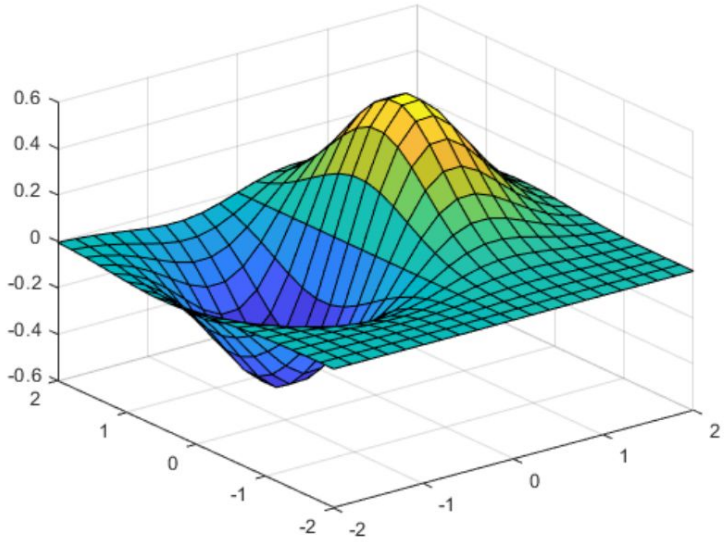
Jak zefektivnit učení NN

Regularizace

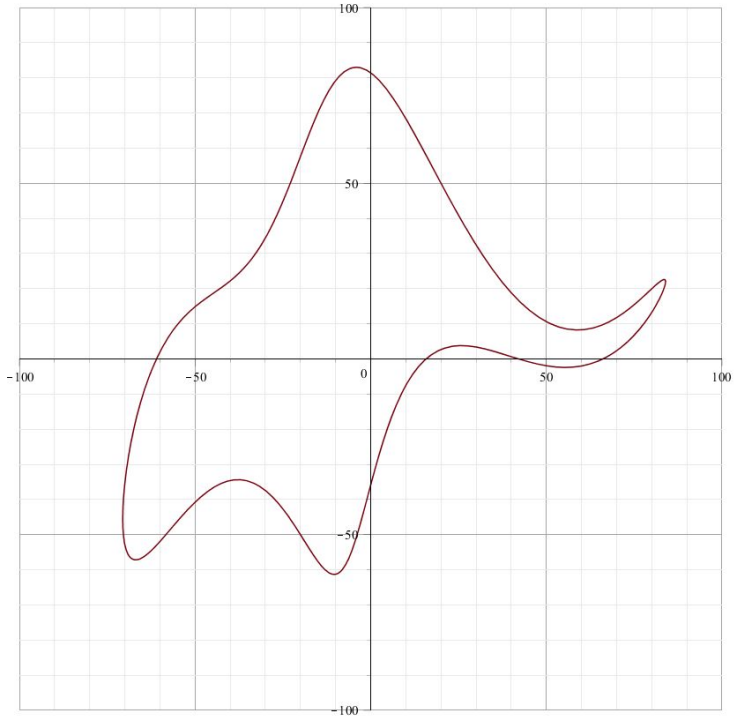


Overfitting

- Model s velkým počtem volných parametrů může popsat spoustu věcí

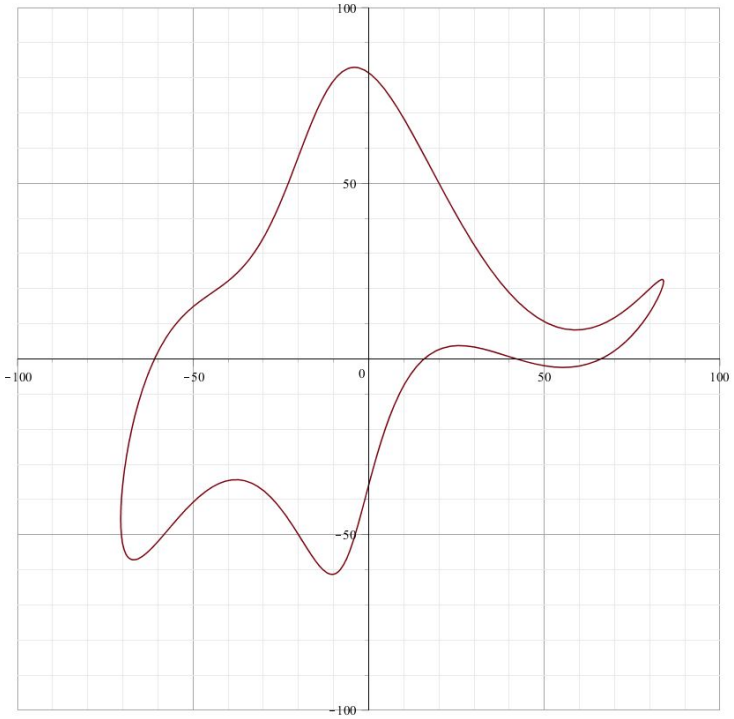
Overfitting

- Model s velkým počtem volných parametrů může popsat spoustu věcí
- Např. Von Neumannův slon, zkonstruovaný se 4 volnými parametry



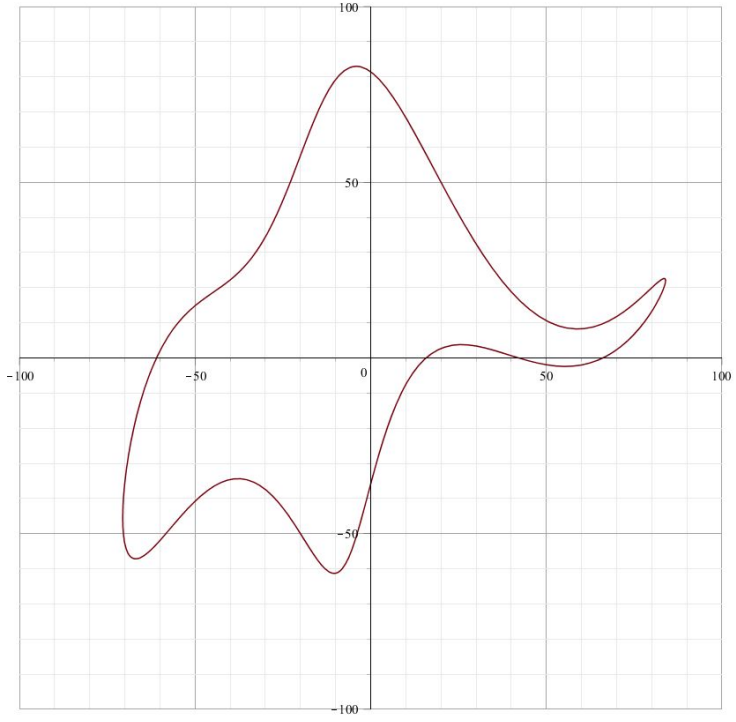
Overfitting

- Model s velkým počtem volných parametrů může popsat spoustu věcí
- Např. Von Neumannův slon, zkonstruovaný se 4 volnými parametry
- Naše síť jich má 24 000



Overfitting

- Model s velkým počtem volných parametrů může popsat spoustu věcí
- Např. Von Neumannův slon, zkonstruovaný se 4 volnými parametry
- Naše síť jich má 24 000
- To, že model dobře fituje data neznamená, že je to dobrý model. Možná má jen příliš stupňů volnosti.



Overfitting

- V takovém případě model dobře funguje na existujících (tréninkových) datech, ale nedokáže zobecnit vzorce v datech a využít je při predikci na nových datech

Overfitting

- V takovém případě model dobře funguje na existujících (tréninkových) datech, ale nedokáže zobecnit vzorce v datech a využít je při predikci na nových datech
- Proto model testujeme na extra datasetu (validační, test)

Overfitting

- V takovém případě model dobře funguje na existujících (tréninkových) datech, ale nedokáže zobecnit vzorce v datech a využít je při predikci na nových datech
- Proto model testujeme na extra datasetu (validační, test)
- Proto monitorujeme průběh tréninku, abychom ho případně včas zastavili

Overfitting

- **Jak odhalit overfitting?**

Overfitting

- **Jak odhalit overfitting?**
  - training cost klesá, validation cost ne (mimo stochastické fluktuace)

Overfitting

- **Jak odhalit overfitting?**
  - training cost klesá, validation cost ne (mimo stochastické fluktuace)
  - training accuracy stoupá, validation accuracy ne (mimo stochastické fluktuace)

Overfitting

- **Jak odhalit overfitting?**
  - training cost klesá, validation cost ne (mimo stochastické fluktuace)
  - training accuracy stoupá, validation accuracy ne (mimo stochastické fluktuace)
  - podezřele vysoká training accuracy

Overfitting

- **Jak odhalit overfitting?**
  - training cost klesá, validation cost ne (mimo stochastické fluktuace)
  - training accuracy stoupá, validation accuracy ne (mimo stochastické fluktuace)
  - podezřele vysoká training accuracy
  - velký rozdíl mezi training a validation accuracy

Overfitting

- **Jak zabránit overfittingu?**

Overfitting

- **Jak zabránit overfittingu?**
  - Předčasné ukončení učení (early stopping) - více možných strategií

Overfitting

- **Jak zabránit overfittingu?**
  - Předčasné ukončení učení (early stopping) - více možných strategií
  - **Regularizace**

Overfitting

- **Jak zabránit overfittingu?**
  - Předčasné ukončení učení (early stopping) - více možných strategií
  - **Regularizace**
  - Data augmentation, pruning, ensembling…

Regularizace

- Přidání extra pravidel během učení sítě pro zlepšení její schopnosti generalizovat

Regularizace

- Přidání extra pravidel během učení sítě pro zlepšení její schopnosti generalizovat
- **L2 regularizace (weight decay):**

Regularizace

- Přidání extra pravidel během učení sítě pro zlepšení její schopnosti generalizovat
- **L2 regularizace (weight decay):**
  - Přidáme do účelové funkce tzv. regularization term

Regularizace

- Přidání extra pravidel během učení sítě pro zlepšení její schopnosti generalizovat
- **L2 regularizace (weight decay):**
  - Přidáme do účelové funkce tzv. regularization term
  - Zaměřme se na cross entropii:

$$C = -\frac{1}{n}\sum_x \left(y_j \ln\left(a_j^L\right) + (1-y_j)\ln\left(1-a_j^L\right)\right) + \frac{\lambda}{2n}\sum_w w^2$$

Regularizace

- Přidání extra pravidel během učení sítě pro zlepšení její schopnosti generalizovat
- **L2 regularizace (weight decay):**
  - Přidáme do účelové funkce tzv. regularization term
  - Zaměřme se na cross entropii:

$$C = -\frac{1}{n}\sum_x \left(y_j \ln\left(a_j^L\right) + (1-y_j)\ln\left(1-a_j^L\right)\right) + \frac{\lambda}{2n}\sum_w w^2$$

  - lambda > 0 je regularizační parametr, w jsou váhy

Regularizace

- Přidání extra pravidel během učení sítě pro zlepšení její schopnosti generalizovat
- **L2 regularizace (weight decay):**
  - Přidáme do účelové funkce tzv. regularization term
  - Zaměřme se na cross entropii:

$$C = -\frac{1}{n}\sum_x \left(y_j \ln\left(a_j^L\right) + (1-y_j)\ln\left(1-a_j^L\right)\right) + \frac{\lambda}{2n}\sum_w w^2$$

  - lambda > 0 je regularizační parametr, w jsou váhy
  - Jak určit lambdu?

L2 regularizace

- Cross entropie: $C = -\frac{1}{n} \sum_x \left(y_j \ln \left(a_j^L \right) + (1 - y_j) \ln \left(1 - a_j^L \right) \right) + \frac{\lambda}{2n} \sum_w w^2$

L2 regularizace

- Cross entropie: $C = -\frac{1}{n} \sum_x \left(y_j \ln \left(a_j^L \right) + (1 - y_j) \ln \left(1 - a_j^L \right) \right) + \frac{\lambda}{2n} \sum_w w^2$
- MSE: $C = \frac{1}{2n} \sum_x \left\| y - a^L \right\|^2 + \frac{\lambda}{2n} \sum_w w^2$

L2 regularizace

- Cross entropie: $C = -\frac{1}{n} \sum_x \left(y_j \ln \left(a_j^L \right) + (1 - y_j) \ln \left(1 - a_j^L \right) \right) + \frac{\lambda}{2n} \sum_w w^2$
- MSE: $C = \frac{1}{2n} \sum_x \left\| y - a^L \right\|^2 + \frac{\lambda}{2n} \sum_w w^2$
- Obecně: $C = C_0 + \frac{\lambda}{2n} \sum_w w^2$

L2 regularizace

- Cross entropie: $C = -\frac{1}{n}\sum_x \left(y_j \ln\left(a_j^L\right) + (1-y_j)\ln\left(1-a_j^L\right)\right) + \frac{\lambda}{2n}\sum_w w^2$
- MSE: $C = \frac{1}{2n}\sum_x \left\|y - a^L\right\|^2 + \frac{\lambda}{2n}\sum_w w^2$
- Obecně: $C = C_0 + \frac{\lambda}{2n}\sum_w w^2$
- Co nám tenhle výraz vlastně říká?

L2 regularizace

- Cross entropie: $C = -\frac{1}{n}\sum_x \left(y_j \ln\left(a_j^L\right) + (1-y_j)\ln\left(1-a_j^L\right)\right) + \frac{\lambda}{2n}\sum_w w^2$
- MSE: $C = \frac{1}{2n}\sum_x \left\|y - a^L\right\|^2 + \frac{\lambda}{2n}\sum_w w^2$
- Obecně: $C = C_0 + \frac{\lambda}{2n}\sum_w w^2$
- Co nám tenhle výraz vlastně říká?
- Snažíme se, aby se síť naučila co nejmenší váhy, respektive aby našla kompromis mezi malými vahami a minimalizací účelové funkce - na základě hodnoty lambdy: malá lambda znamená důraz na minimalizaci účelové funukce, velká lambda důraz na malé váhy

L2 regularizace

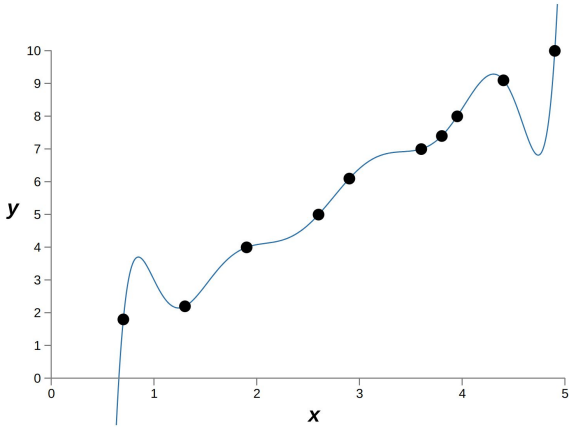
- **K čemu je dobré mít malé váhy?**

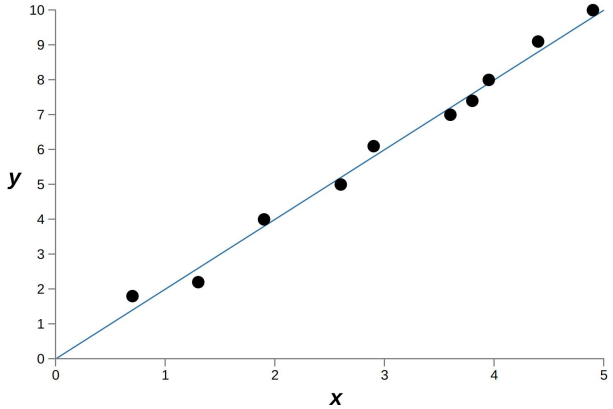
L2 regularizace

- **K čemu je dobré mít malé váhy?**
- Po pravdě… nevíme.

L2 regularizace

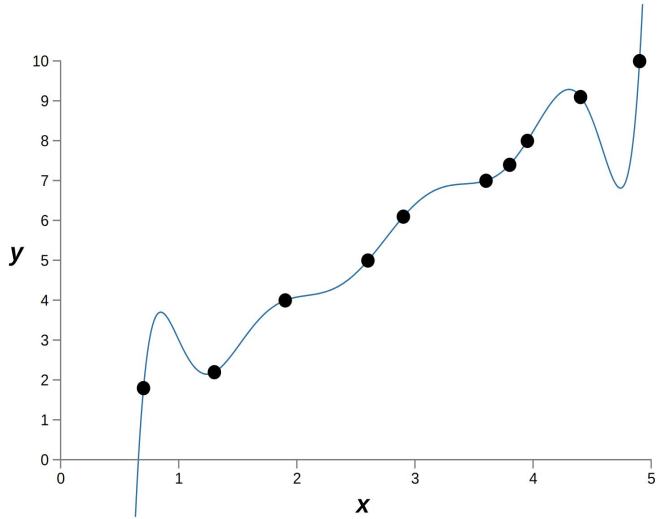
- **K čemu je dobré mít malé váhy?**
- Po pravdě… nevíme.
- Možné vysvětlení: malé váhy nabízejí méně komplexní model, který je tím pádem lépe schopen generalizace. Navíc je robustnější vůči šumu

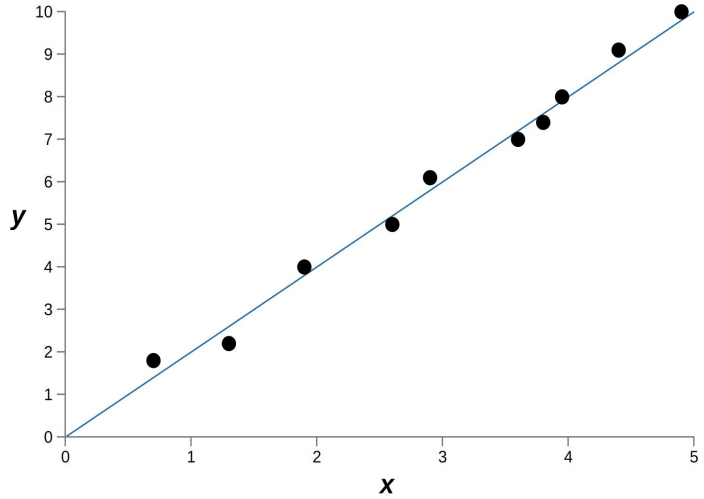




L2 regularizace

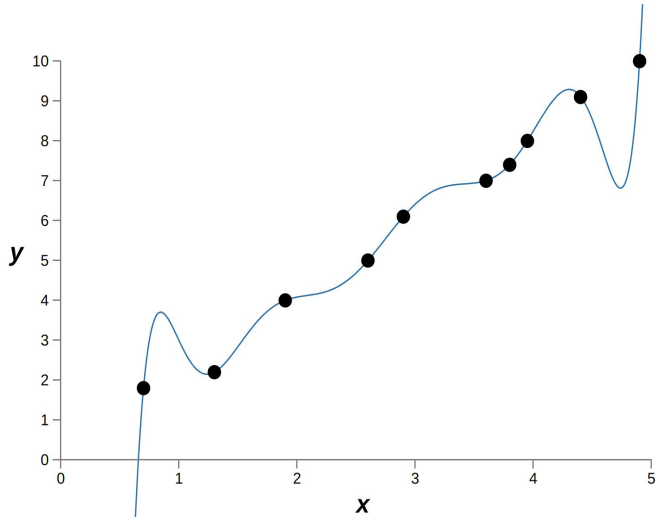
- **Co je lepší model?**

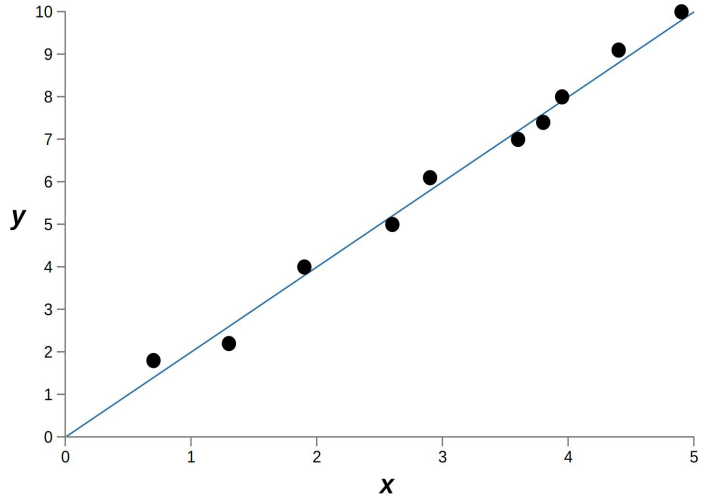




L2 regularizace

- **Co je lepší model?**

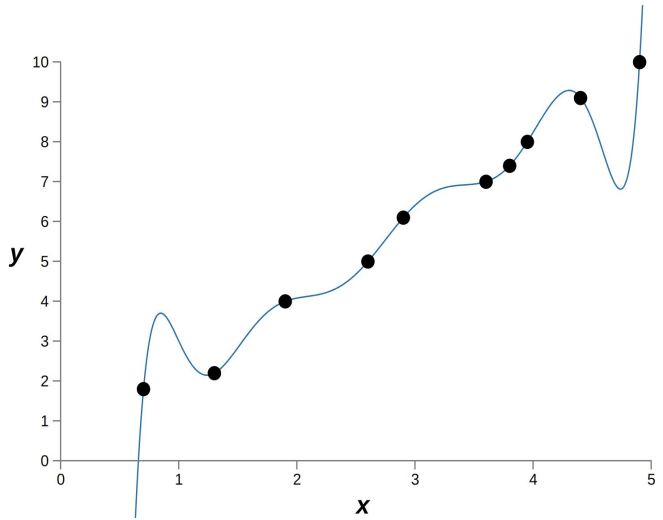


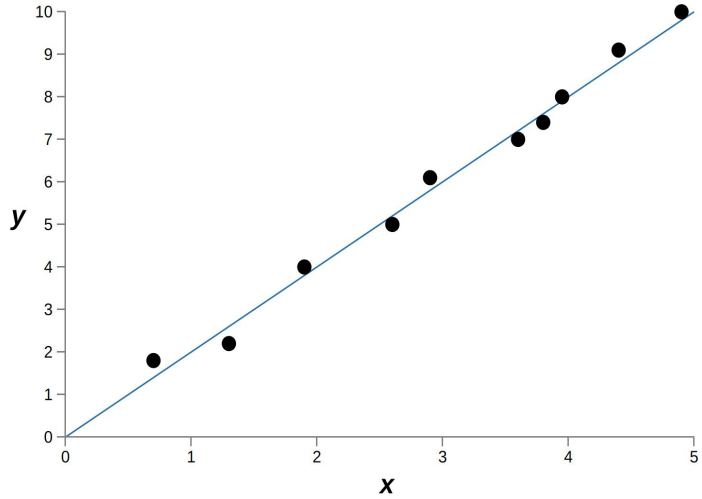


- Nelze jednoznačně určit

L2 regularizace

- **Co je lepší model?**





- Nelze jednoznačně určit
- Predikce pro velké hodnoty x se budou dramaticky lišit

L2 regularizace

- Často prostě chceme co nejjednodušší model, protože se zdá, že popisuje nějaký vzorec chování “skrytý” v datech a je nepravdpodobné, aby to byla jen náhoda.

L2 regularizace

- Často prostě chceme co nejjednodušší model, protože se zdá, že popisuje nějaký vzorec chování “skrytý” v datech a je nepravdpodobné, aby to byla jen náhoda.
- Komplexní modely (jako polynom v předchozím případě) vlastně jen zohledňují šum v datech

L2 regularizace

- Často prostě chceme co nejjednodušší model, protože se zdá, že popisuje nějaký vzorec chování “skrytý” v datech a je nepravdpodobné, aby to byla jen náhoda.
- Komplexní modely (jako polynom v předchozím případě) vlastně jen zohledňují šum v datech
- V případě neuronových sítí malé váhy znamenají, že výstup se příliš nezmění, pokud trochu upravíme vstupní data => síť se nenaučí šum v datech, ale snaží se vytvořit co nejjednodušší model

L2 regularizace

- Často prostě chceme co nejjednodušší model, protože se zdá, že popisuje nějaký vzorec chování “skrytý” v datech a je nepravdpodobné, aby to byla jen náhoda.
- Komplexní modely (jako polynom v předchozím případě) vlastně jen zohledňují šum v datech
- V případě neuronových sítí malé váhy znamenají, že výstup se příliš nezmění, pokud trochu upravíme vstupní data => síť se nenaučí šum v datech, ale snaží se vytvořit co nejjednodušší model
- Ocacamova břitva

L2 regularizace

- Empiricky se regularizované sítě většinou chovají lépe, ale není to pravidlo

L2 regularizace

- Empiricky se regularizované sítě většinou chovají lépe, ale není to pravidlo
- Síť se 100 neurony ve skryté vrstvě má skoro 80 000 parametrů, trénink dataset má 50 000 obrázků. Není to jako fitovat polynom řádu 80 000 skrz 50 000 bodů?

L2 regularizace

- Empiricky se regularizované sítě většinou chovají lépe, ale není to pravidlo
- Síť se 100 neurony ve skryté vrstvě má skoro 80 000 parametrů, trénink dataset má 50 000 obrázků. Není to jako fitovat polynom řádu 80 000 skrz 50 000 bodů?
- Učení MLP pomocí gradientních metod má (empiricky) sebe regularizační efekt - naštěstí!

L2 regularizace

- Empiricky se regularizované sítě většinou chovají lépe, ale není to pravidlo
- Síť se 100 neurony ve skryté vrstvě má skoro 80 000 parametrů, trénink dataset má 50 000 obrázků. Není to jako fitovat polynom řádu 80 000 skrz 50 000 bodů?
- Učení MLP pomocí gradientních metod má (empiricky) sebe regularizační efekt - naštěstí!
- Proč ne biasy? Konvence. Velký bias nezmění příliš chování neuronu (tak jako velká váha) a empiricky se sítě chovají podobně. Navíc to síti dává větší flexibilitu, protože to může zjednodušit saturování neuronu

L2 regularizace

- Jak se změní backpropagation a SGD s regularizovanou účelovou funkcí?

L2 regularizace

- Jak se změní backpropagation (respektive parciální derivace C) a SGD s regularizovanou účelovou funkcí?
- Backpropagation:

$$\frac{\partial C}{\partial w} = \frac{\partial C_0}{\partial w} + \frac{\lambda}{n} w$$

$$\frac{\partial C}{\partial b} = \frac{\partial C_0}{\partial b}$$

L2 regularizace

- Jak se změní backpropagation (respektive parciální derivace C) a SGD s regularizovanou účelovou funkcí?
- Backpropagation:
$$\frac{\partial C}{\partial w} = \frac{\partial C_0}{\partial w} + \frac{\lambda}{n} w$$
$$\frac{\partial C}{\partial b} = \frac{\partial C_0}{\partial b}$$
- GD:
$$b \to b - \eta \frac{\partial C_0}{\partial b}$$
$$w \to w - \eta \frac{\partial C_0}{\partial w} - \frac{\eta \lambda}{n} w = \left(1 - \frac{\eta \lambda}{n}\right) w - \eta \frac{\partial C_0}{\partial w}$$

L2 regularizace

- Jak se změní backpropagation (respektive parciální derivace C) a SGD s regularizovanou účelovou funkcí?
- Backpropagation:
$$\frac{\partial C}{\partial w} = \frac{\partial C_0}{\partial w} + \frac{\lambda}{n} w$$
$$\frac{\partial C}{\partial b} = \frac{\partial C_0}{\partial b}$$
- GD:
$$b \to b - \eta \frac{\partial C_0}{\partial b}$$
$$w \to w - \eta \frac{\partial C_0}{\partial w} - \frac{\eta \lambda}{n} w = \left(1 - \frac{\eta \lambda}{n}\right) w - \eta \frac{\partial C_0}{\partial w}$$
- SGD:
$$b \to b - \frac{\eta}{m} \sum_x \frac{\partial C_x}{\partial b}$$
$$w \to \left(1 - \frac{\eta \lambda}{n}\right) w - \frac{\eta}{m} \sum_x \frac{\partial C_x}{\partial w}, \; C_x \text{ je } \mathrm{C}_0 \text{ pro jeden vstup } x$$

L2 regularizace

- Pro biasy stejné, váhy přeškálujeme výrazem $1 - \frac{\eta\lambda}{n}$

L2 regularizace

- Pro biasy stejné, váhy přeškálujeme výrazem $1 - \frac{\eta\lambda}{n}$
- Parciální derivace C spočítáme pomocí backpropagation (všimněte si, že tento algoritmus se nezmění), regularizační výraz přičteme teprve při výpočtu nových vah (tj. přičteme ho ke gradientu neregularizované C)

L1 regularizace

- $C = C_0 + \frac{\lambda}{n} \sum_w |w|$

L1 regularizace

- $C = C_0 + \frac{\lambda}{n} \sum_w |w|$
- Podobné jako L2: penalizujeme velké váhy

L1 regularizace

- $C = C_0 + \frac{\lambda}{n} \sum_w |w|$
- Podobné jako L2: penalizujeme velké váhy
- $\frac{\partial C}{\partial w} = \frac{\partial C_0}{\partial w} + \frac{\lambda}{n} sgn(w),\ sgn(0) := 0$

L1 regularizace

- $C = C_0 + \frac{\lambda}{n} \sum_w |w|$
- Podobné jako L2: penalizujeme velké váhy
- $\frac{\partial C}{\partial w} = \frac{\partial C_0}{\partial w} + \frac{\lambda}{n} sgn(w),\ sgn(0) := 0$
- GD: $w \to w - \eta \frac{\partial C_0}{\partial w} - \frac{\eta \lambda}{n} sgn(w)$

L1 regularizace

- $C = C_0 + \frac{\lambda}{n} \sum_w |w|$
- Podobné jako L2: penalizujeme velké váhy
- $\frac{\partial C}{\partial w} = \frac{\partial C_0}{\partial w} + \frac{\lambda}{n} sgn(w),\ sgn(0) := 0$
- GD: $w \to w - \eta \frac{\partial C_0}{\partial w} - \frac{\eta \lambda}{n} sgn(w)$
- SGD: $w \to w - \frac{\eta \lambda}{n} sgn(w) - \frac{\eta}{m} \sum_x \frac{\partial C_x}{\partial w}$, C_x je C_0 pro jeden vstup x

L1 regularizace

- $C = C_0 + \frac{\lambda}{n} \sum_w |w|$
- Podobné jako L2: penalizujeme velké váhy
- $\frac{\partial C}{\partial w} = \frac{\partial C_0}{\partial w} + \frac{\lambda}{n} sgn(w),\ sgn(0) := 0$
- GD: $w \to w - \eta \frac{\partial C_0}{\partial w} - \frac{\eta\lambda}{n} sgn(w)$
- SGD: $w \to w - \frac{\eta\lambda}{n} sgn(w) - \frac{\eta}{m} \sum_x \frac{\partial C_x}{\partial w}$, C_x je C_0 pro jeden vstup x

- Rozdíl je v tom, že L2 zmenšuje váhu proporčně její hodnotě, zatímco L1 o konstantní hodnotu

L1 regularizace

- $C = C_0 + \frac{\lambda}{n} \sum_w |w|$
- Podobné jako L2: penalizujeme velké váhy
- $\frac{\partial C}{\partial w} = \frac{\partial C_0}{\partial w} + \frac{\lambda}{n} sgn(w),\ sgn(0) := 0$
- GD: $w \to w - \eta \frac{\partial C_0}{\partial w} - \frac{\eta\lambda}{n} sgn(w)$
- SGD: $w \to w - \frac{\eta\lambda}{n} sgn(w) - \frac{\eta}{m} \sum_x \frac{\partial C_x}{\partial w}$, C_x je C_0 pro jeden vstup x

- Rozdíl je v tom, že L2 zmenšuje váhu proporčně její hodnotě, zatímco L1 o konstantní hodnotu
- Velké váhy se rychleji zmenšují užitím L2, malé užitím L1

Dropout regularizace

- Nemění účelovou funkci, ale přímo síť

Dropout regularizace

- Nemění účelovou funkci, ale přímo síť
- Náhodně “vypneme” část neuronů ve skryté vrstvě - a provedeme forward+backward pass

Dropout regularizace

- Nemění účelovou funkci, ale přímo síť
- Náhodně “vypneme” část neuronů ve skryté vrstvě - a provedeme forward+backward pass
- Jako kdybychom trénovali několik různých sítí a průměrovali jejich output

Dropout regularizace

- Nemění účelovou funkci, ale přímo síť
- Náhodně “vypneme” část neuronů ve skryté vrstvě - a provedeme forward+backward pass
- Jako kdybychom trénovali několik různých sítí a průměrovali jejich output
- Redukuje komplexní závislosti mezi neurony, protože se neurony nemůžou spolehnout na všechny ostatní neurony v okolních vrstvách => tj. model je robustní ke ztrátě nějaké informace

Data augmentation

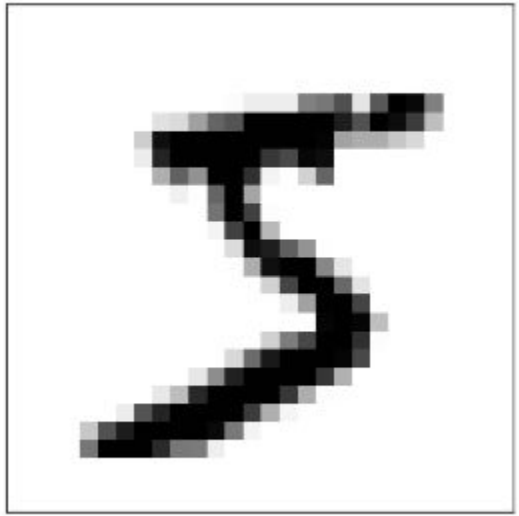
- Umělé rozšíření tréninkových dat

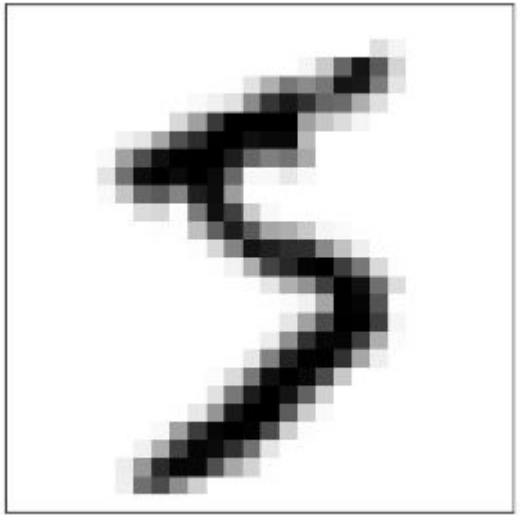
Data augmentation

- Umělé rozšíření tréninkových dat
- Rotace, rozostření, zrcadlení, změna jasu/kontrastu, přiblížení, zešikmení…

Data augmentation

- Umělé rozšíření tréninkových dat
- Rotace, rozostření, zrcadlení, změna jasu/kontrastu, přiblížení, zešikmení…





Data augmentation

- Umělé rozšíření tréninkových dat
- Rotace, rozostření, zrcadlení, změna jasu/kontrastu, přiblížení, zešikmení…
- Cvičení: K jakému problému může dojít při velkých rotacích na MNIST datasetu?

Data augmentation

- Umělé rozšíření tréninkových dat
- Rotace, rozostření, zrcadlení, změna jasu/kontrastu, přiblížení, zešikmení…
- Cvičení: K jakému problému může dojít při velkých rotacích na MNIST datasetu? 6 se může změnit v 9 a naopak