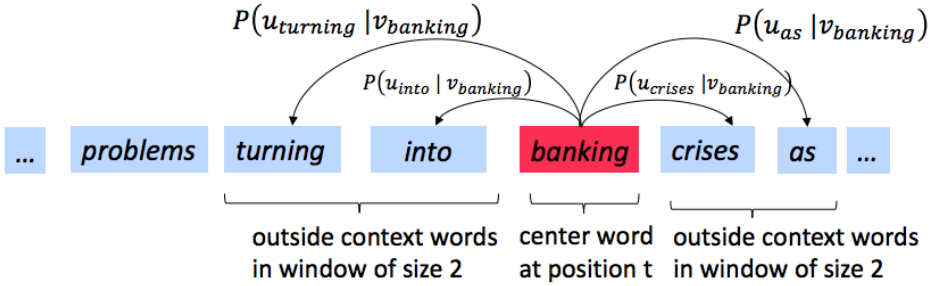
1 手写作业：理解 word2vec

还记得吗？word2vec 背后的核心想法：“一个单词的意义取决于其搭配”，具体来说，对一个中心词 c，它被前后特定长度的上下文围绕。我们将这个上下文窗口中的单词定义为外部词('outside words')。举个例子，在图 1 中，上下文窗口长度为 2，中心词 c 是'banking'，外部词就是'turning', 'into', 'crises'和'as'。



Skip-gram word2vec 的目标是学习一个概率分布 $P(O|C)$。特殊一点的话，给定一个特定的单词 o 和一个特定的单词 c，我们想要预测 $P(O=o|C=c)$，即单词 o 是单词 c 的外部词的概率(单词 o 落在 c 的上下文窗口里)。我们将通过在一系列向量点积上求 softmax 函数对这个概率建模。

$$P(O=o \mid C=c)=\frac{\exp(\boldsymbol{u}_o^{\top}\boldsymbol{v}_c)}{\sum_{w\in \text{Vocab}}\exp(\boldsymbol{u}_w^{\top}\boldsymbol{v}_c)} \tag{1}$$

对每一个单词，我们学习向量 u 和 v，其中 u_o 是代表外部词 o 的外部向量，v_c 是代表中心词的中心向量。我们将这些参数存储在两个矩阵 U 和 V 当中。U 的各列是所有的外部向量 u_w，V 的各列是所有的中心向量 v_w。U 和 V 都包含词汇表中每一个单词 w 对应的向量。

还记得吗？对于一对单词 c 和 o，损失值由以下表达式给出：

$$\boldsymbol{J}_{\text{naive-softmax}}(\boldsymbol{v}_c, o, \boldsymbol{U}) = -\log P(O = o|C = c) \tag{2}$$

我们可以将这个损失视为真实分布$\boldsymbol{y}$和预测分布$\hat{\boldsymbol{y}}$的交叉熵损失，对于一个特定的中心词 c 和一个特定的外部词 o。在此处，$\boldsymbol{y}$和$\hat{\boldsymbol{y}}$都是长度等于词汇表中单词数的向量。而且，这些向量的第 k 个分量表示第 k 个单词成为给定中心词 c 的外部词的条件概率。真正的经验分布$\boldsymbol{y}$是一个独热向量，在真正的外部词对应的分量上的值为 1，其他值为 0。预测分布$\hat{\boldsymbol{y}}$是方程(1)中的模型给出的概率分布$P(O|C = c)$。

注意：在整个作业中，当计算导数时，请使用课程中复习到的方法(不要使用泰勒级数近似)

(a) 请证明方程 2 中的朴素 softmax 损失和$\boldsymbol{y}$-$\hat{\boldsymbol{y}}$交叉熵损失等价(注意$\boldsymbol{y}$和$\hat{\boldsymbol{y}}$都是向量，而$\hat{\boldsymbol{y}}_o$是标量)

$$-\sum_{w \in \text{Vocab}} \boldsymbol{y}_w \log(\hat{\boldsymbol{y}}_w) = -\log(\hat{\boldsymbol{y}}_o).$$

My answer: $\boldsymbol{y}_w = 1 \; if \; w = o \; else \; \boldsymbol{y}_w = 0$

(b)

① 请计算$J_{naive-softmax}(v_c, o, U)$关于$v_c$的偏导数，结果以向量形式表示

$$\frac{\partial J_{naive-softmax}(\boldsymbol{v}_c,\ o,\ U)}{\partial \boldsymbol{v}_c} = -\left[\boldsymbol{u}_o - \sum_{x \in Vocab} \frac{\exp(\boldsymbol{u}_x^T \boldsymbol{v}_c)}{\sum_{w \in Vocab} \exp(\boldsymbol{u}_w^T \boldsymbol{v}_c)} \boldsymbol{u}_x\right]$$
$$= -[\boldsymbol{u}_o - E[\boldsymbol{u}_w]]$$

② 什么情况下计算出的梯度等于 0?

My answer: $\boldsymbol{u}_o = E[\boldsymbol{u}_w]$，即模型对外部词向量的估计值等于外部词向量的实际值时计算出的梯度等于 0。

③ 你找到的梯度是两项的差，当词向量v_c减去梯度时请对梯度中的每一项如何改进词向量进行解释说明

My answer: 负梯度的值为$\boldsymbol{u}_o - E[\boldsymbol{u}_w]$，表示参数的更新将使得模型对外部词向量的估计更接近真实观测到的外部词向量

④ 在许多使用词向量的下游应用中，使用的是 L2 正则化向量而非它们的原始形式。现在，假定你想要将词组分类成正例或者反例。什么时候 L2 正则化向量会从下游任务提取有效信息？什么时候不会？提示：考虑当单词$x \neq y$，存在标量α使得$\boldsymbol{u}_x = \alpha \boldsymbol{u}_y$的情况

My answer: 当不同单词对应的词向量共线时，L2 正则化后不同的单词将对应相同的词向量表示，此时无法从下游任务中提取有效信息；而当这种情况不存在或在小部分单词上存在时可以从下游任务中提取有效信息。

(c) 请计算$J_{naive-softmax}(v_c, o, U)$关于每一个外部词向量$u_w$的偏导数。这将会有两种情况：当 w=o 时，对真正的外部词向量，当 w≠o 时，对所有其他单词。请写出关于$\boldsymbol{y}$，$\hat{\boldsymbol{y}}$和$\boldsymbol{v}_c$的答案。在这个子部分，你可能会用到这些项中特定的元素，比如$y_1, y_2, \ldots$，请注意u_w是一个向量而$y_1, y_2, \ldots$是标量。

My answer:

$$\frac{\partial J_{naive-softmax}(\boldsymbol{v}_c, o, U)}{\partial \boldsymbol{u}_w} = \begin{cases} -[\boldsymbol{u}_w - y_w \boldsymbol{v}_c] & (w = o) \\ -[-y_w \boldsymbol{v}_c] & (w \neq o) \end{cases}$$

$$y_w = \frac{\exp(\boldsymbol{u}_w^T \boldsymbol{v}_c)}{\sum_{w \in Vocab} \exp(\boldsymbol{u}_w^T \boldsymbol{v}_c)}$$

(d) 请写出$J_{naive-softmax}(v_c, o, U)$关于$U$的偏导数。请将答案分解成各个列向量的组合。这里不需要用到具体的导数，只要矩阵形式的答案即可。

My answer:

$$\frac{\partial J_{naive-softmax}(\boldsymbol{v}_c, o, U)}{\partial U} = \boldsymbol{v}_c \hat{\boldsymbol{y}}^T - \boldsymbol{u}_o \boldsymbol{y}$$

(e) Leaky ReLU 激活函数通过方程 4 和图 2 给出，

$$f(x) = \max(\alpha x, x) \tag{4}$$

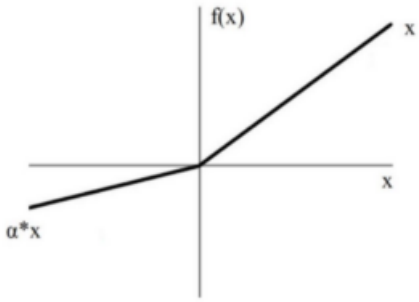


Figure 2: Leaky ReLU

x 是一个标量，$0<\alpha<1$，请计算 f(x)关于 x 的导数，你可以忽略在 0 处导数未定义的情况。

My answer:

$$\frac{df}{dx}=\begin{cases}1 & (x>0)\\ \alpha & (x<0)\end{cases}$$

(f) sigmoid 函数由方程 5 给出：

$$\sigma(x)=\frac{1}{1+e^{-x}}=\frac{e^{x}}{e^{x}+1} \tag{5}$$

请计算$\sigma(x)$关于 x 的导数，x 是一个标量。请写出关于$\sigma(x)$的答案。

My answer:

$$\sigma'(x)=\frac{e^{-x}}{(1+e^{-x})^{2}}=\sigma(x)\,(1-\sigma(x))$$

(g) 现在我们考虑负采样损失，这是朴素 softmax 损失的一个选项。假定从词汇表中抽取 K 个负样本。为了方便表示我们记为 $w_1, w_2, \ldots, w_K$，并将外部词向量记为$\boldsymbol{u}_{w_1}, \boldsymbol{u}_{w_2}, \ldots, \boldsymbol{u}_{w_K}$。对这个问题，假定 K 个负样本相异，即$i\neq j$，有$w_i\neq w_j$，对$i, j\in\{1,2,\ldots,K\}$。

请注意$o \notin \{w_1, w_2, ..., w_K\}$。对于一个中心词 c 和一个外部词 o，负采样损失函数通过下式给出：

$$\boldsymbol{J}_{\text{neg-sample}}(\boldsymbol{v}_c, o, \boldsymbol{U}) = -\log(\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c)) - \sum_{s=1}^{K} \log(\sigma(-\boldsymbol{u}_{w_s}^\top \boldsymbol{v}_c)) \tag{6}$$

① 请重复(b), (c)，计算$J_{neg-sample}$分别关于$\boldsymbol{v}_c$，$\boldsymbol{u}_o$和第 s 个负样本$\boldsymbol{u}_{w_s}$的偏导数。答案关于$\boldsymbol{v}_c$，$\boldsymbol{u}_o$和$\boldsymbol{u}_{w_s}$，$s \in \{1, 2, ..., K\}$。注意：你应该使用(f)中的答案帮助计算这里必需的梯度。

My answer:

$$J_{neg-sample} = -\log(\sigma(\boldsymbol{u}_o^T \boldsymbol{v}_c)) - \sum_{s=1}^{K} \log(\sigma(-\boldsymbol{u}_{w_s}^T \boldsymbol{v}_c))$$

$$\frac{\partial J_{neg-sample}}{\partial \boldsymbol{v}_c} = -[1 - \sigma(\boldsymbol{u}_o^T \boldsymbol{v}_c)]\boldsymbol{u}_o + \sum_{s=1}^{K} [1 - \sigma(-\boldsymbol{u}_{w_s}^T \boldsymbol{v}_c)]\boldsymbol{u}_{w_s}$$

$$\frac{\partial J_{neg-sample}}{\partial \boldsymbol{u}_o} = -[1 - \sigma(\boldsymbol{u}_o^T \boldsymbol{v}_c)]\boldsymbol{v}_c$$

$$\frac{\partial J_{neg-sample}}{\partial \boldsymbol{u}_{w_s}} = [1 - \sigma(-\boldsymbol{u}_{w_s}^T \boldsymbol{v}_c)]\boldsymbol{v}_c$$

② 在课程中，我们学到了反向传播的一种高效实现，这种实现复用了之前计算过的偏导数。你在多大程度上能复用上面计算出的偏导数以最小化计算的冗余程度？请写出关于$U_{o,\{w_1,...,w_K\}} = [u_o, -u_{w_1}, \ldots, -u_{w_K}]$(一个由外部词向量按列放置形成的矩阵)和$1$(一个形状为$(K+1)\times 1$的全 1 向量)的答案，除$U_{o,\{w_1,...,w_K\}}$和$1$之外的其他项也能在答案中使用。

My answer:

$$U_{o,\{w_1,\ldots,w_K\}} = [u_o, -u_{w_1}, \ldots, -u_{w_K}]$$

$$\frac{\partial J_{neg-sample}}{\partial U_{o,\{w_1,\ldots,w_K\}}} = -\boldsymbol{v}_c \cdot [1^T - \sigma(\boldsymbol{v}_c^T U_{o,\{w_1,\ldots,w_K\}})]$$

$$\frac{\partial J_{neg-sample}}{\partial \boldsymbol{v}_c} = -U_{o,\{w_1,\ldots,w_K\}}[1 - \sigma(U_{o,\{w_1,\ldots,w_K\}}^T \boldsymbol{v}_c)]$$

③ 用一句话描述为什么这个损失函数比朴素 softmax 损失在计算上更高效

My answer:

负采样损失函数只在 K 个样本上计算点积，而朴素 softmax 损失函数需要在全部词向量上计算点积。

到目前为止，我们已经研究了通过变量复用和采样近似 softmax 以实现更快的梯度下降。但请注意这些优化的一部分在现代 GPU 上是不必要的，并且在某种程度上，是当时计算资源有限这一背景的产物。

(h) 现在我们将重复之前的练习，但是不使用 K 个采样词相异这一假设。假定从词汇表中抽取 K 个负采样词，为了方便表示我们记为 $w_1, w_2, \ldots, w_K$，并将外部词向量记为 $\boldsymbol{u}_{w_1}, \boldsymbol{u}_{w_2}, \ldots, \boldsymbol{u}_{w_K}$。在这一问题中，你将不再假设这些词相异。换言之，当 $i \neq j$ 时，仍然可能有 $w_i = w_j$，请注意 $o \notin \{w_1, \ldots, w_K\}$。对一个中心词 c 和一个外部词 o，负采样损失函数由下式给出：

$$\boldsymbol{J}_{\text{neg-sample}}(\boldsymbol{v}_c, o, \boldsymbol{U}) = -\log(\sigma(\boldsymbol{u}_o^\top \boldsymbol{v}_c)) - \sum_{s=1}^{K} \log(\sigma(-\boldsymbol{u}_{w_s}^\top \boldsymbol{v}_c)) \tag{7}$$

请计算$J_{neg-sample}$关于$\boldsymbol{u}_{w_s}$的偏导数，请写出关于$\boldsymbol{v}_c$和$\boldsymbol{u}_{w_s}$的答案，$s \in \{1, 2, ..., K\}$。提示：将损失函数的和式分解为两个部分的和：一部分和基于所有等于w_s的采样词，另一部分和基于所有不等于w_s的采样词。为方便表示，你可以在求和符号下写“等于”和“不等于”的条件。

My answer:

$$\begin{aligned} J_{neg-sample} &= -\log(\sigma(\boldsymbol{u}_o^T \boldsymbol{v}_c)) - \sum_{s=1}^{K} \log(\sigma(-\boldsymbol{u}_{w_s}^T \boldsymbol{v}_c)) \\ &= -\log(\sigma(\boldsymbol{u}_o^T \boldsymbol{v}_c)) - \sum_{w_s} \log(\sigma(-\boldsymbol{u}_{w_s}^T \boldsymbol{v}_c)) - \sum_{\neq w_s} \log(\sigma(-\boldsymbol{u}_{w_s}^T \boldsymbol{v}_c)) \end{aligned}$$

$$\frac{\partial J_{neg-sample}}{\partial \boldsymbol{u}_{w_s}} = \sum_{w_s} [1 - \sigma(-\boldsymbol{u}_{w_s}^T \boldsymbol{v}_c)] \boldsymbol{v}_c$$

(i) 假定中心词$c = w_t$，上下文窗口是$[w_{t-m}, ..., w_{t-1}, w_t, w_{t+1}, ..., w_{t+m}]$，其中 m 是上下文窗口的大小。还记得吗？在 skip-gram 版本的 word2vec 中整个上下文窗口的总损失为

$$\boldsymbol{J}_{\text{skip-gram}}(\boldsymbol{v}_c, w_{t-m}, \ldots w_{t+m}, \boldsymbol{U}) = \sum_{\substack{-m \le j \le m \\ j \neq 0}} \boldsymbol{J}(\boldsymbol{v}_c, w_{t+j}, \boldsymbol{U})$$

在这里，$J(v_c, w_{t+j}, U)$代表关于中心词$c = w_t$和外部词 w_{t+j}的任意一个损失项，$J(v_c, w_{t+j}, U)$可以是$J_{naive-softmax}(v_c, w_{t+j}, U)$或者$J_{neg-sample}(v_c, w_{t+j}, U)$，取决于你的实现。

请写出以下三个偏导数：

(i) $\frac{\partial \boldsymbol{J}_{\text{skip-gram}}(\boldsymbol{v}_c, w_{t-m}, \ldots w_{t+m}, \boldsymbol{U})}{\partial \boldsymbol{U}}$

(ii) $\frac{\partial \boldsymbol{J}_{\text{skip-gram}}(\boldsymbol{v}_c, w_{t-m}, \ldots w_{t+m}, \boldsymbol{U})}{\partial \boldsymbol{v}_c}$

(iii) $\frac{\partial \boldsymbol{J}_{\text{skip-gram}}(\boldsymbol{v}_c, w_{t-m}, \ldots w_{t+m}, \boldsymbol{U})}{\partial \boldsymbol{v}_w}$ when $w \neq c$

请写出关于 $\frac{\partial \boldsymbol{J}(\boldsymbol{v}_c, w_{t+j}, \boldsymbol{U})}{\partial \boldsymbol{U}}$ 和 $\frac{\partial \boldsymbol{J}(\boldsymbol{v}_c, w_{t+j}, \boldsymbol{U})}{\partial \boldsymbol{v}_c}$ 的答案。这非常简单，每个答案应该只有一行。

My answer:

$$\frac{\partial J_{skip-gram}(v_c,\ w_{t-m,\ldots,}w_{t+m},U)}{\partial U} = \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \frac{\partial J(v_c, w_{t+j}, U)}{\partial U}$$

$$\frac{\partial J_{skip-gram}(v_c,\ w_{t-m,\ldots,}w_{t+m},U)}{\partial \boldsymbol{v}_c} = \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \frac{\partial J(v_c, w_{t+j}, U)}{\partial \boldsymbol{v}_c}$$

$$\frac{\partial J_{skip-gram}(v_c,\ w_{t-m,\ldots,}w_{t+m},U)}{\partial \boldsymbol{v}_w} = \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \frac{\partial J(v_c, w_{t+j}, U)}{\partial U} \boldsymbol{y}_j \quad \text{(a little puzzle)}$$

一旦你完成了上述所有练习：假定你通过(a)-(c)计算出了 $J(v_c, w_{t+j}, U)$ 关于 U 和 V 所有模型参数的导数，你就已经计算出所有损失函数关于所有参数的导数，你就做好了实现 word2vec 的准备。