PROBLEM 1 **(25 points)**

Let $\mathbf{A}$ be the matrix that projects a vector $\mathbf{y} \in \mathbb{R}^3$ onto the plane $\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$. Provide your reasoning or show the derivation of your answer to the following questions:

(a) What are the eigenvalues of the matrix $\mathbf{A}$? (3 points)

(b) What are the eigenvectors of the matrix $\mathbf{A}$? (3 points)

(c) Find $\det(\mathbf{A})$. (3 points)

(d) Find $\mathbf{A}$. (3 points)

(e) Find $\mathbf{A}^2$. (3 points)

(f) Find $\mathbf{A}^+$ where $^+$ denotes the pseudo-inverse. (3 points)

(g) Find $\|\mathbf{A}\|_F^2$ (3 points)

(h) Find $\mathbf{B}\mathbf{B}^+$, where $\mathbf{B} = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \end{bmatrix}$. (4 points)

Answer

(a) Recall that a pair of eigenvalue λ and eigenvector $\mathbf{v}$ satisfies that $Av = \lambda v$. The vector $\mathbf{n} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix}^T$ is the normal vector of the plane, and thus it would be projected to the zero point by $\mathbf{A}$, namely, $\mathbf{A}\mathbf{n} = \mathbf{0} = 0 \cdot \mathbf{n}$, which immediately implies that $\mathbf{v}_1 = \frac{\mathbf{n}}{\|\mathbf{n}\|} = \mathbf{n}$ is an eigenvector with eigenvalue $\lambda_1 = 0$. Next, notice that the projection of any vector $\mathbf{x}$ on the plane is $\mathbf{x}$ itself, ie. $\mathbf{A}\mathbf{x} = \mathbf{x}$. It follows that any two orthogonal vectors $\mathbf{v}_2$ on the plane could be the rest two eigenvectors of $\mathbf{A}$, both with eigenvalue equal to 1 ($\lambda_2 = \lambda_3 = 1$). To determine $\mathbf{v}_2$ and $\mathbf{v}_3$, we use the property that that $\mathbf{v}_2$ and $\mathbf{v}_3$ both are orthogonal to $\mathbf{v}_1$, and thus a trivial choice would be $\mathbf{v}_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \end{bmatrix}^T$ and $\mathbf{v}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$.

(b) See **(a)**.

(c) $\det(\mathbf{A}) = \Pi_{i=1}^3 \lambda_i = 0 \cdot 1 \cdot 1 = 0$.

(d) Let $\mathbf{Q} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$ and $\mathbf{\Lambda} = diag([\lambda_1, \lambda_2, \lambda_3])$. We have $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$. Simple algebra shows that,

$$\mathbf{A} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(e) Since $\mathbf{A}$ is a projection matrix, $\mathbf{A}^2 = \mathbf{A}$

(f) Since $\mathbf{A}$ is diagonalizable, $\mathbf{A}^+ = \mathbf{Q}\mathbf{\Lambda}^+\mathbf{Q}^T = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T = \mathbf{A}$

(g) $\|\mathbf{A}\|_F^2 = \sum_{i=1}^3 \lambda_i^2 = 0 + 1 + 1 = 2$.

(h) Note that $\mathbf{B}\mathbf{B}^+$ is the projection matrix for the column space of $\mathbf{B}$. Since the columns of $\mathbf{B}$ span the plane $\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$, $\mathbf{B}\mathbf{B}^+ = \mathbf{A}$

PROBLEM 2 **(25 points)**

(a) (15 points)

Consider a set of input vectors in $\mathbb{R}^n$, $\{\mathbf{x}^1, \mathbf{x}^2 \cdots \mathbf{x}^k\}$ and a set of output scalars $\{y^1, y^2 \cdots y^k\}$. We are interested in finding a regression model of the form $\hat{y}(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$.

Assume that, as a method for regularization, you add a noise vector ϵ^i to each input before learning the regression model. ϵ^i are independently drawn from a $\mathcal{N}(\mathbf{0}, \lambda\mathbf{I})$ distribution.

Consider the loss function $J(\mathbf{w}) = \frac{1}{k}\sum_{i=1}^{k}\left[\mathbf{w}^T\left(\mathbf{x}^i + \epsilon^i\right) - \mathbf{y}^i\right]^2$.

Find $\tilde{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}}\, \mathbb{E}\left(J(\mathbf{w})\right)$

(b) (10 points)

Consider $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$.
What unit vector ν when added to the input results in the maximum change in the output i.e $\|f(\mathbf{x}+\nu) - f(\mathbf{x})\|_2^2$?

Possibly helpful identities:-
$\nabla_{\mathbf{x}}\mathbf{a}^T\mathbf{x} = \mathbf{a}$

$\nabla_{\mathbf{x}}\mathbf{x}^T\mathbf{A}\mathbf{x} = (\mathbf{A} + \mathbf{A}^T)\mathbf{x}$

$\underbrace{\mathbf{a}\mathbf{a}^T}_{\text{Matrix}}\ \mathbf{a} = \underbrace{(\mathbf{a}^T\mathbf{a})}_{\text{Scalar}}\ \mathbf{a}$

Answer

(a) Let $\mathbf{X} = [\mathbf{x}^1, \cdots, \mathbf{x}^k]$ and $\mathbf{y} = [y^1, \cdots, y^k]^T$. We first compute the expectation of the loss function,

$$\begin{aligned}
\mathbf{E}\big[J(\mathbf{w})\big] &= \mathbf{E}\big[\frac{1}{k}\sum_{i=1}^{k}\left[\mathbf{w}^T\left(\mathbf{x}^i + \epsilon^i\right) - \mathbf{y}^i\right]^2\big] \\
&= \mathbf{E}\big[\frac{1}{k}\sum_{i=1}^{k}\left[\mathbf{w}^T\left(\mathbf{x}^i - \mathbf{y}^i\right) + \mathbf{w}^T\epsilon^i\right]^2\big] \\
&= \frac{1}{k}\sum_{i=1}^{k}(\mathbf{w}^T\mathbf{x}^i - \mathbf{y}^i)^2 - 2(\mathbf{w}^T\mathbf{x}^i - \mathbf{y}^i)\mathbf{w}^T\mathbf{E}\big[\epsilon^i\big] + \mathbf{w}^T\mathbf{w}\mathbf{E}\big[(\epsilon^i)^2\big] \\
&= \frac{1}{k}(\mathbf{X}^T\mathbf{w} - \mathbf{y})^T(\mathbf{X}^T\mathbf{w} - \mathbf{y}) + \lambda\mathbf{w}^T\mathbf{w} \\
&= \frac{1}{k}(\mathbf{w}^T\mathbf{X}\mathbf{X}^T\mathbf{w} - 2\mathbf{w}^T\mathbf{X}\mathbf{y} + \mathbf{y}^T\mathbf{y}) + \lambda\mathbf{w}^T\mathbf{w}
\end{aligned}$$

Then, we compute the gradient $\nabla_{\mathbf{w}}\mathbf{E}\big[J(\mathbf{w})\big]$,

$$\begin{aligned}
\nabla_{\mathbf{w}}\mathbf{E}\big[J(\mathbf{w})\big] &= \frac{1}{k}(\nabla_{\mathbf{w}}\mathbf{w}^T\mathbf{X}\mathbf{X}^T\mathbf{w} - 2\nabla_{\mathbf{w}}\mathbf{w}^T\mathbf{X}\mathbf{y} + \nabla_{\mathbf{w}}\mathbf{y}^T\mathbf{y}) + \lambda\nabla_{\mathbf{w}}\mathbf{w}^T\mathbf{w} \\
&= \frac{1}{k}\big((\mathbf{X}\mathbf{X}^T + (\mathbf{X}\mathbf{X}^T)^T)\mathbf{w} - 2\mathbf{X}\mathbf{y}\big) + 2\lambda\mathbf{w} \\
&= \frac{1}{k}(2\mathbf{X}\mathbf{X}^T\mathbf{w} - 2\mathbf{X}\mathbf{y}) + 2\lambda\mathbf{w}
\end{aligned}$$

Equating $\nabla_{\mathbf{w}}\mathbf{E}\big[J(\mathbf{w})\big]$ to 0 results in,

$$(\mathbf{X}\mathbf{X}^T + k\lambda\mathbf{I})\mathbf{w} = \mathbf{X}\mathbf{y},$$

and so,

$$\tilde{\mathbf{w}} = (\mathbf{X}\mathbf{X}^T + k\lambda\mathbf{I})^{-1}\mathbf{X}\mathbf{y}.$$

(b) The answer is simply the unit vector along the direction of the gradient of the loss function. The gradient of f equals to $\nabla_{\mathbf{x}}\mathbf{w}^T\mathbf{x} = \mathbf{w}$, and thus $\mathbf{v} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$. In more details, we formulate the constraint optimization problem for the objective function $\|f(\mathbf{x}+\nu) - f(\mathbf{x})\|_2^2 = \|\mathbf{w}^T\mathbf{v}\|_2^2 = \mathbf{v}^T\mathbf{w}\mathbf{w}^T\mathbf{v}$,

$$\mathbf{v}^* = \arg\max_{\mathbf{v}} \big[\mathbf{v}^T\mathbf{w}\mathbf{w}^T\mathbf{v}\big] \text{ s.t } \mathbf{v}^T\mathbf{v} = 1$$

Recall from Chapter 2, the solution to the above problem is the eigenvector of $\mathbf{w}\mathbf{w}^T$ with the largest eigenvalue. Since $\mathbf{w}\mathbf{w}^T$ is formed by a single vector $\mathbf{w}$, it has only one nonzero eigenvalue $\|\mathbf{w}\|$ paired with the eigenvector $\mathbf{v} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$. To see this, we introduce the *Lagrange multiplier* λ and define the *Lagrange* function,

$$\mathbf{L}(\mathbf{v}, \lambda) = \mathbf{v}^T\mathbf{w}\mathbf{w}^T\mathbf{v} - \lambda(1 - \mathbf{v}^T\mathbf{v}),$$

and solve,

$$\begin{aligned} 0 &= \nabla_{\mathbf{v}}\mathbf{L}(\mathbf{v}, \lambda) \\ &= \nabla_{\mathbf{v}}\big[\mathbf{v}^T\mathbf{w}\mathbf{w}^T\mathbf{v} - \lambda(1 - \mathbf{v}^T\mathbf{v})\big] \\ &= 2\mathbf{w}\mathbf{w}^T\mathbf{v} - 2\lambda\mathbf{v}, \end{aligned}$$

which leads to,

$$\mathbf{w}\mathbf{w}^T\mathbf{v} = \lambda\mathbf{v}.$$

The above equation is exactly the formulation of eigenvalue and eigenvector.

PROBLEM 3 **(25 points)**

For each of the following questions, write down T (true) or F (false). Each question is worth 1 point.

1. Like many other machine learning frameworks, the deep learning approach formulates the learning task as an optimization problem minimizing a cost function, and applies the back-propagation algorithm to find the parameters that result in a global minima.

   Ans: False. The back-propogation algorithm is a first order method, which finds a local minumum.

2. Every real-valued symmetric matrix is diagonalizable, and its determinant is nonzero.

   Ans: False. Some of the eigenvalues could be zero, and hence lead to zero determinant, ex. the zero matrix.

3. L^1 regularization induces the sparsity property

   Ans: True

4. The rectified linear unit $\phi(a) = \max(0, a)$ is widely used in hidden layers of the modern deep network frameworks. It is a better choice than *tanh* because it is bounded and gives constant derivative for positive input.

   Ans: False. The reLU is unbounded.

5. A model which achieves minimum training error also achieves minimum test error.

   Ans: False. Cases of overfitting.

6. Unbiased estimators are always more preferable than biased ones.

   Ans: False. Biased estimators might have smaller variance, which, in some cases, would be more preferable.

7. If two random variables are independent, then they are also uncorrelated.

   Ans: True.

8. Given an abstract formulation of deep network model $y = f(x)$, it can be mathematically shown that minimizing the mean squared error of f yields an estimator of the conditional expectation of the output y given the input x.

   Ans: True. Please see eq(6.1) in the textbook.

9. The *softmax* function is a popular choice for the output layer of deep network because it can be interpreted to generate a probability distribution over a finite set of outcomes.

   Ans: True.

10. In order to achieve global optimal solution, deep learning approaches are often expressed in terms of convex optimization. One example is to use the squared norm of the error i.e $L(y, \hat{y}) = \|y - \hat{y}\|_2^2$ for the loss function, where y is the ground truth, and $\hat{y}$ is the outcome of the network.

    Ans: False. Most deep learning problem is difficult to express in terms of convex optimization.

11. Constrained optimization problems can be solved by the Karush-Kuhn-Tucker (KKT) approach, which transforms a constrained problem to an unconstrained one. The KKT approach always succeeds so long as at least one feasible point exists.

    Ans: False. KKT also requires that the objective function does not allow infinite value.

12. The Principle Component Analysis(PCA) is a common method for dimensionality reduction.

    Ans: True.

13. The learning rate of the gradient descent algorithm needs to be set carefully. A good practice is to use the eigenvalues of the Hessian of the cost function to determine the scale of the learning rate.

    Ans: True.

14. The *no free lunch theorem* implies that there is no universal procedure for examining a training set of specific examples and choosing a function that will generalize to points not in the training set.

    Ans: True.

15. The state-of-the-art deep network softwares use minibatch to allow parallelization of forward-propagation and back-propagation across examples.

    Ans: True.

16. In general, deeper models (models with more hidden layers) of deep network tend to perform better than a single hidden layer network having a similar number of parameters.

    Ans: True. Please see Homework, and Figure 6.9 in the textbook.

17. $D_{KL}(P||Q) = D_{KL}(Q||P)$ for all probability distributions $P(x), Q(x)$ over the same random variable

    Ans: False.

18. The MAP estimator is equivalent to the ML estimator when the prior follows uniform distribution over a bounded parameter space.

    Ans: True.

19. *Occam's razor* infers that the feed-forward network with at least one hidden layer and a linear output layer can approximate any continuous function from one finite space to another with arbitrary precision, provided that that the network is given enough hidden units.

    Ans: False. This is referring to the universal approximation theorem.

20. Modern deep learning softwares usually implement the stochastic gradient descent rather than the classic gradient decent. One advantage of stochastic gradient decent is that it uses a predetermined stochastic process to generate a value for learning rate at each iteration.

    Ans: False.

21. Regularization methods in deep learning always involve adding a penalty function to the objective function.

    Ans: False. Other methods include noise injection, data augmentation.

22. The back-propagation algorithm is a systematic method based on the chain rule for computing the derivative of the cost function with respect to the parameters, starting from the last layer and going back to the first layer.

    Ans: True.

23. The deep learning library, Theano, handles automatic differentiation by using a symbolic representation of user-defined cost function to compute a symbolic representation of that function's gradient. This feature of Theano allows user to focus on design of forward path, whereas, historically, researchers used to spend great amount of time on deriving the gradient of complex networks.

    Ans: True.

24. Unsupervised learning algorithms experience a dataset containing features, but each example is also associated with a label or target.

    Ans: False. Supervised learning.

25. Regularization is any component of model training or learning that is introduced to overcome limitations that arise with limited training datasets, such as overfitting.

    Ans: True.

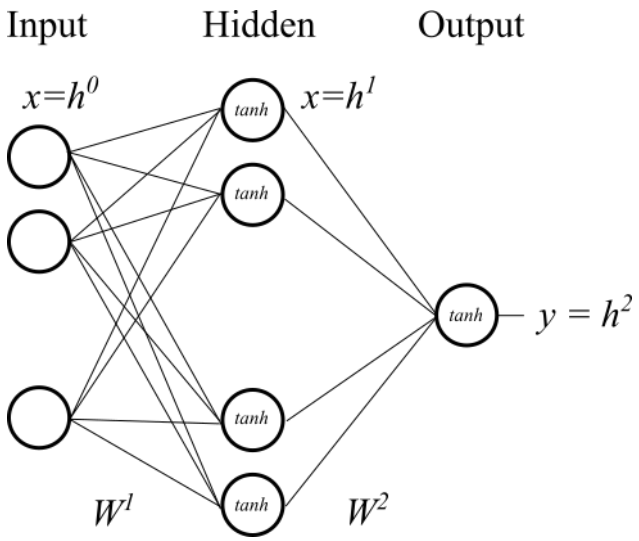
PROBLEM 4 **(25 points)**

Consider a shallow neural network with one input layer, one hidden layer, and one output layer. The input and hidden layers have N and M neurons, respectively, while the output layer has a single neuron.

In this model, we choose to use the hyperbolic tangent activation function:

$$h^2 = \tanh\left(b^2 + (\mathbf{h}^1)^T \mathbf{w}^2\right)$$
$$\mathbf{h}^1 = \tanh\left(\mathbf{b}^1 + (\mathbf{h}^0)^T \mathbf{W}^1\right)$$

$\mathbf{h}^0$ denotes the input, and $\mathbf{h}^1$ and h^2 are the output of the hidden layer and the output layer respectively. Consequently, $\mathbf{h}^0 \in \mathbb{R}^N$, $\mathbf{h}^1 \in \mathbb{R}^M$, and $h^2 \in \mathbb{R}$. $\mathbf{W}^1$ is the weight matrix from input to hidden layer and $\mathbf{w}^2$ is a weight vector from hidden layer to output layer. $\mathbf{b}^1$ and b^2 are the biases for the hidden and output layer respectively. The entire model is expressed as $y = f(\mathbf{x})$



For a given set of input $\{\mathbf{x}^1, ..., \mathbf{x}^k\}$ with their associated output $\{y^1, ..., y^k\}$, we use *mean squared error* as the cost function:

$$L = \frac{1}{k}\sum_{i=1}^{k}(y^i - f(\mathbf{x}^i))^2$$

In this problem, you are asked to derive the back-propagation algorithm.

(a) (10 points)

Derive $\frac{\partial L}{\partial \mathbf{w}^2}$ and $\frac{\partial L}{\partial \mathbf{h}^1}$. You might want to start with applying *chain rule*. $\left(\text{Hint: } \frac{d}{dx}\tanh(x) = \text{sech}^2(x)\right)$

(b) (10 points)

Derive $\frac{\partial L}{\partial \mathbf{W}^1}$. For this part, you might want to express $\mathbf{W}^1$ as

$$\mathbf{W}^1 = \begin{bmatrix} | & & | \\ \mathbf{w}_1^1 & \cdots & \mathbf{w}_M^1 \\ | & & | \end{bmatrix}$$

where $\mathbf{w}_i^1$, for $i = 1, ..., M$, is a column vector of $\mathbf{W}^1$. Then, derive $\frac{\partial L}{\partial \mathbf{w}_i^1}$ in terms of $\frac{\partial L}{\partial \mathbf{h}_i^1}$.

(c) (5 points)

If the loss function was instead defined as

$$L_r = \frac{1}{k}\sum_{i=1}^{k}(y^i - f(\mathbf{x}^i))^2 + \|\mathbf{W}^1\|_F^2 + \|\mathbf{w}^2\|_2^2$$

what will be the expressions for $\frac{\partial L}{\partial \mathbf{w}^2}$ and $\frac{\partial L}{\partial \mathbf{W}^1}$?

Possibly helpful identities:-

$\nabla_{\mathbf{x}}\mathbf{w}^T\mathbf{x} = \mathbf{w}$

$\|\mathbf{W}\|_F^2 = Tr(\mathbf{W}^T\mathbf{W}) = \sum_{i=1}^{k}(\mathbf{w}^i)^T\mathbf{w}^i$, where $\mathbf{w}^i$, for $i = 1, ..., k$, is a column vector of $\mathbf{W}$.

Answer

We first define two intermeidate variables,

$$a^2 = b^2 + (\mathbf{h}^1)^T\mathbf{w}^2$$
$$\mathbf{a}^1 = \mathbf{b}^1 + (\mathbf{h}^0)^T\mathbf{W}^1$$

(a) Let $g = \tanh$, then simply apply the *chain rule*,

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{w}^2} &= \frac{\partial L}{\partial g}\frac{\partial g}{\partial a^2}\frac{\partial a^2}{\partial \mathbf{w}^2} \\
&= \frac{1}{k}\sum_{i=1}^{k}\frac{\partial}{\partial g}(y^i - g(a^2))^2\frac{\partial}{\partial a^2}\tanh(a^2)\frac{\partial}{\partial \mathbf{w}^2}(b^2 + (\mathbf{h}^1)^T\mathbf{w}^2) \\
&= \frac{-2}{k}\sum_{i=1}^{k}(y^i - g(a^2))\operatorname{sech}^2(a^2)\mathbf{h}^1 \\
&= \frac{-2}{k}\sum_{i=1}^{k}\left(y^i - \tanh(b^2 + (\mathbf{h}^1)^T\mathbf{w}^2)\right)\operatorname{sech}^2\left(b^2 + (\mathbf{h}^1)^T\mathbf{w}^2\right)\mathbf{h}^1
\end{aligned}$$

Similarly,

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{h}^1} &= \frac{\partial L}{\partial g}\frac{\partial g}{\partial a^2}\frac{\partial a^2}{\partial \mathbf{h}^1} \\
&= \frac{\partial L}{\partial g}\frac{\partial g}{\partial a^2}\frac{\partial}{\partial \mathbf{h}^1}(b^2 + (\mathbf{h}^1)^T\mathbf{w}^2) \\
&= \frac{-2}{k}\sum_{i=1}^{k}\left(y^i - \tanh(b^2 + (\mathbf{h}^1)^T\mathbf{w}^2)\right)\operatorname{sech}^2\left(b^2 + (\mathbf{h}^1)^T\mathbf{w}^2\right)\mathbf{w}^2
\end{aligned}$$

(b) First, notice that $\mathbf{w}_i^1$ only has effect on $\mathbf{h}_i^1$, and so

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{w}_i^1} &= \frac{\partial L}{\partial \mathbf{h}_i^1}\frac{\partial \mathbf{h}_i^1}{\partial \mathbf{a}_i^1}\frac{\partial \mathbf{a}_i^1}{\partial \mathbf{w}_i^1} \\
&= \frac{\partial L}{\partial \mathbf{h}_i^1}\frac{\partial}{\partial \mathbf{a}_i^1}\tanh(\mathbf{a}_i^1)\frac{\partial}{\partial \mathbf{w}_i^1}\left(\mathbf{b}^1 + (\mathbf{h}^0)^T\mathbf{w}_i^1\right) \\
&= \frac{\partial L}{\partial \mathbf{h}_i^1}\operatorname{sech}^2\left(\mathbf{b}^1 + (\mathbf{h}^0)^T\mathbf{w}_i^1\right)\mathbf{h}^0
\end{aligned}$$

Next, substitute $\frac{\partial L}{\partial \mathbf{h}_i^1}$ with the i^{th} entry of $\frac{\partial L}{\partial \mathbf{h}^1}$ from **(a)**,

$$\frac{\partial L}{\partial \mathbf{w}_i^1} = \frac{-2}{k}\sum_{j=1}^{k}\left(y^j - \tanh(b^2 + (\mathbf{h}^1)^T\mathbf{w}^2)\right)\operatorname{sech}^2\left(b^2 + (\mathbf{h}^1)^T\mathbf{w}^2\right)\mathbf{w}_i^2\operatorname{sech}^2\left(\mathbf{b}^1 + (\mathbf{h}^0)^T\mathbf{w}_i^1\right)\mathbf{h}^0$$

A key point here is to recognize that

$$c = \frac{-2}{k}\sum_{j=1}^{k}\left(y^j - \tanh(b^2 + (\mathbf{h}^1)^T\mathbf{w}^2)\right)\operatorname{sech}^2\left(b^2 + (\mathbf{h}^1)^T\mathbf{w}^2\right)$$

is a scalar, and the i^{th} entry of $\frac{\partial L}{\partial \mathbf{h}_i^1} = c\mathbf{w}^2$ is simply $c\mathbf{w}_i^2$. The overall expression of $\frac{\partial L}{\partial \mathbf{W}^1}$ is given by,

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{W}^1} &= \begin{bmatrix} | & & | \\ \frac{\partial L}{\partial \mathbf{w}_1^1} & \cdots & \frac{\partial L}{\partial \mathbf{w}_M^1} \\ | & & | \end{bmatrix} \\
&= \begin{bmatrix} | & & | \\ c\mathbf{w}_1^2\operatorname{sech}^2\left(\mathbf{b}^1 + (\mathbf{h}^0)^T\mathbf{w}_1^1\right)\mathbf{h}^0 & \cdots & c\mathbf{w}_M^2\operatorname{sech}^2\left(\mathbf{b}^1 + (\mathbf{h}^0)^T\mathbf{w}_M^1\right)\mathbf{h}^0 \\ | & & | \end{bmatrix} \\
&= \underbrace{\left[\frac{-2}{k}\sum_{j=1}^{k}\left(y^j - \tanh(b^2 + (\mathbf{h}^1)^T\mathbf{w}^2)\right)\operatorname{sech}^2\left(b^2 + (\mathbf{h}^1)^T\mathbf{w}^2\right)\right]}_{\text{scalar}}\underbrace{\mathbf{h}^0}_{\text{vector}}\underbrace{\left[\mathbf{w}^2 \circ \operatorname{sech}^2(\mathbf{b}^1 + (\mathbf{h}^0)^T\mathbf{W}^1)\right]^T}_{\text{row vector}}
\end{aligned}$$

where $\circ$ denotes the entrywise product (also known as the Schur product or the Hadamard product).

(c)

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{w}^2}L_r &= \frac{\partial}{\partial \mathbf{w}^2}L + \frac{\partial}{\partial \mathbf{w}^2}\|\mathbf{W}^1\|_F^2 + \frac{\partial}{\partial \mathbf{w}^2}\|\mathbf{w}^2\|_2^2 \\
&= \frac{\partial}{\partial \mathbf{w}^2}L + 2\mathbf{w}^2
\end{aligned}$$

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{W}^1}L_r &= \frac{\partial}{\partial \mathbf{W}^1}L + \frac{\partial}{\partial \mathbf{W}^1}\|\mathbf{W}^1\|_F^2 + \frac{\partial}{\partial \mathbf{W}^1}\|\mathbf{w}^2\|_2^2 \\
&= \frac{\partial}{\partial \mathbf{W}^1}L + 2\mathbf{W}^1
\end{aligned}$$