

Learning inside the Rete algorithm

YKY (Yan King Yin)

general.intelligence@gmail.com

February 17, 2019

Abstract

1 Introduction to the Rete algorithm

The general form of a **production rule** is like this:

$$\overbrace{\text{green circle} \wedge \text{green circle} \wedge \text{green circle} \dots}^{\text{condition}} \rightarrow \overbrace{\text{red circle}}^{\text{action}} \quad (1)$$

where \wedge denotes logical conjunction (AND).

Typically, we would be trying to match a relatively small number of **facts** (that represent the current **state**, or **working memory**) against a very large number of **rules**:

$$\underbrace{\dots \text{yellow circle} \text{yellow circle} \text{yellow circle} \dots}_{\text{queue of WMEs}} \quad \text{match against} \quad \begin{array}{ccccccc} \text{green circle} & \text{green circle} & \text{green circle} & \dots & \rightarrow & \text{red circle} \\ \text{green circle} & \text{green circle} & \text{green circle} & \dots & \rightarrow & \text{red circle} \\ \text{green circle} & \text{green circle} & \text{green circle} & \dots & \rightarrow & \text{red circle} \\ \text{green circle} & \text{green circle} & \text{green circle} & \dots & \rightarrow & \text{red circle} \\ \dots & \dots & \dots & \dots & & \dots \\ \text{green circle} & \text{green circle} & \text{green circle} & \dots & \rightarrow & \text{red circle} \end{array} \quad (2)$$

where $\text{yellow circle} = \text{WME} = \text{working memory element} = \text{fact} = \text{grounded logic formula} = \text{formula not containing variables}$.

Obviously, if the number of rules is large, it would be time-consuming to test each rule one by one to see if they apply.

It would be much more efficient if we could look at each yellow circle and immediately see which rule(s) may apply to it. This is the idea behind Rete.

In other words, we would like to **compile** the rule conditions green circle into a **decision tree**:

$$\begin{array}{ccccccc} \text{green circle} & \text{green circle} & \text{green circle} & \dots & & & \\ \text{green circle} & \text{green circle} & \text{green circle} & \dots & & & \\ \text{green circle} & \text{green circle} & \text{green circle} & \dots & & & \\ \dots & \dots & \dots & \dots & & & \\ \text{green circle} & \text{green circle} & \text{green circle} & \dots & & & \end{array} \xRightarrow{\text{Rete algorithm}} \text{decision tree} \quad (3)$$

The actions red circle of the rules do not figure in the decision process.

1.1 How the Rete graph is constructed

α -nodes:

$$\overbrace{\text{green circle}}^{\alpha_1} \wedge \overbrace{\text{green circle}}^{\alpha_2} \wedge \overbrace{\text{green circle}}^{\alpha_3} \dots \rightarrow \text{red circle} \tag{4}$$

β -nodes:

$$\overbrace{\overbrace{\overbrace{\text{green circle}}^{\beta_1} \wedge \text{green circle}}^{\beta_2} \wedge \text{green circle}}^{\beta_3} \dots \rightarrow \text{red circle} \tag{5}$$

2 Relation to deep learning

A typical **neural network** is:

weights matrix

total # layers

$$F(\vec{x}) = \textcircled{\textcolor{red}{\cap}}(W_1 \textcircled{\textcolor{red}{\cap}}(W_2 \dots \textcircled{\textcolor{red}{\cap}}(W_L \vec{x})))$$

(6)

Its set of **parameters** is $\Theta = \{W_{i,j}^\ell\} \in \mathbb{R}^m$, where $m = \#$ weights.