LING5702: Lecture Notes 24
Models of Grounding

Earlier we saw evidence that people use their language's syntax to learn meanings.

How can we model this?

Contents

24.1 Convolutional models of vision

First we start with a model of vision.

In many animals, the occipital lobe runs sensory signals through progressive filters.

Layers of visual cortex are modeled by **convolving** a $K \times L$ **filter** $\mathbf{W}$ over a **signal** $\mathbf{F}$

$$(\mathbf{F} * \mathbf{W})_{[i,j]} \stackrel{\text{def}}{=} \sum_{k,\ell} \mathbf{F}_{[i-\frac{K}{2}+k,\, j-\frac{L}{2}+\ell]} \cdot \mathbf{W}_{[k,\ell]}$$

So, for example:

$$\overbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}^{\text{signal}} * \overbrace{\begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}}^{\text{filter}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 4 & 2 \\ 0 & 0 & 0 & 2 & 0 \end{bmatrix}$$

A **convolutional neural network** is the same thing, but with a sigmoid $\sigma(x) \stackrel{\text{def}}{=} \frac{1}{1+e^{-x}}$:

$$\Big(\mathrm{CNN}_{\mathbf{W}}(\mathbf{F})\Big)_{[i,j]} \stackrel{\text{def}}{=} \sigma\left(\sum_{k,\ell} \mathbf{F}_{[i-\frac{K}{2}+k,\, j-\frac{L}{2}+\ell]} \cdot \mathbf{W}_{[k,\ell]}\right)$$

These are then chained up to simulate N layers:

$$\mathbf{i} \stackrel{\text{def}}{=} \mathrm{FF}(\mathrm{CNN}_{\mathbf{W}_N}(\mathrm{CNN}_{\mathbf{W}_{N-1}}(\ldots \mathrm{CNN}_{\mathbf{W}_2}(\mathrm{CNN}_{\mathbf{W}_1}(\mathbf{F}))\ldots)))$$

These models backpropagate like regular neural networks.

Low layers learn simple functions (detect edge); high layers learn complex functions (object type).

24.2 Integration with neural grammar inducer (Zhang et al., 2021)

Then we try to meld these images with word sequences allowed by the grammar.

We do this by first calculating an **outside distribution** for each constituent in an N-length sentence:

$$\mathbf{u}_{i,j} \stackrel{\text{def}}{=} \sum_{k=0}^{i} \mathbf{u}_{k,j}{}^{\top} \mathbf{G} \left(\mathbf{v}_{k,i} \otimes \mathbf{I}\right) + \sum_{k=j}^{N} \mathbf{u}_{i,k}{}^{\top} \mathbf{G} \left(\mathbf{I} \otimes \mathbf{v}_{j,k}\right)$$

then calculating **inside likelihood** of each constituent:

$$\mathbf{v}_{i,j} \stackrel{\text{def}}{=} \sum_{k=i+1}^{j-1} \mathbf{G} \left(\mathbf{v}_{i,k} \otimes \mathbf{v}_{k,j}\right)$$

Calculate similarity of each constituent w. image, weighted by constituent **posterior probability**:

$$\mathbf{W}^{(t)} = \mathbf{W}^{(t-1)} - \frac{\partial}{\partial \mathbf{W}^{(t-1)}} \sum_{\sigma \in \mathcal{D}} -\ln \mathsf{P}(\sigma) + \gamma \sum_{i,j} \left(1 - \overbrace{\cos\left(\mathbf{i}, \overbrace{\frac{1}{j-i+1} \sum_{k=i}^{j} \mathbf{w}_k}^{\text{avg. vector}}\right)}^{\text{similarity}}\right) \overbrace{\mathbf{u}_{i,j}{}^{\top} \mathbf{v}_{i,j}}^{\text{posterior of constituent}}$$

where γ is a **regularization weight** and $\mathbf{w}_k$ is a word vector for word k.

Cosine similarity is a normalized inner product: $\cos(\mathbf{i}, \mathbf{w}) = \frac{\mathbf{i}}{\sqrt{\sum_i (\mathbf{i}_{[i]})^2}}^{\top} \frac{\mathbf{w}}{\sqrt{\sum_i (\mathbf{w}_{[i]})^2}}$.

This might allow images to be associated with individual constituents (phrases or clauses)...

References

Zhang, S., Song, L., Jin, L., Xu, K., Yu, D., & Luo, J. (2021). Video-aided unsupervised grammar induction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1513–1524). Online: Association for Computational Linguistics.