TTIC 31230 Fundamentals of Deep Learning
Quiz 3

Problem 1: Generalization Bounds and The Lottery Ticket Hypothesis. Suppose that we want to construct a linear classifier (a linear threshold unit) for binary classification defined by

$$\hat{y}_\alpha(x) = \begin{cases} 1 & \text{if } \sum_{i=1}^{d} \alpha_i f_i(x) >= 0 \\ \\ -1 & \text{otherwise} \end{cases}$$

where each α_i is a scalar weight, $f_i(x)$ is a scalar value, and the functions f_i are (random) features constructed independent of any observed values of x or y.
We will assume a population distribution Pop of pairs $\langle x, y \rangle$ with $y \in \{-1, 1\}$ and a training set Train of N pairs drawn IID from pop.
We can define both test and train losses (error rates).

$$\begin{aligned} \hat{\mathcal{L}}(\alpha) &= E_{x,y\sim\text{Train}}\ \mathbf{1}[\hat{y}_\alpha(x_i) \neq y_i] \\ \\ \mathcal{L}(\alpha) &= E_{x,y\sim\text{Pop}}\ \mathbf{1}[\hat{y}_\alpha(x_i) \neq y_i] \end{aligned}$$

Assume finite precision arithmetic so that we have discrete rather than continuous possible values of α. The course slides state that for any (prior) distribution P on the values of α we have that with probability at least $1 - \delta$ over the draw of the training data the following holds simultneously for all α.

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

We will now incorporate the lottery ticket hypothesis into the prior distribution on α by assuming that low training error can be achieved with some small subset of the random features. More formally, we define a prior favoring sparse α — cases where most weights are zero.

(a) To define $P(\alpha)$, first define a prior probability distribution $P(s)$ over the number s of nonzero values.

Solution: There are of course many solutions. A uniform distribution on the numbers from 1 to d will work giving $P(s) = 1/d$. Another possibility is $P(s) = \epsilon(1-\epsilon)^s$ which defines a distribution on all $s \geq 0$.

(b) Given a specified number s of nonzero values, define a probability distribution $P(U|s)$ where U is a subset of the random features with $|U| = s$.

Solution: A reasonable choice here is a uniform distribution on the $\binom{d}{s}$ possibilities giving $P(U|s) = 1/\binom{d}{s}$.

(c) Assuming that each nonzero value is represented by b bits, give a probability distribution over $P(\alpha|U, s)$.

Solution: Here we can use the uniform distribution on the 2^{bs} ways of assigning numbers to the s nonzero weights in α giving $P(\alpha|U,s) = P(\alpha|U) = 2^{-bs}$.

(d) Combine (a), (b) and (c) to define $P(\alpha)$.

Solution: Under $P(s) = 1/d$ we get $P(\alpha) = \frac{1}{d\binom{d}{s}2^{bs}}$ and using $\binom{d}{s} \leq d^s$ we get $P(\alpha) \geq \frac{1}{dd^s 2^{bs}} = \frac{1}{d^{s+1}2^{bs}}$.

Under $P(s) = \epsilon(1-\epsilon)^s$ we get $P(\alpha) = \frac{\epsilon(1-\epsilon)^s}{\binom{d}{s}2^{bs}}$ and using $\binom{d}{s} \leq d^s$ $P(\alpha) \geq \frac{\epsilon(1-\epsilon)^s}{d^s 2^{bs}}$

(e) Plug your answer to (c) into the above generalization bound to get a bound in terms of the number of random features d, the number s of nonzero values of α, and the number b of bits used to represent each nonzero value and any additional parameters used in defining your distributions.

Solution: Under $P(s) = 1/d$ we get

$$\begin{aligned} \mathcal{L} &\leq \frac{10}{9}\left(\hat{\mathcal{L}} + \frac{5}{N}\left(\ln d + \ln \binom{d}{s} + sb\ln 2 + \ln\frac{1}{\delta}\right)\right) \\ &\leq \frac{10}{9}\left(\hat{\mathcal{L}} + \frac{5}{N}\left((s+1)\ln d + sb\ln 2 + \ln\frac{1}{\delta}\right)\right) \end{aligned}$$

Under $P(s) = \epsilon(1-\epsilon)^s$ we get

$$\begin{aligned} \mathcal{L} &\leq \frac{10}{9}\left(\hat{\mathcal{L}} + \frac{5}{N}\left(\ln\frac{1}{\epsilon} + s\ln\frac{1}{1-\epsilon} + \ln\binom{d}{s} + sb\ln 2 + \ln\frac{1}{\delta}\right)\right) \\ &\leq \frac{10}{9}\left(\hat{\mathcal{L}} + \frac{5}{N}\left(\ln\frac{1}{\epsilon} + s\ln\frac{1}{1-\epsilon} + s\ln d + sb\ln 2 + \ln\frac{1}{\delta}\right)\right) \end{aligned}$$

Note that in either case the bound is logarithmic in d allowing d to be extremely large. The choice of the uniform distribution for s is simpler and gives a completely satisfactory result. However there are regimes in which the second prior on s is very slightly better.

Problem 2. Computing the Partition Function for a Chain Graph. Consider a graphical model defined on a sequence of nodes $n_1, \ldots, n_T$. We are interested in "colorings" $\hat{\mathcal{Y}}$ which assign a color $\hat{\mathcal{Y}}[n]$ to each node n. We will use y to range over the possible colors. Suppose that we assign a score $s(\hat{\mathcal{Y}})$ to each coloring defined by

$$s(\hat{\mathcal{Y}}) = \left(\sum_{t=1}^{T} S^N[t, \hat{\mathcal{Y}}[n_t]]\right) + \left(\sum_{t=1}^{T-1} S^E[t, \hat{\mathcal{Y}}[n_t], \hat{\mathcal{Y}}[n_{t+1}]]\right)$$

In this problem we derive an efficient way to exactly compute the partition function

$$Z = \sum_{\hat{\mathcal{Y}}} e^{s(\hat{\mathcal{Y}})}.$$

Let $\hat{\mathcal{Y}}_t$ range over colorings of $n_1, \ldots n_t$ and define the score of $\hat{\mathcal{Y}}_t$ by

$$s(\hat{\mathcal{Y}}_t) = \left(\sum_{s=1}^{t} S^N[s, \hat{\mathcal{Y}}[n_s]]\right) + \left(\sum_{s=1}^{t-1} S^E[s, \hat{\mathcal{Y}}_t[n_s], \hat{\mathcal{Y}}_t[n_{s+1}]]\right)$$

Now define $Z_t(y)$ by

$$\begin{aligned} Z_1(y) &= e^{S^N[1,y]} \\ Z_{t+1}(y) &= \sum_{\hat{\mathcal{Y}}_t} e^{s(\hat{Y}_t)} e^{S^E[t,\hat{\mathcal{Y}}_t[n_t],y]} e^{S^N[t+1,y]} \end{aligned}$$

(a) Give dynamic programming equations for computing $Z_t(y)$ efficiently. You do not have to prove that your equations are correct — just writing the correct equations gets full credit.

Solution:

$$\begin{aligned} Z_1(y) &= e^{S^N[1,y]} \\ Z_{t+1}(y) &= e^{S^N[t+1,y]} \sum_{y'} Z_t(y') e^{S^E[t,y',y]} \end{aligned}$$

(b) show that $Z = \sum_y Z_T(y)$

Solution:

$$\begin{aligned} \sum_y Z_T(y) &= \sum_y \sum_{\hat{\mathcal{Y}}_{T-1}} e^{s(\hat{Y}_{T-1})} e^{S^E[t,\hat{\mathcal{Y}}_t[n_t],y]} e^{S^N[t+1,y]} \\ &= \sum y \sum_{\hat{\mathcal{Y}}} e^{s(\hat{\mathcal{Y}}[n_T=y])} \\ &= \sum_{\hat{\mathcal{Y}}} e^{s(\hat{\mathcal{Y}})} \\ &= Z \end{aligned}$$

Problem 3. Reshaping Noise in GANs. A GAN generator is typically given a random noise vector $z \sim \mathcal{N}(0, I)$. Give equations defining a method for computing z' from z such that the distribution on z' is a mixture of two Gaussians each with a different mean and diagonal covariance matrix. Hint: use a step-function threshold on the first component of z to compute a binary value and use the other components of z to define the Gaussian variables.

Solution:

$$
\begin{aligned}
y &= \mathbf{1}[z[0] \geq 0] \\
\\
z' &= y(\mu_1 + \sigma_1 \odot z[1:d]) + (1-y)(\mu_2 + \sigma_2 \odot z[1:d])
\end{aligned}
$$