The core component of an RNN is the hidden layer, which is responsible for maintaining and updating the internal state. The hidden state at time step *t* is denoted as h*t* and is computed as follows:

**ht = f( W*hx*x*t* + W*hh*h*t-1* + b*h* )**

* **ht**: Hidden state at time step ***t***.
* **xt** : Input at time step ***t***.
* **W*hx***: Weight matrix for the input.
* **W*hx***: Weight matrix for the hidden state.
* **b*h***: Bias term.
* **f** : Activation function (usually tanh or ReLU).

**Output Computation**

The output **yt** at each time step is usually computed as:

**yt = g( Whyht + by )**

* **Why :** Weight matrix for the hidden state to output.
* **by** : Bias term
* **g :** Activation function (often softmax for classification tasks).