Probabilistic language models

- Represent text as a sequence of atomic **tokens**, typically words or characters.
- Choose a **context size** $n > 0$.
- Suppose that sequences $x_0, \ldots, x_{n-1}, x_n$ of $n+1$ consecutive tokens — $(n+1)$-**grams** — drawn from documents of a large text corpus are distributed according to

$$P(x_n \mid x_0, \ldots, x_{n-1})$$

- Our modeling task is finding an approximation $\widehat{P}$ to P.

The (n+1)-gram model

- Approximate the joint probability mass

$$\widehat{P}(x_0, x_1, \ldots, x_n)$$

by a relative frequency and setting

$$\widehat{P}(x_n \mid x_0, \ldots, x_{n-1}) = \frac{\widehat{P}(x_0, \ldots, x_n)}{\widehat{P}(x_0, \ldots, x_{n-1})}.$$

- When $n = 1$, this is called the **bigram model**.
- When $n = 2$, this is called the **trigram model**.

Text generation

- Having fit the $(n+1)$-gram model to a text corpus, we can generate new text by sampling from the model:
  - Start with tokens $x_0, \ldots, x_{n-1}$.
  - Draw x_n from $\widehat{P}(x_n \mid x_0, \ldots, x_{n-1})$.
  - Draw x_{n+1} from $\widehat{P}(x_{n+1} \mid x_1, \ldots, x_n)$.
  - $\ldots$

"Digestive" language modeling

- Suppose

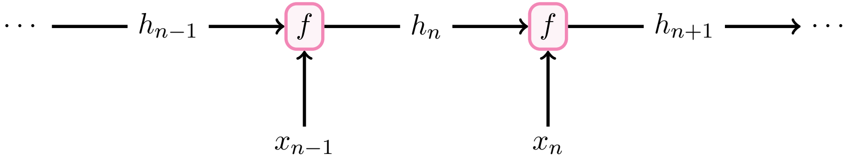
$$\widehat{P}(x_n \mid x_0, \ldots, x_{n-1}) = \widehat{P}(x_n \mid h_n),$$

  where h_n is a "digest" of $x_0, \ldots, x_{n-1}$ that:

  1. h_n depends on $x_0, \ldots, x_{n-2}$ only through h_{n-1}, and
  2. the form of this dependence is independent of n:

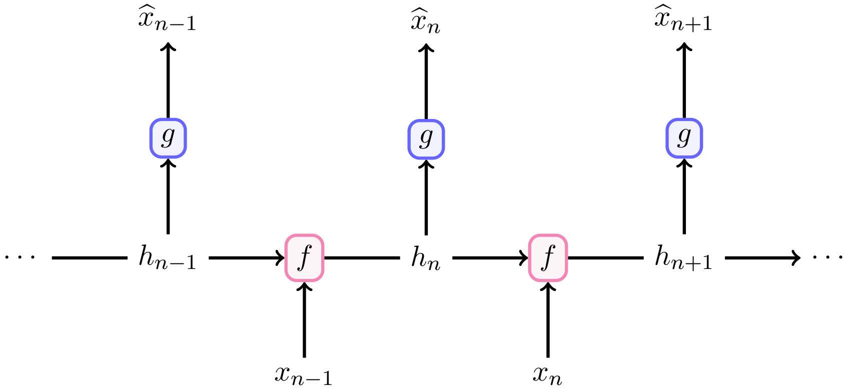
$$h_n = f(x_{n-1}, h_{n-1}).$$

- In practice, f is a neural network.

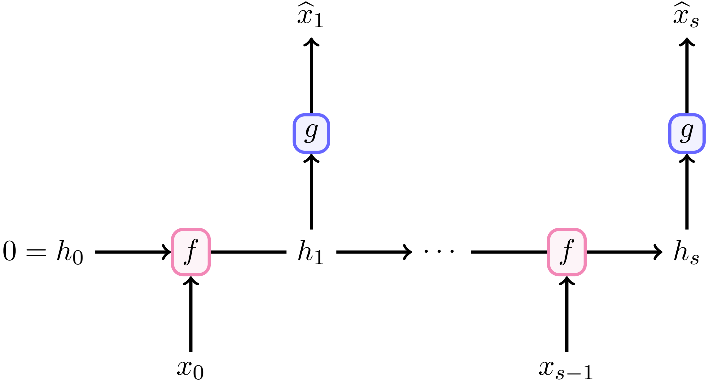


Training f

- We train f to **predict the next token**, using an auxilliary classifier g, trained concurrently.



- We train RNNs on **batches** of **token sequences**.
- Training one sequence with inputs $x_0, \ldots, x_{s-1}$ and targets $x_1, \ldots, x_s$:



Text generation

- We have:

$$h_n = f(x_{n-1}, h_{n-1}),$$
$$\widehat{P}(x_n \mid h_n) = g(h_n)$$

- Having trained f and g, we can generate text, autoregressively, as before.