Phase Equilibria Transformer: Part 1 - Attentive Partitioning

Ali Qajar
ali.qajar @ gmail.com

Abstract

Attentive partitioning model is introduced as a substitute for the Transformer attention block. It is an attention based model and like the Transformer utilizes positional encoding to learn input configurations. However, it departs from the traditional Query-Key-Value (QKV) schema, and instead, leverages a partitioning approach. Attentive partitioning is a linear model with respect to the context window and accounts for the binary and higher order interactions. Additionally, it produces multiple outputs (phases) per partition head. The key takeaway from this paper is that if the model architecture is designed around partitioning input data, the attention can be resolved in linear time. This paper centers on self-attention in discriminative encoders.

Introduction

Transformer attention block considers all binary interactions between the input tokens. This makes the attention block of quadratic time/space complexity (Vaswani et al., 2017). The attention in Transformer is a special case of what in molecular dynamics is called intermolecular interactions. For instance, transition state theory tells us that in condensed fluids with highly reactive intermediates, 3-body interactions can't be ignored.

Several research efforts have focused on accelerating the attention mechanism in the Transformer. Some notable examples include: The Linformer introduces low-rank matrix approximations (Wang et al., 2020). Performer applies kernel-based approximations to reduce complexity of attention calculations (Choromanski et al. 2021). Reformer model utilizes locality-sensitive hashing to reduce the complexity of the attention block (Kitaev et al., 2020). Fastformer speeds up the attention by adopting an additive attention approximation (Wu et al., 2021). Sparse Transformer employs sparse attention patterns to focus computations on a subset of key tokens (Child et al., 2019). The Longformer model introduces an attention mechanism that combines local windowed attention with task-specific global attention mechanisms (Beltagy et al., 2020).

In this paper, I introduce the attentive partitioning algorithm that is linear to the context window for structured inputs and accounts for interactions of any order. The following summarizes the steps taken

- Instead of attention (binary interactions), we focus on partitioning the input.
- Define partition factors as a function of interactions, we call this attentive partitioning.
- Train a feed-forward model to learn the partition factors.
- Generate multiphase outputs using the learned partition factors

This paper covers the self-attention in discriminative encoders which are suitable for feature extraction and classification. Part 2 of this work, in a separate paper, will introduce the 'Generative Partitioning' algorithm suitable for generative encoders and decoder stacks.

Partitioning

Imagine we train a Transformer encoder for image classification. Figure 1, shows one of the images used in training, a fish against a blurred background. Normally, we train the model to pay attention to the fish. In doing so, the Transformer attention block produces filters to extract the features that best describe the fish. We can interpret these filters in different ways. One interpretation is 'semantic segmentation', where the model outputs a probability distribution over the possible classes for each pixel. For instance, one might say a particular pixel, that belongs to the fish, has a probability distribution of 0.98 fish and 0.02 background. Another interpretation is called 'partitioning'. In that, we 'Partition' a pixel into the two groups, fish and background and report it with a partition factor of 0.98/0.02 = 49. The difference between the two interpretations is nuanced. Semantic segmentation is reported as a probability distribution over classes, whereas partitioning is reported in partition-factors against an arbitrary reference group. As it turns out, the 'partitioning' interpretation will help to linearize attention computations.

Figure 1: We assume we partition the image into two phases at equilibrium. Phase 1: Fish, Phase 2: Background.

Attentive Partitioning Algorithm

Assume we have an input with N tokens. For this analysis I refer to a token as a component for consistency. Without losing generality, let's assume the input components are a vector of real numbers. The following analysis is valid with or without considering the embedding dimension, d, as it is dimensionally invariant. For simplicity the analysis is provided for real values and not embedding vectors.

We would like to partition the components into P phases. Each phase is composed of a distribution of the components; we call these distributions, composition. The composition of the input itself is denoted by z_i, for $i \in \{1, 2, ..., N\}$ and the composition of the phases are denoted by x_{ij}, for $i \in \{1, 2, ..., N\}$ and $j \in \{1, 2, ..., P\}$. Figure 2 shows an example of such partitioning, where the initial composition z_i is split into two phases, 1 and 2, with the compositions x_{i1} and x_{i2}. Phase fractions are denoted by L_1 and L_2 where $L_1 + L_2 = 1$. Let's assume phase 2 is the reference phase and define the partition factors as $K_i \equiv x_{i1}/x_{i2}$. The partition factors can tell us how initial composition, z_i will partition into phase compositions x_{i1} and x_{i2} during the phase split.

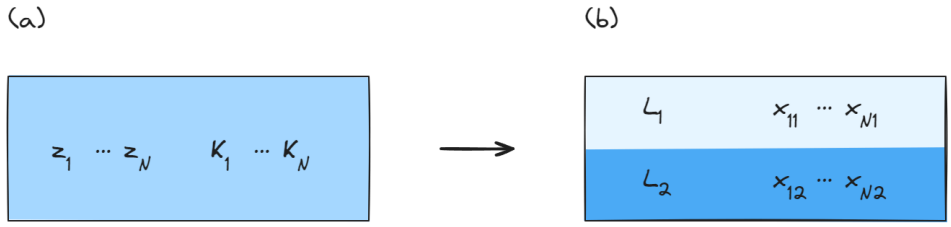

Figure 2: (a) State 1: Input with overall composition of z_i and partition factors K_i (b) State 2: Input after phase split, with phase fractions, L_1 and L_2, and phase compositions x_{i1} and x_{i2}

Based on the above observation we can write the material balance equations for the general case of a phase split to P phases. Assume we have an input with N components. The goal is to partition those components into P phases. The material balance equations can be written as follows

$$\sum_{j=1}^{P} L_j = 1 \tag{1a}$$

$$z_i = \sum_{j=1}^{P} x_{ij} L_j \quad \text{for } i \in \{1, 2, ..., N\} \tag{1b}$$

$$K_{ij} \equiv x_{ij}/x_{ip} \quad \text{for } j \in \{1, 2, ..., P-1\} \tag{1c}$$

$$\sum_{i=1}^{N} z_i = \Lambda \tag{1d}$$

$$\sum_{i=1}^{N} x_{ij} = \Lambda \quad \text{for } j \in \{1, 2, ..., P-1\} \tag{1e}$$

Equation (1a) is a definition, that phase fractions sum up to 1. Equation (1b) is material balance for component i, saying overall composition of component i, z_i, is equal to the sum of the compositions of component i in all phases.

Equation (1c) is the definition of the partition factor for P phases. Assume the partition factor is a real number. Without losing generality we assume that phase P is the reference phase and define the partition factors K_{ij} relative to the composition of the reference phase, x_{ip}. Note that, with this definition we introduce a singularity to the model. Composition of a component can't be zero. Therefore, we assume overall or phase composition of each component, i, is non-zero. To avoid singularity during computations, we can use a smoothing parameter like $\alpha = 10^{-4}$, where : $z_i \rightarrow |z_i| + \alpha$ for $|z_i| < \alpha$.

Equations (1d) and (1e) constrain the overall composition and each phase composition to sum up to a common arbitrary value, Λ, ensuring consistency across the system.

By combining equations (1a), (1b), (1c) one can obtain composition of the reference phase as

$$x_{ip} = z_i/u_i \qquad \text{for } i \in \{1, 2, \ldots, N\} \tag{2a}$$

$$u_i = 1 + \sum_{l=1}^{P-1} L_l(K_{il} - 1) \tag{2b}$$

Using equations (2) and (1e), we can reduce the material balance equations to the following form

$$F_j(L_1, \ldots, L_{P-1}) = \sum_{i=1}^{N} z_i(K_{ij} - 1)/u_i = 0 \tag{3}$$

In solution thermodynamics, the system of equations (3), is known as the Rachford-Rice equation (Whitson and Michelsen, 1989; Michelsen 1994). In the Rachford-Rice equation, the compositions are constrained to the interval of (0, 1). However, during derivation of equation (3) we used assumptions in equation (1) that extend the domain of the variables by allowing z_i and x_{ij} to be any non-zero real number. Additionally, we assumed that composition sums can be any arbitrary real number, Λ, and as long as that number is the same for all compositions the equation (3) still holds.

Based on the above analysis, the domains of the parameters and variables of equation (3) are as follows

$$z_i \in R - \{0\},\ x_{ij} \in R - \{0\},\ K_{ij} \in R - \{0\},\ L_j \in [0, 1] \tag{4}$$

In thermodynamic systems, the initial composition, z_i, is given and the partition factors, K_{ij}, are obtained from analytical or empirical models. Therefore, equation (3) is determined with respect to the variables L_j for $j \in \{1, 2, \ldots, P - 1\}$.

Moreover, in thermodynamic systems, partition factors, K_{ij}, are functions of interactions between the components. They rely on binary or higher-order interactions among the system's components. This relationship can be utilized to incorporate interactions (attention) into the model. By exploiting the fact that the partition factors' dimensionality is linearly related to the context window size, we can leverage this property to decrease the computational complexity from quadratic to linear.

For each phase, j, we can write the partition factors as a function of phase composition of that phase and the reference phase.

$$K_{ij} = \Gamma_j(x_{ij}, x_{ip}) \qquad \text{for } i \in \{1, 2, \ldots, N\} \qquad \text{for each } j \in \{1, 2, \ldots, P-1\} \tag{5}$$

For a machine learning task, the values of K need to be learned, therefore equation (3) is not determined anymore and we can't solve it directly. Instead we need to solve it iteratively through minimization of a loss function.

Let's define the objective function. The input to the model is z and the output is x. Partition factors, K, and phase fractions, L are the parameters that must be learned. For model fitting, we can write the objective function in the form of root mean square (RMS) as follows:

$$minimize\ \mathcal{L} \equiv f(z, K, \theta) + \mu h(\theta) \tag{6a}$$

$$s.t.$$

$$0 \le L_j \le 1 \qquad \text{for } j \in \{1, 2, \ldots, P-1\} \tag{6b}$$

$$h(L) \equiv \left(\sum_{j=1}^{P-1} L_j\right) - 1 \le 0 \tag{6c}$$

where

$$f(K, L) = RMS(F_j) \equiv \sqrt{\frac{1}{P-1}\sum_{j=1}^{P-1} F_j^{\,2}} \tag{6d}$$

$$L_j = \sigma(\theta_j) \qquad \text{for } j \in \{1, 2, \ldots, P\} \tag{6e}$$

Equation (6) is a constrained optimization problem. $\mathcal{L}$ is the Lagrangian objective, f is the root mean square of the phase functions defined in equation (6d). μ is the Lagrange multiplier for the inequality constraint $h(L)$ defined in constraint (5c).

Constraint (6b): To guarantee that the phase fractions *L* remain within their bounds we use the change of variables defined by equation (6e). σ(.) is the sigmoid function. θ is the new unbounded real valued number that substitutes *L* in the objective function.

Constraint (6c): The sum of the phase fractions for $P - 1$ phases should remain less than 1 as expressed in the constraint (6c). The inequality constraints should satisfy Karush-Kuhn-Tucker (KKT) conditions (Bazaraa et al., 2013).

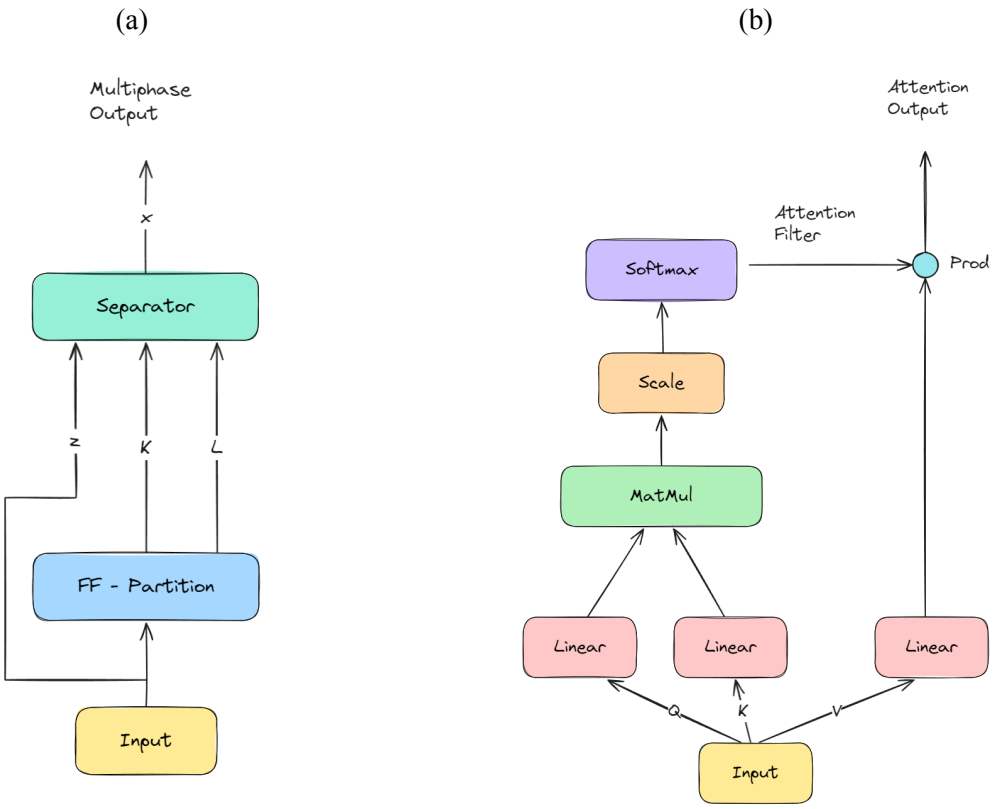

Figure 4: (a) Attentive partition block. FF-Partition produces the partition factors and phase fractions. Separator block evaluates the loss-function during training and evaluates the equation (2) during inference. (b) Transformer self-attention block.

Model Architecture

Attentive partitioning is a multi-task learning algorithm that simultaneously learns to generate both partition factors K and phase fractions L using a single network. By doing so, it enables the network to discover shared representations that are advantageous for each output. This approach compels the model to learn the system's nonlinearities in a coherent manner, thereby reducing the risk of overfitting.

Figure 4 presents a side-by-side comparison of the structures of (a) the attentive partition block and (b) the Transformer self-attention block. The FF-Partition block is a feed-forward network that takes the overall composition z, i.e., the input, and generates approximations for the partition factors K and phase fractions L. The Separator block, on the other hand, receives the overall composition z along with the estimated K and L parameters, and subsequently outputs the phase compositions x.

During training the Separator block evaluates the loss function (6), and back-propagates the error through the FF-Partition network. During inference, Separator block uses the input composition, z, and the

approximated values of K and L from the FF-Partition network and evaluates equations (2) and (1c) to produce phase compositions, x.

The attentive partition and the Transformer self-attention block differ fundamentally in their approach to capturing interactions. While the Transformer self-attention block relies on Query and Key vectors to capture binary interactions, the attentive partition employs the FF-Partition block to capture both binary and higher-order interactions. As outlined in equation (5), the partition factors inherently encode the attention information, and this implicit relationship is learned by the FF-Partition network.

The attentive partition model generates multiple phases that are in thermodynamic equilibrium with one another, indicating that the system has reached its minimum Gibbs free energy state. Thermodynamic equilibrium is characterized by the intensive properties of the system, which are independent of its size or quantity. To illustrate, consider a closed container where a small droplet of a liquid phase can coexist in thermodynamic equilibrium with a substantial amount of a vapor phase. The phase split calculations, as described in equation (3), differentiate between the intensive properties (features) and extensive properties (magnitude) of the system. The phase compositions x encode the features, while the phase fractions L represent the magnitudes.

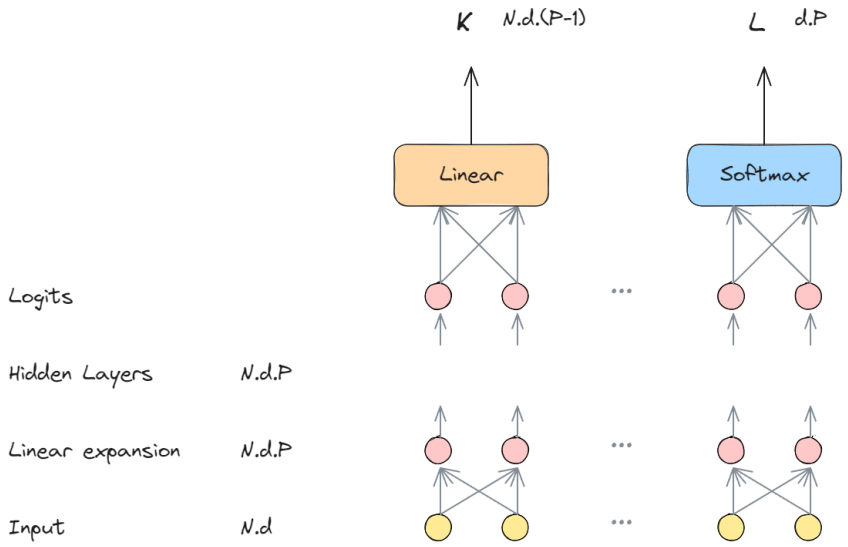

Figure 5: Structure of FF-Partition block

Figure 5 shows the structure and dimensions of FF-Partition network. The input layer is of size $N.d$, where N is the size of the context window. d is the size of the embedding vector. P is the number of phases. For comparison, context window, N, might be in the millions or billions, embedding dimension, d is 512, and the number of phases, P, is a tuning parameter possibly in the range of 2 to 8. It is straightforward to show that the maximum theoretical value for P is $P_{max} = N$ (Smith et al. 2018).

The attentive partition algorithm generates a minimum of two phases, also known as partition heads. In the case of two phases, one interpretation is that the attentive partition block separates the input into interesting and uninteresting components, outputting both. Subsequently, the linear transformation that takes the attention head outputs as input can determine the degree to which it emphasizes or suppresses each of these features.

Next we have the linear expansion layer. We expand the input layer by a factor of P, which makes the linear layer of size $N.d.P$. For high values of P this is a notable expansion. As it is given by equation (5), the partition factors are functions of phase compositions. The linear expansion provides the model with sufficient flexibility to learn the decomposition of the overall composition, z, into the corresponding phase compositions, x. A single linear layer is not enough to learn such highly nonlinear decomposition. Therefore additional hidden layers come after this.

The intermediate hidden layers follow the linear expansion. These layers maintain the same dimensions of $N.d.P$, although their size can be adjusted through experimentation to optimally capture the interactions and relationships among the components (tokens).

Last hidden layer, logits, is fully connected. Dimensions of the logits layer determined by the sum of the dimensions of the output layers (tasks). Attentive partition has two outputs, partition factors, K and phase fractions, L. Phase fractions need to be normalized to 1, therefore a softmax is applied.

Conclusions

In this paper, the attentive partition algorithm was introduced. The model substitutes the Transformer self-attention block. Attentive Partition algorithm employs principles from solution thermodynamics to model data interactions in linear time and space complexity. By converting input data into the composition space and applying phase partitioning, the attentive Partition model efficiently computes binary and higher-order attention. This approach not only reduces computational complexity but also makes the attention mechanism's buffer size tunable to handle input data of various ranks. Finally, by distinguishing between phase composition (features) and phase fraction (magnitude), attentive partition focuses on pertinent information regardless of size.

In the second paper of this series, I will explore the Generative Partition algorithm, which extends this framework to encompass both discriminative and generative tasks while maintaining linear time computation.

Overall, the Partitioning architecture offers a flexible, computationally efficient method for modeling data interactions, suitable for very large context windows. It requires minimal tuning, with all parameters having a clear physical interpretation, facilitating intuitive and predictable design.

References

Bazaraa, M. S., Sherali, H. D., & Shetty, C. M. (2013). Nonlinear Programming: Theory and Algorithms (3rd ed.). John Wiley & Sons.

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The Long-Document Transformer. arXiv preprint arXiv:2004.05150.

Child, R., Gray, S., Radford, A., & Sutskever, I. (2019). Generating Long Sequences with Sparse Transformers. arXiv preprint arXiv:1904.10509.

Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., ... & Weller, A. (2021). Rethinking Attention with Performers. Proceedings of the International Conference on Learning Representations (ICLR).

Kitaev, N., Kaiser, Ł., & Levskaya, A. (2020). Reformer: The Efficient Transformer. Proceedings of the International Conference on Learning Representations (ICLR).

Michelsen, M. L. (1994). Calculation of Multiphase Equilibrium. Computers & Chemical Engineering, 18(7), 545-550.

Smith, J. M., Van Ness, H. C., Abbott, M. M., & Swihart, M. T. (2018). Introduction to Chemical Engineering Thermodynamics (8th ed.). McGraw Hill. ISBN 978-1259696527.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems, 30.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., & Ma, H. (2020). Linformer: Self-Attention with Linear Complexity. arXiv preprint arXiv:2006.04768.

Whitson, C.H., & Michelsen, M.L. (1989). The negative Flash. Fluid Phase Equilibria, 53, 51-71.

Wu, C., Wu, F., Qi, T., Huang, Y., & Xie, X. (2021). Fastformer: Additive Attention Can Be All You Need. arXiv preprint arXiv:2108.09084.