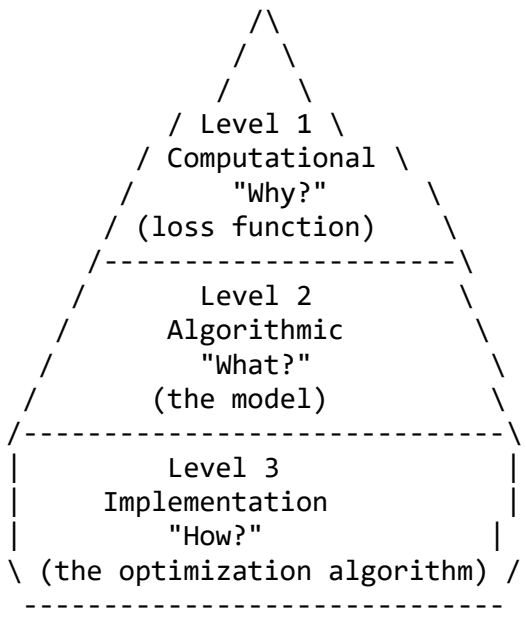
Aside: Marss's Levels of analysis

1. Computations "why?" eg loss fuction
2. Algorithmic "what?" eg the model
3. IMplemenation "whow?" eg the optimization alogo



How is dataset generated?

Let's say ((x, y) p(x, y)), where (x) and (y) is some data.

Training: (= {(x, y) })

Assuming they are i.i.d.,

$$p(\mathcal{D}) = \prod p(x, y) = \prod p(x) p(y \mid x)$$

We are trying to learn (p_(y x)), basically it's a model of the true (p(y x)).

A good model should make the data look probable.

Therefore, we must choose () such that:

$$p(\mathcal{D}) = \prod p(x) p_\theta(y \mid x)$$

is maximized.

But the problem is that (p(x)), multiplying many such terms would make the result closer to zero.

Therefore, we take logarithm:

$$\log p(\mathcal{D}) = \sum \log p(x_i) + \log p_\theta(y_i \mid x_i)$$

$$= \sum \log p_\theta(y_i \mid x_i) + \text{const} \quad (p(x_i) \text{ doesn't depend on } \theta)$$

Then,

$$\theta \leftarrow \text{argmax} \sum \log p_\theta(y_i \mid x_i) \quad \text{(MLE)}$$

$$\theta \leftarrow \text{argmin} - \sum \log p_\theta(y_i \mid x_i) \quad \text{(NLL)}$$

Therefore, what is loss function?

The loss function quantifies how bad () is. We want the least bad ().