Gradient of Multi-class SVM Loss Function

Given:

- W is weight with a shape of D × C.
- β_{nj} is a binary matrix to indicate the sign of margins.
- $\sum_j \beta_{nj}$ is the count of positive margins for the jth sample.

$$f(x_n, W)_c = x_{nd} W_{dc}$$

$$L_n = \sum_{j \neq y_n} \max(0, x_{nd} W_{dj} - x_{nd} W_{d,y_n} + \Delta)$$

$$L = \frac{1}{N} \sum_{n}^{N} \sum_{j \neq y_n} \max(0, x_{nd} W_{dj} - x_{nd} W_{d,y_n} + \Delta) + \lambda \sum_{c=1}^{C} \sum_{d} W_{c,d}^2$$

The gradient of the loss function with respect to the weight matrix W is:

$$\beta_{nj} = \begin{cases} 1 & \text{if } (x_{nd} W_{dj} - x_{nd} W_{d,y_n} + \Delta) > 0 \\ 0 & \text{if } (x_{nd} W_{dj} - x_{nd} W_{d,y_n} + \Delta) < 0 \text{ and } j = y_n \end{cases}$$

Thus,

$$\begin{aligned} L_n &= \sum_{j \neq y_n} \beta_{nj} (x_{nd} W_{dj} - x_{nd} W_{d,y_n} + \Delta) \\ &= \sum_{j} (1 - \delta_{j,y_n}) \, \beta_{nj} (x_{nd} W_{dj} - x_{nd} W_{d,y_n} + \Delta) \end{aligned}$$

The term $1 - \delta_{j,y_n}$ can be encoded using one-hot encoding.

For all samples:

$$L = \frac{1}{N} \sum_{n} \sum_{j} (1 - \delta_{j,y_n}) \, \beta_{nj} (x_{nd} W_{dj} - x_{nd} W_{d,y_n} + \Delta) + \lambda \sum_{c=1}^{C} \sum_{d} W_{c,d}^2$$

The derivatives are:

$$\frac{\partial L_n}{\partial W_{d,y_n}} = - \sum_{j \neq y_n} \beta_{nj} x_{nd}$$

$$\frac{\partial L_n}{\partial W_{dc}} = \beta_{nc} x_{nd}, \text{ when } c \neq y_n$$

For all samples:

$$\frac{\partial L}{\partial W_{d,y_n}} = -\sum_{n=1}^{N} \sum_{j \neq y_n} \beta_{nj} x_{nd}$$

$$\frac{\partial L}{\partial W_{dc}} = \sum_{n=1}^{N} \beta_{nc} x_{nd}, \text{ when } c \neq y_n$$

The final gradient of the loss function with respect to W is:

$$\begin{aligned}\frac{\partial L}{\partial W_{dc}} &= \frac{1}{N} \sum_{n=1}^{N} \left(\beta_{nc} + \delta_{c,y_n} \left(-\sum_{j \neq y_n} \beta_{nj} - \beta_{nc} \right) \right) x_{nd} + 2\lambda W_{dc} \\ &= \frac{1}{N} \sum_{n=1}^{N} \left(\beta_{nc} - \sum_{j} \beta_{nj} \right) x_{nd} + 2\lambda W_{dc}\end{aligned}$$

Question 1:

It is possible that once in a while a dimension in the gradcheck will not match exactly. What could such a discrepancy be caused by? Is it a reason for concern? What is a simple example in one dimension where a gradient check could fail? How would change the margin affect of the frequency of this happening? *Hint: the SVM loss function is not strictly speaking differentiable.

Answer:

Potential Causes of Discrepancies

- **Non-Differentiability:** The SVM loss function involves the max function, which is not differentiable at the point where its arguments are equal. This non-differentiability can cause issues when checking gradients numerically, especially if the numerical gradient calculation falls on the non-differentiable point.
- **Finite Precision:** Numerical gradients are calculated using finite differences, which are approximations. Due to floating-point precision limitations, these approximations can lead to small errors, especially in high-dimensional spaces or with very small gradient values.
- **Choice of Epsilon:** The choice of epsilon (the perturbation value used to compute numerical gradients) can affect accuracy. If epsilon is too large or too small, it can lead to inaccuracies in the gradient approximation.

- **Implementation Errors:** There could be bugs or implementation issues in either the analytical gradient computation or the numerical gradient computation, leading to discrepancies.

Example of Gradient Check Failure in One Dimension

Consider the simplified SVM loss function in one dimension where the margin is Δ. For a single sample and a single dimension, the loss function can be expressed as:

$$L = \max(0, W_1 x - W_2 x + \Delta)$$

where W_1 and W_2 are the weights for the classes, and x is the feature value. Let's calculate the analytical and numerical gradients.

*Analytical Gradient

For simplicity, assume:

$$L = \max(0, W_1 x - W_2 x + \Delta)$$

- If $W_1 x - W_2 x + \Delta > 0$:
  - The gradient with respect to W_1 is x.
  - The gradient with respect to W_2 is $-x$.
- If $W_1 x - W_2 x + \Delta \leq 0$:
  - The gradient is 0 for both W_1 and W_2.

*Numerical Gradient

To compute the numerical gradient, perturb W_1 and W_2 slightly and use the difference quotient:

$$\text{Numerical Gradient} = \frac{L(W + \epsilon) - L(W - \epsilon)}{2\epsilon}$$

Simple Example of Gradient Discrepancy

Consider a simplified case where:

$$\begin{aligned} x &= 1 \\ W_1 &= 1 \\ W_2 &= 1 \\ \Delta &= 0 \end{aligned}$$

The loss function simplifies to:

$$L = \max(0, 1 \cdot 1 - 1 \cdot 1 + 0) = \max(0, 0) = 0$$

Here, the gradient is 0 for both W_1 and W_2 because L is 0 and not positive.

If you perturb W_1 slightly, the loss might become positive if W_1 increases, but it may not affect W_2 if both weights are initially equal. Due to the non-differentiable point at $L = 0$, the gradient check might show discrepancies near this point, especially if numerical gradient computations are sensitive to perturbations.

Impact of Changing the Margin

The margin Δ affects the range over which the max function evaluates positive values. Changing the margin affects:

- **Frequency of Non-Differentiable Points:** A larger margin increases the range where L can be positive, potentially affecting the frequency of non-differentiable points.
- **Gradient Magnitudes:** A larger margin can increase the value of the loss, which might result in larger gradients and potentially reduce the relative impact of numerical approximation errors.

Question 2:

Describe what your visualized SVM weights look like, and offer a brief explanation for why they look the way they do.

*A*nswer:

The weights can be thought of as defining a template or prototype for each class. For example, if you're classifying images of cars and frogs, the weight vectors might align in such a way that they represent features typical of each class. A "car" weight vector might highlight features common to car images, while a "frog" weight vector might emphasize characteristics common to frog images.