

Simple essence of AD

Artur Ezequiel Nelson

Universidade do Minho

29 de Maio

Index

Relembrar

Conceitos base

Categorias e funtores

Dedução de instâncias

RAD e FAD generalizados

Demonstração prática

Relembrar

Documento estudado: "The simple essence of automatic differentiation"([Elliott 2018][2])

Objetivo: Estudo e implementação de um algoritmo AD genérico

Definição de \mathcal{D} e conversão em \mathcal{D}^+

\mathcal{D} - aproximação linear de uma função

Definição

Seja $f :: a \rightarrow b$ uma função onde a e b são espaços vetoriais sobre um corpo comum. A primeira definição de derivada é:

$$\mathcal{D} :: (a \rightarrow b) \rightarrow (a \rightarrow (a \multimap b))$$

\mathcal{D}^+ - versão de \mathcal{D} com composição eficiente

$$\mathcal{D}^+ :: (a \rightarrow b) \rightarrow (a \rightarrow (b \times (a \multimap b)))$$

$$\mathcal{D}^+ f a = (f a, \mathcal{D} f a)$$

Corolários associados a \mathcal{D}^+

Corolário 1.1

$$\mathcal{D}^+ (g \circ f) a = \mathbf{let} \{ (b, f') = \mathcal{D}^+ f a; (c, g') = \mathcal{D}^+ g b \} \\ \mathbf{in} (c, g' \circ f')$$

Corolário 2.1

$$\mathcal{D}^+ (f \times g) (a, b) = \mathbf{let} \{ (c, f') = \mathcal{D}^+ f a; (d, g') = \mathcal{D}^+ g b \} \\ \mathbf{in} ((c, d), f' \times g')$$

Corolário 3.1

Para todas as funções lineares f , $\mathcal{D}^+ f = \lambda a \rightarrow (fa, f)$.

Objetivo

Criar uma implementação de \mathcal{D}^+ através da transcrição dos seus corolários para teoria de categorias de modo a obter um algoritmo generalizado para AD.

Categorias e funtores

Categoria: conjunto de objetos e morfismos com duas operações base(id e composição) e 2 regras:

- $id \circ f = f \circ id = f$
- $f \circ (g \circ h) = (f \circ g) \circ h$

Funtor: mapeia uma categoria noutra, preservando a estrutura

- Dado um objeto $t \in \mathcal{U}$ existe um objeto correspondente $F t \in \mathcal{V}$
- Dado um morfismo $m :: a \rightarrow b \in \mathcal{U}$ existe um morfismo correspondente $F m :: F a \rightarrow F b \in \mathcal{V}$
- $F id (\in \mathcal{U}) = id (\in \mathcal{V})$
- $F (f \circ g) = F f \circ F g$

Passos para obter a instância a partir da definição do funtor

- Passo 1 - Assumir que $\hat{\mathcal{D}}$ é funtor de uma instância de \mathcal{D} a determinar
- Passo 2 - Substituir pelo que determinamos nos corolários
- Passo 3 - Generalizar condições se necessário para obtermos instância

Exemplo para categorias

Passo 1

$$id = \hat{\mathcal{D}} id = \mathcal{D} (\mathcal{D}^+ id)$$

$$\hat{\mathcal{D}} g \circ \hat{\mathcal{D}} f = \hat{\mathcal{D}} (g \circ f) = \mathcal{D} (\hat{\mathcal{D}} (g \circ f))$$

Passo 2

$$id = \mathcal{D} (\lambda a \rightarrow (id\ a, id))$$

$$\hat{\mathcal{D}} g \circ \hat{\mathcal{D}} f = \mathcal{D} (\lambda a \rightarrow \mathbf{let}\ \{(b, f') = \mathcal{D}^+ f\ a; (c, g') = \mathcal{D}^+ g\ b\} \mathbf{in}\ (c, g' \circ f'))$$

Passo 3

Para a nossa instância a primeira equação que determinamos serve como definição da identidade.

Para definir a composição generalizamos a condição:

$$\mathcal{D} g \circ \mathcal{D} f = \mathcal{D} (\lambda a \rightarrow \mathbf{let}\ \{(b, f') = f\ a; (c, g') = g\ b\} \mathbf{in}\ (c, g' \circ f'))$$

Exemplo para categorias

Passo 1

$$id = \hat{\mathcal{D}} id = \mathcal{D} (\mathcal{D}^+ id)$$

$$\hat{\mathcal{D}} g \circ \hat{\mathcal{D}} f = \hat{\mathcal{D}} (g \circ f) = \mathcal{D} (\hat{\mathcal{D}} (g \circ f))$$

Passo 2

$$id = \mathcal{D} (\lambda a \rightarrow (id\ a, id))$$

$$\hat{\mathcal{D}} g \circ \hat{\mathcal{D}} f = \mathcal{D} (\lambda a \rightarrow \mathbf{let}\ \{(b, f') = \mathcal{D}^+ f\ a; (c, g') = \mathcal{D}^+ g\ b\} \mathbf{in}\ (c, g' \circ f'))$$

Passo 3

Para a nossa instância a primeira equação que determinamos serve como definição da identidade.

Para definir a composição generalizamos a condição:

$$\mathcal{D} g \circ \mathcal{D} f = \mathcal{D} (\lambda a \rightarrow \mathbf{let}\ \{(b, f') = f\ a; (c, g') = g\ b\} \mathbf{in}\ (c, g' \circ f'))$$

Exemplo para categorias

Passo 1

$$id = \hat{\mathcal{D}} \, id = \mathcal{D} \, (\mathcal{D}^+ \, id)$$

$$\hat{\mathcal{D}} g \circ \hat{\mathcal{D}} f = \hat{\mathcal{D}} (g \circ f) = \mathcal{D} (\hat{\mathcal{D}} (g \circ f))$$

Passo 2

$$id = \mathcal{D} (\lambda a \rightarrow (id\ a, id))$$

$$\hat{\mathcal{D}} g \circ \hat{\mathcal{D}} f = \mathcal{D} (\lambda a \rightarrow \mathbf{let} \{ (b, f') = \mathcal{D}^+ f a; (c, g') = \mathcal{D}^+ g b \} \mathbf{in} (c, g' \circ f'))$$

Passo 3

Para a nossa instância a primeira equação que determinamos serve como definição da identidade.

Para definir a composição generalizamos a condição:

$$\mathcal{D} g \circ \mathcal{D} f = \mathcal{D} (\lambda a \rightarrow \mathbf{let} \{ (b, f') = f \ a; (c, g') = g \ b \} \mathbf{in} (c, g' \circ f'))$$

Definição da classe de categoria

class *Category* *k* **where**

id :: (*a*'*k*'*a*)

(*○*) :: (*b*'*k*'*c*) → (*a*'*k*'*b*) → (*a*'*k*'*c*)

Instância deduzida para a categoria

instance *Category* *D* **where**

id = *linearD id*

D g ○ *D f* =

D (λ*a* → **let** { (*b*, *f'*) = *f a*; (*c*, *g'*) = *g b* } **in** (*c*, *g'* ○ *f'*))

Definição da classe de categoria monoidal

class *Category* $k \Rightarrow \text{Monoidal } k$ **where**

$$(\times) :: (a' \text{ k}' c) \rightarrow (b' \text{ k}' d) \rightarrow ((a \times b)' \text{ k}' (c \times d))$$

Instância deduzida para a categoria monoidal

instance *Monoidal* \mathcal{D} where
$$\mathcal{D} f \times \mathcal{D} g = \mathcal{D} (\lambda(a, b) \rightarrow \mathbf{let} \{ (c, f') = f \ a; (d, g') = g \ b \} \\ \mathbf{in} ((c, d), f' \times g'))$$

Definição da classe de categoria cartesiana

```

class Monoidal  $k \Rightarrow \textit{Cartesian}$   $k$  where
     $exl :: (a, b) \rightarrow k \rightarrow a$ 
     $exr :: (a, b) \rightarrow k \rightarrow b$ 
     $dup :: a \rightarrow k \rightarrow (a, a)$ 
    
```

Instância deduzida para a categoria cartesiana

```

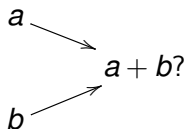
instance Cartesian  $D$  where
     $exl = \textit{linearD} \textit{exl}$ 
     $exr = \textit{linearD} \textit{exr}$ 
     $dup = \textit{linearD} \textit{dup}$ 
    
```


Definição da classe de categoria cocartesiana

```

class Category  $k \Rightarrow \text{Cocartesian } k$  where
     $\text{inl} :: a \text{ ' } k \text{ ' } (a, b)$ 
     $\text{inr} :: b \text{ ' } k \text{ ' } (a, b)$ 
     $\text{jam} :: (a, a) \text{ ' } k \text{ ' } a$ 
    
```

Dedução de (\rightarrow^+)



- Queremos uma categoria que os seus objetos tenham adição e seja capaz de multiplicar por constantes (scaling).

Instância deduzida para AD genérico

Dedução de D_k

- Queremos ter um D_k tal que seja o \mathcal{D} mas com uma categoria de genérica, pois nunca fizemos nada de específico com a \multimap .
- Queremos definir também as operações usuais nesta categoria, como a soma, multiplicação, sin, cos, etc.

Generalizando RAD e FAD

Obter RAD e FAD de algoritmo AD genérico: forçar a direção da composição de morfismos

Conversão da escrita de morfismos

$f :: a' \text{ k}' b \Rightarrow (\circ f) :: (b' \text{ k}' r) \rightarrow (a' \text{ k}' r)$ para r objeto de categoria k .

Definição de novo tipo

newtype $Cont_k^f a b = Cont ((b' \text{ k}' r) \rightarrow (a' \text{ k}' r))$

Functor derivado dele

$cont :: Category \text{ k} \Rightarrow (a' \text{ k}' b) \rightarrow Cont_k^f a b$
 $cont f = Cont (\circ f)$

Demonstração prática

Demonstração prática

Link do projeto git: [LEI 2019] [1]



Projeto LEI.

https://github.com/Ezequiel-Moreira/LEI_2019_Uminho.

Accessed: 2019-05-28.



ELLIOTT, C.

The simple essence of automatic differentiation.

Proc. ACM Program. Lang. 2, ICFP (July 2018), 70:1–70:29.