CS 540 Fall 2023
Homework 1 Problem 1

Matthew Tang (mhtang2)
Dicussion with:
Jacob Levine (jlevine4)
Colin Lu (colinyl2)

Solution:

(a)

$$\inf_f \mathscr{R}(f) = \inf_f \mathbb{E}[\ell(f(x), y)] \tag{1}$$
$$= \inf_f \mathbb{E}[\ell(f(x), y) - \ell(g(x), y) + \ell(g(x), y)] \tag{2}$$
$$\leq \mathbb{E}[\ell(g(x), y)] \tag{3}$$
$$\leq \inf_g \mathbb{E}[\ell(g(x), y)] \tag{4}$$
$$\leq \inf_g \mathscr{R}(g) \tag{5}$$

(2) Introduce new terms which sum to 0, introducing arbitrary $g \in \mathscr{G}$.
(3) Because ℓ is rho-Lipschitz, $|\ell(f(x)) - \ell(g(x))| \leq K|g(x) - f(x)| <= 0$ since f is a universal approximator and we can make it arbitrarily close to g. It can be replaced with 0, since that is the f which will minimize the expectation.
(4) LHS is a lower bound for arbitrary g, so it must be less than infimum over g.
(5) Definition of $\mathscr{R}$

Because networks are continuous, $f \in \mathscr{G} \implies \inf_g \mathscr{R}(g) \leq \inf_f \mathscr{R}(f)$ as well, so $\inf_g \mathscr{R}(g) = \inf_f \mathscr{R}(f)$

(b) Consider probability distribution as the Dirac delta, where all probability is concentrated on point $x = \vec{0}, y = +1$.

Pick $f(x)$ as a constant value (This is possible by setting outer layer weights to 0, and assigning the bias to be the desired constant). Call this constant value $f(x) = f$. Select $g(0) = 0$, the rest will be defined later. Then find the constant f so that the risk inequality is satisfied:

$$R(f) \geq 1/\epsilon + R(g)$$
$$\mathbb{E}[\log(1 + \exp(-yf))] \geq 1/\epsilon + \mathbb{E}[\log(1 + \exp(-yg(x)))]$$
$$\log(1 + \exp(-f)) \geq 1/\epsilon + \log(1 + \exp(0)) \qquad \text{Only 1 point has probability}$$
$$f \leq -\log(e^{1/\epsilon} + \log(2) - 1)$$

Now define the rest of $g(x)$ so that $\int_x |f(x) - g(x)| \leq \epsilon$. Let $g(x)$ be the $d+1$ dimensional hypercone with height $-f$ (f is negative). The base of the hypercone is an n-ball of dimension d and radius r centered at $(0, ..., 0, f)$.

$$g(\boldsymbol{x}) = \begin{cases} f \cdot \|\frac{\boldsymbol{x}}{r}\|_2 & \|\boldsymbol{x}\|_2^2 \leq r \\ f & \text{Otherwise} \end{cases}$$

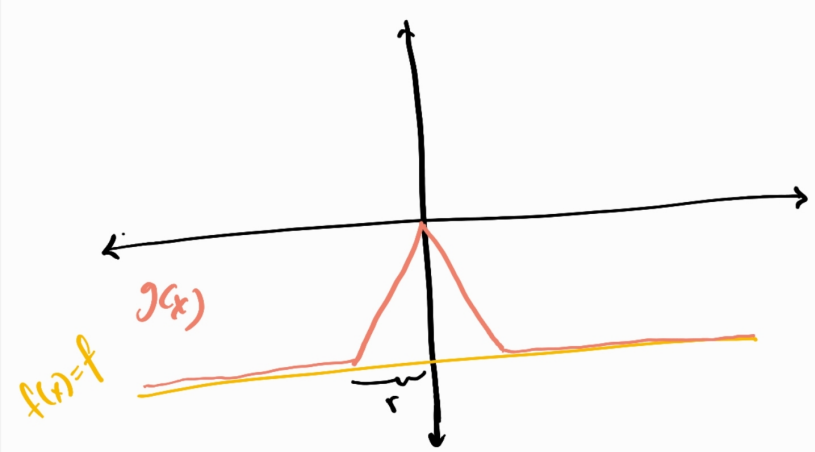

Figure 1. Illustration for $d = 1$

Chose r small enough to bound the L_1 between f, g:
Since $g(x) = f$ outside of the cone's base, $\int_x |f - g(x)|dx$ is just the volume of the cone, which is $\frac{1}{d+1} *$ Base N-Ball Volume $* |f| = \frac{1}{d+1} * \frac{\pi^{d/2}}{\Gamma(1+d/2)} r^d * |f|$. Now upperbound this by ϵ and solve for $r \leq ((\frac{1}{d+1}\frac{\pi^{d/2}}{\Gamma(1+d/2)}|f|)^{-1}\epsilon)^{1/d}$
Picking any radius satisfying this bound will satisfy the error bound between f and g.

Now f(x),g(x) have been picked which satisfy both inequalities. Note that $x \in [0, 1]$, but this proof assumes $x \in [-1, 1]$, which does not change the proof, since the functions can be shifted and rescaled.

Source for volume of cone:
Su, Francis E., et al. "Volume of a Cone in N Dimensions." Math Fun Facts. <https://www.math.hmc.edu/funfacts>

(c) First show that f is linear for large x.
First note $f(x) = \sum_j^m a_j \sigma(v_j x + b_j)$ is a sum of a finite number of ReLUs. Note ReLU$(vx + b)$ changes at only one value of x, when $vx + b = 0$, or $x = -b/v$. Then for all $x > -b/v$, ReLU$(vx + b)$ evaluates to either 0 or $vx + b$, but will not switch between the two. For very large x beyond the changing point of these ReLUs (i.e. $x > \max(-v_j/b_j)$), then $f(x) = \sum_i^m a_i(v_i x + b_i)$ where i are nodes with non-zero ReLU activations. This is a sum of affine functions, so f is affine. Combine constants to call $f(x) = kx + c$
In the case $k \neq 0$, $|(kx + c) - sin(x)|$ is unbounded, so $|f(x) - sin(x)| > 1$ for some large x.
In the case $k = 0$, $f(x) = c$ is constant. If $c \geq 0$, then consider large x s.t. $sin(x) = -1$ (which must exist as sine is periodic), then $f(x) - sin(x) \geq 1$. If $c \leq 0$, then consider large x s.t. $sin(x) = 1$, then $sin(x) - f(x) \geq 1$.

(d) The general idea is to use each layer of the narrow network to compute the output of one node in the shallow network. Within each layer of the narrow network there are $d + 3$ nodes: use d nodes to constantly store the d inputs. Of the other 3 nodes, use 1 node to store the total positive outputs, 1 node to store the total negative outputs, and 1 node to perform calculations. At the end of the network, the postive and negative parts of f are stored in the accumulators, and can be subtracted to obtain f.
Let $w_{i,j}^{(l)}$ denote the weight between the i^{th} node of layer $l - 1$ to the j^{th} node of layer l. Let

$+$ denote the positive accumulator node, $-$ the negative accumulator, and C the calculation node. For example $w^{(l)}_{C,+}$ be the weight between the calculation node on layer $l-1$ and positive accumulation node on layer l.

Let shallow network $f(x) = \sum_j^m a_j \sigma(v_j^T x + b_j)$

Then the output of the positive accumulator in layer (l) is the sum of positive outputs of the first l hidden nodes in f:

The output of positive accumulator node on layer l is $\sum_i^l a_i \sigma(v_i^T x)$ where $a_i \sigma(v_i^T x) \geq 0$.

Similarly, the output of the negative accumulator is $-\sum_i^l a_i \sigma(v_i^T x)$ where $a_i \sigma(v_i^T x) \leq 0$

Now define the weights of the narrow network. To pass forward the inputs, set w as follows. For all $l \leq m$ and $i, j \leq d$,

$$w^{(l)}_{i,j} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

Then in every layer, the output of the ith node will be the original input x_i.

Now define the weights going into the calculation node. The calculation node performs $\sigma(v_j^T x + b_j)$

$$w^{(l)}_{i,c} = (v_l)_i \quad \forall i \leq d, l \leq m$$

This is also the only time biases are needed in the narrow network:

$$b^{(l)}_c = b_l$$

Now define the weights going into the positive and negative accumulators. This performs the calculation $a_j * \sigma(v_j^T x + b_j)$ and adds it to the running total.

$$w^{(l)}_{C,+} = \begin{cases} a_l & a_l \geq 0 \\ 0 & a_l < 0 \end{cases}$$

$$w^{(l)}_{+,+} = 1$$

The idea is to accumulate positive values from previous layers. Since the ReLU is always positive, the sign is determined by a. We only want to accumulate if $a \geq 0$ then. Define the negative accumulator similarly

$$w^{(l)}_{C,-} = \begin{cases} -a_l & a_l \leq 0 \\ 0 & a_l > 0 \end{cases}$$

$$w^{(l)}_{-,-} = 1$$

Finally the output of the network is the difference of the last positive and negative accumulators:

$$w^{(m+1)}_{+,out} = 1$$

$$w^{(m+1)}_{-,out} = -1$$

Any weights/biases not explicitly defined are not needed and set to 0

■

Matthew Tang (mhtang2)
Dicussion with:
Jacob Levine (jlevine4)
Colin Lu (colinyl2)

CS 540 Fall 2023
Homework 1 Problem 2

Solution:

(a) Case 1: $f(x)$ monotone. By mean value theorem: $\exists c$ s.t. $f'(c) = \frac{f(1)-f(0)}{1-0} = f(1)-f(0)$ Then $|f'(c)| = |f(1)-f(0)| \implies \sup_x |f'(x)| \geq |f(1)-f(0)|$.

Case 2: $f(x)$ not monotone. Then one of g,h must be increasing and the other decreasing (if they were both increasing, then f would be increasing and monotone). WLOG let g be increasing and h decreasing. Define

$$g'(x) = \begin{cases} f'(x) & f'(x) \geq 0 \\ 0 & f'(x) \leq 0 \end{cases}$$

$$h'(x) = \begin{cases} f'(x) & f'(x) \leq 0 \\ 0 & f'(x) \geq 0 \end{cases}$$

Note that $g'(x) \geq 0$, $h'(x) \leq 0$ meaning they are monotone. Also $f'(x) = g'(x) + h'(x)$ so $f(x) = g(x) + h(x)$. Then if these specific g, h satisfies $\|f\|_{LIP} \geq \|g\|_{BV} + \|h\|_{BV}$, the infimum over all possible g, h must be less: $\|f\|_{LIP} \geq \inf(\|g\|_{BV} + \|h\|_{BV})$.

Show that these specific g, h satisfy the inequality.

$$\begin{aligned}
\sup_{x_1} |f'(x_1)| &= \int_0^1 \sup_{x_1} |f'(x_1)| dx && (1) \\
&\geq \int_0^1 |f'(x)| dx && (2) \\
&= \int_0^1 |g'(x)| + |h'(x)| dx && (3) \\
&= \int_0^1 g'(x) dx + - \int_0^1 h'(x) dx && (4) \\
&= g(1) - g(0) + h(0) - h(1) \\
&= |g(1) - g(0)| + |h(0) - h(1)| && (5) \\
&= \|g\|_{BV} + \|h\|_{BV}
\end{aligned}$$

(1) Integral of constant. x_1 is used to not be confused with integrating variable x.
(2) $\sup_{x_1} |f'(x_1)| \geq |f'(x)| \forall x$.
(3) When $f'(x) > 0$,by def $h'(x) = 0$, so $|f'(x)| = |g'(x) + 0| + 0 = |g'(x)| + |h'(x)|$. Similar proof for case $f'(x) \leq 0$
(4) $g(x) \geq 0$ and $h(x) \leq 0$
(5) Both differences of endpoints positive since g increasing and h decreasing

(b) Two cases:
Case 1: $\epsilon \geq 1 : \frac{1}{\epsilon} \leq \epsilon$.
Pick $f(x) := \epsilon x$.
Then $\|f\|_{BV} = \epsilon \leq \epsilon$ and $\|f\|_{LIP} = \epsilon \geq 1/\epsilon$
Case 2: $\epsilon < 1 : \frac{1}{\epsilon} > \epsilon$
Pick

$$f(x) = \begin{cases} x/\epsilon & 0 \leq x \leq \epsilon^2 \\ \epsilon & \epsilon^2 \leq x \leq 1 \end{cases}$$

Note $f(x)$ is increasing. Then $\|f\|_{BV} = f(1) - f(0) = \epsilon \leq \epsilon$

$$\|f\|_{LIP} = \max(0, 1/\epsilon) = 1/\epsilon \geq 1/\epsilon$$

(c) Since BV-Norm of g is finite, $g(x)$ can be decomposed into an increasing and decreasing function. If each monotone function can be approximated using a network within error $\epsilon/2$ using $2\lceil \|g\|_{BV}/\epsilon \rceil$ nodes, then the sum of the two networks uses $4\lceil \|g\|_{BV}/\epsilon \rceil$ nodes and can approximate the sum of the two monotone functions, $g(x)$, with approximation error at most $\epsilon/2 + \epsilon/2 = \epsilon$. (The construction for summing networks is found in the answer to 4.a)

What's left is to show that monotone function $h(x)$ can be approximated using a network within error $\epsilon/2$ using $2\lceil \|h\|_{BV}/\epsilon \rceil$ nodes. WLOG assume $h(x)$ increasing. Construct $f(x)$ similarly to the univariate approximation of Lipschitz smooth functions in lecture notes. Let $a_0 = 0$, and $a_i > a_{i-1}$ be such that $h(a_i) - h(a_{i-1}) = \epsilon/2$. This network would be $f(x) = \sum_i^m \frac{\epsilon}{2} \mathbb{1}[x - a_i \geq 0]$. Notice at each step, $h(x) - f(x) \leq \epsilon/2$. Now find the number of nodes m. Since $f(x)$ takes steps of size $\epsilon/2$, it needs to take $m = \lceil \frac{h(1)-h(0)}{\epsilon/2} \rceil$ steps. Since h is monotone, rewrite it as BV norm: $m = 2\lceil \frac{\|h\|_{BV}}{\epsilon} \rceil$

(d) In part (c) an approximation network $f(x)$ was constructed using indicator as activation. Show that by only replacing the activations in $f(x)$ with ReLU, we can construct new network $F(x) = \int f(x)dx$ (Number of nodes don't change).

$$\begin{aligned} \int f(x)dx &= \int \sum_i^m \frac{\epsilon}{2} \mathbb{1}[x - a_i \geq 0]dx \\ &= \sum_i^m \frac{\epsilon}{2} \int \mathbb{1}[x - a_i \geq 0]dx \\ &= \sum_i^m \frac{\epsilon}{2} \text{ReLU}(x - a_i) \\ &= F(x) \end{aligned}$$

From (c) we know g' can be approximated using $f(x)$ s.t. $\forall x \quad |g(x)' - f(x)| \leq \epsilon$ using $4\lceil \frac{\|g'\|_{BV}}{\epsilon} \rceil$ nodes. Construct $F(x) = \int f(x)dx$ using the same number of nodes by switching

the activations of $f(x)$ with Relus. Show that $F(x)$ approximates g to the same error.

$$\begin{aligned}
|F(x)-g(x)| &= |\int_0^x f(t)dt - \int_0^t g'(t)dt| \\
&= |\int_0^x f(t)-g'(t)dt| \\
&\leq |\int_0^x \epsilon dt| && \text{Bounded appx error} \\
&= |x\epsilon| && x \in [0,1] \\
&\leq \epsilon
\end{aligned}$$

■

CS 540 Fall 2023
Homework 1 Problem 3

Matthew Tang (mhtang2)
Dicussion with:
Jacob Levine (jlevine4)
Colin Lu (colinyl2)

Solution: Let x_1, x_2 have angle δ, and $y_1 = +1, y_2 = -1$. Consider the case when none of the w_j can separate x_1, x_2: $\forall j : \sigma'(w_j^T x_1) = \sigma'(w_j^T x_2) \iff \text{sgn}(w_j^T x_1) = \text{sgn}(w_j^T x_2)$. Note w_j is able to separate x_1, x_2 if the normal plane to w_j is within the two points. This happens over angle δ, so this probability is $\frac{\delta}{2\pi}$. Because w_j are iid, the probability no w_j can separate is $(1 - \frac{\delta}{2\pi})^m$. This probability is above the bound in the problem $(1 - \frac{m\delta}{\pi})$, so it is satisfied.
In this case it is still possible for v_j to separate the points and predict them correctly. That's true when $\sum v_j^T x_1 \sigma'(w_j^T x_1) = +1$ and $\sum v_j^T x_2 \sigma'(w_j^T x_2) = -1$. Since some indicators go to 0, consider only the sum over non-zero indictors: $\sum_i v_i^T x_1 = +1$ and $\sum_i v_i^T x_2 = -1$, where the sum is over i s.t. $w_i^T x_1 \geq 0$. Note i is the same in both equations since $\text{sgn}(w_j^T x_1) = \text{sgn}(w_j^T x_2)$, so both have non-zero indicators at the same indices i. Then subtract the two equations to get

$$\begin{aligned}
\sum v_i^T x_1 - \sum v_i^T x_2 &= 2 \\
\sum v_i^T (x_1 - x_2) &= 2 \\
\sum \|v_i\| \|(x_1 - x_2)\| &\geq 2 && (1) \\
\sum \|v_i\| &\geq \frac{2}{\|(x_1 - x_2)\|} \\
&\geq \frac{1}{\delta} && (2)
\end{aligned}$$

(1) Cauchy schwartz.
(2) is shown here:

$$\begin{aligned}
\frac{2}{\|(x_1 - x_2)\|} &\geq \frac{1}{\delta} \\
\delta &\geq \frac{\|(x_1 - x_2)\|}{2} \\
&= \frac{\sqrt{2 - 2\cos\delta}}{2} && \text{Law of cosine} \\
&= |\sin(\delta/2)| \\
\delta &\geq |\sin(\delta/2)| && \text{True from small angle, or check taylor approx. of sine.}
\end{aligned}$$

Just showed with probability $(1 - \frac{\delta}{2\pi})^m \geq (1 - \frac{m\delta}{\pi})$, x_1 and x_2 can be assigned to correct labels only if $\sum \|v_i\| \geq \frac{1}{\delta}$. ■

CS 540 Fall 2023
Homework 1 Problem 4

Matthew Tang (mhtang2)
Dicussion with:
Jacob Levine (jlevine4)
Colin Lu (colinyl2)

Solution:

(a) $h(x) = \frac{\partial}{\partial w} f(wx) = \lim_{e \to 0} \frac{f((w+e)x) - f(wx)}{e}$
Then $\frac{f((w+e)x)-f(wx)}{e}$ can be constructed as a new two layer network $g(x)$. Summing two networks can be done by stacking the hidden layer, and scaling can be done in the outer layer. If $f(x) = \sum_j a_j \sigma(v_j x)$, then define g with weights (v', a'), where v' is the concatenation $\begin{pmatrix} (w+e)\boldsymbol{v} \\ w\boldsymbol{v} \end{pmatrix}$, and a' is the concatenation $\begin{pmatrix} \boldsymbol{a}/e \\ -\boldsymbol{a}/e \end{pmatrix}$ Show this construction works:

$$\begin{aligned} g(x) &= \sum_i a_i' \sigma(v_i' x) \\ &= \sum_j \frac{a_j}{e} \sigma((w+e)v_j x) + \sum_j \frac{-a_j}{e} \sigma(w v_j x) \\ &= \frac{\sum a_j \sigma((w+e)v_j x) - \sum a_j \sigma(w v_j x)}{e} \\ &= \frac{f((w+e)x) - f(wx)}{e} \end{aligned}$$

Using the limit definition of derivative, since $h(x) = \lim_{e \to 0} g(x)$, e can be picked arbitrarily small so that $\|g(x) - h(x)\|_u$ is ϵ small. This can be done by picking e such that $\sup_x |\frac{f((w+e)x)-f(wx)}{e} - \frac{\partial}{\partial w} f(wx)| \leq \epsilon$. This e must exist since f is differentiable, so the limit definition of the derivative must converge to the derivative at any x.

This result also shows that 2 layer networks are closed under addition and scalar multiplication, which will be used later.

(b) Weak Inductive hypothesis:
Given $\frac{\partial^n}{\partial w^n} \sigma(wx + b)$ and $\epsilon > 0$, there is a $k(x, w) \in \mathscr{F}$ s.t. $\|\frac{\partial^n}{\partial w^n} \sigma(wx + b) - k(x, w)\|_u \leq \epsilon$.
Base case for $n = 0$ is trivial since $\sigma(wx + b) \in \mathscr{F}$ so there is no error.
Induction step:
Define $h(x, w) = \frac{\partial^n}{\partial w^n} \sigma(wx + b)$.
We need to show $\frac{\partial^{(n+1)}}{\partial w^{(n+1)}} \sigma(wx + b) = \frac{\partial}{\partial w} h(x, w)$ can be approximated to uniform norm ϵ using some $g(x) \in \mathscr{F}$.
Use two approximations: First $\frac{\partial}{\partial w} h(x, w)$ which can be approximated using limit definition of the derivative: $\lim_{e \to 0} \frac{h(x,w+e)-h(x,w)}{e}$. Pick e s.t. this approximation error $\|\frac{h(x,w+e)-h(x,w)}{e} - \frac{\partial}{\partial w} h(x, w)\|_u \leq \epsilon/3$. As reasoned in (a), this must be possible.
By inductive hypothesis, $h(x, w)$ can be approximated within error $\frac{e \cdot \epsilon}{3}$ using a network $k(x, w)$.
Use this for the second approximation $\frac{h(x,w+e)-h(x,w)}{e} \approx \frac{k(x,w+e)-k(x,w)}{e}$. The error is

$\|\frac{h(x,w+e)-h(x,w)}{e}-\frac{k(x,w+e)-k(x,w)}{e}\|_u \le \|\frac{h(x,w+e)-h(x,w)}{e}-\left(\frac{(h(x,w+e)+e\epsilon/3)-(h(x,w)-e\epsilon/3)}{e}\right)\|_u \le 2\epsilon/3$
Then the total error of using $\frac{k(x,w+e)-k(x,w)}{e}$ to approximate $\frac{\partial}{\partial w}h(x,w)$ is at most $\epsilon/3+2\epsilon/3 = \epsilon$. Since k is a neural network, we can exactly construct this approximation network $g(x) = \frac{k(x,w+e)-k(x,w)}{e}$ using the same method in (a) using the closure under addition and scalar multiplication.

(c) Let b be the point such that $\sigma^{(n)}(b) \neq 0$ so there will be no divide by zeros. Next $x^n = \frac{x^n\sigma(wx+b)}{\sigma(wx+b)}$. Pick $w = 0$, now $x^n = \frac{x^n\sigma(b)}{\sigma(b)}$. By (b), find $g_1(x) \in \mathscr{F}$ s.t. $\|g_1(x) - x^n\sigma(b)\| \le \epsilon$. Construct $g(x) = \frac{g_1(x)}{\sigma(b)}$ exactly using closure under scalar multiplication shown in (a) (There is no approximation error). Then $\|g(x) - \frac{x^n\sigma(b)}{\sigma(b)}\|_u \le \epsilon$

(d) Perform two approximations: the taylor approximation of $\exp(x)$ and the approximation of the polynomial terms in the taylor approximation. First: Pick N s.t.

$$\sup_{|x|<r} \|\sum_{n=1}^{N} \frac{x^n}{n!} - \exp(x)\| \le \epsilon/2$$

Next approximate each x^n using $p_n(x)$ s.t. $\|x^n - p_n(x)\|_u \le \frac{\epsilon}{2N}$. Then a network $f(x) = \sum_{n=1}^{N} \frac{p_n(x)}{n!}$ can be constructed exactly, since it is a linear combination of networks p_n. The error of this approximation is bounded $\|\sum_{n=1}^{N} \frac{p_n(x)}{n!} - \sum_{n=1}^{N} \frac{x^n}{n!}\| \le \frac{\epsilon}{2N} * N = \frac{\epsilon}{2}$. Then the error $\|f(x) - \exp x\|$ is less than the combined error of these two approximations: $\epsilon/2 + \epsilon/2 = \epsilon$.

■