TTIC 31230 Fundamentals of Deep Learning

Transformer Problems.

Problem 1. The Transformer computes layers of sequences of vectors $M[\ell, t, j]$ where ℓ ranger over layers, t ranges over "time" (the index into the sequence), and j is the index of a particular dimension of the vector at layer ℓ and time t.

We also let h range over "heads" — each head is computing an attention for a different purpose.

We compute $M[\ell+1, T, J]$ from $M[\ell, T, J]$ as follows:

$$\begin{aligned}
\text{Query}[\ell, h, t, k] &= \sum_j W^Q[\ell, h, k, j] M[\ell, t, j] \\
\text{Key}[\ell, h, t, k] &= \sum_j W^K[\ell, h, k, j] M[\ell, t, j] \\
\text{Value}[\ell, h, t, i] &= \sum_j W^V[\ell, h, i, j] M[\ell, t, j] \\
s[\ell, h, t, t'] &= \frac{1}{\sqrt{K}} \sum_k \text{Query}[\ell, h, t, k] \text{Key}[\ell, h, t', k] \\
\alpha[\ell, h, t, t'] &= \underset{t'}{\text{softmax}}\ s[\ell, h, t, t'] \quad \text{the attention} \\
V[\ell, h, t, i] &= \sum_{t'} \alpha[\ell, h, t, t'] \text{Value}[\ell, h, t', i]
\end{aligned}$$

$$M[\ell+1, t, J] = V[\ell, 1, t, I]; V[\ell, 2, t, I]; \cdots; V[\ell, H, t, I] \quad J = HI$$

(a) assume that summations are done serially on a GPU while other computations are maximally parallel. Under these assumptions what is the order of the parallel running time of the Transformer as a function of L, T, H, J and K (we have $I = J/H$).

Solution: The computation must be serial over the layers but we need to determine the parallel running time for computing $M[\ell+1, T, J]$ from $M[\ell, T, J]$ The first three equations can be computed in parallel over t, h, k and i. We have assumed that each summation takes time proportional to J. The fourth equation can be computed in parallel over h, t and t' and takes time proportional to K. The softmax operation can be parallelized over h and t but requires computing the normalizing constant Z by summing over t' and takes time proportional to T. Computing $V[\ell, h, t, i]$ can be parallelized over h, t and i but sums over t' and hence takes time proportional to T. The concatenation operation can be

fully parallelized and so takes unit time in parallel. Putting this all together we get

$$O(L(J + K + T))$$

(b) Actually a summation over N terms can be done in parallel in $O(\log N)$ time. Repeat part (a) but under the assumption that the summations take logarithmic parallel time.

Solution: $O(L(\log J + \log K + \log T))$

The Transformer is much faster than an RNN. It is comparable in speed to a CNN but with each position in each layer taking input from all positions in the previous layer.

Problem 2. Just as CNNs can be done in two dimensions for vision and in one dimension for language, the Transformer can be done in two dimensions for vision — the so-called spatial transformer.

(a) Rewrite the equations from problem 1 so that the time index t is replaced by spatial dimensions x and y.

(b) Assuming that summations take logarithmic parallel time, give the parallel order of run time for the spatial transformer as a function of L, X, Y, H, J, and K.