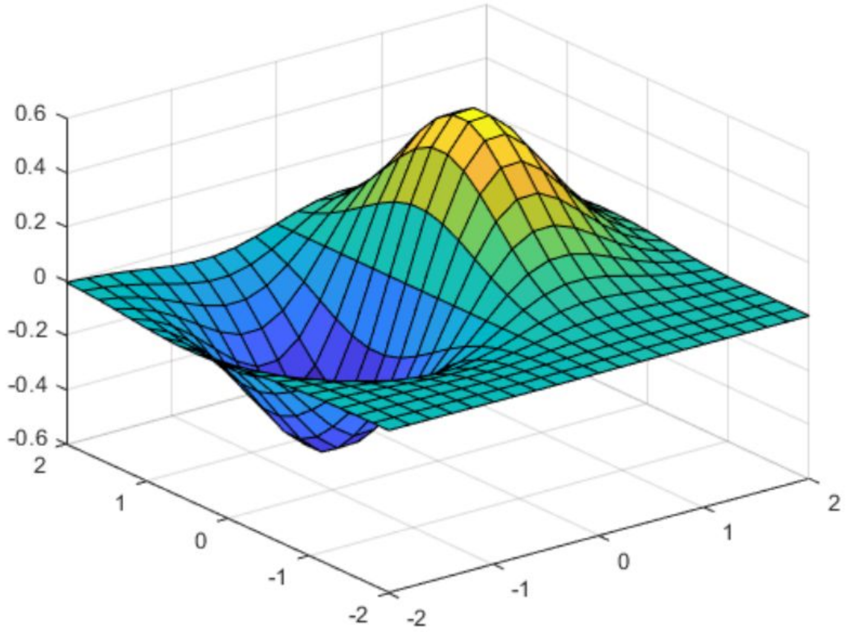
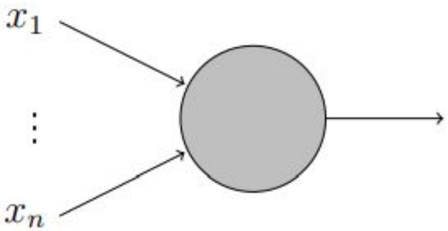
Jak se neuronové sítě učí



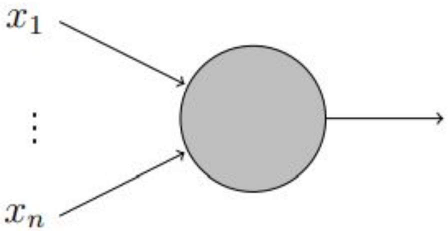
Model neuronu - spojitý neuron



- *n* vstupů, 1 výstup
- n+1 parametrů
- **Jak určit počet parametrů MLP?**

Model neuronu - spojitý neuron

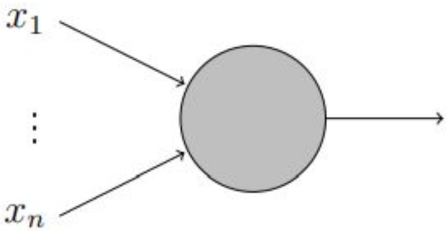
- *n* vstupů, 1 výstup
- n+1 parametrů
- **Jak určit počet parametrů MLP?**



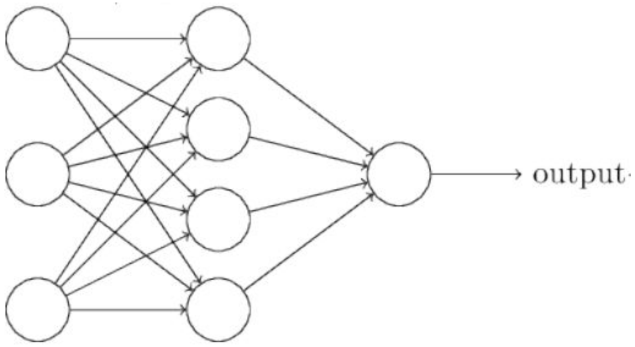
$$\sum_{i=2}^{k} N_i \cdot (N_{i-1} + 1)$$

Model neuronu - spojitý neuron

- *n* vstupů, 1 výstup
- n+1 parametrů
- **Jak určit počet parametrů MLP?**

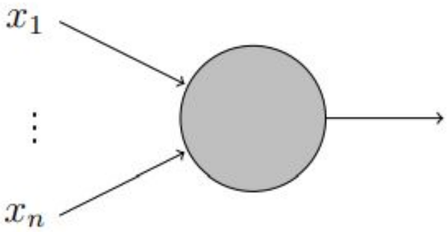


$$\sum_{i=2}^{k} N_i \cdot (N_{i-1} + 1)$$

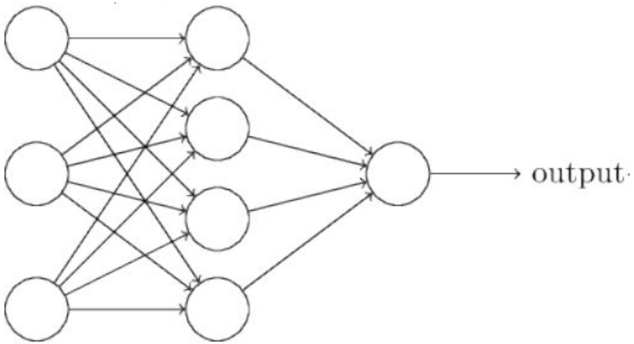


Model neuronu - spojitý neuron

- *n* vstupů, 1 výstup
- n+1 parametrů
- **Jak určit počet parametrů MLP?**



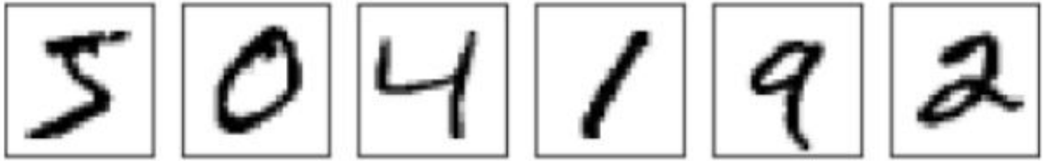
$$\sum_{i=2}^{k} N_i \cdot (N_{i-1} + 1)$$



$$4 \cdot (3 + 1) + 1 \cdot (4 + 1) = 21$$

Rozpoznávání ručně psaných číslic

- MNIST dataset - 60 000 obrázků ručně psaných číslic od 250 lidí pro trénink, 10 000 obrázků pro test psaných jinými 250 lidmi => chceme robustní systém, co rozpozná ručně psané číslice!
- vstup budeme značit jako **x**, výstup **y**



Rozpoznávání ručně psaných číslic

- Vstupní obrázek má 28x28 = 784 pixelů => vstupní vrstva 784 neuronů
- **Kolik výstupní vrstva?**

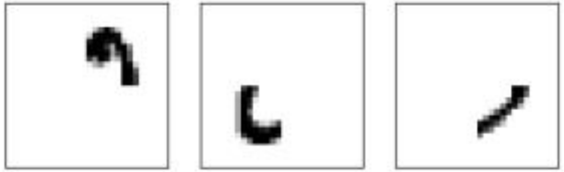
Rozpoznávání ručně psaných číslic

- Vstupní obrázek má 28x28 = 784 pixelů => vstupní vrstva 784 neuronů
- **Kolik výstupní vrstva?**
  a. **10 neuronů - 1 za každou číslici**
  b. 4 neurony - v binárním vyjádření
  c. 1 neuron - přímá hodnota

10 neuronů ve výstupní vrstvě

- Divoká fabulace: Představte si, že první neuron v předposlední vrstvě detekuje (pomocí váženého součtu předchozích neuronů) podobný tvar 

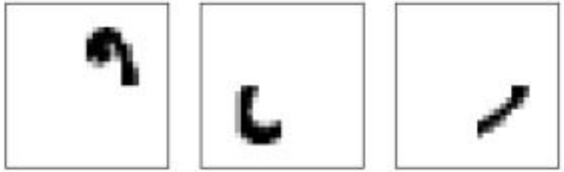
10 neuronů ve výstupní vrstvě

- Divoká fabulace: Představte si, že první neuron v předposlední vrstvě detekuje (pomocí váženého součtu předchozích neuronů) podobný tvar 
- Další neurony tyto tvary 

10 neuronů ve výstupní vrstvě

- Divoká fabulace: Představte si, že první neuron v předposlední vrstvě detekuje (pomocí váženého součtu předchozích neuronů) podobný tvar

- Další neurony tyto tvary



- Váženým součtem přes ně bychom zjistili, že tento tvar je 0

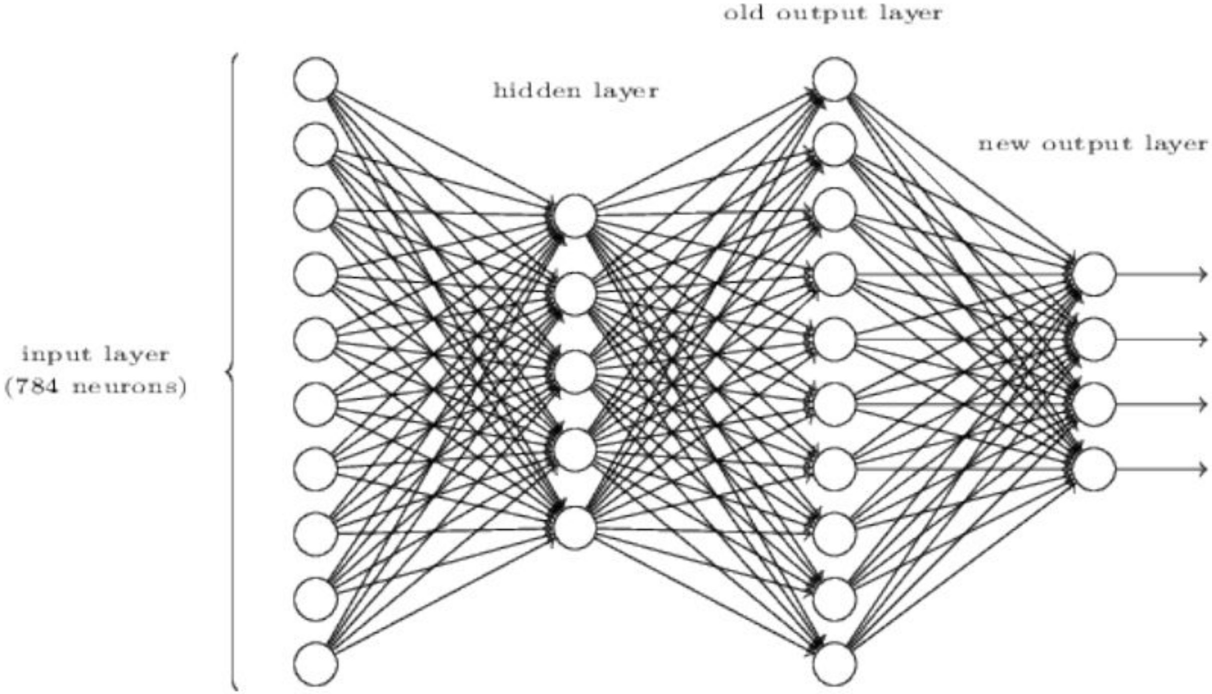
10 neuronů ve výstupní vrstvě

Jak udělat tenhle intuitivní přechod pro binární reprezentaci?

Cvičení 3

- Existuje způsob, jak zakódovat output sítě s deseti výstupními neurony do binární soustavy přidáním další vrstvy obsahující čtyři neurony. Určete jejich váhy a biasy. Předpokládejte, že správná číslice má aktivaci alespoň 0.99.

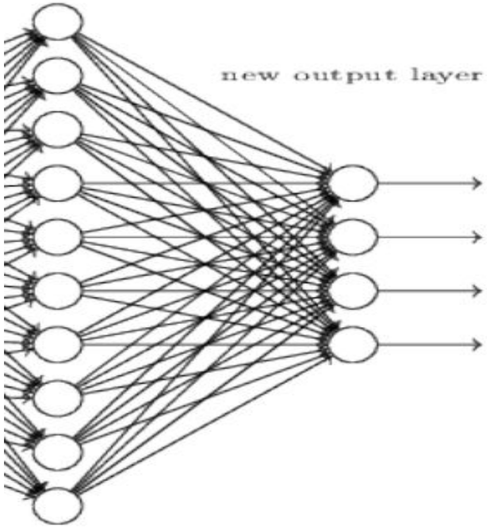
Cvičení 3


old output layer
hidden layer
new output layer
input layer
(784 neurons)

Cvičení 3

- Reprezentace čísel 0-9 v binární soustavě:

0 - 0000,
1 - 0001,
2 - 0010,
3 - 0011,
4 - 0100,
5 - 0101,
6 - 0110,
7 - 0111,
8 - 1000,
9 - 1001.



Cvičení 3

0 - 0000,	5 - 0101,
1 - 0001,	6 - 0110,
2 - 0010,	7 - 0111,
3 - 0011,	8 - 1000,
4 - 0100,	9 - 1001.

- 4 výstupní neurony - chceme zvolit váhy tak, aby např. první neuron aktivovala číslice 1, 3, 5, 7, 9 =>

Cvičení 3

0 - 0000,
1 - 0001,
2 - 0010,
3 - 0011,
4 - 0100,
5 - 0101,
6 - 0110,
7 - 0111,
8 - 1000,
9 - 1001.

- 4 výstupní neurony - chceme zvolit váhy tak, aby např. první neuron aktivovala číslice 1, 3, 5, 7, 9 =>

spojení mezi nimi přiřadíme váhu 1, ostatním 0, bias zvolíme např. 0.99

Cvičení 3

0 - 0000,
1 - 0001,
2 - 0010,
3 - 0011,
4 - 0100,
5 - 0101,
6 - 0110,
7 - 0111,
8 - 1000,
9 - 1001.

- 4 výstupní neurony - chceme zvolit váhy tak, aby např. první neuron aktivovala číslice 1, 3, 5, 7, 9 => spojení mezi nimi přiřadíme váhu 1, ostatním 0, bias zvolíme např. 0.99
- druhý neuron aktivujeme pomocí 2, 3, 6, 7 => váhy těchto spojení zvolíme 1, ostatní 0 a bias 0.99

Učení NN

- vstup **x**, výstup **y** = **y**(**x**)

Učení NN

- vstup **x**, výstup **y = y(x)**
- výstup z naší NN je deset čísel (neuronů), proto i naše **y** musí být vektor o deseti číslech se samými nulami a jednou jedničkou

Učení NN

- vstup **x**, výstup **y = y(x)**
- výstup z naší NN je deset čísel (neuronů), proto i naše **y** musí být vektor o deseti číslech se samými nulami a jednou jedničkou
- hledáme algoritmus, který nám umožní najít váhy a biasy tak, aby síť aproximovala **y(x)** pro všechna tréninková data v datasetu

Učení NN

- vstup **x**, výstup **y = y(x)**
- výstup z naší NN je deset čísel (neuronů), proto i naše **y** musí být vektor o deseti číslech se samými nulami a jednou jedničkou
- hledáme algoritmus, který nám umožní najít váhy a biasy tak, aby síť aproximovala **y(x)** pro všechna tréninková data v datasetu **=> účelová funkce**

Učení NN - účelové funkce

- účelová funkce (cost, loss or objective function)

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

Učení NN - účelové funkce

- účelová funkce (cost, loss or objective function)

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

- ***w, b*** je vektor všech vah, respektive biasů, ***n*** počet tréninkových dat, ***a*** výstup sítě, ***a = a(w,b,x)***
- quadratic cost function, mean squared error (MSE)

Učení NN - účelové funkce

- platí: $C(w,b) \geqslant 0$

Učení NN - účelové funkce

- platí: $C(w,b) \geqslant 0$

$$C(w,b) \approx 0 \iff y(x) \approx a(w,b,x)$$

Učení NN - účelové funkce

- platí: $C(w,b) \geqslant 0$

$$C(w,b) \approx 0 \iff y(x) \approx a(w,b,x)$$

$$C(w,b) \rightarrow \ min$$

Učení NN - účelové funkce

- proč účelová funkce a ne přímo počet správně klasifikovaných číslic?

Učení NN - účelové funkce

- proč účelová funkce a ne přímo počet správně klasifikovaných číslic? **Potřebujeme spojitost vůči vahám a biasům**

Učení NN - účelové funkce

- proč účelová funkce a ne přímo počet správně klasifikovaných číslic? **Potřebujeme spojitost vůči vahám a biasům**
- proč zrovna tahle účelová funkce?

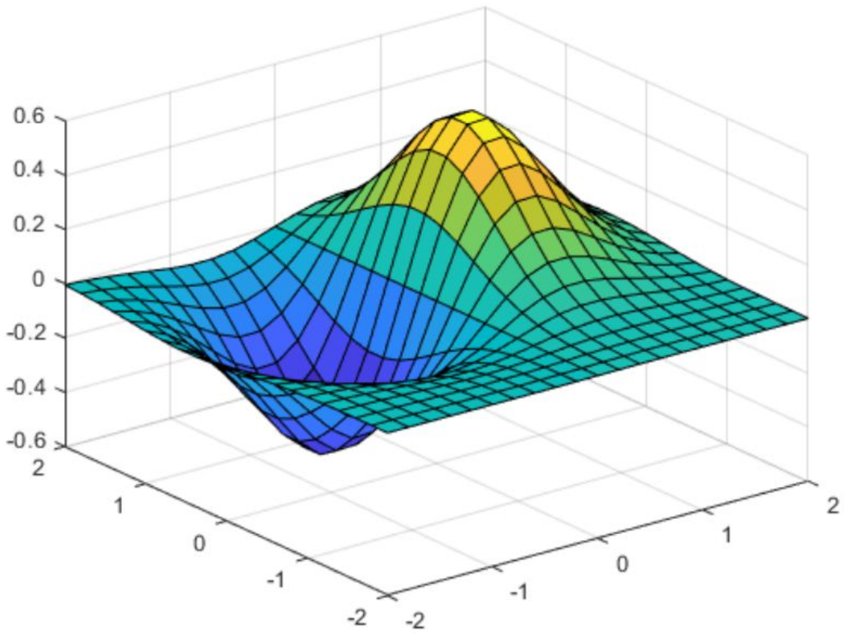
Učení NN - účelové funkce

- proč účelová funkce a ne přímo počet správně klasifikovaných číslic? **Potřebujeme spojitost vůči vahám a biasům**
- proč zrovna tahle účelová funkce? **Možností je víc a jiná účelová funkce povede k jinému učení (a následně výkonu). Prozatím neřešíme**

Učení NN - gradient descent

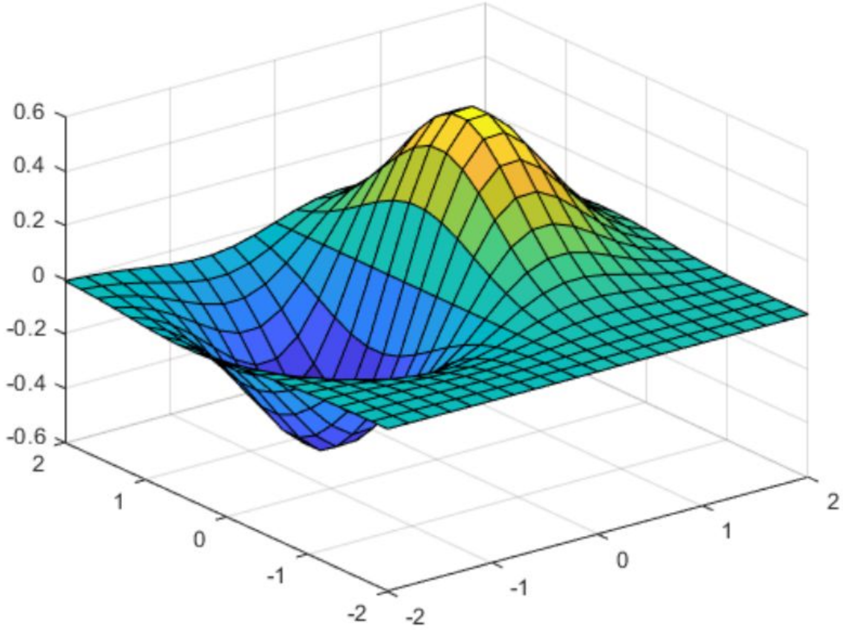
Učení NN - gradient descent

- Začneme lehčí úlohou: minimalizovat funkci dvou proměnných **f(x,y)**



Učení NN - gradient descent

- Začneme lehčí úlohou: minimalizovat funkci dvou proměnných **f(x,y)**
- **Jak tuto úlohu vyřešit?**



Učení NN - gradient descent

- analyticky to může být těžké, zvlášť budeme-li chtít postup zobecnit pro libovolný počet proměnných

Učení NN - gradient descent

- analyticky to může být těžké, zvlášť budeme-li chtít postup zobecnit pro libovolný počet proměnných
- heuristika: začneme z libovolného bodu a budeme se vždy posouvat tak, aby se hodnota **f(x,y)** snižovala (kutálející míč)

Učení NN - gradient descent

- analyticky to může být těžké, zvlášť budeme-li chtít postup zobecnit pro libovolný počet proměnných
- heuristika: začneme z libovolného bodu a budeme se vždy posouvat tak, aby se hodnota **f(x,y)** snižovala (kutálející míč)
- Takhle nalezneme alespoň lokální minimum

Učení NN - gradient descent

- Chceme zvolit Δx a Δy tak, aby Δf bylo negativní

Učení NN - gradient descent

- Chceme zvolit Δx a Δy tak, aby Δf bylo negativní
- Díky spojitosti:

$$\Delta f \approx \frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial y} \Delta y$$

Učení NN - gradient descent

- Využijeme gradient (směr nejprudšího růstu “steepest ascent”)

Učení NN - gradient descent

- Využijeme gradient (směr nejprudšího růstu “steepest ascent”)

$$\overrightarrow{\nabla f} = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$$

$$\Rightarrow$$

$$\overrightarrow{\Delta v} := \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

Učení NN - gradient descent

- Využijeme gradient (směr nejprudšího růstu "steepest ascent")

$$\overrightarrow{\nabla f} = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$$

$$\overrightarrow{\Delta v} := \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

$$\Rightarrow \quad \Delta f \approx \overrightarrow{\nabla f}^T \cdot \overrightarrow{\Delta v}.$$

Učení NN - gradient descent

- Chceme najít takové $\overrightarrow{\Delta v}$, aby byla $\Delta f \approx \overrightarrow{\nabla f}^T \cdot \overrightarrow{\Delta v}.$ negativní bez závislosti na znaménku gradientu

Učení NN - gradient descent

- Chceme najít takové $\overrightarrow{\Delta v}$, aby byla $\Delta f \approx \overrightarrow{\nabla f}^T \cdot \overrightarrow{\Delta v}.$ negativní bez závislosti na znaménku gradientu
- volba $\overrightarrow{\Delta v} := -\xi \overrightarrow{\nabla f}$, tzv. *learning rate*

Učení NN - gradient descent

- Chceme najít takové $\overrightarrow{\Delta v}$, aby byla $\Delta f \approx \overrightarrow{\nabla f}^T \cdot \overrightarrow{\Delta v}.$ negativní bez závislosti na znaménku gradientu
- volba $\overrightarrow{\Delta v} := -\xi \overrightarrow{\nabla f}$, tzv. *learning rate*

$$\Delta f \approx \overrightarrow{\nabla f}^T \cdot \overrightarrow{\Delta v} = -\overrightarrow{\nabla f}^T \cdot \xi \overrightarrow{\nabla f} = -\xi \left\| \overrightarrow{\nabla f} \right\|^2$$

Učení NN - gradient descent

- Chceme najít takové $\overrightarrow{\Delta v}$, aby byla $\Delta f \approx \overrightarrow{\nabla f}^T \cdot \overrightarrow{\Delta v}.$ negativní bez závislosti na znaménku gradientu
- volba $\overrightarrow{\Delta v} := -\xi \overrightarrow{\nabla f}$, tzv. *learning rate*

$$\Delta f \approx \overrightarrow{\nabla f}^T \cdot \overrightarrow{\Delta v} = -\overrightarrow{\nabla f}^T \cdot \xi \overrightarrow{\nabla f} = -\xi \left\| \overrightarrow{\nabla f} \right\|^2$$

- Pravidlo pro změnu **x** a **y** $\vec{v} \rightarrow \vec{\tilde{v}} = \vec{v} - \xi \overrightarrow{\nabla f}$

Shrnutí

$$\Delta f \approx \frac{\partial f}{\partial x}\Delta x + \frac{\partial f}{\partial y}\Delta y$$

Shrnutí

$$\Delta f \approx \frac{\partial f}{\partial x}\Delta x + \frac{\partial f}{\partial y}\Delta y$$

$$\overrightarrow{\nabla f} = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$$

$$\overrightarrow{\Delta v} := \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

Shrnutí

$$\Delta f \approx \frac{\partial f}{\partial x}\Delta x + \frac{\partial f}{\partial y}\Delta y$$

$$\Delta f \approx \overrightarrow{\nabla f}^T \cdot \overrightarrow{\Delta v}.$$

$$\overrightarrow{\nabla f} = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$$

$$\overrightarrow{\Delta v} := \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

Shrnutí

$$\overrightarrow{\nabla f} = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$$

$$\Delta f \approx \frac{\partial f}{\partial x}\Delta x + \frac{\partial f}{\partial y}\Delta y$$

$$\overrightarrow{\Delta v} := \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

$$\Delta f \approx \overrightarrow{\nabla f}^T \cdot \overrightarrow{\Delta v}.$$

$$\overrightarrow{\Delta v} := -\xi \overrightarrow{\nabla f}$$

Shrnutí

$$\Delta f \approx \frac{\partial f}{\partial x}\Delta x + \frac{\partial f}{\partial y}\Delta y$$

$$\overrightarrow{\nabla f} = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$$

$$\Delta f \approx \overrightarrow{\nabla f}^T \cdot \overrightarrow{\Delta v}.$$

$$\overrightarrow{\Delta v} := \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

$$\overrightarrow{\Delta v} := -\xi \overrightarrow{\nabla f}$$

$$\Delta f \approx \overrightarrow{\nabla f}^T \cdot \overrightarrow{\Delta v} = -\overrightarrow{\nabla f}^T \cdot \xi \overrightarrow{\nabla f} = -\xi \left\| \overrightarrow{\nabla f} \right\|^2$$

Shrnutí

$$\overrightarrow{\nabla f} = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$$

$$\overrightarrow{\Delta v} := \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

$$\Delta f \approx \frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial y} \Delta y$$

$$\Delta f \approx \overrightarrow{\nabla f}^T \cdot \overrightarrow{\Delta v}.$$

$$\overrightarrow{\Delta v} := -\xi \overrightarrow{\nabla f}$$

$$\Delta f \approx \overrightarrow{\nabla f}^T \cdot \overrightarrow{\Delta v} = -\overrightarrow{\nabla f}^T \cdot \xi \overrightarrow{\nabla f} = -\xi \left\| \overrightarrow{\nabla f} \right\|^2$$

$$\vec{v} \rightarrow \tilde{\vec{v}} = \vec{v} - \xi \overrightarrow{\nabla f}$$

Učení NN - gradient descent

- Zobecníme algoritmus pro funkci **n** parametrů

$$\overrightarrow{\nabla f} = \left(\frac{\partial f}{\partial v_1}, \ldots, \frac{\partial f}{\partial v_n}\right)^T$$

Učení NN - gradient descent

- Zobecníme algoritmus pro funkci **n** parametrů
$$\overrightarrow{\nabla f} = \left(\frac{\partial f}{\partial v_1}, \ldots, \frac{\partial f}{\partial v_n}\right)^T$$
- Algoritmus můžeme použít na naší účelovou funkci $C(w, b) \equiv \frac{1}{2n} \sum_x \|y(x) - a\|^2$

Učení NN - gradient descent

- Zobecníme algoritmus pro funkci **n** parametrů

$$\overrightarrow{\nabla f} = \left(\frac{\partial f}{\partial v_1}, \ldots, \frac{\partial f}{\partial v_n}\right)^T$$

- Algoritmus můžeme použít na naší účelovou funkci $C(w,b) \equiv \frac{1}{2n} \sum_x \|y(x) - a\|^2$
- Všimněme si, že má tvar $C = \frac{1}{n} \sum_{i=1}^n C_i$
- Musíme spočítat gradient přes všechna tréninková data $\overrightarrow{\nabla C} = \frac{1}{n} \sum_{i=1}^n \overrightarrow{\nabla C_i}$

Učení NN - stochastic gradient descent

- Vylepšení: stochastic gradient descent
- Použijeme aproximaci pomocí *mini-batch*, náhodný výběr **m** tréninkových dat

Učení NN - stochastic gradient descent

- Vylepšení: stochastic gradient descent
- Použijeme aproximaci pomocí *mini-batch*, náhodný výběr **m** tréninkových dat

$$\overrightarrow{\nabla C} = \frac{1}{n}\sum_{i=1}^{n}\overrightarrow{\nabla C_i} \approx \frac{1}{m}\sum_{j=1}^{m}\overrightarrow{\nabla C_{i_j}},\ i_j$$ je náhodný index dat

Učení NN - gradient descent

- Pravidlo pro update vah a biasu: $\vec{w} = (w_1, ..., w_p)^T$,

$\vec{b} = (b_1, ..., b_q)^T$, $C = C(\vec{w}, \vec{b})$, $\nabla C_{i_j} = \left(\frac{\partial C_{i_j}}{\partial w_1}, \ldots, \frac{\partial C_{i_j}}{\partial p} \right)$ a $\nabla C_{i_j}^{(k)}$ je k-tá složka gradientu

Učení NN - gradient descent

- Pravidlo pro update vah a biasu: $\vec{w} = (w_1, ..., w_p)^T,$

$\vec{b} = (b_1, ..., b_q)^T,$ $C = C(\vec{w}, \vec{b}),$ $\nabla C_{i_j} = \left(\frac{\partial C_{i_j}}{\partial w_1}, \ldots, \frac{\partial C_{i_j}}{\partial p} \right)$ a $\nabla C_{i_j}^{(k)}$ je k-tá složka gradientu

$$w_k \to \tilde{w_k} = w_k - \frac{\xi}{m} \nabla C_{i_j}^{(k)} = w_k - \frac{\xi}{m} \sum_{j=1}^{m} \frac{\partial C_{i_j}}{\partial w_k}$$

$$b_l \to \tilde{b_l} = w_k - \frac{\xi}{m} \nabla C_{i_j}^{(l)} = b_l - \frac{\xi}{m} \sum_{j=1}^{m} \frac{\partial C_{i_j}}{\partial b_l},$$

Cvičení 4

- Chceme udělat krok Δv, přičemž $\|\Delta v\| = \epsilon,\ \epsilon > 0$, tak, abychom minimalizovali $\Delta C \approx \nabla C \cdot \Delta v$. Dokažte, že $\Delta v := -\xi \nabla C,\ \xi = \dfrac{\epsilon}{\|\nabla C\|}$

Cvičení 4

- Chceme udělat krok Δv, přičemž $\|\Delta v\| = \epsilon,\ \epsilon > 0$, tak, abychom minimalizovali $\Delta C \approx \nabla C \cdot \Delta v$. Dokažte, že $\Delta v := -\xi \nabla C,\ \xi = \frac{\epsilon}{\|\nabla C\|}$
- Jinými slovy dokažte, že “pohyb” ve směru gradientu je optimální krok (tj. vede k největšímu poklesu účelové funkce). Pozor, jen v lokální lineární aproximaci

Cvičení 4

$\|\Delta v\| = \epsilon$ $\quad \epsilon > 0$ $\quad \Delta C \approx \nabla C \cdot \Delta v.$

Nápověda: Cauchy-Schwarzova nerovnost

Cvičení 4

$$\|\Delta v\| = \epsilon \qquad \epsilon > 0 \qquad \Delta C \approx \nabla C \cdot \Delta v.$$

Nápověda: Cauchy-Schwarzova nerovnost

$$-\|u\|\|v\| \leqslant |\langle u, v\rangle| \leqslant \|u\|\|v\|,$$
$$|\langle u, v\rangle| = \|u\|\|v\| \iff u = \alpha \cdot v,\ \alpha \in \mathbb{R}$$

Cvičení 4

$$\|\Delta v\| = \epsilon \qquad \epsilon > 0 \qquad \Delta C \approx \nabla C \cdot \Delta v.$$

Nápověda: Cauchy-Schwarzova nerovnost

$$-\|u\|\|v\| \leqslant |\langle u, v\rangle| \leqslant \|u\|\|v\|,$$
$$|\langle u, v\rangle| = \|u\|\|v\| \iff u = \alpha \cdot v,\ \alpha \in \mathbb{R}$$

Dosadíme aproximaci účelové funkce do CS nerovnosti:

Cvičení 4

$$\|\Delta v\| = \epsilon \qquad \epsilon > 0 \qquad \Delta C \approx \nabla C \cdot \Delta v.$$

Nápověda: Cauchy-Schwarzova nerovnost

$$-\|u\|\|v\| \leqslant |\langle u, v\rangle| \leqslant \|u\|\|v\|,$$
$$|\langle u, v\rangle| = \|u\|\|v\| \iff u = \alpha \cdot v,\ \alpha \in \mathbb{R}$$

Dosadíme aproximaci účelové funkce do CS nerovnosti:

$$-\|\nabla C\| \cdot \|\Delta v\| = -\|\nabla C\| \cdot \epsilon \leq |\langle \nabla C, \Delta v\rangle| = |\nabla C \cdot \Delta v| \leq \|\nabla C\| \cdot \|\Delta v\| = \|\nabla C\| \cdot \epsilon$$

Cvičení 4

$$\|\Delta v\| = \epsilon \qquad \epsilon > 0 \qquad \Delta C \approx \nabla C \cdot \Delta v.$$

Nápověda: Cauchy-Schwarzova nerovnost

$$-\|u\|\|v\| \leqslant |\langle u, v\rangle| \leqslant \|u\|\|v\|,$$
$$|\langle u, v\rangle| = \|u\|\|v\| \iff u = \alpha \cdot v,\ \alpha \in \mathbb{R}$$

Dosadíme aproximaci účelové funkce do CS nerovnosti:

$$-\|\nabla C\| \cdot \|\Delta v\| = -\|\nabla C\| \cdot \epsilon \leq |\langle \nabla C, \Delta v\rangle| = |\nabla C \cdot \Delta v| \leq \|\nabla C\| \cdot \|\Delta v\| = \|\nabla C\| \cdot \epsilon$$

Chceme rovnost, tedu musí platit $\Delta v = \alpha \nabla C,\ \alpha \in \mathbb{R}$

Cvičení 4

$$\|\Delta v\| = \epsilon \qquad \epsilon > 0 \qquad \Delta C \approx \nabla C \cdot \Delta v.$$

Nápověda: Cauchy-Schwarzova nerovnost

$$-\|u\|\|v\| \leqslant |\langle u, v\rangle| \leqslant \|u\|\|v\|,$$
$$|\langle u, v\rangle| = \|u\|\|v\| \iff u = \alpha \cdot v,\ \alpha \in \mathbb{R}$$

Dosadíme aproximaci účelové funkce do CS nerovnosti:

$$-\|\nabla C\| \cdot \|\Delta v\| = -\|\nabla C\| \cdot \epsilon \leq |\langle \nabla C, \Delta v\rangle| = |\nabla C \cdot \Delta v| \leq \|\nabla C\| \cdot \|\Delta v\| = \|\nabla C\| \cdot \epsilon$$

Chceme rovnost, tedu musí platit $\Delta v = \alpha \nabla C,\ \alpha \in \mathbb{R}$

$$\|\nabla C\| \cdot \|\Delta v\| = \|\nabla C\| \cdot \|\alpha \cdot \nabla C\| = |\alpha| \cdot \|\nabla C\|^2 \implies \|\nabla C\| \cdot \epsilon = |\alpha| \cdot \|\nabla C\|^2$$

Cvičení 4

$$\|\Delta v\| = \epsilon \qquad \epsilon > 0 \qquad \Delta C \approx \nabla C \cdot \Delta v.$$

Nápověda: Cauchy-Schwarzova nerovnost

$$-\|u\|\|v\| \leqslant |\langle u, v\rangle| \leqslant \|u\|\|v\|,$$
$$|\langle u, v\rangle| = \|u\|\|v\| \iff u = \alpha \cdot v,\ \alpha \in \mathbb{R}$$

Dosadíme aproximaci účelové funkce do CS nerovnosti:

$$-\|\nabla C\| \cdot \|\Delta v\| = -\|\nabla C\| \cdot \epsilon \leq |\langle \nabla C, \Delta v\rangle| = |\nabla C \cdot \Delta v| \leq \|\nabla C\| \cdot \|\Delta v\| = \|\nabla C\| \cdot \epsilon$$

Chceme rovnost, tedu musí platit $\Delta v = \alpha \nabla C,\ \alpha \in \mathbb{R}$

$$\|\nabla C\| \cdot \|\Delta v\| = \|\nabla C\| \cdot \|\alpha \cdot \nabla C\| = |\alpha| \cdot \|\nabla C\|^2 \implies \|\nabla C\| \cdot \epsilon = |\alpha| \cdot \|\nabla C\|^2$$
$$\epsilon = |\alpha| \cdot \|\nabla C\| \iff \pm\alpha = \frac{\epsilon}{\|\nabla C\|} \implies \Delta v = \pm\frac{\epsilon}{\|\nabla C\|} \cdot \nabla C$$

Cvičení 5

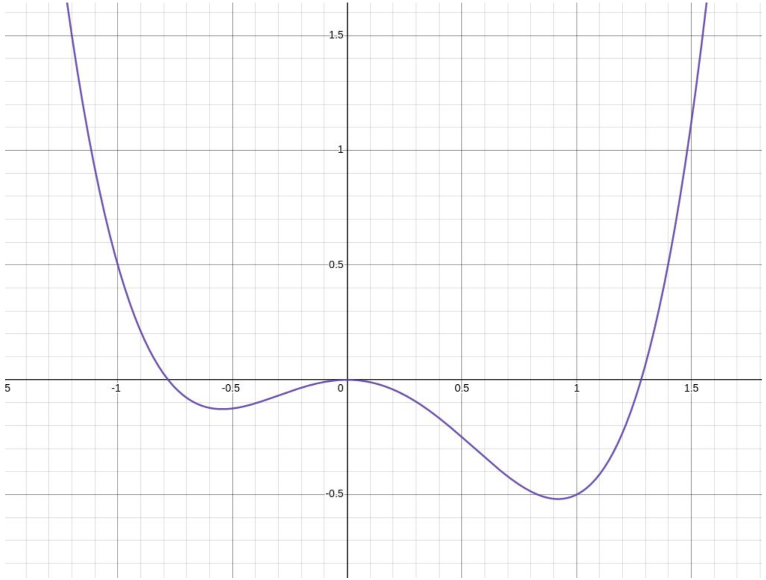
- Jak to bude v případě funkce jedné proměnné?

Cvičení 5

- Jak to bude v případě funkce jedné proměnné?

$$\nabla C(x) = C'(x) \text{ a zároveň } \Delta v(x) = -\frac{\epsilon}{\|\nabla C(x)\|}\nabla C(x) = -\frac{\epsilon \cdot C'(x)}{\|C'(x)\|}$$

pokud $C'(x) = 0$, jsme v lokálním minimu, pokud $C'(x) > 0$, $\Delta v = -\epsilon$ a v opačném případě $\Delta v = \epsilon$



Cvičení 6

- Jaký výhody a nevýhody má online learning (batch velikosti 1)?

Cvičení 6

- Jaký výhody a nevýhody má online learning (batch velikosti 1)?
  a. Výhody: učení je rychlé

Cvičení 6

- Jaký výhody a nevýhody má online learning (batch velikosti 1)?
  a. Výhody: učení je rychlé
  b. Nevýhody: nepřesnost aproximace gradientu, i extrémy v datech mají vliv na úpravu parametrů sítě

Jenže jak spočítáme gradient přes všechny váhy a biasy?