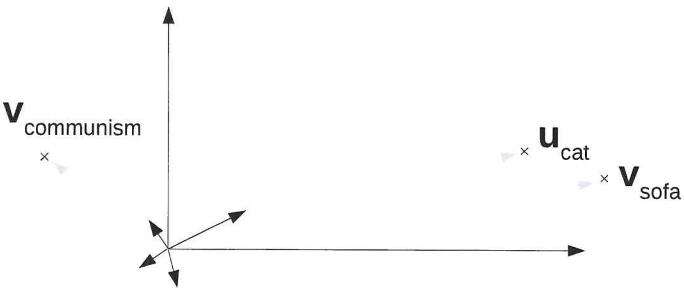
Das Word2vec-Modell



- Jeder Term t besitzt <u>zwei</u> **Embeddings** $\mathbf{u}_t, \mathbf{v}_t \in \mathbb{R}^{300}$
- Ein einfacher Sigmoid wird verwendet um $P(t_2|t_1)$ zu schätzen

$$P(t_2|t_t) := \sigma\Big(< \mathbf{u}_{t_1}, \mathbf{v}_{t_2} > \Big) = \frac{1}{1 + exp(- < \mathbf{u}_{t_1}, \mathbf{v}_{t_2} >)}$$

- Ziel **Lerne** die Embeddings durch **Backpropagation**!

Das Word2vec-Modell: Illustration

- Initialisiere $u_t, v_{t'}$ zufällig
- Ermittle die Häufigkeit h_t jedes Terms. Definiere eine Verteilung:

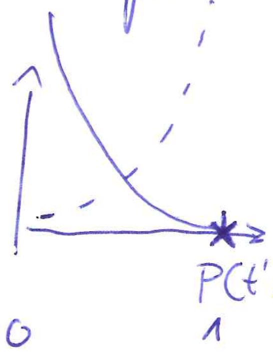
$$P_t = \frac{1}{Z} \cdot \left(\frac{h_t}{\sum_{t'} h_{t'}} \right)^{\beta} \quad \text{mit } \beta = 3/4$$

Normalisierung

- repeat:
  - Wähle ein Termpaar (t, t')
  - Sample "negative" Terme $t_1, \ldots, t_k \sim P$
  - Wir definiere den Loss:

$$L = -\log P(t'|t) + \sum_k -\log(1 - P(t_k|t$$

  - ...



Das Word2vec-Modell: Lernen

– Rechne:

$$(\cancel{NR})\quad u_t := u_t - \lambda \cdot \nabla_{u_t} L = u_t + \lambda \cdot \Big[(1 - P(t'|t)) \cdot v_{t'} - \sum_k P(t_k|t) \cdot v_{t_k} \Big]$$

$$(NR)\quad v_{t'} := v_{t'} - \lambda \cdot \nabla_{v_{t'}} L = v_{t'} + \lambda \cdot \underline{(1 - P(t'|t))} \cdot u_t$$

$$v_{t_k} := v_{t_k} - \lambda \cdot \nabla_{v_{t_k}} L = v_{t_k} - \lambda \cdot P(t_k|t) \cdot u_t$$

$$\boxed{\sigma' = \sigma \cdot (1 - \sigma)}$$

$$(NR)\quad \nabla_{v_{t'}} L = \nabla_{v_{t'}} - \log(P(t'|t))$$

$$= \text{〃} - \log(\sigma(\langle u_{t'} v_{t'} \rangle))$$

$$= -\frac{1}{\sigma(\dots)} \cdot \nabla_{v_{t'}} \sigma(\langle u_{t'} v_{t'} \rangle)$$

$$= -\frac{1}{\cancel{\sigma(\dots)}} \cdot \cancel{\sigma(\dots)} \cdot (1 - \sigma(\dots)) \cdot \nabla_{v_{t'}} \langle u_{t'}, v_{t'} \rangle$$

Das Word2vec-Modell: Lernen

$$= -(1 - \sigma(\dots)) \cdot u_t$$

$$= -(1 - P(t'|t)) \cdot u_t$$