Due Date : February 16th, 2019

Instructions

- *For all questions, show your work!*
- *Use a document preparation system such as LaTeX.*
- *Submit your answers electronically via the course studium page, and via Gradescope.*

Question 1. Using the following definition of the derivative and the definition of the Heaviside step function :

$$\frac{d}{dx}f(x) = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon} \qquad H(x) = \begin{cases} 1 & \text{if } x > 0 \\ \frac{1}{2} & \text{if } x = 0 \\ 0 & \text{if } x < 0 \end{cases}$$

1. Show that the derivative of the rectified linear unit $g(x) = \max\{0, x\}$, **wherever it exists**, is equal to the Heaviside step function.
2. Give two alternative definitions of $g(x)$ using $H(x)$.
3. Show that $H(x)$ can be well approximated by the sigmoid function $\sigma(x) = \frac{1}{1+e^{-kx}}$ asymptotically (i.e for large k), where k is a parameter.

*4. Although the Heaviside step function is not differentiable, we can define its **distributional derivative**. For a function F, consider the functional $F[\phi] = \int_{\mathbb{R}} F(x)\phi(x)dx$, where ϕ is a smooth function (infinitely differentiable) with compact support ($\phi(x) = 0$ whenever $|x| \geq A$, for some $A > 0$).

Show that whenever F is differentiable, $F'[\phi] = -\int_{\mathbb{R}} F(x)\phi'(x)dx$. Using this formula as a definition in the case of non-differentiable functions, show that $H'[\phi] = \phi(0)$. ($\delta[\phi] \doteq \phi(0)$ is known as the Dirac delta function.)

Answer 1. Write your answer here.

1. the derivative of the rectified linear unit g(x) according to the following definition of the derivative :

$$\frac{d}{dx}f(x) = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon}$$

$$\frac{dg(x)}{d(x)} = \begin{cases} \lim\limits_{\epsilon \to 0} \frac{max(0,x+\epsilon) - max(0,x)}{\epsilon} = \lim\limits_{\epsilon \to 0} \frac{(x+\epsilon) - x}{\epsilon} = 1, & \text{if x>0} \\ \lim\limits_{\epsilon \to 0} \frac{max(0,x+\epsilon) - max(0,x)}{\epsilon} = \lim\limits_{\epsilon \to 0} \frac{0-0}{\epsilon} = 0, & \text{if x<0} \\ \text{not exists,} & \text{if x=0} \end{cases}$$

We knew,

$$H(x) = \begin{cases} 1 & \text{if x>0, g(x)=x} \\ \frac{1}{2} & \text{if x =0} \\ 0 & \text{if x<=0, g(x)=0} \end{cases}$$

So wherever it exists, it equals to the Heaviside step function.

2. As we know :

$$H(x) = \begin{cases} 1 & \text{if } x > 0 \\ \frac{1}{2} & \text{if } x = 0 \\ 0 & \text{if } x < 0 \end{cases}$$

definition 1 :

$Relu(x) = max(0, x) = x * H(x)$

definition 2 :

$Relu(x) = max(0, x) = x * (1 - H(-x))$

3. H(x) can be well approximated by the sigmoid $\phi(x) = \frac{1}{1+e^{-kx}}$ asymptotically while $k \to +\infty$:

when $x > 0$, $k \to +\infty$,$kx \to \infty$, $e^{-kx} \to 0$, $\phi(x) = \frac{1}{1+e^{-kx}} \to 1$

when $x < 0$, $k \to +\infty$,$kx \to -\infty$, $e^{-kx} \to \infty$, $\phi(x) = \frac{1}{1+e^{-kx}} \to 0$

when $x = 0$, $k \to +\infty$,$kx = 0$, $e^{-kx} = 1$, $\phi(x) = \frac{1}{1+e^{-kx}} = \frac{1}{2}$

4. Because :

$d(uv) = udv + vdu \Rightarrow \int d(uv) = \int udv + \int vdu \Rightarrow uv = \int udv + \int vdu \Rightarrow \int udv = uv - \int vdu$

When F is differentiable, we set :

$u = \phi(x), dv = F'(x)dx$

$du = \phi'(x), v = F(x),$

$(F', \phi) = \int_R F'(x)\phi(x)dx = \phi(x)F(x)|_{-\infty}^{+\infty} - \int_R F(x)\phi'(x)dx$

Because $\phi(x)$ is a smooth function with compact support : $\phi(x)F(x)|_{-\infty}^{+\infty} = 0$, then

$(F', \phi) = -\int_R F(x)\phi'(x)dx = -(F, \phi')$

So when $F[\phi] = \int_R F(x)\phi(x)dx$, then we could know :

$F'[\phi] = -\int_R F(x)\phi'(x)dx$

Using this formula as a definition in the case of non-differentiable functions,

$H'[\phi] = -\int_R H(x)\phi'(x)dx$

$= -\int_0^{+\infty} H(x)\phi'(x)dx$

$= -\phi(x)|_0^{+\infty}$

$= \phi(0)$

Question 2. Let x be an n-dimentional vector. Recall the softmax function : $S : \boldsymbol{x} \in \mathbb{R}^n \mapsto S(\boldsymbol{x}) \in \mathbb{R}^n$ such that $S(\boldsymbol{x})_i = \frac{e^{\boldsymbol{x}_i}}{\sum_j e^{\boldsymbol{x}_j}}$; the diagonal function : $\text{diag}(\boldsymbol{x})_{ij} = \boldsymbol{x}_i$ if $i = j$ and $\text{diag}(\boldsymbol{x})_{ij} = 0$ if $i \neq j$; and the Kronecker delta function : $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$.

1. Show that the derivative of the softmax function is $\frac{dS(\boldsymbol{x})_i}{d\boldsymbol{x}_j} = S(\boldsymbol{x})_i\,(\delta_{ij} - S(\boldsymbol{x})_j)$.
2. Express the Jacobian matrix $\frac{\partial S(\boldsymbol{x})}{\partial \boldsymbol{x}}$ using matrix-vector notation. Use $\text{diag}(\cdot)$.
3. Compute the Jacobian of the sigmoid function $\sigma(\boldsymbol{x}) = 1/(1 + e^{-\boldsymbol{x}})$.
4. Let $\boldsymbol{y}$ and $\boldsymbol{x}$ be $n-$dimensional vectors related by $\boldsymbol{y} = f(\boldsymbol{x})$, L be an unspecified differentiable loss function. According to the chain rule of calculus, $\nabla_{\boldsymbol{x}} L = (\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}})^\top \nabla_{\boldsymbol{y}} L$, which takes up $\mathcal{O}(n^2)$ computational time in general. Show that if $f(\boldsymbol{x}) = \sigma(\boldsymbol{x})$ or $f(\boldsymbol{x}) = S(\boldsymbol{x})$, the above matrix-vector multiplication can be simplified to a $\mathcal{O}(n)$ operation.

Answer 2. Write your answer here.

1. There are two situations :

if $i = j, \delta_{ij} = 1$:

$\frac{dS(x)_i}{dx_j} = \frac{\partial}{\partial x_j}(e^{x_i}((\sum_j e^{x_j})^{-1} = (e^{x_i})' * (\sum_j e^{x_j})^{-1} + (e^{x_i})(\sum_j e^{x_j})^{-1})' = e^{x_i}(\sum_j e^{x_j})^{-1} + e^{x_i}(-(\sum_j e^{x_j})^{-2})e^{x_j} = \frac{e^{x_i}}{\sum_j e^{x_j}} - \frac{e^{x_i}e^{x_j}}{(\sum_j e^{x_j})^2} = \frac{e^{x_i}}{\sum_j e^{x_j}} - \frac{(e^{x_j})^2}{(\sum_j e^{x_j})^2} = \frac{e^{x_i}}{\sum_j e^{x_j}}(1 - \frac{e^{x_i}}{\sum_j e^{x_j}}) = S(x)_i(1 - S(x)_j) = S(x)_i(\delta_{ij} - S(x)_j)$

if $i \neq j, \delta_{ij} = 0$:

$\frac{dS(x)_i}{dx_j} = \frac{\partial}{\partial x_j}(e^{x_i}((\sum_j e^{x_j})^{-1} = (e^{x_i})' * (\sum_j e^{x_j})^{-1} + (e^{x_i})(\sum_j e^{x_j})^{-1})' = -e^{x_i}(\sum_j e^{x_j})^{-2}e^{x_j} = \frac{e^{x_i}}{\sum_j e^{x_j}}(0 - \frac{e^{x_j}}{\sum_j e^{x_j}} = S(x)_i(\delta_{ij} - S(x)_j)$

Shows that the derivative of the softmax function is $\frac{dS(x)_i}{dx_j} = S(x)_i(\delta_{ij} - S(x)_j)$

2. Jacobian matrix $\frac{\partial S(x)}{\partial (x)}$:

$$\begin{bmatrix} \frac{\partial S(x)_1}{\partial x_1} & \cdots & \frac{\partial S(x)_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial S(x)_n}{\partial x_1} & \cdots & \frac{\partial S(x)_n}{\partial x_n} \end{bmatrix} = \begin{bmatrix} S(x)_1(1 - S(x)_1) & -S(x)_1 S(x)_2 & \cdots & -S(x)_1 S(x)_n \\ -S(x)_2 S(x)_1 & S(x)_2(1 - S(x)_2) & \cdots & -S(x)_2 S(x)_n \\ \vdots & \vdots & \ddots & \vdots \\ -S(x)_n S(x)_1 & -S(x)_n S(x)_2 & \cdots & S(x)_n(1 - S(x)_n) \end{bmatrix} =$$

$$\begin{bmatrix} S(x)_1 - (S(x)_1)^2 & -S(x)_1 S(x)_2 & \cdots & -S(x)_1 S(x)_n \\ -S(x)_2 S(x)_1 & S(x)_2 - (S(x)_2)^2 & \cdots & -S(x)_2 S(x)_n \\ \vdots & \vdots & \ddots & \vdots \\ -S(x)_n S(x)_1 & -S(x)_n S(x)_2 & \cdots & S(x)_n - (S(x)_n)^2 \end{bmatrix} = diag(S(x)) - S(x)S(x)^T$$

3. Derivative of the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$:

$$\frac{\partial \sigma(x_i)}{\partial x_j} = \begin{cases} \sigma(x_i)(1-\sigma(x_i)) & \textit{if } i = j \\ 0 & \textit{if } \mathrm{i} \neq j \end{cases}$$

The Jacobian of the sigmoid function :

$$\begin{bmatrix} \frac{\partial \sigma(x)_1}{\partial x_1} & \cdots & \frac{\partial \sigma(x)_n}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \sigma(x)_n}{\partial x_1} & \cdots & \frac{\partial \sigma(x)_n}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \sigma(x_1)(1-\sigma(x_1)) & 0 & \cdots & 0 \\ 0 & \sigma(x_2)(1-\sigma(x_2)) & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma(x_n)(1-\sigma(x_n)) \end{bmatrix}$$

So, for element-wise to the vector : $\frac{\partial \sigma(x)}{\partial x} = diag(\sigma(x) \odot (I - \sigma(x)))$

$I = (1, ..., 1)^T \in R^n$

4. $\nabla_x L = (\frac{\partial y}{\partial x})^T \nabla_y L = (\begin{matrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial x_1} & \cdots & \frac{\partial y_n}{\partial x_n} \end{matrix})^T (\begin{matrix} \frac{\partial L}{\partial y_1} \\ \vdots \\ \frac{\partial L}{\partial y_n} \end{matrix})$

- if $y = f(x) = \sigma(x)$, from 2.3, we know :

$\nabla_x L = (\frac{\partial y}{\partial x})^T \nabla_y L = (diag(\sigma(x) \odot (1 - \sigma(x))))\nabla_y L) = (\sigma(x) \odot (1-\sigma(x))) \odot \nabla_y L$

It shows that the matrix-vector multiplication could operate by element-wise product which takes up O(n) computational time.

- If $y = f(x) = S(x)$, from 2.2, we know :

$\nabla_x L = (diag(S(x)) - S(x)S(x)^T)\nabla_y L$

$=$ diag(S(x))$\nabla_y L - S(x)S(x)^T \nabla_y L$

$=$ S(x)$\odot \nabla_y L - S(x) < S(x), \nabla_y L >$

It shows that the matrix-vector multiplication could be operate by element-wise product which takes up O(n) computational time, and inner product operation is scalar, scalar-vector product takes up O(n) computational time. So totally takes up O(n) computational time.

Question 3. Recall the definition of the softmax function : $S(\boldsymbol{x})_i = e^{\boldsymbol{x}_i} / \sum_j e^{\boldsymbol{x}_j}$.

1. Show that softmax is translation-invariant, that is : $S(\boldsymbol{x}+c) = S(\boldsymbol{x})$, where c is a scalar constant.
2. Show that softmax is not invariant under scalar multiplication. Let $S_c(\boldsymbol{x}) = S(c\boldsymbol{x})$ where $c \geq 0$. What are the effects of taking c to be 0 and arbitrarily large ?
3. Let $\boldsymbol{x}$ be a 2-dimentional vector. One can represent a 2-class categorical probability using softmax $S(\boldsymbol{x})$. Show that $S(\boldsymbol{x})$ can be reparameterized using sigmoid function, i.e. $S(\boldsymbol{x}) = [\sigma(z), 1-\sigma(z)]^\top$ where z is a scalar function of $\boldsymbol{x}$.

4. Let $\boldsymbol{x}$ be a K-dimentional vector ($K \geq 2$). Show that $S(\boldsymbol{x})$ can be represented using $K-1$ parameters, i.e. $S(\boldsymbol{x}) = S([0, y_1, y_2, ..., y_{K-1}]^\top)$ where y_i is a scalar function of $\boldsymbol{x}$ for $i \in \{1, ..., K-1\}$.

Answer 3. Write your answer here.

1. $S(x+c) = \frac{e^{x_i+c}}{\sum_j e^{x_j+c}} = \frac{e^{x_i}e^c}{\sum_j e^{x_j}e^c} = \frac{e^{x_i}}{\sum_j e^{x_j}} = S(x)$

It show that softmax is translation-invariant.

2. $S_c(x) = S(cx) = \frac{(e^{x_i})^c}{(\sum_j e^{x_j})^c} = \frac{1}{\sum_j e^{c(x_j - x_i)}}$

It shows that softmax is not invariant under scalar multiplication.

While $c \to 0$: $S_c(x) = S(x)^c = \frac{1}{\sum_j e^{c(x_j-x_i)}} = 1/n$

While $c \to \infty$: $S_c(x) = S(x)^c = \frac{1}{\sum_j \lim_{c\to+\infty} e^{c(x_j-x_i)}}$

if $x_j - x_i < 0, lim_{c\to+\infty} e^{c(x_j-x_i)} \to 0,$

if $x_j - x_i = 0, lim_{c\to+\infty} e^{c(x_j-x_i)} = 1,$

if $x_j - x_i > 0, lim_{c\to+\infty} e^{c(x_j-x_i)} \to +\infty$

So, $\frac{1}{\sum_j \lim_{c\to+\infty} e^{c(x_j-x_i)}}$ depends on the value of $(x_j - x_i)$.

If there exists $x_j - x_i > 0$, then the $\frac{1}{\sum_j \lim_{c\to+\infty} e^{c(x_j-x_i)}} \to 0$;

If not exist $x_j - x_i > 0$, but there are k $x_j = x_i$, then$\frac{1}{\sum_j \lim_{c\to+\infty} e^{c(x_j-x_i)}} = \frac{1}{k}$

3. As we know, softmax S(x) represent a 2-class categorical probability :

$S(x) = [\frac{e^{x_1}}{e^{x_1}+e^{x_2}}, \frac{e^{x_2}}{e^{x_1}+e^{x_2}}]^T$, while $z = x_1 - x_2$:

$\frac{e^{x_1}}{e^{x_1}+e^{x_2}} = \frac{1}{\frac{e^{x_1}+e^{x_2}}{e^{x_1}}} = \frac{1}{1+\frac{e^{x_2}}{e^{x_1}}} = \frac{1}{1+e^{(x_2-x_1)}} = \frac{1}{1+e^{-(x_1-x_2)}} = \frac{1}{1+e^{-z}} = \phi(z)$

$\frac{e^{x_2}}{e^{x_1}+e^{x_2}} = \frac{e^{x_1}+e^{x_2}-e^{x_1}}{e^{x_1}+e^{x_2}} = 1 - \frac{e^{x_1}}{e^{x_1}+e^{x_2}} = 1 - \phi(z),$

So we could show that $S(x) = [\phi(z), 1-\phi(z)]^T$

4. According to 3.1, we know softmax is translation-invariant, $S(x+c) = S(x)$, we set $c = -x_0$, then :

$S(x) = S(x+c) = S(x-x_0) = S([x_0 - x_0, x_1 - x_0, ..., x_{k-1} - x_0]^T)$

let $y_i = x_i - x_0$, where $1 \leq i \leq k-1$, then : $S(x) = ([0, y_1, y_2, ..., y_{k-1}]^T)$, it shows that S(x) can be represented using K-1 parameters.

Question 4. Consider a 2-layer neural network $y : \mathbb{R}^D \to \mathbb{R}^K$ of the form :

$$y(x, \Theta, \sigma)_k = \sum_{j=1}^{M} \omega_{kj}^{(2)} \sigma \left(\sum_{i=1}^{D} \omega_{ji}^{(1)} x_i + \omega_{j0}^{(1)} \right) + \omega_{k0}^{(2)}$$

for $1 \leq k \leq K$, with parameters $\Theta = (\omega^{(1)}, \omega^{(2)})$ and logistic sigmoid activation function σ. Show that there exists an equivalent network of the same form, with parameters $\Theta' = (\tilde{\omega}^{(1)}, \tilde{\omega}^{(2)})$ and tanh activation function, such that $y(x, \Theta', \tanh) = y(x, \Theta, \sigma)$ for all $x \in \mathbb{R}^D$, and express Θ' as a function of Θ.

Answer 4. Write your answer here.

The connection between sigmoid and tanh :

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

$tanh(x) = \frac{e^x - e^{-x}}{e^x+e^{-x}} = \frac{e^{2x}-1}{e^{2x}+1}$

So, $1 - 2\sigma(x) = -tanh(\frac{x}{2})$, $\sigma(x) = \frac{1+tanh(\frac{x}{2})}{2}$

$\sigma\left(\sum_{i=1}^{D} \omega_{ji}^{(1)} x_i + \omega_{j0}^{(1)}\right) = \sum_{i=1}^{D} \omega_{ji}^{(1)} (\frac{1+tanh(\frac{x_i}{2})}{2}) + \omega_{j0}^{(1)}$

$y(x, \Theta, \sigma)_k = \sum\limits_{j=1}^{M} \omega_{kj}^{(2)} \sigma \left(\sum\limits_{i=1}^{D} \omega_{ji}^{(1)} x_i + \omega_{j0}^{(1)} \right) + \omega_{k0}^{(2)} = \sum\limits_{j=1}^{M} \omega_{kj}^{(2)} (\sum\limits_{i=1}^{D} \omega_{ji}^{(1)} (\frac{1+tanh(\frac{x_i}{2})}{2}) + \omega_{j0}^{(1)}) + \omega_{k0}^{(2)} =$ $\sum\limits_{j=1}^{M} \omega_{kj}^{(2)} (\frac{1+tanh(\frac{\sum\limits_{i=1}^{D} \omega_{ji}^{(1)} x_i + \omega_{j0}^{(1)}}{2})}{2}) + \omega_{k0}^{(2)} = \frac{1}{2}\sum\limits_{j=1}^{M} \omega_{kj}^{(2)} + \frac{1}{2}\sum\limits_{j=1}^{M} \omega_{kj}^{(2)} tanh(\frac{\sum\limits_{i=1}^{D} \omega_{ji}^{(1)} x_i + \omega_{j0}^{(1)}}{2}) + \omega_{k0}^{(2)}$

So there exists an equivalent network of the same form with parameters $\Theta' = (\tilde{\omega^{(1)}}, \tilde{\omega^{(2)}})$ and the tanh activation function, such that $y(x, \Theta', tanh) = y(x, \Theta, \sigma)$ for all x$\in R^D$.Θ' is expressed as : $\tilde{\omega^{(1)}} = [\frac{1}{2}\sum\limits_{i=1}^{D} \omega_{ji}^{(1)}], \tilde{b^{(1)}} = [\frac{1}{2}\omega_{j0}^{(1)}]$; $\tilde{\omega^{(2)}} = [\frac{1}{2}\sum\limits_{j=1}^{M} \omega_{kj}^{(2)}], \tilde{b^{(2)}} = [\omega_{k0}^{(2)} + \frac{1}{2}\sum\limits_{j=1}^{M} \omega_{kj}^{(2)}]$

Question 5. Given $N \in \mathbb{Z}^+$, we want to show that for any $f : \mathbb{R}^n \to \mathbb{R}^m$ and any sample set $\mathcal{S} \subset \mathbb{R}^n$ of size N, there is a set of parameters for a two-layer network such that the output $y(\boldsymbol{x})$ matches $f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{S}$. That is, we want to interpolate f with y on any finite set of samples $\mathcal{S}$.

1. Write the generic form of the function $y : \mathbb{R}^n \to \mathbb{R}^m$ defined by a 2-layer network with $N-1$ hidden units, with linear output and activation function ϕ, in terms of its weights and biases $(\boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)})$ and $(\boldsymbol{W}^{(2)}, \boldsymbol{b}^{(2)})$.
2. In what follows, we will restrict $\boldsymbol{W}^{(1)}$ to be $\boldsymbol{W}^{(1)} = [\boldsymbol{w}, \cdots, \boldsymbol{w}]^T$ for some $\boldsymbol{w} \in \mathbb{R}^n$ (so the rows of $\boldsymbol{W}^{(1)}$ are all the same). Show that the interpolation problem on the sample set $\mathcal{S} = \{\boldsymbol{x}^{(1)}, \cdots \boldsymbol{x}^{(N)}\} \subset \mathbb{R}^n$ can be reduced to solving a matrix equation : $\boldsymbol{M}\tilde{\boldsymbol{W}}^{(2)} = \boldsymbol{F}$, where $\tilde{\boldsymbol{W}}^{(2)}$ and $\boldsymbol{F}$ are both $N \times m$, given by

$$\tilde{\boldsymbol{W}}^{(2)} = [\boldsymbol{W}^{(2)}, \boldsymbol{b}^{(2)}]^\top \qquad \boldsymbol{F} = [f(\boldsymbol{x}^{(1)}), \cdots, f(\boldsymbol{x}^{(N)})]^\top$$

Express the $N \times N$ matrix $\boldsymbol{M}$ in terms of $\boldsymbol{w}$, $\boldsymbol{b}^{(1)}$, ϕ and $\boldsymbol{x}^{(i)}$.

*3. **Proof with Relu activation.** Assume $\boldsymbol{x}^{(i)}$ are all distinct. Choose $\boldsymbol{w}$ such that $\boldsymbol{w}^\top \boldsymbol{x}^{(i)}$ are also all distinct (Try to prove the existence of such a $\boldsymbol{w}$, although this is not required for the assignment - See Assignment 0). Set $\boldsymbol{b}_j^{(1)} = -\boldsymbol{w}^\top \boldsymbol{x}^{(j)} + \epsilon$, where $\epsilon > 0$. Find a value of ϵ such that $\boldsymbol{M}$ is triangular with non-zero diagonal elements. Conclude. (Hint : assume an ordering of $\boldsymbol{w}^\top \boldsymbol{x}^{(i)}$.)

*4. **Proof with sigmoid-like activations**. Assume ϕ is continuous, bounded, $\phi(-\infty) = 0$ and $\phi(0) > 0$. Decompose $\boldsymbol{w}$ as $\boldsymbol{w} = \lambda \boldsymbol{u}$. Set $\boldsymbol{b}_j^{(1)} = -\lambda \boldsymbol{u}^\top \boldsymbol{x}^{(j)}$. Fixing $\boldsymbol{u}$, show that $\lim_{\lambda \to +\infty} \boldsymbol{M}$ is triangular with non-zero diagonal elements. Conclude. (Note that doing so preserves the distinctness of $\boldsymbol{w}^\top \boldsymbol{x}^{(i)}$.)

Answer 5. Write your answer here.

1. $y(x) = W^{(2)} \phi(W^{(1)} x + b^{(1)}) + b^{(2)}$

The dimensiion :

$x : n * N$

$W^{(1)} : (N-1) * n \, ; \, b^{(1)} : (N-1) * 1$

$W^{(2)} : m * (N-1) \, ; \, b^{(2)} : m * 1$

2. We express below formula with matrix :

$$W^{(1)} = \begin{bmatrix} w_{1,1}^{(1)T} & \cdots & w_{n,1}^{(1)T} \\ \vdots & \ddots & \vdots \\ w_{1,N-1}^{(1)T} & \cdots & w_{n,N-1}^{(1)T} \end{bmatrix} ((N-1)*n), \text{ Sample set } S = \begin{bmatrix} x_1^{(1)} & \cdots & x_1^{(N)} \\ \vdots & \ddots & \vdots \\ x_n^{(1)} & \cdots & x_n^{(N)} \end{bmatrix} (n*N),$$

$$\mathrm{b}^{(1)} = \begin{bmatrix} b_1^{(1)} \\ \vdots \\ b_{N-1}^{(1)} \end{bmatrix} ((N-1)*1),$$

$$W^{(1)} S + b^{(1)} = \begin{bmatrix} w_{1,1}^{(1)T} x_1^{(1)} + b_1^{(1)} & \cdots & w_{n,1}^{(1)T} x_n^{(1)} + b_1^{(1)} \\ \vdots & \ddots & \vdots \\ w_{1,N-1}^{(1)T} x_1^{(N)} + b_{N-1}^{(1)} & \cdots & w_{n,N-1}^{(1)T} x_n^{(N)} + b_{N-1}^{(1)} \end{bmatrix} ((N-1)*N)$$

$$h_1 = \phi(W^{(1)} S + b^{(1)}) = \begin{bmatrix} \phi(w_{1,1}^{(1)T} x_1^{(1)} + b_1^{(1)}) & \cdots & \phi(w_{n,1}^{(1)T} x_n^{(N)} + b_1^{(1)}) \\ \vdots & \ddots & \vdots \\ \phi(w_{1,N-1}^{(1)T} x_1^{(1)} + b_{N-1}^{(1)}) & \cdots & \phi(w_{n,N-1}^{(1)T} x_n^{(N)} + b_{N-1}^{(1)}) \end{bmatrix} ((N-1)*N)$$

$$W^{(2)} = \begin{bmatrix} w_{1,1}^{(2)} & \cdots & w_{1,N-1}^{(2)} \\ \vdots & \ddots & \vdots \\ w_{m,1}^{(2)} & \cdots & w_{m,N-1}^{(2)} \end{bmatrix} (m*(N-1)), \, b^{(2)} = \begin{bmatrix} b_1^{(2)} \\ \vdots \\ b_m^{(2)} \end{bmatrix} (m*1),$$

$f(x) = W^{(2)}h_1 + b^{(2)} = W^{(2)}\phi(W^{(1)}S + b^{(1)}) + b^{(2)}, (m * N),$

Given by $\hat{W}^{(2)} = [W^2, b^{(2)}]^T$, $F = [f(x^{(1)}), ..., f(x^{(N)})]^T$, where $\hat{W}^{(2)}$ and F are both $N \times m$, we could get the $N \times N$ matrix M in terms of $w, b^{(1)}, \phi$ and $x^{(i)}$ to solving a matrix equation : $M\hat{W}^{(2)} = F$

$$\hat{W}^2 = \begin{bmatrix} w_{1,1}^{(2)} & \cdots & w_{m,1}^{(2)} \\ \vdots & \ddots & \vdots \\ w_{1,N-1}^{(2)} & \cdots & w_{m,N-1}^{(2)} \\ b_1^{(2)} & \cdots & b_m^{(2)} \end{bmatrix} (N * m)$$

$$F = \begin{bmatrix} w_{1,1}^{(2)}\phi(w_{1,1}^{(1)T}x_1^{(1)} + b_1^{(1)}) + b_1^{(2)} & \cdots & w_{1,N-1}^{(2)}\phi(w_{1,N-1}^{(1)T}x_1^{(N)} + b_{N-1}^{(1)}) + b_1^{(2)} \\ \vdots & \ddots & \vdots \\ w_{m,1}^{(2)}\phi(w_{n,1}^{(1)T}x_n^{(1)} + b_1^{(1)}) + b_m^{(2)} & \cdots & w_{m,N-1}^{(2)}\phi(w_{n,N-1}^{(1)T}x_n^{(N)} + b_{N-1}^{(1)}) + b_m^{(2)} \end{bmatrix}^T (N * m)$$

Then we can get $M = [(\phi(W^{(1)}S + b^{(1)})^T, 1]$, is $N \times N$ matrix, express :

$$M = \begin{bmatrix} \phi(w_{1,1}^{(1)T}x_1^{(1)} + b_1^{(1)}) & \cdots & \phi(w_{1,N-1}^{(1)T}x_1^{(1)} + b_{N-1}^{(1)}) & ,1 \\ \vdots & \ddots & \vdots & ,1 \\ \phi(w_{n,1}^{(1)T}x_n^{(N)} + b_1^{(1)}) & \cdots & \phi(w_{n,N-1}^{(1)T}x_n^{(N)} + b_{N-1}^{(1)}) & ,1 \end{bmatrix}$$

3. Set $b_j^{(1)} = -w^Tx^{(j)} + \epsilon$, where $\epsilon > 0$. Then we could get the matrix M with $b_j^{(1)}$:

$$M = \begin{bmatrix} \phi(w_{1,1}^{(1)T}x_1^{(1)} + (-w^Tx^{(1)} + \epsilon)) & \cdots & \phi(w_{1,N-1}^{(1)T}x_1^{(1)} + (-w^Tx^{(N-1)} + \epsilon)) & ,1 \\ \vdots & \ddots & \vdots & ,1 \\ \phi(w_{n,1}^{(1)T}x_n^{(N)} + (-w^Tx^{(1)} + \epsilon)) & \cdots & \phi(w_{n,N-1}^{(1)T}x_n^{(N)} + (-w^Tx^{(N-1)} + \epsilon)) & ,1 \end{bmatrix}$$

$$= \begin{bmatrix} \phi(\epsilon)) & \cdots & \phi(w_{1,N-1}^{(1)T}x_1^{(1)} + (-w^Tx^{(N-1)} + \epsilon)) & ,1 \\ \vdots & \ddots & \vdots & ,1 \\ \phi(w_{n-1,1}^{(1)T}x_{n-1}^{(N-1)} + (-w^Tx^{(1)} + \epsilon)) & \cdots & \phi(\epsilon)) & ,1 \\ \phi(w_{n,1}^{(1)T}x_n^{(N)} + (-w^Tx^{(1)} + \epsilon)) & \cdots & \phi(w_{n,N-1}^{(1)T}x_n^{(N)} + (-w^Tx^{(N-1)} + \epsilon)) & ,1 \end{bmatrix}$$

Assume and ordering of $w^Tx^{(i)}$, with Relu activation funcion : $\phi(z) = max\{0, z\}$. If we would like to make M be triangular with non-zero diagonal elements, Lower triangular should be less than 0.

$$M = \begin{bmatrix} \epsilon & \cdots & \phi(w_{1,N-1}^{(1)T}x_1^{(1)} + (-w^Tx^{(N-1)} + \epsilon)) & ,1 \\ \vdots & \ddots & \vdots & ,1 \\ 0 & \cdots & \epsilon & ,1 \\ 0 & \cdots & 0 & ,1 \end{bmatrix}$$

So we know : $\epsilon < w^T||x^{(N)} - x^{(N-1)}|| < w^T||x^{(N)} - x^{(1)}||,$

then we could get the range of ϵ : $0 < \epsilon < w^T||x^{(i)} - x^{(i-1)}||, 2 \leq i \leq N$

4. Decompose $\boldsymbol{w}$ as $\boldsymbol{w} = \lambda \boldsymbol{u}$. Set $\boldsymbol{b}_j^{(1)} = -\lambda \boldsymbol{u}^\top \boldsymbol{x}^{(j)}$. Fixing $\boldsymbol{u}$, we get the matrix M :

$$M = \begin{bmatrix} \phi((\lambda \boldsymbol{u})^T x_1^{(1)} - \lambda \boldsymbol{u}^T x^{(1)}) & \cdots & \phi((\lambda \boldsymbol{u})^T x_1^{(1)} - \lambda \boldsymbol{u}^T x^{(N-1)}) & ,1 \\ \vdots & \ddots & \vdots & ,1 \\ \phi((\lambda \boldsymbol{u})^T x_n^{(N)} - \lambda \boldsymbol{u}^T x^{(1)}) & \cdots & \phi((\lambda \boldsymbol{u})^T x_n^{(N)} - \lambda \boldsymbol{u}^T x^{(N-1)}) & ,1 \end{bmatrix}$$

Assume ϕ is continuous, bounded, $\phi(-\infty) = 0$ and $\phi(0) > 0$, and an ordering of $w^T x^i$, Fixing $\boldsymbol{u}$, while $\lim_{\lambda \to +\infty}$, M would be :

$$M = \begin{bmatrix} \phi((\lambda \boldsymbol{u})^T x_1^{(1)} - \lambda \boldsymbol{u}^T x^{(1)}) & \cdots & \phi((\lambda \boldsymbol{u})^T x_1^{(1)} - \lambda \boldsymbol{u}^T x^{(N-1)}) & ,1 \\ 0 & \cdots & \phi((\lambda \boldsymbol{u})^T x_2^{(2)} - \lambda \boldsymbol{u}^T x^{(N-1)}) & ,1 \\ \vdots & \ddots & \vdots & ,1 \\ 0 & \cdots & \phi((\lambda \boldsymbol{u})^T x_{n-1}^{(N-1)} - \lambda \boldsymbol{u}^T x^{(N-1)}) & ,1 \\ 0 & \cdots & 0 & ,1 \end{bmatrix}$$

It shows that M is triangular with non-zero diagonal elements.

Question 6. Compute the *full*, *valid*, and *same* convolution (with kernel flipping) for the following 1D matrices : $\begin{bmatrix}1, 2, 3, 4\end{bmatrix} * \begin{bmatrix}1, 0, 2\end{bmatrix}$

Answer 6. Write your answer here.

full convolution : [0, 0, 1, 2, 3, 4, 0, 0] × [2, 0, 1] = [1, 2, 5, 8, 6, 8]

valid convolution : [1, 2, 3, 4] × [2, 0, 1] = [5, 8]

same convolution : [0, 1, 2, 3, 4, 0] × [2, 0, 1] = [2, 5, 8, 6]

Question 7. Consider a convolutional neural network. Assume the input is a colorful image of size 256×256 in the RGB representation. The first layer convolves 64 8×8 kernels with the input, using a stride of 2 and no padding. The second layer downsamples the output of the first layer with a 5×5 non-overlapping max pooling. The third layer convolves 128 4×4 kernels with a stride of 1 and a zero-padding of size 1 on each border.

1. What is the dimensionality (scalar) of the output of the last layer ?
2. Not including the biases, how many parameters are needed for the last layer ?

Answer 7. Write your answer here.

1. We know the input is 3@256*256, after the first layer convolves 64 8×8.

For layer1, the dimension should be : $o = \frac{i+2p-k}{s} + 1 = \frac{256-8}{2} + 1 = 125$, 64@125*125.

Layer2, with a 5×5 non-overlapping max pooling, the dimension should be : 64@25*25

Layer3, the dimension should be : $o = \frac{i+2p-k}{s} + 1 = \frac{25+2-4}{1} + 1 = 24$

So the dimensionality (scalar) of the output of the last layer :

128 x 24 x 24 = 128 x 576 = 73728

2. With parameter sharing, it introduced F*F*D weights per filter, a total of (F*F*D)*k weights.

F : kernel size ; D : Volume of size : W x H x D ; k : filter number

So for Layer 1 : F = 8, D = 3, k = 64, there are (8*8*3)*64 = 12288 parameters

Layer 2 is pooling layer, it introduces 0 parameters

Layer3 : F = 4, D = 64, k = 128, there are (4*4*64)*128 = 131072 parameters

Not including the biases, there are 131072 parameters are needed for the last layer.

Question 8. Assume we are given data of size $3 \times 64 \times 64$. In what follows, provide the correct configuration of a convolutional neural network layer that satisfies the specified assumption. Answer with the window size of kernel (k), stride (s), padding (p), and dilation (d, with convention $d = 0$ for no dilation). Use square windows only (e.g. same k for both width and height).

1. The output shape of the first layer is $(64, 32, 32)$.
   (a) Assume $k = 8$ without dilation.
   (b) Assume $d = 7$, and $s = 2$.
2. The output shape of the second layer is $(64, 8, 8)$. Assume $p = 0$ and $d = 1$.
   (a) Specify k and s for pooling with non-overlapping window.
   (b) What is output shape if $k = 8$ and $s = 4$ instead ?
3. The output shape of the last layer is $(128, 4, 4)$.
   (a) Assume we are not using padding or dilation.
   (b) Assume $d = 2$, $p = 2$.
   (c) Assume $p = 1$, $d = 1$.

Answer 8. Write your answer here.

1. (a) From question, assume k = 8, without dilation, convolutional neural network layer transform from 3 x 64 x 64 to 64 x 32 x 32, we know :

input size : $i * i$ kernel of size : $k * k$ output size : $((i - k + 2p)/s + 1) * ((i - k + 2p)/s + 1)$

i = 64, k = 8, (i-k+2p)/s + 1 = 32, so :

$(64 - 8 + 2p)/s = 31 \Rightarrow 56 + 2p = 31s \Rightarrow p = 3, s = 2$

(b) A kernel of size k dilated by a factor d has an effective size :

$\hat{k} = k + (k - 1)(d - 1)$

the output size :

$o = \lfloor \frac{i+2p-k-(k-1)(d-1)}{s} \rfloor + 1$

as we know, $i = 64, s = 2, d = 7, i = 64$, so we can get :

$\lfloor \frac{64+2p-k-(k-1)*6}{2} \rfloor + 1 = 32 \Rightarrow \lfloor \frac{70+2p-7k}{2} \rfloor = 31$:

For example, k=2, p=3 ; k=3, p=7 ; etc

2. (a) Pooling with non-overlapping window, we could get below fomular :

$o = \lfloor \frac{i-k}{s} \rfloor + 1$, o=8, i=32, non-overlapping means $s = k$. So : $\lfloor \frac{32-k}{s} \rfloor = 7$, we could get :

$k = 4, s = 4$

(b) according to this formular :

$o = \lfloor \frac{i+2p-k-(k-1)(d-1)}{s} \rfloor + 1 = \lfloor \frac{32+2*0-8}{4} \rfloor + 1 = 6 + 1 = 7$

so, the output shape is 7*7.

3. (a) assume we are not using padding or dilation, then p =0, d=1. o=4, i=8, with below formula :
$o = \lfloor \frac{i+2p-k-(k-1)(d-1)}{s} \rfloor + 1$, we could get :

$\lfloor \frac{8+0-k}{s} \rfloor + 1 = 4 \Rightarrow \lfloor \frac{8-k}{s} \rfloor = 3$, to satisfy this, there are :

$k = 2, s = 2,\ k = 5, s = 1$

(b) Assume d=2,p=2, we could get :

$4 = \lfloor \frac{8+2*2-k-(k-1)(2-1)}{s} \rfloor + 1 \Rightarrow \lfloor \frac{13-2k}{s} \rfloor = 3$, there are :

$k = 2, s = 3\,;\ k = 3, s = 2\,;\ k = 5, s = 1$

(c) Assume p=1,d=1, we could get :

$4 = \lfloor \frac{8+2*1-k-(k-1)(1-1)}{s} \rfloor + 1 \Rightarrow \lfloor \frac{10-k}{s} \rfloor = 3$, there are :

$k = 7, s = 1$