Numerička provjera gradijenata

Marin Kačan

Sadržaj

Ispitivanje implementacije algoritma *backpropagation*

Ispitivanje implementacije *backpropagationa*

- Implementacija postupka analitičkog određivanja gradijenata u neuronskoj mreži nije uvijek jednostavna i lako je pogriješiti.

Ispitivanje implementacije *backpropagationa*

- Implementacija postupka analitičkog određivanja gradijenata u neuronskoj mreži nije uvijek jednostavna i lako je pogriješiti.
- Debuggiranje implementacije može biti mukotrpno.

Ispitivanje implementacije *backpropagationa*

- Implementacija postupka analitičkog određivanja gradijenata u neuronskoj mreži nije uvijek jednostavna i lako je pogriješiti.
- Debuggiranje implementacije može biti mukotrpno.
- Često se mreža može činiti ispravnom (funkcija gubitka pada usprkos pogrešnoj implementaciji).

Ispitivanje implementacije *backpropagationa*

- Implementacija postupka analitičkog određivanja gradijenata u neuronskoj mreži nije uvijek jednostavna i lako je pogriješiti.
- Debuggiranje implementacije može biti mukotrpno.
- Često se mreža može činiti ispravnom (funkcija gubitka pada usprkos pogrešnoj implementaciji).
- **Kako provjeriti ispravnost?**

Provjera ispravnosti

Opcije:

1. Izračunavanje prolaza unaprijed i unatrag ”na papiru”

Provjera ispravnosti

Opcije:

1. Izračunavanje prolaza unaprijed i unatrag ”na papiru”
    - Sporo

Provjera ispravnosti

Opcije:

1. Izračunavanje prolaza unaprijed i unatrag ”na papiru”
    - Sporo
    - Pretpostavlja da smo točno izveli izraze za gradijente

Provjera ispravnosti

Opcije:

1. Izračunavanje prolaza unaprijed i unatrag ”na papiru”
   - Sporo
   - Pretpostavlja da smo točno izveli izraze za gradijente
2. **Usporedba s numeričkom procjenom gradijenta**

Provjera ispravnosti

Opcije:

1. Izračunavanje prolaza unaprijed i unatrag ”na papiru”
   - Sporo
   - Pretpostavlja da smo točno izveli izraze za gradijente
2. **Usporedba s numeričkom procjenom gradijenta**
   - Pretpostavlja samo točno implementiran unaprijedni prolaz

Provjera ispravnosti

Opcije:

1. Izračunavanje prolaza unaprijed i unatrag ”na papiru”
   - Sporo
   - Pretpostavlja da smo točno izveli izraze za gradijente
2. **Usporedba s numeričkom procjenom gradijenta**
   - Pretpostavlja samo točno implementiran unaprijedni prolaz
   - Jednom implementiran, postupak se vrlo lako može koristiti neovisno o modelu

Provjera ispravnosti

Ideja postupka:

1. Nekakvim **postupkom** numerički aproksimirati vrijednost gradijenta

Provjera ispravnosti

Ideja postupka:

1. Nekakvim **postupkom** numerički aproksimirati vrijednost gradijenta
2. Nekakvom **mjerom pogreške** izračunati razliku vrijednosti gradijenata dobivenih analitičkim i numeričkim postupkom

Provjera ispravnosti

Ideja postupka:

1. Nekakvim **postupkom** numerički aproksimirati vrijednost gradijenta
2. Nekakvom **mjerom pogreške** izračunati razliku vrijednosti gradijenata dobivenih analitičkim i numeričkim postupkom
3. Zabrinuti se ako je pogreška veća od neke unaprijed utvrđene **granice**

Numerička aproksimacija gradijenta

Derivacije

- Egzaktno:

$$\frac{d}{dx}f(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \tag{1}$$

$$\frac{\partial}{\partial x_i} f\left(\vec{x}\right) = \lim_{h \to 0} \frac{f\left(x_i + h, \vec{x}\right) - f\left(\vec{x}\right)}{h} \tag{2}$$

Derivacije

- Egzaktno:

$$\frac{d}{dx}f(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \tag{1}$$

$$\frac{\partial}{\partial x_i} f\left(\vec{x}\right) = \lim_{h \to 0} \frac{f\left(x_i + h, \vec{x}\right) - f\left(\vec{x}\right)}{h} \tag{2}$$

- Približno:

$$\frac{d}{dx}f(x) \approx \frac{f(x+h) - f(x)}{h} \tag{3}$$

$$\frac{\partial}{\partial x_i} f\left(\vec{x}\right) \approx \frac{f\left(x_i + h, \vec{x}\right) - f\left(\vec{x}\right)}{h} \tag{4}$$

Za ”mali” h

Konačne diferencije

Aproksimacija derivacije:

- Unaprijednom diferencijom

$$\frac{d}{dx}f(x) \approx \frac{f(x+h) - f(x)}{h}$$

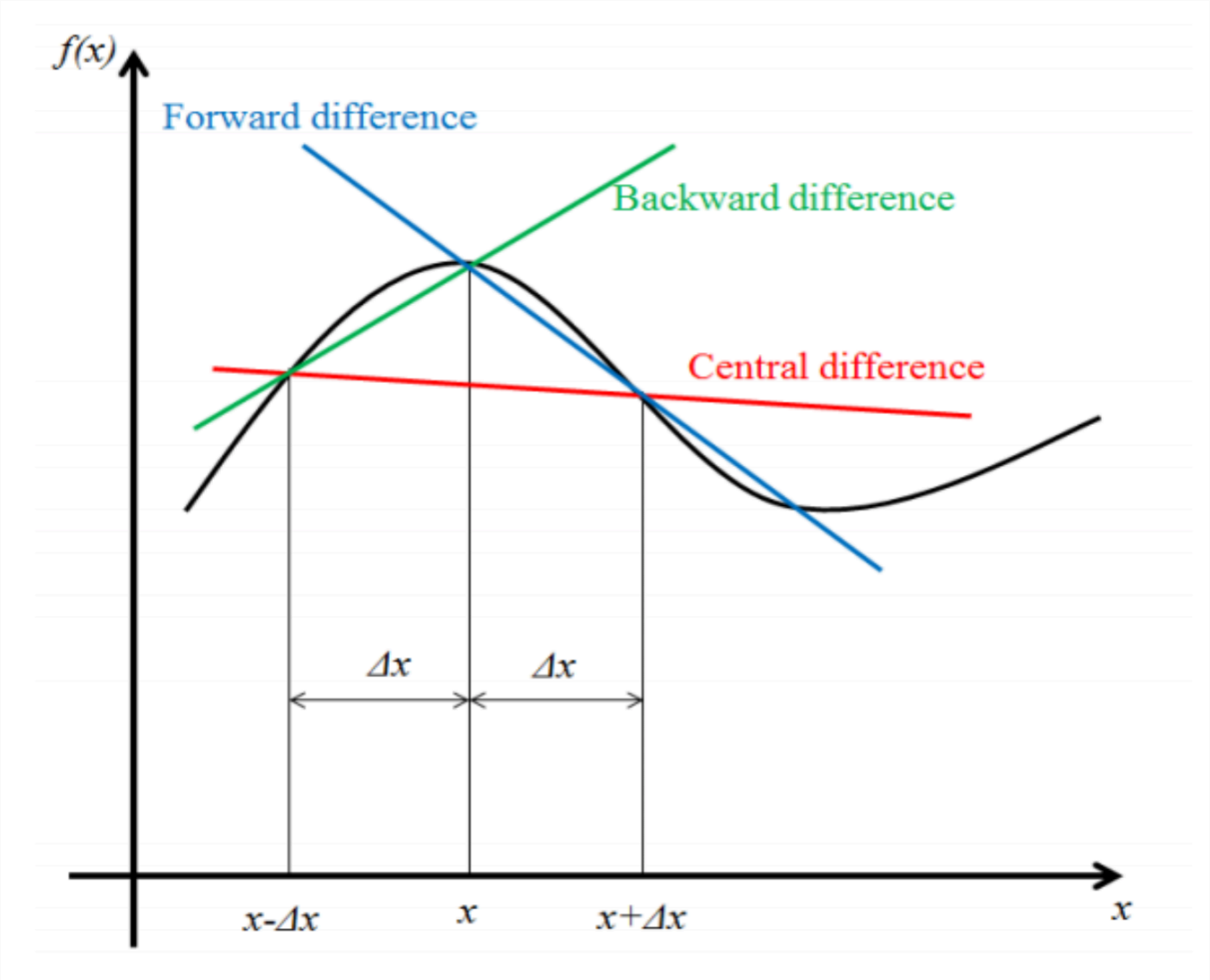
- Unatražnom diferencijom

$$\frac{d}{dx}f(x) \approx \frac{f(x) - f(x-h)}{h}$$

- **Centralnom diferencijom**

$$\frac{d}{dx}f(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

Konačne diferencije



Zašto centralna diferencija

Taylor:

$$f(x+h) = f(x) + \frac{f'(x)}{1!}h + \frac{f''(x)}{2!}h^2 + \dots \tag{5}$$

$$f(x-h) = f(x) - \frac{f'(x)}{1!}h + \frac{f''(x)}{2!}h^2 - \dots \tag{6}$$

Zašto centralna diferencija

Unaprijedna diferencija:

$$\frac{f(x+h)-f(x)}{h} = f'(x) + \frac{f''(x)}{2!}h + \frac{f'''(x)}{3!}h^2 + \dots \tag{7}$$

$$\frac{f(x+h)-f(x)}{h} = f'(x) + O(h) \tag{8}$$

Centralna diferencija:

$$\frac{f(x+h)-f(x-h)}{2h} = f'(x) + \frac{f'''(x)}{3!}h^2 + \dots \tag{9}$$

$$\frac{f(x+h)-f(x-h)}{2h} = f'(x) + O(h^2) \tag{10}$$

Osnovni algoritam numeričke provjere

Pseudokod

```
grad_check(forward_f, forward_backward_f, θ, h, tol):
    a_grads = forward_backward_f(θ)

    ZA SVAKI indeks idx U θ:
        θ_plus = elementu θ[idx] dodaj h
        f_plus = forward_f(θ_plus)

        θ_minus = elementu θ[idx] oduzmi h
        f_minus = forward_f(θ_minus)

        n_grad = (f_plus - f_minus) / (2h)
        a_grad = a_grads[idx]

        AKO error(n_grad, a_grad) > tol:
            # zabrini se !
```

Mjera pogreške

- Apsolutna pogreška

Mjera pogreške

- Apsolutna pogreška
  - $abs\_err(a\_grad, n\_grad) = |a\_grad - n\_grad|$

Mjera pogreške

- Apsolutna pogreška
  - $abs\_err(a\_grad, n\_grad) = |a\_grad - n\_grad|$
  - Nije dovoljno informativna:
    $abs\_err(10.001, 10.0001) = abs\_err(10^{-3}, 10^{-4})$

Mjera pogreške

- Apsolutna pogreška
  - $abs\_err(a\_grad, n\_grad) = |a\_grad - n\_grad|$
  - Nije dovoljno informativna:
    $abs\_err(10.001, 10.0001) = abs\_err(10^{-3}, 10^{-4})$
- Relativna pogreška

Mjera pogreške

- Apsolutna pogreška
  - $abs\_err(a\_grad, n\_grad) = |a\_grad - n\_grad|$
  - Nije dovoljno informativna:
    $abs\_err(10.001, 10.0001) = abs\_err(10^{-3}, 10^{-4})$
- Relativna pogreška
  - $\mathbf{rel\_err(a\_grad, n\_grad) = abs\_err(a\_grad, n\_grad)/Z}$

Mjera pogreške

- Apsolutna pogreška
  - $abs\_err(a\_grad, n\_grad) = |a\_grad - n\_grad|$
  - Nije dovoljno informativna:
    $abs\_err(10.001, 10.0001) = abs\_err(10^{-3}, 10^{-4})$
- Relativna pogreška
  - $\mathbf{rel\_err(a\_grad, n\_grad) = abs\_err(a\_grad, n\_grad)/Z}$
  - $Z = |a\_grad|$

Mjera pogreške

- Apsolutna pogreška
  - $abs\_err(a\_grad, n\_grad) = |a\_grad - n\_grad|$
  - Nije dovoljno informativna:
    $abs\_err(10.001, 10.0001) = abs\_err(10^{-3}, 10^{-4})$
- Relativna pogreška
  - $\mathbf{rel\_err(a\_grad, n\_grad) = abs\_err(a\_grad, n\_grad)/Z}$
  - $Z = |a\_grad|$
  - $Z = max(|a\_grad|, |n\_grad|)$

Mjera pogreške

- Apsolutna pogreška
  - $abs\_err(a\_grad, n\_grad) = |a\_grad - n\_grad|$
  - Nije dovoljno informativna:
    $abs\_err(10.001, 10.0001) = abs\_err(10^{-3}, 10^{-4})$
- Relativna pogreška
  - $\mathbf{rel\_err(a\_grad, n\_grad) = abs\_err(a\_grad, n\_grad)/Z}$
  - $Z = |a\_grad|$
  - $Z = max(|a\_grad|, |n\_grad|)$
  - $Z = |a\_grad| + |n\_grad|$

Mjera pogreške

- Apsolutna pogreška
  - $abs\_err(a\_grad, n\_grad) = |a\_grad - n\_grad|$
  - Nije dovoljno informativna:
    $abs\_err(10.001, 10.0001) = abs\_err(10^{-3}, 10^{-4})$
- Relativna pogreška
  - $\mathbf{rel\_err(a\_grad, n\_grad) = abs\_err(a\_grad, n\_grad)/Z}$
  - $Z = |a\_grad|$
  - $Z = max(|a\_grad|, |n\_grad|)$
  - $Z = |a\_grad| + |n\_grad|$
- U slučaju da je $n\_grad = a\_grad = 0$

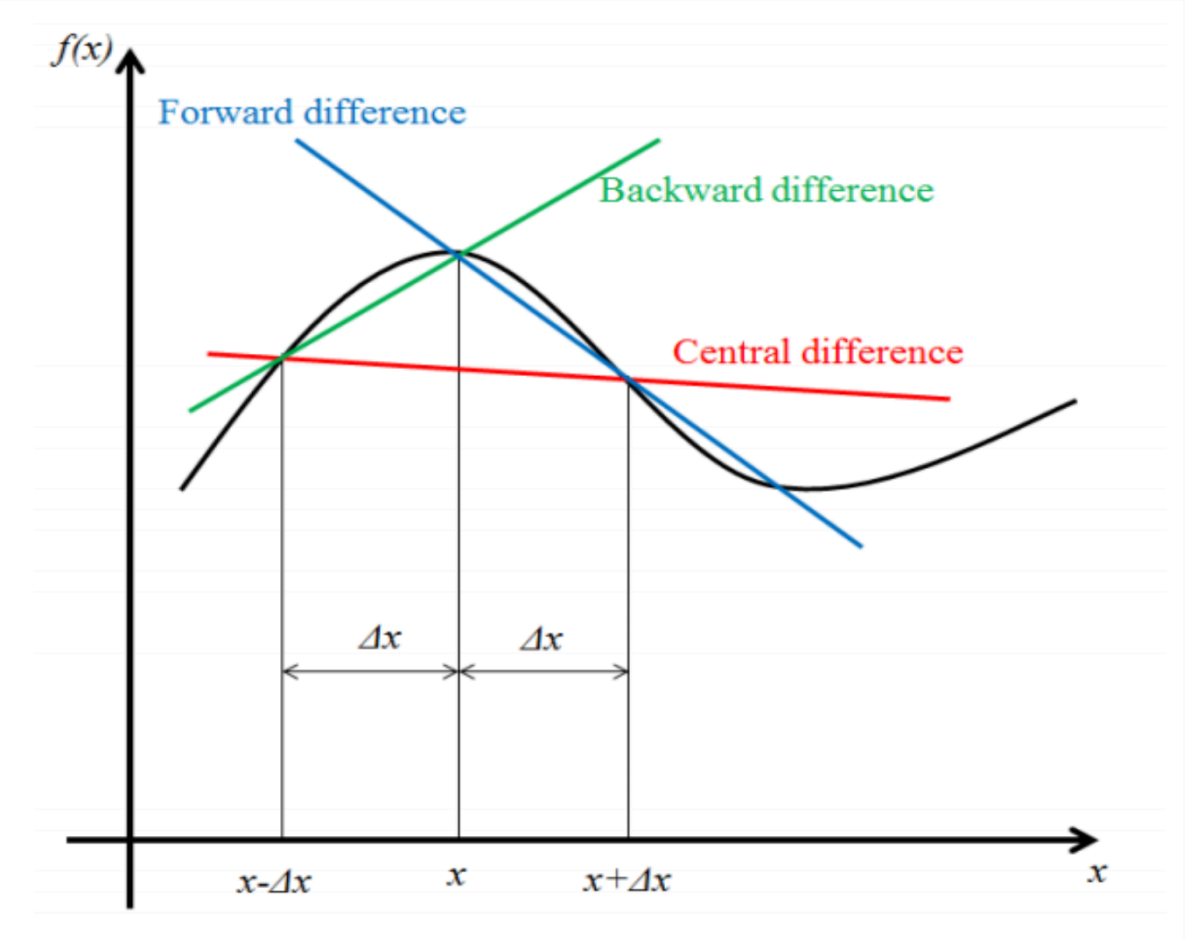
Mjera pogreške

- Apsolutna pogreška
  - $abs\_err(a\_grad, n\_grad) = |a\_grad - n\_grad|$
  - Nije dovoljno informativna:
    $abs\_err(10.001, 10.0001) = abs\_err(10^{-3}, 10^{-4})$
- Relativna pogreška
  - $\mathbf{rel\_err(a\_grad, n\_grad) = abs\_err(a\_grad, n\_grad)/Z}$
  - $Z = |a\_grad|$
  - $Z = max(|a\_grad|, |n\_grad|)$
  - $Z = |a\_grad| + |n\_grad|$
- U slučaju da je $n\_grad = a\_grad = 0$
  - $\mathbf{Z = max(10^{-8}, max(|a\_grad|, |n\_grad|))}$

Napomene

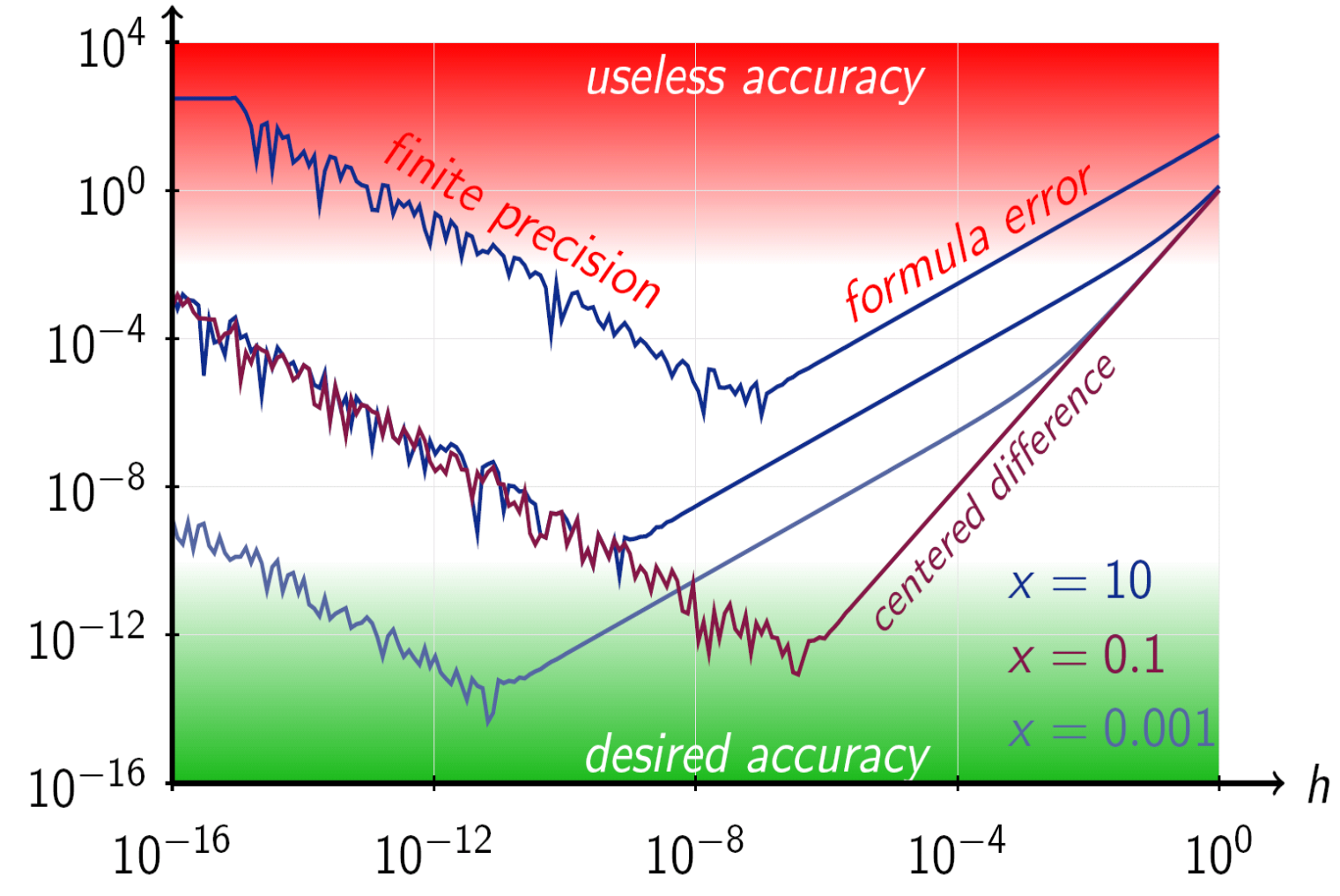
Korak h

- Što je manji, to je bolja aproksimacija derivacije.



Korak h

Ipak, ne želimo premali h - numerički problemi



Korak h. Cancellation.

...the relative error committed when subtracting two nearby quantities can be very large. In other words, the evaluation of any expression containing a subtraction (or an addition of quantities with opposite signs) could result in a relative error so large that all the digits are meaningless...

— David Goldberg, *What Every Computer Scientist Should Know About Floating-Point Arithmetic*

Korak h

Dakle:

- Ako je h prevelik, loše aproksimiramo derivaciju

Korak h

Dakle:

- Ako je h prevelik, loše aproksimiramo derivaciju
  - Procjena nagiba tangente sekantom je lošija

Korak h

Dakle:

- Ako je h prevelik, loše aproksimiramo derivaciju
  - Procjena nagiba tangente sekantom je lošija
- Ako je h premalen, opasnost od numeričkih pogrešaka

Korak h

Dakle:

- Ako je h prevelik, loše aproksimiramo derivaciju
  - Procjena nagiba tangente sekantom je lošija
- Ako je h premalen, opasnost od numeričkih pogrešaka
  - Za premali h, razlika f(x + h) - f(x - h) će se zaokružiti na 0 (**cancellation**)

Korak h

Dakle:

- Ako je h prevelik, loše aproksimiramo derivaciju
  - Procjena nagiba tangente sekantom je lošija
- Ako je h premalen, opasnost od numeričkih pogrešaka
  - Za premali h, razlika f(x + h) - f(x - h) će se zaokružiti na 0 (**cancellation**)
- **U praksi:** 10^{-4}, 10^{-5}

Tolerancija (tol)

Dozvoljeno odstupanje gradijenta od željenog:

- Bilo bi lijepo da je manje od 10^{-5}.

Tolerancija (tol)

Dozvoljeno odstupanje gradijenta od željenog:

- Bilo bi lijepo da je manje od 10^{-5}.
- Kod mreža s glatkim funkcijama kao tanh i sigmoidom odstupanje će vjerojatno biti manje nego kod mreža s neglatkim funkcijama (ReLU).

ReLU, SVM loss, ...

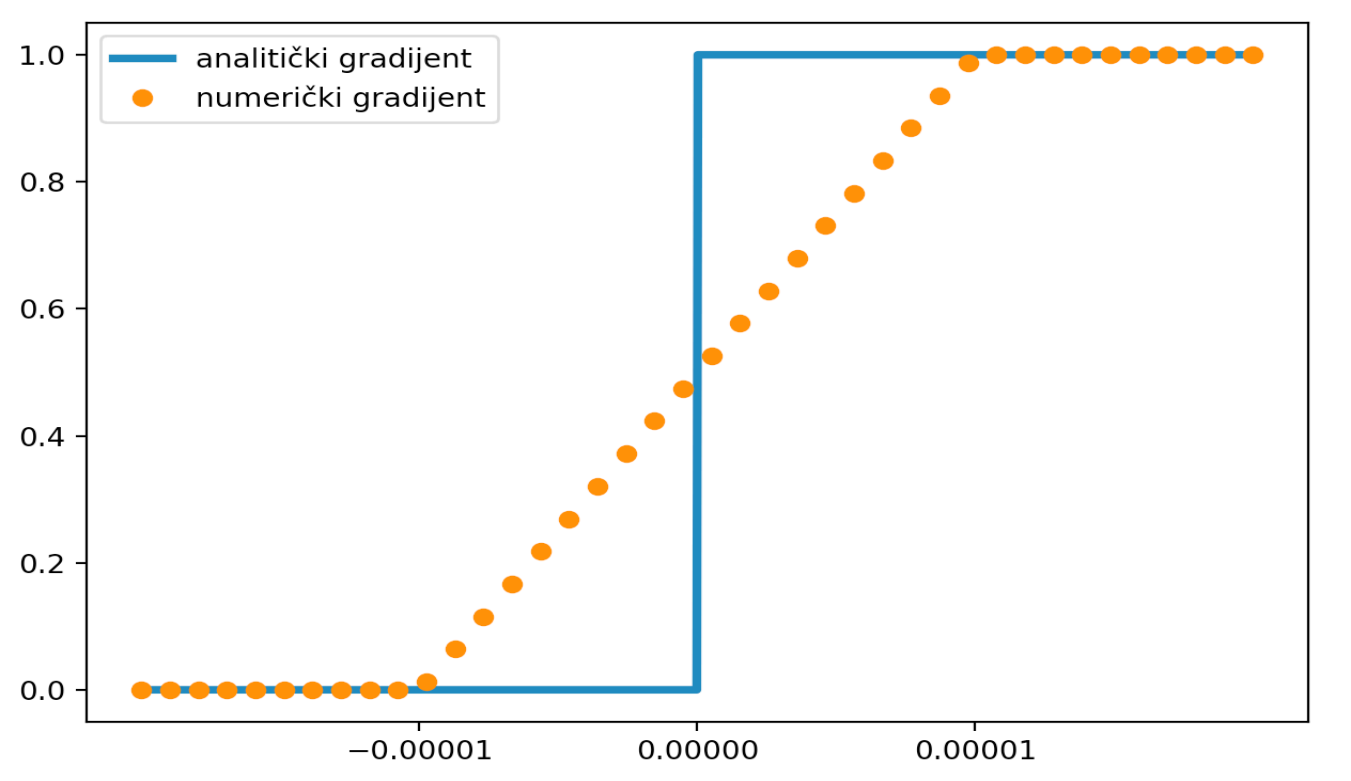
”Neglatke” funkcije

- ReLU nije diferencijabilna u 0. Kod analitičkog gradijenta to nije problem.

ReLU, SVM loss, ...

"Neglatke" funkcije

- ReLU nije diferencijabilna u 0. Kod analitičkog gradijenta to nije problem.
- Kod numeričkog, cijeli interval (-h, h) je problematičan.



ReLU

Ako je $x + h > 0$, a $x - h < 0$ onda nastaje problem.
Rješenje:

- Manji h.

ReLU

Ako je $x + h > 0$, a $x - h < 0$ onda nastaje problem.
Rješenje:

- Manji h.
- U algoritamu pratiti, detektirati taj problematični slučaj i ignorirati ga.

Ostale napomene

- Gradient check usred treniranja, a ne na samom početku - ne želimo patološke slučajeve.

Ostale napomene

- Gradient check usred treniranja, a ne na samom početku - ne želimo patološke slučajeve.
- Premali gradijenti

Ostale napomene

- Gradient check usred treniranja, a ne na samom početku - ne želimo patološke slučajeve.
- Premali gradijenti
  - Ako su sami gradijenti iznosom premali, može doći do numeričkih problema

Ostale napomene

- Gradient check usred treniranja, a ne na samom početku - ne želimo patološke slučajeve.
- Premali gradijenti
  - Ako su sami gradijenti iznosom premali, može doći do numeričkih problema
  - Pomnožiti loss velikom konstantom da se gradijenti dovedu u bolje područje

Ostale napomene

- Gradient check usred treniranja, a ne na samom početku - ne želimo patološke slučajeve.
- Premali gradijenti
  - Ako su sami gradijenti iznosom premali, može doći do numeričkih problema
  - Pomnožiti loss velikom konstantom da se gradijenti dovedu u bolje područje
- Stohastički elementi i slojevi - dropout, ...

Ostale napomene

- Gradient check usred treniranja, a ne na samom početku - ne želimo patološke slučajeve.
- Premali gradijenti
  - Ako su sami gradijenti iznosom premali, može doći do numeričkih problema
  - Pomnožiti loss velikom konstantom da se gradijenti dovedu u bolje područje
- Stohastički elementi i slojevi - dropout, ...
  1. Isključiti ih

Ostale napomene

- Gradient check usred treniranja, a ne na samom početku - ne želimo patološke slučajeve.
- Premali gradijenti
  - Ako su sami gradijenti iznosom premali, može doći do numeričkih problema
  - Pomnožiti loss velikom konstantom da se gradijenti dovedu u bolje područje
- Stohastički elementi i slojevi - dropout, ...
  1. Isključiti ih
  2. Prije svake evaluacije (f(x+h), f(x-h), analitički gradijent) fiksirati isti seed

Ostale napomene

- Gradient check usred treniranja, a ne na samom početku - ne želimo patološke slučajeve.
- Premali gradijenti
  - Ako su sami gradijenti iznosom premali, može doći do numeričkih problema
  - Pomnožiti loss velikom konstantom da se gradijenti dovedu u bolje područje
- Stohastički elementi i slojevi - dropout, ...
  1. Isključiti ih
  2. Prije svake evaluacije (f(x+h), f(x-h), analitički gradijent) fiksirati isti seed
- Provjera samo podskupa parametara

Ostale napomene

- Gradient check usred treniranja, a ne na samom početku - ne želimo patološke slučajeve.
- Premali gradijenti
  - Ako su sami gradijenti iznosom premali, može doći do numeričkih problema
  - Pomnožiti loss velikom konstantom da se gradijenti dovedu u bolje područje
- Stohastički elementi i slojevi - dropout, ...
  1. Isključiti ih
  2. Prije svake evaluacije (f(x+h), f(x-h), analitički gradijent) fiksirati isti seed
- Provjera samo podskupa parametara
  - Imati na umu da svi različiti parametri trebaju biti zastupljeni (W, b, ...)

Jednostavan primjer s RNN-om

Potrebne funkcije

```
def rnn_forward_function(params):
    x, y = data.get_next()
    return rnn.forward_loss(h0, x, y, params)

def rnn_forward_backward_function(params):
    x, y = data.get_next()
    return rnn.forward_backward(h0, x, y, params)

initial_params = {
    "U": rnn.U,
    "W": rnn.W,
    "b": rnn.b,
    "V": rnn.V,
    "c": rnn.c
}
```

Gradient check postupak

```
def grad_check(forward, forward_back, params, h, tol):
  a_grads = forward_back(params)

  for tup in param_plus_minus(params, a_grads, h):
    param_id, param_minus, param_plus, a_grad = tup

    f_plus = forward(param_plus)
    f_minus = forward(param_minus)

    n_grad = (f_plus - f_minus) / (2 * h)

    if not are_equal(a_grad, n_grad, tol):
      print_alert(a_grad, n_grad, param_id)
```

Variranje vrijednosti parametara

```
def param_plus_minus(params, a_grad_dict, h):
  for key in params.keys():
    param = params[key]
    a_grads = a_grad_dict[key]

    for idx, el in np.ndenumerate(param_array):
      a_grad = a_grads[idx]

      minus_array = np.copy(param)
      minus_array[idx] -= h
      plus_array = np.copy(param)
      plus_array[idx] += h

      # ... p_minus, p_plus, param_id, ...

      yield (param_id, p_minus, p_plus, a_grad)
```

Pogreška

```
def are_equal(a_grad, n_grad, tol):
    abs_err = abs(a_grad - n_grad)
    max_grad = max(abs(a_grad), abs(n_grad))
    abs_max = max(1e-10, max_grad)

    return (abs_err / abs_max) < tol
```

Kompleksni brojevi

Aproksimacija derivacije kompleksnim brojevima

Ako programski jezik podržava operacije s kompleksnim brojevima:

- $f(x + ih) = f(x) + ihf'(x) + O(h^2)$

Aproksimacija derivacije kompleksnim brojevima

Ako programski jezik podržava operacije s kompleksnim brojevima:

- $f(x + ih) = f(x) + ihf'(x) + O(h^2)$
- $Im[f(x + ih)/h] = f'(x) + O(h^2)$

Aproksimacija derivacije kompleksnim brojevima

Ako programski jezik podržava operacije s kompleksnim brojevima:

- $f(x + ih) = f(x) + ihf'(x) + O(h^2)$
- $Im[f(x + ih)/h] = f'(x) + O(h^2)$
- Nema cancellation effecta - h se može smanjiti na jako malen broj (10^{-150})

Aproksimacija derivacije kompleksnim brojevima

Ako programski jezik podržava operacije s kompleksnim brojevima:

- $f(x + ih) = f(x) + ihf'(x) + O(h^2)$
- $Im[f(x + ih)/h] = f'(x) + O(h^2)$
- Nema cancellation effecta - h se može smanjiti na jako malen broj (10^{-150})
- Nema problema s ReLU-om

Aproksimacija derivacije kompleksnim brojevima

Ako programski jezik podržava operacije s kompleksnim brojevima:

- $f(x + ih) = f(x) + ihf'(x) + O(h^2)$
- $Im[f(x + ih)/h] = f'(x) + O(h^2)$
- Nema cancellation effecta - h se može smanjiti na jako malen broj (10^{-150})
- Nema problema s ReLU-om
- `num_grad = (forward_f(x + h * 1j) / h).imag`

Zaključak

Zaključak

- Krenuli smo od jednostavne ideje.

Zaključak

- Krenuli smo od jednostavne ideje.
- Putem smo se dotakli raznih detalja koji nam možda na prvi pogled nisu bili očiti.

Zaključak

- Krenuli smo od jednostavne ideje.
- Putem smo se dotakli raznih detalja koji nam možda na prvi pogled nisu bili očiti.
- Metoda očito ima svojih nesavršenosti, ali je i dalje korisna, pogotovo jer se te nesavršenosti rijetko manifestiraju. Ipak moramo biti svjesni da one postoje i znati kako im doskočiti.

Hvala.
Pitanja?