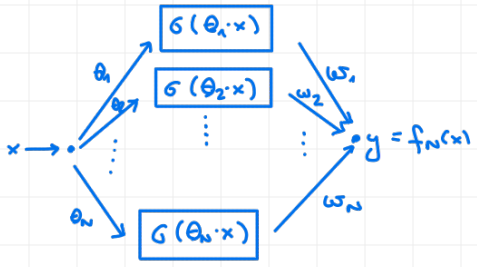
EXTRA

A (fully connected two-layer) neural network with $N \in \mathbb{N}$ neurons is a function of the form:

$$\mathbb{R}^n \ni x \longmapsto f_N(x) = \frac{1}{N} \sum_{i=1}^{N} w_i \sigma(\theta_i \cdot x) = \frac{1}{N} \sum_{i=1}^{N} \Phi(\xi_i, x)$$



where $w_i \in \mathbb{R}$, $\theta_i \in \mathbb{R}^n$ are parameters to be chosen,

and $\xi_i = (w_i, \theta_i) \in \mathbb{R}^{n+1}$, $\Phi(\xi, x) = w\,\sigma(\theta \cdot x)$.

σ is a (nonlinear) activation function that is given.

Training a neural network is the optimization of the parameters ξ_i so that f_N approximates a given function f (for example, wrt the ℓ^2 loss):

$$F((\xi_i)_i) = \int (f(x) - f_N(x))^2 \, dx \quad \leftarrow \text{Choose } \xi_i \text{ to minimize this.}$$

If we denote $\mu_N := \frac{1}{N} \sum_{i=1}^{N} \delta_{\xi_i} \in \mathcal{P}(\mathbb{R}^{n+1})$ (the empirical measure of the weights), then:

$$f_N(x) = \int \Phi(\xi, x) \mu_N(d\xi).$$

So $F = F(\mu_N) = \int \left(f(x) - \int \Phi(\xi, x) \mu_N(d\xi)\right)^2$ is a functional in the space of measures.

The gradient descent of F wrt the parameters is the Wasserstein gradient flow of F wrt the measure. As such, it is governed by a PDE:

$$\boxed{\partial_t \mu_t = \operatorname{div}(\mu_t \nabla \mathcal{L}(\mu_t)) + \operatorname{div}(\mu_t \nabla S),} \quad \text{where}$$

$$\mathcal{L}(\mu_t)(\xi) = 2 \int K(\xi, \bar{\xi}) \, d\mu_t(\bar{\xi}), \quad K(\xi, \bar{\xi}) = \frac{1}{2} \int \Phi(\xi, x) \Phi(\bar{\xi}, x) \, dx$$

$$S(\xi) = -\int \Phi(\xi, x) f(x) \, dx$$