Julia

Высокопроизводительный язык для очень больших данных.

СИМОНОВ АЛЕКСАНДР

КОРОЛЁВ КОНСТАНТИН

Отличительные особенности

- Множественная диспетчеризация. Множественная диспетчеризация означает, что вариант определенной функции, который будет исполнен, определяется всем набором типов параметров этой функции.JIT-компиляция. Это технология, которая позволяет увеличить производительность путём компиляции байт-кода в машинный.
- Метапрограммированиею. Это когда мы в программе создаем программы и на ходу их запускаем. Это мощный метод, который позволяет сделать много разных интересных вещей. Классический пример — Django ORM, в котором создаются поля с помощью метаклассов.
- Параллелизация. Позволяет производить несколько вычислений параллельно друг другу, что в разы сокращает время выполнения программы.

Технологии машинного обучения

- Flux

  В данном языке он обычно используется как уровень абстракции для создания нейронных сетей. Flux.jl входит в число моих любимых пакетов, и он был одним из первых, с которыми я познакомился, начиная работу с Julia. Его огромным преимуществом является не вероятно маленький размер.

  Легковесность Flux позволяет прекрасно использовать его на серверах, так как он не займет много места, и его запросто можно поместить в крошечные виртуальные среды при помощи Pkg.

Технологии машинного обучения

- Merlin.jl

  Merlin — еще один фреймворк для глубокого обучения, предназначенный для создания нейронных сетей. При выполнении многих операций он опережает Flux, но это совсем не означает, что он всегда будет быстрее.

  Как и в случае с Flux, Merlin имеет встроенную GPU поддержку с CUDA. Модели Merlin обычно отличаются от моделей Flux способностью к более быстрому развертыванию. Если вы планируете развернуть API, использующее нейронную сеть, при помощи Genie и Julia, стоить выбрать Merlin. Помимо всех выше перечисленных превосходных достоинств, он также включает маленькую дистрибутивную библиотеку.

Технологии машинного обучения

- KNet.jl

  В отличие от Flux и Merlin, пакет Knet немного тяжелее. Кроме этого, он написан не чисто на Julia, но и на других языках, таких как C и MATLAB, выполняющихся под кодом Julia.

  Следует отметить, что только очень малая часть Knet, а именно 1,8%, написана на C и MATLAB. Остается добавить, что Knet лучше подойдет новичкам, поскольку очень прост в использовании. Knet был создан в университете Коч (Стамбул). Вследствие чего, данный пакет, похоже, намного лучше поддерживается, чем большинство его аналогов

Knet.jl

Технологии машинного обучения

- Lathe

  Lathe сопровождается довольно большой библиотекой статистики, включающей хи-квадрат тесты, байесовскую статистику, t-тесты, f-тесты и даже малоизвестные тесты по критерию знаков. Более того, в него также включена проверка с модулем статистики, в котором есть метрики точности для непрерывных и категориальных прогнозов. И наконец, Lathe.jl включает небольшую скромную библиотеку дистрибутивов.

  Данный пакет также написан на чистом Julia и имеет довольно основательную документацию. Можно сказать, что если Flux — это нечто иное, как TensorFlow для Julia, то Lathe — это ее Sklearn. А значит, он весьма содержателен и предоставляет все необходимые инструменты для работы. Более того, благодаря простоте в использовании, Lathe будет легко освоить даже начинающим.

Волновое уравнение гиперболического типа

Волновое уравнение гиперболического типа имеет следующий вид:

$$\frac{\partial^2 U(x,t)}{\partial t^2} = c^2 \frac{\partial^2 U(x,t)}{\partial x^2}$$

Описывает одномерные линейные волны без дисперсии. Например, колебания струны, звук в жидкости (газе) или электромагнитные волны в вакууме (в последнем случае уравнение должно быть записано в векторном виде).

Волновое уравнение гиперболического типа

Простейшей разностной схемой, аппроксимирующей данное уравнение, является явная пятиточечная схема :

$$\frac{U_i^{n+1} - 2U_i^n + U_i^{n-1}}{\tau^2} = c^2 \frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{h^2}$$

$$x_i = ih,\ t_n = \tau$$

Эта схема, получившая название «крест», имеет второй порядок точности по времени и по пространственной координате и является трехслойной по времени.

Волновое уравнение гиперболического типа

Пример программной части решения волнового уравнения гиперболического типа:

```
# функция задающая начальное условие
ψ = x -> x^2 * exp( -(x-0.5)^2/0.01 )
# поведение на границах
ϕ(x) = 0
c = x -> 1
```

Волновое уравнение гиперболического типа

Пример программной части решения волнового уравнения гиперболического типа:

```
# решение одномерного волнового уравнения
function pdesolver(N = 100, K = 100, L = 2pi, T = 10, a = 0.1 )

    dx = L/N;
    dt = T/K;
    gam(x) = c(x)*c(x)*a*a*dt*dt/dx/dx;
    print("Kurant-Fridrihs-Levi: $(dt*a/dx) dx = $dx dt = $dt")
    u = zeros(N,K);
```

Волновое уравнение гиперболического типа

Пример программной части решения волнового уравнения гиперболического типа:

```
x = [i for i in range(0, length = N, step = dx)]
# инициализируем первые два временных слоя
u[:,1] = ψ.(x);
u[:,2] = u[:,1] + dt*ψ.(x);
# задаём поведение на границах

fill!( u[1,:], 0);
fill!( u[N,:], ϕ(L) );
```

Волновое уравнение гиперболического типа

Пример программной части решения волнового уравнения гиперболического типа:

```
    for t = 2:K-1, i = 2:N-1
        u[i,t+1] = -u[i,t-1] + gam( x[i] )* (u[i-1,t] + u[i+1,t]) + (2-2*gam
( x[i] ) )*u[i,t];
    end
    x, u
end
```

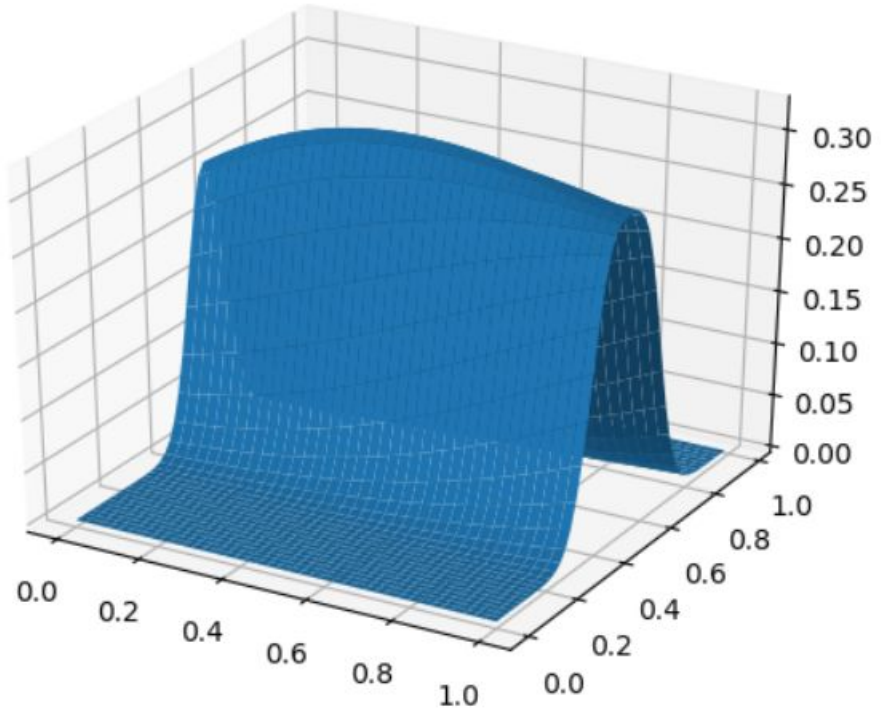
Волновое уравнение гиперболического типа

Пример программной части решения волнового уравнения гиперболического типа:

```
N = 50; # количество шагов по координате
K = 40; # и по времени
a = 0.1; # скорость распространения волны
L = 1; # длина образца
T = 1; # длительность эксперимента

t = [i for i in range(0, length = K, stop = T)]

X, U = pdesolver(N, K, L, T, a) # вызываем расчетную функцию

plot(X, U[:,1])
plot!(X, U[:,40])
```

Волновое уравнение гиперболического типа

Чтобы построить поверхность, воспользуемся PyPlot'ом не как окружением Plots, а непосредственно:




```
using PyPlot
surf(t, X, U)
```

Статьи с arxiv.org

- https://arxiv.org/abs/2103.00915
- https://arxiv.org/abs/2010.07516
- https://arxiv.org/abs/1209.5145
- https://arxiv.org/abs/2103.09948

Обучение

- https://julialang.org/learning/
- https://dmkpress.com/files/download/978-5-97060-370-3_1-27.pdf
- https://mnmc.hse.ru/mirror/pubs/share/415043701.pdf
- https://geekbrains.ru/posts/julia_lang