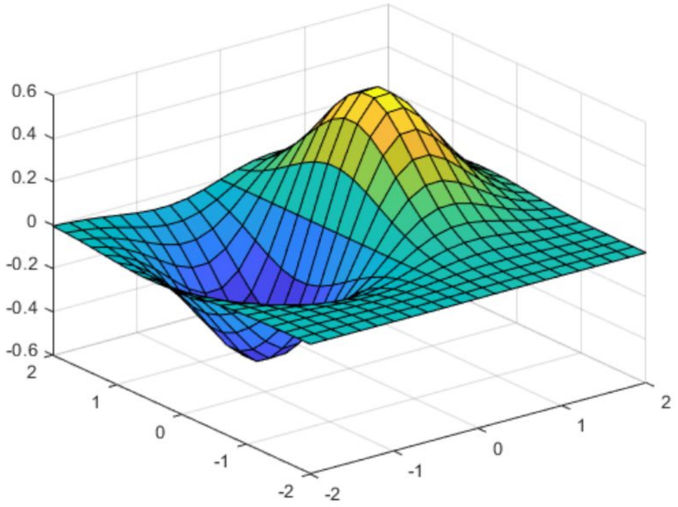
Proč je těžké učit hluboké sítě

Vanishing/exploding gradient problem



Deep learning

- Mělké sítě sice dokáží aproximovat jakékoliv funkce, praktické využití ale nacházejí hlavně hluboké sítě

Deep learning

- Mělké sítě sice dokáží aproximovat jakékoliv funkce, praktické využití ale nacházejí hlavně hluboké sítě
- Zkusíme do naší sítě přidat vrstvu/vrstvy

Deep learning

- Mělké sítě sice dokáží aproximovat jakékoliv funkce, praktické využití ale nacházejí hlavně hluboké sítě
- Zkusíme do naší sítě přidat vrstvu/vrstvy
- Zlepšení není žádné, nebo jen minimální… co se děje?

Vanishing gradient problem

- Různé vrstvy se učí různou rychlostí (čím pozdější vrstva, tím rychlejší učení)

Vanishing gradient problem

- Různé vrstvy se učí různou rychlostí (čím pozdější vrstva, tím rychlejší učení)
- Prozkoumáme tuto rychlost (gradient)

Vanishing gradient problem

- Různé vrstvy se učí různou rychlostí (čím pozdější vrstva, tím rychlejší učení)
- Prozkoumáme tuto rychlost (gradient)
- 3. rovnice backpropagation: $\frac{\partial C}{\partial b_j^l} = \delta_j^l$
- 4. rovnice backpropagation: $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1}\delta_j^l = a_k^{l-1}\frac{\partial C}{\partial b_j^l} < \frac{\partial C}{\partial b_j^l}$

Vanishing gradient problem

- Různé vrstvy se učí různou rychlostí (čím pozdější vrstva, tím rychlejší učení)
- Prozkoumáme tuto rychlost (gradient)
- 3. rovnice backpropagation: $\frac{\partial C}{\partial b_j^l} = \delta_j^l$
- 4. rovnice backpropagation: $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1}\delta_j^l = a_k^{l-1}\frac{\partial C}{\partial b_j^l} < \frac{\partial C}{\partial b_j^l}$
- Velikost gradientu je tedy přímo úměrná parciální derivaci účelové funkce dle biasu

Vanishing gradient problem

- V naší síti se dřívější vrstvy učí pomaleji, než pozdější (gradient se při zpětné propagaci chyb snižuje).

Vanishing gradient problem

- V naší síti se dřívější vrstvy učí pomaleji, než pozdější (gradient se při zpětné propagaci chyb snižuje).
- Tomuto jevu říkáme “vanishing gradient problem”

Vanishing gradient problem

- V naší síti se dřívější vrstvy učí pomaleji, než pozdější (gradient se při zpětné propagaci chyb snižuje).
- Tomuto jevu říkáme “vanishing gradient problem”
- Pokud při optimalizaci ***f*** je derivace **f** malá, možná už jsme blízko optima => to v našem případě neplatí, protože váhy a biasy jsme inicializovali náhodně a je velmi nepravděpodobné, že bychom náhodou trefili “správné” hodnoty

Vanishing gradient problem

- První vrstvy tak spíš “zahazují” informace, než že by pomáhaly učení

Vanishing gradient problem

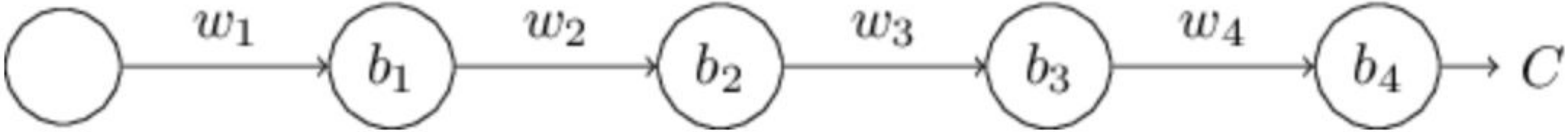
- První vrstvy tak spíš “zahazují” informace, než že by pomáhaly učení
- **Co se děje?**

Vanishing gradient problem

- První vrstvy tak spíš “zahazují” informace, než že by pomáhaly učení
- **Co se děje?**
- Uvažujme jednoduchou síť s jedním neuronem v každé vrstvě

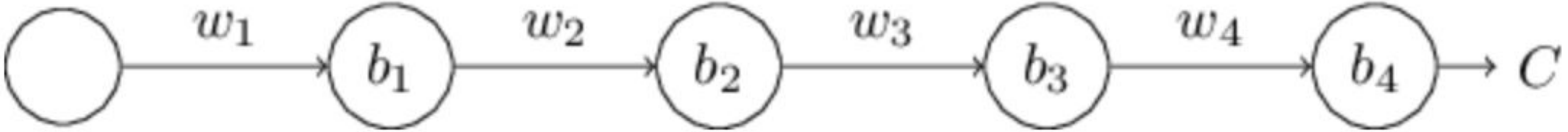
Vanishing gradient problem

- První vrstvy tak spíš “zahazují” informace, než že by pomáhaly učení
- **Co se děje?**
- Uvažujme jednoduchou síť s jedním neuronem v každé vrstvě



Vanishing gradient problem

- První vrstvy tak spíš "zahazují" informace, než že by pomáhaly učení
- **Co se děje?**
- Uvažujme jednoduchou síť s jedním neuronem v každé vrstvě

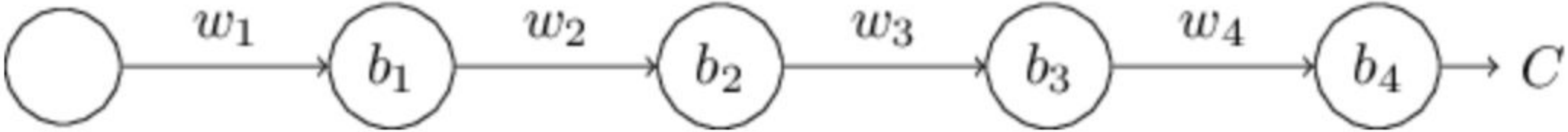


- Podíváme se na chybu neuronu v 1. vrstvě $\delta^1 = \frac{\partial C}{\partial b_1}$

$$\delta^1 = \frac{\partial C}{\partial b_1}$$

Vanishing gradient problem

- Změna biasu Δb_1 povede ke změně Δa_1, to povede ke změně Δz_2 atd., až se změna projeví ve výstupu sítě a tím pádem i v hodnotě účelové funkce ΔC



$\delta^1 = \frac{\partial C}{\partial b_1}$

Vanishing gradient problem

- Změna biasu Δb_1 povede ke změně Δa_1, to povede ke změně Δz_2 atd., až se změna projeví ve výstupu sítě a tím pádem i v hodnotě účelové funkce ΔC
- Máme tedy $\frac{\partial C}{\partial b_1} \approx \frac{\Delta C}{\Delta b_1}$

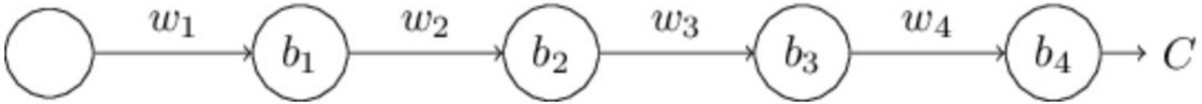
$$\delta^1 = \frac{\partial C}{\partial b_1}$$

Vanishing gradient problem

- Změna biasu Δb_1 povede ke změně Δa_1, to povede ke změně Δz_2 atd., až se změna projeví ve výstupu sítě a tím pádem i v hodnotě účelové funkce ΔC
- Máme tedy $\frac{\partial C}{\partial b_1} \approx \frac{\Delta C}{\Delta b_1}$
- Chybu neuronu tedy můžeme určit tak, že budeme sledovat, jak se změna v prvním biasu propaguje skrz síť

$$\delta^1 = \frac{\partial C}{\partial b_1}$$

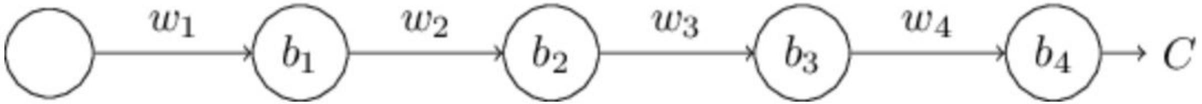
Vanishing gradient problem



- Pro první skrytou vrstvu: $a_1 = \sigma(z_1) = \sigma(w_1 a_0 + b_1)$,

$$\delta^1 = \frac{\partial C}{\partial b_1}$$

Vanishing gradient problem

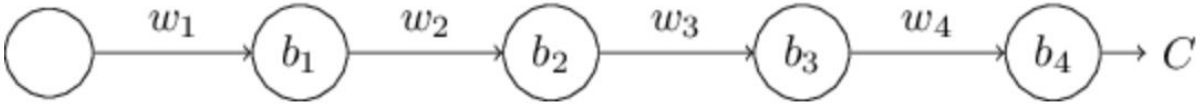


- Pro první skrytou vrstvu: $a_1 = \sigma(z_1) = \sigma(w_1 a_0 + b_1)$,

tj. $\Delta a_1 \approx \frac{\partial \sigma(w_1 a_0 + b_1)}{\partial b_1} \Delta b_1 = \sigma'(z_1) \Delta b_1$

$$\delta^1 = \frac{\partial C}{\partial b_1}$$

Vanishing gradient problem



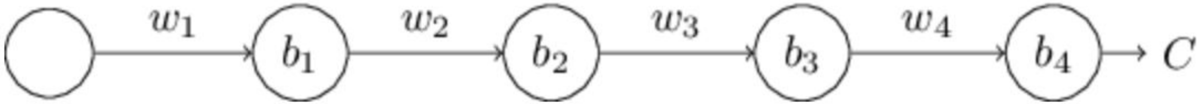
- Pro první skrytou vrstvu: $a_1 = \sigma(z_1) = \sigma(w_1 a_0 + b_1)$,

  tj. $\Delta a_1 \approx \dfrac{\partial \sigma(w_1 a_0 + b_1)}{\partial b_1} \Delta b_1 = \sigma'(z_1) \Delta b_1$

- Pro z druhé skryté vrstvy: $z_2 = w_2 a_1 + b_2$,

$$\delta^1 = \frac{\partial C}{\partial b_1}$$

Vanishing gradient problem



- Pro první skrytou vrstvu: $a_1 = \sigma(z_1) = \sigma(w_1 a_0 + b_1)$,

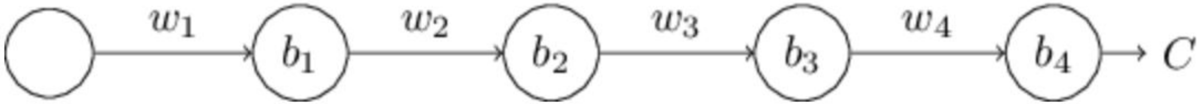
  tj. $\Delta a_1 \approx \dfrac{\partial \sigma(w_1 a_0 + b_1)}{\partial b_1} \Delta b_1 = \sigma'(z_1) \Delta b_1$

- Pro z druhé skryté vrstvy: $z_2 = w_2 a_1 + b_2$,

  tj. $\Delta z_2 \approx \dfrac{\partial z_2}{\partial a_1} \Delta a_1 = w_2 \Delta a_1$

$$\delta^1 = \frac{\partial C}{\partial b_1}$$

Vanishing gradient problem



- Pro první skrytou vrstvu: $a_1 = \sigma(z_1) = \sigma(w_1 a_0 + b_1)$,

  tj. $\Delta a_1 \approx \dfrac{\partial \sigma(w_1 a_0 + b_1)}{\partial b_1} \Delta b_1 = \sigma'(z_1) \Delta b_1$

- Pro z druhé skryté vrstvy: $z_2 = w_2 a_1 + b_2$,

  tj. $\Delta z_2 \approx \dfrac{\partial z_2}{\partial a_1} \Delta a_1 = w_2 \Delta a_1$

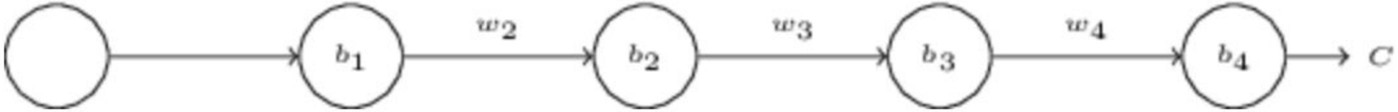
- Dohromady: $\Delta z_2 \approx w_2 \Delta a_1 = w_2 \sigma'(z_1) \Delta b_1$

$$\delta^1 = \frac{\partial C}{\partial b_1}$$

Vanishing gradient problem

- Pokračujeme dál:

$$\frac{\partial C}{\partial b_1} = \sigma'(z_1) \times w_2 \times \sigma'(z_2) \times w_3 \times \sigma'(z_3) \times w_4 \times \sigma'(z_4) \times \frac{\partial C}{\partial a_4}$$

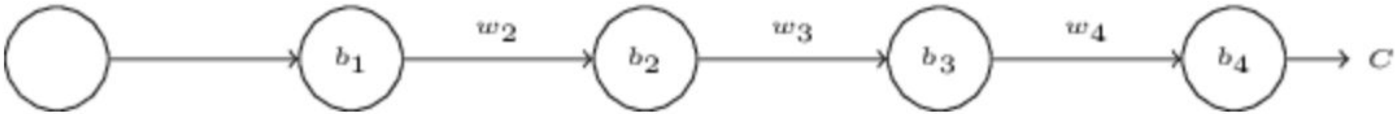


$$\delta^1 = \frac{\partial C}{\partial b_1}$$

Vanishing gradient problem

- Pokračujeme dál:

$$\frac{\partial C}{\partial b_1} = \sigma'(z_1) \times w_2 \times \sigma'(z_2) \times w_3 \times \sigma'(z_3) \times w_4 \times \sigma'(z_4) \times \frac{\partial C}{\partial a_4}$$



$$\Delta C \approx \sigma'(z_1) w_2 \sigma'(z_2) w_3 \sigma'(z_3) w_4 \sigma'(z_4) \frac{\partial C}{\partial a_4} \Delta b_1$$

$$\frac{\partial C}{\partial b_1} \approx \frac{\Delta C}{\Delta b_1} \approx \sigma'(z_1) w_2 \sigma'(z_2) w_3 \sigma'(z_3) w_4 \sigma'(z_4) \frac{\partial C}{\partial a_4}$$

Vanishing gradient problem

- Proč gradient mizí? $$\frac{\partial C}{\partial b_1} = \sigma'(z_1) w_2 \sigma'(z_2) w_3 \sigma'(z_3) w_4 \sigma'(z_4) \frac{\partial C}{\partial a_4}$$

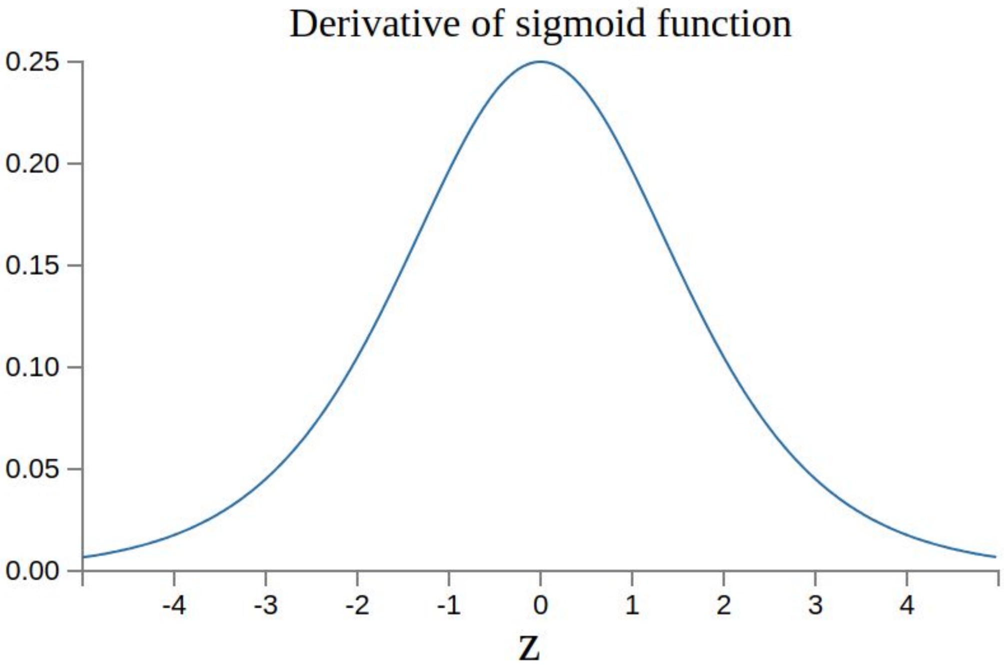
Vanishing gradient problem

- Proč gradient mizí? $$\frac{\partial C}{\partial b_1} = \sigma'(z_1) w_2 \sigma'(z_2) w_3 \sigma'(z_3) w_4 \sigma'(z_4) \frac{\partial C}{\partial a_4}$$
- Produkt výrazů $w_j \sigma'(z_j)$ (až na poslední člen)

Vanishing gradient problem

- Proč gradient mizí? $\frac{\partial C}{\partial b_1} = \sigma'(z_1) w_2 \sigma'(z_2) w_3 \sigma'(z_3) w_4 \sigma'(z_4) \frac{\partial C}{\partial a_4}$
- Produkt výrazů $w_j \sigma'(z_j)$ (až na poslední člen)

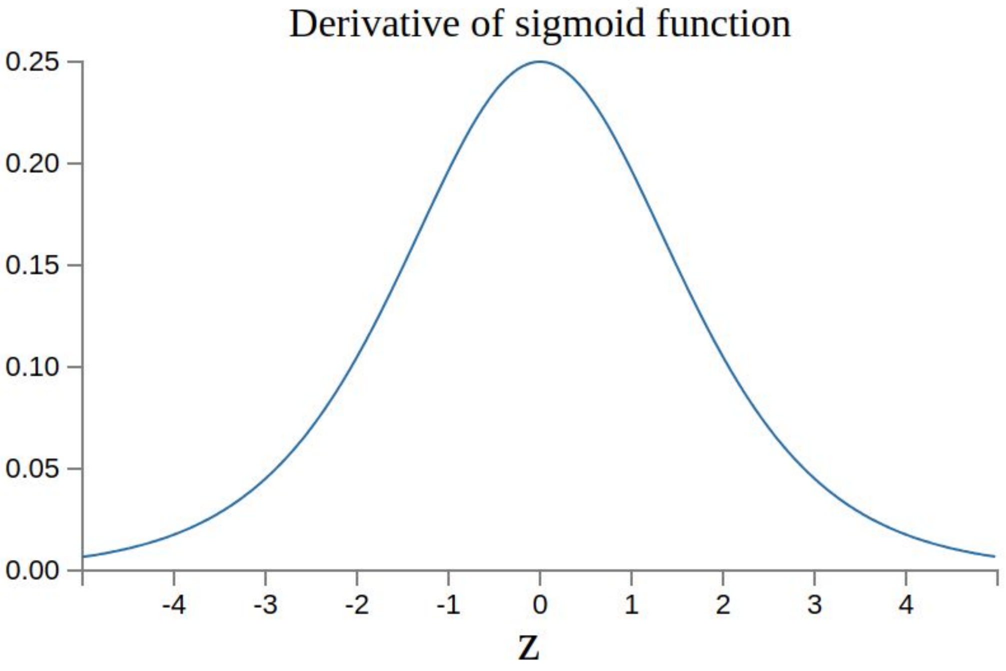
- $\max(\sigma') = \sigma'(0) = \frac{1}{4}$



Vanishing gradient problem

- Proč gradient mizí? $\frac{\partial C}{\partial b_1} = \sigma'(z_1) w_2 \sigma'(z_2) w_3 \sigma'(z_3) w_4 \sigma'(z_4) \frac{\partial C}{\partial a_4}$
- Produkt výrazů $w_j \sigma'(z_j)$ (až na poslední člen)

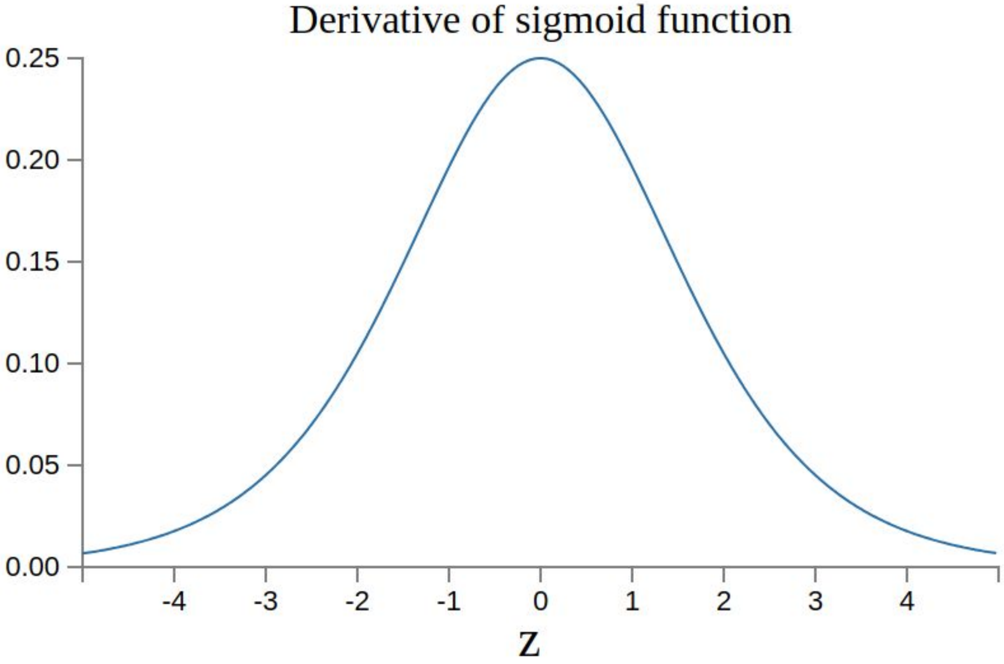
- $\max(\sigma') = \sigma'(0) = \frac{1}{4}$
- inicializace: $w_j \sim N(0,1) \implies$ většina vah splňuje $|w_j| < 1$



Vanishing gradient problem

- Proč gradient mizí? $\frac{\partial C}{\partial b_1} = \sigma'(z_1) w_2 \sigma'(z_2) w_3 \sigma'(z_3) w_4 \sigma'(z_4) \frac{\partial C}{\partial a_4}$
- Produkt výrazů $w_j \sigma'(z_j)$ (až na poslední člen)

- $\max(\sigma') = \sigma'(0) = \frac{1}{4}$
- inicializace: $w_j \sim N(0,1) \implies$ většina vah splňuje $|w_j| < 1$
- většinou $|w_j \sigma'(z_j)| < \frac{1}{4}$



Vanishing gradient problem

- Můžeme spočítat derivaci podle třetího biasu:

$$\frac{\partial C}{\partial b_1} = \sigma'(z_1) w_2 \sigma'(z_2) w_3 \sigma'(z_3) w_4 \sigma'(z_4) \frac{\partial C}{\partial a_4}$$

$$\frac{\partial C}{\partial b_3} = \sigma'(z_3) w_4 \sigma'(z_4) \frac{\partial C}{\partial a_4}$$

Vanishing gradient problem

- Můžeme spočítat derivaci podle třetího biasu:

$$\frac{\partial C}{\partial b_1} = \sigma'(z_1) w_2 \sigma'(z_2) w_3 \sigma'(z_3) w_4 \sigma'(z_4) \frac{\partial C}{\partial a_4}$$

$$\frac{\partial C}{\partial b_3} = \sigma'(z_3) w_4 \sigma'(z_4) \frac{\partial C}{\partial a_4}$$

- Gradienty (chyby neuronu) se exponenciálně snižují

Vanishing gradient problem

- Můžeme spočítat derivaci podle třetího biasu:

$$\frac{\partial C}{\partial b_1} = \sigma'(z_1) w_2 \sigma'(z_2) w_3 \sigma'(z_3) w_4 \sigma'(z_4) \frac{\partial C}{\partial a_4}$$

$$\frac{\partial C}{\partial b_3} = \sigma'(z_3) w_4 \sigma'(z_4) \frac{\partial C}{\partial a_4}$$

- Gradienty (chyby neuronu) se exponenciálně snižují
- Pokud budou váhy růst tak, že $|w_j \sigma'(z_j)| > 1$, situace se nazývá ***exploding gradient problem***

Vanishing gradient problem

- Obecně hovoříme o problému nestabilního gradientu (unstable gradient problem)

Vanishing gradient problem

- Obecně hovoříme o problému nestabilního gradientu (unstable gradient problem)
- Při použití sigmoidu jako aktivační funkce většinou dochází k vymizení gradientu (derivace sigmoidu závisí rovněž na váze, takže čím vyšší váha, tím nižší $\sigma'(z_j)$)

Vanishing gradient problem: cvičení 1

- **Pomůže s problémem změna aktivační funkce?**

Vanishing gradient problem: cvičení 1

- **Pomůže s problémem změna aktivační funkce?**
- Ano. Průměrně váhy inicializujeme $\sqrt{\frac{2}{\pi}} \approx 0.8$, chceme tedy takovou aktivační funkci, aby její derivace byla ~1
- **Zkusme relu**

Vanishing gradient problem: problém 1

- **Dokažte, že** $|w_j \sigma'(z_j)| > 1 \implies w_j \geq 4$

Vanishing gradient problem: problém 1

- **Dokažte, že** $|w_j \sigma'(z_j)| > 1 \implies w_j \geq 4$

  Protože $\forall z \in \mathbb{R}, \sigma'(z) \leq \frac{1}{4}$

Vanishing gradient problem: problém 1

- **Dokažte, že** $|w_j\sigma'(z_j)| > 1 \implies w_j \geq 4$
  Protože $\forall z \in \mathbb{R}, \sigma'(z) \leq \frac{1}{4}$
- **Nechť** $|w| \geq 4$ **. Dokažte, že množina aktivací, pro které je** $|w_j\sigma'(z_j)| > 1$ **spadá do intervalu**
  $$\frac{2}{|w|}\ln\left(\frac{|w|\left(1+\sqrt{1-\frac{4}{|w|}}\right)}{2}-1\right)$$
- **Numericky ukažte, že je tento interval největší pro** $|w| \approx 6.9$

Unstable gradient problem

- Stejný problém, jaký jsme pozorovali pro malou síť nastane i pro komplexnější síť:

$$\delta^l = \Sigma'(z^l)(w^{l+1})^T \Sigma'(z^{l+1})(w^{l+2})^T \ldots \Sigma'(z^L) \nabla_a C$$

kde sigmy jsou diagonální matice s derivací sigmoidu na diagonále, tj. matice s hodnotami <¼

Unstable gradient problem

- Co s tím?
  a. lepší aktivační funkce
  b. batch normalization
  c. inicializace vah
  d. skip connectioins
  e. gradient clipping
  f. Layer-wise Pretraining