Mathematical Foundations of Deep Neural Networks, M1407.001200
E. Ryu
Spring 2024

Midterm Exam
Saturday, April 13, 2024, 9:00am–1:00pm
4 hours, 7 questions, 100 points, 11 pages

This exam is open-book in the sense that you may use any non-electronic resource.
While we don't expect you will need more space than provided,
you may continue on the back of the pages.

Name: _____

Do not turn to the next page until the start of the exam.

1. (10 points) *Bias-variance tradeoff for moment estimators.* Let $g^0, g^1, g^2, \ldots \in \mathbb{R}$ be IID random variables with mean μ and standard deviation σ. (In RMSprop or Adam, $g^0, g^1, g^2, \ldots \in \mathbb{R}^p$ are vectors, but, for simplicity, we assume they are scalars in this problem.) Let

$$m^{k+1} = \beta m^k + (1-\beta) g^k$$

for $k = 0, 1, \ldots$, with $m^0 = 0$. Assume $0 < \beta < 1$. Show the following.

(a) $\mathbb{E}[m^k] = (1-\beta^k)\mu$ for $k = 0, 1, \ldots$.

(b) $\mathrm{Var}(m^k) = \sigma^2 \frac{1-\beta}{1+\beta}(1-\beta^{2k})$ for $k = 0, 1, \ldots$.

(c)

$$\lim_{k\to\infty} \mathbb{E}[m^k] = \mu, \qquad \lim_{k\to\infty} \mathrm{Var}(m^k) = \sigma^2 \frac{1-\beta}{1+\beta}.$$

Hint. Recall that the classical geometric sum formula is

$$\sum_{i=0}^{k-1} r^i = \frac{1-r^k}{1-r}, \qquad \text{for } |r| < 1.$$

Hint. Obviously, $\mathrm{Var}(m^0) = 0$. Show, for $k = 0, 1, \ldots$,

$$\mathrm{Var}(m^{k+1}) = \beta^2 \mathrm{Var}(m^k) + (1-\beta)^2 \sigma^2.$$

Clarification. The notation used here and in the lecture, unfortunately, can be confusing. Here, m^k and g^k means the k-th m and g, not the k-th power. On the other hand, β^k or β^{2k} do refer to the k-th or $2k$-th powers of β.

Solution: Note that m^k and g^k are independent. This fact comes from that m^k depends on only $g^0, g^1, g^2, \ldots, g^{k-1}$ and $g^0, g^1, g^2, \ldots, g^{k-1}, g^k$ are IID.

(a) Using *mathematical induction*.

Base case: For $k = 0$, $\mathbb{E}[m^0] = \mathbb{E}[0] = 0$. Also, $(1 - \beta^0)\mu = (1 - 1)\mu = 0$.
Induction step: Let's assume that the statement is true for k. Then

$$\mathbb{E}[m^{k+1}] = \mathbb{E}[\beta m^k + (1-\beta)g^k] = \beta\mathbb{E}[m^k] + (1-\beta)\mathbb{E}[g^k] = \beta(1-\beta^k)\mu + (1-\beta)\mu = (1-\beta^{k+1})\mu$$

Since both the base case and the induction step have been proved as true, by *mathematical induction*, it holds for every non-negative integer k.

(b) Using *mathematical induction*.

Base case: For $k = 0$, $\mathrm{Var}(m^0) = 0$. Also, $\sigma^2 \frac{1-\beta}{1+\beta}(1 - \beta^{2\cdot 0}) = 0$.
Induction step: Let's assume that the statement is true for k. Then

$$\begin{aligned}
\mathrm{Var}(m^{k+1}) &= \mathrm{Var}\left(\beta m^k + (1-\beta)g^k\right) = \mathrm{Var}(\beta m^k) + \mathrm{Var}\left((1-\beta)g^k\right) = \beta^2\mathrm{Var}(m^k) + (1-\beta)^2\mathrm{Var}(g^k) \\
&= \beta^2\sigma^2\frac{1-\beta}{1+\beta}(1-\beta^{2k}) + (1-\beta)^2\sigma^2 = \sigma^2(1-\beta)\left(\frac{\beta^2 - \beta^{2(k+1)}}{1+\beta} + 1 - \beta\right) \\
&= \sigma^2(1-\beta)\left(\frac{\beta^2 - \beta^{2(k+1)}}{1+\beta} + \frac{1-\beta^2}{1+\beta}\right) = \sigma^2\frac{1-\beta}{1+\beta}\left(\beta^2 - \beta^{2(k+1)} + 1 - \beta^2\right) \\
&= \sigma^2\frac{1-\beta}{1+\beta}(1-\beta^{2(k+1)})
\end{aligned}$$

Since both the base case and the induction step have been proved as true, by *mathematical induction*, it holds for every non-negative integer k.

(c) Since $0 < \beta < 1$, $\lim_{k\to\infty} \beta^k = 0$,

$$\lim_{k\to\infty} \mathbb{E}[m^k] = \lim_{k\to\infty} (1-\beta^k)\mu = (1 - \lim_{k\to\infty} \beta^k)\mu = (1-0)\mu = \mu.$$

Also, since $0 < \beta^2 < 1$, $\lim_{k\to\infty} \beta^{2k} = 0$,

$$\lim_{k\to\infty} \mathrm{Var}(m^k) = \lim_{k\to\infty} = \lim_{k\to\infty} \sigma^2\frac{1-\beta}{1+\beta}(1-\beta^{2k}) = \sigma^2\frac{1-\beta}{1+\beta}(1-\lim_{k\to\infty}\beta^{2k}) = \sigma^2\frac{1-\beta}{1+\beta}(1-0) = \sigma^2\frac{1-\beta}{1+\beta}$$

Note that $\mathbb{E}[m^k]$ and $\mathrm{Var}(m^k)$ are increasing, but bounded.

Bias is a measure of how far on average m_k is from the original value, m_0. i.e.,

$$\text{Bias} = m_0 - \mathbb{E}[m^k]$$

As the variance $\mathrm{Var}(m^k)$ decreases, the bias $m_0 - \mathbb{E}[m^k]$ increases, and vice-versa. We call this phenomenon as a *Bias-variance tradeoff*.

2. (10 points) *Multiple labels per data point for binary classification.* Let $X_1, \ldots, X_N \in \mathbb{R}^m$ be IID data points and let $Y_{ij} \in \{-1, +1\}$ for $j = 1, \ldots, n$ be independent labels of X_i for $i = 1, \ldots, N$, where $n \geq 1$. (Imagine we ask n different human labelers to provide a label for X_i. This would make sense if there is a possibility of disagreement among the human labelers.) Define the empirical distribution

$$\mathcal{P}(\{Y_{ij}\}_{j=1}^n) = \frac{1}{n}\begin{bmatrix}\text{Count of } Y_{ij} = -1 \text{ among } j = 1, \ldots, n \\ \text{Count of } Y_{ij} = +1 \text{ among } j = 1, \ldots, n\end{bmatrix}.$$

For $\theta \in \mathbb{R}^p$, let $f_\theta \colon \mathbb{R}^m \to \mathbb{R}$ and

$$\mu(f_\theta(X)) = \begin{bmatrix} \frac{1}{1+e^{f_\theta(X)}} \\ \frac{1}{1+e^{-f_\theta(X)}} \end{bmatrix}.$$

Show that the minimization problem

$$\underset{\theta \in \mathbb{R}^p}{\text{minimize}} \quad \frac{1}{N}\sum_{i=1}^N D_{\mathrm{KL}}\big(\mathcal{P}(\{Y_{ij}\}_{j=1}^n) \big\| \mu(f_\theta(X_i))\big)$$

is equivalent to

$$\underset{\theta \in \mathbb{R}^p}{\text{minimize}} \quad \frac{1}{nN}\sum_{i=1}^N \sum_{j=1}^n \ell(Y_{ij} f_\theta(X_i))$$

with $\ell(z) = \log(1 + e^{-z})$.

Solution: Let's first define an auxiliary function such that $\hat{\mathcal{P}}(-1) = \begin{bmatrix}1\\0\end{bmatrix}$ and $\hat{\mathcal{P}}(+1) = \begin{bmatrix}0\\1\end{bmatrix}$. Observe $\mathcal{P}(\{Y_{ij}\}_{j=1}^n) = \frac{1}{n}\sum_{j=1}^n \hat{\mathcal{P}}(Y_{ij})$. Now by performing direct computation,

$$\begin{aligned}
& \underset{\theta \in \mathbb{R}^p}{\text{minimize}} && \frac{1}{N}\sum_{i=1}^N D_{\mathrm{KL}}\big(\mathcal{P}(\{Y_{ij}\}_{j=1}^n) \big\| \mu(f_\theta(X_i))\big) \\
\iff & \underset{\theta \in \mathbb{R}^p}{\text{minimize}} && \frac{1}{N}\sum_{i=1}^N \mathcal{H}\big(\mathcal{P}(\{Y_{ij}\}_{j=1}^n) \big\| \mu(f_\theta(X_i))\big) + (\text{terms independent of } \theta) \\
\overset{(1)}{\iff} & \underset{\theta \in \mathbb{R}^p}{\text{minimize}} && \frac{1}{nN}\sum_{i=1}^N \sum_{j=1}^n \mathcal{H}\big(\hat{\mathcal{P}}(Y_{ij}) \big\| \mu(f_\theta(X_i))\big) \\
\overset{(2)}{\iff} & \underset{\theta \in \mathbb{R}^p}{\text{minimize}} && \frac{1}{nN}\sum_{i=1}^N \sum_{j=1}^n \log(1 + e^{-Y_{ij} f_\theta(X_i)}) \\
\iff & \underset{\theta \in \mathbb{R}^p}{\text{minimize}} && \frac{1}{nN}\sum_{i=1}^N \sum_{j=1}^n \ell(Y_{ij} f_\theta(X_i)),
\end{aligned}$$

where $\mathcal{H}(p, q) = -\sum_{i=1}^N p_i \log(q_i)$ is the cross entropy of q relative to p. (1) comes from the fact that $\mathcal{H}$ is linear with respect to the first input and (2) can be checked by plugging in the two possible values $Y_{ij} \in \{-1, 1\}$.

3. (10 points) *Logistic regression with separable data.* Consider the binary classification problem with data $X_1, \ldots, X_N \in \mathbb{R}^p$ and corresponding labels $Y_1, \ldots, Y_N \in \{-1, +1\}$. Assume the training data is linearly separable, i.e., there is $a_{\text{true}} \in \mathbb{R}^p$ and $b_{\text{true}} \in \mathbb{R}$ such that

$$Y_i = \text{sign}(a_{\text{true}}^\mathsf{T} X_i + b_{\text{true}}) \qquad \text{for } i = 1, \ldots, N,$$

where we define $\text{sign}(0) = 0$. For $a \in \mathbb{R}^p$ and $b \in \mathbb{R}$, define $f_{a,b}$ as

$$f_{a,b}(x) = a^\mathsf{T} x + b.$$

Consider the logistic regression training optimization problem

$$\underset{a \in \mathbb{R}^p,\, b \in \mathbb{R}}{\text{minimize}} \quad \sum_{i=1}^{N} \ell(Y_i f_{a,b}(X_i))$$

with $\ell(z) = \log(1 + e^{-z})$.

(a) Show that $p^\star = 0$, where $p^\star$ denotes the optimal (minimal) value.

(b) Show that there is no $(a, b) \in \mathbb{R}^p \times \mathbb{R}$ attaining the optimal value 0.

Solution:

(a) Note that $Y_i(a_{true}^T X_i + b_{true}) = |a_{true}^T X_i + b_{true}| > 0$. (The norm cannot equal 0 because $Y_i \in \{\pm 1\}$.) Since N is finite, we can choose some positive lower bound

$$0 < \exists t \le |a_{true}^T X_i + b_{true}|, \quad (i = 1, \cdots, N).$$

For any $\varepsilon > 0$, there is a constant C such that

$$e^{-Ct} < e^{\frac{\varepsilon}{N}} - 1,$$

in other words, $\log(1 + e^{-Ct}) < \varepsilon/N$. Take $(Ca_{true}, Cb_{true}) \in \mathbb{R}^p \times \mathbb{R}$, then

$$\begin{aligned} \sum_{i=1}^{N} \ell(Y_i f_{Ca,Cb}(X_i)) &= \sum_i \log(1 + e^{-C|a_{true}^T X_i + b_{true}|}) \\ &\le \sum_i \log(1 + e^{-Ct}) \\ &< \sum_i \frac{\varepsilon}{N} = \varepsilon. \end{aligned}$$

Since ε is arbitrary, we have $p^\star \le 0$. Certainly $p^\star \ge 0$ because $\ell(z) > 0$.

(b) Since $\ell(z)$ is always positive, $\sum_{i=1}^{N} \ell(Y_i f_{a,b}(X_i))$ cannot be 0 for any $(a, b) \in \mathbb{R}^p \times \mathbb{R}$.

4. (10 points) *Log-softmax is idempotent.* Let the softmax function $\mu\colon \mathbb{R}^k \to \Delta^k$ be defined as

$$\big(\mu(z)\big)_i = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}}$$

for all $z \in \mathbb{R}^k$ and $i = 1, \ldots, k$. Let $\log \mu$ be the *log-softmax* function, defined by applying log elementwise. To clarify,

$$\big(\log \mu(z)\big)_i = \log\left(\frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}}\right)$$

for all $z \in \mathbb{R}^k$ and $i = 1, \ldots, k$. Show that

$$\log \mu(\log \mu(z)) = \log \mu(z)$$

for all $z \in \mathbb{R}^k$.

Remark. The idempotence property makes sense if you think of softmax as a kind of normalization.

Solution: Take $z \in \mathbb{R}^k$ and $i = 1, \ldots, k$. From the definition of $\log \mu$ we know

$$e^{\big(\log \mu(z)\big)_i} = e^{\log\big(\mu(z)\big)_i} = \big(\mu(z)\big)_i.$$

Using the fact $\sum_{j=1}^k \big(\mu(z)\big)_j = 1$, we have

$$\big(\log \mu(\log \mu(z))\big)_i = \log\left(\frac{e^{\big(\log \mu(z)\big)_i}}{\sum_{j=1}^k e^{\big(\log \mu(z)\big)_j}}\right) = \log\left(\frac{\big(\mu(z)\big)_i}{\sum_{j=1}^k \big(\mu(z)\big)_j}\right) = \log\big(\mu(z)\big)_i = \big(\log \mu(z)\big)_i.$$

Since z and i are chosen arbitrarily, we get the desired conclusion.

5. (20 points) *LeCun initialization for convolutional layers.* Consider the layer

$$Y = \text{Conv2D}_{w,b}(X)$$

where $\text{Conv2D}_{w,b}$ denotes a 3×3 convolution with padding 1, $w \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times 3 \times 3}$, and $b \in \mathbb{R}^{C_{\text{out}}}$. Assume $X \in \mathbb{R}^{C_{\text{in}} \times m \times m}$. Then, of course, $Y \in \mathbb{R}^{C_{\text{out}} \times m \times m}$. Assume X_{cij} has mean 0 and variance 1 for all c, i, j. Initialize

$$\begin{aligned} w_{ijk\ell} &\sim \mathcal{N}(0, \sigma_w^2) \qquad \text{for all } i, j, k, l \\ b_i &\sim \mathcal{N}(0, \sigma_b^2) \qquad \text{for all } i, \end{aligned}$$

such that $w_{ijk\ell}$ and b_i are IID independent, and w and b are independent from X. Show that

$$\mathbb{E}[Y_{cij}] = 0 \qquad \text{for } c \in \{1, \ldots, C_{\text{out}}\},\ i, j \in \{1, 2, \ldots, m-1, m\}$$

and

$$\mathbb{E}[Y_{cij}^2] = 3^2 C_{\text{in}} \sigma_w^2 + \sigma_b^2 \qquad \text{for } c \in \{1, \ldots, C_{\text{out}}\},\ i, j \in \{2, 3, \ldots, m-2, m-1\}.$$

Solution: For convenience, put $X_{c,0,j}, X_{c,m+1,j}, X_{c,i,0}, X_{c,i,m+1}, X_{c,0,0}, X_{c,m+1,m+1} = 0$. Then

$$Y_{cij} = \sum_{p=1}^{C_{\text{in}}} \sum_{q=1}^{3} \sum_{r=1}^{3} w_{cpqr} X_{p,i+q-2,j+r-2} + b_c.$$

Then,

$$\begin{aligned} \mathbb{E}\left[Y_{cij}\right] &= \mathbb{E}_X \left[\mathbb{E} \left[\sum_{p=1}^{C_{\text{in}}} \sum_{q=1}^{3} \sum_{r=1}^{3} w_{cpqr} X_{p,i+q-2,j+r-2} + b_c \,|\, X \right] \right] \\ &= \mathbb{E}_X \left[\sum_{p=1}^{C_{\text{in}}} \sum_{q=1}^{3} \sum_{r=1}^{3} \mathbb{E}\left[w_{cpqr} \,|\, X\right] X_{p,i+q-2,j+r-2} + \mathbb{E}\left[b_c \,|\, X\right] \right] = 0. \end{aligned}$$

$$\begin{aligned} \mathbb{E}\left[Y_{cij}^2\right] &= \mathbb{E}_X \left[\mathbb{E}\left[Y_{cij}^2 \,|\, X\right] \right] \\ &= \mathbb{E}_X \left[\mathbb{E} \left[\left(\sum_{p=1}^{C_{\text{in}}} \sum_{q=1}^{3} \sum_{r=1}^{3} w_{cpqr} X_{p,i+q-2,j+r-2} + b_c \right)^2 \,\Big|\, X \right] \right] \\ &= \mathbb{E}_X \left[\mathbb{E} \left[\left(\sum_{p=1}^{C_{\text{in}}} \sum_{q=1}^{3} \sum_{r=1}^{3} w_{cpqr} X_{p,i+q-2,j+r-2} \right)^2 \,\Big|\, X \right] \right] \\ &\quad + \mathbb{E}_X \left[2\mathbb{E} \left[\left(\sum_{p=1}^{C_{\text{in}}} \sum_{q=1}^{3} \sum_{r=1}^{3} w_{cpqr} X_{p,i+q-2,j+r-2} \right) b_c \,\Big|\, X \right] \right] + \mathbb{E}_X \left[\mathbb{E}\left[b_c^2 \,|\, X\right] \right] \\ &= \mathbb{E}_X \left[\mathbb{E} \left[\sum_{k=1}^{C_{\text{in}}} \sum_{l=1}^{3} \sum_{n=1}^{3} \sum_{p=1}^{C_{\text{in}}} \sum_{q=1}^{3} \sum_{r=1}^{3} w_{ckln} w_{cpqr} \,|\, X \right] X_{k,i+l-2,j+n-2} X_{p,i+q-2,j+r-2} \right] \\ &\quad + \mathbb{E}_X \left[2\mathbb{E} \left[\left(\sum_{p=1}^{C_{\text{in}}} \sum_{q=1}^{3} \sum_{r=1}^{3} w_{cpqr} b_c \,|\, X \right) X_{p,i+q-2,j+r-2} \right] \right] + \mathbb{E}_X \left[\mathbb{E}\left[b_c^2 \,|\, X\right] \right] \\ &= \mathbb{E}_X \left[\sum_{k=1}^{C_{\text{in}}} \sum_{l=1}^{3} \sum_{n=1}^{3} \sum_{p=1}^{C_{\text{in}}} \sum_{q=1}^{3} \sum_{r=1}^{3} \sigma_w^2 \delta_{kp} \delta_{lq} \delta_{nr} X_{k,i+l-2,j+n-2} X_{p,i+q-2,j+r-2} \right] + 0 + \sigma_b^2 \\ &= 3^2 C_{\text{in}} \sigma_w^2 + \sigma_b^2, \end{aligned}$$

where $\delta_{kp}, \delta_{lq}, \delta_{nr}$ are the Kronecker deltas.

6. (20 points) *ReLU CNNs without biases are positive homogeneous.* Consider a deep ReLU CNN architecture without biases as defined by

```
class Net(nn.Module) :
  def __init__(self) :
    super(Net, self).__init__()
    self.conv_layer1 = nn.Sequential(
        nn.Conv2d(3, 64, kernel_size=3, padding=1, bias=False),
        nn.ReLU(),
        nn.Conv2d(64, 64, kernel_size=3, padding=1, bias=False),
        nn.ReLU(),
        nn.MaxPool2d(kernel_size=2, stride=2) )
    self.conv_layer2 = nn.Sequential(
        nn.Conv2d(64, 128, kernel_size=3, padding=1, bias=False),
        nn.ReLU(),
        nn.Conv2d(128, 128, kernel_size=3, padding=1, bias=False),
        nn.ReLU(),
        nn.MaxPool2d(kernel_size=2, stride=2) )
    self.conv_layer3 = nn.Sequential(
        nn.Conv2d(128, 256, kernel_size=3, padding=1, bias=False),
        nn.ReLU(),
        nn.Conv2d(256, 256, kernel_size=3, padding=1, bias=False),
        nn.ReLU(),
        nn.Conv2d(256, 256, kernel_size=3, padding=1, bias=False),
        nn.ReLU(),
        nn.MaxPool2d(kernel_size=2, stride=2) )
    self.conv_layer4 = nn.Sequential(
        nn.Conv2d(256, 512, kernel_size=3, padding=1, bias=False),
        nn.ReLU(),
        nn.Conv2d(512, 512, kernel_size=3, padding=1, bias=False),
        nn.ReLU(),
        nn.Conv2d(512, 512, kernel_size=3, padding=1, bias=False),
        nn.ReLU(),
        nn.MaxPool2d(kernel_size=2, stride=2) )
    self.fc_layer1 = nn.Sequential(
        nn.Linear(512*2*2, 4096, bias=False),
        nn.ReLU(),
        nn.Dropout() )
    self.fc_layer2 = nn.Sequential(
        nn.Linear(4096, 4096, bias=False),
        nn.ReLU(),
        nn.Dropout() )
    self.fc_layer3 = nn.Linear(4096, 10, bias=False)

  def forward(self, x) :
    output = self.conv_layer1(x)
    output = self.conv_layer2(output)
    output = self.conv_layer3(output)
    output = self.conv_layer4(output)
    output = output.view(-1, 512*2*2)
    output = self.fc_layer1(output)
    output = self.fc_layer2(output)
    return self.fc_layer3(output)
```

where the input X has dimension $B \times 3 \times 32 \times 32$ with batch size B. Write f_θ to denote this neural network, i.e., $f_\theta(X)$ denotes the output of the forward evaluation. Show that

$$f_\theta(\alpha X) = \alpha f_\theta(X) \qquad \text{for all } \alpha \geq 0.$$

Solution: The given function $f_\theta(X)$ is a function composed with sequences of $ReLU$, CNN without any bias, Linear mapping without any bias, MaxPool, and Dropout. To prove that f_θ is positive homogeneous, all we need to show is that each functions are positive homogeneous.

1. **ReLU is positive homogeneous.** For $\alpha > 0$, $\alpha x \geq 0$ iff when $x \geq 0$, thus

$$ReLU(\alpha x) = \begin{cases} \alpha x & \alpha x \geq 0 \\ 0 & \alpha x < 0 \end{cases} = \alpha \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} = \alpha ReLU(x).$$

   Also when $\alpha = 0$, $ReLU(\alpha x) = ReLU(0) = 0 = \alpha ReLU(x)$. Therefore, if $\alpha \geq 0$,

$$ReLU(\alpha X) = \alpha ReLU(X).$$

2. **CNN without any bias is homogeneous.** Consider any window $[i, i + f_1 - 1] \times [j, j + f_2 - 1]$ of X and weight $A = (a_{\gamma,\alpha,\beta}) \in \mathbb{R}^{C_{in} \times f_1 \times f_2}$. Then, for the summation over $(\gamma, \alpha, \beta) \in [1, C_{in}] \times [1, f_1] \times [1, f_2]$, we have

$$\sum_{\gamma,\alpha,\beta} a_{\gamma,\alpha,\beta}(\alpha X)_{\gamma,i+\alpha-1,j+\beta-1} = \alpha \sum_{\gamma,\alpha,\beta} a_{\gamma,\alpha,\beta}(X)_{\gamma,i+\alpha-1,j+\beta-1},$$

   which demonstrates that a CNN without any bias is homogeneous. (The operation is independent across elements of the batch and indices of output channels.)

3. **Linear mapping without any bias is homogeneous.** Since

$$A(\alpha X) = \alpha AX,$$

   linear mapping without any bias is homogeneous.

4. **MaxPool is positive homogeneous** For $\alpha > 0$, consider any set of numbers x_i, $i = 1, \ldots n$. Then, since $\alpha x \geq \alpha y$ iff $x \geq y$, we have

$$\max(\alpha x_1, \ldots, \alpha x_n) = \alpha \max(x_1, \ldots, x_n).$$

   Thus, $MaxPool(\alpha X) = \alpha MaxPool(X)$ and MaxPool is positive homogeneous.

5. **Dropout is homogeneous.** If an element is chosen to be zero, $0(x) \equiv 0$ is homogeneous since $\alpha \times 0 = 0$. If an element is not chosen to be zero, Dropout is an identity, which is homogeneous.

Since homogeneous function is also positive homogeneous and composite of positive homogeneous functions are also positive homogeneous:

$$g_1(g_2(\alpha X)) = g_1(\alpha g_2(X)) = \alpha g_1(g_2(X)),$$

we can conclude that $f_\theta(\alpha X) = \alpha f_\theta(X)$ for $\alpha \geq 0$.

7. (20 points) *Weight decay and ℓ^2 regularization are not the same for SGD with momentum.* Recall that SGD with momentum has the form

$$\begin{aligned} g^k &= \text{stochastic gradient of loss at } \theta^k \\ v^{k+1} &= g^k + \beta v^k \\ \theta^{k+1} &= \theta^k - \alpha v^{k+1} \end{aligned}$$

for $k = 0, 1, \ldots$, where $v^0 = 0$. Define SGD with momentum and weight decay as

$$\begin{aligned} g^k &= \text{stochastic gradient of loss at } \theta^k \\ v^{k+1} &= g^k + \beta v^k \\ \theta^{k+1} &= \theta^k - \alpha v^{k+1} - \mu\theta^k \end{aligned}$$

for $k = 0, 1, \ldots$, where $v^0 = 0$. Define SGD with momentum and ℓ^2 regularization as

$$\begin{aligned} g^k &= \text{stochastic gradient of loss at } \theta^k \\ v^{k+1} &= (g^k + \lambda\theta^k) + \beta v^k \\ \theta^{k+1} &= \theta^k - \alpha v^{k+1} \end{aligned}$$

for $k = 0, 1, \ldots$, where $v^0 = 0$. Suppose $\alpha > 0$ and $\beta > 0$ are fixed. Show that there is no choice of $\mu > 0$ and $\lambda > 0$ such that [SGD with momentum and weight decay] is equivalent to [SGD with momentum and ℓ^2 regularization].

Hint. Consider the case $0 = g^0 = g^1 = g^2 = \cdots$ and $v^0 = 0$. Show that $\theta^k = (1-\mu)^k\theta^0$ for SGD with momentum and weight decay while there is no $\gamma \in \mathbb{R}$ such that $\theta^k = \gamma^k\theta^0$ for SGD with momentum and ℓ^2 regularization.

Solution:

Consider the case where the stochastic gradient g^k always equals 0 for any iteration number k and any choice of parameter θ^k and, which might happen when the loss function is a constant function. We will show that in this case, we can find the counterexample where [SGD with momentum and weight decay] cannot be equivalent to [SGD with momentum and ℓ^2 regularization] in a sense that given the same initial setups (e.g. θ^0, $\alpha > 0$, $\beta > 0$), there is no choice of $\mu > 0$ and $\lambda > 0$ which can make the iterates v^k and θ^k to be the same for two different SGD variants.

Suppose we start from the same initial parameter θ^0 for both SGD variants. Starting with $v^0 = 0$ and $g^k = 0$ for all $k = 0, 1, \ldots$, we use mathematical induction on k to prove that [SGD with momentum and weight decay] results in

$$\theta^k = (1-\mu)^k\theta^0, \quad v^k = 0, \quad k = 0, 1, \ldots. \tag{1}$$

First of all, for $k = 0$, $\theta^0 = (1-\mu)^0\theta^0$ and $v^0 = 0$ holds trivially. Suppose the inductive hypothesis (1) holds true for $k = n \geq 0$. Then

$$\begin{aligned} v^{n+1} &= g^n + \beta v^n = 0 + \beta \cdot 0 = 0 \\ \theta^{n+1} &= \theta^n - \alpha v^{n+1} - \mu\theta^n = (1-\mu)\theta^n = (1-\mu)^{n+1}\theta^0, \end{aligned}$$

so (1) also holds for $k = n + 1$ as well. From mathematical induction, the (1) holds true for all $k \geq 0$.

Now let's observe the iterates v^k and θ^k generated from [SGD with momentum ℓ^2 regularization]. Starting from $v^0 = 0$ and $g^0 = 0$, we get

$$\begin{aligned} v^1 &= (g^0 + \lambda\theta^0) + \beta v^0 = \lambda\theta^0 \\ \theta^1 &= \theta^0 - \alpha v^1 = (1 - \alpha\lambda)\theta^0. \end{aligned}$$

If [SGD with momentum and ℓ^2 regularization] were to satisfy $\theta^k = \gamma^k \theta^0$ for all $k = 0, 1, \ldots$ for some $\gamma \in \mathbb{R}$, such γ should be $\gamma = 1 - \alpha\lambda$. Proceeding with one more update,

$$\begin{aligned} v^2 &= (g^1 + \lambda\theta^1) + \beta v^1 = \lambda(1 - \alpha\lambda)\theta^0 + \beta\lambda\theta^0 \\ &= \lambda(1 + \beta - \alpha\lambda)\theta^0 \\ \theta^2 &= \theta^1 - \alpha v^2 = (1 - \alpha\lambda)\theta^0 - \alpha\lambda(1 + \beta - \alpha\lambda)\theta^0 \\ &= \left\{(1 - \alpha\lambda)^2 - \alpha\beta\lambda\right\}\theta^0. \end{aligned}$$

For $\theta^2 = \gamma^2\theta^0$ to hold true for any initial parameter θ^0,

$$(1 - \alpha\lambda)^2 = \gamma^2 = \left\{(1 - \alpha\lambda)^2 - \alpha\beta\lambda\right\}$$

or equivalently, $\alpha\beta\lambda = 0$ should hold. However, α and β are nonzero, so $\lambda = 0$, which violates the condition $\lambda > 0$.

To conclude, there is no possible way for the θ^k of two different methods to equal each other given the same setup (same θ^0, $\alpha > 0$, and $\beta > 0$) for any $\lambda > 0$ and $\mu > 0$, so two methods are never equivalent.