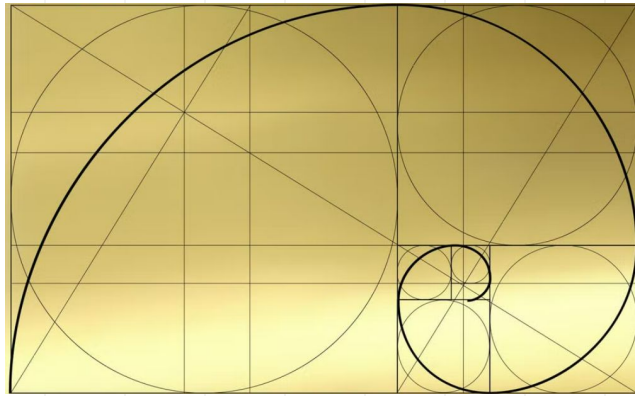
Recursion

Question

What is a recursive equation in Math?

What is Recursion?

- Recursion: The calling of a function within the function until a specific goal is reached.
- Recursion is used in Object-Oriented Programming Languages to allow the coder to write more complex code



When is Recursion used?

- MergeSorts and QuickSort
- Binary Search
- Algorithms that require a loop that decreases in repetitions every completion

What it looks like (Pseudocode)

```
Method a {
Method b(parameters)

}
Method b (parameters){
If (base case)
Return number
Else
Return method b(parameters)
}
```

When to and when not use recursion

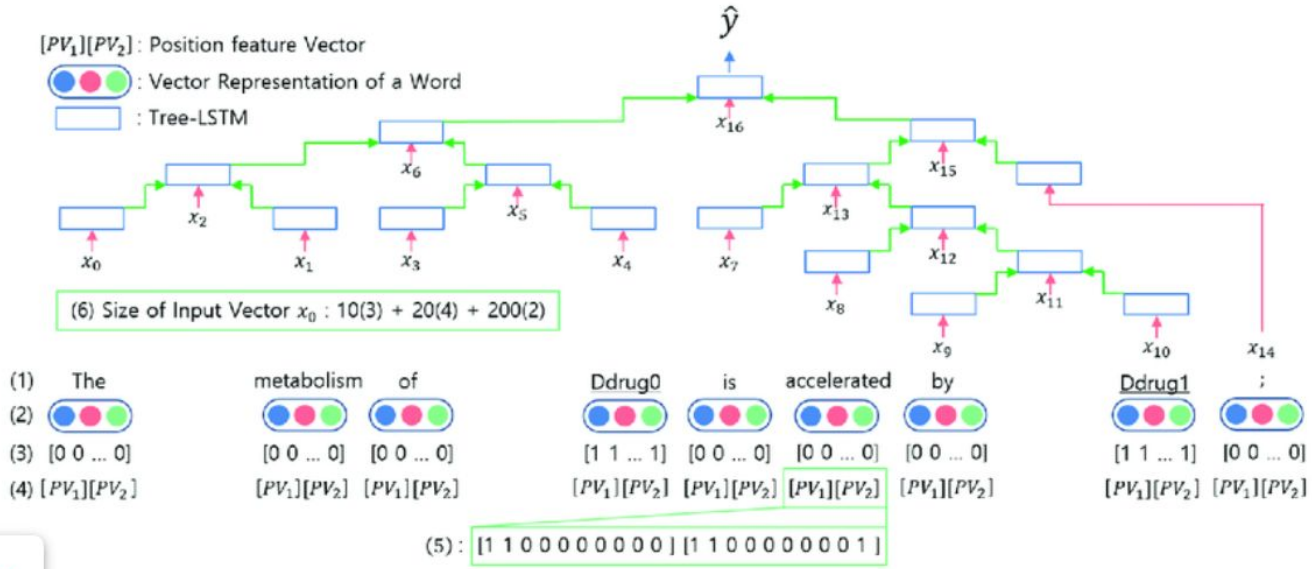
- Use when dealing with large sets of data, where the approach must be simple and it can use as much memory as necessary
- Don’t use when dealing with small sets of data or sets that require more logic and memory.

Neural Networks

How does Recursion fit into Machine Learning?

- Deep NN created by applying the same set of weights recursively over a structured input,
- Produces a structured prediction over variable-size input structures
  - Topological order, deep tree structure.
  - Requiring complete sentence parsing
  - Predicting Stock Markets
  - Generating Pokemon Names

Recursive Neural Network

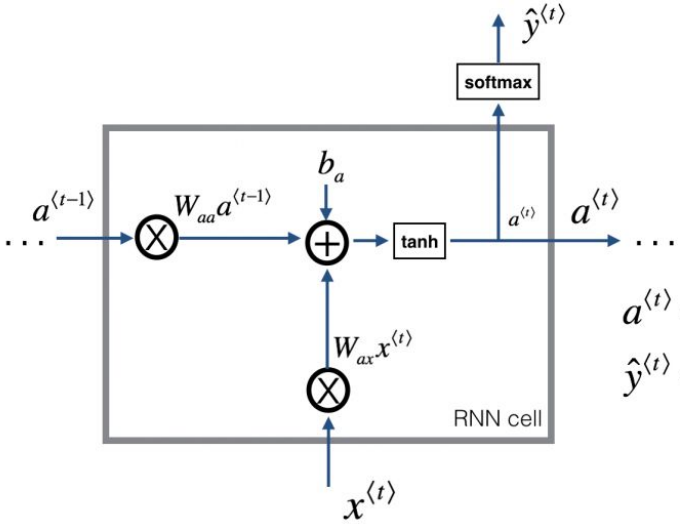


Example

Model

- RNN Name Generator
  - To simulate real people
  - Given n inputs, iterates for each one to compute n outputs
- n inputs are letters, model tries to predict next letter
- Main Methods
  - Forward Prop
  - Loss Calc
  - Back Prop
  - Clip Gradients

Forward Prop



$$a^{\langle t \rangle} = \tanh(W_{ax}x^{\langle t \rangle} + W_{aa}a^{\langle t-1 \rangle} + b_a)$$

$$\hat{y}^{\langle t \rangle} = soft\max(W_{ya}a^{\langle t \rangle} + b_y)$$

- For each Epoch, iterate with formulas for each letter in word
- ‘a’ = hidden state

Loss Calc

$$l_t(y_{pred,t},\ y_t) = y_t * \log(y_{pred,t}) + (y_t - 1) * \log(1 - y_{pred,t})$$

$$\text{Loss} = L(y_{pred},\ y) = \Sigma(l_t(y_{pred,t},\ y_t))$$

Back Prop

loss ~~fn~~ deriv

$$= \frac{\partial J}{\partial y} = y_{pred} - y$$

(hidden)

$$= \frac{\partial J}{\partial W_{y,a}} = \frac{\partial y}{\partial W_{y,a}} \cdot \frac{\partial J}{\partial y} = a \cdot \frac{\partial J}{\partial y}$$

weight of hid

bias of y_{pred}

and

$$\frac{\partial J}{\partial b_y} = \frac{\partial J}{\partial y}$$

$$\partial \text{ of hidd} = \frac{\partial J}{\partial a} = \frac{\partial y}{\partial a} \cdot \frac{\partial J}{\partial y} = W_{y,a} \cdot \frac{\partial J}{\partial y} + \underbrace{(\text{const}}_{\text{set}=} \left(\frac{\partial J}{\partial a_{t+1}}\right)$$

$$\frac{\partial J}{\partial \tanh(x,a)} = (1-a^2) \cdot \frac{\partial J}{\partial a}$$

and grad + bias

$$= \frac{\partial J}{\partial W_{i,x,a}} = X \cdot \frac{\partial J}{\partial \tanh(x,a)}$$

use stochastic grad descen

$$\frac{\partial J}{\partial W_{a,a}} = a_{t-1} \cdot \frac{\partial J}{\partial \tanh(x,a)}$$