6.7900 Fall 2024: Lecture Notes 12

1 Transformers

1.1 Attention Mechanism For Images

This section is to illustrate the attention mechanism, and how it is built up to be the full transformer model. The sentence so far is "A woman is throwing a " with "frisbee" being the desired next word which can be seen in the image. Let $h \in \mathbb{R}^d$ be the context vector given by an RNN. Let an image $M \in \mathbb{R}^{H \times W}$ be convolved with a set of d filters or kernels to form $Z \in \mathbb{R}^{H \times W \times d}$. Here $z_{ij} \in \mathbb{R}^d$ is the result of the inner product between the d filters at position $(i, j) \in \mathbb{R}^{H \times W}$. We then score each relative position $s_{ij} = \text{score}(h, z_{ij})$. Then we compute attention weights defined as the softmax of the scores s_{ij}.

$$a_{ij} = \frac{e^{s_{ij}}}{\sum_{k \in [H], \ell \in [W]} e^{s_{k\ell}}}$$

The weighted aggregation is defined as the following d dimensional vector

$$c = \sum_{ij} a_{ij} z_{ij}$$

Then we combine c with the state vector h to create the word "frisbee".

1.2 Key-Query-Value

The key-query-value attention mechanism is typically explained via an analogy to querying databases. In lecture we first introduce the attention mechanism with no learnable parameters and then with learnable parameters. A query q representing a word such as "food" is used to find a key $k_{\hat{i}}$ where $\hat{i} = \arg\max_i \{\langle q, k_i \rangle\}$ and return the value $v_{\hat{i}}$.

We define the attention mechanism as follows. Let $x_1, x_2, ..., x_n \in \mathbb{R}^d$ be the vector embeddings of a sentence. Let $W_q, W_k, W_v \in \mathbb{R}^{d \times d}$ be query, key, and value

matrices . Let $q = W_q x_n$. Let $v_i = W_v x_i$, let $k_i = W_k x_i$. Let K and V be the set of k_i's and v_i's respectively. Then we define

$$A(q, K, V) = \sum_i \frac{e^{q \cdot k_i}}{\sum_j e^{q \cdot k_j}} v_i$$

As stated, the attention mechanism is permutation invariant. To make this a sensible model for sequence modeling we add positional encodings to the words that can be chosen to be a sinusoidal function. For the full transformer model, we utilize multiple heads per layer which are concatenated and transformed before being fed into an MLP.

1.3 Generalization

We make the following definitions. A training set is the data used by the learning algorithm to fit the model parameters. The validation set is our proxy for the test set, used for selecting modeling choices, including types of features, hyperparameters, etc. Finally the test set is for deployment and final performance assessment, which can't be used iteratively to revise architectures.

Let $\hat{\mathcal{F}}_M$ be the hypothesis class comprised of predictors $\hat{f}_\ell$ for $\ell \in [M]$. We use the validation set to evaluate which of the $\hat{f}_\ell$ we should adopt. The validation set $S = \{(x^i, y^i), i = 1, ..., N\}$ for (x^i, y^i) drawn i.i.d from a distribution P. The test examples are also drawn from the same distribution P.

Let the loss function $L(y, f(x))$ return either a class label or a probability distribution over labels.

$$R_S(f) = \frac{1}{N} \sum_{i=1}^{N} L(y^i, f(x^i))$$

Let the expected test risk be

$$R(f) = E_{(x,y)\sim P}[L(y, f(x))]$$

We typically select $\hat{f} = \arg\min_{f \in \hat{\mathcal{F}}_M} R_S(f)$ and hope that $R_S(\hat{f}) \approx R(\hat{f})$. We would like to ensure the empirical validation risk $R_S(\hat{f})$ for the chosen classifier is close to the corresponding test risk $R(\hat{f})$ with high probability over the randomness in drawing the validation set.

Key Idea: we consider a stronger requirement, that this holds *uniformly* for all $f \in \hat{\mathcal{F}}_M$

$$\mathbb{P}_{S\sim P}(\forall f \in \hat{\mathcal{F}}_M, |R_S(f) - R(f)| \leq \epsilon) \geq 1 - \delta$$

Or equivalently

$$\mathbb{P}_{S\sim P}(\exists f \in \hat{\mathcal{F}}_M, |R_S(f) - R(f)| \geq \epsilon) \leq \delta$$

The key is to understand what is the smallest ϵ for which this statement holds as a function of $\delta, N,$ and M. We denote this quantity $\epsilon(\delta, N, M)$. For intuition, as N increases, ϵ decreases. With more data there is smaller generalization error. As δ increases, ϵ decreases. As the probability that there is an estimator that fails, the smaller the generalization error on the successful estimators. Finally as M increases, ϵ increases. As the number of estimators that are being considered increases, the greater the generalization error.

Now we derive a bound on $\epsilon(\delta, N, M)$. We start by replacing $\exists$ with $\forall$ as is a standard with union bound.

$$\mathbb{P}_{S\sim P}(\exists f \in \hat{\mathcal{F}}_M, |R_S(f) - R(f)| \geq \epsilon) \leq \mathbb{P}_{S\sim P}(\cup_{\forall f\in\hat{\mathcal{F}}} |R_S(f) - R(f)| \geq \epsilon) \tag{1}$$

Applying union bound yields

$$\leq \sum_{f\in\hat{\mathcal{F}}} \mathbb{P}_{S\sim P}(|R_S(f) - R(f)| \geq \epsilon) \tag{2}$$

Then applying hoeffding's inequality for the average of independent bounded random variables (we can assume the loss is bounded in $[0, 1]$ without loss of generality). We obtain

$$\leq \sum_{f\in\hat{\mathcal{F}}} 2\exp(-2N\epsilon^2) = 2M\exp(-2N\epsilon^2) \tag{3}$$

Solving $2M\exp(-2N\epsilon^2) \leq \delta$ for ϵ we find $\epsilon \leq \sqrt{\frac{1}{N}\log(\frac{M}{\delta})}$.