Developing an automatic differentiation and parameter optimisation pipeline for the particle shower model

Oleksii Kiva, Kyiv Academic University

Mentors: Dr. Lukas Heinrich and Dr. Michael Kagan

Stochastic program $f\colon \mathcal{X} \times \Omega \times \Theta \rightarrow \mathcal{Y}$

$\mathcal{X}$ is the input and $\mathcal{Y}$ is the program's output space

$\Omega \ni \omega$ is a random element drawn from a distribution $P(\ .\ ;\ \theta)$

samples of uniform (pseudo) random numbers in [0, 1] + inversion of CDF (in practice)

Θ is the set of parameters on which f depends either explicitly

or implicitly via the probability distribution on Ω

Basic examples

```
function f_0(θ)
    return rand(Bernoulli(θ))
end


function f_1(θ)
    p = θ^2 / (1 + θ^2)
    return rand(Bernoulli(p))
end
```

```
function f_2(x, θ)
    a = θ^2
    m = x * 3 + 11
    return rand(Normal(m, a))
end


function f_3(x, θ)
    a = θ^2
    b = rand(Binomial(10, θ))
    c = 2 * b + 3 * rand(Bernoulli(θ))
    return x * a * c * rand(Normal(b, a))
end
```

What unbiased estimate to put inside the expectation?

$$\frac{\partial}{\partial\theta}\mathbb{E}_\theta\left[f\left(x,\theta\right)\right]=\mathbb{E}\left[\ ?\ \right]$$

everything depends on smoothness (differentiability properties)

of $f\left(x,\theta\right)$ and $P(\ .\ ;\ \theta)$ with respect to θ

Gradient Estimation Using Stochastic Computation Graphs

John Schulman[1,2]
joschu@eecs.berkeley.edu

Nicolas Heess[1]
heess@google.com

Theophane Weber[1]
theophane@google.com

Pieter Abbeel[2]
pabbeel@eecs.berkeley.edu

- Finally θ might appear both in the probability distribution and inside the expectation, e.g., in $\frac{\partial}{\partial\theta}\mathbb{E}_{z\sim p(\cdot;\,\theta)}\left[f(x(z,\theta))\right]$. Then the gradient estimator has two terms:

$$\frac{\partial}{\partial\theta}\mathbb{E}_{z\sim p(\cdot;\,\theta)}\left[f(x(z,\theta))\right] = \mathbb{E}_{z\sim p(\cdot;\,\theta)}\left[\frac{\partial}{\partial\theta}f(x(z,\theta)) + \left(\frac{\partial}{\partial\theta}\log p(z;\theta)\right)f(x(z,\theta))\right]. \quad (4)$$

When $f(x, \theta)$ and $P(\ .\ ;\ \theta)$ are both smooth with respect to θ

Condition 1 (Differentiability Requirements). *Given input node $\theta \in \Theta$, for all edges (v, w) which satisfy $\theta \prec^D v$ and $\theta \prec^D w$, then the following condition holds: if w is deterministic, Jacobian $\frac{\partial w}{\partial v}$ exists, and if w is stochastic, then the derivative of the probability mass function $\frac{\partial}{\partial v} p(w \mid \text{PARENTS}_w)$ exists.*

Theorem 1. *Suppose that $\theta \in \Theta$ satisfies 1. Then the following two equivalent equations hold:*

$$\frac{\partial}{\partial \theta} \mathbb{E}\left[\sum_{c \in \mathcal{C}} c\right] = \mathbb{E}\left[\sum_{\substack{w \in \mathcal{S}, \\ \theta \prec^D w}} \left(\frac{\partial}{\partial \theta} \log p(w \mid \text{DEPS}_w)\right) \hat{Q}_w + \sum_{\substack{c \in \mathcal{C} \\ \theta \prec^D c}} \frac{\partial}{\partial \theta} c(\text{DEPS}_c)\right] \tag{5}$$

$$= \mathbb{E}\left[\sum_{c \in \mathcal{C}} \hat{c} \sum_{\substack{w \prec c, \\ \theta \prec^D w}} \frac{\partial}{\partial \theta} \log p(w \mid \text{DEPS}_w) + \sum_{\substack{c \in \mathcal{C}, \\ \theta \prec^D c}} \frac{\partial}{\partial \theta} c(\text{DEPS}_c)\right]. \tag{6}$$

Corollary 1. *Let* $L(\Theta, \mathcal{S}) := \sum_w \log p(w \mid \text{DEPS}_w)\hat{Q}_w + \sum_{c \in \mathcal{C}} c(\text{DEPS}_c)$*. Then differentiation of* L *gives us an unbiased gradient estimate:* $\frac{\partial}{\partial\theta}\mathbb{E}\left[\sum_{c\in\mathcal{C}} c\right] = \mathbb{E}\left[\frac{\partial}{\partial\theta}L(\Theta, \mathcal{S})\right]$.

One practical consequence of this result is that we can apply a standard automatic differentiation procedure to L to obtain an unbiased gradient estimator. In other words, we convert the stochastic computation graph into a deterministic computation graph, to which we can apply the backpropagation algorithm.

And what if the differentiability requirements are not satisfied?

What if the differentiability requirements are not satisfied?

```
function f_0(θ)
    return rand(Bernoulli(θ))
end


function f_1(θ)
    p = θ^2 / (1 + θ^2)
    return rand(Bernoulli(p))
end
```

```
function f_2(x, θ)
    a = θ^2
    m = x * 3 + 11
    return rand(Normal(m, a))
end


function f_3(x, θ)
    a = θ^2
    b = rand(Binomial(10, θ))
    c = 2 * b + 3 * rand(Bernoulli(θ))
    return x * a * c * rand(Normal(b, a))
end
```

More advanced examples

```julia
function p_interact_material(x, y, θ)
    par_radial = 10
    par_azimutal = 10
    r = sqrt(x^2 + y^2)

    alpha = atan2(x, y)

    sampling1 = 1 / (1 + exp(10*sin(par_radial*(alpha + 2*r))))

    sampling2 = 1 / (1 + exp(10*sin(par_azimutal*(r-2))))

    start = 1 / (1 + exp(-10*(r - θ)))

    tail = 1 / (1 + exp(10*(r - (θ + 10))))

    return 0.5 * start * sampling1 * sampling2 * tail
end
```

```julia
function simulate(T, θ)
    T_new = propagate(T)

    dointeract = rand(Binomial(p_interact_material(x, y, θ)))

    if dointeract
        push!(hits, (x, y))

        dosplit = rand(Binomial(p_split(T)))

        if dointeract
            T_1, T_2 = split(T_new)

            simulate(T_1, θ)
            simulate(T_2, θ)

        else
            simulate(T_new, θ)

        end
    end

end
```

It is still possible to obtain meaningful estimates

```
samples = [derivative_estimate(f_0, 0.6) for i in 1:1000]
println("d/dθ of E[f_0(θ)]: $(mean(samples)) ± $(std(samples) / sqrt(1000))")

samples = [derivative_estimate(f_1, 0.6) for i in 1:1000]
println("d/dθ of E[f_1(θ)]: $(mean(samples)) ± $(std(samples) / sqrt(1000))")

samples = [derivative_estimate(θ -> f_2(5, θ), -20) for i in 1:1000]
println("d/dθ of E[f_2(θ)]: $(mean(samples)) ± $(std(samples) / sqrt(1000))")

samples = [derivative_estimate(θ -> f_3(1, θ), 0.6) for i in 1:1000]
println("d/dθ of E[f_3(θ)]: $(mean(samples)) ± $(std(samples) / sqrt(1000))")
```

```
d/dθ of E[f_0(θ)]: 1.0025 ± 0.038765274363745904
d/dθ of E[f_1(θ)]: 0.6529411764705884 ± 0.012245094114506663
d/dθ of E[f_2(θ)]: 2.476388047193479 ± 1.2892228953295273
d/dθ of E[f_3(θ)]: 202.98243353789118 ± 1.2607083445925458
```

How?

$$\frac{\mathrm{d}\mathbb{E}\left[X(p)\right]}{\mathrm{d}p} = \mathbb{E}[\delta + w\left(Y - X(p)\right)].$$

Automatic Differentiation of Programs with Discrete Randomness

Gaurav Arya
Massachusetts Institute of Technology, USA
aryag@mit.edu

Moritz Schauer
Chalmers University of Technology, Sweden
University of Gothenburg, Sweden
smoritz@chalmers.se

Frank Schäfer
Massachusetts Institute of Technology, USA
University of Basel, Switzerland
franksch@mit.edu

Chris Rackauckas
Massachusetts Institute of Technology, USA
Julia Computing Inc., USA
Pumas-AI Inc., USA
crackauc@mit.edu

Definition 2.2 (Stochastic derivative)**.** Suppose $X(p) \in E$ is a stochastic program with index set I a closed real interval. We say that the triple of random variables (δ, w, Y), with $w \in \mathbb{R}$ and $Y \in E$, is a right (left) *stochastic derivative* of X at the input $p \in I$ if $\mathrm{d}X(\varepsilon)/\varepsilon \to \delta$ almost surely as $\varepsilon \to 0$, and there is an integrable (i.e. of bounded expectation) random variable $B > |\delta|$ such that for all bounded functions $f\colon E \to \mathbb{R}$ with bounded derivative it holds almost surely that

$$\mathbb{E}\left[w\left(f(Y) - f(X(p))\right) \mid X(p)\right] = \lim_{\varepsilon \to 0^{+/-}} \mathbb{E}\left[\frac{f(X(p+\varepsilon)) - f(X(p))}{\varepsilon} \mathbf{1}_{A_B(\varepsilon)} \,\middle|\, X(p)\right], \quad (2.4)$$

with limit taken from above (below), where $\mathbb{P}\left(A_B(\varepsilon) \mid X(p)\right)/\varepsilon$ is dominated by an integrable random variable for all $\varepsilon > 0$ ($\varepsilon < 0$).

```
struct StochasticTriple
    value # primal evaluation
    δ # "infinitesimal" component
    Δs # component of discrete change
       # with "infinitesimal"
       # probability
end
```

```
using Distributions
function X(p)  p = 0.6 + ε
    a = p^2  0.36 + 1.2ε
    b = rand(Binomial(10, p))
    6 + (1 with probability 10.0ε)
    c = 2 * b + 3 * rand(Bernoulli(p))
    12 + (3 with probability 12.5ε)
    return a * c * rand(Normal(b, a))
end
```

Thanks for attention!