...Machine Learning...

Artur Ezequiel Nelson

Universidade do Minho

26 de Abril

Indice

titulo

titulo

Fork e Join

- $(\triangle)$ :: Cartesian k $\Rightarrow$ (a 'k' c) $\rightarrow$ (a 'k' d) $\rightarrow$ (a 'k' (c $\times$ d))
- $(\triangledown)$ :: Cartesian k $\Rightarrow$ (c 'k' a) $\rightarrow$ (d 'k' a) $\rightarrow$ ((c $\times$ d) 'k' a)

instancia de $\rightarrow^+$

newtype a $\rightarrow^+$ b = AddFun (a $\rightarrow$ b)

instance Category ($\rightarrow^+$) **where**
    type Obj ($\rightarrow^+$) = Additive
    id = AddFun id
    AddFun g $\circ$ AddFun f = AddFun (g $\circ$ f)

instance Monoidal ($\rightarrow^+$) **where**
    AddFun f $\times$ AddFun g = AddFun (f $\times$ g)

instance Cartesian ($\rightarrow^+$) **where**
    exl = AddFun exl
    exr = AddFun exr
    dup = AddFun dup

instancia de $\rightarrow^+$

instance Cocartesian ($\rightarrow^+$) **where**
    inl = AddFun inlF
    inr = AddFun inrF
    jam = AddFun jamF

inlF :: Additive b $\Rightarrow$ a $\rightarrow$ a $\times$ b
inrF :: Additive a $\Rightarrow$ b $\rightarrow$ a $\times$ b
jamF :: Additive a $\Rightarrow$ a $\times$ a $\rightarrow$ a

inlF = λa $\rightarrow$ (a, 0)
inrF = λb $\rightarrow$ (0, b)
jamF = λ(a, b) $\rightarrow$ a + b

definição de NumCat

```
class NumCat k a where
        negateC :: a `k` a
        addC :: (a × a) `k` a
        mulC :: (a × a) `k` a
        ...

instance Num a ⇒ NumCat (→) a where
        negateC = negate
        addC = uncurry (+)
        mulC = uncurry (·)
        ...
```

D (negate u) = negate (D u)
D (u + v) = D u + D v
D (u · v) = u · D v + v · D u

- Impreciso na natureza de u e v.
- Algo mais preciso seria defenir a diferenciação das operações em si.

```
class Scalable k a where
    scale :: a → (a `k` a)

instance Num a ⇒ Scalable (→⁺) a where
    scale a = AddFun (λda → a · da)

instance NumCat D where
    negateC = linearD negateC
    addC = linearD addC
    mulC = D (λ(a, b) → (a · b, scale b ▽ scale a))
```

Generalizing Automatic Differentiation

newtype D_k a b = D (a $\rightarrow$ b $\times$ (a ‘k‘ b))

linearD :: (a $\rightarrow$ b) $\rightarrow$ (a ‘k‘ b) $\rightarrow$ D_k a b
linearD f f’= D (λa $\rightarrow$ (f a, f’))

instance Category k $\Rightarrow$ Category D_k **where**
    type Obj D_k = Additive $\wedge$ Obj k ...

instance Monoidal k $\Rightarrow$ Monoidal D_k **where** ...

instance Cartesian k $\Rightarrow$ Cartesian D_k **where** ...

instance Cocartesian k $\Rightarrow$ Cocartesian D_k **where**
    inl = linearD inlF inl
    inr = linearD inrF inr
    jam = linearD jamF jam

instance Scalable k s $\Rightarrow$ NumCat D_k s **where**
negateC = linearD negateC negateC
addC = linearD addC addC
mulC = D (λ(a, b) $\rightarrow$ (a $\cdot$ b, scale b $\bigtriangledown$ scale a))

Exemplos

Generalizar