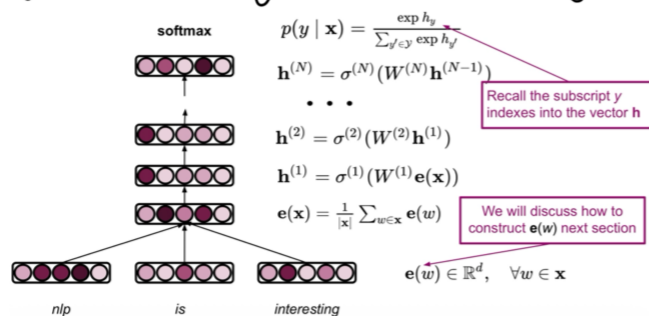


3. MLP and Sentiment Analysis

Multilayer Perceptions

$h^{(N)} = \sigma^{(N)}(W^{(N)} \dots \sigma^{(2)}(W^{(2)} \sigma^{(1)}(W^{(1)} e(x))))$, $W^{(i)} \in \mathbb{R}^{d_{i+1} \times d_i}$
 $\mathbb{R}^{1 \times 1}$ activation function final layer $\mathbb{R}^{1 \times 1 \times d_N}$ first layer $\mathbb{R}^{d_1 \times d_1}$ encodes input x $e(x) = \frac{1}{|x|} \sum_{w \in x} e(w)$ mean pooling
 $p(y|x) = \frac{\exp(h_y)}{\sum_{y' \in Y} \exp(h_{y'})} = \text{softmax}(h^{(N)}, y)$ ← This is the MLP

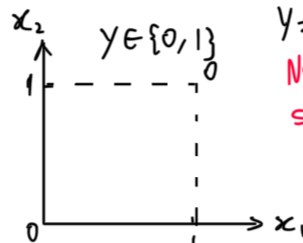
- A MLP is a log-linear model where we also learn the feature function f .
- The final layer is generally a softmax.
- Maximizing log-likelihood with gradient methods, using back-propagation.
- Structure:



* The perceptron is a log-linear model with a specific setting of hyperparameters.
A special case for log-linear models.

XOR Problem

x_1	x_2	$y = x_1 \text{ XOR } x_2$
0	0	0
0	1	1
1	0	1
1	1	0



$y = \alpha_1 x_1 + \alpha_2 x_2$ * A single-layer MLP cannot solve XOR problem.
Not linearly separable!

But activations help - non-linearity.

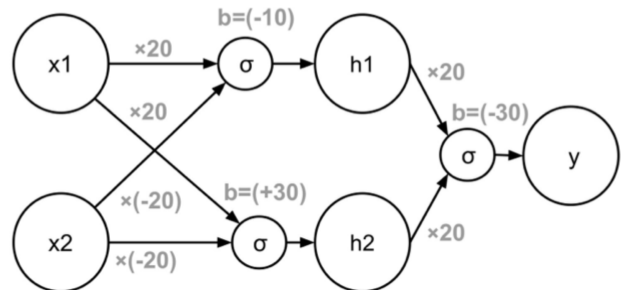
$\tanh(x) = 2\sigma(2x) - 1$

$y = (-1)x_1 + (-1)x_2 + (2)\tanh(x_1 + x_2)$

⇒

x_1	x_2	y
0	0	$\lceil (-1) \cdot 0 + (-1) \cdot 0 + (2) \cdot \tanh(0) \rceil \approx 0$
0	1	$\lceil (-1) \cdot 0 + (-1) \cdot 1 + (2) \cdot \tanh(1) \rceil \approx 1$
1	0	$\lceil (-1) \cdot 1 + (-1) \cdot 0 + (2) \cdot \tanh(1) \rceil \approx 1$
1	1	$\lceil (-1) \cdot 1 + (-1) \cdot 1 + (2) \cdot \tanh(2) \rceil \approx 0$

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



Skip-gram Model

Preprocessing (Example)

Pre-process with a window of size 3. Consider both left and right context.



Training Data

(super, NLP)
 (super, is)
 (super, a)
 (super, fun)
 (super, lecture!)

- Result is a collection of N word-context pairs
- Window size k , with N is $O(k \cdot C)$, C is corpus size.

(super, fun)
(super, lecture)
(super, !)

Model

A log-bilinear model. Ex: $f(x, y) = xy$.

$$p(c | w) = \frac{1}{z(w)} \exp(e_{\text{word}}(w) \cdot e_{\text{ctx}}(c))$$

$\mathbb{R}^d$, embedding
from word w

$\mathbb{R}^d$, embedding
from context c .

- * Every word can appear as a word or context.
- * If we have $|V|$ words, then $\geq |V|d$ parameters.

Estimation

Maximize the log-likelihood. Using gradient (backpropagation) on $e_{\text{word}}(w)$ and $e_{\text{ctx}}(c)$.

$$\sum_{n=1}^N \log p(c^{(n)} | w^{(n)}) = \sum_{n=1}^N (e_{\text{word}}(w^{(n)}) \cdot e_{\text{ctx}}(c^{(n)}) - \log z(w^{(n)}))$$

Output

Two collections of word embeddings *Non-one-hot representations*

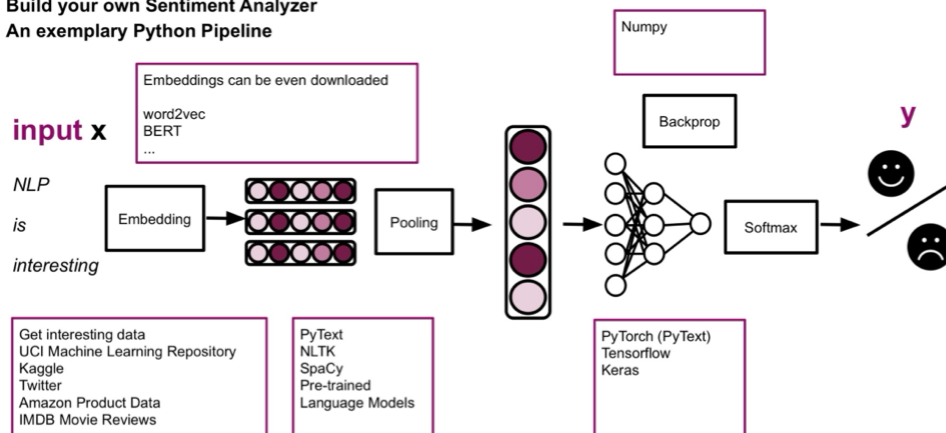
- $\{e_{\text{word}}(w)\}_{w \in V}$
- $\{e_{\text{ctx}}(w)\}_{w \in V}$ *← throw away or concatenate to form an embedding of $\mathbb{R}^{2d}$.*
- Where V is the set of word types in the corpus.

Sentiment Analysis

Classifying utterances according to how they make the interlocutor feel. Ex: movie review.

- Polarity: identifying the sentiment orientation.
- Subjectivity: expression of personal feelings

Build your own Sentiment Analyzer
An exemplary Python Pipeline



Exercise Q2

(b) $h = \text{ReLU}(W^{(1)} \cdot x + b^{(1)})$, $\hat{y} = W^{(2)} \cdot h + b^{(2)}$, $x \in \mathbb{R}^2$. Give a set of $W^{(1)}$, $W^{(2)}$, $b^{(1)}$, and $b^{(2)}$ such that this model can accurately model the XOR problem.

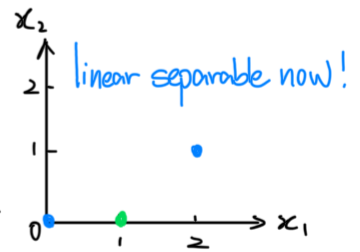
$\begin{bmatrix} x_1 & x_2 \\ 1 & 1 \end{bmatrix}$

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}_{4 \times 2}, \quad W^{(1)} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}_{2 \times 2}, \quad b^{(1)} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}_{2 \times 1}$$

$$\Rightarrow XW = \begin{bmatrix} 0 \cdot 1 + 0 \cdot 1 = 0 & 0 \cdot 1 + 0 \cdot 1 = 0 \\ 0 \cdot 1 + 1 \cdot 1 = 1 & 0 \cdot 1 + 1 \cdot 1 = 1 \\ 1 \cdot 1 + 0 \cdot 1 = 1 & 1 \cdot 1 + 0 \cdot 1 = 1 \\ 1 \cdot 1 + 1 \cdot 1 = 2 & 1 \cdot 1 + 1 \cdot 1 = 2 \end{bmatrix}_{4 \times 2} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix}_{4 \times 2}$$

$$\Rightarrow XW + b^{(1)} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix}_{4 \times 2} + \begin{bmatrix} 0 & -1 \\ 0 & -1 \\ 0 & -1 \\ 0 & -1 \end{bmatrix}_{4 \times 2} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}_{4 \times 2}$$

$$h = \text{ReLU}(W^{(1)} \cdot x + b^{(1)}) = \text{ReLU} \left(\begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}_{4 \times 2} \right) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}_{4 \times 2}$$



$$W^{(2)} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}_{2 \times 1}, \quad b^{(2)} = 0$$

$$\hat{y} = W^{(2)} \cdot h + b^{(2)} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}_{4 \times 2} \times \begin{bmatrix} 1 \\ -2 \end{bmatrix}_{2 \times 1} + \mathbf{0} = \begin{bmatrix} 0 \cdot 1 + 0 \cdot (-2) = 0 \\ 1 \cdot 1 + 0 \cdot (-2) = 1 \\ 1 \cdot 1 + 0 \cdot (-2) = 1 \\ 2 \cdot 1 + 1 \cdot (-2) = 0 \end{bmatrix}_{4 \times 1} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}_{4 \times 1}$$

Homework | Question 1 (c)

Show that a single-layer MLP cannot solve XOR problem.

$x_1 \quad x_2 \quad y = x_1 \text{ XOR } x_2$

0	0	0
0	1	1
1	0	1
1	1	0

Define a linear boundary as $y = \alpha_1 x_1 + \alpha_2 x_2 + b$

$$y = \begin{cases} \alpha_1 x_1 + \alpha_2 x_2 + b \geq 0 \Rightarrow y = 1 \\ \alpha_1 x_1 + \alpha_2 x_2 + b < 0 \Rightarrow y = 0 \end{cases}$$

$x_1 \quad x_2 \quad y = \alpha_1 x_1 + \alpha_2 x_2$

0	0	$\alpha_1 \cdot 0 + \alpha_2 \cdot 0 + b < 0 \Rightarrow y = 0$
0	1	$\alpha_1 \cdot 0 + \alpha_2 \cdot 1 + b \geq 0 \Rightarrow y = 1$
1	0	$\alpha_1 \cdot 1 + \alpha_2 \cdot 0 + b \geq 0 \Rightarrow y = 1$

$b < 0$

$$\alpha_1 \cdot 0 + \alpha_2 \cdot 1 + b + \alpha_1 \cdot 1 + \alpha_2 \cdot 0 + b \geq 0$$

|

|

$$\alpha_1 + \alpha_2 + b < 0 \rightarrow y = 0$$

$$\alpha_1 + \alpha_2 < -b$$

$$\Rightarrow \alpha_2 + \alpha_1 + 2b \geq 0$$

$$\Rightarrow \alpha_2 + \alpha_1 + 2b < -b + 2b$$

$$\Rightarrow \alpha_2 + \alpha_1 + 2b < b$$

$$\Rightarrow b \leq \alpha_2 + \alpha_1 + 2b < b$$

$$\Rightarrow b > 0$$

???

Not possible to satisfy both!