Computer-Linguistische Anwendungen

CLA | B.Sc. | LMU

Gradient Descent

Immobilienpreise in München

Eingabe Variable X, Größe (m^2)	Ausgabe Variable Y, Preis (€) in 1000s
60	555
75	695
90	830
100	925
120	1110

60m^2 kosten im Schnitt 555.000€

Wir verwenden *m* hier als die Anzahl der Trainings-Instanzen

Setup um den Immobilienpreis Prädiktor mit Gradient Descent zu lernen

- Hypothesis (Vorhersage):

$$h_\theta = \theta_0 + \theta_1 x$$

$f(x) = m^* x + b$

, wobei $\theta = [\,\theta_0, \theta_1]$ (θ ist ein Vektor), θ_0 ist der Bias und θ_1 ist der Koeffizient (wie bei linearer Funktion)

- Parameter: $\theta = (\,\theta_0, \theta_1)$

- Kostenfunktion (cost function: mean squared error - Wie gut unsere Vorhersage ist):

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

, wobei $h_\theta(x^{(i)})$ unsere Vorhersage ist als $\hat{y}$ und $y^{(i)}$ ist der Wert aus unseren Trainingsdaten (Label)

- Ziel: Minimisation von $J\,(\,\theta_0, \theta_1)$

Parameter der zwei Lern-Settings

Immobilienpreise

$$\theta = (\theta_0, \theta_1)$$

word2vec skipgram:

$\theta_{11}, \theta_{12}, \dots, \theta_{1d}$
$\theta_{21}, \theta_{22}, \dots, \theta_{2d}$
...
$\theta_{n1}, \theta_{n2}, \dots, \theta_{nd}$

$\eta_{11}, \eta_{12}, \dots, \eta_{1d}$
$\eta_{21}, \eta_{22}, \dots, \eta_{2d}$
...
$\eta_{n1}, \eta_{n2}, \dots, \eta_{nd}$

Dimensionalität der Embeddings:	d
Größe des Vokabulars:	n
Word Embeddings:	θ
Context Embeddings:	η

Parameter der zwei Lern-Settings

Immobilienpreise

$$h_\theta = \theta_0 + \theta_1 x$$

(lineare Funktion)

Word2Vec Skipgram:

$$h_{\theta,\eta}(w, c) = \mathrm{sim}(\theta(w), \eta(c))$$

Lies: “Die Ähnlichkeit des Wort-Vektors θ und des Kontext-Vektors η für das Wort-Kontext-Paar w, c “

Diese Ähnlichkeit sollte ~ 1 sein, wenn das Wort-Kontext-Paar im Korpus vorhanden ist (≈ P(GOOD|w,c)), und sollte ~ 0 sein, wenn es eine zufällige Kombination zweier Wörter aus dem Vokabular ist.

Cost Function

Immobilienpreise

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

word2vec skipgram:

Ähnlichkeit maximieren

Kosten minimieren

$$-[\sum_{(w,c)\in D} \log \sigma(\vec{v}_w \cdot \vec{v}_c) + \beta \sum_{(w,c)\in V\times V} \log \sigma(-\vec{v}_w \cdot \vec{v}_c)]$$

$$J(\theta, \eta) = -[\sum_{(w,c)\in D} \log \sigma(\theta(w) \cdot \eta(c)) + \beta \sum_{(w,c)\in V\times V} \log \sigma(-\theta(w) \cdot \eta(c))]$$

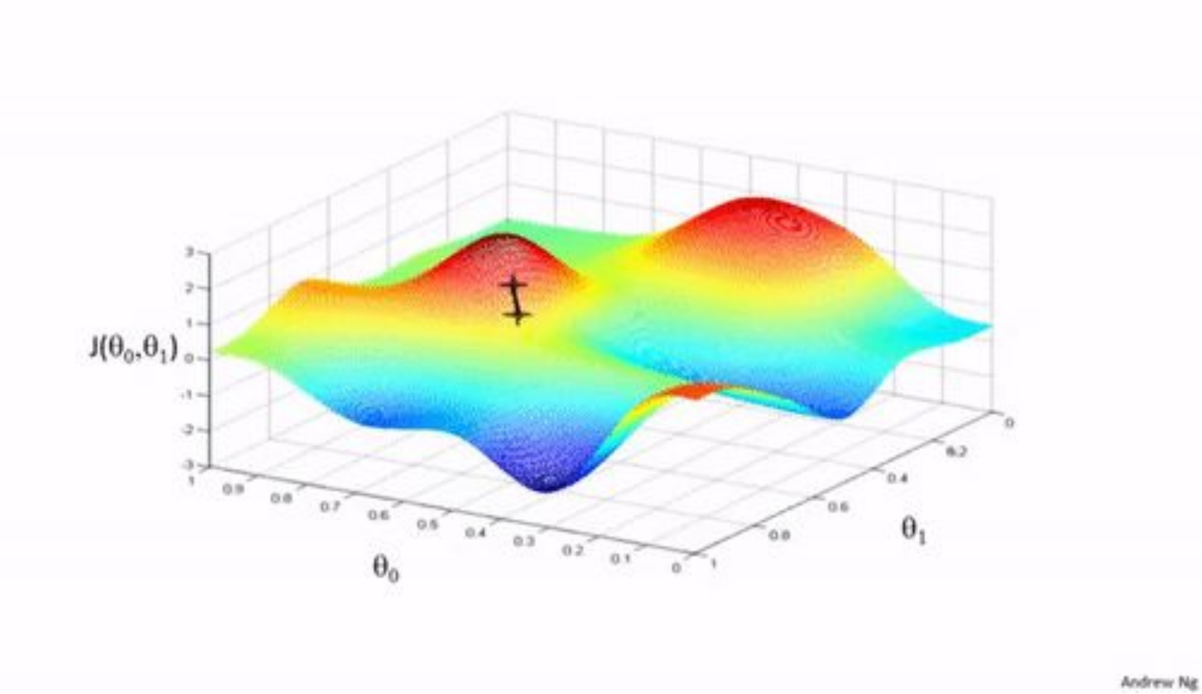
Ziel: Gradient Descent

Gradient Descent: Algorithmus um die Parameter einer Funktion zu optimieren

Immobilienpreise: $\text{minimiere}_{\theta 0,\, \theta 1}\, J\,(\theta_0, \theta_1)$

Word2vec skipgram: $\text{minimiere}_{\theta,\eta}\ J\,(\theta_0, \theta_\eta)$

Gradient Descent: Intuition



Gradient Descent

- Beginne mit einer zufälligen Wahl der Parameter (Starte irgendwo auf der Karte)
- Berechne die Kosten die durch diese Parameter entstehen (Berechne die aktuelle Höhe)
- Wir tasten nun in alle Richtungen die Steigung ab und wählen einen Schritt in die Richtung mit der größten Neigung - die Schrittgröße ist ein entscheidender Hyperparameter
- Wir wiederholen diese Schritte bis zu dem Punkt an dem die Steigung 0 ist, wir eine Fläche erreichen, oder nicht weniger tief kommen, wir erreichen ein lokales Minimum/Optimum (ein Tal)
- Welches lokale Minimum wir finden, hängt vom Startpunkt und der Schrittweite ab.

Gradient Descent

	house prices	word2vec skipgram
Parameter	θ_0, θ_1	$2\|V\|d$ parameters: θ, η
Vorhersage	$h_\theta(x) = \theta_0 + \theta_1 x$	$h_{\theta,\eta}(w,c) = sim(\theta(w), \eta(c))$
Cost Function	$J(\theta) =$ $1/(2m)\sum(h_\theta(x^{(i)}) - y^{(i)})^2$ (mean squared error)	$J(\theta, \eta) =$ $-[\sum_{(w,c)\in D} \log \sigma(\theta(w) \cdot \eta(c))$ $+\beta \sum_{(w,c)\in V\times V}$ $\log \sigma(-\theta(w) \cdot \eta(c))]$ (negative log likelihood)
Ziel	$\text{argmin}_\theta J(\theta)$	$\text{argmin}_{\theta,\eta} J(\theta, \eta)$

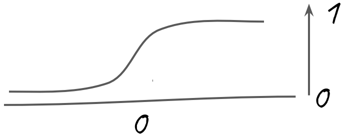
Aufgabe

- Was ist der minimale Wert, den der Wert als Ziel des Word2Vec skip grams erreichen kann? Man betrachtet den Term unten.
- Ist es wahrscheinlich, dass wir Parameter finden, die das Minimum erreichen?
- Man bedenke: $\sigma(x) = 1/(1 + e^{-x}))$

$$J(\theta, \eta) = -[\sum_{(w,c)\in D} \log \sigma(\theta(w) \cdot \eta(c)) + \beta \sum_{(w,c)\in V\times V} \log \sigma(-\theta(w) \cdot \eta(c))]$$

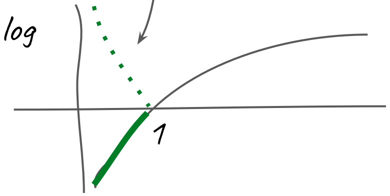
Aufgabe

- Was ist der minimale Wert, den der Wert als Ziel des Word2Vec skip grams erreichen kann? Man betrachtet den Term unten.
- Ist es wahrscheinlich, dass wir Parameter finden, die das Minimum erreichen?
- Man bedenke: $\sigma(x) = 1/(1 + e^{-x}))$



Minimal cost = 0
when word-vectors have high dot product

$$J(\theta, \eta) = -[\sum_{(w,c)\in D} \log \sigma(\theta(w) \cdot \eta(c)) + \beta \sum_{(w,c)\in V\times V} \log \sigma(-\theta(w) \cdot \eta(c))]$$



High dot product of words that occur together ⇔ low cost