CNN HW2-

1. c. let's set activation function to be ReLU and the layer size as 2, then $w_i \in M_{2,2}^{(d)}\ i \in \{1, \dots, d-1\}$ and $w_d \in \mathbb{R}_2$.

Now we will define the empirical loss as a function of as $E(w)$.

If we find w_1, w_2 with $E(w_1) = E(w_2) = 0$ (no loss) then we have $tE(w_1) + (1-t)E(w_2) = 0$ and if for some t the loss $E(tw_1 + (1-t)w_2) \neq 0$ then we are done.

Notice that if we set the first d-2 layers to be the identity transformation:

$$w_{i,j} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\ i \in \{1,2\}\ j \in \{1, \dots, d-1\}$$

then after applying ReLU on $w_{i,j}x$ and x is positive we still have the identity function.
from the above claim we get that as long as $x > 0$ we can choose $d \geq 2$ as we like and generalize the claim to every $d' > d$ by setting the first $d' - d$ layers to be the identity (if $d = 1$ the claim is incorrect as $f_w(x)$ is just a linear transformation of x and the logistic loss is convex in w) transformation.

Set $d = 2$

Now let's look at the following counter example, we choose the dataset and the classifiers

w_1 and w_2 and show that the loss is not convex for these examples:

$$S = \left\{\left(x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, y = -1\right)\right\}, \qquad w_1 = \left(\begin{pmatrix} -5 & -5 \\ 5 & 5 \end{pmatrix}, \begin{pmatrix} 5 \\ 0 \end{pmatrix}\right)\ w_2 = \left(\begin{pmatrix} 5 & -5 \\ -5 & -10 \end{pmatrix}, \begin{pmatrix} -10 \\ 5 \end{pmatrix}\right)$$

we have $m = 1$ so $E(w) = l(f_w(x), y)$

Output from classifier:

$$\mathrm{w}_{1,2}\max(w_{1,1}x, 0) = w_{1,2}\max\left(\begin{pmatrix} -10 \\ 10 \end{pmatrix}, \mathbf{0}\right) = w_{1,2}\begin{pmatrix} 0 \\ 10 \end{pmatrix} = 0$$

$$\mathrm{w}_{2,2}\max(w_{2,1}x, 0) = w_{2,2}\max\left(\begin{pmatrix} 0 \\ -15 \end{pmatrix}, \mathbf{0}\right) = w_{1,2}\mathbf{0} = 0$$

Loss:

$$\log\left(1 + e^{\mathrm{w}_{1,2}\max(w_{1,1}x, \mathbf{0})}\right) = \log\left(1 + e^{\mathrm{w}_{2,2}\max(w_{2,1}x, \mathbf{0})}\right) = \log(2)$$

Now we define a new classifier like this - $w' = tw_1 + (1-t)w_2$ and choose $t = \frac{4}{5}$

$$w' = tw_1 + (1-t)w_2 = \frac{4}{5}w_1 + \frac{1}{5}w_2 = \left(\begin{pmatrix} -3 & -5 \\ 3 & 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix}\right)$$

the output for the classifier is:

$$\mathrm{w}_2'\max(w''_1 x, 0) = w_2'\max\left(\begin{pmatrix} -8 \\ 5 \end{pmatrix}, \mathbf{0}\right) = w_2'\begin{pmatrix} 0 \\ 5 \end{pmatrix} = 5$$

And the loss is:

$$E\left(\frac{4}{5}w_1 + \frac{1}{5}w_2\right) = E(w') = (\log\left(1 + e^{\mathrm{w}_2'\max(w_1'x, \mathbf{0})}\right) = \log(1 + e^5) > \log(2)$$
$$= \frac{4}{5}E(w_1) + \frac{1}{5}E(w_2)$$

Hence the empirical loss is non convex with respect to w

2. we will compute the gradient of $\left\|W_3\left(\sigma\left(W_2\left(\sigma(W_1\boldsymbol{x})\right)\right)\right)-\boldsymbol{y}\right\|_2^2$ step by step.

mark the dimensions:

$$d(x)=n_x \;\; d(W_1)=(n_1,n_x) \quad d(W_2)=(n_2,n_1) \quad d(W_3)=\left(n_y,n_2\right) \quad d(\boldsymbol{y})=n_y$$

first let's define $L_i(\boldsymbol{x})=W_i\boldsymbol{x}$ and we get:

$$||L_3\left(\sigma\left(L_2\left(\sigma\left(L_1(\boldsymbol{x})\right)\right)\right)\right)-\boldsymbol{y}||_2^2$$

Let's write the analytical derivatives we will use:

$$\frac{\partial}{\partial\boldsymbol{x}}||\boldsymbol{x}-\boldsymbol{y}||^2=2\boldsymbol{x}$$

$$\frac{\partial L_i(\boldsymbol{x})}{\partial\boldsymbol{x}}=W_i$$

We'll mark $\boldsymbol{w_{i,r}}$ as the r-th row of matrix W_i and compute the gradient row wise

$$\frac{\partial L_i(\boldsymbol{x})}{\partial\boldsymbol{w_{i,r}}}=\begin{array}{c}0\\ \vdots\\ r\\ \vdots\\ n_i\end{array}\begin{bmatrix}0 & & \cdots & 0\\ \vdots & \vdots & \vdots & \vdots\\ x_0 & x_1 & \dots & x_n\\ \vdots & \vdots & \vdots & \vdots\\ 0 & & \cdots & 0\end{bmatrix}$$

When x is a scalar we can use the following identity:

$$\frac{\partial\sigma(x)}{\partial x}=\sigma(x)\left(1-\sigma(x)\right)=\frac{1}{1+e^{-x}}\left(\frac{e^{-x}}{1+e^{-x}}\right)=\frac{e^{-x}}{1+2e^{-x}+e^{-2x}}$$

and when x is a vector of length n we get:

$$\frac{\partial\sigma(\boldsymbol{x})}{\partial\boldsymbol{x}}=\begin{bmatrix}\frac{\partial\sigma(x_0)}{\partial x_0} & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \frac{\partial\sigma(x_n)}{\partial x_n}\end{bmatrix}=\begin{bmatrix}\frac{e^{-x_0}}{1+2e^{-x_0}+e^{-2x_0}} & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \frac{e^{-x_n}}{1+2e^{-x_n}+e^{-2x_n}}\end{bmatrix}$$

All of the above involves no computation.

Now we start computing the gradients, we make a forward pass and save the intermediate results of the form $\frac{\partial\sigma(x)}{\partial x}$. (no need to save $\frac{\partial L_i(x)}{\partial x}$ as we saw earlier that $\frac{\partial L_i(x)}{\partial x}=W_i$ and we have that from the net state).

This takes $O(n_x n_1+n_1 n_2+n_2 n_y)$ time.

saving the intermediate results will take $O(n_1+n_2+n_y)$ space

for comfort we will mark the output of the t-th sigmoid layer as z_t

Now we will compute the gradients backward using the chain rule and save intermediate matrix multiplication that we will use in the future from each calculation

Gradients w.r.t W_3:

$$\frac{\partial}{\partial\boldsymbol{w_{3,r}}}||L_3(\boldsymbol{z_2})-\boldsymbol{y}||_2^2=\frac{\partial||L_3(z_2)-\boldsymbol{y}||_2^2}{\partial L_3(z_2)}\frac{\partial L_3(z_2)}{\partial\boldsymbol{w_{3,r}}}$$

For every $\frac{\partial}{\partial w_{3,r}}||L_3(\boldsymbol{z_2}) - \boldsymbol{y}||_2^2$ calculation we multiply a vector by a sparse matrix where only the r-th row is non zero, basically we multiply the r-th row by the r-th index of the vector this takes, $O(n_2)$ time

We will do this n_y time so overall $O(n_y^2 n_2)$ time

Gradients w.r.t W_2:

$$\frac{\partial}{\partial \boldsymbol{w_{2,r}}}||L_3(\boldsymbol{z_2}) - \boldsymbol{y}||_2^2 = \frac{\partial||L_3(\boldsymbol{z_2}) - \boldsymbol{y}||_2^2}{\partial L_3(\boldsymbol{z_2})}\frac{\partial L_3(\boldsymbol{z_2})}{\partial \boldsymbol{z_2}}\frac{\partial \sigma\big(L_2(\boldsymbol{z_1})\big)}{\partial L_2(\boldsymbol{z_1})}\frac{\partial L(\boldsymbol{z_1})}{\partial \boldsymbol{w_{2,r}}}$$

We need to compute $\left(\frac{\partial||L_3(\boldsymbol{z_2})-\boldsymbol{y}||_2^2}{\partial L_3(\boldsymbol{z_2})}\frac{\partial L_3(\boldsymbol{z_2})}{\partial \boldsymbol{z_2}}\right)\frac{\partial \sigma(L_2(\boldsymbol{z_1}))}{\partial L_2(\boldsymbol{z_1})}$ once and save it ($O(n_2)$ space) for later use, this is done in $O(n_y n_2)$ as the last multiplication is vector by a diagonal matrix.
Than we multiply the result by the final part for every r (n_2 times) in $O(n_2 n_1)$ as $\frac{\partial L(\boldsymbol{z_1})}{\partial w_{2,r}}$ is mostly zeros except row r , overall we have $O(n_2 n_1 + n_y n_2)$ for this part

Gradients w.r.t W_1:

$$\frac{\partial}{\partial \boldsymbol{w_{1,r}}}||L_3(\boldsymbol{z_2}) - \boldsymbol{y}||_2^2 = \frac{\partial||L_3(\boldsymbol{z_2}) - \boldsymbol{y}||_2^2}{\partial L_3(\boldsymbol{z_2})}\frac{\partial L_3(\boldsymbol{z_2})}{\partial \boldsymbol{z_2}}\frac{\partial \sigma\big(L_2(\boldsymbol{z_1})\big)}{\partial L_2(\boldsymbol{z_1})}\frac{\partial L(\boldsymbol{z_1})}{\partial \boldsymbol{z_1}}\frac{\partial \sigma\big(L_1(\boldsymbol{x})\big)}{\partial L_1(\boldsymbol{x})}\frac{\partial L_1(\boldsymbol{x})}{\partial \boldsymbol{w_{1,r}}}$$

We already calculated $\frac{\partial||L_3(\boldsymbol{z_2})-\boldsymbol{y}||_2^2}{\partial L_3(\boldsymbol{z_2})}\frac{\partial L_3(\boldsymbol{z_2})}{\partial \boldsymbol{z_2}}\frac{\partial \sigma(L_2(\boldsymbol{z_1}))}{\partial L_2(\boldsymbol{z_1})}$ so in order to calculate $\frac{\partial||L_3(\boldsymbol{z_2})-\boldsymbol{y}||_2^2}{\partial L_3(\boldsymbol{z_2})}\frac{\partial L_3(\boldsymbol{z_2})}{\partial \boldsymbol{z_2}}\frac{\partial \sigma(L_2(\boldsymbol{z_1}))}{\partial L_2(\boldsymbol{z_1})}\frac{\partial L(\boldsymbol{z_1})}{\partial \boldsymbol{z_1}}\frac{\partial \sigma(L_1(\boldsymbol{x}))}{\partial L_1(\boldsymbol{x})}$ we only need 2 more martrix multiplications where one is diagonal. So similarly to last step (with different dimensions) we need to perform $O(n_2 n_1)$ calculations and then $O(n_1 n_x)$ for a total of $O(n_2 n_1 + n_1 n_x)$.
we saved one vector of length n_1 so $O(n_1)$ space.

Let's sum It all up: $O\big(n_1 + n_2 + n_x + n_y\big)$ space, $O(n_1 n_2 + n_1 n_x + n_y n_2)$