Computation Graph

y3162

Namespace:CG

Class Inheritance Hierarchy

- Node
  - Leaf
  - MMtoM
    - Add
    - Sub
  - MMto1
    - Dots
    - MLE
  - MtoM
    - ReLU
    - Softmax
  - Mto1
    - Norm2
  - Affine

Class:Node

This is a base class of following classes.

Type	Property	Description
vec1<dtype>	data	A vector processed by this node.
vec1<dtype>	grad	The partial derivative of the cost function with respect to *data*.
vec1<Node*>	forward	A vector of pointers to the next layer's nodes.
vec1<Node*>	backward	A vector of pointers to the previous layer's nodes.
size_t	domsize	The dimension of the domain.
int	f_count	The number of nodes in the next layer during backpropagation.
int	b_count	The number of nodes in the previous layer during forwardpropagation.

Type	Method	Description
void	pushThis(Node *node)	Add this as a new argument's new next layer.
virtual void	calcData()	Calculate *data* by using previous layer's nodes.
void	forwardPropagation()	After all forward propagations from the previous layer are completed, propagate to the next layer.
virtual void	calcPartialDerivative()	Calculate the *grad* for the previous layer by using the *grad* at this node.
void	backwardPropagation()	After receiving all backward propagations from the next layer, propagate backward to the previous layer.
virtual void	updateParameters(dtype eta)	Update this node's parameters.
void	update(dtype eta)	After updating all parameters of the next layer, update the parameters of the previous layer.

Class:Leaf extends Node

This class represents a leaf node.

Type	Method	Description
void	getInput(vec1<dtype> input)	Provide the argument as input to the *data*.

Class:MMtoM extends Node

This class represents a node that processes the *data* from the previous layer's nodes and updates the *data* for this node. $\mathbb{R}^m \times \mathbb{R}^m \longrightarrow \mathbb{R}^m$.

Class:Add extends MMtoM

For $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{d} \in \mathbb{R}^m$, $\boldsymbol{x}$ is the first and $\boldsymbol{y}$ is the second node in the previous layer and $\boldsymbol{d}$ is *data* in this node.

$$\boldsymbol{d} = \boldsymbol{x} + \boldsymbol{y} \Longrightarrow d_i = x_i + y_i$$

Let L be the cost function,

$$\frac{\partial L}{\partial x_i} = \frac{\partial d_i}{\partial x_i}\frac{\partial L}{\partial d_i} = \frac{\partial L}{\partial d_i}$$
$$\frac{\partial L}{\partial y_i} = \frac{\partial d_i}{\partial y_i}\frac{\partial L}{\partial d_i} = \frac{\partial L}{\partial d_i}$$

Class:Sub extends MMtoM

For $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{d} \in \mathbb{R}^m$, $\boldsymbol{x}$ is the first and $\boldsymbol{y}$ is the second node in the previous layer and $\boldsymbol{d}$ is *data* in this node.

$$\boldsymbol{d} = \boldsymbol{x} - \boldsymbol{y} \Longrightarrow d_i = x_i - y_i$$

Let L be the cost function,

$$\frac{\partial L}{\partial x_i} = \frac{\partial d_i}{\partial x_i}\frac{\partial L}{\partial d_i} = \frac{\partial L}{\partial d_i}$$
$$\frac{\partial L}{\partial y_i} = \frac{\partial d_i}{\partial y_i}\frac{\partial L}{\partial d_i} = -\frac{\partial L}{\partial d_i}$$

Class:MMto1 extends Node

This class represents a node that processes the *data* from the previous layer's nodes and updates the *data* for this node. $\mathbb{R}^m \times \mathbb{R}^m \longrightarrow \mathbb{R}$.

Class:Dots extends MMto1

For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m$ and $d \in \mathbb{R}$, $\boldsymbol{x}$ is the first and $\boldsymbol{y}$ is the second node in the previous layer and d is *data* in this node.

$$d = \boldsymbol{x} \cdot \boldsymbol{y} = \sum_{i=1}^{m} x_i y_i$$

Let L be the cost function,

$$\frac{\partial L}{\partial x_i} = \frac{\partial d}{\partial x_i}\frac{\partial L}{\partial d} = y_i\frac{\partial L}{\partial d}$$
$$\frac{\partial L}{\partial y_i} = \frac{\partial d}{\partial y_i}\frac{\partial L}{\partial d} = x_i\frac{\partial L}{\partial d}$$

Class:MLE extends MMto1

For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m$ and $d \in \mathbb{R}$, $\boldsymbol{x}$ is the first and $\boldsymbol{y}$ is the second node in the previous layer and d is *data* in this node.

$$d = \frac{1}{m}\left\|\boldsymbol{x} - \boldsymbol{y}\right\|_2^2 = \frac{1}{m}\sum_{i=1}^{m}\left(x_i - y_i\right)^2$$

Let L be the cost function,

$$\frac{\partial L}{\partial x_i} = \frac{\partial d}{\partial x_i}\frac{\partial L}{\partial d} = \frac{2\left(x_i - y_i\right)}{m}\frac{\partial L}{\partial d}$$
$$\frac{\partial L}{\partial y_i} = \frac{\partial d}{\partial y_i}\frac{\partial L}{\partial d} = -\frac{2\left(x_i - y_i\right)}{m}\frac{\partial L}{\partial d}$$

Class:MtoM extends Node

This class represents a node that processes the *data* from the previous layer's node and updates the *data* for this node. $\mathbb{R}^m \longrightarrow \mathbb{R}^m$.

Class:ReLU extends MtoM

For $\boldsymbol{x}, \boldsymbol{d} \in \mathbb{R}^m$, $\boldsymbol{x}$ is the first node in the previous layer and $\boldsymbol{d}$ is *data* in this node.

$$\boldsymbol{d} = \mathrm{ReLU}\left(\boldsymbol{x}\right) \Longrightarrow d_i = \mathrm{ReLU}\left(x_i\right) = \begin{cases} x_i & x_i \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Let L be the cost function,

$$\frac{\partial L}{\partial x_i} = \frac{\partial d_i}{\partial x_i}\frac{\partial L}{\partial d_i} = 1_{[x_i \geq 0]}\frac{\partial L}{\partial d_i}$$

Class:Softmax extends MtoM

For $\boldsymbol{x}, \boldsymbol{d} \in \mathbb{R}^m$, $\boldsymbol{x}$ is the first node in the previous layer and $\boldsymbol{d}$ is *data* in this node.

$$\boldsymbol{d} = \mathrm{Softmax}\left(\boldsymbol{x}\right) \Longrightarrow d_i = \mathrm{Softmax}\left(\boldsymbol{x}\right)_i = \frac{\exp\left(x_i\right)}{\sum_{j=1}^{m}\exp\left(x_j\right)}$$

Let L be the cost function,

$$\frac{\partial L}{\partial x_i} = \sum_{j=1}^{m}\frac{\partial d_j}{\partial x_i}\frac{\partial L}{\partial d_j} = \sum_{j=1}^{m}\left(\delta_{i,j} - d_i\right) d_j \frac{\partial L}{\partial d_j}$$

Class:Mto1 extends Node

This class represents a node that processes the *data* from the previous layer's node and updates the *data* for this node. $\mathbb{R}^m \longrightarrow \mathbb{R}$.

Class:Norm2 extends Mto1

For $\boldsymbol{x} \in \mathbb{R}^m$ and $d \in \mathbb{R}$, $\boldsymbol{x}$ is the first node in the previous layer and d is *data* in this node.

$$d = \left\|\boldsymbol{x}\right\|_2 = \sqrt{\sum_{i=1}^{m} x_i^2}$$

Let L be the cost function,

$$\frac{\partial L}{\partial x_i} = \frac{\partial d}{\partial x_i}\frac{\partial L}{\partial d} = \frac{x_i}{d}\frac{\partial L}{\partial d}$$

Class:Affine extends Node

This class represents a node that performs an affine transformation.

Type	Property	Description
vec2<dtype>	Weight	The affine transformation matrix for this affine transformation.
vec2<dtype>	gradWeight	The sum of the gradients of the parameters.
dtype	bias	The bias in this affine transformation.

For $\boldsymbol{x} \in \mathbb{R}^m, b \in \mathbb{R}, \boldsymbol{W} \in \mathbb{R}^{(m+1)\times n}$ and $\boldsymbol{d} \in \mathbb{R}^n$, $\boldsymbol{x}$ is the first node in the previous layer, b is the bias and $\boldsymbol{d}$ is the *data* in this node.$\mathbb{R}^m \longrightarrow \mathbb{R}^n$

$$\boldsymbol{d} = \boldsymbol{W}^T \begin{pmatrix} \boldsymbol{x} \\ b \end{pmatrix} \Longrightarrow d_i = \sum_{j=1}^{m} W_{j,i} x_j + W_{m+1,i} b$$

Let L be the cost function,

$$\frac{\partial L}{\partial x_i} = \sum_{j=1}^{n} \frac{\partial d_j}{\partial x_i} \frac{\partial L}{\partial d_j} = \sum_{j=1}^{n} W_{i,j} \frac{\partial L}{\partial d_j}$$

$$\frac{\partial L}{\partial W_{i,j}} = \sum_{k=1}^{n} \frac{\partial d_k}{\partial W_{i,j}} \frac{\partial L}{\partial d_k} = \begin{cases} x_i \frac{\partial L}{\partial d_j} & 1 \le i \le m \\ b \frac{\partial L}{\partial d_j} & \text{otherwise} \end{cases}$$