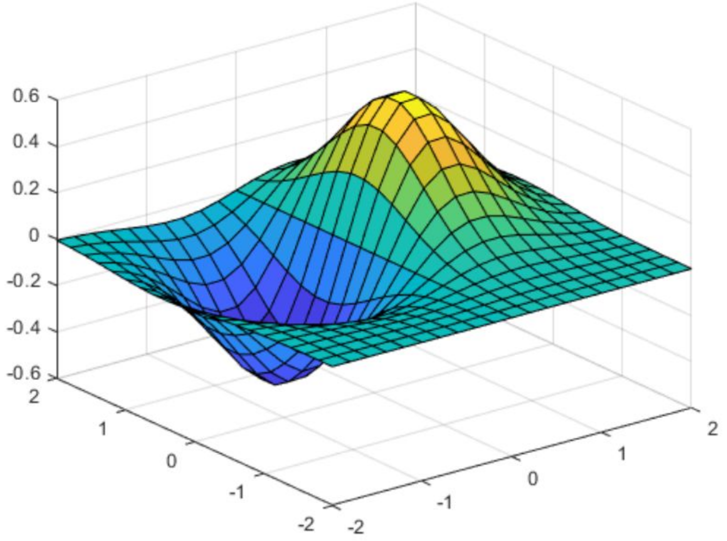
Jak zefektivnit učení

neuronových sítí



Jak zeefktivnit učení neuronové sítě

- Jiná účelová funkce

Jak zeefktivnit učení neuronové sítě

- Jiná účelová funkce
- Jiné aktivační funkce (zobecnění neuronu)

Jak zeefktivnit učení neuronové sítě

- Jiná účelová funkce
- Jiné aktivační funkce (zobecnění neuronu)
- Regularizace

Jak zeefktivnit učení neuronové sítě

- Jiná účelová funkce
- Jiné aktivační funkce (zobecnění neuronu)
- Regularizace
- Lepší inicializace vah

Jak zeefktivnit učení neuronové sítě

- Jiná účelová funkce
- Jiné aktivační funkce (zobecnění neuronu)
- Regularizace
- Lepší inicializace vah
- Heuristiky pro výběr hyperparametrů

Účelová funkce - je MSE dobrá?

- Chceme prozkoumat, jak dobře se síť s MSE učí. Vezmeme si jednoduchou úlohu - pouze jeden neuron, který má za úkol z 0 udělat 1

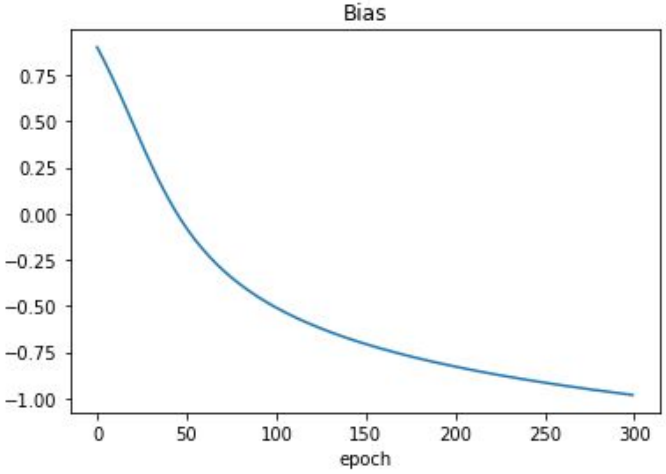
Účelová funkce - je MSE dobrá?

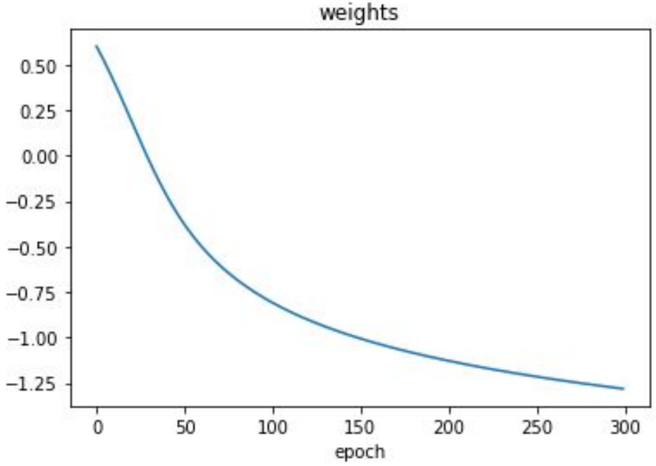
- Chceme prozkoumat, jak dobře se síť s MSE učí. Vezmeme si jednoduchou úlohu - pouze jeden neuron, který má za úkol z 0 udělat 1
- Váhu i bias bychom snadno určili, ale chceme prozkoumat chování učení

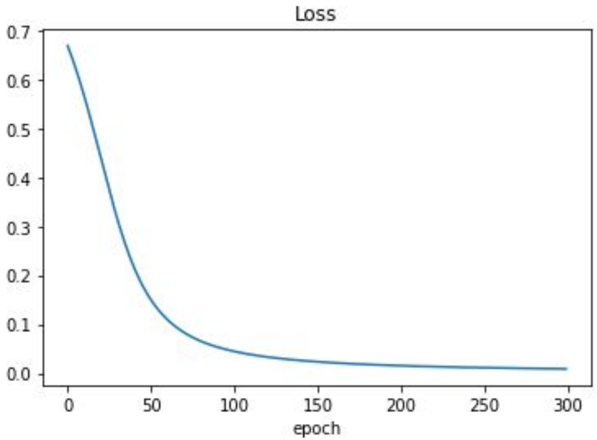
Účelová funkce - je MSE dobrá?

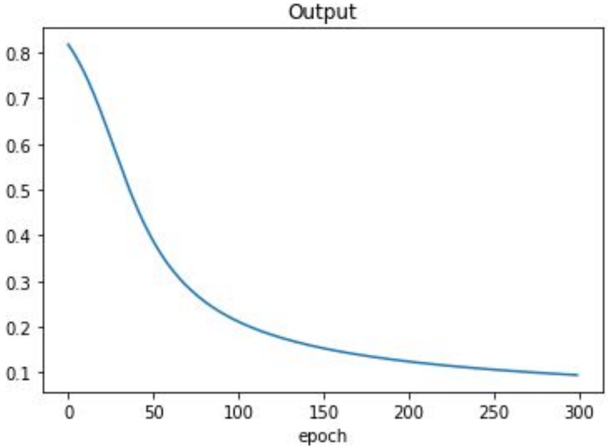
- Chceme prozkoumat, jak dobře se síť s MSE učí. Vezmeme si jednoduchou úlohu - pouze jeden neuron, který má za úkol z 0 udělat 1
- Váhu i bias bychom snadno určili, ale chceme prozkoumat chování učení
- Bias a váhu inicializujeme libovolně, např. 0.9 a 0.6, ale pevně.

Účelová funkce - je MSE dobrá?



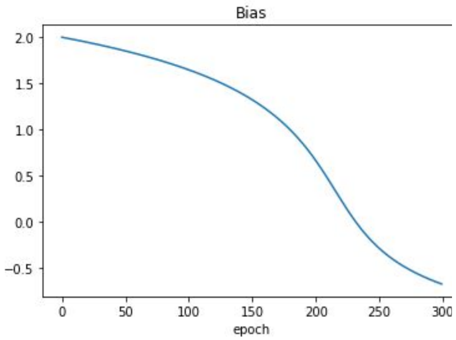


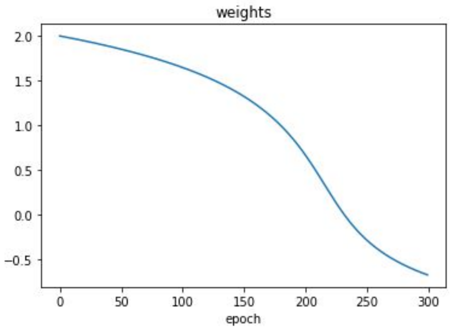


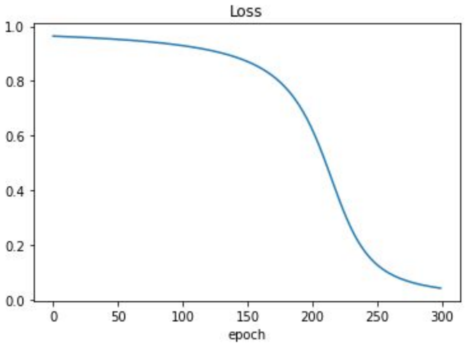


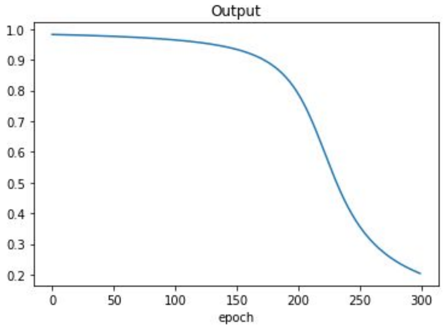
Účelová funkce - je MSE dobrá?

- Závisí na inicializace váhy a biasu. Pokud pro váhu i bias zvolíme 2, vidíme, že cca 150 epoch se síť prakticky neučí



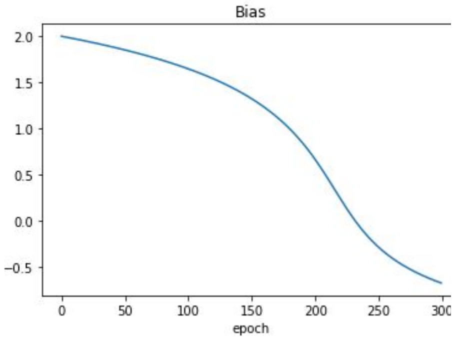


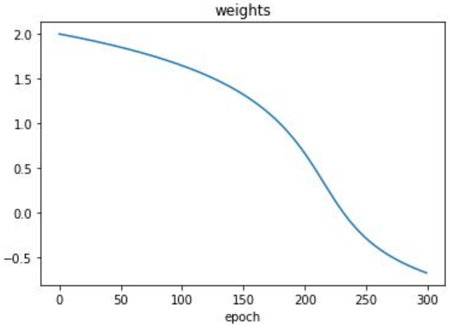


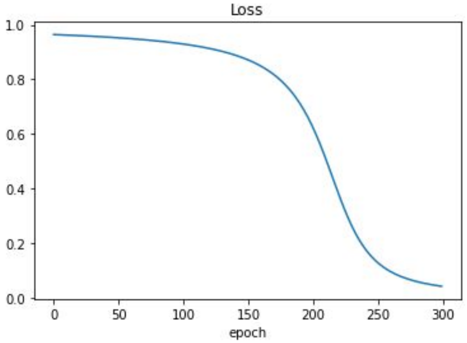


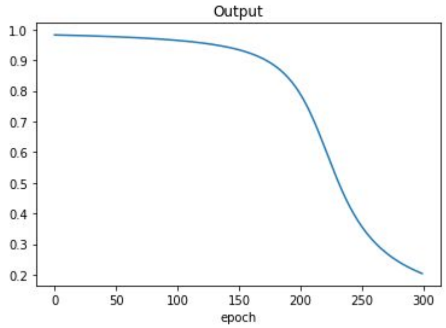
Účelová funkce - je MSE dobrá?

- Závisí na inicializace váhy a biasu. Pokud pro váhu i bias zvolíme 2, vidíme, že cca 150 epoch se síť prakticky neučí
- Jinými slovy vidíme, že čím hůř se síť chová, tím hůř se učí









Účelová funkce - je MSE dobrá?

- Závisí na inicializace váhy a biasu. Pokud pro váhu i bias zvolíme 2, vidíme, že cca 150 epoch se síť prakticky neučí
- Jinými slovy vidíme, že čím hůř se síť chová, tím hůř se učí
- Toto pozorování platí obecně

Účelová funkce - je MSE dobrá?

- Závisí na inicializace váhy a biasu. Pokud pro váhu i bias zvolíme 2, vidíme, že cca 150 epoch se síť prakticky neučí
- Jinými slovy vidíme, že čím hůř se síť chová, tím hůř se učí
- Toto pozorování platí obecně
- **Co se děje?**

Účelová funkce - je MSE dobrá?

- Závisí na inicializace váhy a biasu. Pokud pro váhu i bias zvolíme 2, vidíme, že cca 150 epoch se síť prakticky neučí
- Jinými slovy vidíme, že čím hůř se síť chová, tím hůř se učí
- Toto pozorování platí obecně
- **Co se děje?**
- Neuron se učí tím, že se mění jeho váha a bias v závislosti na parciální derivaci účelové funkce => pomalé učení je tedy ekvivalentní tomu, že $\frac{\partial C}{\partial w}$ a $\frac{\partial C}{\partial b}$ jsou malé

Účelová funkce - je MSE dobrá?

- Závisí na inicializace váhy a biasu. Pokud pro váhu i bias zvolíme 2, vidíme, že cca 150 epoch se síť prakticky neučí
- Jinými slovy vidíme, že čím hůř se síť chová, tím hůř se učí
- Toto pozorování platí obecně
- **Co se děje?**
- Neuron se učí tím, že se mění jeho váha a bias v závislosti na parciální derivaci účelové funkce => pomalé učení je tedy ekvivalentní tomu, že $\frac{\partial C}{\partial w}$ a $\frac{\partial C}{\partial b}$ jsou malé
- **Proč?**

Účelová funkce - je MSE dobrá?

- Podíváme se na předpis MSE: $C(w,b) \equiv \frac{1}{2n} \sum_x \|y(x) - a\|^2$

Účelová funkce - je MSE dobrá?

- Podíváme se na předpis MSE: $C(w,b) \equiv \frac{1}{2n} \sum_x \|y(x) - a\|^2$
- Pro síť s jedním neuronem:

$$C = (y-a)^2 = (y-\sigma(z))^2 = (y-\sigma(wx+b))^2$$

Účelová funkce - je MSE dobrá?

- Podíváme se na předpis MSE: $C(w,b) \equiv \frac{1}{2n} \sum_x \|y(x) - a\|^2$
- Pro síť s jedním neuronem:

$$C = (y-a)^2 = (y - \sigma(z))^2 = (y - \sigma(wx+b))^2$$

- Spočteme parciální derivace (přitom x=1 a y=0)

Účelová funkce - je MSE dobrá?

- Podíváme se na předpis MSE: $C(w,b) \equiv \frac{1}{2n}\sum_x \|y(x)-a\|^2$
- Pro síť s jedním neuronem:

$$C = (y-a)^2 = (y-\sigma(z))^2 = (y-\sigma(wx+b))^2$$

- Spočteme parciální derivace (přitom x=1 a y=0)

$$\frac{\partial C}{\partial w} = (a-y)\sigma'(z)x = a\sigma'(z)$$

$$\frac{\partial C}{\partial b} = (a-y)\sigma'(z) = a\sigma'(z)$$

Účelová funkce - je MSE dobrá?

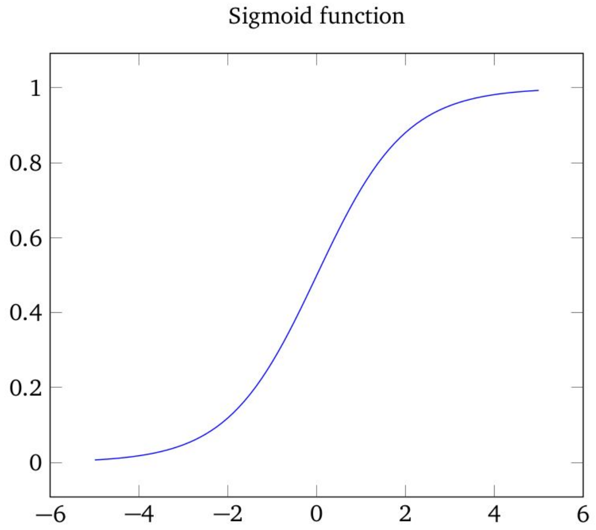
- Podíváme se na předpis MSE: $C(w,b) \equiv \frac{1}{2n}\sum_x \|y(x)-a\|^2$
- Pro síť s jedním neuronem:

$$C = (y-a)^2 = (y-\sigma(z))^2 = (y-\sigma(wx+b))^2$$

- Spočteme parciální derivace (přitom x=1 a y=0)

$$\frac{\partial C}{\partial w} = (a-y)\sigma'(z)x = a\sigma'(z)$$

$$\frac{\partial C}{\partial b} = (a-y)\sigma'(z) = a\sigma'(z)$$



Účelová funkce - je MSE dobrá?

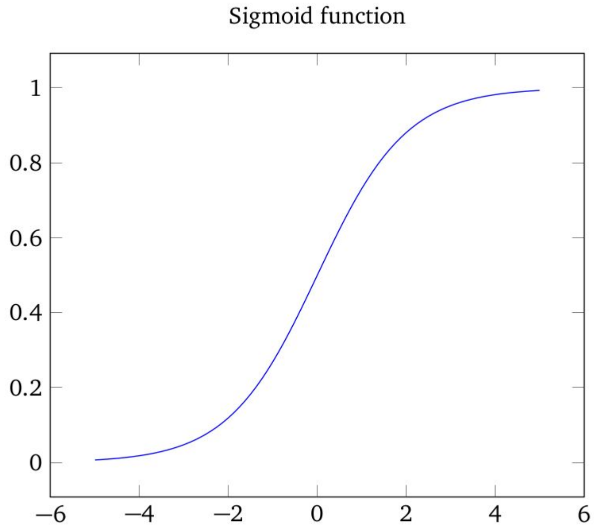
- Podíváme se na předpis MSE: $C(w,b) \equiv \frac{1}{2n}\sum_x \|y(x)-a\|^2$
- Pro síť s jedním neuronem:

$$C = (y-a)^2 = (y-\sigma(z))^2 = (y-\sigma(wx+b))^2$$

- Spočteme parciální derivace (přitom x=1 a y=0)

$$\frac{\partial C}{\partial w} = (a-y)\sigma'(z)x = a\sigma'(z)$$

$$\frac{\partial C}{\partial b} = (a-y)\sigma'(z) = a\sigma'(z)$$



- Když je output neuronu $\approx$ 1, je křivka velmi plochá

Účelová funkce - je MSE dobrá?

- To nám říká, že parciální derivace budou pro saturovaný neuron (output okolo 0 a 1) velmi malé => učení je pomalé

Účelová funkce - je MSE dobrá?

- To nám říká, že parciální derivace budou pro saturovaný neuron (output okolo 0 a 1) velmi malé => učení je pomalé
- Toto opět platí obecně

Účelová funkce - je MSE dobrá?

- To nám říká, že parciální derivace budou pro saturovaný neuron (output okolo 0 a 1) velmi malé => učení je pomalé
- Toto opět platí obecně
- **Jak to vyřešit?**

Účelová funkce - je MSE dobrá?

- To nám říká, že parciální derivace budou pro saturovaný neuron (output okolo 0 a 1) velmi malé => učení je pomalé
- Toto opět platí obecně
- **Jak to vyřešit?** Zkusíme najít účelovou funkci, tak, aby v jejích parciálních derivacích vypadla závislost na sigmoidu, tj. cost jednoho training samplu bude:

Účelová funkce - je MSE dobrá?

- To nám říká, že parciální derivace budou pro saturovaný neuron (output okolo 0 a 1) velmi malé => učení je pomalé
- Toto opět platí obecně
- **Jak to vyřešit?** Zkusíme najít účelovou funkci, tak, aby v jejích parciálních derivacích vypadla závislost na sigmoidu, tj. cost jednoho training samplu bude:

$$\frac{\partial C_{x_j}}{\partial w} = (a-y)\sigma'(z)x_j \longrightarrow (a-y)x_j$$

$$\frac{\partial C_{x_j}}{\partial b} = (a-y)\sigma'(z) \longrightarrow (a-y)$$

Účelová funkce - Hledáme novou

- Pokud najdeme účelovou funkci, která to splňuje, díky (a-y) se bude učit síť tím rychleji, čím větší bude původní chyba

$$\frac{\partial C_{x_j}}{\partial w} = (a-y)\sigma'(z)x_j \longrightarrow (a-y)x_j$$

$$\frac{\partial C_{x_j}}{\partial b} = (a-y)\sigma'(z) \longrightarrow (a-y)$$

Účelová funkce - Hledáme novou

- Pokud najdeme účelovou funkci, která to splňuje, díky (a-y) se bude učit síť tím rychleji, čím větší bude původní chyba
- Jdeme odvozovat:

Účelová funkce - Hledáme novou

- Pokud najdeme účelovou funkci, která to splňuje, díky (a-y) se bude učit síť tím rychleji, čím větší bude původní chyba
- Jdeme odvozovat:

$$\text{derivace složené funkce: } \frac{\partial C}{\partial b} = \frac{\partial C}{\partial a}\sigma'(z) \text{ a } \sigma'(z) = \sigma(z)(1-\sigma(z)) = a(1-a)$$

$$C = (y-a)^2 = (y-\sigma(z))^2 = (y-\sigma(wx+b))^2$$

Účelová funkce - Hledáme novou

- Pokud najdeme účelovou funkci, která to splňuje, díky (a-y) se bude učit síť tím rychleji, čím větší bude původní chyba
- Jdeme odvozovat:

derivace složené funkce: $\frac{\partial C}{\partial b} = \frac{\partial C}{\partial a}\sigma'(z)$ a $\sigma'(z) = \sigma(z)(1-\sigma(z)) = a(1-a)$

dosadíme: $\frac{\partial C}{\partial b} = \frac{\partial C}{\partial a}a(1-a)$, to dosadíme do našeho požadavku na parciální derivaci: $\frac{\partial C}{\partial a} = \frac{\partial C}{\partial b}\frac{1}{a(1-a)} = \frac{a-y}{a(1-a)}$

$$\frac{\partial C_{x_j}}{\partial w} = (a-y)\sigma'(z)x_j \longrightarrow (a-y)x_j$$
$$\frac{\partial C_{x_j}}{\partial b} = (a-y)\sigma'(z) \longrightarrow (a-y)$$

Účelová funkce - Hledáme novou

$$C = \int \frac{a-y}{a(1-a)} da = \int \frac{a}{a(1-a)} da - y \int \frac{1}{a(1-a)} da =$$
$$= -\ln(1-a) - y \int \frac{1}{a^2\left(\frac{1}{a} - 1\right)} da + const = \left[t = \frac{1}{a} - 1,\ dt = \frac{1}{a^2} da\right] =$$
$$= -\ln(1-a) - y \int \frac{1}{t} dt + const = -\ln(1-a) - y\ln(t) =$$
$$= -\ln(1-a) - y\ln\left(\frac{1}{a} - 1\right) + const = -\ln(1-a) - y\ln\left(\frac{1-a}{a}\right) + const =$$
$$= -\ln(1-a) - y(\ln(1-a) + y\ln(a)) + const =$$
$$= -(y\ln(a) + (1-y)\ln(1-a)) + const$$

Účelová funkce - Hledáme novou

$$C = \int \frac{a-y}{a(1-a)} da = \int \frac{a}{a(1-a)} da - y \int \frac{1}{a(1-a)} da =$$
$$= -\ln(1-a) - y \int \frac{1}{a^2\left(\frac{1}{a}-1\right)} da + const = \left[t = \frac{1}{a} - 1,\ dt = \frac{1}{a^2} da\right] =$$
$$= -\ln(1-a) - y \int \frac{1}{t} dt + const = -\ln(1-a) - y \ln(t) =$$
$$= -\ln(1-a) - y \ln\left(\frac{1}{a} - 1\right) + const = -\ln(1-a) - y \ln\left(\frac{1-a}{a}\right) + const =$$
$$= -\ln(1-a) - y(\ln(1-a) + y\ln(a)) + const =$$
$$= -(y\ln(a) + (1-y)\ln(1-a)) + const$$

To platí pro jeden training sample. Abychom dostali celou účelovou funkci, vezmeme průměr přes všechny training samply:

$$C = -\frac{1}{n} \sum_x (y\ln(a) + (1-y)\ln(1-a))) + const$$

Účelová funkce - Hledáme novou

$$C = -\frac{1}{n}\sum_x \left(y \ln(a) + (1-y)\ln(1-a)\right)) + const$$

- Konstanta je průměr konstant jednotlivých integrandů

Účelová funkce - Hledáme novou

$$C = -\frac{1}{n}\sum_x (y \ln(a) + (1-y)\ln(1-a))) + const$$

- Konstanta je průměr konstant jednotlivých integrandů
- Tato účelová funkce je jednoznačně daná (až na konstantu) našimi požadavky na parciální derivace

Účelová funkce - Cross Entropy

$$C = -\frac{1}{n}\sum_x (y \ln(a) + (1-y)\ln(1-a))) + const$$

- Konstanta je průměr konstant jednotlivých integrandů
- Tato účelová funkce je jednoznačně daná (až na konstantu) našimi požadavky na parciální derivace
- Říkáme jí cross-entropy

Účelová funkce - Cross Entropy

$$C = -\frac{1}{n}\sum_x (y \ln(a) + (1-y)\ln(1-a))) + const$$

- Konstanta je průměr konstant jednotlivých integrandů
- Tato účelová funkce je jednoznačně daná (až na konstantu) našimi požadavky na parciální derivace
- Říkáme jí cross-entropy
- Známe z teorie informace: entropie jako míra informace neboli “překvapení” získáním nového pozorování (tj. srovnání očekávané pravděpodobnosti nějaké události vs pozorování)

Účelová funkce - Cross Entropy

$$C = -\frac{1}{n} \sum_x (y \ln(a) + (1-y) \ln(1-a))) + const$$

- Konstanta je průměr konstant jednotlivých integrandů
- Tato účelová funkce je jednoznačně daná (až na konstantu) našimi požadavky na parciální derivace
- Říkáme jí cross-entropy
- Známe z teorie informace: entropie jako míra informace neboli “překvapení” získáním nového pozorování (tj. srovnání očekávané pravděpodobnosti nějaké události vs pozorování)
- konstantu zahodíme

Účelová funkce - Cross Entropy

$$C = -\frac{1}{n}\sum_{x}(y\ln{(a)} + (1-y)\ln{(1-a)}))$$

- Vyřeší cross-entropy naše problémy s pomalým učením?

Účelová funkce - Cross Entropy

$$C = -\frac{1}{n}\sum_x \left(y \ln(a) + (1-y)\ln(1-a)\right)$$

- Vyřeší cross-entropy naše problémy s pomalým učením?
- Ujistíme se, že je to vhodná účelová funkce:

Účelová funkce - Cross Entropy

$$C = -\frac{1}{n}\sum_x (y\ln(a) + (1-y)\ln(1-a)))$$

- Vyřeší cross-entropy naše problémy s pomalým učením?
- Ujistíme se, že je to vhodná účelová funkce:

$$C > 0,\ \text{protože}\ \ln(x) < 0,\ x \in (0,1)$$

$$a \to y \implies C \to 0\,(\text{pro y} \in \{0,1\},\ \text{tj. hlavně klasifikační problém})$$

Účelová funkce - Cross Entropy

$$C = -\frac{1}{n}\sum_x (y \ln(a) + (1-y)\ln(1-a)))$$

- Vyřeší cross-entropy naše problémy s pomalým učením?
- Ujistíme se, že je to vhodná účelová funkce:

$$C > 0, \text{ protože } \ln(x) < 0,\ x \in (0,1)$$
$$a \to y \implies C \to 0\,(\text{pro y} \in \{0,1\},\ \text{tj. hlavně klasifikační problém})$$

- Když se neuron chová dobře, tak:
    a. pro y=0 a a~0, zůstane jen -ln(1-a)~0
    b. pro y=1 a a~1, zůstane jen -ln(a)~0

Účelová funkce - Cross Entropy

$$C = -\frac{1}{n}\sum_{x}(y\ln(a) + (1-y)\ln(1-a)))$$

- Podíváme se na parciální derivace:

Účelová funkce - Cross Entropy

$$C = -\frac{1}{n}\sum_x (y \ln(a) + (1-y)\ln(1-a)))$$

- Podíváme se na parciální derivace:

$$\frac{\partial C}{\partial w_j} = \frac{\partial C}{\partial w_j}\left(-\frac{1}{n}\sum_x y\ln(\sigma(z(w,b))) + (1-y)\ln(1-\sigma(z(w,b)))\right) =$$

$$= -\frac{1}{n}\sum_x \left(\frac{y}{\sigma(z)} - \frac{1-y}{1-\sigma(z)}\right)\frac{\partial\sigma}{\partial w_j} = -\frac{1}{n}\sum_x \left(\frac{y}{\sigma(z)} - \frac{1-y}{1-\sigma(z)}\right)\sigma'(z)x_j =$$

$$= -\frac{1}{n}\sum_x \left(\frac{y - y\sigma(z) - \sigma(z) + y\sigma(z)}{\sigma(z)(1-\sigma(z))}\right)\sigma'(z)x_j =$$

$$= \frac{1}{n}\sum_x \left(\frac{\sigma'(z)x_j}{\sigma(z)(1-\sigma(z))}\right)(\sigma(z)-y) = \frac{1}{n}\sum_x x_j(\sigma(z)-y)$$

Účelová funkce - Cross Entropy

$$C = -\frac{1}{n}\sum_x (y \ln(a) + (1-y)\ln(1-a)))$$

- Podíváme se na parciální derivace:

$$\frac{\partial C}{\partial w_j} = \frac{\partial C}{\partial w_j}\left(-\frac{1}{n}\sum_x y\ln(\sigma(z(w,b))) + (1-y)\ln(1-\sigma(z(w,b)))\right) =$$

$$= -\frac{1}{n}\sum_x \left(\frac{y}{\sigma(z)} - \frac{1-y}{1-\sigma(z)}\right)\frac{\partial \sigma}{\partial w_j} = -\frac{1}{n}\sum_x \left(\frac{y}{\sigma(z)} - \frac{1-y}{1-\sigma(z)}\right)\sigma'(z)x_j =$$

$$= -\frac{1}{n}\sum_x \left(\frac{y - y\sigma(z) - \sigma(z) + y\sigma(z)}{\sigma(z)(1-\sigma(z))}\right)\sigma'(z)x_j =$$

$$= \frac{1}{n}\sum_x \left(\frac{\sigma'(z)x_j}{\sigma(z)(1-\sigma(z))}\right)(\sigma(z)-y) = \frac{1}{n}\sum_x x_j(\sigma(z)-y)$$

- Učení tedy závisí na chybě výstupu $\sigma(z) - y$

Účelová funkce - Cross Entropy

$$C = -\frac{1}{n}\sum_x (y \ln(a) + (1-y)\ln(1-a)))$$

- Podobně můžeme odvodit parciální derivaci dle biasu:

$$\frac{\partial C}{\partial b} = \frac{1}{n}\sum_x (\sigma(z) - y)$$

Účelová funkce - Cross Entropy

$$C = -\frac{1}{n}\sum_x \left(y \ln(a) + (1-y)\ln(1-a)\right)$$

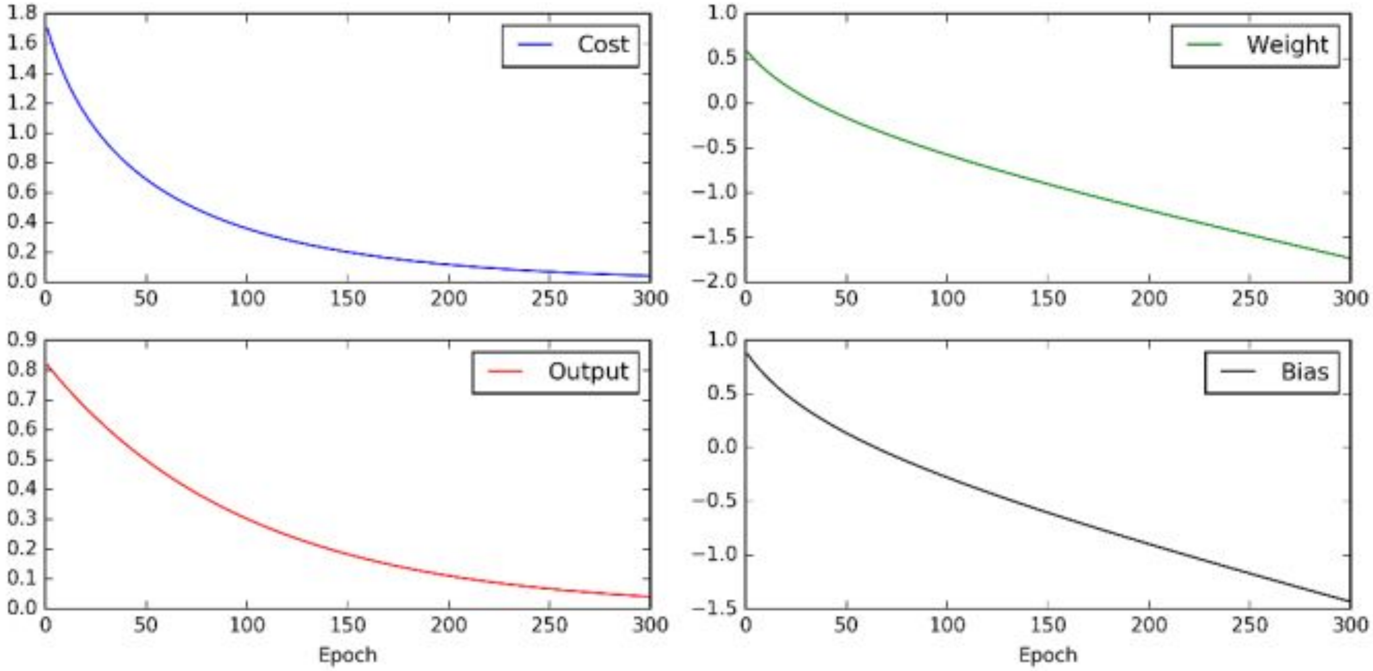
- Podobně můžeme odvodit parciální derivaci dle biasu:

$$\frac{\partial C}{\partial b} = \frac{1}{n}\sum_x (\sigma(z) - y)$$

- Zkusíme, jak se změna projeví na našem příkladu s jedním neuronem

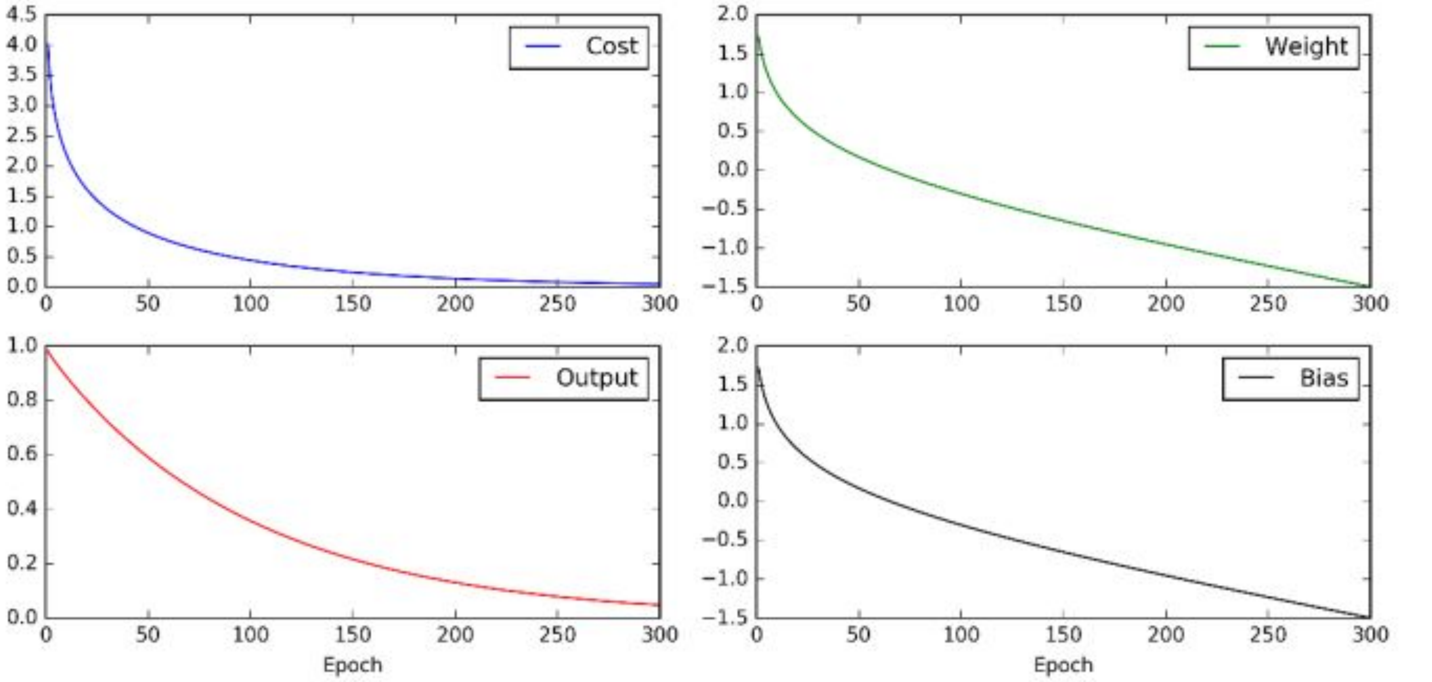
Účelová funkce - Cross Entropy

- Pro bias=0.9 a váhu=0.6



Účelová funkce - Cross Entropy

- Pro bias=2 a váhu=2



Účelová funkce - Cross Entropy

- Vypadá to dobře. K získání grafů jsme ale použili jinou learning rate, takže nic není rigorózní - nám jde ale o křivku účelové funkce

Účelová funkce - Cross Entropy

- Vypadá to dobře. K získání grafů jsme ale použili jinou learning rate, takže nic není rigorózní - nám jde ale o křivku účelové funkce
- learning rate nelze mezi účelovými funkcemi přímo porovnávat

Účelová funkce - Cross Entropy

- Vypadá to dobře. K získání grafů jsme ale použili jinou learning rate, takže nic není rigorózní - nám jde ale o křivku účelové funkce
- learning rate nelze mezi účelovými funkcemi přímo porovnávat
- Zobecníme cross entropii pro libovolnou síť:

Účelová funkce - Cross Entropy

- Vypadá to dobře. K získání grafů jsme ale použili jinou learning rate, takže nic není rigorózní - nám jde ale o křivku účelové funkce
- learning rate nelze mezi účelovými funkcemi přímo porovnávat
- Zobecníme cross entropii pro libovolnou síť:

$C = -\frac{1}{n} \sum_x \sum_j \left(y_j \ln \left(a_j^L \right) + (1 - y_j) \ln \left(1 - a_j^L \right) \right)$, $y_1, \ldots, y_n$ jsou labely pro poslední vrstvu a $a_1^L, \ldots, a_n^L$ jsou skutečné outputy

Účelová funkce - Cross Entropy

- Vypadá to dobře. K získání grafů jsme ale použili jinou learning rate, takže nic není rigorózní - nám jde ale o křivku účelové funkce
- learning rate nelze mezi účelovými funkcemi přímo porovnávat
- Zobecníme cross entropii pro libovolnou síť:

$$C = -\frac{1}{n}\sum_x \sum_j \left(y_j \ln\left(a_j^L\right) + (1-y_j)\ln\left(1-a_j^L\right)\right),\ y_1,\dots,y_n \text{ jsou labely pro poslední vrstvu}$$

a $a_1^L,\dots,a_n^L$ jsou skutečné outputy

- skoro vždy je lepší, než MSE

Cross Entropy: cvičení

- Co se stane, když v předpisu prohodíme y, a

$$C_x = -(y\ln(a) + (1-y)\ln(1-a)) \text{ vs } -(a\ln(y) + (1-a)\ln(1-y))$$

Cross Entropy: cvičení

- Co se stane, když v předpisu prohodíme y, a

$$C_x = -(y \ln(a) + (1-y)\ln(1-a)) \text{ vs } -(a \ln(y) + (1-a)\ln(1-y))$$

- není definováno, protože pro y=0 a y=1 bychom počítali logaritmus z nuly. Pro aktivaci se to nikdy nestane, díky sigmoid funkci, která nikdy nedává krajní hodnoty intervalu (0,1)

Cross Entropy: 2. cvičení

- Ukažte, že cross entropie je dobrá, i když $0 < y < 1.$

$$C_x = -(y \ln(a) + (1-y)\ln(1-a)) \text{ je minimální pro } a = y$$

Cross Entropy: 2. cvičení

- Ukažte, že cross entropie je dobrá, i když $0 < y < 1$.

$$C_x = -(y \ln(a) + (1-y)\ln(1-a))\text{ je minimální pro } a = y$$

spočteme $C'(a) = -\frac{y}{a} + \frac{1-y}{1-a}$ a položíme rovno 0:

$$\begin{aligned} -\frac{y}{a} + \frac{1-y}{1-a} = 0 &\iff \frac{y}{a} = \frac{1-y}{1-a} \\ &\iff y - ay = a - ay \\ &\iff a = y \end{aligned}$$

Cross Entropy: 2. cvičení

- Ukažte, že cross entropie je dobrá, i když $0 < y < 1$.

$$C_x = -(y \ln(a) + (1-y)\ln(1-a)) \text{ je minimální pro } a = y$$

spočteme $C'(a) = -\frac{y}{a} + \frac{1-y}{1-a}$ a položíme rovno 0:

$$-\frac{y}{a} + \frac{1-y}{1-a} = 0 \iff \frac{y}{a} = \frac{1-y}{1-a}$$
$$\iff y - ay = a - ay$$
$$\iff a = y$$

Spočteme druhou derivaci $C''(a) = \frac{y}{a^2} + \frac{1-y}{(1-a)^2}$ a protože 0<a<1 a 0<y<1 to je kladné => nalezli jsme lokální (a tedy i globální) minimum

Cross Entropy: problém

- Ve videu o backpropagation jsme dokázali, že:

  $$\frac{\partial C}{\partial w_{jk}^L} = \frac{1}{n} \sum_x a_k^{L-1}(a_j^L - y_j)\sigma'(z_j^L)$$

  přičemž derivace aktivační funkce způsobuje pomalé učení pokud je neuron saturován na nesprávné hodnotě. Dokažte, že pro cross entropii je chyba neuronu $\delta^L = a^L - y$ (pro 1 training sample). Pomocí toho dokažte, že parciální derivace C podle váhy v poslední vrstvě je

  $$\frac{\partial C}{\partial w_{jk}^L} = \frac{1}{n} \sum_x a_k^{L-1}(a_j^L - y_j)$$

Cross Entropy: problém

- Ve videu o backpropagation jsme dokázali, že:

  $$\frac{\partial C}{\partial w_{jk}^L} = \frac{1}{n}\sum_x a_k^{L-1}(a_j^L - y_j)\sigma'(z_j^L)$$

  přičemž derivace aktivační funkce způsobuje pomalé učení pokud je neuron saturován na nesprávné hodnotě. Dokažte, že pro cross entropii je chyba neuronu $\delta^L = a^L - y$ (pro 1 training sample). Pomocí toho dokažte, že parciální derivace C podle váhy v poslední vrstvě je

  $$\frac{\partial C}{\partial w_{jk}^L} = \frac{1}{n}\sum_x a_k^{L-1}(a_j^L - y_j)$$

  máme tedy dokázat, že cross entropie řeší problém pomalého učení pro libovolnou síť

Cross Entropy: problém

- **Rovnice backpropagation (pro MSE)**

1. $\delta_j^L = \frac{\partial C}{\partial a_j^L}\sigma'\left(z_j^L\right)$ zapíšeme jako $\nabla_a C \odot \sigma'\left(z^L\right)$

2. $\delta^l = \left(\left(w^{l+1}\right)^T \delta^{l+1}\right) \odot \sigma'\left(z^l\right)$, po prvcích $\delta_j^l = \sum_{k=1}^{n_{l+1}} w_{kj}^{l+1} \delta_k^{l+1} \sigma'\left(z_j^l\right)$

3. $\frac{\partial C}{\partial b_j^l} = \delta_j^l$

4. $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l$

Cross Entropy: problém

- **Rovnice backpropagation (pro MSE)**

1. $\delta_j^L = \frac{\partial C}{\partial a_j^L}\sigma'\left(z_j^L\right)$ zapíšeme jako $\nabla_a C \odot \sigma'\left(z^L\right)$

2. $\delta^l = \left(\left(w^{l+1}\right)^T \delta^{l+1}\right) \odot \sigma'\left(z^l\right)$, po prvcích $\delta_j^l = \sum_{k=1}^{n_{l+1}} w_{kj}^{l+1} \delta_k^{l+1} \sigma'\left(z_j^l\right)$

3. $\frac{\partial C}{\partial b_j^l} = \delta_j^l$

4. $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l$

$$C(w,b) \equiv \frac{1}{2n}\sum_x \|y(x) - a\|^2$$

$$\frac{\partial C_{i_j}}{\partial a^{(k)}} = a^{(k)} - y$$

Cross Entropy: problém

- **Rovnice backpropagation (pro MSE)**

1. $\delta_j^L = \frac{\partial C}{\partial a_j^L} \sigma'\left(z_j^L\right)$ zapíšeme jako $\nabla_a C \odot \sigma'\left(z^L\right)$

4. $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l$

$$C(w,b) \equiv \frac{1}{2n} \sum_x \|y(x) - a\|^2$$

$$\frac{\partial C_{i_j}}{\partial a^{(k)}} = a^{(k)} - y$$

$$\frac{\partial C_x}{\partial w_{jk}^L} = a_k^{L-1} \delta_j^L = a_k^{L-1} \frac{\partial C_x}{\partial a_j^L} \sigma'(z_j) = a_k^{L-1} \left(a_j^L - y_j\right) \sigma'(z_j)$$

$$\frac{\partial C}{\partial w_{jk}^L} = \frac{1}{n} \sum_x \frac{\partial C_x}{\partial w_{jk}^L} = \frac{1}{n} \sum_x a_k^{L-1} \left(a_j^L - y_j\right) \sigma'(z_j)$$

Cross Entropy: problém

- Musíme tedy přepočítat 1. a 4. rovnici backpropagation pro cross entropii jako účelovou funkci:

Backpropagation - 1. rovnice algoritmu: důkaz

- **Důkaz 1. :** $\delta_j^L = \frac{\partial C}{\partial a_j^L}\sigma'\left(z_j^L\right)$
- Vyjdeme z definice a aplikujeme derivaci složené funkce:

$$\delta_j^L \equiv \frac{\partial C}{\partial z_j^L} = \sum_{k=1}^{n_L} \frac{\partial C}{\partial a_k^L}\frac{\partial a_k^L}{\partial z_j^L}$$

přitom aktivace k-tého neuronu poslední vrstvy závisí pouze na $z_k^L \implies \frac{\partial a_k^L}{\partial z_{j^L}} = 0 \text{ pokud } j \neq k$

Dohromady tedy $\delta_j^L \equiv \frac{\partial C}{\partial z_j^L} = \sum_{k=1}^{n_L} \frac{\partial C}{\partial a_k^L}\frac{\partial a_k^L}{\partial z_j^L} = \frac{\partial C}{\partial a_j^L}\frac{\partial a_j^L}{\partial z_j^L}$

Dále víme, že $a_j^L = \sigma\left(z_j^L\right)$, což můžeme dosadit do předchozího vztahu

$$\delta_j^L \equiv \frac{\partial C}{\partial z_j^L} = \sum_{k=1}^{n_L} \frac{\partial C}{\partial a_k^L}\frac{\partial a_k^L}{\partial z_j^L} = \frac{\partial C}{\partial a_j^L}\frac{\partial a_j^L}{\partial z_j^L} = \frac{\partial C}{\partial a_j^L}\frac{\partial \sigma}{\partial z_j^L}\left(z_j^L\right) = \frac{\partial C}{\partial a_j^L}\sigma'\left(z_j^L\right)$$

Cross Entropy: problém

- Musíme tedy přepočítat 1. a 4. rovnici backpropagation pro cross entropii jako účelovou funkci: $C = -\frac{1}{n}\sum_x (y \ln(a) + (1-y)\ln(1-a)))$

Cross Entropy: problém

- Musíme tedy přepočítat 1. a 4. rovnici backpropagation pro cross entropii jako účelovou funkci: $C = -\frac{1}{n}\sum_x (y \ln(a) + (1-y)\ln(1-a)))$
- Vidíme, že v důkazu jsme nikde nepoužili konkrétní tvar C, takže rovnost $\delta_j^L = \frac{\partial C}{\partial a_j^L}\sigma'(z_j^L)$ platí. Spočteme $\frac{\partial C}{\partial a_j^L}$ pro 1 training sample:

Cross Entropy: problém

- Musíme tedy přepočítat 1. a 4. rovnici backpropagation pro cross entropii jako účelovou funkci: $C = -\frac{1}{n}\sum_x \left(y \ln(a) + (1-y)\ln(1-a)\right)$
- Vidíme, že v důkazu jsme nikde nepoužili konkrétní tvar C, takže rovnost $\delta_j^L = \frac{\partial C}{\partial a_j^L}\sigma'(z_j^L)$ platí. Spočteme $\frac{\partial C}{\partial a_j^L}$ pro 1 training sample:

$$\frac{\partial C_x}{\partial a_j^L} = -\left(\frac{y_j}{a_j^L} - \frac{1-y_j}{1-a_j^L}\right) = -\left(\frac{y_j}{\sigma\left(z_j^L\right)} - \frac{1-y_j}{1-\sigma\left(z_j^L\right)}\right)$$

Cross Entropy: problém

- Musíme tedy přepočítat 1. a 4. rovnici backpropagation pro cross entropii jako účelovou funkci: $C = -\frac{1}{n}\sum_x (y\ln(a) + (1-y)\ln(1-a)))$
- Vidíme, že v důkazu jsme nikde nepoužili konkrétní tvar C, takže rovnost $\delta_j^L = \frac{\partial C}{\partial a_j^L}\sigma'(z_j^L)$ platí. Spočteme $\frac{\partial C}{\partial a_j^L}$ pro 1 training sample:

$$\frac{\partial C_x}{\partial a_j^L} = -\left(\frac{y_j}{a_j^L} - \frac{1-y_j}{1-a_j^L}\right) = -\left(\frac{y_j}{\sigma\left(z_j^L\right)} - \frac{1-y_j}{1-\sigma\left(z_j^L\right)}\right)$$ dosadíme do 1. rovnice:

Cross Entropy: problém

- Musíme tedy přepočítat 1. a 4. rovnici backpropagation pro cross entropii jako účelovou funkci: $C = -\frac{1}{n}\sum_x (y \ln(a) + (1-y)\ln(1-a)))$
- Vidíme, že v důkazu jsme nikde nepoužili konkrétní tvar C, takže rovnost $\delta_j^L = \frac{\partial C}{\partial a_j^L}\sigma'(z_j^L)$ platí. Spočteme $\frac{\partial C}{\partial a_j^L}$ pro 1 training sample:

$$\frac{\partial C_x}{\partial a_j^L} = -\left(\frac{y_j}{a_j^L} - \frac{1-y_j}{1-a_j^L}\right) = -\left(\frac{y_j}{\sigma\left(z_j^L\right)} - \frac{1-y_j}{1-\sigma\left(z_j^L\right)}\right)$$ dosadíme do 1. rovnice:

$$\delta_j^L = \frac{\partial C_x}{\partial a_j^L}\sigma'(z_j^L) = -\left(\frac{y_j}{\sigma\left(z_j^L\right)} - \frac{1-y_j}{1-\sigma\left(z_j^L\right)}\right)\sigma(z_j^L)(1-\sigma(z_j^L)) = (1-y_j)\sigma(z_j^L) - (1-\sigma(z_j^L))y_j =$$
$$= \sigma(z_j^L) - y_j = a_j^L - y_j$$

Cross Entropy: problém

- $\delta_j^L = a_j^L - y_j$ dosadíme do 4. rovnice: $\dfrac{\partial C}{\partial w_{jk}^l} = a_k^{l-1}\delta_j^l$

Cross Entropy: problém

- $\delta_j^L = a_j^L - y_j$ dosadíme do 4. rovnice: $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1}\delta_j^l$

$$\frac{\partial C_x}{\partial w_{jk}^L} = a_k^{L-1}\delta_j^L = a_k^{L-1}(a_j^L - y_j) \implies \frac{\partial C}{\partial w_{jk}^L} = \frac{1}{n}\sum_x a_k^{L-1}(a_j^L - y_j)$$

Cross Entropy: problém

- $\delta_j^L = a_j^L - y_j$ dosadíme do 4. rovnice: $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1}\delta_j^l$

$$\frac{\partial C_x}{\partial w_{jk}^L} = a_k^{L-1}\delta_j^L = a_k^{L-1}(a_j^L - y_j) \implies \frac{\partial C}{\partial w_{jk}^L} = \frac{1}{n}\sum_x a_k^{L-1}(a_j^L - y_j)$$

- Pro bias stejný postup, akorát dosadíme do 3. rovnice: $\frac{\partial C}{\partial b_j^l} = \delta_j^l$

Cross Entropy: problém

- $\delta_j^L = a_j^L - y_j$ dosadíme do 4. rovnice: $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1}\delta_j^l$

$$\frac{\partial C_x}{\partial w_{jk}^L} = a_k^{L-1}\delta_j^L = a_k^{L-1}(a_j^L - y_j) \implies \frac{\partial C}{\partial w_{jk}^L} = \frac{1}{n}\sum_x a_k^{L-1}(a_j^L - y_j)$$

- Pro bias stejný postup, akorát dosadíme do 3. rovnice: $\frac{\partial C}{\partial b_j^l} = \delta_j^l$

$$\frac{\partial C_x}{\partial b_j^L} = \delta_j^L = a_j^L - y_j \implies \frac{\partial C}{\partial b_k^L} = \frac{1}{n}\sum_x (a_j^L - y_j)$$

Cross Entropy: 2. problém

- MSE a lineární neurony: máme síť s lineárními neurony v poslední vrstvě (neaplikuje se tam sigmoid), tj. $a_j^L = z_j^L$ Ukažte, že pro jeden training sample x platí $\delta^L = a^L - y.$ Dále dokažte, stejně jako v předchozím bodě, že parciální derivace C se rovnají:

$$\frac{\partial C}{\partial w_{jk}^L} = \frac{1}{n} \sum_x a_k^{L-1} (a_j^L - y_j)$$

$$\frac{\partial C}{\partial b_j^L} = \frac{1}{n} \sum_x (a_j^L - y_j).$$

Cross Entropy: 2. problém

- Dosadíme do 1. rovnice tvar naší aktivační funkce (identita):

Cross Entropy: 2. problém

- Dosadíme do 1. rovnice tvar naší aktivační funkce (identita):

$$\delta_j^L = \frac{\partial C}{\partial a_j^L} id'\left(z_j^L\right) = \frac{\partial C}{\partial a_j^L} \cdot z_j' = \frac{\partial C}{\partial a_j^L} \cdot 1 = \frac{\partial C}{\partial a_j^L} = a_j^L - y_j$$, protože

$C = \frac{1}{2}\sum_i (a_i^L - y_i)^2$. Vektorově dostáváme $\delta^L = a^L - y$

Cross Entropy: 2. problém

- Dosadíme do 1. rovnice tvar naší aktivační funkce (identita):

$$\delta_j^L = \frac{\partial C}{\partial a_j^L} id'\left(z_j^L\right) = \frac{\partial C}{\partial a_j^L} \cdot z_j' = \frac{\partial C}{\partial a_j^L} \cdot 1 = \frac{\partial C}{\partial a_j^L} = a_j^L - y_j$$, protože

$C = \frac{1}{2}\sum_i (a_i^L - y_i)^2$. Vektorově dostáváme $\delta^L = a^L - y$

- Dosadíme do 4. rovnice: $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1}\delta_j^l$

Cross Entropy: 2. problém

- Dosadíme do 1. rovnice tvar naší aktivační funkce (identita):

$$\delta_j^L = \frac{\partial C}{\partial a_j^L} id'\left(z_j^L\right) = \frac{\partial C}{\partial a_j^L} \cdot z_j' = \frac{\partial C}{\partial a_j^L} \cdot 1 = \frac{\partial C}{\partial a_j^L} = a_j^L - y_j$$, protože

$C = \frac{1}{2} \sum_i (a_i^L - y_i)^2$. Vektorově dostáváme $\delta^L = a^L - y$

- Dosadíme do 4. rovnice: $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l$

$$\frac{\partial C_x}{\partial w_{jk}^L} = a_k^{L-1} \delta_j^L = a_k^{L-1}\left(a_j^L - y_j\right) \implies \frac{\partial C}{\partial w_{jk}^L} = \frac{1}{n} \sum_x a_k^{L-1}\left(a_j^L - y_j\right)$$

Cross Entropy: 2. problém

- Pro biasy to stejné, ale na konci dosadíme do 3. rovnice: $\frac{\partial C}{\partial b_j^l} = \delta_j^l$

Cross Entropy: 2. problém

- Pro biasy to stejné, ale na konci dosadíme do 3. rovnice: $\frac{\partial C}{\partial b_j^l} = \delta_j^l$

$$\frac{\partial C_x}{\partial b_j^L} = \delta_j^L = a_j^L - y_j \implies \frac{\partial C}{\partial b_j^L} = \frac{1}{n} \sum_x a_j^L - y_j$$

Cross Entropy: 2. problém

- Pro biasy to stejné, ale na konci dosadíme do 3. rovnice: $\frac{\partial C}{\partial b_j^l} = \delta_j^l$

$$\frac{\partial C_x}{\partial b_j^L} = \delta_j^L = a_j^L - y_j \implies \frac{\partial C}{\partial b_j^L} = \frac{1}{n} \sum_x a_j^L - y_j$$

- To nám říká, že v případě lineárních neuronů výstupní vrstvy u MSE nedojde ke zpomalení učení (není to tedy jen chyba MSE, ale kombinace MSE+sigmoid). V takovém případě tedy MSE můžeme klidně použít.

Cross Entropy: 3. problém

- Díky cross entropii jsme se zbavili závislosti na derivaci aktivační funkci v rovnici $\frac{\partial C}{\partial w_j} = \frac{1}{n} \sum_x x_j(\sigma(z) - y)$
  Dokažte, že není možné zbavit se závislosti na Xj.

Cross Entropy: 3. problém

- Díky cross entropii jsme se zbavili závislosti na derivaci aktivační funkci v rovnici $\frac{\partial C}{\partial w_j} = \frac{1}{n} \sum_x x_j(\sigma(z) - y)$
  Dokažte, že není možné zbavit se závislosti na Xj.
- Při odvozování jsme použili

$$\frac{\partial C}{\partial w_j} = \frac{\partial C}{\partial \sigma} \frac{\partial \sigma}{\partial w_j}, \text{ přitom } \frac{\partial \sigma}{\partial w_j} = x_j \sigma' \left(\sum_i w_i x_i + b \right) = x_j \sigma'(z)$$

Cross Entropy: 3. problém

- Díky cross entropii jsme se zbavili závislosti na derivaci aktivační funkci v rovnici $\frac{\partial C}{\partial w_j} = \frac{1}{n}\sum_x x_j(\sigma(z) - y)$
  Dokažte, že není možné zbavit se závislosti na Xj.
- Při odvozování jsme použili
  $$\frac{\partial C}{\partial w_j} = \frac{\partial C}{\partial \sigma}\frac{\partial \sigma}{\partial w_j}, \text{ přitom } \frac{\partial \sigma}{\partial w_j} = x_j\sigma'\left(\sum_i w_i x_i + b\right) = x_j\sigma'(z)$$
- cross entropii jsme definovali tak, aby $\frac{\partial C}{\partial a} = \frac{\text{něco}}{\sigma'(z)}$

Cross Entropy: 3. problém

- Díky cross entropii jsme se zbavili závislosti na derivaci aktivační funkci v rovnici $\frac{\partial C}{\partial w_j} = \frac{1}{n} \sum_x x_j(\sigma(z) - y)$
  Dokažte, že není možné zbavit se závislosti na Xj.
- Při odvozování jsme použili

$$\frac{\partial C}{\partial w_j} = \frac{\partial C}{\partial \sigma} \frac{\partial \sigma}{\partial w_j}, \text{ přitom } \frac{\partial \sigma}{\partial w_j} = x_j \sigma' \left(\sum_i w_i x_i + b \right) = x_j \sigma'(z)$$

- cross entropii jsme definovali tak, aby $\frac{\partial C}{\partial a} = \frac{\text{něco}}{\sigma'(z)}$
- teď chceme, aby nějaká jiná účelová funkce splnila $\frac{\partial C}{\partial a} = \frac{\text{něco}}{x_j}$

Cross Entropy: 3. problém

- Díky cross entropii jsme se zbavili závislosti na derivaci aktivační funkci v rovnici $\frac{\partial C}{\partial w_j} = \frac{1}{n}\sum_x x_j(\sigma(z) - y)$
  Dokažte, že není možné zbavit se závislosti na Xj.
- Při odvozování jsme použili
  $$\frac{\partial C}{\partial w_j} = \frac{\partial C}{\partial \sigma}\frac{\partial \sigma}{\partial w_j}, \text{ přitom } \frac{\partial \sigma}{\partial w_j} = x_j\sigma'\left(\sum_i w_i x_i + b\right) = x_j\sigma'(z)$$
- cross entropii jsme definovali tak, aby $\frac{\partial C}{\partial a} = \frac{\text{něco}}{\sigma'(z)}$
- teď chceme, aby nějaká jiná účelová funkce splnila $\frac{\partial C}{\partial a} = \frac{\text{něco}}{x_j}$
- Jenže C může záviset pouze na outputu sítě (**a**) a labelu (**y**). Protože nekonečně mnoho kombinací **Xj** vede ke stejnému z a tedy i aktivaci, přínos jednotlivých **Xj** nelze vzít v potaz.