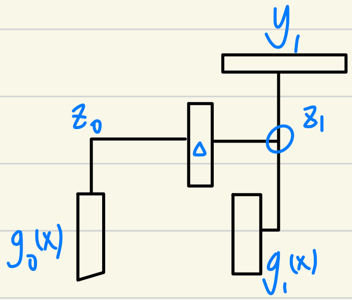
Pipeline :

initial input : $x \in d^{in}$, $d^{in} = 2$ or 3 $\Leftarrow$ 2d or 3d coordinate



$z_0 \xrightarrow{g_1} z_1$: $z_1 = g_1(x) \circ (W_1 z_0 + b_1)$

Δ : $W_1 z_0 + b_1$

$g_0(x) = \sin(w_0 x + \phi_0)$

$g_0 = z_0$

$\underbrace{w_0}_{d^h \times d^{in}} \underbrace{x}_{d^{in}} + \phi_0 \in d^h$

$w_0 \in [-B_0, B_0]$, if w_1 also $\in [-B_0, B_0]$

$2B_0 = B \Rightarrow z_1$'s bandwidth will be $\le 2B$

$y_1 = W_1^{out} z_1 + b_1^{out}$

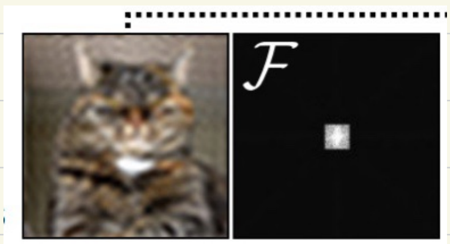
$w_0 X$

din

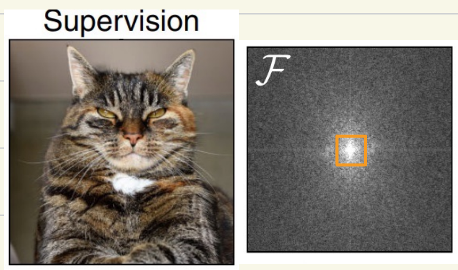
$d_h \underbrace{[\qquad]}_{W_0} \underbrace{[\ ]}_{X} d_{in}$

what are entries of ω_i?
random uniform initialization from [-B_0, B_0], or in coding
random.uniform(-B_i, B_i, d_h * d_in)

y_1 and its spectrum



Supervisory



learn for each (u, v) coordinate

Guess_1: the output dimension should be 1, which represents the value of frequency magnitude because “the network is characterized entirely by its Fourier specturm”. We should train in the Fourier frequency domain, and visualize the performance in image domain by inverse Fourier transform.

supervisory image

set manually

$$f_\theta(w_i \mathbf{X}) \xrightarrow{FFT} \tilde{F}(u,v) = \tilde{F} \quad - \quad F \xleftarrow{FFT} f(\mathbf{X})$$

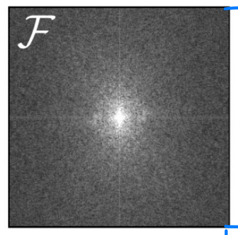
unknown,
what we want to train

predicted output

real frequency value by
supervisory image

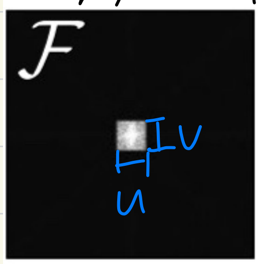
$\mathbf{X} = (x,y)$ $\quad w_i \mathbf{X}$ limit the possible coordinate in frequency domain (u, v) will not appear out of expected frequency domain

$f_\theta(\mathbf{X}) \xrightarrow{FFT} F(u,v)$



(u,v) domain

$f_\theta(w_i \mathbf{X}) \xrightarrow{FFT} F(u,v)$



(u,v) domain

that's how ω works

$$g_i(\mathbf{X}) = \sin(w_i \mathbf{X} + \phi_i)$$

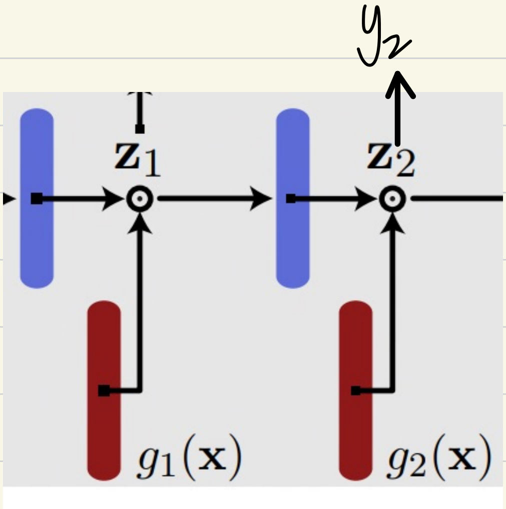
$$z_i = g_i(\mathbf{X})(W_i z_{i-1} + b_i)$$

$$y_i = W_i^{out} z_i + b_i^{out}$$

— : freeze

— : trainable

$f_\theta(w_i \mathbf{X})$



$$z_2 = g_2(x) \circ (\underbrace{W_2 z_1 + b_2}_{\le 2B})$$

$\downarrow$

$\underbrace{\sin(w_2 x + \phi_2)}_{B}$ $\quad \Rightarrow z_2$ and $y_2 \le 3B$

frequency

$f_\theta(x)$ = the value of signal (maybe grayscale or RGB, need to be checked)

where x is input coordinate

$y_i = f_\theta(\omega_i; \mathbf{X})$

4.1 image fitting loss: $\sum_{i=1/4, 4} \| y_i - y_{i,GT} \|$

4.2 NeRF :

$f_{NeRF}(\mathbf{X}) = (c, \sigma) \Rightarrow C(r)$

$f_{BACON}(\mathbf{X}) = c$, where to get σ? Got by NeRF

$(c, \sigma) \Rightarrow C(r)$