TRAINING

$\underline{x}$ features

$\underline{y}$ labels

$\underline{w}$ weights

$\mathcal{D}$ data

NN = model $p(\underline{y} \mid \underline{x}, \underline{w}, \mathcal{D})$

"STANDARD" NN : learn ML estimator

$$\underline{w}^* = \arg\max_w \log \mathcal{L}(\mathcal{D} \mid \underline{w})$$

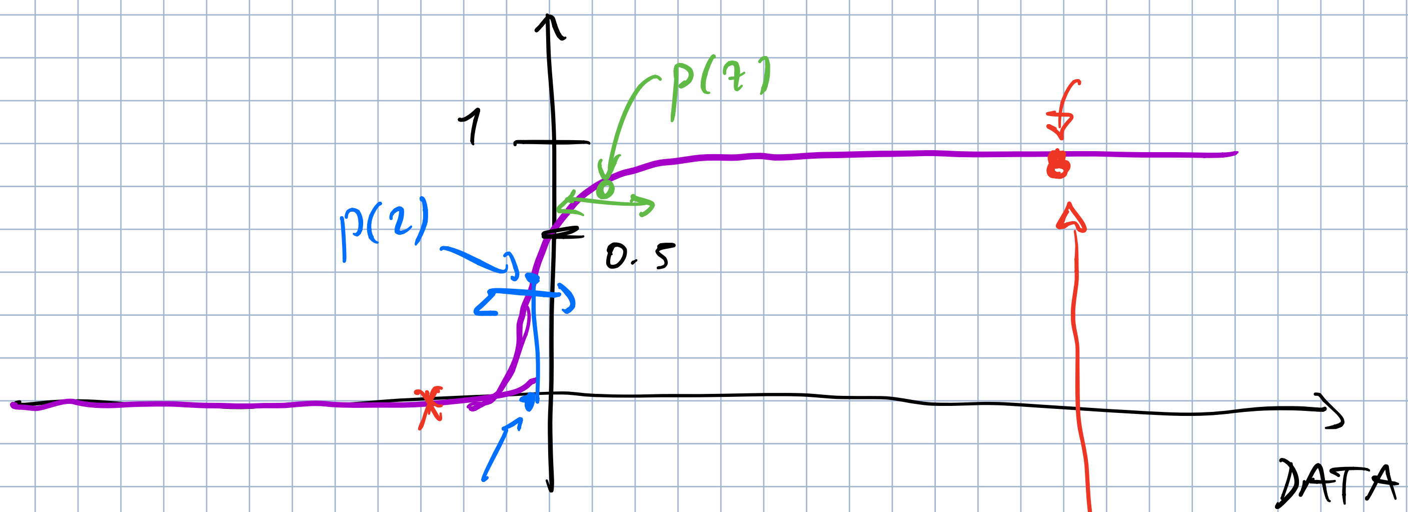
$$= \arg\max_w \sum p(y_i \mid x_i, w_i, \mathcal{D})$$

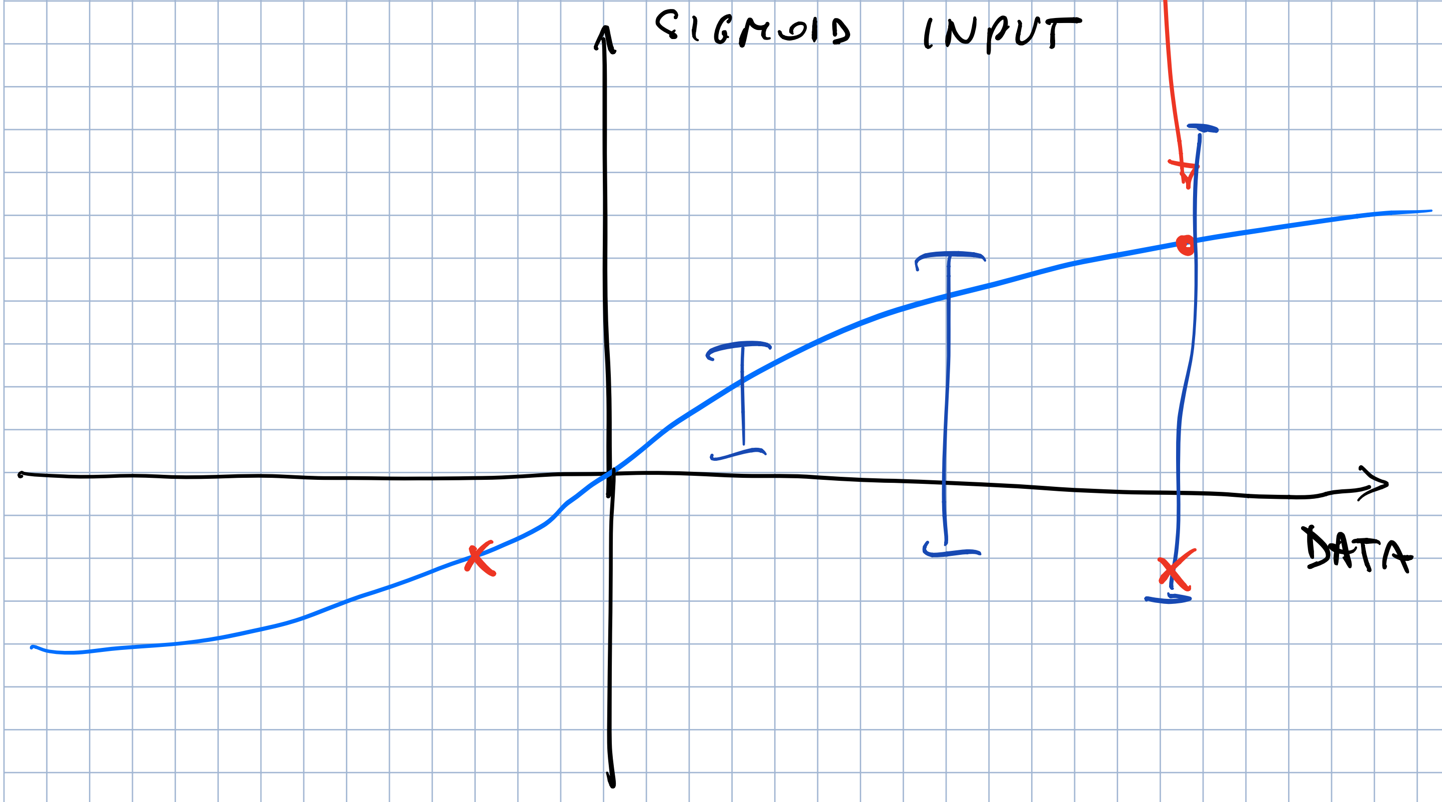
NO GLOBAL MINIMUM $\forall$ $\mathcal{D}$!

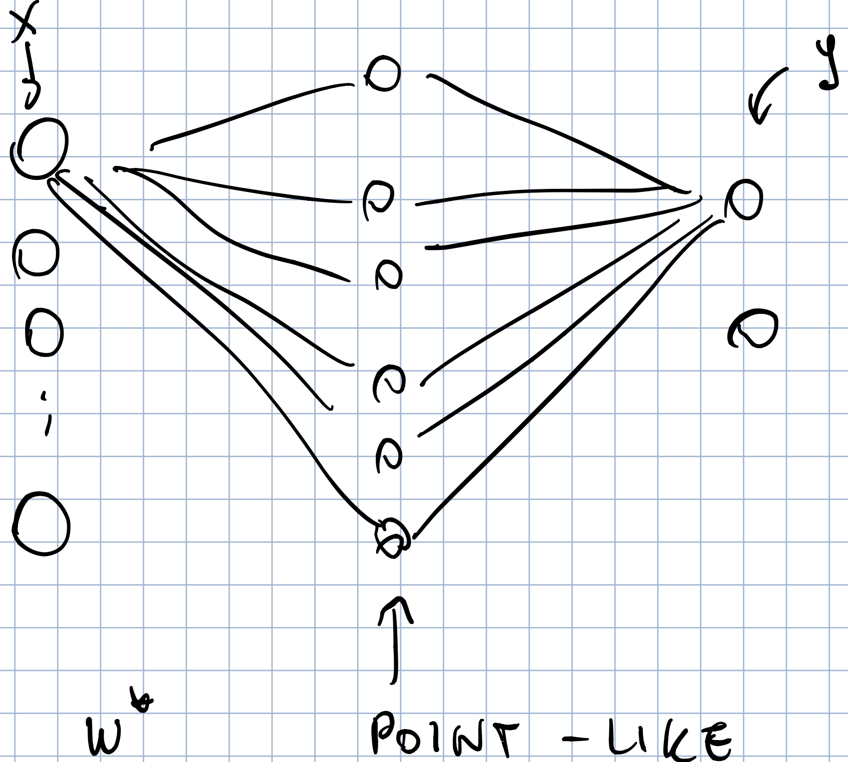
or MAP w. prior

$$w^{MAP} = \arg\max_w \log \mathcal{L}(\mathcal{D} \mid \underline{w}) + \log p(\underline{w})$$

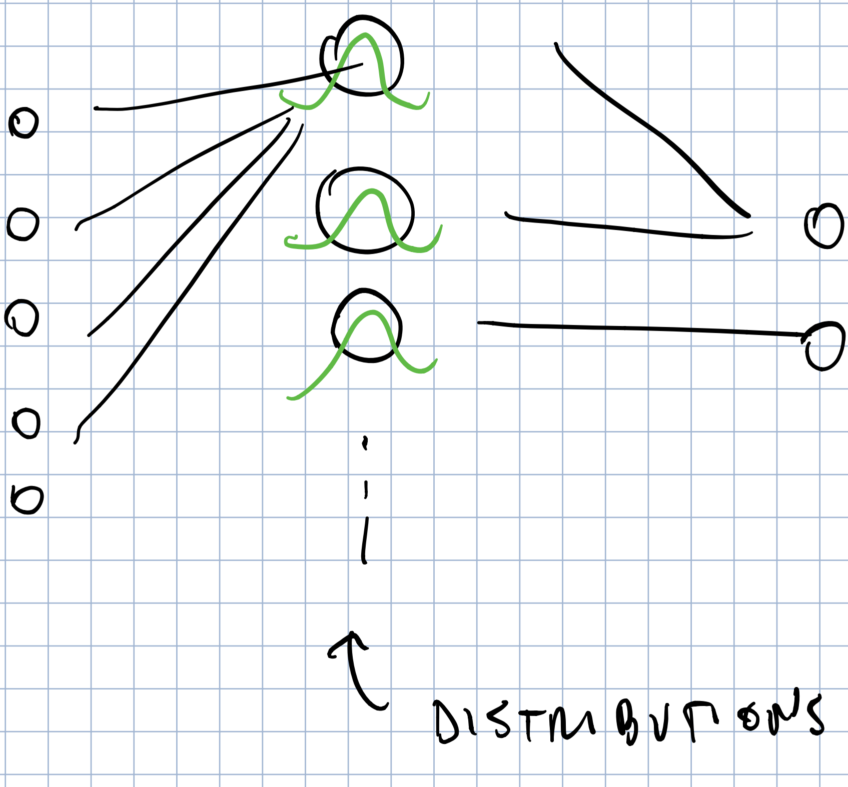
$p(7) \sim 56\%$ $p(2) \sim 43\%$







x
y
W
POINT-LIKE

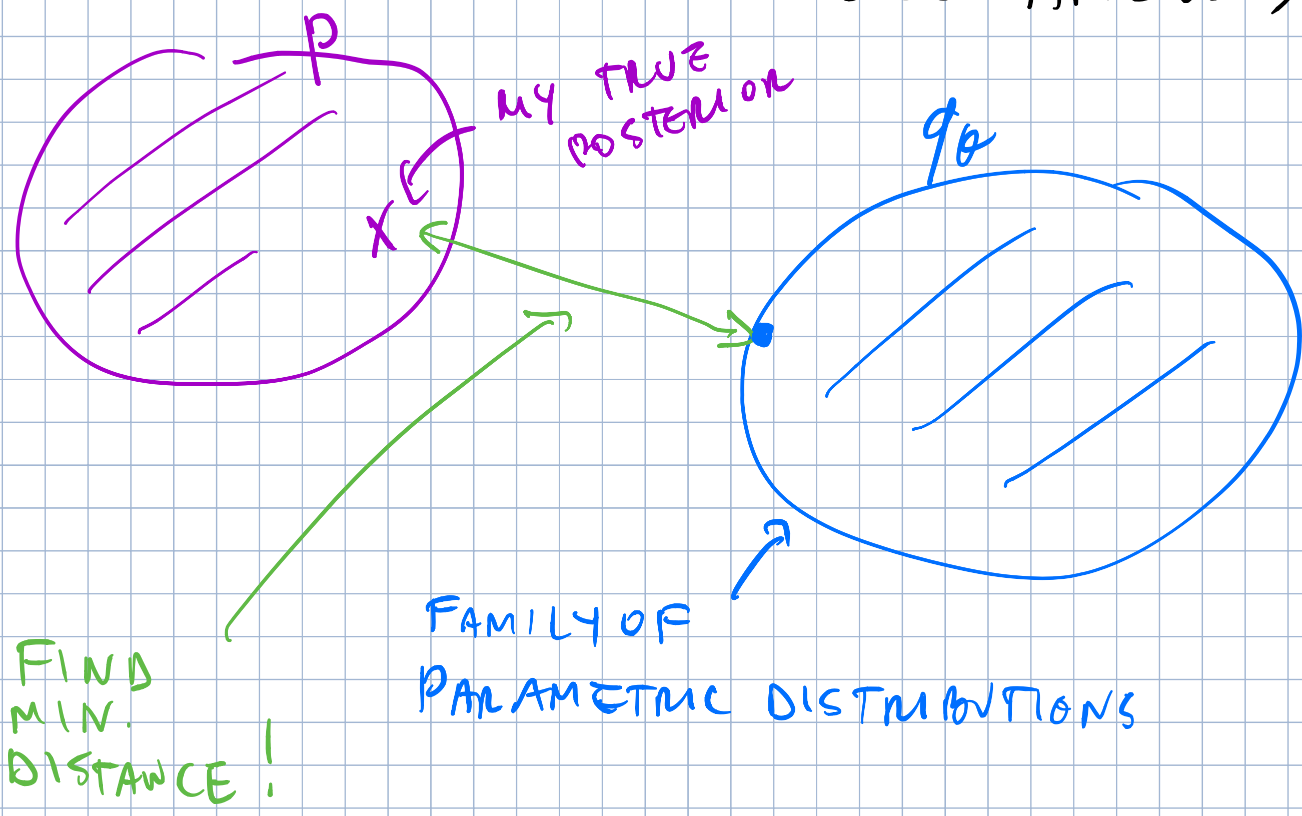


DISTRIBUTIONS

NOW: LEARN $\boxed{p(\underline{w} \mid D)}$ POSTERIOR ON WEIGHTS

$$\boxed{p(\underline{y} \mid \underline{x}) = \mathbb{E}_{p(\underline{w} \mid D)}\left[p(\underline{y} \mid \underline{x}, \underline{w}) \right]}$$

$p(\underline{w}, D)$ full unknown distribution (COULD DO HMC ...)



$q(\underline{w} \mid \theta)$ PARAMETRIC DISTRIBUTION THAT "BEST APPROXIMATES" $p(\underline{w} \mid D)$

KL DIVERGENCE

$$\boxed{KL[f \| g] = \int f(x) \log \frac{f(x)}{g(x)} dx}$$

MINIMIZE:

$$KL[q(\underline{w}|\theta) \| p(\underline{w}|D)] =$$

$$= \int dw\, q(\underline{w}|\underline{\theta}) \log \frac{q(\underline{w}|\underline{\theta})}{p(\underline{w}|D)}$$

BAYES: $p(\underline{w}|D) \propto \mathcal{L}(D|w)\, p(w)$

$$= \int dw\, q(\underline{w}|\underline{\theta}) \log \frac{q(\underline{w}|\underline{\theta})}{\mathcal{L}(D, \underline{w})\, p(\underline{w})}$$

$$= \underbrace{\int dw\, q(\underline{w}|\underline{\theta}) \log \frac{q(\underline{w}|\underline{\theta})}{p(\underline{w})}} - \int dw\, q(w,\theta) \times \log \mathcal{L}(D|\underline{w})$$

$$= KL[q \| p(\underline{w})] + \mathbb{E}_q[-\log \mathcal{L}]$$

OPTIMIZE:

$$\boxed{\mathcal{F}(\mathcal{D},\theta) = KL\left[q_\theta(\underline{w}\,|\,\mathcal{D})\,\|\,P(\underline{w})\right] + \mathbb{E}_{q_\theta}\left[-\log \mathcal{L}(\mathcal{D}\,|\,\underline{w})\right]}$$

$$\approx \sum_{(i)\in \text{SAMPLES}} \underbrace{\log q_\theta(\underline{w}^{(i)}\,|\,\mathcal{D})}_{\text{TARGET}} - \underbrace{\log p(\underline{w}^{(i)})}_{\text{PRIOR}} - \underbrace{\log \mathcal{L}(\mathcal{D}\,|\,\underline{w}^{(i)})}_{\text{LOSS}}$$

$$= - \text{ELBO} \quad (\text{Evidence Lower Bound})$$

- BERNOULLI VIA DROPOUT 1506.02142

Proof that $\exists$ $p(\underline{w})$ such that

$$\mathcal{F} \simeq \sum_{\text{DATA}} \mathcal{L}_{\tilde{i}}(D, w) - \lambda \sum_{\text{WEIGHTS}} \| w_i \|^2$$

+ DROPOUT AFTER EACH LAYER

NOT SO "BAYESIAN" ...

- BAYES BY BACKPROP 1505.05424

Sample weights in back-propagation (Once for batch)

$$\mathcal{F} \simeq \sum \log q_\theta(w^{(i)} | D) - \log p(w^i) - \log \mathcal{L}(D | w^{(i)})$$

Milions of samples needed + same sample for every batch

- LOCAL REPARAMETRIZATION TRICK 1506.02557

Sample layer activations instead of $w^{(i)}$

- FLIPOUT 1803.04386

Samples weight independently within a mini-batch.

- NORMALIZING FLOWS 1802.04908

Differentiable mapping to complex distributions

—

SEE NOTEBOOK!

tfp : FLIPOUT + LOCAL R.T.

PREDICTIONS & UNCERTAINTY

$$p(\underline{y}^*, \underline{x}^*) = \mathbb{E}_{q_\theta}\left[p(\underline{y}^*, \underline{x}^*) \right]$$

$$\simeq \frac{1}{N_{samples}} \sum p(\underline{y}^* \mid \underline{x}^*, \underline{w}^{(i)})$$

Each time draw 1 sample

$$Var_q\left(p(\underline{y}^*, \underline{x}^*) \right) = \mathbb{E}\left[\underline{y}^* \underline{y}^{*T} \right] - \mathbb{E}_q(\underline{y}^*)\, \mathbb{E}_q(\underline{y}^*)^T$$

- Train last layer to learn μ, σ $\forall$ label
- Then:

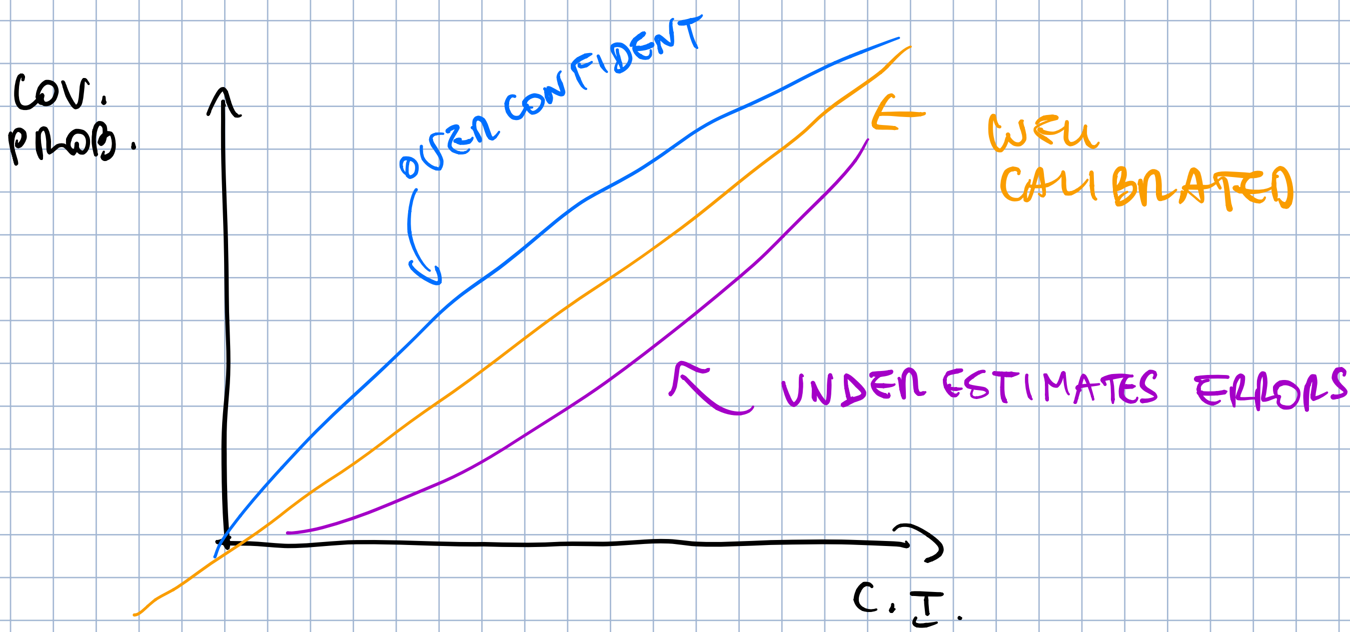
$$Var_q\left(p(y^*, x^*) \right) \simeq \underbrace{\frac{1}{N} \sum_i \sigma_i^2}_{\text{Aleatoric}} + \underbrace{\frac{1}{N} \sum (\mu_i - \bar{\mu}_i)^2}_{\text{Epistemic}}$$

?

CALIBRATION

1708.08843
1911.08508

- Get an uncertainty
- Coverage probability: fraction of test examples where the true value lies within a confidence interval

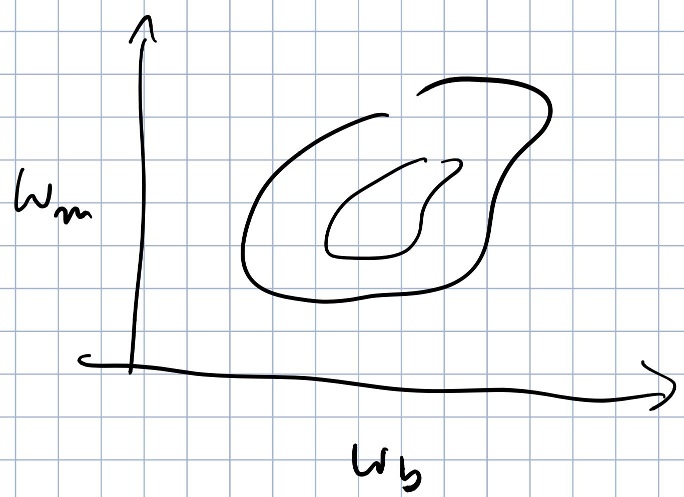


- Tune parameters to get correct result (e.g. Dropout rate)
- Methods for calibrating after training

PARAMETER ESTIMATION & MEANING OF "BAYESIAN"

Say now the last layer gives you some parameters (Ω_b, Ω_m, A_s)

Can draw samples :



- CONDITIONED ON DATA
- WHERE IS PRIOR?
- NOT "EXACT" POSTERIOR BUT q_θ