单词集合，对 $C_t$ 中所有词向量取平均，就得到了 $w_t$ 的上下文表示：

$$\boldsymbol{v}_{C_t} = \frac{1}{|C_t|} \sum_{w \in C_t} \boldsymbol{v}_w \tag{5-10}$$

**（3）输出层。**输出层根据上下文表示对目标词进行预测（分类），与前馈神经网络语言模型基本一致，唯一的不同在于丢弃了线性变换的偏置项。令 $\boldsymbol{E}' \in \mathbb{R}^{|\mathbb{V}| \times d}$ 为隐含层到输出层的权值矩阵，记 $\boldsymbol{v}'_{w_i}$ 为 $\boldsymbol{E}'$ 中与 $w_i$ 对应的行向量，那么输出 $w_t$ 的概率可由下式计算：

$$P(w_t|C_t) = \frac{\exp(\boldsymbol{v}_{C_t} \cdot \boldsymbol{v}'_{w_t})}{\sum_{w' \in \mathbb{V}} \exp(\boldsymbol{v}_{C_t} \cdot \boldsymbol{v}'_{w'})} \tag{5-11}$$

在 CBOW 模型的参数中，矩阵 $\boldsymbol{E}$ 和 $\boldsymbol{E}'$ 均可作为词向量矩阵，它们分别描述了词表中的词在作为条件上下文或目标词时的不同性质。在实际中，通常只用 $\boldsymbol{E}$ 就能够满足应用需求，但是在某些任务中，对两者进行组合得到的向量可能会取得更好的表现。

### 2. Skip-gram 模型

绝大多数词向量学习模型本质上都是在建立词与其上下文之间的联系。CBOW 模型使用上下文窗口中词的集合作为条件输入预测目标词，即 $P(w_t|C_t)$，其中 $C_t = \{w_{t-k}, \cdots, w_{t-1}, w_{t+1}, \cdots, w_{t+k}\}$。而 Skip-gram 模型在此基础之上作了进一步的简化，使用 $C_t$ 中的每个词作为独立的上下文对目标词进行预测。因此，Skip-gram 模型建立的是词与词之间的共现关系，即 $P(w_t|w_{t+j})$，其中 $j \in \{\pm 1, \cdots, \pm k\}$。原文献[6] 对于 Skip-gram 模型的描述是根据当前词 $w_t$ 预测其上下文中的词 $w_{t+j}$，即 $P(w_{t+j}|w_t)$。这两种形式是等价的，本章采用后一种形式对 Skip-gram 模型进行解释与分析。

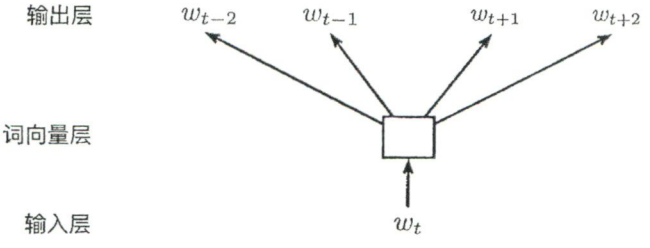


图 5-5 Skip-gram 模型示意图

仍然以 $k = 2$ 为例，Skip-gram 模型可以表示为图 5-5 的结构，其中输入层是当前时刻 $w_t$ 的独热编码，通过矩阵 $\boldsymbol{E}$ 投射至隐含层。此时，隐含层向量即为 $w_t$ 的词向量 $\boldsymbol{v}_{w_t} = \boldsymbol{E}_{w_t}^{\top}$。根据 $\boldsymbol{v}_{w_t}$，输出层利用线性变换矩阵 $\boldsymbol{E}'$ 对上下文窗口内的词进行独立的预测：