Naive Bayes example: $|V| = 20$ $P(w|-) = 14$ $P(w|+) = 9$

Training
- just plain boring
- entirely predictable and lacks energy
- no surprises and very few surprises
+ very powerful
+ the most fun film of the summer

Testing ? predictable ~~with~~ no fun

Training Prior: $P(-) = 3/5$ $P(+) = 2/5$

$$\hat{P}(c_j) = \frac{N_{c_j}}{N_{total}}$$

· drop "with"

Likelihoods from training:

$$P(w_i|c) = \frac{count(w_i, c) + 1}{\left(\sum_{w \in V} count(w, c)\right) + |V|}$$

$$P(\text{"predictable"}|-) = \frac{1+1}{14+20} = \frac{2}{34} = \frac{1}{17}$$

$$P(\text{"predictable"}|+) = \frac{0+1}{9+20} = \frac{1}{29}$$

$$P(\text{"no"} \mid -) = \frac{1+1}{14+20} = \frac{2}{34} = \frac{1}{17}$$

$$P(\text{"no"} \mid +) = \frac{0+1}{9+20} = \frac{1}{29}$$

$$P(\text{"fun"} \mid -) = \frac{0+1}{14+20} = \frac{1}{34}$$

$$P(\text{"fun"} \mid +) = \frac{1+1}{9+20} = \frac{2}{29}$$

- scoring from test set

$$P(-)\,P(S \mid -) = \frac{3}{5}\left(\frac{2 \times 2 \times 1}{34^3}\right) = 6.1 \times 10^{-5}$$

$$P(+)\,P(S \mid +) = \frac{2}{5}\left(\frac{1 \times 1 \times 2}{29^3}\right) = 3.2 \times 10^{-5}$$

Logistic Regression

Sigmoid function:

$$y = s(z) = \frac{1}{1+e^{-z}}$$

$$\sigma(w \cdot x + b)$$

Set of classes: $\{+, -\}$

Vector x of features: $[x_1, x_2, \ldots, x_n]$

$x_1 = \text{count}(\text{"happy"})$

$x_2 = \log(\text{number of words in review})$

Vector w of weights: $[w_1, w_2, \ldots, w_n]$

- w_i for each feature f_i

$$P(y=1) = \sigma(x \cdot w + b)$$

$$= \frac{1}{1+e^{-(x \cdot w + b)}}$$

- Loss function: cross-entropy loss
- optimization algo: Stochastic gradient descent

$$L_{CE}(\hat{y}, y) = -\log p(y|x)$$

$$= -[y \log \hat{y} + (1-y) \log(1-\hat{y})]$$

plug in sigmoid for y

$$L_{CE}(\hat{y}, y) = -[y \log \sigma(w \cdot x + b) + (1-y) \log(1 - \sigma(w \cdot x + b))]$$

- want to minimize

example:

Document: "The food was absolutely delightful, and the service was outstanding"

positive words : ['absolutely', 'delightful', 'outstanding']

negative words: [' ']

feature vector rep(x) : [3, 0]

weights (w) : [0.5, -0.4]
bias (b) : -0.1

$$\sigma(w \cdot x + b) = \sigma([0.5, -0.4] \cdot [3, 0] - 0.1) = \sigma(1.5 - 0.1)$$

$$= \sigma(1.4)$$

$$P(y=1) = \frac{1}{1 + e^{-1.4}} = 0.802 \quad \text{(positive class score)}$$

$$P(y=0) = 1 - 0.802 = 0.198 \quad \text{(negative class score)}$$

$$L_{CE}(\hat{y}, y) = L_{CE}(.802, 1) = -[1 \cdot \log(.802) + 0]$$

when the label is 'positive'

$$= -[\ln(.802)] = \boxed{.221}$$

↑ using ln

update weights & bias using g.d. update rule.

$$\theta_{t+1} = \theta_t - \eta \nabla L(f(x;\theta), y) \quad \text{where} \quad \eta = 0.1 \text{ learn rate}$$

$$\frac{\partial L_{CE}(\hat{y}, y)}{\partial w_j} = [\sigma(w \cdot x + b) - y] x_j$$

$$\theta' = \begin{bmatrix} w_1 \\ w_2 \\ b \end{bmatrix} \qquad \sigma(1.4) = .802$$

$$\Delta W = \begin{bmatrix} [\sigma(1.4) - 1] x_1 \\ [\sigma(1.4) - 1] x_2 \\ [\sigma(1.4) - 1] b \end{bmatrix} = \begin{bmatrix} (.802 - 1)\, 0.5 \\ (.802 - 1)(-0.4) \\ (.802 - 1)\, 0.1) \end{bmatrix}$$

(0.802)

$$\Delta W = \begin{bmatrix} -0.099 \\ 0.079 \\ 0.02 \end{bmatrix}$$

$$\theta_{t+1} = \theta_t - \eta \Delta w$$

$$= \begin{bmatrix} 0.5 \\ -0.4 \\ -0.1 \end{bmatrix} - 0.1 \begin{bmatrix} -0.099 \\ 0.079 \\ 0.02 \end{bmatrix}$$

$$= \begin{bmatrix} 0.5 \\ -0.4 \\ -0.1 \end{bmatrix} - \begin{bmatrix} -0.0099 \\ 0.0079 \\ +0.002 \end{bmatrix} = \boxed{\begin{bmatrix} 0.510 \\ -0.408 \\ -0.102 \end{bmatrix}} = \theta_{t+1}$$

updated weights, bias

Vector semantics

- word2vec
  - dense vectors
  - training a classifier to predict whether a word is likely to appear nearby.

- Term matrix
- Dot product

word2vec example:

Joy (target): [0.2, 0.1, 0.3]

Happy (positive pair): [0.1, 0.3, 0.2]

course (negative pair): [0.2, 0.2, 0.1]

exam (negative pair): [0.3, 0.1, 0.2]

- update word vectors using one step of skip-gram with negative sampling algo.
- Learning rate (α) is 0.1

- compute dot product for positive pair and negative pairs

Joy · Happy = .02 + .03 + .06 = .11

Joy · course = .04 + .02 + .03 = .09

Joy · exam = .06 + .01 + .06 = .13

· Compute cross-entropy loss

$$L_{CE} = -\left[\log \sigma(c_{pos} \cdot w) + \sum_{i=1}^{K} \log \sigma(-c_{neg_i} \cdot w)\right]$$

$$= -\left[\log(\sigma(.11)) + \left[\log(\sigma(0.09)) + \log(\sigma(-0.13))\right]\right]$$

$$= -\left[\ln(0.527) + \left[\ln(0.478) + \ln(0.468)\right]\right]$$

$$= -\left[-0.641 + (-1.497)\right] = 2.14$$

: compute gradients for the target word.

$$\frac{d L_{CE}}{dw} = \left[\sigma(c_{pos} \cdot w) - 1\right] c_{pos} + \sum_{i=1}^{K} \left[\sigma(c_{neg_i} \cdot w)\right] c_{neg_i}$$

$$\Delta w = \left[\sigma(.11) - 1\right][0.1, 0.3, 0.2] + \Big(\sigma(.09)[0.2, 0.2, 0.1] + \sigma(.13)[0.3, 0.1, 0.2]\Big)$$

$$= \left[.527[0.1, 0.3, 0.2] + \Big(.478[0.2, 0.2, 0.1] + .468[0.3, 0.1, 0.2]\Big)\right.$$

$$= [.053, .158, .105] + \Big([.096, .096, .048] + [.140, -.047, .094]\Big)$$

$$= [.053, .158, .105] + \Big([0.236, 0.143, 0.142]\Big)$$

$$\Delta w = [0.289, 0.301, 0.247]$$

- Update target word vector using computed gradients. What is new rep vector for joy with learning rate of 0.1?

w_{joy}: [0.2, 0.1, 0.3]

ΔW: [0.289, 0.301, 0.247]

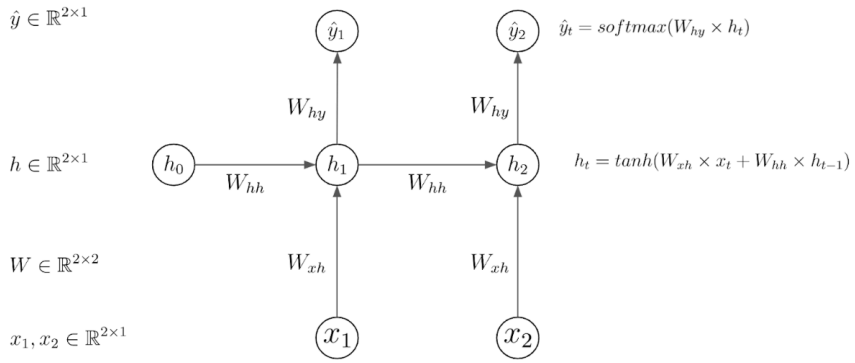
$$W_{t+1} = W_t - \eta \Delta W$$

$$= \begin{bmatrix} 0.2 \\ 0.1 \\ 0.3 \end{bmatrix} - 0.1 \begin{bmatrix} 0.289 \\ 0.301 \\ 0.247 \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.1 \\ 0.3 \end{bmatrix} - \begin{bmatrix} 0.0289 \\ 0.0301 \\ 0.0247 \end{bmatrix}$$

$$= \begin{bmatrix} 0.171 \\ 0.070 \\ 0.275 \end{bmatrix}$$

Recurrent Neural Networks example (practice)

Consider a simple RNN with one hidden layer. It receives input vectors from a sequence of two time steps. Cross entropy is used as the loss function.



Given the following values:

$$x_1 = \begin{bmatrix}0.1\\0.2\end{bmatrix},\ x_2 = \begin{bmatrix}0.3\\0.4\end{bmatrix},\ y_1 = \begin{bmatrix}y_1^1\\y_1^2\end{bmatrix} = \begin{bmatrix}1\\0\end{bmatrix},\ y_2 = \begin{bmatrix}0\\1\end{bmatrix},\ h_0 = \begin{bmatrix}0\\0\end{bmatrix}$$

$$w_{xh} = \begin{bmatrix}0.1 & 0.1\\0.2 & 0.2\end{bmatrix},\ w_{hh} = \begin{bmatrix}0.3 & 0.3\\0.4 & 0.4\end{bmatrix},\ w_{hy} = \begin{bmatrix}0.5 & 0.5\\0.6 & 0.6\end{bmatrix}$$

1. (2 points) Write the formula for the tanh activation function, and its derivatives.

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)}$$

$$\frac{d}{dx}\tanh(x) = \frac{\cosh(x)\cdot\cosh(x) - \sinh(x)(-\sinh(x))}{\cosh^2(x)}$$

$$= \frac{\cosh^2(x) + \sinh^2(x)}{\cosh^2(x)}$$

$$= 1 + \tanh^2(x) = 1 + \left(\frac{e^x - e^{-x}}{e^x + e^{-x}}\right)^2$$

- compute forward pass for this sequence.

At $t=1$:

$$h_1 = \tanh(W_{xh} \cdot x(1) + W_{hh} \cdot h(0)) = \tanh\left(\begin{bmatrix} 0.1 & 0.1 \\ 0.2 & 0.2 \end{bmatrix} \cdot \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} + \begin{bmatrix} 0.3 & 0.3 \\ 0.4 & 0.4 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right)$$

$$= \tanh\begin{pmatrix} 0.03 \\ 0.06 \end{pmatrix} = \begin{bmatrix} 0.0299 \\ 0.0599 \end{bmatrix} = h_1$$

$$\hat{y}_1 = \text{Softmax}(W_{hy} \cdot h(1)) = \text{Softmax}\left(\begin{bmatrix} 0.5 & 0.5 \\ 0.6 & 0.6 \end{bmatrix} \cdot \begin{bmatrix} 0.0299 \\ 0.0599 \end{bmatrix}\right)$$

$$= \text{Softmax}\left(\begin{bmatrix} 0.0449 \\ 0.0539 \end{bmatrix}\right)$$

$$= \begin{bmatrix} e^{0.0449} / (e^{0.0449} + e^{0.0539}) \\ e^{0.0539} / (e^{0.0449} + e^{0.0539}) \end{bmatrix}$$

$$\hat{y}_1 = \begin{bmatrix} 0.498 \\ 0.502 \end{bmatrix}$$

At $t=2$:

$$h_2 = \tanh(W_{xh} \cdot x(2) + W_{hh} \cdot h(1))$$

$$= \tanh\left(\begin{bmatrix} 0.1 & 0.1 \\ 0.2 & 0.2 \end{bmatrix} \cdot \begin{bmatrix} 0.3 \\ 0.4 \end{bmatrix} + \begin{bmatrix} 0.3 & 0.3 \\ 0.4 & 0.4 \end{bmatrix} \cdot \begin{bmatrix} 0.0299 \\ 0.0599 \end{bmatrix}\right)$$

$$= \tanh\left(\begin{bmatrix} 0.07 \\ 0.14 \end{bmatrix} + \begin{bmatrix} 0.02694 \\ 0.03592 \end{bmatrix}\right)$$

$$= \tanh\left(\begin{bmatrix} 0.09694 \\ 0.17592 \end{bmatrix}\right) = \begin{bmatrix} 0.0966 \\ 0.174 \end{bmatrix} = h_2$$

$$\hat{y}_2 = \text{softmax}(W_{hy} \cdot h_2)$$

$$= \text{softmax}\left(\begin{bmatrix} 0.5 & 0.5 \\ 0.6 & 0.6 \end{bmatrix} \cdot \begin{bmatrix} 0.0966 \\ 0.174 \end{bmatrix}\right)$$

$$= \text{softmax}\left(\begin{bmatrix} 0.1353 \\ 0.1624 \end{bmatrix}\right)$$

$$= \begin{bmatrix} e^{.1353} / (e^{.1353} + e^{.1624}) \\ e^{.1624} / (e^{.1353} + e^{.1624}) \end{bmatrix} = \begin{bmatrix} 0.493 \\ 0.507 \end{bmatrix} = \hat{y}_2$$

- Compute total loss for this seq.

Loss at timestep t: $L_t = -\sum_{i=1}^{2} y_t^i \cdot \log(\hat{y}_t^i)$

Total loss: $L = \sum L_t$

I used natural log (but doesn't matter).

$$L_1 = -\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \ln \begin{bmatrix} .498 \\ .502 \end{bmatrix}\right) = .697$$

$$L_2 = -\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix} \ln \begin{bmatrix} .493 \\ .507 \end{bmatrix}\right) = .679$$

$$L = 0.697 + 0.679 = \underline{1.376}$$