SGD for Probabilistic Neural Networks

As in the previous lecture, we consider probabilistic neural networks

$$p(x^{1\dots\ell} \mid x^0; \theta) = \prod_{k=1}^{\ell} p(x^k \mid x^{k-1}; \theta^k)$$

with

$$p(x^k \mid x^{k-1}; \theta^k) = \prod_{i=1}^{n_k} p(x_i^k \mid x^{k-1}; \theta^k)$$

The network „head function" denotes the remainder not including stochastic terms, which is denoted by $f(x^\ell; \theta^{\ell+1})$. It can be combined of an affine transform and e.g. the softmax + cross entropy loss.

Our central problem is to compute

$$\nabla_\theta \mathbb{E}_x[f(x^\ell; \theta)] = \nabla_\theta \sum_{x^1,\dots,x^\ell} p(x^{1\dots\ell} \mid x^0; \theta) f(x^\ell; \theta)$$

A. Robbins-Monro (1951)

Consider the task

$$F(\theta) = \mathbb{E}_{x\sim p(x)}[f(x, \theta)] \to \min_\theta$$

where $f(x, \theta)$ is convex in θ for each x.

Let $g(\theta, x)$ be an unbiased estimator of $\nabla_\theta F(\theta)$ i.e.

$$\mathbb{E}_x[g(\theta, x)] = \nabla_\theta F(\theta)$$

if e.g. f is differentiable in $\theta \Rightarrow$
$g(\theta, x) = \nabla_\theta f(\theta, x)$ is an unbiased estimator.

Algorithm for minimising F:

$$\theta_{t+1} = \theta_t + \alpha_t g(\theta_t, x_{t+1}), \quad \text{where } x_{t+1} \sim p(x)$$

It converges to the minimum of F if α_t are chosen so that $\sum_t \alpha_t = \infty$, $\sum_t \alpha_t^2 < \infty$.

Nice result, but in our case $x \sim p(x;\theta)$!

B. Score function

Consider the task

$$F(\theta) = \mathbb{E}_{x \sim p(x;\theta)}[f(x,\theta)] \to \min_\theta$$

We can construct a stochastic gradient estimator by

$$\nabla_\theta \mathbb{E}_{x \sim p(x,\theta)}[f(x,\theta)] = \nabla_\theta \sum_x p(x;\theta) f(x,\theta)$$

$$= \sum_x p(x;\theta) \nabla_\theta f(x,\theta) + \sum_x f(x,\theta) \nabla_\theta p(x;\theta)$$

$$= \sum_x p(x;\theta) \left[\nabla_\theta f(x,\theta) + f(x,\theta) \nabla_\theta \log p(x;\theta) \right]$$

i.e.

$$g(\theta, x) = \nabla_\theta f(x,\theta) + f(x,\theta) \nabla_\theta \log p(x;\theta)$$

It is unbiased, but has <u>high variance</u> because the second term does not utilise the gradient of f.

C. Pathwise gradient estimators

Consider a continuous r.v. $x \sim p(x;\theta)$ for which $\exists$ a mapping $x = t(\varepsilon, \theta)$ such that

$$\varepsilon \sim p(\varepsilon) \Rightarrow x = t(\varepsilon, \theta) \sim p(x;\theta)$$

Then we can write

$$\mathbb{E}_{x\sim p(x;\theta)}[f(x)] = \mathbb{E}_{\varepsilon\sim p(\varepsilon)}[f(t(\varepsilon,\theta))]$$

and it follows

$$\nabla_\theta \mathbb{E}_{x\sim p(x;\theta)}[f(x)] = \mathbb{E}_{\varepsilon\sim p(\varepsilon)}[\nabla_\theta f(t(\varepsilon,\theta))]$$

Example 1 $\quad x \sim \mathcal{N}(\mu, \sigma^2), \quad \varepsilon \sim \mathcal{N}(0,1)$

$$x = \sigma\varepsilon + \mu$$

$$\nabla_\mu \mathbb{E}_{\mathcal{N}(\mu,\sigma^2)}[f(x)] = \mathbb{E}_{\mathcal{N}(0,1)}[\nabla_\mu f(\sigma\varepsilon+\mu)]$$

$$\nabla_\sigma \mathbb{E}_{\mathcal{N}(\mu,\sigma^2)}[f(x)] = \mathbb{E}_{\mathcal{N}(0,1)}[\nabla_\sigma f(\sigma\varepsilon+\mu)]$$

Pathwise gradients have small variance.
However, they can not be used if x are binary and/or f is not differentiable

D. Stochastic binary neurons

For probabilistic networks with binary units we have to trade in bias for small variance.

E.g. straight through gradient estimator:

Let $y = H(a+\varepsilon)$ be a binary neuron with, H - Heaviside step function, a - pre-activation and ε - injected noise with c.d.f. F_ε. Then

$$\frac{d}{da}\int f(H(a+\varepsilon))\,p(\varepsilon)\,d\varepsilon \overset{?}{=}$$

$$= \int f'(H(a+\varepsilon))\left(\frac{dH(a+\varepsilon)}{da}\right)p(\varepsilon)\,d\varepsilon$$

??

$$\approx \int f'(H(a+\varepsilon))\,\frac{d}{da}F_\varepsilon(a)\,p(\varepsilon)\,d\varepsilon$$

I.e.

- in forward pass → sample neuron outputs
- in backpropagation → use derivative of

$$\nabla_\theta \mathbb{E}_\varepsilon[H(a(\theta)+\varepsilon)] = \nabla_\theta F_\varepsilon(a(\theta))$$