TTIC 31230 Fundamentals of Deep Learning, Autumn 2021

Exam 2

Problem 1: 25 pts. This problem is on interaction of learning rate and scaling of the loss function.

(a) Consider vanilla SGD on cross entropy loss for classification with batch size 1 and no moment in which case we have

$$\Phi_{t+1} = \Phi_t - \eta \nabla_\Phi \ln P_\Phi(y|x)$$

Now suppose someone uses log base 2 (to get loss in bits) and uses the update

$$\Phi_{t+1} = \Phi_t - \eta' \nabla_\Phi \log_2 P_\Phi(y|x)$$

Suppose that we find that leatning rate η works well for the natural log version (with loss in nats). What value of η' should be used in the second version with loss measured in bits? You can use the relation that $\log_b z = \ln z / \ln b$.

Solution: We have

$$\begin{aligned} -\Delta\Phi &= \eta' \nabla_\Phi \log_2 P(\Phi) \\ &= \eta' \nabla_\Phi \ln P(\Phi) / \ln 2 \\ &= \frac{\eta'}{\ln 2} \nabla_\Phi \ln P(\Phi) \end{aligned}$$

To make the two updates the same we set $\eta' = \eta \ln 2$

(b) Now consider the following simplified version of RMSprop where for each parameter $\Phi[i]$ we have

$$\Phi_{t+1}[i] = \Phi_t[i] - \frac{\eta}{\sigma_i} \nabla_\Phi \mathcal{L}_\Phi(x_t, y_t)$$

where σ_i is exactly the standard deviation of ith component of the gradient as defined by

$$\begin{aligned} \mu_i &= E_{x,y}\left[\nabla_{\Phi[i]}\, \mathcal{L}_\Phi(x,y)\right] \\ \sigma_i &= \sqrt{E_{x,y}\left[\left(\nabla_{\Phi[i]}\, \mathcal{L}_\Phi(x,y) - \mu_i\right)^2\right]} \end{aligned}$$

If we replace $\mathcal{L}$ by $2\mathcal{L}$ what learning rate η' should we use with loss $2\mathcal{L}$ to get the same temperature?

Solution: If we double the loss function we also double σ_i and we have $\eta' = \eta$. For RMSprop we get that the learning rate is (approximately) invariant to scaling the loss function. It is not clear whether this has any significance.

Problem 2. 25 pts This problem is on a non-standard form of adaptive learning rates. In general when we consider the significance of a change Δx to a number x it is reasonable to consider the change as a percentage of x. For example, a baseline annual raise in salary is often a percentage raise when different employees have significantly different salaries. So we might consider the following "multiplicative update SGD" which we will write here for batch size 1.

$$\Phi^{t+1}[i] = \Phi^t[i] - \eta \ \max(\epsilon, |\Phi^t[i]|) \ \ \hat{g}(\Phi, x_t, y_t)[i] \tag{1}$$

where $\hat{g}(\Phi, x, y)$ abbreviates the gradient $\nabla_\Phi \mathcal{L}(\Phi, x, y)$ where $\mathcal{L}(\Phi, x, y)$ is the loss for the training point (x, y) at parameter setting Φ, and where and $\hat{g}(\Phi, x, y)[i]$ is the ith component of the gradient. For $|\Phi^t[i]| >> \epsilon$ this is a multiplicative update. Multiplicative updates have a long history and rich theory for mixtures of experts prior to the deep revolution. However, I do not know of a citation for the above multiplicative variant of SGD (let me know if you find one later). The parameter ϵ allows a weight to flip sign — to pass through zero more easily. Recall that a stationary point is a parameter setting where the total gradient is zero.

$$\sum_{(x,y)\sim \text{Train}} \nabla_\Phi \ \mathcal{L}(x, y) = 0 \tag{2}$$

(a) At a stationary point of the loss function, is the expected update of equation (1) over a random draw of (x_t, y_t) always equal to zero. In other words, is a stationary point of the loss function also a stationary point of the update equation?

Solution: Yes, a stationary point of the loss function is also a stationary point of the update equation.

$$\begin{aligned} & E_{(x,y)\sim\text{Train}} \ \ \eta \ \min(\epsilon, |\Phi^t[i]|) \ \left(\nabla_\Phi \ \mathcal{L}(\Phi, x, y)\right)[i] \\ = \ & \eta \ \min(\epsilon, |\Phi[i]|) \ E_{(x,y)\sim\text{Train}} \left(\nabla_\Phi \mathcal{L}(\Phi, x, y)\right)[i] \\ = \ & 0 \end{aligned}$$

(b) Consider an adaptive algorithm which makes the update proportional to the loss. i.e.,

$$\Phi^{t+1} = \Phi^t - \eta \ \mathcal{L}(\Phi, x_t, y_t) \ \hat{g}^t \tag{3}$$

Is a stationary point of the loss function always a stationary point of the update defined by (3)? Justify your answer.

You can assume that there exists a training set of two points (x_1, y_1) and (x_2, y_2) and a stationary point of the loss function Φ with $\mathcal{L}(\Phi, x_1, y_1) \neq \mathcal{L}(\Phi, x_2, y_2)$ and $\nabla_\Phi(\Phi, x_1, y_1) \neq \nabla_\Phi(\Phi, x_2, y_2)$.

Solution: No, the expected update can be non-zero at a stationary point of the loss function. Weighing the updates by something that depends on the draw of (x, y) effectively changes the weighting on the training points which changes the stationarity condition. Writing this in English counts as a correct solution. A formal counter example can be given using the assumed conditions:

$$\begin{aligned}
& E_{(x,y)\sim\text{Train}}\;\; \eta\, \mathcal{L}(\Phi, x, y)\;\; \nabla_\Phi\; \mathcal{L}(\Phi, x, y) \\
= \;\; & \eta\, \frac{1}{2} \left(\mathcal{L}(\Phi, x_1, y_1)\;\; (\nabla_\Phi\; \mathcal{L}(\Phi, x_1, y_1)) + \mathcal{L}(\Phi, x_2, y_2)\;\; (\nabla_\Phi\; \mathcal{L}(\Phi, x_2, y_2))\right) \\
= \;\; & \eta\, \frac{1}{2} \left(\mathcal{L}_1\;\; (\nabla_\Phi\; \mathcal{L}(\Phi, x_2, y_2)) + \mathcal{L}_2\;\; (\nabla_\Phi\; \mathcal{L}(\Phi, x_2, y_2))\right) \\
= \;\; & \eta(\mathcal{L}_1 + \mathcal{L}_2)\, \frac{1}{2} \left(\frac{\mathcal{L}_1}{\mathcal{L}_1 + \mathcal{L}_2}\;\; (\nabla_\Phi\; \mathcal{L}(\Phi, x_2, y_2)) + \frac{\mathcal{L}_2}{\mathcal{L}_1 + \mathcal{L}_2}\;\; (\nabla_\Phi\; \mathcal{L}(\Phi, x_2, y_2))\right) \\
\neq \;\; & \eta\; (\mathcal{L}_1 + \mathcal{L}_2) \frac{1}{2} (\; \nabla_\Phi\; \mathcal{L}(\Phi, x_1, y_1) + \nabla_\Phi\; \mathcal{L}(\Phi, x_2, y_2)) \\
= \;\; & 0
\end{aligned}$$

In Adam and RMSProp we have a weighting that depends on a moving average of the second moment of the gradients. This is essentially a weighting that depends on a random draw over the training data. It has been shown that stationary points of Adam and RMSProp updates do not necessarily correspond to stationary points of the loss function.

Problem 3. (25 pts) This problem is on robust loss functions. With a robust loss one identifies "outliers" in the data and "gives up" on modeling the outliers. In particular we can consider the following bounded version of cross-entropy loss

$$\begin{aligned}
\mathcal{L}(\Phi, x, y) &= L_{\max}\;\tanh\left(\frac{-\ln P_\Phi(y|x)}{L_{\max}}\right) \\
\tanh(z) &= \frac{2}{1 + e^{-2z}} - 1.
\end{aligned}$$

For $z \geq 0$ we have $\tanh(z) \geq 0$ and we have that the above robust loss is non-negative and can never be larger than $L_{\max}$.

(a) Consider the function $L_{\max}\;\tanh(\frac{z}{L_{\max}})$. Use a first order Taylor expansion of the tanh function about zero to show that for $|z| << L_{\max}$ we have

$$L_{\max}\;\tanh\left(\frac{z}{L_{\max}}\right) \approx z$$

This implies that the robust cross entropy loss is essentially equal to the cross entropy loss when the cross entropy loss is small compared to $L_{\max}$.

Solution: The first order Taylor expansion of the tanh function about zero is

$$\tanh(u) \approx u$$

yielding the desired result.

(b) Consider the case where the cross-entropy loss is large compared to $L_{\max}$. For $z >> 1$ we have that the derivative $\tanh'(z)$ is essentially zero. What parameter update is made on a training point whose cross entropy loss is large compared to $L_{\max}$ if we model $\tanh'(z) = 0$ in such cases.

Solution: The update on a data point (x, y) is

$$\Phi^{t+1} = \Phi^t - \eta \nabla_\Phi \mathcal{L}(\Phi, x, y)$$

At a point where the derivative of the sigmoid is essentially zero this update will be essentially zero. So "outliers" do not effect the model parameters.

(c) Look up the PAC-Bayesian generalization guarantee that is stated in terms of the L_2 norm of the weight vector. Explain why the robust loss function comes with a better PAC-Bayesian generalization guarantee. Intuitively, the improvement in generalization is due to insensitivity to "outliers" (or things the model cannot understand).

Solution: The L_2 PAC-Bayeisan guarantee in the notes is

$$\mathcal{L}_\sigma(\Phi) \leq \frac{10}{9}\left(\hat{\mathcal{L}}_\sigma(\Phi) + \frac{5L_{\max}}{N}\left(\frac{||\Phi - \Phi_{\mathrm{init}}||^2}{2\sigma^2} + \ln\frac{1}{\delta}\right)\right)$$

Reducing $L_{\max}$ both reduces $\hat{L}_\sigma(\Phi)$ and reduces the penalty for the model complexity (the norm squared of the distance from the initialization).

(d) Curriculum learning is the idea that one first learns how to solve easy problems and then gradually learns ever harder problems. At a high informal level describe a learning algorithm based on the above robust loss function which can be intuitively motivated as curriculum learning.

Solution: Easier problems should correspond to cases where the cross entropy loss can be made small. So setting $L_{\max}$ to be smallish will focus the model on the easy problems while ignoring the hard problems. Gradually increasing $L_{\max}$ will gradually pay more attention to the harder problems.

Another possible answer is that holding $L_{\max}$ fixed will focus first on the easy problems ignoring the hard problems but as the understanding of easy problems improves the harder problems become easy problems and we automatically gradually pay attention to harder and harder problems even with a fixed value of $L_{\max}$.

Problem 4. **(25 pts)** This problem is on PAC-Bayes bounds for classifiers built on CLIP using **prompt engineering**. CLIP is a joint probability model on images and English descriptions (image captions). Clip is trained on a large corpus of captioned images drawn from the web and defines a probability distribution over captions c given an image x. We can use CLIP for image classification (as in ImageNet) using "prompt engineering". A "prompt" is caption specific to an image label. For example the caption "this is an image of a cat" for the label "cat" or "this is an image of a dog" for the label "dog". For each image class y we have a prompt (hypothetical caption) $c(y)$. We can then label an image x with class $\hat{y}$ using the rule

$$\hat{y}(x) = \underset{y}{\text{argmax}}\; P_{\text{CLIP}}(c(y)|x)$$

Suppose that we search (somehow) over the captions $c(y_1), \ldots, c(y_n)$ assigned to the n image classes $y_1, \ldots, y_n$ to find a set of captions minimizing the error rate (0-1 loss) on a set of N labeled training images. Let $\hat{\mathcal{L}}$ be the error rate on the training data. Also suppose that CLIP assigns a prior probability $P_{\text{CLIP}}(c)$ to any caption c independent of any image. Consider the PAC-Bayes bound on generalization loss for predictive rule h where the bound is guaranteed to hold for all h with probability at least $1 - \delta$.

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

Apply this rule to the CLIP image classifier using CLIP's "prior probability" on the caption space.

Solution:

$$\mathcal{L}(h) \leq \frac{10}{9}\left(\hat{\mathcal{L}}(h) + \frac{5}{N}\left(\left(\sum_y -\ln P_{\text{CLIP}}(c(y))\right) + \ln\frac{1}{\delta}\right)\right)$$

I am not proposing that searching over all captions is a good idea. Some narrower prior is called for.