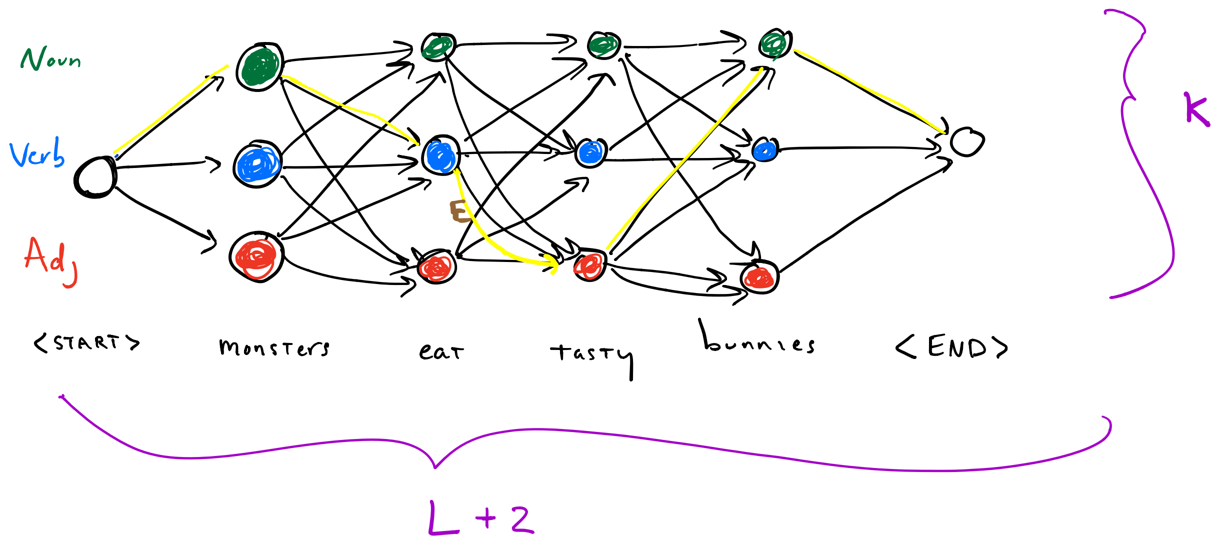


Labels = States

Observation 1 Can only be in one state at time $\tau$

Observation 2 By construction we have assumed that we only care about features relating adjacent words and labels.

Observation 3 Any path through this trellis corresponds to a unique labeling of $x$.

Consider E the edge from Verb → Adj b/w eat and tasty

$$\text{weight}(E) = w \cdot \phi(\text{tasty}, \text{adj.}, \text{verb} \to \text{adj.} \ldots)$$

How does this allow us to compute arg max?

Naively -- if we just considered all unique paths -- it wouldn't!

But this structure permits Dynamic Programming for efficient computation.

## Viterbi Algorithm

Define

$$\alpha_{\ell,k} = \text{best possible output up to and including } \ell \text{ for label } k.$$

$$= \max_{\hat{y}_{1:\ell-1}} w \cdot \phi_\ell(x, \hat{y} \oplus k)$$

$\oplus$ denotes concatenation

Consider $\ell = 2$; Assume we have computed $\alpha$s up to this point

The word at $\ell = 2$ is "eat"

Now we want to derive $\alpha_{3,\text{Adj}}$; How to calculate this score?

→ max over possible previous states!

$$\alpha_{3,\text{Adj}} \leftarrow \max \{ \alpha_{2,\text{Noun}} + w_{\text{Tasty}/\text{Adj}} + w_{\text{Noun}\to\text{Adj}},$$
$$\alpha_{2,\text{Verb}} + w_{\text{Tasty}/\text{Adj}} + w_{\text{Verb}\to\text{Adj}},$$
$$\alpha_{2,\text{Adj}} + w_{\text{Tasty}/\text{Adj}} + w_{\text{Adj}\to\text{Adj}} \}$$

We can generalize this for a recursive defintion

Let $y^*_{l,k}$ denote the label at position l-1 that acheives the max

Init

$$\alpha_{0,k} = 0 \quad \forall k$$

$$y^*_{0,k} = \emptyset \quad \forall k$$

Feature vector for position $\ell$ given that we are coming from k' and going to k.

Best score for k' at $\ell$

Now

$$\alpha_{\ell+1,k} = \max_{k'} \left[ \alpha_{\ell,k'} + w \cdot \phi_{\ell+1}(x \mid k, k') \right]$$

Where to transition from

$$y^*_{\ell+1,k} = \text{Same, but } \underline{\text{argmax}}$$

At the end, can find best seq:

$$\underset{k}{\text{argmax}}\ \alpha_{Lk} \quad \text{(last state)}$$

and trace backwards.

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{matrix}$$

**Unary** features?

- Pixel-value/face terms
- others? Maybe coordinate info

**Markov** features

- Transition to face/not-face from neighbors

$$y_{(i-1)(j-1)},\ y_{(i-1)j},\ y_{i(j-1)}$$

Dynamic program

Start from $(0,0)$ proceed