TTIC 31230 Fundamentals of Deep Learning

Quiz 3

Problem 1 (25 points). Consider the following running update equation.

$$\begin{aligned} y_0 &= 0 \\ y_t &= \left(1 - \frac{1}{N}\right) y_{t-1} + x_t \end{aligned}$$

(a) If the input sequence is constant, i.e., if $x_t = c$ for all $t \geq 1$, what is $\lim_{t \to \infty} y_t$?

Solution:
The limit y must satisfy

$$y = \left(1 - \frac{1}{N}\right) y + c$$

giving $y = Nc$.

(b) y_t is a running average of what quantity?

Solution: The update can be rewritten as

$$y_t = \left(1 - \frac{1}{N}\right) y_{t-1} + \frac{1}{N}(Nx_t)$$

so y_t is the running average of Nx_t.

(c) Express y_t as a function of μ_t where μ_t is defined by

$$\begin{aligned} \mu_0 &= 0 \\ \mu_t &= \left(1 - \frac{1}{N}\right) \mu_{t-1} + \frac{1}{N} x_t \end{aligned}$$

Solution: y_t is the running average of Nx_t which equals N times the running average of x_t so we have

$$y_t = N\mu_t$$

Problem 2 (25 points). Consider any probability distribution $P(h)$ over an discrete class $\mathcal{H}$. Assume $0 \leq \mathcal{L}(h, x, y) \leq L_{\max}$. Define

$$\begin{aligned} \mathcal{L}(h) &= E_{(x,y) \sim \text{Pop}}\, \mathcal{L}(h, x, y) \\ \hat{\mathcal{L}}(h) &= E_{(x,y) \sim \text{Train}}\, \mathcal{L}(h, x, y) \end{aligned}$$

We now have the theorem that with probability at least $1-\delta$ over the draw of training data the following holds simultaneously for all h.

$$\mathcal{L}(h) \leq \frac{10}{9}\left(\hat{\mathcal{L}}(h) + \frac{5L_{\max}}{N}\left(\ln\frac{1}{P(h)} + \ln\frac{1}{\delta}\right)\right) \quad (1)$$

This motivates

$$h^* = \operatorname*{argmin}_h \; \hat{\mathcal{L}}(h) + \frac{5L_{\max}}{N}\ln\frac{1}{P(h)} \quad (2)$$

The Bayesian maximum a-posteriori (MAP) rule is

$$h^* = \operatorname*{argmax}_h \; P(h) \prod_{(x,y)\in \mathrm{Train}} P(y|x,h) \quad (3)$$

For $\mathcal{L}(h,x,y) = -\ln P(y|x,h)$ (cross entropy loss) rewrite (2) so as to be as similar to (3) as possible. Keep in mind that

$$\hat{\mathcal{L}}(h) = \frac{1}{N}\sum_{(x,y)\in \mathrm{Train}} -\ln P(y|x,h)$$

Solution:

$$\begin{aligned}
& \operatorname*{argmin}_h \left(\frac{1}{N}\sum_{(x,y)\sim \mathrm{Train}} -\ln P(y|x,h)\right) + \frac{5L_{\max}}{N}\ln\frac{1}{P(h)} \\
= \; & \operatorname*{argmax}_h \left(\frac{1}{N}\sum_{(x,y)\sim \mathrm{Train}} \ln P(y|x,h)\right) + \frac{5L_{\max}}{N}\ln P(h) \\
= \; & \operatorname*{argmax}_h \left(\sum_{(x,y)\sim \mathrm{Train}} \ln P(y|x,h)\right) + 5L_{\max}\ln P(h) \\
= \; & \operatorname*{argmax}_h \ln\left(P(h)^{5L_{\max}} \prod_{(x,y)\sim \mathrm{Train}} P(y|x,h)\right) \\
= \; & \operatorname*{argmax}_h \; P(h)^{5L_{\max}} \prod_{(x,y)\sim \mathrm{Train}} P(y|x,h)
\end{aligned}$$

Problem 3 (25 points).

(a) Consider a model with d parameters each of which is represented by a 32 bit floating point number. Express the bound (1) in problem 2 in terms of the dimension d assuming all representable parameter vectors are equally likely.

Solution:
$$\mathcal{L}(h) \leq \frac{10}{9}\left(\hat{\mathcal{L}}(h) + \frac{5L_{\max}}{N}\left(32d\ln 2 + \ln\frac{1}{\delta}\right)\right)$$

(b) Repeat part (a) but for a model with d parameters represented by $\Phi_i = z[J[i]]$ where $J[i]$ is an integer index with $0 \leq J[i] < 32$ and where $z[j]$ is a 32 bit floating point number and where all parameter vectors are equally likely.

Solution:

$$\mathcal{L}(h) \leq \frac{10}{9}\left(\hat{\mathcal{L}}(h) + \frac{5L_{\max}}{N}\left((32^2 + 5d)\ln 2 + \ln\frac{1}{\delta}\right)\right)$$

Problem 4 (25 points). This problem is on dynamic programming for hidden Markov models (HMMs). Assume we have an input sequence $x_1, \ldots, x_T$ and a phoneme gold label $y_1, \ldots, y_T$ with $y_t \in \mathcal{P}$. This problem is simpler than CTC because the gold label has the same length as the input sequence.

In an HMM we assume a hidden state sequence $s_1, \ldots, s_T$ with $s_t \in \mathcal{S}$ where $\mathcal{S}$ is some finite sets of "hidden states". Here will assume that then some deep network has computed transition probabilities and emission probabilities.

$$P_{\text{Trans}}(s_{t+1} \mid s_t)$$

$$P_{\text{Emit}}(y_t \mid s_t)$$

We assume an initial state s_{init} and a stop state s_{stop} such that $s_1 = s_{\text{init}}$ (before emitting any phonemes). The length T is determined by when the hidden state becomes s_{stop} giving $s_{T+1} = s_{\text{stop}}$.

For a given gold sequence $y_1, \ldots, y_T$ we define a "forward tensor" as

$$F[t, s] = P(y_1, \ldots, y_{t-1} \ \wedge s_t = s)$$

We have

$$\begin{aligned} F[1, s_{\text{init}}] &= 1 \\ F[1, s] &= 0 \quad \text{for } s \neq s_{\text{init}} \end{aligned}$$

(a) Write a dynamic programming equation to compute $F[t, s]$ from $F[t-1, s']$ for various values of s'.

Solution:
$$F[t, s] = \sum_{s'} F[t-1, s'] P_{\text{Emit}}(y_{t-1}|s') P_{\text{Trans}}(s|s')$$

(b) Express $P(y_1, \ldots, y_T)$ in terms of $F[t, s]$.

Solution:
$$P(y_1, \ldots y_T) = F[T+1, s_{\text{stop}}]$$

(c) Explain why, if the forward equations are written in a framework, we do not need to also implement "backward" equations to compute

$$B[t, s] = P(y_t, \ldots, y_T \mid s_t = s).$$

Solution: Once we have expressed the loss $-\ln P(y_1, \ldots, y_T)$ in a framework we can train the model by SGD using the framework's implementation of backpropagation. Nothing more is needed.