Supplementary Data

DeepSig: deep learning improves signal peptide detection in proteins

Castrense Savojardo, Pier Luigi Martelli, Piero Fariselli and Rita Casadio

1. Evaluating residue positional relevance with deep Taylor decomposition

The DCNN described in Section 2.2 of the Main Text provides a prediction of the presence/absence of the signal peptide sequence in the N-terminus of any input protein. With DCNN, some of the elements of the input sequence (i.e. individual residues) may be more determinant than others, in driving the model classification towards one specific class. An important question is then how this piece of information can be extracted from the analysis of the internal neuronal activity of DCNN.

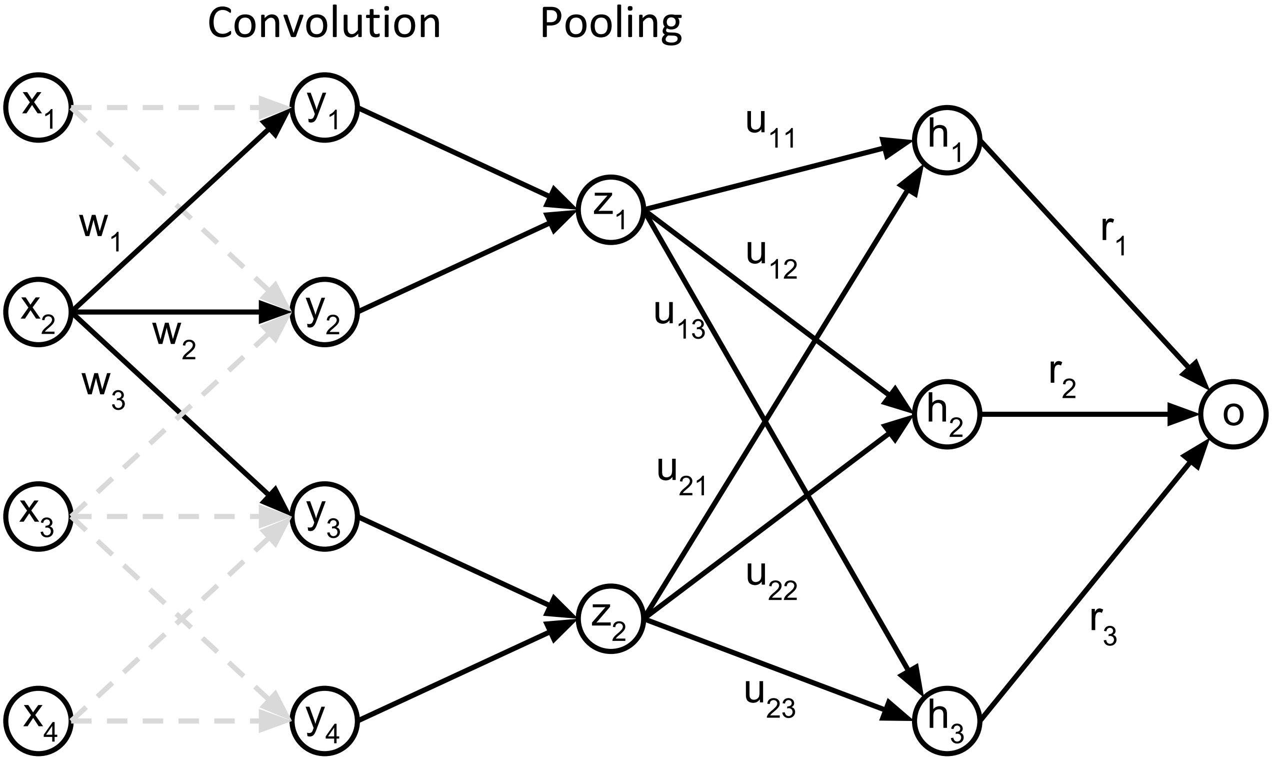
Here, we adopted the deep Taylor decomposition (Montavon *et al.*, 2017), a hybrid functional/message passing approach that has been recently introduced for the analysis of deep neural networks. The method focuses on image classification, but it can be easily extended to other types of prediction scenarios, such as protein sequence classification. We briefly describe here its main aspects and refer to the original paper for a complete mathematical description of the method (Montavon *et al.*, 2017).

Let be $\mathbf{x} = [\mathbf{x}_1, \cdots, \mathbf{x}_l]$ an input protein sequence of length l where each $\mathbf{x}_i \in \mathbb{R}^{20}$ is a 20-channel vector representing a residue in the sequence. $f(\mathbf{x}) \in \mathbb{R}$ is the scalar function implemented by the DCNN and evaluated on the input $\mathbf{x}$. The function $f(\mathbf{x})$ quantifies the evidence (or score) that a signal peptide is present in the N-terminus of the sequence $\mathbf{x}$. We want to assign to each residue $\mathbf{x}_i$ a *relevance score* $R_{\mathbf{x}_i}$ that quantifies the individual contribution of that residue to the total predicted evidence function $f(\mathbf{x})$.

The main properties of deep Taylor decomposition can be summarized in few key points (Montavon *et al.*, 2017). Firstly, the global evidence function $f(\mathbf{x})$ is decomposed into a set of sub-functions on the basis of the specific connectivity between neurons of the network. Secondly, a relevance score is propagated from upper to lower layers of the network through local evaluations of Taylor expansions of the relevance function. Thirdly, explicit propagation rules exploiting the local network connectivity are defined to propagate the total relevance evaluated at the network output back to the input variables. These rules are defined in order to take into account all possible (i.e from other neurons) contributions to the relevance computed at a given internal neuron. At each propagation stage, the relevance is redistributed from neurons at a given layer to neurons in the connected lower layer.

To understand how deep Taylor decomposition works consider the network showed in Supplementary Fig. 1.

Supplementary Fig. 1. A simple DCNN mapping four input variables to a single output through one convolution-pooling stage.



The network is a simplified version of our DCNN, processing a single-channel input sequence of length four with a single convolutional layer stage with one motif of width three, sum pooling and final linear mapping to a single output through three hidden units.

Mathematically, the function $f(x_1, x_2, x_3, x_4)$ computed by the network can be decomposed as follows:

$$y_j = \max\left\{0, \sum_{i=j-1}^{j+1} x_i w_{j-i+1}\right\} \quad (1)$$

$$z_k = \sum_{j=k}^{k+1} y_j \quad (2)$$

$$h_l = \max\{0, \sum_k z_k u_{kl} + b_l\} \quad (3)$$

$$o = \sum_l r_l h_l \quad (4)$$

where w is the motif weight matrix and b_l are bias parameters.

Firstly, the total relevance assigned by the model is the predicted output, i.e. $R_o = f(x_1, x_2, x_3, x_4) = o$. From Eq. 4, the relevance R_o can be expressed as a function of the hidden-layer neurons as follows:

$$R_o = g(h_1, h_2, h_3) = \sum_l r_l h_l \tag{5}$$

In Equation 5 a direct mapping is established between neurons h_l and the relevance R_o. This allows to defined the relevance value R_{h_l} for a given hidden-neuron h_l in terms of a local first-order Taylor expansion of the mapping function $g = R_o$ at some well-chosen *root point* $\tilde{h}_l$ (i.e. a point where $g(\tilde{h}_l) = 0$):

$$R_{h_l} = \frac{\partial R_o}{\partial h_l} | _{\tilde{h}_l} \times (h_l - \tilde{h}_l) \tag{6}$$

where $\frac{\partial g}{\partial h_l} | _{\tilde{h}_l}$ is the gradient of R_o with respect to h_l evaluated at the root point $\tilde{h}_l$. It can be shown that, given the functional form of h_l (Eq. 3) the only admissible root point is $\tilde{h}_l = 0$ (Montavon *et al.*, 2016).

Since $\frac{\partial g}{\partial h_l} = 1$, it follows that:

$$R_{h_l} = h_l = \max\{0, \sum_k z_k u_{kl} + b_l\} \tag{7}$$

In other words, relevance R_{h_l} is proportional to the actual activation of each hidden-neuron. Going one step backward, relevance scores R_{h_l} are redistributed to pooling layer neurons z_k. Again, computing local Taylor expansion of the function R_{h_l} we have that:

$$R_{z_k} = \sum_l \frac{\partial R_{h_l}}{\partial z_k} | _{\tilde{z}_k^{(l)}} \left(z_k - \tilde{z}_k^{(l)}\right) \tag{8}$$

where $\tilde{z}_k^{(l)}$ is a well-chosen root point (here depending on k and l), and l is an index for all neurons to which z_k is connected to. Several methods are presented by authors for choosing the proper root point at this stage, each method giving rise to different relevance propagation rules (Montavon *et al.*, 2017). In particular, when the input space is unconstrained the root point can be always chosen as the closest point by Euclidean distance to the z_k point. In contrast, when the space is constrained, the search domain must be restricted in order to consider feasible root points. Here, since the Rectified Linear Unit (ReLU) activations are used in the previous layer, the pooling layer output is a constrained input space ($\mathbb{R}_+^2$). We can then apply the so-called *z-rule* (Montavon *et al.*, 2017), leading to the following propagation formula:

$$R_{z_k} = \sum_l \frac{u_{kl}^+ z_k}{\sum_{k'} u_{k'l}^+ z_{k'}} R_{h_l} \tag{9}$$

where u_{kl}^+ denotes the positive part of u_{kl}.

Adopting the same concepts, relevance scores are back-propagated to convolution and input layers. Analogously to the hidden-layer, for the convolution relevance R_{y_j} we have the following:

$$R_{y_j} = y_j = \max\left\{0, \sum_{i=j-1}^{j+1} x_i w_{j-i+1}\right\} \tag{10}$$

Finally, for the input layer, according to the z-rule we have that:

$$R_{x_i} = \sum_{j=i-1}^{i+1} \frac{w_{i-j+1}^+ x_i}{\sum_{i'=j-1}^{j+1} w_{j-i'+1}^+ x_{i'}} R_{y_j} \tag{11}$$

where, again, w_i^+ denotes the positive part of w_i. Note that in Eq. 11, the relevance R_{x_i} takes into consideration both the contribution of x_i to connected y_j (the numerator of the fraction) and the contributions of inputs $x_{i'}$ surrounding x_i (the normalizing denominator of the fraction) to each y_j.

In summary, deep Taylor decomposition allows to assign to each neuron in a deep network a relevance score proportional to the contribution of the neuron to the total predicted score. Neuron relevance scores are computed by establishing local, connectivity-dependent functional mapping between neurons activations and propagated relevance values from upper-layers. Taylor expansions of local mappings at neuron-specific root points are then computed. Depending on the functional form of the mappings and on the nature of the input domain, different relevance propagation rules are defined.

We apply this procedure to our signal peptide DCNN to evaluate the contribution of each residue position to the detection of the signal sequence. In other words, when applied to a given input sequence of length $l = 96$, deep Taylor decomposition gives a vector:

$$\left(R_{\mathbf{x}_1}, \cdots, R_{\mathbf{x}_l}\right) \tag{12}$$

where the component $R_{\mathbf{x}_\mathrm{i}}$ is the relevance of the residue in position i.