Your Presentation

You

Where You're From

Date of Presentation

Outline

Taylor decomposition

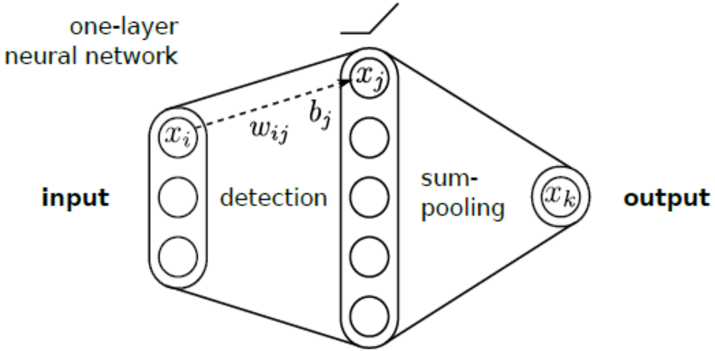
Example

Task description

Taylor Decomposition

- Redistribute the neural network output onto the input variables
- Taylor expansion of a function $f(x)$ at a:
  $f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \cdots$
- $f(\mathbf{x}) = f(\tilde{\mathbf{x}}) + \left(\frac{\partial f}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\tilde{\mathbf{x}}}\right)^T (\mathbf{x} - \tilde{\mathbf{x}}) + \epsilon$
  $0 + \sum_p \underbrace{\frac{\partial f}{x_p}\Big|_{\mathbf{x}=\tilde{\mathbf{x}}}(x_p - \tilde{x}_p)}_{R_p(\mathbf{x})} + \epsilon$

Example (1/2)



- $x_j = max(0, \sum_i x_i w_{ij} + b_j)$ (ReLU nonlinearity)
- $x_k = \sum_j x_j$ (Sum pooling)

Example (2/2)

R_k of output layer: Total relevance that must be backpropagated:

- $R_k = x_k = \sum_j x_j$

R_j of hidden layer: Taylor decomposition on $\{\tilde{x}_j\} = 0$:

- $R_j = \frac{\partial R_k}{\partial x_j}\Big|_{\{\tilde{x}_j\}} \cdot (x_j - \tilde{x}_j) = x_j = max(0, \sum_i x_i w_{ij} + b_j)$

R_i of input layer:

- $R_i = \sum_j \frac{\partial R_j}{\partial x_i}\Big|_{\{\tilde{x}_i\}^{(j)}} \cdot (x_i - \tilde{x}_i^{(j)})$
- $R_i = \sum_j \frac{w_{ij}^2}{\sum_{i'} w_{i'j}^2} R_j$

Task

The task contains two parts

1. Numerical task
   - Use the equations above to compute numerically the relevance of all layers of the network depicted in figure 4.
   - Use your own weight values (w_{ij}), but think on weighting schemes that are typically used in neural networks.
   - Verify that the conservation and positivity rules properties apply.
   - Provide descriptions of the interpretations
2. Programmatic task
   - Install, run.
   - Change the number of training steps and see how the computed relevance changes.
   - Provide descriptions of the interpretations of the relevance images with respect to the input images.

Literature