2-layer:

\[ h = \sigma \left( \begin{array}{c} W_1 x_1 + W_2 x_2 + b \\ \end{array} \right) \]

\[ y = \text{sigmoid}(h) \]
Language Modeling
Predict the next word

Output P(y|X)

U: \(1 \times d_u\)

hidden layer

\(h_1, \ldots, h_n\)

Embedding
\(d_k \times 2 |v|\)

Input trigram
\(w, o_k, w_1\)

\(W_1, W_2\)

ReLU
\(1 \times d_h\)

Sca \(1 \times 1\)

next word

\(1 \times 2 \cdot |w|\)
training

Loss (Cross-Entropy)

\[ L_{ce}(y, \hat{y}) = -\log \frac{e^{\hat{y}_i}}{\sum_{j=1}^{c} e^{\hat{y}_j}} \]

PyTorch / torch (Python)

Tensorflow
adjustable training properties
Length - Batching
**Figure 7.9**  Computation graph for the function $L(a, b, c) = c(a + 2b)$, with values for input nodes $a = 3$, $b = 1$, $c = -2$, showing the forward pass computation of $L$.

**Figure 7.10**  Computation graph for the function $L(a, b, c) = c(a + 2b)$, showing the backward pass computation of $\frac{\partial L}{\partial a}$, $\frac{\partial L}{\partial b}$, and $\frac{\partial L}{\partial c}$. 
Figure 7.11  Sample computation graph for a simple 2-layer neural net (= 1 hidden layer) with two input dimensions and 2 hidden dimensions.