

Neuronové sítě
(10. přednáška)

O co jde?

O co jde?

Máme **model výpočtu**

O co jde?

Máme **model výpočtu** (t.j. výpočetní postup jednoznačně daný vstupy a nějakými parametry)

O co jde?

Máme **model výpočtu** (t.j. výpočetní postup jednoznačně daný vstupy a nějakými parametry), chceme najít vhodné nastavení parametrů, aby postup (model) dával řešení našeho problému.

O co jde?

Máme **model výpočtu** (t.j. výpočetní postup jednoznačně daný vstupy a nějakými parametry), chceme najít vhodné nastavení parametrů, aby postup (model) dával řešení našeho problému.

Při strojovém **učení s učitelem** zvolíme dostatečně silný model a nastavení parametrů chceme nalézt automaticky na základě vzorových řešení daného problému.

Slogan

Chceme, aby se stroj naučil řešit zadaný problém na základě vzorových řešení

Slogan

Chceme, aby se stroj naučil řešit zadaný problém na základě vzorových řešení

Proč?

Slogan

Chceme, aby se stroj naučil řešit zadaný problém na základě vzorových řešení

Proč?

- řešení je příliš komplikované
- problém se často mění, vyvíjí
- lidská práce je drahá (v porovnání se strojovou)

- Rozpoznávání vzorců
 - věci na fotkách
 - osoby na fotkách
 - výrazy tváře
 - mluvená slova

- Rozpoznávání vzorců
 - věci na fotkách
 - osoby na fotkách
 - výrazy tváře
 - mluvená slova
- Rozpoznávání anomálií
 - netypické sekvence finančních transakcí
 - netypická data přicházející ze senzorů v atomové elektrárně

- Rozpoznávání vzorců
 - věci na fotkách
 - osoby na fotkách
 - výrazy tváře
 - mluvená slova
- Rozpoznávání anomálií
 - netypické sekvence finančních transakcí
 - netypická data přicházející ze senzorů v atomové elektrárně
- Předpovídání
 - vývoj ceny akcií na burze / vývoj měnového kurzu
 - jaké filmy bude mít daný člověk rád

Výpočetní modely

Výpočetní modely

regrese

Výpočetní modely

regrese proložení funkce danými body:

Výpočetní modely

regrese proložení funkce danými body:

- *lineární regrese*, hledání optima pomocí

Výpočetní modely

regrese proložení funkce danými body:

- *lineární regrese*, hledání optima pomocí *klesání podle gradientu* nebo

Výpočetní modely

regrese proložení funkce danými body:

- *lineární regrese*, hledání optima pomocí *klesání podle gradientu* nebo *genetických algoritmů*

Výpočetní modely

regrese proložení funkce danými body:

- *lineární regrese*, hledání optima pomocí *klesání podle gradientu* nebo *genetických algoritmů*
- *rozhodovací stromy*

Výpočetní modely

regrese proložení funkce danými body:

- *lineární regrese*, hledání optima pomocí *klesání podle gradientu* nebo *genetických algoritmů*
- *rozhodovací stromy*

klasifikace

Výpočetní modely

regrese proložení funkce danými body:

- *lineární regrese*, hledání optima pomocí *klesání podle gradientu* nebo *genetických algoritmů*
- *rozhodovací stromy*

klasifikace přiřazení vzorku do některé z konečné množiny tříd:

Výpočetní modely

regrese proložení funkce danými body:

- *lineární regrese*, hledání optima pomocí *klesání podle gradientu* nebo *genetických algoritmů*
- *rozhodovací stromy*

klasifikace přiřazení vzorku do některé z konečné množiny tříd:

- *rozhodovací stromy*

Výpočetní modely

regrese proložení funkce danými body:

- *lineární regrese*, hledání optima pomocí *klesání podle gradientu* nebo *genetických algoritmů*
- *rozhodovací stromy*

klasifikace přiřazení vzorku do některé z konečné množiny tříd:

- rozhodovací stromy
- metoda k nejbližších sousedů

Výpočetní modely

regrese proložení funkce danými body:

- *lineární regrese*, hledání optima pomocí *klesání podle gradientu* nebo *genetických algoritmů*
- *rozhodovací stromy*

klasifikace přiřazení vzorku do některé z konečné množiny tříd:

- rozhodovací stromy
- metoda k nejbližších sousedů
- Bayesovské sítě

Výpočetní modely

regrese proložení funkce danými body:

- *lineární regrese*, hledání optima pomocí *klesání podle gradientu* nebo *genetických algoritmů*
- *rozhodovací stromy*

klasifikace přiřazení vzorku do některé z konečné množiny tříd:

- rozhodovací stromy
- metoda k nejbližších sousedů
- Bayesovské sítě
- Support Vector Machines

Výpočetní modely

regrese proložení funkce danými body:

- *lineární regrese*, hledání optima pomocí *klesání podle gradientu* nebo *genetických algoritmů*
- *rozhodovací stromy*

klasifikace přiřazení vzorku do některé z konečné množiny tříd:

- rozhodovací stromy
- metoda k nejbližších sousedů
- Bayesovské sítě
- Support Vector Machines
- neuronové sítě

Výpočetní modely

regrese proložení funkce danými body:

- *lineární regrese*, hledání optima pomocí *klesání podle gradientu* nebo *genetických algoritmů*
- *rozhodovací stromy*

klasifikace přiřazení vzorku do některé z konečné množiny tříd:

- rozhodovací stromy
- metoda k nejbližších sousedů
- Bayesovské sítě
- Support Vector Machines
- neuronové sítě

shlukování

Výpočetní modely

regrese proložení funkce danými body:

- *lineární regrese*, hledání optima pomocí *klesání podle gradientu* nebo *genetických algoritmů*
- *rozhodovací stromy*

klasifikace přiřazení vzorku do některé z konečné množiny tříd:

- rozhodovací stromy
- metoda k nejbližších sousedů
- Bayesovské sítě
- Support Vector Machines
- neuronové sítě

shlukování *k-means algoritmus*

snížení dimenze *Principal Component Analysis*

Proč neuronové sítě?

- studium mozku

Proč neuronové sítě?

- studium mozku
- lidského myšlení

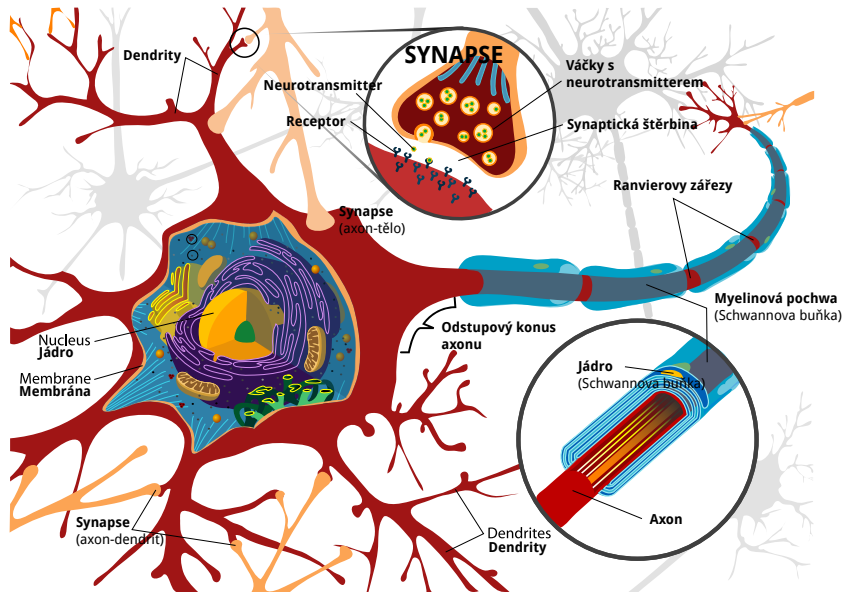
Proč neuronové sítě?

- studium mozku
- lidského myšlení
- zajímavý výpočetní model

Proč neuronové sítě?

- studium mozku
- lidského myšlení
- zajímavý výpočetní model

Biologie neuronu



Matematický neuron — $N : X_1 \times \cdots \times X_n \rightarrow Y$

Matematický neuron — $N : X_1 \times \cdots \times X_n \rightarrow Y$

- Lineární neurony $N(X) = w_0 + \sum_{i=1}^n w_i x_i$
 - $N(x)$ tedy je **aktivace neuronu** (z)

Matematický neuron — $N : X_1 \times \dots \times X_n \rightarrow Y$

- Lineární neurony $N(X) = w_0 + \sum_{i=1}^n w_i x_i$
 - $N(x)$ tedy je **aktivace neuronu** (z)
- Binární prahové neurony (**perceptrony**)

Matematický neuron — $N : X_1 \times \cdots \times X_n \rightarrow Y$

- Lineární neurony $N(X) = w_0 + \sum_{i=1}^n w_i x_i$
 - $N(x)$ tedy je **aktivace neuronu** (z)
- Binární prahové neurony (**perceptrony**)
 - McCulloch & Pitts, 1943

Matematický neuron — $N : X_1 \times \dots \times X_n \rightarrow Y$

- Lineární neurony $N(X) = w_0 + \sum_{i=1}^n w_i x_i$
 - $N(x)$ tedy je **aktivace neuronu** (z)
- Binární prahové neurony (**perceptrony**)
 - McCulloch & Pitts, 1943
 - je-li aktivace nižší než práh, vrací 0, jinak vrací 1

Matematický neuron — $N : X_1 \times \dots \times X_n \rightarrow Y$

- Lineární neurony $N(X) = w_0 + \sum_{i=1}^n w_i x_i$
 - $N(x)$ tedy je **aktivace neuronu** (z)
- Binární prahové neurony (**perceptrony**)
 - McCulloch & Pitts, 1943
 - je-li aktivace nižší než práh, vrací 0, jinak vrací 1
 - spike (akční potenciál) \approx pravdivostní hodnota výroku, neuron počítá pravdivostní hodnotu jiného výroku ze vstupních výroků

Matematický neuron — $N : X_1 \times \dots \times X_n \rightarrow Y$

- Lineární neurony $N(X) = w_0 + \sum_{i=1}^n w_i x_i$
 - $N(x)$ tedy je **aktivace neuronu** (z)
- Binární prahové neurony (**perceptrony**)
 - McCulloch & Pitts, 1943
 - je-li aktivace nižší než práh, vrací 0, jinak vrací 1
 - spike (akční potenciál) \approx pravdivostní hodnota výroku, neuron počítá pravdivostní hodnotu jiného výroku ze vstupních výroků
- Lineární prahové neurony

Matematický neuron — $N : X_1 \times \dots \times X_n \rightarrow Y$

- Lineární neurony $N(X) = w_0 + \sum_{i=1}^n w_i x_i$
 - $N(x)$ tedy je **aktivace neuronu** (z)
- Binární prahové neurony (**perceptrony**)
 - McCulloch & Pitts, 1943
 - je-li aktivace nižší než práh, vrací 0, jinak vrací 1
 - spike (akční potenciál) \approx pravdivostní hodnota výroku, neuron počítá pravdivostní hodnotu jiného výroku ze vstupních výroků
- Lineární prahové neurony
 - po práh vrací 0, nad ním nějakou lineární funkci aktivace

Matematický neuron — $N : X_1 \times \dots \times X_n \rightarrow Y$

- Lineární neurony $N(X) = w_0 + \sum_{i=1}^n w_i x_i$
 - $N(x)$ tedy je **aktivace neuronu** (z)
- Binární prahové neurony (**perceptrony**)
 - McCulloch & Pitts, 1943
 - je-li aktivace nižší než práh, vrací 0, jinak vrací 1
 - spike (akční potenciál) \approx pravdivostní hodnota výroku, neuron počítá pravdivostní hodnotu jiného výroku ze vstupních výroků
- Lineární prahové neurony
 - po práh vrací 0, nad ním nějakou lineární funkci aktivace
- Sigmoidy (logistická funkce $N(X) = \frac{1}{1+e^{-z}}$, příp. tanh)

Matematický neuron — $N : X_1 \times \dots \times X_n \rightarrow Y$

- Lineární neurony $N(X) = w_0 + \sum_{i=1}^n w_i x_i$
 - $N(x)$ tedy je **aktivace neuronu** (z)
- Binární prahové neurony (**perceptrony**)
 - McCulloch & Pitts, 1943
 - je-li aktivace nižší než práh, vrací 0, jinak vrací 1
 - spike (akční potenciál) \approx pravdivostní hodnota výroku, neuron počítá pravdivostní hodnotu jiného výroku ze vstupních výroků
- Lineární prahové neurony
 - po práh vrací 0, nad ním nějakou lineární funkci aktivace
- Sigmoidy (logistická funkce $N(X) = \frac{1}{1+e^{-z}}$, příp. tanh)
 - mají hezké limity ($\lim_{N \rightarrow -\infty} N(X) = 0$, $N(0) = 0.5$, $\lim_{N \rightarrow \infty} N(X) = 1$)

Matematický neuron — $N : X_1 \times \dots \times X_n \rightarrow Y$

- Lineární neurony $N(X) = w_0 + \sum_{i=1}^n w_i x_i$
 - $N(x)$ tedy je **aktivace neuronu** (z)
- Binární prahové neurony (**perceptrony**)
 - McCulloch & Pitts, 1943
 - je-li aktivace nižší než práh, vrací 0, jinak vrací 1
 - spike (akční potenciál) \approx pravdivostní hodnota výroku, neuron počítá pravdivostní hodnotu jiného výroku ze vstupních výroků
- Lineární prahové neurony
 - po práh vrací 0, nad ním nějakou lineární funkci aktivace
- Sigmoidy (logistická funkce $N(X) = \frac{1}{1+e^{-z}}$, příp. tanh)
 - mají hezké limity ($\lim_{N \rightarrow -\infty} N(X) = 0$, $N(0) = 0.5$, $\lim_{N \rightarrow \infty} N(X) = 1$) a derivace

Matematický neuron — $N : X_1 \times \dots \times X_n \rightarrow Y$

- Lineární neurony $N(X) = w_0 + \sum_{i=1}^n w_i x_i$
 - $N(x)$ tedy je **aktivace neuronu** (z)
- Binární prahové neurony (**perceptrony**)
 - McCulloch & Pitts, 1943
 - je-li aktivace nižší než práh, vrací 0, jinak vrací 1
 - spike (akční potenciál) \approx pravdivostní hodnota výroku, neuron počítá pravdivostní hodnotu jiného výroku ze vstupních výroků
- Lineární prahové neurony
 - po práh vrací 0, nad ním nějakou lineární funkci aktivace
- Sigmoidy (logistická funkce $N(X) = \frac{1}{1+e^{-z}}$, příp. tanh)
 - mají hezké limity ($\lim_{N \rightarrow -\infty} N(X) = 0$, $N(0) = 0.5$, $\lim_{N \rightarrow \infty} N(X) = 1$) a derivace
- Stochastické neurony

Matematický neuron — $N : X_1 \times \dots \times X_n \rightarrow Y$

- Lineární neurony $N(X) = w_0 + \sum_{i=1}^n w_i x_i$
 - $N(x)$ tedy je **aktivace neuronu** (z)
- Binární prahové neurony (**perceptrony**)
 - McCulloch & Pitts, 1943
 - je-li aktivace nižší než práh, vrací 0, jinak vrací 1
 - spike (akční potenciál) \approx pravdivostní hodnota výroku, neuron počítá pravdivostní hodnotu jiného výroku ze vstupních výroků
- Lineární prahové neurony
 - po práh vrací 0, nad ním nějakou lineární funkci aktivace
- Sigmoidy (logistická funkce $N(X) = \frac{1}{1+e^{-z}}$, příp. tanh)
 - mají hezké limity ($\lim_{N \rightarrow -\infty} N(X) = 0$, $N(0) = 0.5$, $\lim_{N \rightarrow \infty} N(X) = 1$) a derivace
- Stochastické neurony
 - jako sigmoidální, ale výstup se interpretuje jako pravděpodobnost, že se v určitém intervalu vygeneruje spike (akční potenciál)

Jednoduché rozpoznávání znaků

- dvouvrstvá síť

Jednoduché rozpoznávání znaků

- dvouvrstvá síť
- vstupní neurony = pixely

Jednoduché rozpoznávání znaků

- dvouvrstvá síť
- vstupní neurony = pixely
- výstupní neurony = jednotlivé znaky

Jednoduché rozpoznávání znaků

- dvouvrstvá síť
- vstupní neurony = pixely
- výstupní neurony = jednotlivé znaky
- pixel může hlasovat pokud je zbarvený

Jednoduché rozpoznávání znaků

- dvouvrstvá síť
- vstupní neurony = pixely
- výstupní neurony = jednotlivé znaky
- pixel může hlasovat pokud je zbarvený
- pixel může hlasovat pro víc znaků

Jednoduché rozpoznávání znaků

- dvouvrstvá síť
- vstupní neurony = pixely
- výstupní neurony = jednotlivé znaky
- pixel může hlasovat pokud je zbarvený
- pixel může hlasovat pro víc znaků
- znak s největším počtem hlasů vyhrává

Jednoduché rozpoznávání znaků

- dvouvrstvá síť
- vstupní neurony = pixely
- výstupní neurony = jednotlivé znaky
- pixel může hlasovat pokud je zbarvený
- pixel může hlasovat pro víc znaků
- znak s největším počtem hlasů vyhrává

Proces učení

Jednoduché rozpoznávání znaků

- dvouvrstvá síť
- vstupní neurony = pixely
- výstupní neurony = jednotlivé znaky
- pixel může hlasovat pokud je zbarvený
- pixel může hlasovat pro víc znaků
- znak s největším počtem hlasů vyhrává

Proces učení

V každém kroku:

Jednoduché rozpoznávání znaků

- dvouvrstvá síť
- vstupní neurony = pixely
- výstupní neurony = jednotlivé znaky
- pixel může hlasovat pokud je zbarvený
- pixel může hlasovat pro víc znaků
- znak s největším počtem hlasů vyhrává

Proces učení

V každém kroku:

- zvyš váhy z aktivních pixelů do správné třídy

Jednoduché rozpoznávání znaků

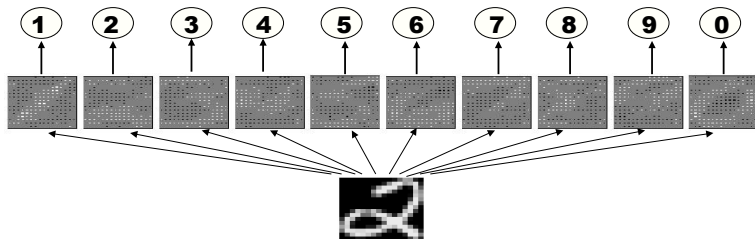
- dvouvrstvá síť
- vstupní neurony = pixely
- výstupní neurony = jednotlivé znaky
- pixel může hlasovat pokud je zbarvený
- pixel může hlasovat pro víc znaků
- znak s největším počtem hlasů vyhrává

Proces učení

V každém kroku:

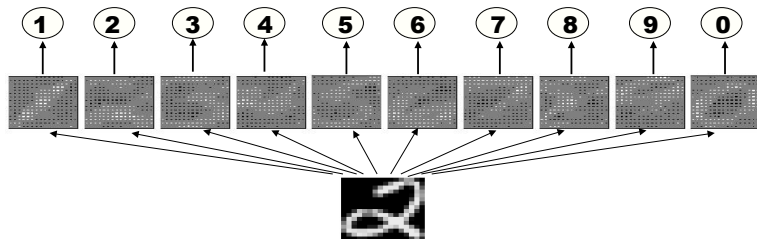
- zvyš váhy z aktivních pixelů do správné třídy
- následně sniž váhy z aktivních pixelů do aktuálně uhádnuté třídy

Rozpoznávání znaků — učení v obrázcích



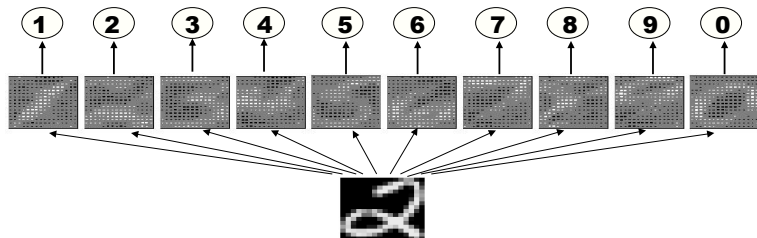
(zdroj: G. Hinton, Neural Networks for ML)

Rozpoznávání znaků — učení v obrázcích



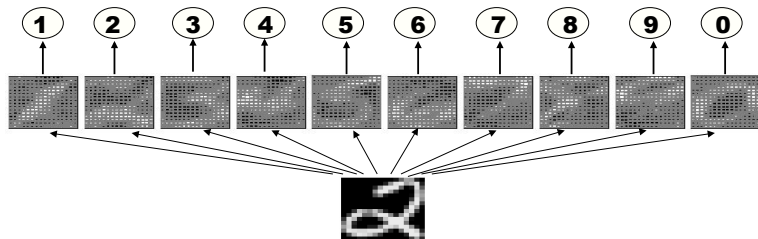
(zdroj: G. Hinton, Neural Networks for ML)

Rozpoznávání znaků — učení v obrázcích



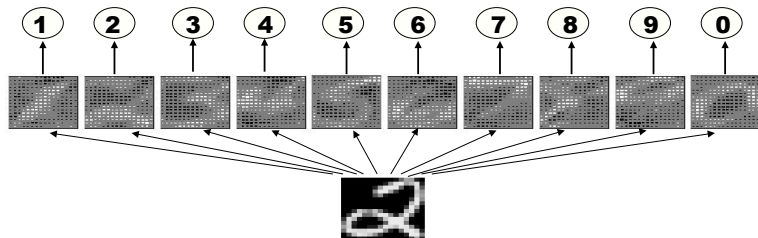
(zdroj: G. Hinton, Neural Networks for ML)

Rozpoznávání znaků — učení v obrázcích



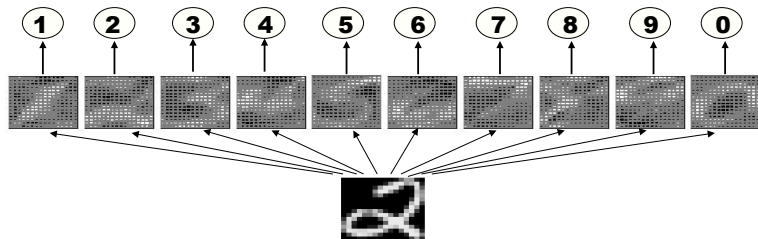
(zdroj: G. Hinton, Neural Networks for ML)

Rozpoznávání znaků — učení v obrázcích



(zdroj: G. Hinton, Neural Networks for ML)

Rozpoznávání znaků — učení v obrázcích



(zdroj: G. Hinton, Neural Networks for ML)

Typy neuronových sítí

- feed forward

Typy neuronových sítí

- feed forward (hluboké mají >1 skryté vrstvy)

Typy neuronových sítí

- feed forward (hluboké mají >1 skryté vrstvy)
- rekurentní neuronové sítě

Typy neuronových sítí

- feed forward (hluboké mají >1 skryté vrstvy)
- rekurentní neuronové sítě
- symetrické (Hopfieldova síť nemá skryté neurony)

I. Sutskever (2011) trénoval Neuronovou síť tak, aby uhodla následující znak v posloupnosti znaků.

I. Sutskever (2011) trénoval Neuronovou síť tak, aby uhodla následující znak v posloupnosti znaků. Trénovací data:

I. Sutskever (2011) trénoval Neuronovou síť tak, aby uhodla následující znak v posloupnosti znaků. Trénovací data: Wikipedie.

I. Sutskever (2011) trénoval Neuronovou síť tak, aby uhodla následující znak v posloupnosti znaků. Trénovací data: Wikipedie.

Výsledek (generovaný po jednom znaku)

I. Sutskever (2011) trénoval Neuronovou síť tak, aby uhodla následující znak v posloupnosti znaků. Trénovací data: Wikipedie.

Výsledek (generovaný po jednom znaku)

In 1974 Northern Denver had been overshadowed by CNL, and several Irish intelligence agencies in the Mediterranean region. However, on the Victoria, Kings Hebrew stated that Charles decided to escape during an alliance. The mansion house was completed in 1882, the second in its bridge are omitted, while closing is the proton reticulum composed below it aims, such that it is the blurring of appearing on any well-paid type of box printer.

(zdroj: G. Hinton, Neural Networks for ML)

Perceptron

Perceptron (binární prahový neuron)

Proces učení

Napřed si rozmyslet, jaké parametry by mohly být užitečné. Ručně vybrat/naprogramovat nástroje, které z dostupných dat tyto parametry extrahují.

Perceptron (binární prahový neuron)

Proces učení

Napřed si rozmyslet, jaké parametry by mohly být užitečné. Ručně vybrat/naprogramovat nástroje, které z dostupných dat tyto parametry extrahují.

Váha w_0 není ničím výjimečná: je to váha vstupního parametru, jehož hodnota je vždy 1.

Perceptron (binární prahový neuron)

Proces učení

Napřed si rozmyslet, jaké parametry by mohly být užitečné. Ručně vybrat/naprogramovat nástroje, které z dostupných dat tyto parametry extrahují.

Váha w_0 není ničím výjimečná: je to váha vstupního parametru, jehož hodnota je vždy 1.

Postupně po jednom probírej trénovací vzorky.

Perceptron (binární prahový neuron)

Proces učení

Napřed si rozmyslet, jaké parametry by mohly být užitečné. Ručně vybrat/naprogramovat nástroje, které z dostupných dat tyto parametry extrahují.

Váha w_0 není ničím výjimečná: je to váha vstupního parametru, jehož hodnota je vždy 1.

Postupně po jednom probírej trénovací vzorky.

- pokud je odpověď pro daný vzorek správná, nedělej nic

Perceptron (binární prahový neuron)

Proces učení

Napřed si rozmyslet, jaké parametry by mohly být užitečné. Ručně vybrat/naprogramovat nástroje, které z dostupných dat tyto parametry extrahují.

Váha w_0 není ničím výjimečná: je to váha vstupního parametru, jehož hodnota je vždy 1.

Postupně po jednom probírej trénovací vzorky.

- pokud je odpověď pro daný vzorek správná, nedělej nic
- pokud je odpověď 0, ale má být 1, zvyš váhy sítě o vektor vstupních parametrů

Perceptron (binární prahový neuron)

Proces učení

Napřed si rozmyslet, jaké parametry by mohly být užitečné. Ručně vybrat/naprogramovat nástroje, které z dostupných dat tyto parametry extrahují.

Váha w_0 není ničím výjimečná: je to váha vstupního parametru, jehož hodnota je vždy 1.

Postupně po jednom probírej trénovací vzorky.

- pokud je odpověď pro daný vzorek správná, nedělej nic
- pokud je odpověď 0, ale má být 1, zvyš váhy sítě o vektor vstupních parametrů
- pokud je odpověď 1, ale má být 0, sniž váhy sítě o vektor vstupních parametrů

Perceptron

Perceptron (binární prahový neuron)

Proces učení

Napřed si rozmyslet, jaké parametry by mohly být užitečné. Ručně vybrat/naprogramovat nástroje, které z dostupných dat tyto parametry extrahují.

Váha w_0 není ničím výjimečná: je to váha vstupního parametru, jehož hodnota je vždy 1.

Postupně po jednom probírej trénovací vzorky.

- pokud je odpověď pro daný vzorek správná, nedělej nic
- pokud je odpověď 0, ale má být 1, zvyš váhy sítě o vektor vstupních parametrů
- pokud je odpověď 1, ale má být 0, sniž váhy sítě o vektor vstupních parametrů

Pokud existuje nastavení vah, při kterém je odpověď pro všechny trénovací vzorky správná, tento postup jej najde.

<https://class.coursera.org/neuralnets-2012-001/>