TAM 598 Lecture 25:

Neural Networks, cont'd

Announcements:

- HW 7 covers lectures 24-26; due on Fri May 16th

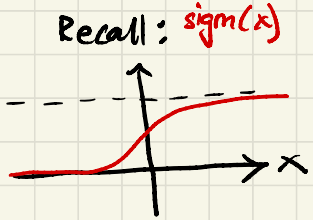
Classification using Neural Networks

features $\underline{x}_{1:n}$, discrete targets $y_{1:n}$, N.N. $f(\underline{x};\theta)$

I) Binary Classification $y=0$ or $y=1$

single point

$$p(y=1 \mid \underline{x},\theta) = \text{sigm}(f(\underline{x};\theta)) = \frac{e^{f(\underline{x};\theta)}}{1+e^{f(\underline{x};\theta)}}$$



$$p(y=0 \mid \underline{x},\theta) = 1-\text{sigm}(f(\underline{x};\theta))$$

likelihood

$$p(y \mid \underline{x},\theta) = [\text{sigm}(f(\underline{x};\theta))]^{y}\,[1-\text{sigm}(f(\underline{x};\theta))]^{1-y}$$

all points likelihood

$$p(\underline{y}_{1:n} \mid \underline{x}_{1:n},\theta) = \prod_{i=1}^{n} p(y_i \mid \underline{x}_i,\theta)$$

train network by maximizing log likelihood, ie minimizing the cross entropy loss

$$\ell(\theta) = -\log p(y_{1:n} | \underline{x}_{1:n}, \theta)$$

$$= -\sum_{i=1}^{n} \left\{ y_i \log \text{sigm}(f(\underline{x}_i; \theta)) + (1-y_i) \log [1 - \text{sigm}(f(\underline{x}_i; \theta))] \right\}$$

II) Multiclass Classification $y = 0, 1, \ldots, K-1$ K values

need: $p(y=k | \underline{x})$

use: NN $\underline{f}(\underline{x}; \theta)$ with K outputs

$$\underline{f}(\underline{x}; \theta) = (f_0(\underline{x}; \theta), \ldots, f_{K-1}(\underline{x}; \theta))$$ K scalars

Construct a probability vector:

$$p(y=k \mid \underline{x}, \theta) = \text{softmax}_k\left(\underline{f}(\underline{x}; \theta)\right) = \frac{\exp\left(f_k(\underline{x}, \theta)\right)}{\sum_{k'=0}^{K-1} \exp\left(f_k(\underline{x}, \theta)\right)}$$

Likelihood of full dataset

$$p(y_{1:n} \mid \underline{x}_{1:n}, \theta) = \prod_{i=1}^{n} p(y_i \mid \underline{x}, \theta)$$

Minimize the softmax cross entropy loss

$$\ell(\theta) = -\log p(y_{1:n} \mid \underline{x}_{1:n}, \theta)$$

$$= -\sum_{i=1}^{n} \log\left[\text{softmax}_{y_i}\left(\underline{f}(\underline{x}_i; \theta)\right)\right]$$

III) Regularization

since NN's are highly flexible $\Rightarrow$ prone to overfitting

often minimize

$$J(\theta) = L(\theta) + \lambda R(\theta)$$

λ: positive param

$R(\theta)$: penalizes big weights, eg

$$R(\theta) = \sum_j \theta_j^2$$

IV) Bayesian interpretation of regularization - regularization can be seen as a maximum posterior approach, given prior $p(\theta)$

$$\underbrace{p(\theta \mid \underline{x}_{1:n}, y_{1:n})}_{\text{maximize this}} \propto p(y_{1:n} \mid \underline{x}_{1:n}, \theta)\, p(\theta)$$

equivalent to minimize loss

$$J(\theta) = \underbrace{-\log p(y_{1:n} \mid x_{1:n}, \theta)}_{L(\theta)} \underbrace{- \log p(\theta)}_{\lambda R(\theta)}$$

examples: CNNs → process grid like data. Used in image recog, etc. extract spatial features. apply filters that detect edges, shapes...

eg) prior $p(\theta)$ = gaussian w/ mean $\mu = 0$, variance $1/\alpha$

$$= N(\theta \mid 0, \alpha^{-1} I)$$

then $-\log p(\theta) = \frac{\alpha}{2\pi} \sum_j \theta_j^2 + \text{const}$

ie) L2 regularizer