Mathematical Foundations of Deep Neural Networks, M1407.001200
E. Ryu
Fall 2021

Homework 6
Due 5pm, Tuesday, November 9, 2021

Problem 1: *Removing BN after training.* During training, the addition of batch norm adds additional operations that were otherwise not present and therefore increases the computational cost per iteration. During testing, however, the effect of batch normalization can be combined with the preceding convolutional or linear layer so that no additional computational cost is incurred. Download the starter code `bn_remove.py` and the save file `smallNetSaved` and carry out the removal of the batchnorm layers. Specifically, load the pre-trained `smallNetTrain` model and set the weights and parameters of `smallNetTest` so that the two models produce exactly the same outputs on the test set.

Problem 2: *Default weight initialization.* Consider the multi-layer perceptron

$$\begin{aligned} y_L &= A_L y_{L-1} + b_L \\ y_{L-1} &= \sigma(A_{L-1} y_{L-2} + b_{L-1}) \\ &\vdots \\ y_2 &= \sigma(A_2 y_1 + b_2) \\ y_1 &= \sigma(A_1 x + b_1), \end{aligned}$$

where $x \in \mathbb{R}^{n_0}$, $A_\ell \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$, $b_\ell \in \mathbb{R}^{n_\ell}$, and $n_L = 1$. For the sake of simplicity, let

$$\sigma(z) = z.$$

Assume $x_1, \dots, x_{n_0}$ are IID with zero-mean and unit variance. If this network is initialized with the default weight initialization of PyTorch, what will the mean and variance of y_L be?

Clarification. For this problem, you are being asked to read the PyTorch source code
`https://pytorch.org/docs/stable/_modules/torch/nn/modules/linear.html`
to identify the default initialization behavior and then to perform calculations.

Problem 3: *Backprop with convolutions.* Consider 1D convolutions with single input and output channels, stride 1, and padding 0. Let $w_1, \dots, w_L$ be convolutional filters with sizes $f_1, \dots, f_L$. Let $A_{w_\ell} \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$, where $n_\ell = n_{\ell-1} - f_\ell + 1$, be the matrix representing convolution with w_ℓ, i.e., multiplication by A_{w_ℓ} is equivalent to convolution with w_ℓ, for $\ell = 1, \dots, L$. Let $\sigma\colon \mathbb{R} \to \mathbb{R}$ be a differentiable activation function. Consider the convolutional neural network

$$\begin{aligned} y_L &= A_{w_L} y_{L-1} + b_L \mathbf{1}_{n_L} \\ y_{L-1} &= \sigma(A_{w_{L-1}} y_{L-2} + b_{L-1} \mathbf{1}_{n_{L-1}}) \\ &\vdots \\ y_2 &= \sigma(A_{w_2} y_1 + b_2 \mathbf{1}_{n_2}) \\ y_1 &= \sigma(A_{w_1} x + b_1 \mathbf{1}_{n_1}), \end{aligned}$$

where $x \in \mathbb{R}^{n_0}$, $b_\ell \in \mathbb{R}$, $\mathbf{1}_{n_\ell} \in \mathbb{R}^{n_\ell}$ is the vector with all entries being 1, and $n_L = 1$. Assume x is fixed and $y_1, \dots, y_L$ have been computed in a forward pass. For notational convenience, define $x = y_0$. Find formulae for

$$\frac{\partial y_L}{\partial w_\ell}, \qquad \frac{\partial y_L}{\partial b_\ell}$$

for $\ell = 1, \dots, L$ and describe how to compute them using backpropagation. As discussed in homework 1, forming the full matrix A_{w_ℓ} is wasteful and should be avoided. In the description, make clear when matrix-vector or vector-matrix products with respect to A_{w_i} or $A_{w_i}^\intercal$ are used.

Clarification. A matrix-vector product $A_{w_i} v$ should be computed by performing convolution. A vector-matrix product $u^\intercal A_{w_i} = (A_{w_i}^\intercal u)^\intercal$ should be computed by performing transpose-convolution, which was discussed in homework 1.

Hint. Define $A_\ell(w_\ell) = A_{w_\ell}$ and $\beta_\ell(b_\ell) = b_\ell \mathbf{1}_{n_\ell}$ and write

$$\begin{aligned} y_L &= A_L(w_L) y_{L-1} + \beta_L(b_L) \\ y_{L-1} &= \sigma(A_{L-1}(w_{L-1}) y_{L-2} + \beta_{L-1}(b_{L-1}) \\ &\vdots \\ y_2 &= \sigma(A_2(w_2) y_1 + \beta_2(b_2)) \\ y_1 &= \sigma(A_1(w_1) x + \beta_1(b_1)). \end{aligned}$$

Compute

$$\frac{\partial y_L}{\partial A_\ell}, \qquad \frac{\partial A_\ell}{\partial w_\ell}, \qquad \frac{\partial y_L}{\partial \beta_\ell}, \qquad \frac{\partial \beta_\ell}{\partial b_\ell}.$$

and use the chain rule.

Problem 4: *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 5: D_{KL} *of continuous random variables.* The KL-divergence between continuous random variables $X \sim f$ and $Y \sim g$, where f and g are probability density functions in $\mathbb{R}^d$, is

$$D_{\mathrm{KL}}\left(X\|Y\right) = \int_{\mathbb{R}^d} f(x) \log\left(\frac{f(x)}{g(x)}\right)\, dx.$$

(a) Show that

$$D_{\mathrm{KL}}\left(X\|Y\right) \geq 0.$$

(b) Show that if $X = (X_1, \ldots, X_d)$ is a continuous random variable such that $X_1, \ldots, X_d$ are independent and $Y = (Y_1, \ldots, Y_d)$ is a continuous random variable such that $Y_1, \ldots, Y_d$ are independent, then

$$D_{\mathrm{KL}}(X\|Y) = D_{\mathrm{KL}}(X_1\|Y_1) + \cdots + D_{\mathrm{KL}}(X_d\|Y_d).$$

Problem 6: D_{KL} *of Gaussian random variables.* Let $\mathcal{N}(\mu, \Sigma)$ denote the Gaussian distribution with mean μ and covariance Σ. So if $X \sim \mathcal{N}(\mu, \Sigma)$, then

$$\mathbb{E}[X] = \mu, \qquad \mathbb{E}[(X-\mu)(X-\mu)^\intercal] = \Sigma.$$

Show that

$$D_{\mathrm{KL}}\left(\mathcal{N}(\mu_0, \Sigma_0)\|\mathcal{N}(\mu_1, \Sigma_1)\right) = \frac{1}{2}\left(\operatorname{tr}\left(\Sigma_1^{-1}\Sigma_0\right) + (\mu_1 - \mu_0)^\intercal \Sigma_1^{-1}(\mu_1 - \mu_0) - d + \log\left(\frac{\det \Sigma_1}{\det \Sigma_0}\right)\right),$$

where d is the underlying dimension of the random variables $\mathcal{N}(\mu_0, \Sigma_0)$ and $\mathcal{N}(\mu_1, \Sigma_1)$. Assume Σ_0 and Σ_1 are positive definite.