Homework 3

Hijun (Jane) Seo
1001423284

Q1 : Robustness & Regularization

1.1.2 $\nabla_x f(x;w) = w$

$x' = x - \epsilon w$

$\therefore \; f(x';w) = w^T x' = w^T(x - \epsilon w)$

1.2.2 Gradient of loss $= \frac{1}{n} X^T (Xw^* - t) + 2\lambda w^* = 0$

$X^T X w^* - X^T t + 2\lambda n w^* = 0$

$(X^T X + 2\lambda n I) w^* = X^T t$

$\therefore \; w^* = (X^T X + 2\lambda n I)^{-1} X^T t$

1.2.3 $f(x';w^*) = w^{*T} x - \epsilon w^{*T} w^* = 0$

*Note: w^*_{ridge} written as w^*

$\epsilon = w^{*T} x \, (w^{*T} w^*)^{-1}$

If considering 1-D model,

$\epsilon = \frac{x}{w^*} = \frac{x(\mathbb{X}^2 + 2\lambda n)}{\mathbb{X} t}$

*note: x = input, $\mathbb{X}$ = design matrix

Weight decay makes the model more robust since the addition of the $2\lambda n$ term makes ϵ bigger than it would be for plain regression

Q2: Trading off Resources in Neural Net Training

2.1.1 a) As batch size increases, so does optimal learning rate.
The larger the batch size, the less the gradient noise dominates.
Thus, with larger batch size, we can converge with fewer steps
$\therefore$ optimal learning rate increases.

2.1.2 a) C is the point that represents optimal batch size for data parallelism. This is because C is the point where batch size has been increased enough to reduce the total # of training steps, but is small enough that it isn't within the region of curvature where there is less benefit from data parallelism.

b) Point A: Regime = noise-dominated
Potential way to accelerate training = seek parallel compute

Point B: Regime = curvature dominc [illegible]
Potential way to accelerate training = use higher order optimizers

2.2. Figure 4 shows that training the same model with the same batch size for more steps doesn't lead to a huge decrease in the test loss as the curves seem "saturated".
Figure 3 (left) also shows that test loss does not change much after a certain # of tokens processed, so this information & information from Figure 4 suggests that there is a critical batch size (b_{crit}) & batches greater than b_{crit} don't lead to a significant decrease in test loss. Thus, the best option is to increase model size since larger models lead to steeper decrease in test loss with # of tokens processed & # of training steps &, with high amount of compute, can reach lowest test losses.

Q3 : Dropout as Gaussian noise

3.2 From lecture 6, bias-variance decomposition gives

$$\mathbb{E}_\pi[\tilde{y}] = \frac{1}{2N}\sum_{i=1}^{N}(\mathbb{E}_\pi[\tilde{y}^{(i)}] - t^{(i)})^2 + \frac{1}{2N}\sum_{i=1}^{N}\mathrm{var}_\pi[\tilde{y}^{(i)}]$$

$$\mathbb{E}_\pi[\tilde{y}_\pi^{(i)}] = y^{(i)} \quad \text{if} \quad \tilde{y}_\pi^{(i)} = \sum_j (1+\pi_j^{(i)}) w_j x_j^{(i)} \quad \text{where } \pi_j^{(i)} \overset{iid}{\sim} \mathcal{N}(0,\sigma^2)$$

↳ mean is same as $\mathbb{E}_m[\tilde{y}^{(i)}]$

$\therefore$ only need to look at variance term.

$$\frac{1}{2N}\sum_{i=1}^{N}\mathrm{var}_m\left[\frac{1}{p}\sum_j m_j^{(i)} x_j^{(i)} w_j\right] = \frac{1}{2N}\sum_{i=1}^{N}\mathrm{var}_\pi\left[\sum_j (1+\pi_j^{(i)}) x_j^{(i)} w_j\right]$$

$$\frac{1}{p^2}\sum_j \mathrm{var}_m[m_j^{(i)}](x_j^{(i)} w_j)^2 = \sum_j \mathrm{var}_\pi[1+\pi_j^{(i)}](x_j^{(i)} w_j)^2$$

$$\frac{1}{p^2}\sum_j p(1-p)(x_j^{(i)} w_j)^2 = \sum_j \sigma^2 (x_j^{(i)} w_j)^2$$

$$\therefore \sigma^2 = \frac{1-p}{p}$$

$$\therefore \sigma = \sqrt{\frac{1-p}{p}}$$