

NN parameter estimation for regression

- **Model:** $\mathbf{Y} = \hat{\mathbf{Y}} + \epsilon = f(\mathbf{X}; \boldsymbol{\theta}) + \epsilon$ where $f(\mathbf{X}; \boldsymbol{\theta})$ expresses our neural network

$$\mathbf{Y} = \hat{\mathbf{Y}} + \epsilon = f(\mathbf{X}; \theta) + \epsilon$$

where

$f(\mathbf{X}; \boldsymbol{\theta})$ expresses our neural network

$$\epsilon \sim N(\mathbf{Y} - f(\mathbf{X}; \boldsymbol{\theta}), \boldsymbol{\Sigma})$$

$$= \sqrt{\frac{1}{(2\pi)^N \det \Sigma}} e^{-\frac{1}{2}(\mathbf{X} - f(\mathbf{X}; \boldsymbol{\theta}))^T \Sigma^{-1}(\mathbf{X} - f(\mathbf{X}; \boldsymbol{\theta}))}$$

- Optimization: $\hat{\theta} = \underset{\theta}{\operatorname{argmin}} -\log P(\mathbf{Y} \mid \mathbf{X}; \theta)$

$$:= P(\mathbf{Y} | \mathbf{X}; \boldsymbol{\theta}) \leftarrow \mathcal{L}(\mathbf{D}; \boldsymbol{\theta})$$

$$= \operatorname{argmin}_{\boldsymbol{\theta}} -\log\left(\sqrt{\frac{1}{(2\pi)^N \det \boldsymbol{\Sigma}}} \exp\left[-\frac{1}{2}(\mathbf{X} - f(\mathbf{X}; \boldsymbol{\theta}))^T \boldsymbol{\Sigma}^{-1}(\mathbf{X} - f(\mathbf{X}; \boldsymbol{\theta}))\right]\right)$$

$$= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \quad -\frac{1}{2} \log((2\pi)^N \det \boldsymbol{\Sigma}) - \frac{1}{2} (\mathbf{X} - f(\mathbf{X}; \boldsymbol{\theta}))^T \boldsymbol{\Sigma}^{-1} (\mathbf{X} - f(\mathbf{X}; \boldsymbol{\theta}))$$

$$= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \quad - (\mathbf{X} - f(\mathbf{X}; \boldsymbol{\theta}))^T (\mathbf{X} - f(\mathbf{X}; \boldsymbol{\theta}))$$

$$= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \quad - \sum_{i=1}^M (\mathbf{y}^{(i)} - f(\mathbf{x}^{(i)}; \boldsymbol{\theta}))^T (\mathbf{y}^{(i)} - f(\mathbf{x}^{(i)}; \boldsymbol{\theta}))$$

$$= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} - \sum_{i=1}^M \sum_{j=1}^N (\mathbf{y}_j^{(i)} - f(\mathbf{x}^{(i)}; \boldsymbol{\theta}))_j^2 \quad \leftarrow \text{least squares}$$

→ Σ is diagonal, strictly positive, independent of \mathbf{x} ; it doesn't affect $\hat{\theta}$

NN parameter estimation for classification

- Model: $P(\mathbf{y} | \mathbf{x}; \boldsymbol{\theta})$ where $\mathbf{x}, \mathbf{y} \sim P_D$, $\mathbf{y} \in \{0,1\}^N$
- Optimization:
$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \operatorname{argmin}_{\boldsymbol{\theta}} \sum_{i=1}^M -\log P(\mathbf{y}_i | \mathbf{x}_i; \boldsymbol{\theta}) \\ &= \operatorname{argmin}_{\boldsymbol{\theta}} -\mathbb{E}_{\mathbf{x}, \mathbf{y} \sim P_D} [\log P(\mathbf{y} | \mathbf{x}; \boldsymbol{\theta})] \\ &= \operatorname{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x}, \mathbf{y} \sim P_D} \left[\log \frac{1}{P(\mathbf{y} | \mathbf{x}; \boldsymbol{\theta})} \right]\end{aligned}$$

one hot encoding



Cross entropy between data distribution and the model output distribution, $H(P_D(\mathbf{y} | \mathbf{x}), P(\mathbf{y} | \mathbf{x}; \boldsymbol{\theta}))$. Note, this holds for arbitrary PMFs (softmax, Bernoulli, etc...). Because the labels, \mathbf{y} , are one hot, they have zero entropy, and thus in this setting minimizing the cross entropy is equivalent to minimizing the KL divergence, $D_{KL}(P_D(\mathbf{y} | \mathbf{x}), P(\mathbf{y} | \mathbf{x}; \boldsymbol{\theta}))$.