Student: Hena Ghonia(20213256)

Question 1 (6-9-6)**.** This question is about activation functions and vanishing/exploding gradients in recurrent neural networks (RNNs). Let $\sigma : \mathbb{R} \to \mathbb{R}$ be an activation function. When the argument is a vector, we apply σ element-wise. Consider the following recurrent unit:

$$\boldsymbol{h}_t = \boldsymbol{W}\sigma(\boldsymbol{h}_{t-1}) + \boldsymbol{U}\boldsymbol{x}_t + \boldsymbol{b}$$

1.1 Show that applying the activation function in this way results in an equivalent recurrence as the conventional way of applying the activation function: $\boldsymbol{g}_t = \sigma(\boldsymbol{W}\boldsymbol{g}_{t-1} + \boldsymbol{U}\boldsymbol{x}_t + \boldsymbol{b})$ (i.e. express $\boldsymbol{g}_t$ in terms of $\boldsymbol{h}_t$). More formally, you need to prove it using mathematical induction. You only need to prove the induction step in this question, assuming your expression holds for time step $t-1$.

Solution:

Assuming our expression holds for t-1 time step and $\sigma(h_t) = g_t$.

$$\boldsymbol{h}_t = \boldsymbol{W}\sigma(\boldsymbol{h}_{t-1}) + \boldsymbol{U}\boldsymbol{x}_t + \boldsymbol{b}$$

Using Recurrence relation we substitute below h_{t-1} value.

$$\boldsymbol{h}_{t-1} = \boldsymbol{W}\sigma(\boldsymbol{h}_{t-2}) + \boldsymbol{U}\boldsymbol{x}_{t-1} + \boldsymbol{b}$$

$$\boldsymbol{h}_t = \boldsymbol{W}\sigma(\boldsymbol{W}\sigma(\boldsymbol{h}_{t-2}) + \boldsymbol{U}\boldsymbol{x}_{t-1} + \boldsymbol{b}) + \boldsymbol{U}\boldsymbol{x}_t + \boldsymbol{b}$$

Substituting $h_{t-2} = g_{t-2}$

$$\boldsymbol{h}_t = \boldsymbol{W}\sigma(\boldsymbol{W}\boldsymbol{g}_{t-2} + \boldsymbol{U}\boldsymbol{x}_{t-1} + \boldsymbol{b}) + \boldsymbol{U}\boldsymbol{x}_t + \boldsymbol{b}$$

As given $\boldsymbol{g}_t = \sigma(\boldsymbol{W}\boldsymbol{g}_{t-1} + \boldsymbol{U}\boldsymbol{x}_t + \boldsymbol{b})$, we can write $\boldsymbol{g}_{t-1} = \sigma(\boldsymbol{W}\boldsymbol{g}_{t-2} + \boldsymbol{U}\boldsymbol{x}_{t-1} + \boldsymbol{b})$, substituting in above equation:

$$\boldsymbol{h}_t = \boldsymbol{W}g_{t-1} + \boldsymbol{U}\boldsymbol{x}_t + \boldsymbol{b}$$

Taking sigmoid both the sides

$$\sigma(\boldsymbol{h}_t) = \sigma(\boldsymbol{W}g_{t-1} + \boldsymbol{U}\boldsymbol{x}_t + \boldsymbol{b}) = g_t$$

Thus it is true for t-1 step by induction.

*1.2 Let $||\boldsymbol{A}||$ denote the L_2 operator norm [1] of matrix $\boldsymbol{A}$ ($||\boldsymbol{A}|| := \max_{\boldsymbol{x}:||\boldsymbol{x}||=1} ||\boldsymbol{A}\boldsymbol{x}||$). Assume $\sigma(x)$ has bounded derivative, i.e. $|\sigma'(x)| \leq \gamma$ for some $\gamma > 0$ and for all x. We denote as $\lambda_1(\cdot)$ the largest eigenvalue of a symmetric matrix. Show that if the largest eigenvalue of the weights is bounded by $\frac{\delta^2}{\gamma^2}$ for some $0 \leq \delta < 1$, gradients of the hidden state will vanish over time, i.e.

$$\lambda_1(\boldsymbol{W}^\top \boldsymbol{W}) \leq \frac{\delta^2}{\gamma^2} \quad \implies \quad \left|\left|\frac{\partial \boldsymbol{h}_T}{\partial \boldsymbol{h}_0}\right|\right| \to 0 \text{ as } T \to \infty$$

Use the following properties of the L_2 operator norm

$$||\boldsymbol{A}\boldsymbol{B}|| \leq ||\boldsymbol{A}||\,||\boldsymbol{B}|| \qquad \text{and} \qquad ||\boldsymbol{A}|| = \sqrt{\lambda_1(\boldsymbol{A}^\top \boldsymbol{A})}$$

1. The L_2 operator norm of a matrix $\boldsymbol{A}$ is is an *induced norm* corresponding to the L_2 norm of vectors. You can try to prove the given properties as an exercise.

Solution:

$$\boldsymbol{h}_t = \boldsymbol{W}\sigma(\boldsymbol{h}_{t-1}) + \boldsymbol{U}\boldsymbol{x}_t + \boldsymbol{b}$$

From previous question $\sigma(\boldsymbol{h}_t) = \boldsymbol{g}_t$.

$$\frac{\partial \boldsymbol{h}_T}{\partial \boldsymbol{h}_0} = \frac{\partial \boldsymbol{h}_t}{\partial \boldsymbol{h}_{t-1}} \frac{\partial \boldsymbol{h}_{t-1}}{\partial \boldsymbol{h}_2} \cdots \frac{\partial \boldsymbol{h}_1}{\partial \boldsymbol{h}_0} \tag{1}$$

where $t \in [1, ..., T]$

$$\frac{\partial \boldsymbol{h}_t}{\partial \boldsymbol{h}_{t-1}} = \frac{\partial \boldsymbol{W}\sigma(\boldsymbol{h}_{t-1}) + \boldsymbol{U}\boldsymbol{x}_t + \boldsymbol{b}}{\partial \boldsymbol{h}_{t-1}} = \boldsymbol{W}\frac{\partial \sigma(\boldsymbol{h}_{t-1})}{\partial h_{t-1}} = \boldsymbol{W}\sigma^{'}(\boldsymbol{h}_{t-1})$$

Similarly for $\frac{\partial \boldsymbol{h}_{t-1}}{\partial \boldsymbol{h}_{t-2}} = \boldsymbol{W}\sigma^{'}(\boldsymbol{h}_{t-2}) ... \frac{\partial \boldsymbol{h}_1}{\partial \boldsymbol{h}_0} = \boldsymbol{W}\sigma^{'}(\boldsymbol{h}_0)$. Substituting these values in equation (1).

$$\frac{\partial \boldsymbol{h}_T}{\partial \boldsymbol{h}_0} = \boldsymbol{W}\sigma^{'}(\boldsymbol{h}_{t-1}) \cdot \boldsymbol{W}\sigma^{'}(\boldsymbol{h}_{t-2}) \cdots \boldsymbol{W}\sigma^{'}(\boldsymbol{h}_0)$$

Taking L2 norm on both the sides

$$\left\| \frac{\partial \boldsymbol{h}_T}{\partial \boldsymbol{h}0} \right\| = \left\| \boldsymbol{W}\sigma^{'}(\boldsymbol{h}_{t-1}) \cdot \boldsymbol{W}\sigma^{'}(\boldsymbol{h}_{t-2}) \cdots \boldsymbol{W}\sigma^{'}(\boldsymbol{h}_0) \right\|$$

$$\text{Using} \qquad ||\boldsymbol{A}\boldsymbol{B}|| \leq ||\boldsymbol{A}||\,||\boldsymbol{B}||$$

$$\left\| \frac{\partial \boldsymbol{h}_T}{\partial \boldsymbol{h}0} \right\| \leq \|\boldsymbol{W}\| \left\| \boldsymbol{W}\sigma^{'}(\boldsymbol{h}_{t-1}) \right\| \cdot \|\boldsymbol{W}\| \left\| \boldsymbol{W}\sigma^{'}(\boldsymbol{h}_{t-2}) \right\| \cdot ... \|\boldsymbol{W}\| \left\| \boldsymbol{W}\sigma^{'}(\boldsymbol{h}_0) \right\|$$

$$\left\| \frac{\partial \boldsymbol{h}_T}{\partial \boldsymbol{h}0} \right\| \leq \|\boldsymbol{W}\|^T \gamma^T$$

$$\text{Using} \qquad ||\boldsymbol{A}|| = \sqrt{\lambda_1(\boldsymbol{A}^\top \boldsymbol{A})}$$

$$\|\boldsymbol{W}\|^T = (\lambda_1 \boldsymbol{W}^\top \boldsymbol{W})^{T/2}$$

$$\left\| \frac{\partial \boldsymbol{h}_T}{\partial \boldsymbol{h}0} \right\| \leq (\lambda_1 \boldsymbol{W}^\top \boldsymbol{W})^{T/2} (\gamma^2)^{T/2}$$

As it is given that largest eigenvalue of the weights is bounded by $\frac{\delta^2}{\gamma^2}$ for some $0 \leq \delta < 1$, $\|\boldsymbol{W}\|^T \gamma^T \leq (\frac{\delta^2}{\gamma^2})^{T/2} \gamma^T$

$$\left\| \frac{\partial \boldsymbol{h}_T}{\partial \boldsymbol{h}0} \right\| \leq \delta^T$$

Upper bound of derivative $\left\| \frac{\partial \boldsymbol{h}_T}{\partial \boldsymbol{h}0} \right\|$ is δ^T, taking its limit:

$$\lim_{T \to \infty} \left\| \frac{\partial \boldsymbol{h}_T}{\partial \boldsymbol{h}0} \right\| \leq \lim_{T \to \infty} \left\| \frac{\partial \boldsymbol{h}_T}{\partial \boldsymbol{h}0} \right\| \delta^T = 0$$

Since $0 \leq \delta < 1$, upper bound of derivative tends to 0 as T tends to infinity and consecutive derivative won't be larger than upper bound and hence all those derivatives will vanish which causes vanishing gradient problem.

1.3 What do you think will happen to the gradients of the hidden state if the condition in the previous question is reversed, i.e. if the largest eigenvalue of the weights is larger than $\frac{\delta^2}{\gamma^2}$? Is this condition *necessary* and/or *sufficient* for the gradient to explode ? (Answer in 1-2 sentences).

Solution: If gradients were to explode, then upper bound of $\left\|\frac{\partial \boldsymbol{h}_T}{\partial \boldsymbol{h}0}\right\|$in previous question will tend to ∞ over time. But if upper bound of $\lim_{T\to\infty}\left\|\frac{\partial \boldsymbol{h}_T}{\partial \boldsymbol{h}0}\right\|$ is finite then the gradients cannot exceed that value and gradient won't explode. So necessary condition is gradient in previous question becomes unbounded or upper bound is not zero. And if the largest eigenvalue of the weights is larger than $\frac{\delta^2}{\gamma^2}$ then upper bound will not tend to zero and the gradient might explode. Hence it is a *necessary* condition for gradients to explode.

Question 2 (8-8-8)**.** In this question you will demonstrate that an estimate of the first moment of the gradient using an (exponential) running average is equivalent to using momentum, and is biased by a scaling factor. The goal of this question is for you to consider the relationship between different optimization schemes, and to practice noting and quantifying the effect (particularly in terms of bias/variance) of *estimating* a quantity.

Let $\boldsymbol{g}_t$ be an unbiased sample of gradient at time step t and $\Delta\boldsymbol{\theta}_t$ be the update to be made. Initialize $\boldsymbol{v}_0$ to be a vector of zeros.

2.1 For $t \geq 1$, consider the following update rules:

— SGD with momentum:

$$\boldsymbol{v}_t = \alpha\boldsymbol{v}_{t-1} + \epsilon\boldsymbol{g}_t \qquad \Delta\boldsymbol{\theta}_t = -\boldsymbol{v}_t$$

where $\epsilon > 0$ and $\alpha \in (0, 1)$.

— SGD with running average of $\boldsymbol{g}_t$:

$$\boldsymbol{v}_t = \beta\boldsymbol{v}_{t-1} + (1-\beta)\boldsymbol{g}_t \qquad \Delta\boldsymbol{\theta}_t = -\delta\boldsymbol{v}_t$$

where $\beta \in (0, 1)$ and $\delta > 0$.

Express the two update rules recursively ($\Delta\boldsymbol{\theta}_t$ as a function of $\Delta\boldsymbol{\theta}_{t-1}$). Show that these two update rules are equivalent ; i.e. express (α, ϵ) as a function of (β, δ).

Solution: For SGD with momentum:

$$\boldsymbol{v}_t = \alpha\boldsymbol{v}_{t-1} + \epsilon\boldsymbol{g}_t \qquad \Delta\boldsymbol{\theta}_t = -\boldsymbol{v}_t$$

As given $\Delta\boldsymbol{\theta}_t = -\boldsymbol{v}_t$, we can write $\boldsymbol{v}_t = -\Delta\boldsymbol{\theta}_t$ $\boldsymbol{v}_{t-1} = -\Delta\boldsymbol{\theta}_{t-1}$

$$\boldsymbol{v}_t = \alpha(-\Delta\boldsymbol{\theta}_{t-1}) + \epsilon\boldsymbol{g}_t \qquad \Delta\boldsymbol{\theta}_t = -\boldsymbol{v}_t$$

$$-\Delta\boldsymbol{\theta}_t = \alpha(-\Delta\boldsymbol{\theta}_{t-1}) + \epsilon\boldsymbol{g}_t \tag{2}$$

For SGD with running average:

$$\boldsymbol{v}_t = \beta\boldsymbol{v}_{t-1} + (1-\beta)\boldsymbol{g}_t \qquad \Delta\boldsymbol{\theta}_t = -\delta\boldsymbol{v}_t$$

where $\beta \in (0, 1)$ and $\delta > 0$. As given $\Delta\boldsymbol{\theta}_t = -\boldsymbol{v}_t$, we can write $\boldsymbol{v}_t = -\frac{\Delta\boldsymbol{\theta}_t}{\delta}$, $\boldsymbol{v}_{t-1} = -\frac{\Delta\boldsymbol{\theta}_{t-1}}{\delta}$

$$\boldsymbol{v}_t = -\beta\frac{\Delta\boldsymbol{\theta}_{t-1}}{\delta} + (1-\beta)\boldsymbol{g}_t$$

$$-\frac{\Delta\boldsymbol{\theta}_t}{\delta} = -\beta\frac{\Delta\boldsymbol{\theta}_{t-1}}{\delta} + (1-\beta)\boldsymbol{g}_t$$
$$-\Delta\boldsymbol{\theta}_t = -\beta\Delta\boldsymbol{\theta}_{t-1} + \delta(1-\beta)\boldsymbol{g}_t \tag{3}$$

Comparing equation 2 and 3 we get $\alpha = \beta$, $\epsilon = \delta(1-\beta)$

2.2 Unroll the running average update rule, i.e. express $\boldsymbol{v}_t$ as a linear combination of $\boldsymbol{g}_i$'s $(1 \leq i \leq t)$.
Solution SGD with running average

$$\boldsymbol{v}_t = \beta\boldsymbol{v}_{t-1} + (1-\beta)\boldsymbol{g}_t \qquad \Delta\boldsymbol{\theta}_t = -\delta\boldsymbol{v}_t$$

$$\boldsymbol{v}_t = \beta\boldsymbol{v}_{t-1} + (1-\beta)\boldsymbol{g}_t$$
$$\boldsymbol{v}_t = \beta(\beta\boldsymbol{v}_{t-2} + (1-\beta)\boldsymbol{g}_{t-1}) + (1-\beta)\boldsymbol{g}_t$$
$$\boldsymbol{v}_t = \beta(\beta\boldsymbol{v}_{t-2}) + \beta(1-\beta)\boldsymbol{g}_{t-1} + (1-\beta)\boldsymbol{g}_t$$
$$\boldsymbol{v}_t = \beta(\beta(\beta\boldsymbol{v}_{t-3} + (1-\beta)\boldsymbol{g}_{t-2})) + \beta(1-\beta)\boldsymbol{g}_{t-1} + (1-\beta)\boldsymbol{g}_t$$
$$\boldsymbol{v}_t = (\beta^3\boldsymbol{v}_{t-3}) + \beta^2(1-\beta)\boldsymbol{g}_{t-2} + \beta(1-\beta)\boldsymbol{g}_{t-1} + (1-\beta)\boldsymbol{g}_t$$

For t terms, $\boldsymbol{v}_t$ can be represented as:

$$\boldsymbol{v}_t = \beta^t\boldsymbol{v}_0 + (1-\beta)\sum_{i=1}^{t}\beta^{t-i}g_i$$

Since $v_0 = 0$,

$$\boldsymbol{v}_t = (1-\beta)\sum_{i=1}^{t}\beta^{t-i}\boldsymbol{g}_i$$

2.3 Assume $\boldsymbol{g}_t$ has a stationary distribution independent of t. Show that the running average is biased, i.e. $\mathbb{E}[\boldsymbol{v}_t] \neq \mathbb{E}[\boldsymbol{g}_t]$. Propose a way to eliminate such a bias by rescaling $\boldsymbol{v}_t$.
Solution:

$$\mathbb{E}[\boldsymbol{v}_t] = \mathbb{E}[(1-\beta)\sum_{i=1}^{t}\beta^{t-i}\boldsymbol{g}_i]$$

$$\mathbb{E}[\boldsymbol{v}_t] = \mathbb{E}[\boldsymbol{g}_i](1-\beta)\sum_{i=1}^{t}\beta^{t-i} + \zeta$$

$$\mathbb{E}[\boldsymbol{v}_t] = \mathbb{E}[\boldsymbol{g}_i](1-\beta)\sum_{i=1}^{t}\beta^{t-i} + \zeta$$

$\sum_{i=1}^{t}\beta^{t-i} = 1 + \beta + \beta^2 + ... + \beta^{t-1} = \frac{(1-\beta^t)}{1-\beta}$

$$\mathbb{E}[\boldsymbol{v}_t] = \mathbb{E}[\boldsymbol{g}_i](1-\beta)\sum_{i=1}^{t}\beta^{t-i} + \zeta = \mathbb{E}[\boldsymbol{g}_i](1-\beta)\frac{(1-\beta^t)}{1-\beta} + \zeta$$

$$\mathbb{E}[\boldsymbol{v}_t] = \mathbb{E}[\boldsymbol{g}_i](1-\beta^t) + \zeta$$

$\mathbb{E}[\boldsymbol{v}_t] \neq \mathbb{E}[\boldsymbol{g}_t]$, hence running average is biased. A way to eliminate bias is to rescale v_t. As given $\mathbb{E}[\boldsymbol{v}_t]$ has stationary distribution and from above equation i.e true first moment is stationary so

$\zeta = 0$ otherwise ζ can be small as exponential decayβ can be choosen such that the exponential moving average assigns small weights to gradient too far in past. $(1-\beta^t)$ is caused by initializing the vector of zeros. So we divide v_t by $1-\beta^t$ to correct intialization bias. Hence rescaled v_t is $\frac{v_t}{1-\beta^t}$

Question 3 (8-8-6-9-3)**.** In this question, you will analyze the performance of dot-product attention and derive an efficient approximation of it. Consider that *multi-head* dot-product attention for a sequence of length n is defined as follows:

$$\begin{aligned}\text{MultiHead}(\bar{\boldsymbol{Q}}, \bar{\boldsymbol{K}}, \bar{\boldsymbol{V}}) &= \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\boldsymbol{W}^O \\ \text{where} \quad \text{head}_i &= \text{Attention}_{\text{std}}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) \quad (\text{here, } \boldsymbol{Q} := \bar{\boldsymbol{Q}}\boldsymbol{W}_i^Q, \boldsymbol{K} := \bar{\boldsymbol{K}}\boldsymbol{W}_i^K, \boldsymbol{V} := \bar{\boldsymbol{V}}\boldsymbol{W}_i^V) \\ &= \text{softmax}_{\text{row}}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_k}}\right)\boldsymbol{V}\end{aligned}$$

where $\bar{\boldsymbol{Q}}, \bar{\boldsymbol{K}}, \bar{\boldsymbol{V}} \in \mathbb{R}^{n \times d_{\text{model}}}$ are the queries, keys, and values, and $\boldsymbol{W}_i^Q, \boldsymbol{W}_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\boldsymbol{W}_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ $\forall i$, and $\boldsymbol{W}_O \in \mathbb{R}^{hd_v \times d_{model}}$ are the weights. The softmax subscript "row" indicates that the softmax is computed along the rows, and the Attention subscript "std" indicates that this is the standard variant (we will see other variants later in the question). For this question, you can assume that $d_k = d_v = d_{\text{model}}$ and call the value d.

For calculating the time and space complexities, you can also assume that matrix multiplications are performed naively. As an example, for $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{B}$ where $\boldsymbol{A} \in \mathbb{R}^{p \times q}$, $\boldsymbol{B} \in \mathbb{R}^{q \times r}$, and $\boldsymbol{C} \in \mathbb{R}^{p \times r}$, the time complexity is $\Theta(pqr)$ due to the three nested loops, and the space complexity is $\Theta(pq + qr + pr)$ from storing the inputs and the result.

3.1 What is the time and space complexity of the attention operation carried out by a single head in Θ-notation in terms of n and d? Use your answer to calculate the time and space complexity of multi-head dot-product attention in terms of n, d, and h, assuming that the heads are computed sequentially. For very long sequences, where does the bottleneck lie?

Solution: From above given equation time complexity for head_i is $\Theta(n^2 d)$ since $\boldsymbol{Q}$ and $\boldsymbol{K}$ is of size $n \times d$.
For single head time complexity: $\Theta(n^2 d) + \Theta(nd^2) = \Theta(n^2 d + nd^2)$
Space complexity: $\Theta(n^2 + 2nd) + \Theta(d^2 + 2nd) \approx \Theta(n^2 + d^2 + nd)$
For h heads:
time complexity: $\Theta(n^2 dh + nhd^2)$
Space complexity: $\Theta(n^2 + d^2 + hd^2 + nhd + nd)$
Bottleneck: time and space complexity will be in $\Theta(n^2)$ for larger sequence.

For the remaining parts, let us focus on the attention operation carried out by a single head. Furthermore, you can omit the scaling factor $\sqrt{d}$ without loss of generality by considering that $\boldsymbol{Q}$ and $\boldsymbol{K}$ can be scaled as desired.

3.2 Let us consider an alternative form of attention, one that performs row-wise softmax on $\boldsymbol{Q}$ and column-wise softmax on $\boldsymbol{K}$ separately as follows:

$$\text{Attention}_{\text{separable}}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}_{\text{row}}(\boldsymbol{Q})\text{softmax}_{\text{col}}(\boldsymbol{K})^\top \boldsymbol{V}.$$

Prove that $\text{softmax}_{\text{row}}(\boldsymbol{Q})\text{softmax}_{\text{col}}(\boldsymbol{K})^\top$ produces valid categorical distributions in every row, like $\text{softmax}_{\text{row}}(\boldsymbol{Q}\boldsymbol{K}^\top)$. If $n \gg d$, show that $\text{Attention}_{\text{separable}}$ can be faster and requires less space than $\text{Attention}_{\text{std}}$. Is $\text{Attention}_{\text{separable}}$ as expressive as $\text{Attention}_{\text{std}}$?
(Hint: For a valid categorical distribution $\boldsymbol{p} \in \mathbb{R}^d$ over d categories, $p_i \geq 0 \, \forall i \in \{1, \ldots, d\}$ and $\sum_{i=1}^d p_i = 1$.)

Solution: Suppose

$$\boldsymbol{Q} = \begin{bmatrix} q_{11} & q_{12} & \ldots & q_{1d} \\ q_{21} & q_{22} & \ldots & q_{2d} \\ \ldots & \ldots & \ldots & \ldots \\ q_{n1} & q_{n2} & \ldots & q_{nd} \end{bmatrix}_{n \times d} \quad \text{and } \boldsymbol{K} = \begin{bmatrix} k_{11} & k_{12} & \ldots & k_{1d} \\ k_{21} & k_{22} & \ldots & k_{2d} \\ \ldots & \ldots & \ldots & \ldots \\ k_{n1} & k_{n2} & \ldots & k_{nd} \end{bmatrix}_{n \times d}$$

$$\text{softmax}_{\text{row}}(\boldsymbol{Q}\boldsymbol{K}^\top) = \text{softmax}_{\text{row}} \begin{bmatrix} \sum_{i=1}^d q_{1i}k_{1i} & \sum_{i=1}^d q_{1i}k_{2i} & \ldots & \sum_{i=1}^d q_{1i}k_{ni} \\ \sum_{i=1}^d q_{2i}k_{1i} & \sum_{i=1}^d q_{2i}k_{2i} & \ldots & \sum_{i=1}^d q_{2i}k_{ni} \\ \ldots & \ldots & \ldots & \ldots \\ \sum_{i=1}^d q_{ni}k_{1i} & \sum_{i=1}^d q_{ni}k_{2i} & \ldots & \sum_{i=1}^d q_{ni}k_{ni} \end{bmatrix}_{n \times n}$$

$$\text{softmax}_{\text{row}}(\boldsymbol{Q}\boldsymbol{K}^\top) = \begin{bmatrix} \frac{e^{\sum_{i=1}^d q_{1i}k_{1i}}}{\sum_{j=1}^n e^{\sum_{i=1}^d q_{1i}k_{ji}}} & \frac{e^{\sum_{i=1}^d q_{1i}k_{2i}}}{\sum_{j=1}^n e^{\sum_{i=1}^d q_{1i}k_{ji}}} & \ldots & \frac{e^{\sum_{i=1}^d q_{1i}k_{ni}}}{\sum_{j=1}^n e^{\sum_{i=1}^d q_{1i}k_{ji}}} \\ \frac{e^{\sum_{i=1}^d q_{2i}k_{1i}}}{\sum_{j=1}^n e^{\sum_{i=1}^d q_{2i}k_{ji}}} & \frac{e^{\sum_{i=1}^d q_{2i}k_{2i}}}{\sum_{j=1}^n e^{\sum_{i=1}^d q_{2i}k_{ji}}} & \ldots & \frac{e^{\sum_{i=1}^d q_{2i}k_{ni}}}{\sum_{j=1}^n e^{\sum_{i=1}^d q_{2i}k_{ji}}} \\ \ldots & \ldots & \ldots & \ldots \\ \frac{e^{\sum_{i=1}^d q_{ni}k_{1i}}}{\sum_{j=1}^n e^{\sum_{i=1}^d q_{ni}k_{ji}}} & \frac{e^{\sum_{i=1}^d q_{ni}k_{2i}}}{\sum_{j=1}^n e^{\sum_{i=1}^d q_{ni}k_{ji}}} & \ldots & \frac{e^{\sum_{i=1}^d q_{ni}k_{ni}}}{\sum_{j=1}^n e^{\sum_{i=1}^d q_{ni}k_{ji}}} \end{bmatrix}_{n \times n}$$

Above matrix $\text{softmax}_{\text{row}}(\boldsymbol{Q}\boldsymbol{K}^\top)$ produces valid categorical distribution as summing over row values gives 1. $\frac{e^{\sum_{i=1}^d q_{1i}k_{1i}}}{\sum_{j=1}^n e^{\sum_{i=1}^d q_{1i}k_{ji}}} + \frac{e^{\sum_{i=1}^d q_{1i}k_{2i}}}{\sum_{j=1}^n e^{\sum_{i=1}^d q_{1i}k_{ji}}} + \ldots + \frac{e^{\sum_{i=1}^d q_{1i}k_{ni}}}{\sum_{j=1}^n e^{\sum_{i=1}^d q_{1i}k_{ji}}} = 1$ over n categories.

$$\text{softmax}_{\text{row}}(\boldsymbol{Q}) = \begin{bmatrix} \frac{e^{q_{11}}}{\sum_{i=1}^d e^{q_{1i}}} & \frac{e^{q_{12}}}{\sum_{i=1}^d e^{q_{1i}}} & \ldots & \frac{e^{q_{1d}}}{\sum_{i=1}^d e^{q_{1i}}} \\ \frac{e^{q_{21}}}{\sum_{i=1}^d e^{q_{2i}}} & \frac{e^{q_{22}}}{\sum_{i=1}^d e^{q_{2i}}} & \ldots & \frac{e^{q_{2d}}}{\sum_{i=1}^d e^{q_{2i}}} \\ \ldots & \ldots & \ldots & \ldots \\ \frac{e^{q_{n1}}}{\sum_{i=1}^d e^{q_{ni}}} & \frac{e^{q_{n2}}}{\sum_{i=1}^d e^{q_{ni}}} & \ldots & \frac{e^{q_{nd}}}{\sum_{i=1}^d e^{q_{ni}}} \end{bmatrix}_{n \times d}$$

$$\text{softmax}_{\text{col}}(\boldsymbol{K})^\top = \begin{bmatrix} \frac{e^{k_{11}}}{\sum_{i=1}^n e^{k_{i1}}} & \frac{e^{k_{21}}}{\sum_{i=1}^n e^{k_{i1}}} & \ldots & \frac{e^{k_{n1}}}{\sum_{i=1}^n e^{k_{i1}}} \\ \frac{e^{k_{12}}}{\sum_{i=1}^n e^{k_{i2}}} & \frac{e^{k_{22}}}{\sum_{i=1}^n e^{k_{i1}}} & \ldots & \frac{e^{k_{n2}}}{\sum_{i=1}^n e^{k_{i1}}} \\ \ldots & \ldots & \ldots & \ldots \\ \frac{e^{k_{1d}}}{\sum_{i=1}^n e^{k_{id}}} & \frac{e^{k_{2d}}}{\sum_{i=1}^n e^{k_{id}}} & \ldots & \frac{e^{k_{nd}}}{\sum_{i=1}^n e^{k_{id}}} \end{bmatrix}_{d \times n}$$

$$\text{softmax}_{\text{row}}(\boldsymbol{Q})\text{softmax}_{\text{col}}(\boldsymbol{K})^\top = \begin{bmatrix} \sum_{j=1}^d \frac{e^{q_{1j}+k_{1j}}}{\sum_{i=1}^d e^{q_{1i}} \sum_{l=1}^n e^{k_{lj}}} & \ldots & \sum_{j=1}^d \frac{e^{q_{1j}+k_{nj}}}{\sum_{i=1}^d e^{q_{1i}} \sum_{l=1}^n e^{k_{lj}}} \\ \ldots & \ldots & \ldots \\ \sum_{j=1}^d \frac{e^{q_{nj}+k_{1j}}}{\sum_{i=1}^d e^{q_{ni}} \sum_{l=1}^n e^{k_{lj}}} & \ldots & \sum_{j=1}^d \frac{e^{q_{nj}+k_{nj}}}{\sum_{i=1}^d e^{q_{ni}} \sum_{l=1}^n e^{k_{lj}}} \end{bmatrix}_{n \times n}$$

Above matrix $\text{softmax}_{\text{row}}(\boldsymbol{Q})\text{softmax}_{\text{col}}(\boldsymbol{K})^\top$ produces valid categorical distribution over n categories like $\text{softmax}_{\text{row}}(\boldsymbol{Q}\boldsymbol{K}^\top)$:

$\sum_{j=1}^{d} \frac{e^{q_{1j}+k_{1j}}}{\sum_{i=1}^{d} e^{q_{1i}} \sum_{l=1}^{n} e^{k_{lj}}} + \sum_{j=1}^{d} \frac{e^{q_{1j}+k_{2j}}}{\sum_{i=1}^{d} e^{q_{1i}} \sum_{l=1}^{n} e^{k_{lj}}} + ... + \sum_{j=1}^{d} \frac{e^{q_{1j}+k_{nj}}}{\sum_{i=1}^{d} e^{q_{1i}} \sum_{l=1}^{n} e^{k_{lj}}} = 1$
$\sum_{j=1}^{d} \frac{e^{q_{2j}+k_{1j}}}{\sum_{i=1}^{d} e^{q_{2i}} \sum_{l=1}^{n} e^{k_{lj}}} + ... + \sum_{j=1}^{d} \frac{e^{q_{2j}+k_{nj}}}{\sum_{i=1}^{d} e^{q_{2i}} \sum_{l=1}^{n} e^{k_{lj}}} = 1$
, ..., $\sum_{j=1}^{d} \frac{e^{q_{nj}+k_{1j}}}{\sum_{i=1}^{d} e^{q_{ni}} \sum_{l=1}^{n} e^{k_{lj}}} + ... + \sum_{j=1}^{d} \frac{e^{q_{nj}+k_{nj}}}{\sum_{i=1}^{d} e^{q_{ni}} \sum_{l=1}^{n} e^{k_{lj}}} = 1$

If $n \gg d$, time complexity of $\text{Attention}_{\text{std}}$ is $\Theta(n^2)$ and space complexity of $\Theta(n^2)$. For $\text{Attention}_{\text{separable}}$, we perform multiplication after calculating softmax of $\boldsymbol{Q}$ and $\boldsymbol{K}$. As $n \gg d$ and considering associative property of multiplication time complexity($Theta(n^2d)$) and space complexity($Theta(n^2+d^2+nd)$) reduces to $\Theta(n)$ for $\text{Attention}_{\text{separable}}$ $\text{Attention}_{\text{separable}}$ is not as expressive as $\text{Attention}_{\text{std}}$ i.e it does not have as many mapping as $\text{Attention}_{\text{std}}$ but $\text{Attention}_{\text{separable}}$ does not alter the parameter costs of the model since the model still retains the Q, K, V transforms from the original Transformer model.

3.3 Verify that the standard attention can be written as

$$\text{Attention}_{\text{std}}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{V}$$

where $\boldsymbol{A} = \exp\left(\boldsymbol{Q}\boldsymbol{K}^\top\right)$ and $\boldsymbol{D} = \text{diag}(\boldsymbol{A}\mathbf{1})$, where exp is an element-wise operation, diag creates a diagonal matrix from a vector, and $\mathbf{1}$ is a vector of ones. Note that you can store diagonal matrices in linear space and compute matrix multiplications with them in linear time.

Let us now consider a variant $\text{Attention}_{\text{approx}}$ where the elements a_{ij} of $\boldsymbol{A}$ can be represented as $a_{ij} = f(\boldsymbol{q}_i)^\top f(\boldsymbol{k}_j)$ for some $f : \mathbb{R}^d \to \mathbb{R}^m_+$, where $\boldsymbol{q}_i$ and $\boldsymbol{k}_j$ are the ith row of $\boldsymbol{Q}$ and the jth row of $\boldsymbol{K}$ respectively.

If $n \gg m$ and $n \gg d$, how can you use this formulation to make attention efficient ? What is the time and space complexity of $\text{Attention}_{\text{approx}}$?

(Hint: Decompose the matrix $\boldsymbol{A}$.)

Solution: $f(\boldsymbol{q}_i)$ and $f(\boldsymbol{k}_j)$ will be in $\mathbb{R}^{n\times m}$ as $f : \mathbb{R}^d \to \mathbb{R}^m_+$.
Time complexity to calculate $\boldsymbol{A}$ is $\Theta(nm^2)$ and space complexity of calculating $\boldsymbol{A}$ is $\Theta(2nm+n^2)$.
Time complexity of $\text{Attention}_{\text{std}} = \Theta(nmd + nm^2)$.
Space complexity of $\text{Attention}_{\text{std}} = \Theta(2n^2 + nd + 2nm + n^2)$.
If $n \gg m$ and $n \gg d$, then Space complexity of $\text{Attention}_{\text{std}} = \Theta(nmd)$ and time complexity of $\text{Attention}_{\text{std}} = \Theta(n^2)$, Since $(\Theta(2n^2 + nd + 2nm + n^2) \approx \Theta(n^2))$ which makes the formulation to make attention efficient.

*3.4 Prove that in $\text{Attention}_{\text{std}}$,

$$a_{ij} = \exp\left(\frac{-\|\boldsymbol{q}_i\|^2}{2}\right) \cdot \mathbb{E}_{\boldsymbol{x}\in\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\exp(\boldsymbol{x}^\top \boldsymbol{q}_i)\exp(\boldsymbol{x}^\top \boldsymbol{k}_j)\right] \cdot \exp\left(\frac{-\|\boldsymbol{k}_j\|^2}{2}\right).$$

Use this result to devise the function $f : \mathbb{R}^d \to \mathbb{R}^m_+$ introduced in the previous part, such that $\text{Attention}_{\text{approx}}$ approximates the expectation in $\text{Attention}_{\text{std}}$ by sampling.
(Hint 1: If $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{I})$, $p(\boldsymbol{x}) = (2\pi)^{-d/2}\exp\left(-\frac{1}{2}\|\boldsymbol{x} - \boldsymbol{\mu}\|^2\right)$ and $\int_{\boldsymbol{x}} p(\boldsymbol{x})d\boldsymbol{x} = 1$.)
(Hint 2: $\boldsymbol{x}^\top\boldsymbol{y} = -\frac{1}{2}(\boldsymbol{x}^\top\boldsymbol{x} - (\boldsymbol{x}+\boldsymbol{y})^\top(\boldsymbol{x}+\boldsymbol{y}) + \boldsymbol{y}^\top\boldsymbol{y})$.)
Solution: To prove $\text{Attention}_{\text{std}} = a_{ij} = \exp\left(\frac{-\|\boldsymbol{q}_i\|^2}{2}\right) \cdot \mathbb{E}_{\boldsymbol{x}\in\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\exp(\boldsymbol{x}^\top \boldsymbol{q}_i)\exp(\boldsymbol{x}^\top \boldsymbol{k}_j)\right] \cdot \exp\left(\frac{-\|\boldsymbol{k}_j\|^2}{2}\right)$
As given in previous question $\boldsymbol{A} = exp(\boldsymbol{Q}\boldsymbol{K}^\top)$ where elements a_{ij} of $\boldsymbol{A}$ can be expressed as

$a_{ij} = exp(\boldsymbol{q}_i \boldsymbol{k}_j^\top)$ where $\boldsymbol{q}_i$ and $\boldsymbol{k}_j$ are the ith row of $\boldsymbol{Q}$ and jth row of $\boldsymbol{K}$.

$$\begin{aligned}
\boldsymbol{A} &= exp(\boldsymbol{Q}\boldsymbol{K}^\top) \\
a_{ij} &= exp(\boldsymbol{q}_i \boldsymbol{k}_j^\top), \text{ Using Hint 2:} \\
&= exp(-\frac{1}{2}(\boldsymbol{q}_i^\top \boldsymbol{q}_i - (\boldsymbol{q}_i + \boldsymbol{k}_j)^\top (\boldsymbol{q}_i + \boldsymbol{k}_j) + \boldsymbol{k}_j^\top \boldsymbol{k}_j)) \\
&= exp(\frac{-\left\|\boldsymbol{q}_i\right\|^2}{2} + \frac{\left\|\boldsymbol{q}_i + \boldsymbol{k}_j\right\|^2}{2} + \frac{-\left\|k_j\right\|^2}{2}) \\
&= exp(\frac{-\left\|\boldsymbol{q}_i\right\|^2}{2}) \cdot exp(\frac{\left\|\boldsymbol{q}_i + \boldsymbol{k}_j\right\|^2}{2}) \cdot exp(\frac{-\left\|\boldsymbol{j}_j\right\|^2}{2})
\end{aligned} \tag{4}$$

Using Hint 1: Given that $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{I})$, $p(\boldsymbol{x}) = (2\pi)^{-d/2}\exp\left(-\frac{1}{2}\|\boldsymbol{x} - \boldsymbol{\mu}\|^2\right)$ and $\int_{\boldsymbol{x}} p(\boldsymbol{x})d\boldsymbol{x} = 1$ $p(\boldsymbol{x}) = (2\pi)^{-d/2}\exp\left(-\frac{1}{2}\|\boldsymbol{x} - (\boldsymbol{q}_i + \boldsymbol{k}_j)\|^2\right)$ and $\int_{\boldsymbol{x}}(2\pi)^{-d/2}\exp\left(-\frac{1}{2}\|\boldsymbol{x} - (\boldsymbol{q}_i + \boldsymbol{k}_j)\|^2\right) d\boldsymbol{x} = 1$. Replacing value of 1:

$$\begin{aligned}
\exp\left(\frac{\left\|\boldsymbol{q}_i + \boldsymbol{k}_j\right\|^2}{2}\right) &= \exp\left(\frac{\left\|\boldsymbol{q}_i + \boldsymbol{k}_j\right\|^2}{2}\right) \int_{\boldsymbol{x}} (2\pi)^{-d/2} \cdot \exp\left(-\frac{1}{2}\|\boldsymbol{x} - (\boldsymbol{q}_i + \boldsymbol{k}_j)\|^2\right) d\boldsymbol{x} \\
&= (2\pi)^{-d/2} \int_{\boldsymbol{x}} \exp\left(\frac{\left\|\boldsymbol{q}_i + \boldsymbol{k}_j\right\|^2}{2}\right) \cdot \exp\left(-\frac{1}{2}\|\boldsymbol{x} - (\boldsymbol{q}_i + \boldsymbol{k}_j)\|^2\right) d\boldsymbol{x} \\
&= (2\pi)^{-d/2} \int_{\boldsymbol{x}} \exp\left(\frac{\left\|\boldsymbol{q}_i + \boldsymbol{k}_j\right\|^2}{2} - \frac{\left\|\boldsymbol{x}\right\|^2}{2} + \boldsymbol{x}^\top(\boldsymbol{q}_i + \boldsymbol{k}_j) - \frac{\left\|\boldsymbol{q}_i + \boldsymbol{k}_j\right\|^2}{2}\right) d\boldsymbol{x} \\
&= (2\pi)^{-d/2} \int_{\boldsymbol{x}} \exp\left(-\frac{\left\|\boldsymbol{x}\right\|^2}{2} + \boldsymbol{x}^\top(\boldsymbol{q}_i + \boldsymbol{k}_j)\right) d\boldsymbol{x} \\
&= (2\pi)^{-d/2} \int_{\boldsymbol{x}} \exp\left(-\frac{\left\|\boldsymbol{x}\right\|^2}{2} + \boldsymbol{x}^\top(\boldsymbol{q}_i) + \boldsymbol{x}^\top(\boldsymbol{k}_j)\right) d\boldsymbol{x} \\
&= (2\pi)^{-d/2} \int_{\boldsymbol{x}} \exp\left(-\frac{\left\|\boldsymbol{x}\right\|^2}{2}\right) \cdot \exp\left(\boldsymbol{x}^\top(\boldsymbol{q}_i)\right) \cdot \exp\left(\boldsymbol{x}^\top(\boldsymbol{k}_j)\right) d\boldsymbol{x} \\
&= \mathbb{E}_{x \in \mathcal{N}(\mathbf{0}, \mathbf{I})}\left[\exp\left(\boldsymbol{x}^\top \boldsymbol{q}_i\right) \cdot \exp\left(\boldsymbol{x}^\top \boldsymbol{k}_j\right)\right]
\end{aligned} \tag{5}$$

Substituting value of equation (5) in equation (4) we get:

$$a_{ij} = \exp\left(\frac{-\|\boldsymbol{q}_i\|^2}{2}\right) \cdot \mathbb{E}_{\boldsymbol{x} \in \mathcal{N}(\mathbf{0}, \mathbf{I})}\left[\exp(\boldsymbol{x}^\top \boldsymbol{q}_i)\exp(\boldsymbol{x}^\top \boldsymbol{k}_j)\right] \cdot \exp\left(\frac{-\|\boldsymbol{k}_j\|^2}{2}\right)$$

We can represent $\text{attention}_{std} = \boldsymbol{K}(\boldsymbol{q}_i^\top, \boldsymbol{k}_j^\top)$ with $\boldsymbol{q}_i/\boldsymbol{k}_j$ standing for the ith/jth, query/key row-vector in $\boldsymbol{Q}, \boldsymbol{K}$ and kernel $\boldsymbol{K} : R^d \times R^d \to R_+$ defined for the (usually randomized) mapping: $\phi : R^d \to R_+^m$ (for some r > 0) as:

$$\boldsymbol{K}(\boldsymbol{q}_i^\top, \boldsymbol{k}_j^\top) = E[\phi(\boldsymbol{q}_i^\top)^\top \phi(\boldsymbol{k}_j^\top)^\top]$$

To approximate softmax kernels for attention we write ϕ of following form for functions $f_1, ...f_l : R \to R$ for deterministic vectors $\omega_1, ...\omega_m$ comes from iid distribution :D $= \mathcal{N}(0, I_d)$ which is

isotropic, we can model most kernals used in practise.

$$\phi(x) = \frac{h(x)}{\sqrt{m}}\left(f_1(\omega_1^\top x), ..., f_1(\omega_m^\top x), ..., f_l(\omega_1^\top x), ..., f_l(\omega_m^\top x)\right)$$

where $h(x) = 1, l = 2, f_1 = sin,\ f_2 = cos$ correspond to shift-invariant kernels, in particular D $= \mathcal{N}(0, I_d)$ leads to the Gaussian kernel. The softmax-kernel i.e attention matrix is given as: $exp(\boldsymbol{Q}^\top \boldsymbol{K})$. In this equation we remove $\sqrt{d}$ renormalization since we can equivalently renormalize input keys and queries. Hence we obtain random feature map: $\exp\left(\frac{-\|\boldsymbol{q}_i\|^2}{2}\right) \cdot \boldsymbol{K}(\boldsymbol{q}_i^\top, \boldsymbol{k}_j^\top) \cdot \exp\left(\frac{-\|\boldsymbol{k}_j\|^2}{2}\right)$ which approximates the expectation in Attention_{std} by sampling.

3.5 Discuss the implications of the choice of m for $\text{Attention}_{\text{approx}}$. What are the trade-offs to think about ?

Solution: Time complexity does depend on choice of m. Bigger m results in higher computation costs, but also in a lower variance of the estimate of $\boldsymbol{A}$. To decrease computation, choosing lower m will be beneficial but it will result in higher variance which is not desirable. Hence we choice of m and variance trade off.

Question 4 (4-5-6-6)**.** In this question, you will reconcile the relationship between L2 regularization and weight decay for the Stochastic Gradient Descent (SGD) and Adam optimizers. Imagine you are training a neural network (with learnable weights θ) with a loss function $L(f(\mathbf{x}^{(i)}, \theta), \mathbf{y}^{(i)})$, under two different schemes. The *weight decay* scheme uses a modified SGD update rule: the weights θ decay exponentially by a factor of λ. That is, the weights at iteration $i + 1$ are computed as

$$\theta_{i+1} = \theta_i - \eta \frac{\partial L(f(\mathbf{x}^{(i)}, \theta_i), \mathbf{y}^{(i)})}{\partial \theta_i} - \lambda \theta_i$$

where η is the learning rate of the SGD optimizer. The *L2 regularization* scheme instead modifies the loss function (while maintaining the typical SGD or Adam update rules). The modified loss function is

$$L_{\text{reg}}(f(\mathbf{x}^{(i)}, \theta), \mathbf{y}^{(i)}) = L(f(\mathbf{x}^{(i)}, \theta), \mathbf{y}^{(i)}) + \gamma \|\theta\|_2^2$$

4.1 Prove that the *weight decay* scheme that employs the modified SGD update is identical to an *L2 regularization* scheme that employs a standard SGD update rule.

Solution: Considering SGD rule:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta L_{reg}$$

Modified loss is given as :

$$L_{\text{reg}}(f(\mathbf{x}^{(i)}, \theta), \mathbf{y}^{(i)}) = L(f(\mathbf{x}^{(i)}, \theta), \mathbf{y}^{(i)}) + \gamma \|\theta\|_2^2$$

Taking gradient of modified loss:

$$\nabla_\theta L_{reg} = \frac{\partial L(f(\mathbf{x}^{(i)}, \theta), \mathbf{y}^{(i)})}{\partial \theta} + \gamma \frac{\partial \|\theta\|_2^2}{\partial \theta}$$

$$\nabla_\theta L_{reg} = \frac{\partial L(f(\mathbf{x}^{(i)}, \theta), \mathbf{y}^{(i)})}{\partial \theta} + 2\gamma\theta$$

where $\|\theta\|_2^2 = \theta^\top\theta$ and $\frac{\partial\|\theta\|_2^2}{\partial\theta} = 2\theta$
Substituting value in SGD rule i.e the modified SGD update rule.

$$\theta_{i+1} = \theta_i - \eta\frac{\partial}{\partial\theta_i}[L(f(\mathbf{x}^{(i)},\theta),\mathbf{y}^{(i)}) + \gamma\|\theta\|_2^2]$$

$$\theta_{i+1} = \theta_i - \eta\frac{\partial}{\partial\theta_i}[L(f(\mathbf{x}^{(i)},\theta),\mathbf{y}^{(i)})] + \gamma 2\eta\theta_i$$

For $2\gamma\eta = \lambda$, *L2 regularization* scheme also follows similar weight update rule as modified SGD rule.

4.2 This question refers to the Adam algorithm as described in the lecture slide (also identical to Algorithm 8.7 of the deep learning book). It turns out that a one-line change to this algorithms gives us Adam with an L2 regularization scheme. Identify the line of the algorithm that needs to change, and provide this one-line modification.
Solution: To change line where g_t is computed i.e to change gradient. Line to be modified:

$$g_t \leftarrow \frac{1}{m}\Delta_\theta\sum_i L(f(x^i;\theta),y^{(i)})$$

Updated modification:

$$g_t \leftarrow \frac{1}{m}\Delta_\theta\sum_i L(f(x^i;\theta),y^{(i)}) + \lambda\theta_i$$

4.3 Consider a "decoupled" weight decay scheme for the original Adam algorithm (see lecture slides, or equivalently, Algorithm 8.7 of the deep learning book) with the following two update rules.

— The **Adam-L2-reg** scheme computes the update by employing an L2 regularization scheme (same as the question above).

— The **Adam-weight-decay** scheme computes the update as $\boldsymbol{\Delta}\theta = -\left(\epsilon\frac{\hat{\mathbf{s}}}{\sqrt{\hat{\mathbf{r}}}+\delta} + \lambda\theta\right)$.

Now, assume that the neural network weights can be partitioned into two disjoint sets based on their gradient magnitude: $\theta = \{\theta_{\text{small}}, \theta_{\text{large}}\}$, where each weight $\theta_s \in \theta_{\text{small}}$ has a much smaller gradient magnitude than each weight $\theta_l \in \theta_{\text{large}}$. Using this information provided, answer the following questions. In each case, provide a brief explanation as to why your answer holds.

(a) Under the **Adam-L2-reg** scheme, which set of weights among θ_{small} and θ_{large} would you expect to be regularized (i.e., driven closer to zero) more strongly than the other ? Why ?

Solution: According to formula of $\Delta\theta$ from deep learning book we write $\widehat{s}$ and $\widehat{r}$ used in $\Delta\theta$:

$$\widehat{s} = \frac{\rho_1 s + (1-\rho_1)g}{(1-\rho_2)^t}$$

$$\widehat{r} = \frac{\rho_2 s + (1-\rho_2)g}{(1-\rho_2)^t}$$

For Adam L2 reg scheme weight update will happen as follows:

$$\theta_{t+1} = \theta_t - \epsilon\frac{\widehat{s}}{\sqrt{\widehat{r}}+\delta}$$

From above equation $\widehat{r}$ will be smaller for θ_{small} which gives large gradient update and for θ_{large} it might give small gradient update. Hence, for **Adam-L2-reg** scheme, weights that tend to have large gradient i.e θ_{large} do not get regularized as much as they would with decoupled weight decay as the gradient of the regularizer gets scaled along with gradient $L(f(\mathbf{x}^{(i)}, \theta), \mathbf{y}^{(i)})$ mentioned in above equation. This leads to an inequivalence of L2 and decoupled weight decay regularization for adaptive gradient algorithms. Hence θ_{small} gets regularized more strongly than other.

(b) Would your answer change for the **Adam-weight-decay** scheme ? Why/why not ?
Solution: Adam-weight-decay regularizes both Θ_{large} and Θ_{small} or all weights equally with the same rate λ as given in weight update equation: $\mathbf{\Delta}\theta = -\left(\epsilon \frac{\hat{\mathbf{s}}}{\sqrt{\hat{\mathbf{r}}}+\delta} + \lambda\theta\right)$

(Note: for the two sub-parts above, we are interested in the rate at which the weights are regularized, *relative* to their initial magnitudes.)

4.4 In the context of all of the discussion above, argue that weight decay is a better scheme to employ as opposed to L2 regularization ; particularly in the context of adaptive gradient based optimizers. (Hint: think about how each of these schemes regularize each parameter, and also about what the overarching objective of regularization is).

Solution: In context of adaptive gradient based optimizers, with L2 regularization both type of gradient(θ_{large} and θ_{small}) are normalized by their magnitudes and weights θ_{i+1} with large gradient magnitude are regularized by a smaller relative amount than other weights whereas for **Adam- weight decay scheme**, it regularizes weight with same rate λ.
Also, Adam with decoupled weight decay largely decouples weight decay and learning rate.
In paper - Loshchilov, Ilya, and Frank Hutter. "Decoupled weight decay regularization.", Adam with weight decay yielded better generalization performance for similar training loss values compared to Adam with L2. Hence weight decay is a better scheme to employ as opposed to L2 regularization for adaptive gradient based method.