Mathematical Foundations of Machine Learning
Assignment

David Felipe Alvear Goyes

2. RNN For generation

RNN Equations Block

$$O = H W_{hq} + b_q$$

$$H = \phi(X W_{xh} + b_h)$$

RNN Generation Model

Input combined $\rightarrow X \in \mathbb{R}^{n \times d}$

$n \rightarrow$ Batch Size
$d \rightarrow$ inputs

$$\boxed{O = \sigma\left(W_o\left(x w_i + b_i + x w_h + b_h\right) + b_o\right)}$$

$\rightarrow W_i \in \mathbb{R}^{d \times h_i}$, $b_i \in \mathbb{R}^{1 \times h_i}$

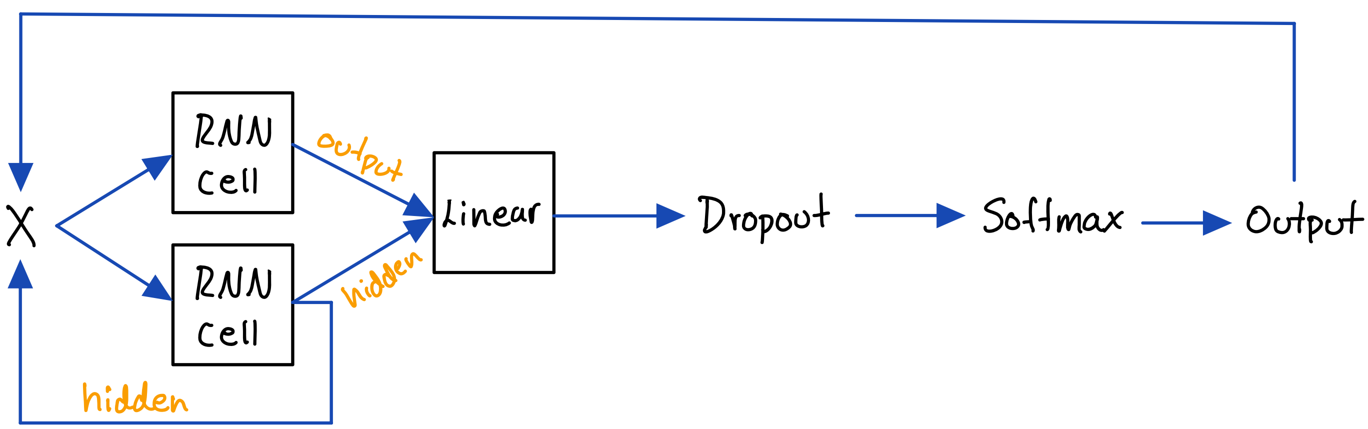
$\rightarrow W_h \in \mathbb{R}^{d \times h_h}$, $b_h \in \mathbb{R}^{1 \times h_h}$

$\rightarrow W_o \in \mathbb{R}^{h_i + h_h \times h_o}$, $b_o \in \mathbb{R}^{h_o}$

$$X = C + I + h$$

$h \rightarrow$ hidden
$C \rightarrow$ category
$I \rightarrow$ input

$\rightarrow$ Sketch RNN Network using RNN cell



3.

15.1 Word embeddings

15.1.3 Skip-gram model

→ Given word generate surroundings

→ Use conditional probability for generate context words

Word i → 2 d-dim vector
- ↳ $v_i \in \mathbb{R}^d$ → For being center word
- ↳ $u_i \in \mathbb{R}^d$ → For being context word

→ Conditional Probability given w_c generate w_o

$$\boxed{P(w_o \mid w_c) = \frac{\exp(u_o^T v_c)}{\sum_{i \in V} \exp(u_i^T v_c)}} \quad \Big\} \text{ Softmax operation}$$

* Vocab $V = \{0, 1, \cdots |V|-1\}$

Likelihood fn Skip-gram model → $\boxed{\prod_{t=1}^{T} \prod_{-m \le j \le m} P(w^{(t+j)} \mid w^{(t)})}$ } Context window m

Training

Skip-gram model Parameters → Word → Center word vec, → Context word vec

Learn maximizing likelihood fn:

$$\boxed{-\sum_{t=1}^{T} \sum_{-m \le j \le m} \log P(w^{(t+j)} \mid w^{(t)})}$$

→ SGD use short subsequence

1. Explain how the softmax training loss is the cross entropy

$H(p,q) = -\sum_{x} p(x) \log q(x)$

True distribution

→ for skip-gram model $p(x) = 1$ for the actual context word w_o and cero for others

Predicted distribution

→ $q(x) = P(w^{t+j} \mid w^t)$ Model's estimate of the probability of each context word given current word

The softmax training loss extends the cross-entropy calculation to a sum over words in a context window.

2. Derive gradient expression

→ Calculate gradient log cond probability

$$\log P(w_o|w_c) = u_o^T v_c - \log\left(\sum_{i \in V} \exp(u_i^T v_c)\right)$$

$$\frac{\partial \log P(w_o|w_c)}{\partial v_c} = u_o^T - \frac{\sum_{j \in V} \exp(u_j^T v_c)\, u_j^T}{\sum_{i \in V} \exp(u_i^T v_c)}$$

$$= u_o^T - \sum_{j \in V} \underbrace{\left[\frac{\exp(u_j^T v_c)}{\sum_{i \in V} \exp(u_i^T v_c)}\right]}_{P(w_j|w_c)} u_j$$

$$\boxed{\frac{\partial \log P(w_o|w_c)}{\partial v_c} = u_o^T - \sum_{j \in V} P(w_j|w_c)\, u_j}$$

→ Requires the conditional probabilities of all words in the dictionary

3. Interpretation, How driving it to zero improves the learned vector v_c

$$\frac{\partial \log P(w_o|w_c)}{\partial v_c} = u_o^T - \sum_{j \in V} P(w_j|w_c)\, u_j = 0$$

u_o^T → Vector representation of the context word

$\sum_{j \in V} P(w_j|w_c)\, u_j$ → Estimated Representation

Driving the gradient towards to zero means that the estimated context word given the center word match the current vector representation (u_o) with the estimated. It improves the learned vector because we are using the conditional probability to estimate a context word given a center word.

4. Compotational complexity, if vocab size is large?

$$\Delta = u_o^T - \sum_{j \in V} \left[\frac{\overbrace{\exp(u_j^T v_c)}^{Vd}}{\underbrace{\sum_{i \in V} \exp(u_i^T v_c)}_{V.V.d}} \right] u_j$$

→ Num: $V.d$

→ denom: $V^2 d$ $\quad O(V^2 d)$

→ Exponentials: $V + V^2 \rightarrow O(V^2)$
$O(1)$

The most expensive part is the double summation over the vocabulary with time complexity $O(V^2 d)$.

The vocabulary size is dominant in the compotatinal complexity. If it's large then the complexity will be higher.

15.2.1 Negative sampling

$P(D=1 \mid w_c, w_o) = \sigma(u_o^T v_c)$ → Probability that w_o comes from the context window given w_c.

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

→ S: event w_o comes from contex window given w_c

→ N_K: event noise word w_K doen't come from context window of w_c

Joint Probability:

$$\prod_{t=1}^{T} \prod_{-m \leq j \leq m} P(w^{(t+j)} \mid w^{(t)})$$

$$P(w^{(t+j)} \mid w^{(t)}) = P(D=1 \mid w^{(t)}, w^{(t+j)}) \prod_{K=1,\, w_K \sim P(w)}^{K} P(D=0 \mid w^{(t)}, w_K)$$

5. Alternative loss function

The new loss function change the approach, from predicting the probability of the target word from the vocabulary to distinguish the target word from randomly chosen negative words, this means words that are no present in the context window. The idea is learn to differentiate context words from noise words.

6. Gradient negative sampling loss

$$-\log P(w^{(t+j)} \mid w^{(t)}) = -\log P(D=1 \mid w^t, w^{t+j}) - \sum_{K=1}^{K} \log P(D=0 \mid w^t, w_K)$$

$$= -\log \sigma(u_{i_{t+j}}^T v_{i_t}) - \sum_{K=1}^{K} \log(1 - \sigma(u_{n_K}^T v_{i_t}))$$

$$= -\log \sigma(u_{i_{t+j}}^T v_{i_t}) - \sum_{K=1}^{K} \log \sigma(-u_{n_K}^T v_{i_t})$$

$$\frac{d \log P(w_o \mid w_c)}{d v_c} = \frac{d}{d v_c}\left[\log \sigma(u_o^T v_c) + \sum_{\substack{K=1 \\ w_K \sim P(w)}}^{K} \log \sigma(-u_{n_K}^T v_c)\right]$$

$$= \frac{\cancel{\sigma(u_o^T v_c)}(1 - \sigma(u_o^T v_c)) u_o}{\cancel{\sigma(u_o^T v_c)}} + \sum_{K=1}^{K} \frac{\cancel{\sigma(-u_c^T v_{n_i})}(1 - \sigma(-u_{n_K}^T v_c))}{\cancel{\sigma(-u_c^T v_{n_i})}} (-u_{n_K})$$

$$= (1 - \sigma(u_o^T v_c)) u_o + \sum_{K=1}^{K} (\sigma(-u_{n_K}^T v_c) - 1) u_{n_K}$$

$$\boxed{\frac{d \log P(w_o \mid w_c)}{d v_c} = (1 - \sigma(u_o^T v_c)) u_o + \sum_{K=1}^{K} (\sigma(-u_{n_K}^T v_c) - 1) u_{n_K}}$$