Homework 2

1. **(3 points)** *Designing convolution filters by hand*. Consider an input 2D image and a 3×3 filter (say w) applied to this image. The goal is to guess good filters which implement each of the following elementary image processing operations.

   a. Write down the weights of w which acts as a *blurring* filter, i.e., the output is a blurry form in the input.

   b. Write down the weights of w which acts as a *sharpening* filter in the horizontal direction.

   c. Write down the weights of w which acts as a *sharpening* filter in the vertical direction.

   d. Write down the weights of w which act as a *sharpening* filter in a *diagonal* (bottom-left to top-right) direction.

   Solution

   These are not unique solutions; others are also possible.

   a.
   $$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

   b.
   $$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

   c.
   $$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

   d.
   $$\begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$$

2. **(3 points)** *Weight decay*. The use of ℓ_2 regularization for training multi-layer neural networks has a special name: *weight decay*. Assume an arbitrary dataset $\{(x_i, y_i)\}_{i=1}^n$ and a loss function $\mathcal{L}(w)$ where w is a vector that represents all the trainable weights (and biases).

   a. Write down the ℓ_2 regularized loss, using a weighting parameter λ for the regularizer.

   b. Derive the gradient descent update rules for this loss.

   c. Conclude that in each update, the weights are "shrunk" or "decayed" by a multiplicative factor before applying the descent update.

d. What does increasing λ achieve algorithmically, and how should the learning rate be chosen to make the updates stable?

Solution

a. The new loss is $\bar{\mathcal{L}}(w) = \mathcal{L}(w) + \lambda\|w\|_2^2$.

b. If the learning rate is η, then the gradient update rule is:

$$\begin{aligned} w_{t+1} &= w_t - \eta\nabla\bar{\mathcal{L}}(w_t) \\ &= w_t - \eta\nabla\mathcal{L}(w_t) - 2\eta\lambda w_t \\ &= (1 - 2\eta\lambda)w_t - \eta\nabla\mathcal{L}(w_t). \end{aligned}$$

c. From the above update equations, we can see that the updated w consists of shrinking/decaying the current w by a factor $(1 - 2\eta\lambda)$ and then updating in the direction of the gradient.

d. Increasing λ penalizes the ℓ_2 norm of the weight vector, and therefore enforces smaller weights on average. In order for the gradient dynamics to be stable, the contraction factor should be smaller than 1, i.e., $\eta < 1/2\lambda$.

3. **(2 points)** *The IoU metric*. In class we defined the IoU metric (or the Jaccard similarity index) for comparing bounding boxes.

a. Using elementary properties of sets, argue that the IoU metric between any two pair of bounding boxes is always a non-negative real number in $[0, 1]$.

b. If we represent each bounding box as a function of the top-left and bottom-right coordinates (assume all coordinates are real numbers) then argue that the IoU metric is *non-differentiable* and hence cannot be directly optimized by gradient descent.

Solution

a. The definition of IOU for any two bounding boxes A and B is given by:

$$IOU(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

Since the right hand side is non-negative, this number has to be bigger than (or equal to) 0. Moreover, $A \cap B \subseteq A \cup B$, and hence the numerator has to be no bigger than the denominator. Therefore the IOU is bounded between 0 and 1 (inclusive).

b. Here is a simple counter-example. Let's take two identical size boxes A and B (say, square), both aligned at the same horizontal level. Fix B and then imagine "sliding" A from left to right. As A moves, the IOU will start from zero (no overlap), increase (until there is perfect overlap), and then decrease (until there is no overlap again). So, if we plot IOU as a function of horizontal displacement, we should get a curve like this:

which has 3 kinks and hence is non-differentiable.

4. **(4 points)** *Training AlexNet*. Open the (incomplete) Jupyter notebook provided as an attachment to this homework in Google Colab (or other Python IDE of your choice) and complete the missing items.

Solution