Loss Function

Our loss function is defined as follow:

$$C(X) = \left(\frac{1}{n}\right) \sum_{j}^{dmax} (|SAD(X_j - X_1) - Y_j|)$$

Where X and Y are the input and target vectors respectively.
Wrt to 3D stereo matching context, every element X_j is a 2D-image matrix representing 9x9 patches of the target image whilst X_1 is the patch of the reference image and SAD is the 'Sum of Absolute Differences' cost. So, leaving out the average - which is optional - and the outer sum that computes the total error, the function on which we have to compute the gradient will be:

$$\nabla(C(X)) = \nabla(|SAD(X_j - X_1) - Y_j|)$$

The gradient contains $dmax$ derivatives, each for each X_j patch. Hence, let's now focus on a single image patch X_j.
Given $g(X_j) = (|f(X_j) - y|)$, $f(X_j) = SAD(X_j - X_1) = \sum_i^P |x_{ij} - x_{i1}|$ and $h(X_j) = g(f(X_j))$ we can write the derivative of each example following the chain-rule:

$$\frac{\mathrm{d}h(X_j)}{\mathrm{d}X_j} = g'_{(}f(X_j)) \cdot f'(X_j)$$
$$g'(X_j) = sgn(f(X_j) - Y_j)$$
$$f'(X_j) = sgn(X_j - X_1)$$

Note that the derivative of the SAD function boils down to the derivative of the abs function. The former is motivated by the fact that our input is the entire image X_j, not the single pixels x_{ij}, thus the external sum can be ignored.
To obtain the final gradient, we need to compute the above derivatives for every image patch (i.e. every input X_j)

Implementation in Torch Criterion

```
for i=1,self.dmax do
   local diff = input[i] - ref
   SAD[i] = (torch.sum(torch.abs(input[i])) / self.maxDiff)
   d_SAD[i] =  torch.sign(input[i]) / self.maxDiff
 end
 --chain-rule derivatives
 local dg_dx  = torch.sign(SAD - target) --(DMAX)x1
 local df_dx = d_SAD -- (DMAX)x1x9x9

 for i=1,self.dmax do
     gradInput[i] = dg_dx[i] * df_dx[i]
 end
```