IFT 6135 - Homework 1

Adel Nabli

16/02/2019

1 Question 1

1. We define the function g as follows: $\forall x \in \mathbb{R},\ g(x) = \max\{0, x\}$. Thus, we can write:
$$\begin{cases} \forall x > 0,\ g(x) = x \text{ and } \lim\limits_{\epsilon \to 0} \dfrac{g(x+\epsilon) - g(x)}{\epsilon} = \lim\limits_{\epsilon \to 0} \dfrac{x + \epsilon - x}{\epsilon} = 1 \\ \\ \forall x < 0,\ g(x) = 0 \text{ and } \lim\limits_{\epsilon \to 0} \dfrac{g(x+\epsilon) - g(x)}{\epsilon} = \lim\limits_{\epsilon \to 0} \dfrac{g(x+\epsilon)}{\epsilon} - 0 = 0 \end{cases}$$
For $x = 0$, the derivative is not defined as the left limit and the right limit don't give the same result. Thus, $\forall x \neq 0,\ g'(x) = H(x)$.

2. As $\{0\}$ is a countable set, $\lambda(\{0\}) = 0$ with λ the Lebesgue measure and $g' = H$ almost everywhere. Hence, as g' and $H \in \mathcal{L}^1_{loc}$, $\forall a, b \in \mathbb{R}^2$, $\int_a^b H(u) d\lambda_u = \int_a^b g'(u) d\lambda_u$. Especially, as $g(0) = 0$, we can write:
$$\forall x \in \mathbb{R}, g(x) = \int_0^x g'(u) d\lambda_u = \int_0^x H(u) d\lambda_u$$
An other way of writing g using H is the following one:
$$\forall x \in \mathbb{R}, g(x) = xH(x)$$

3. As $\lim\limits_{u \to +\infty} e^{-u} = 0$ and $\lim\limits_{u \to -\infty} e^{-u} = +\infty$, we have:
$$\begin{cases} \forall x > 0,\ \lim\limits_{k \to \infty} \sigma(kx) = 1 \\ \\ \forall x < 0,\ \lim\limits_{k \to \infty} \sigma(kx) = 0 \\ \\ \forall k,\ \sigma(0) = \dfrac{1}{1+1} = \dfrac{1}{2} \end{cases}$$
Thus, $\forall x \in \mathbb{R},\ H(x) = \lim\limits_{k \to \infty} \sigma_k(x)$ (with $\sigma_k(x) = \sigma(kx)\ \forall x$) and we have a pointwise convergence of the sequence σ_k towards H.

4. Let's take $\phi \in \mathcal{D}(\mathbb{R}) = \{f \in \mathcal{C}^\infty(\mathbb{R}),\ \textit{supp}(f) \textit{ bounded}\}$ (with $\textit{supp}(f)$ being the support of f). Hence, by integration by parts, we have:
$$\forall F \in \mathcal{C}^1(\mathbb{R}),\ -\int_{\mathbb{R}} F(x)\phi'(x) = -\underbrace{[F\phi]_{-\infty}^{\infty}}_{=0 \text{ as } supp(\phi) \text{ bounded}} + \int_{\mathbb{R}} F'(x)\phi(x)dx = F'[\phi]$$
As $\{1/2\}$ is a null set with respect to the Lebesgue measure, we have that $\int H(x)dx = \int \mathbb{1}_{x>0}(x)dx$. Thus, extending the formula for $F'[\phi]$ to non differentiable functions, we write:
$$H'[\phi] = -\int_{\mathbb{R}} H(x)\phi'(x)dx = -\int_{\mathbb{R}} \mathbb{1}_{x>0}(x)\phi'(x)dx = -\int_0^\infty \phi'(x)dx = -\underbrace{\phi(\infty)}_{=0} + \phi(0)$$

2 Question 2

1. As $\forall i \in [\![1,n]\!],\ S(x)_i = \dfrac{e^{x_i}}{\sum_k e^{x_k}}$ and $x \mapsto e^x \ \in \mathcal{C}^\infty(\mathbb{R})$, using the formulas for computing the derivative of a fraction of differentiable functions, we have:

$$\forall i,j \in [\![1,n]\!],\ \frac{dS(x)_i}{dx_j} = \frac{\delta_{ij}e^{x_i}}{\sum_k e^{x_k}} - \frac{e^{x_j}e^{x_i}}{\left(\sum_k e^{x_k}\right)^2} = S(x)_i\big[\delta_{ij} - S(x)_j\big]$$

2. As $\forall i,j \in [\![1,n]\!], S(x)_i\delta_{ij} \neq 0 \Leftrightarrow i = j$, we have:

$$\frac{\partial S(x)}{\partial x} = diag(S) - SS^T$$

3. We have: $\forall x \in \mathbb{R}, \sigma(x) = \dfrac{1}{1+e^{-x}}$. Thus, we have:

$$\frac{d\sigma(x)_i}{dx_j} = \delta_{ij}\left(\frac{e^{-x_i}}{\left(1+e^{-x_i}\right)^2}\right) = \delta_{ij}\sigma(x)_i\big(1-\sigma(x)_i\big)$$

Then, we can write:

$$\frac{\partial \sigma(x)}{\partial x} = diag\Big(\sigma(x)\big(1-\sigma(x)\big)\Big)$$

4. As computing a dot product of two vectors of $\mathbb{R}^n$, computing the product of a diagonal matrix of $\mathbb{R}^{n\times n}$ with a vector of $\mathbb{R}^n$, and multiplying a vector of $\mathbb{R}^n$ with a scalar can be done in $O(n)$ operations, we have:

$$\begin{cases} with\ S: \quad \nabla_x L = \underbrace{diag(S)\nabla_y L}_{O(n)\ operations} - \underbrace{S\ \overbrace{S^T\nabla_y L}^{=<S,\nabla_y L>}}_{O(n)\ operations} \\[2ex] with\ \sigma: \quad \nabla_x L = \underbrace{diag\Big(\sigma(x)\big(1-\sigma(x)\big)\Big)\nabla_y L}_{O(n)\ operations} \end{cases}$$

3 Question 3

1. We have:

$$\forall c \in \mathbb{R}, \forall i \in [\![1,n]\!], S(x+c)_i = \frac{e^{x_i+c}}{\sum_k e^{x_k+c}} = \frac{\cancel{e^c}e^{x_i}}{\cancel{e^c}\sum_k e^{x_k}} = S(x)_i$$

2. We can write that:

$$\forall c \in \mathbb{R}, \forall i \in [\![1,n]\!],\ S(xc)_i = \frac{(e^{x_i})^c}{\sum_k (e^{x_k})^c} \neq S(x)_i\ \forall c \neq 1$$

From that, we deduce that if $c = 0$, then $S(x)_i = \dfrac{1}{n}\forall i \in [\![1,n]\!]$.

Now, let's consider the case of $c \to \infty$. Without loss of generality, we can suppose that $x_1 \geq$

$x_2 ... \geq x_n$. Let's name $J = |\max\{x_1, ..., x_n\}|$ ($J \geq 1$ as there might be several maximal values). Thus, we have two cases to consider:

$$\begin{cases} \forall i \in [\![1, J]\!],\ S(sc)_i = \dfrac{1}{\sum_{k=1}^{J} (\underbrace{e^{x_k - x_i}}_{=1})^c + \sum_{k=J+1}^{n} (\underbrace{e^{x_k - x_i}}_{\in]0,1[})^c} \underset{c \to \infty}{\rightarrow} \dfrac{1}{J} \\ \\ \forall i \in [\![J+1, n]\!],\ S(sc)_i = \dfrac{1}{\sum_{k=1}^{J} (\underbrace{e^{x_k - x_i}}_{>1})^c + \sum_{k=J+1}^{n} (\underbrace{e^{x_k - x_i}}_{\in]0,1]})^c} \underset{c \to \infty}{\rightarrow} 0 \end{cases}$$

3. For $n = 2$, we have $\forall x \in \mathbb{R}^2$:

$$\begin{cases} S(x)_1 = \dfrac{e^{x_1}}{e^{x_1} + e^{x_2}} = \dfrac{1}{1 + e^{x_2 - x_1}} \\ \\ S(x)_2 = \dfrac{e^{x_2}}{e^{x_1} + e^{x_2}} = \dfrac{1}{1 + e^{x_1 - x_2}} \end{cases}$$

Thus, by taking $z = x_1 - x_2$ and by recalling that $\forall z \in \mathbb{R}^2, \sigma(-z) = 1 - \sigma(z)$, we have that $S(x) = \big[\sigma(z), 1 - \sigma(z)\big]^T$.

4. Let's use $\forall i \in [\![0, K-1]\!],\ y_i = x_{i+1} - x_1$ (thus, $y_0 = 0$). Hence, we can write:

$$\forall j \in [\![1, K]\!], S(x)_j = \frac{e^{-x_1} e^{x_j}}{e^{-x_1} \sum_{k=1}^{K} e^{x_k}} = \frac{e^{y_{j-1}}}{1 + \sum_{k=1}^{K-1} e^{y_k}}$$

And thus, $\forall x \in \mathbb{R}^K,\ S(x) = S([0, y_1, ... y_{K-1}]^T)$.

4 Question 4

We can write that:

$$\forall x \in \mathbb{R}, \sigma(x) = \frac{1}{1 + e^{-x}} = \frac{e^{x/2}}{e^{x/2} + e^{-x/2}} = \underbrace{\frac{e^{x/2} - e^{-x/2}}{e^{x/2} + e^{-x/2}}}_{=\tanh(x/2)} + \underbrace{\frac{e^{-x/2}}{e^{x/2} + e^{-x/2}}}_{=\sigma(-x)}$$

Thus, by recalling that $\sigma(-x) = 1 - \sigma(x)$, we derive that:

$$\forall x \in \mathbb{R},\ \sigma(x) = \frac{1}{2}\big[\tanh(x/2) + 1\big]$$

Hence, if we take the expression of the neural network, we have:

$$\forall x \in \mathbb{R}^D,\ \forall k \in [\![1, K]\!],\ y(x, \theta, \sigma)_k = \sum_{j=1}^{M} w_{kj}^{(2)} \left[\frac{1}{2} \tanh\left(\sum_{i=1}^{D} \frac{w_{ji}^{(1)}}{2} x_i + \frac{w_{j0}^{(1)}}{2}\right) + \frac{1}{2}\right] + w_{k0}^{(2)}$$

And deduce that $\forall k, j, i \in [\![1, K]\!] \times [\![1, M]\!] \times [\![1, D]\!]$:

$$w_{kj}^{(2)'} = \frac{w_{kj}^{(2)}}{2} \quad ; \quad w_{k0}^{(2)'} = w_{k0} + \frac{1}{2} \sum_{j=1}^{M} w_{kj}^{(2)} \quad ; \quad w_{ji}^{(1)'} = \frac{w_{ji}^{(1)}}{2} \quad ; \quad w_{j0}^{(1)'} = \frac{w_{j0}^{(1)}}{2}$$

5 Question 5

1. The generic form of a two layer network with $N-1$ hidden units $y : \mathbb{R}^n \to \mathbb{R}^m$ is:

$$\forall x \in \mathbb{R}^n,\ y(x) = W^{(2)}\phi\Big[W^{(1)}x + b^{(1)}\Big] + b^{(2)}$$

with $W^{(1)} \in \mathbb{R}^{N-1\times n},\ b^{(1)} \in \mathbb{R}^{N-1},\ W^{(2)} \in \mathbb{R}^{m\times N-1},\ b^{(2)} \in \mathbb{R}^m$.

2. As $W^{(1)} = \underbrace{[w, ..., w]}_{N-1\ times}^T$, we have that $\forall x \in \mathbb{R}^n,\ W^{(1)}x =< w, x > \underbrace{[1, ..., 1]}_{N-1\ times}^T$. Thus, if we call the design matrix $X = [x^{(1)}, ..., x^{(N)}]^T$, we have that $W^{(1)}X^T = \underbrace{[1, ..., 1]}_{N-1\ times}^T[< w, x^{(1)} >, ..., < w, x^{(N)} >]$.

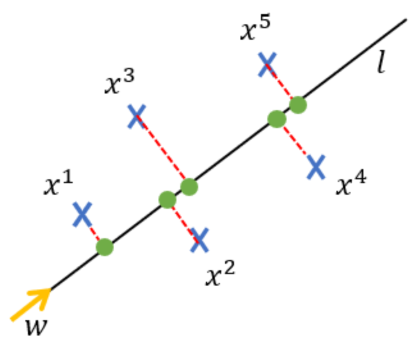
As we have:

$$\forall x \in \underbrace{\{x^{(1)}, ..., x^{(N)}\}}_{=\mathcal{S}},\ y(x) = f(x) \Leftrightarrow W^{(2)}\phi\Big[W^{(1)}X^T + b^{(1)}\underbrace{[1, ..., 1]}_{N\ times}\Big] + b^{(2)}\underbrace{[1, ..., 1]}_{N\ times} = F^T$$

We deduce that:

$$\forall x \in \mathcal{S},\ y(x) = f(x) \Leftrightarrow M\tilde{W}^{(2)} = F\ with\ M = \begin{pmatrix} \phi\big(< w, x^{(1)} > +b_1^{(1)}\big) & ... & \phi\big(< w, x^{(1)} > +b_{N-1}^{(1)}\big) & 1 \\ \vdots & & \vdots & \vdots \\ \phi\big(< w, x^{(N)} > +b_1^{(1)}\big) & ... & \phi\big(< w, x^{(N)} > +b_{N-1}^{(1)}\big) & 1 \end{pmatrix}$$

3. Let's assume that the $\{x^{(i)}\}_{1\leq i\leq N}$ are all different. Thus, they form a cloud of N distinct points in $\mathbb{R}^n$. We want to find $w \in \mathbb{R}^n$ s.t the $\{< w, x^{(i)} >\}_{1\leq i\leq N}$ are all different. Geometrically speaking, this is equivalent to finding a line l in $\mathbb{R}^n$ of direction w such that the orthogonal projections of all the $x^{(i)}$ on l create N different coordinates on that line.



Schema of the situation in $\mathbb{R}^2$ with 5 points

But we know that two distinct points $x^{(i_1)}, x^{(i_2)}$ with $i_1 \neq i_2$ have the same coordinates on l if and only if l is orthogonal to the vector $x^{(i_1)} - x^{(i_2)}$ ($< w, x^{(i_1)} >=< w, x^{(i_2)} >\Leftrightarrow< w, x^{(i_1)} - x^{(i_2)} >= 0$). As there are $\dfrac{N(N-1)}{2}$ different vectors of the type $x^{(i_1)} - x^{(i_2)}$ in a cloud of N distinct points *(number of edges in a clique of size N)* and an uncountable amount of possible directions w, it suffices to pick one w which has a different direction from the $\dfrac{N(N-1)}{2}$ "prohibited" ones to have an w that will verify the wanted property.

Having picked such an w, we fix the value of $b^{(1)}$: $\forall j \in [\![1, N-1]\!]$, $b_j^{(1)} = - < w, x^{(j)} > +\epsilon$ with $\epsilon > 0$. We want to find ϵ such that the matrix M is triangular with non zero diagonal elements. Let's say that we want the lower half of M being equal to zero.

First, let's call a_i the $< w, x^{(i)} >$ for $i \in [\![1, N]\!]$. Thus, $\forall i, j \in [\![1, N]\!] \times [\![1, N-1]\!]$, $m_{ij} = \phi(a_i - a_j + \epsilon)$. As $\phi = RELU$, our purpose is to have $\forall i < j, a_i - a_j + \epsilon \leq 0$. Without loss of generality, we can assume that the $x^{(i)}$ are ordered in a way that $a_1 < ... < a_N$. Thus, if we choose $\epsilon = \min\limits_{i \in [\![1, N-1]\!]} (a_{i+1} - a_i)$, we have that the lower half of M equals to zero with a diagonal made of $N-1$ $\epsilon > 0$ and one 1.

Hence, all the eigenvalues of M are strictly positive and M is invertible. Thus, we have $\tilde{W}^{(2)} = M^{-1}F$ and all the parameters of the second layers are fixed which solves the interpolation problem *(the parameters of the first layers were already found as they only depend on w, X and ϵ).*

4. This time, we write $w = \lambda u$, $\forall i \in [\![1, N]\!]$, $< u, x^{(i)} >= c_i$ and $\forall j \in [\![1, N-1]\!]$, $b_j^{(1)} = -\lambda c_j$. Again, we can assume that the $x^{(i)}$ are ordered in a way that $c_1 < ... < c_N$. Thus, we have that $\forall i < j$, $\lambda(c_i - c_j) \underset{\lambda \to \infty}{\to} -\infty$ and $\forall i = j$, $\lambda(c_i - c_j) = 0$. Hence, as $\phi(-\infty) = 0$ and $\phi(0) > 0$, we have that $\lim\limits_{\lambda \to \infty} M_\lambda = M'$ is triangular with non zero diagonal elements, which means that M' is invertible.

   Lemma 5.1. *The set $GL_N(\mathbb{R})$ of invertible matrices of $\mathcal{M}_N(\mathbb{R})$ is open.*

   Proof. Let's call d the mapping $\begin{array}{cccc} d & : & \mathcal{M}_N(\mathbb{R}) & \to & \mathbb{R} \\ & & M & \mapsto & \det(M) \end{array}$. We have that d is a continuous function on $\mathcal{M}_N(\mathbb{R})$ as it is a polynomial of the elements of M $\Big($*it suffices to write* $d(M) = \sum_{\sigma \in \mathfrak{S}_N} \varepsilon(\sigma) \prod_{k=1}^N m_{k,\sigma(k)}$ *to see it* $\Big)$. As $\mathbb{R}^*$ is an open set of $\mathbb{R}$, $d^{-1}(\mathbb{R}^*)$ is an open set of $\mathcal{M}_N(\mathbb{R})$. But $d^{-1}(\mathbb{R}^*) = GL_N(\mathbb{R})$, which concludes the proof. $\square$

   Using this lemma, we have that there exists an open ball of radius $r > 0$ centered on M' for any norm *(as $\mathcal{M}_N(\mathbb{R})$ is of finite dimension, all the norms are equivalent)* such that any matrix within that ball is invertible. But as $\lim\limits_{\lambda \to \infty} M_\lambda = M'$, $\exists \lambda' \in \mathbb{R}^*$ *s.t* $||M_{\lambda'} - M'|| < r$. Then, for such a λ', $M_{\lambda'}$ is invertible and again, we have that $\tilde{W}^{(2)} = M_{\lambda'}^{-1} F$ which solves the interpolation problem.

6 Question 6

In this question, we will use a stride $s = 1$. We have a $1D$ matrix $[1, 2, 3, 4]$ and a kernel $[1, 0, 2]$. We do the convolution with kernel flipping which leads the following results for the three types of padding:

- **Valid:** We do not add a padding which leads to:

$$[1, 2, 3, 4] * [1, 0, 2] = [2 \times 1 + 0 \times 2 + 3 \times 1 \ , \ 2 \times 2 + 0 \times 3 + 1 \times 4] = [5, 6]$$

- **Same:** We add a padding of $p = 1$ to have an output matrix of the same size as the input matrix:

$$[0, 1, 2, 3, 4, 0] * [1, 0, 2] = [2, 5, 8, 6]$$

- **Full:** We add a padding of size $p = k - 1 = 2$:

$$[0, 0, 1, 2, 3, 4, 0, 0] * [1, 0, 2] = [1, 2, 5, 8, 6, 8]$$

7 Question 7

1. In this question, we will use the formula for computing the size of the output of a convolution o depending on the size of the input matrix i, the padding p, the kernel k and the stride s:
$o = \left\lfloor \frac{i+2p-k}{s} \right\rfloor + 1$. For the first layer, we have: $i = 256,\ s = 2,\ p = 0,\ k = 8$ which leads to $o_1 = 125$. We then use a 5×5 non overlapping max pooling, which leads to $o_2 = 25$. Then, for the last layer we use 128 kernels of size $k = 4$ and have $i = 25,\ s = 1,\ p = 1$. In the end, we thus have 128 matrices of size 24×24 which means that the dimension of the output is $d = 128 \times 24 \times 24 = 73728$.

2. For the last layer, we take as input 64 matrices and we use 128 kernels of size 4×4, without counting the bias, that leads to a total of $64 \times 128 \times 4 \times 4 = 161792$ parameters for this layer.

8 Question 8

In this question, we will use the formula for computing the size of the output of a convolution o depending on the size of the input matrix i, the padding p, the kernel k, the stride s and the dilatation d:
$o = \left\lfloor \frac{i+2p-k-(k-1)(d-1)}{s} \right\rfloor + 1$.

1. We have an input image of size $i = 64$ and we want $o = 32$.
   (a) We have $k = 8$ and $d = 1$. Thus, $32 = \left\lfloor \frac{64+2p-8}{s} \right\rfloor + 1$ and taking $s = 2$ and $p = 3$ works.
   (b) We have $d = 7$ and $s = 2$. Thus $32 = \left\lfloor \frac{64+2p-k-(k-1)6}{2} \right\rfloor + 1$ and taking $p = 3$ and $k = 2$ works.

2. We have an input image of size $i = 32$ and we want $o = 8$. We have $p = 0$ and $d = 1$.
   (a) For pooling with non overlapping windows, we must have $k = s$. As we have $\frac{i}{o} = 4$ we take $k = s = 4$.
   (b) If we have $k = 8$ and $s = 4$, then $o = \left\lfloor \frac{32-8}{4} \right\rfloor + 1 = 7$

3. We have an input image of size $i = 8$ and we want $o = 4$.
   (a) We have $d = 1$ and $p = 0$. $4 = \left\lfloor \frac{8-k}{s} \right\rfloor + 1$ and taking $k = 2$ and $s = 2$ works.
   (b) We have $d = 2$ and $p = 2$. $4 = \left\lfloor \frac{8+4-2k+1}{s} \right\rfloor + 1$ and taking $k = 2$ and $s = 3$ works.
   (c) We have $d = 1$ and $p = 1$. $4 = \left\lfloor \frac{8+2-k}{s} \right\rfloor + 1$ and taking $k = 4$ and $s = 2$ works.