Neural network knowledge distillation in tensor networks

Dereck Piché

March 12, 2023

Abstract

1 Introduction

void

2 Knowledge Distillation

Knowledge Distillation is a machine learning practice which involves taking a trained model and using it's parameters to train another one. The already trained model is refered to as the "teacher", and the model in which his "knowledge" is to be distilled is called the "student".

2.1 Response-Based Knowledge Distillation

The first approach is to look exclusively at the outputs of the student and teachers. Now, we apply the logits of our functions element-wise to the outputs and the softmax.

$$\mathtt{softmax}(v_i) = \frac{e^{v_i}}{\sum_j e^{v_i}} \tag{1}$$

The softmax function's objective is to transform the logits into probability distributions for the different classes. We will now apply a loss function L to these two functions. Since we are trying to reduce the divergence between two distributions, we will use the Kullback-Leibler divergence loss

$$KL(P, Q) = \frac{1}{n} \sum_i^n Q \frac{\log_e(Q)}{\log_e(P)} \tag{2}$$

Here, Q is the distribution we are aiming for and P is the one we have.

3 Tensor Networks

Tensor Networks come from the study of quantum phenomena. They started being used recently as machine learning models.

Consider a input vector x.

3.1 Tensor Products and Transformations

Consider an input vector $x \in R^d$. Let $\phi(x) : R \mapsto R^{d_\phi}$. Then, let us take the tensor product of apply $\phi(x)$ applied to every element of the vector x.

$$\Phi(x) = \phi(x_1) \otimes \phi(x_2) \otimes (\dots)\phi(x_d) \tag{3}$$

We obtain a tensor $\Phi(x)$, of which the sum of it's element contain the basis of a space of products of the elements of the transformed elements of $\phi(x)$.

In order to reduce the abstractness of this statement, we can take say that $\otimes$ refers to the Kronecker Product.

A particular dimension is of interest here, the transformation

$$\phi(x) = \begin{bmatrix} 1 \\ x \end{bmatrix} \tag{4}$$

With, this particular transformation, the Kronecker Product gives us a $\Phi(x)$ which is a matrix where each element is a basis of the space of multilinear functions on the elements of the vector x.

3.2 Linear Combinations

How do we use $\Phi(x)$, how does it become useful? Well, it becomes useful when we take a linear map of it's elements. Let our $\Phi(x)$ be an arbitrely shaped tensor of d_Φ dimensions. The final form of our function $f(x)$ will be

$$f(x) = \sum_{i}^{d_\Phi} \theta_i [\Phi(x)]_i \tag{5}$$

Now, we have a function that can, under some choice of transformation ϕ, become extremely expressive.

3.3 The *Matrix Product State* Tensor Network

3.4 Expressivity of MPS combinations with transformation $[1, x]^t$

Let $x \in R^d$. Let $f(x)$ be a function that returns a vector v, which contains every element required to form a basis of the space of x_i-variate multinear functions.

Let

$$\Theta = \{\theta_1, \theta_2, (\dots), \theta_{d_{theta}}\} \tag{6}$$

in

$$M(f(x), \Theta) = \begin{bmatrix} m(f(x), \theta_1) \\ m(f(x), \theta_2) \\ (\dots) \\ m(f(x), \theta_{d_{theta}}) \end{bmatrix} \tag{7}$$

Where $m(f(x), \theta_i) : \ R^{2^d} \mapsto (R^d \mapsto R)$ is of the form

$$m(f(x), \theta_i) = m(v, \theta_i) = \sum_j \theta_{i,j} \cdot v_j \tag{8}$$

We can trivially choose $\Theta^* \in \Theta$ such that

$$M(f(x), \Theta^*) = \begin{bmatrix} \left.\begin{matrix} x_1 \\ (\dots) \\ x_1 \end{matrix}\right\} \textit{repeated } z \textit{ times} \\ \left.\begin{matrix} x_2 \\ (\dots) \\ x_2 \end{matrix}\right\} \textit{repeated } z \textit{ times} \\ (\dots) \\ \left.\begin{matrix} x_n \\ (\dots) \\ x_n \end{matrix}\right\} \textit{repeated } z \textit{ times} \end{bmatrix} = \lambda \tag{9}$$

We can rewrite λ as

$$\lambda = \begin{bmatrix} \varepsilon_{1,1} \\ (\dots) \\ \varepsilon_{1,z} \\ \varepsilon_{2,1} \\ (\dots) \\ \varepsilon_{2,z} \\ (\dots) \\ \varepsilon_{d,z} \end{bmatrix} = \lambda^* \tag{10}$$

Let Z be the space of x_i-variate polynomial functions of degree $\leq z$. Then every monomial of any x_i-variate polynomial function $\zeta \in Z$ is of the form.

$$x_1^{k_1} x_2^{k_2} (\dots) x_d^{k_d} \tag{11}$$

under the condition $\sum_i k_i \leq z$. However, for every $K = \{k_1, k_2, (\dots), k_d\}$ meeting this condition, we can rewrite the monomial as

$$\left(\prod_{i_1=1}^{k_1} \varepsilon_{1,i_1} \right) \left(\prod_{i_2=1}^{k_2} \varepsilon_{2,i_2} \right) (\dots) \left(\prod_{i_d=1}^{k_d} \varepsilon_{d,i_d} \right) \tag{12}$$

by using the elements of λ^* from (10). However, we clearly see that this term is a multilinear monomial of the variables in λ^*. This implies that $f(\lambda*)$ returns a basis-vector of x_i-variate polynomial function of degree $\leq z$.

In other words,

$$\xi(x, \Theta) = M(f(M(f(x), \Theta^*)), \Theta) \tag{13}$$

can express any x_i-variate polynomial function of degree $\leq z$ under fixed Θ.