}^{d_{v_{l+1}} \times d_{v_l}}$.

V2:

$$(\mathcal{K}_l[v_l])(x) = \int_\Omega \kappa^{(l)}(x,y,a(x),a(y))v_l(y)\,dy$$

where $k^{(l)} : \Omega \times \Omega \times \mathbb{R}^{d_a} \times \mathbb{R}^{d_a} \to \mathcal{R}^{d_{v_{l+1}} \times d_{v_l}}$.

Neural Operators

Integral Kernel Operators

V1:

$$(\mathscr{K}_l[v_l])(x) = \int_\Omega \kappa^{(l)}(x,y)v_l(y)\,dy \qquad \forall x \in \Omega\,.$$

where $k^{(l)} : \Omega \times \Omega \to \mathscr{R}^{d_{v_{l+1}} \times d_{v_l}}$.

V2:

$$(\mathscr{K}_l[v_l])(x) = \int_\Omega \kappa^{(l)}(x,y,a(x),a(y))v_l(y)\,dy$$

where $k^{(l)} : \Omega \times \Omega \times \mathbb{R}^{d_a} \times \mathbb{R}^{d_a} \to \mathscr{R}^{d_{v_{l+1}} \times d_{v_l}}$.

V3:

Neural Operators

Integral Kernel Operators

V1:

$$(\mathcal{K}_l[v_l])(x) = \int_\Omega \kappa^{(l)}(x, y) v_l(y)\, dy \qquad \forall x \in \Omega\,.$$

where $k^{(l)} : \Omega \times \Omega \to \mathcal{R}^{d_{v_{l+1}} \times d_{v_l}}$.

V2:

$$(\mathcal{K}_l[v_l])(x) = \int_\Omega \kappa^{(l)}(x, y, a(x), a(y)) v_l(y)\, dy$$

where $k^{(l)} : \Omega \times \Omega \times \mathbb{R}^{d_a} \times \mathbb{R}^{d_a} \to \mathcal{R}^{d_{v_{l+1}} \times d_{v_l}}$.

V3:

$$(\mathcal{K}_l[v_l])(x) = \int_\Omega \kappa^{(l)}(x, y, v_l(x), v_l(y)) v_l(y)\, dy$$

Neural Operators

Integral Kernel Operators

V1:

$$(\mathcal{K}_l[v_l])(x) = \int_\Omega \kappa^{(l)}(x,y)v_l(y)\,dy \qquad \forall x \in \Omega\,.$$

where $k^{(l)} : \Omega \times \Omega \to \mathcal{R}^{d_{v_{l+1}} \times d_{v_l}}$.

V2:

$$(\mathcal{K}_l[v_l])(x) = \int_\Omega \kappa^{(l)}(x,y,a(x),a(y))v_l(y)\,dy$$

where $k^{(l)} : \Omega \times \Omega \times \mathbb{R}^{d_a} \times \mathbb{R}^{d_a} \to \mathcal{R}^{d_{v_{l+1}} \times d_{v_l}}$.

V3:

$$(\mathcal{K}_l[v_l])(x) = \int_\Omega \kappa^{(l)}(x,y,v_l(x),v_l(y))v_l(y)\,dy$$

where $k^{(l)} : \Omega \times \Omega \times \mathbb{R}^{d_l} \times \mathbb{R}^{d_l} \to \mathcal{R}^{d_{v_{l+1}} \times d_{v_l}}$.

Neural Operators

Integral Kernel Operators

V1:

$$(\mathcal{K}_l[v_l])(x) = \int_\Omega \kappa^{(l)}(x, y) v_l(y)\, dy \qquad \forall x \in \Omega\,.$$

where $k^{(l)} : \Omega \times \Omega \to \mathscr{R}^{d_{v_{l+1}} \times d_{v_l}}$.

V2:

$$(\mathcal{K}_l[v_l])(x) = \int_\Omega \kappa^{(l)}(x, y, a(x), a(y)) v_l(y)\, dy$$

where $k^{(l)} : \Omega \times \Omega \times \mathbb{R}^{d_a} \times \mathbb{R}^{d_a} \to \mathscr{R}^{d_{v_{l+1}} \times d_{v_l}}$.

V3:

$$(\mathcal{K}_l[v_l])(x) = \int_\Omega \kappa^{(l)}(x, y, v_l(x), v_l(y)) v_l(y)\, dy$$

where $k^{(l)} : \Omega \times \Omega \times \mathbb{R}^{d_l} \times \mathbb{R}^{d_l} \to \mathscr{R}^{d_{v_{l+1}} \times d_{v_l}}$.

$$v_{l+1}(x) = \sigma(W_l v_l(x) + \mathcal{K}_l([v_l])(x) + b_l(x))$$

Neural Operators

Integral Kernel Operators

V1:

$$(\mathcal{K}_l[v_l])(x) = \int_\Omega \kappa^{(l)}(x, y)v_l(y)\, dy \qquad \forall x \in \Omega\,.$$

where $k^{(l)} : \Omega \times \Omega \to \mathcal{R}^{d_{v_{l+1}} \times d_{v_l}}$.

V2:

$$(\mathcal{K}_l[v_l])(x) = \int_\Omega \kappa^{(l)}(x, y, a(x), a(y))v_l(y)\, dy$$

where $k^{(l)} : \Omega \times \Omega \times \mathbb{R}^{d_a} \times \mathbb{R}^{d_a} \to \mathcal{R}^{d_{v_{l+1}} \times d_{v_l}}$.

V3:

$$(\mathcal{K}_l[v_l])(x) = \int_\Omega \kappa^{(l)}(x, y, v_l(x), v_l(y))v_l(y)\, dy$$

where $k^{(l)} : \Omega \times \Omega \times \mathbb{R}^{d_l} \times \mathbb{R}^{d_l} \to \mathcal{R}^{d_{v_{l+1}} \times d_{v_l}}$.

$$v_{l+1}(x) = \sigma(W_l v_l(x) + \mathcal{K}_l([v_l])(x) + b_l(x))$$

Fourier Neural Operator

Motivation

Fourier Neural Operator

Motivation

Define for f:

Fourier Neural Operator

Motivation

Define for f:

$$(\mathcal{F}f)(k) = \int f(x)\exp(-2\pi i\langle x, k\rangle)$$

$$(\mathcal{F}^{-1}f)(x) = \int f(k)\exp(2\pi i\langle x, k\rangle)$$

Fourier Neural Operator

Motivation

Define for f:

$$(\mathcal{F}f)(k) = \int f(x) \exp(-2\pi i \langle x, k \rangle)$$

$$(\mathcal{F}^{-1}f)(x) = \int f(k) \exp(2\pi i \langle x, k \rangle)$$

Fourier integral operator $\mathcal{K}$:

Fourier Neural Operator

Motivation

Define for f:

$$(\mathscr{F}f)(k) = \int f(x)\exp(-2\pi i\langle x, k\rangle)$$

$$(\mathscr{F}^{-1}f)(x) = \int f(k)\exp(2\pi i\langle x, k\rangle)$$

Fourier integral operator $\mathscr{K}$:

$$(\mathscr{K}(\phi)v_l)(x) = \mathscr{F}^{-1}(R_\theta \cdot \mathscr{F}(v_l))(x)$$

Fourier Neural Operator

Motivation

Define for f:

$$(\mathcal{F}f)(k) = \int f(x)\exp(-2\pi i\langle x, k\rangle)$$

$$(\mathcal{F}^{-1}f)(x) = \int f(k)\exp(2\pi i\langle x, k\rangle)$$

Fourier integral operator $\mathcal{K}$:

$$(\mathcal{K}(\phi)v_l)(x) = \mathcal{F}^{-1}(R_\theta \cdot \mathcal{F}(v_l))(x)$$

For frequency mode $k \in \mathbb{Z}^d$ and $v_l : \Omega \subset \mathbb{R}^d \to \mathbb{R}^{d_{v_l}}$, we have $(\mathcal{F}v_l)(k) \in \mathbb{C}^{d_{v_l}}$ and $R_\theta(k) \in \mathbb{C}^{d_{v_l} \times d_{v_{l+1}}}$

Fourier Neural Operator

Choice for $\boldsymbol{R}$

Fourier Neural Operator

Choice for $\boldsymbol{R}$

Direct: Define the parameters $\theta_k \in \mathbb{C}^{d_{v_l} \times d_{v_{l+1}}}$ for each wave number k:

Fourier Neural Operator

Choice for $\boldsymbol{R}$

Direct: Define the parameters $\theta_k \in \mathbb{C}^{d_{v_l} \times d_{v_{l+1}}}$ for each wave number k:

$$R_\theta(k, \mathcal{F} v_l(k)) := \theta_k$$

Fourier Neural Operator

Choice for $\boldsymbol{R}$

Direct: Define the parameters $\theta_k \in \mathbb{C}^{d_{v_l} \times d_{v_{l+1}}}$ for each wave number k:

$$R_\theta(k, \mathcal{F} v_l(k)) := \theta_k$$

Linear: Deine the parameters $\theta_{1,k} \in \mathbb{C}^{d_{v_{l+1}} \times d_{v_l} \times d_a}$ and $\theta_{2,k} \in \mathbb{C}^{d_{v_{l+1}} \times d_{v_l}}$ for each wave number k:

Fourier Neural Operator

Choice for $\boldsymbol{R}$

Direct: Define the parameters $\theta_k \in \mathbb{C}^{d_{v_l} \times d_{v_{l+1}}}$ for each wave number k:

$$R_\theta(k, \mathcal{F} v_l(k)) := \theta_k$$

Linear: Deine the parameters $\theta_{1,k} \in \mathbb{C}^{d_{v_{l+1}} \times d_{v_l} \times d_a}$ and $\theta_{2,k} \in \mathbb{C}^{d_{v_{l+1}} \times d_{v_l}}$ for each wave number k:

$$R_\theta(k, \mathcal{F} v_l(k)) := \theta_{1,k}(\mathcal{F} a)(k) + \theta_{2,k}$$

Fourier Neural Operator

Choice for $\boldsymbol{R}$

Direct: Define the parameters $\theta_k \in \mathbb{C}^{d_{v_l} \times d_{v_{l+1}}}$ for each wave number k:

$$R_\theta(k, \mathcal{F} v_l(k)) := \theta_k$$

Linear: Deine the parameters $\theta_{1,k} \in \mathbb{C}^{d_{v_{l+1}} \times d_{v_l} \times d_a}$ and $\theta_{2,k} \in \mathbb{C}^{d_{v_{l+1}} \times d_{v_l}}$ for each wave number k:

$$R_\theta(k, \mathcal{F} v_l(k)) := \theta_{1,k}(\mathcal{F} a)(k) + \theta_{2,k}$$

Neural Network: Let $\Phi_\theta : \mathbb{Z}^d \times \mathbb{C}^{d_a} \to \mathbb{C}^{d_{v_l} \times d_{v_{l+1}}}$ then:

Fourier Neural Operator

Choice for $\boldsymbol{R}$

Direct: Define the parameters $\theta_k \in \mathbb{C}^{d_{v_l} \times d_{v_{l+1}}}$ for each wave number k:

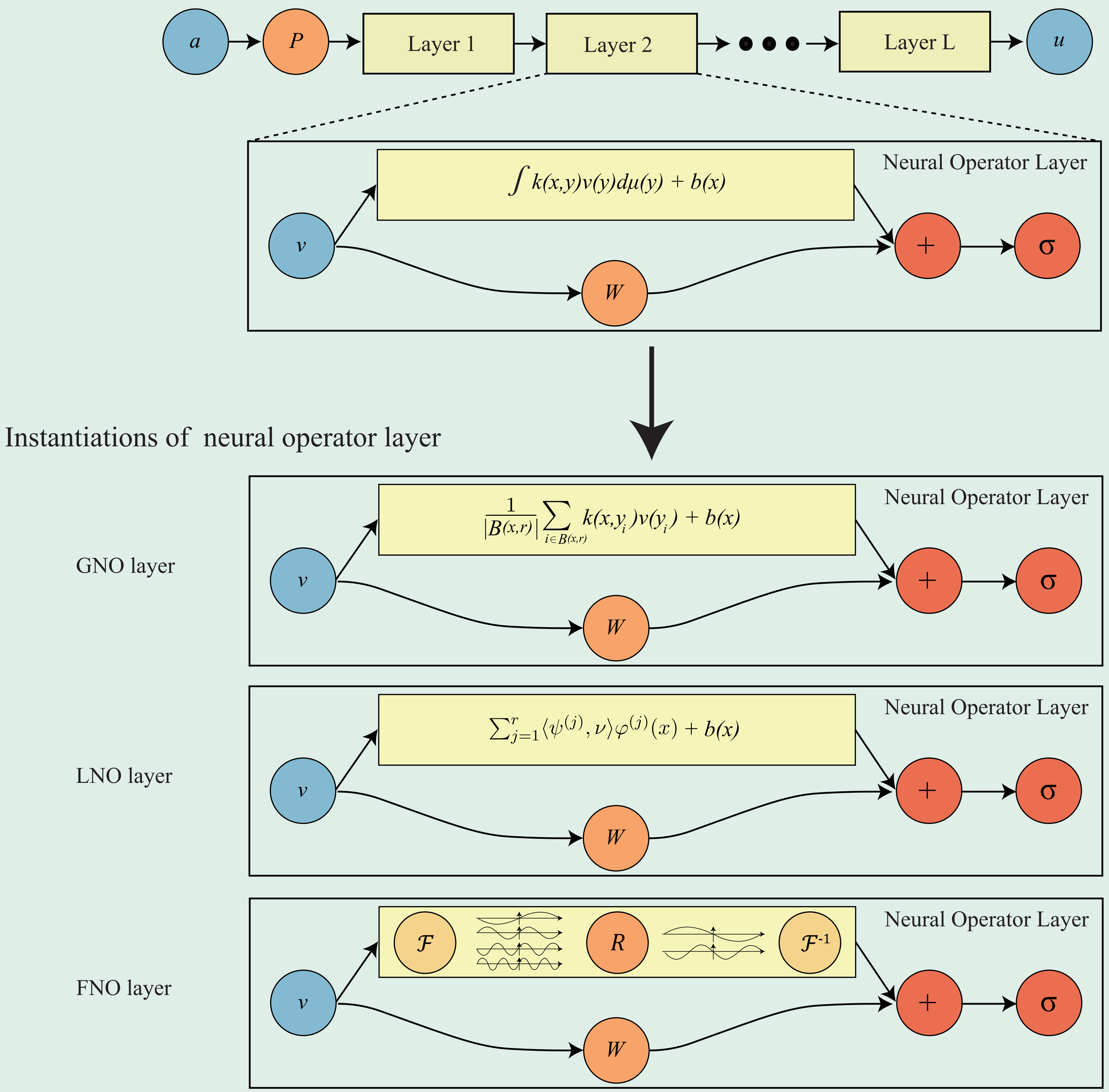
$$R_\theta(k, \mathcal{F}v_l(k)) := \theta_k$$

Linear: Deine the parameters $\theta_{1,k} \in \mathbb{C}^{d_{v_{l+1}} \times d_{v_l} \times d_a}$ and $\theta_{2,k} \in \mathbb{C}^{d_{v_{l+1}} \times d_{v_l}}$ for each wave number k:

$$R_\theta(k, \mathcal{F}v_l(k)) := \theta_{1,k}(\mathcal{F}a)(k) + \theta_{2,k}$$

Neural Network: Let $\Phi_\theta : \mathbb{Z}^d \times \mathbb{C}^{d_a} \to \mathbb{C}^{d_{v_l} \times d_{v_{l+1}}}$ then:

$$R_\theta(k, \mathcal{F}v_l(k)) := \Phi_\theta(k, (\mathcal{F}a)(k))$$



a
P
Layer 1
Layer 2
Layer L
u
Neural Operator Layer
∫ k(x,y)v(y)dμ(y) + b(x)
v
W
+
σ
Instantiations of neural operator layer
GNO layer
Neural Operator Layer
1/|B(x,r)| Σ_{i∈B(x,r)} k(x,y_i)v(y_i) + b(x)
LNO layer
Neural Operator Layer
Σ_{j=1}^r ⟨ψ^(j), ν⟩φ^(j)(x) + b(x)
FNO layer
Neural Operator Layer
ℱ
R
ℱ^-1