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

Due 5pm, Monday, May 06, 2024

Problem 1: *Transpose of downsampling.* Consider the downsampling operator $\mathcal{T}\colon \mathbb{R}^{m\times n} \to \mathbb{R}^{(m/2)\times(n/2)}$, defined as the average pool with a 2×2 kernel and stride 2. For the sake of simplicity, assume m and n are even. Describe the action of $\mathcal{T}^\top$. More specifically, describe how to compute $\mathcal{T}^\top(Y)$ for any $Y \in \mathbb{R}^{(m/2)\times(n/2)}$.

Clarification. The downsampling operator $\mathcal{T}$ is a linear operator (why?). Therefore, $\mathcal{T}$ has a matrix representation $A \in \mathbb{R}^{(mn/4)\times(mn)}$ such that

$$\mathcal{T}(X) = (A(X.\text{reshape}(mn))).\text{reshape}(m/2, n/2)$$

for all $X \in \mathbb{R}^{m\times n}$. The adjoint $\mathcal{T}^\top$ has two equivalent definitions. One definition is

$$\mathcal{T}^\top(Y) = (A^\top(Y.\text{reshape}(mn/4))).\text{reshape}(m, n)$$

for all $Y \in \mathbb{R}^{(m/2)\times(n/2)}$. Another is

$$\sum_{i=1}^{m/2}\sum_{j=1}^{n/2} Y_{ij}(\mathcal{T}(X))_{ij} = \sum_{i=1}^{m}\sum_{j=1}^{n}(\mathcal{T}^\top(Y))_{ij}(X)_{ij}$$

for all $X \in \mathbb{R}^{m\times n}$ and $Y \in \mathbb{R}^{(m/2)\times(n/2)}$.
Hint. To spoil the suspence, $\mathcal{T}^\top$ is a constant times the nearest neighbor upsampling. Explain why in your answer.

Problem 2: *Nearest neighbor upsampling.* How is the nearest neighbor upsampling operator an instance of transpose convolution? Specifically, describe how

```
layer = nn.Upsample(scale_factor=r, mode='nearest')
```

where r is a positive integer, can be equivalently represented by

```
layer = nn.ConvTranspose2d(...)
layer.weight.data = ...
```

with `...` appropriately filled in.

Problem 3: *f-divergence.* Let X and Y be two continuous random variables with densities p_X and p_Y. The f-divergence of X from Y is defined as

$$D_f(X\|Y) = \int f\left(\frac{p_X(x)}{p_Y(x)}\right) p_Y(x)\, dx,$$

where f is a convex function such that $f(1) = 0$.

(a) Show that $D_f(X\|Y) \geq 0$.

(b) Show that $f = -\log t$ and $f = t \log t$ correspond to the KL divergence.

Problem 4: *Generalized inverse transform sampling.* Let $F\colon \mathbb{R} \to [0,1]$ be the CDF of a random variable and let $U \sim \text{Uniform}([0,1])$. If F is continuous and strictly increasing and therefore invertible, then $F^{-1}(U)$ is a random variable with CDF F, because

$$\mathbb{P}(F^{-1}(U) \leq t) = \mathbb{P}(U \leq F(t)) = F(t).$$

When F is not necessarily invertible, the *generalized inverse* of F is $G\colon (0,1) \to \mathbb{R}$ with

$$G(u) = \inf\{x \in \mathbb{R} \,|\, u \leq F(x)\}.$$

Show that $G(U)$ is a random variable with CDF F.

Hint. Use the fact that F is right-continuous, i.e., $\lim_{h\to 0^+} F(x+h) = F(x)$ for all $x \in \mathbb{R}$, and that $\lim_{x\to -\infty} F(x) = 0$.

Problem 5: *Change of variables formula for Gaussians.* If $\varphi\colon \mathbb{R}^n \to \mathbb{R}^n$ is a one-to-one differentiable function, $Y = \varphi(X)$, and Y is a continuous random variable with density function p_Y, then X is a continuous random variable with density function

$$p_X(x) = p_Y(\varphi(x)) \left|\det \frac{\partial \varphi}{\partial x}(x)\right|.$$

Let $Y \in \mathbb{R}^n$ be a continuous random vector with density

$$p_Y(y) = \frac{1}{(2\pi)^{n/2}} e^{-\frac{1}{2}\|y\|^2},$$

i.e., $Y \sim \mathcal{N}(0, I)$. Let $X = AY + b$ with an invertible matrix $A \in \mathbb{R}^{n\times n}$ and a vector $b \in \mathbb{R}^n$. Define $\Sigma = AA^\intercal$. Show that X is a continuous random vector with density

$$p_X(x) = \frac{1}{\sqrt{(2\pi)^n \det \Sigma}} e^{-\frac{1}{2}(x-b)^\intercal \Sigma^{-1}(x-b)}.$$

Problem 6: *Inverse permutation.* Let S_n denote the group of length-n permutations. Note that the map $i \mapsto \sigma(i)$ is a bijection. Define $\sigma^{-1} \in S_n$ as the permutation representing the inverse of this map, i.e, $\sigma^{-1}(\sigma(i)) = i$ for $i = 1, \ldots, n$. Describe an algorithm for computing σ^{-1} given σ.

Clarification. In this class, we defined σ as a list of length n containing the elements of $\{1, \ldots, n\}$ exactly once. The output of the algorithm, σ^{-1}, should also be provided as a list.

Clarification. For this problem, it is sufficient to describe the algorithm in equations or pseudocode. There is no need to submit a Python script for this problem.

Problem 7: *Permutation matrix.* Given a permutation $\sigma \in S_n$, the *permutation matrix* of σ is defined as

$$P_\sigma = \begin{bmatrix} e_{\sigma(1)}^\intercal \\ e_{\sigma(2)}^\intercal \\ \vdots \\ e_{\sigma(n)}^\intercal \end{bmatrix} \in \mathbb{R}^{n \times n},$$

where $e_1, \ldots, e_n \in \mathbb{R}^n$ are the standard unit vectors. Show

(a) $(P_\sigma x)_i = x_{\sigma(i)}$ for all $x \in \mathbb{R}^n$ and $i = 1, \ldots, n$,

(b) $P_\sigma^\intercal = P_\sigma^{-1} = P_{\sigma^{-1}}$ and

(c) $|\det P_\sigma| = 1$.

Hint. If the rows of $U \in \mathbb{R}^{n \times n}$ are orthonormal, we say U is an orthogonal matrix. Orthogonal matrices satisfy $UU^\intercal = U^\intercal U = I$.