DOUBLY-NORMALIZED ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Models based on the Transformer architecture have achieved better accuracy than models based on competing architectures. A unique feature of the Transformer is its universal application of a self-attention mechanism, which allows for free information flow at arbitrary distances. In this paper, we provide two alternative views of the attention mechanism: one from the probabilistic view via the Gaussian mixture model, the other from the optimization view via optimal transport. Following these insights, we propose a new attention scheme that requires normalization on both the upper and lower layers, called the doubly-normalized attention scheme. We analyze the properties of both the original and the new attention schemes, and find that the doubly-normalized attention mechanism directly mitigates two unwanted effects: it resolves the explaining-away effect and alleviates mode collapse. We conduct empirical studies that quantify numerical advantages for the doubly-normalized attention model, as well as for a hybrid model that dynamically combines both attention schemes to achieve improved performance on several well-known benchmarks.

1 INTRODUCTION

The Transformer architecture (Vaswani et al., 2017) has been successfully used to improve state-of-the-art performance in a variety of machine learning tasks, such as machine translation (Vaswani et al., 2017; Dehghani et al., 2019), language modeling (Devlin et al., 2019; Yang et al., 2019), summarization (Cohan et al., 2018; Goodman et al., 2019), dialog (Mazaré et al., 2018; Cheng et al., 2019), image captioninig (Sharma et al., 2018; Zhao et al., 2019), and visual question answering (Yu et al., 2019b; Tan & Bansal, 2019). One of the most important components of the Transformer architecture is its self-attention mechanism, applied universally to both the encoder and the decoder components. This attention mechanism allows for information to freely flow between inputs at arbitrary distances, which is intuitively appealing for modeling natural language or tasks that need to model cross-modal relationships between their inputs (such as visual question answering).

Despite the empirical success of the self-attention mechanism, little formal work has been done to analyze its statistical properties and relate it to previously known classical models. Better understanding its properties can lead to insights into what it does and does not do well. This in turn can lead to improvements to the attention mechanism and ultimately to a better-performing Transformer network. In this paper, we closely study the current attention formulation from both a probabilistic view via the Gaussian mixture model and from an optimization view via optimal transport.

First, we reveal the mathematical connection between the Transformer attention and the Gaussian mixture model. In particular, we show that the output neurons (from the upper layer) of an attention unit can be regarded as the most likely data generated by a Gaussian mixture model (GMM), while the input neurons (from the lower layer) of the attention unit act as the Gaussian centers.

A formulation in which the upper layer acts as the generated data while the lower layer acts as the Gaussian centers aligns well with the purpose of a decoder mechanism, where a final upper layer ultimately generates the outputs. However, this design does not correspond well to the purpose of an encoder mechanism, in which the bottom layer takes input data and the upper layer encodes the data. To address this mismatch, we describe in this paper a new attention mechanism, in which the role of the upper and lower layers in the GMM formulation are reversed: the lower layer represents the generated data, while the upper layer represents the Gaussian centers. It is worth noting that a similar design was also applied in the Capsule Networks algorithm (Hinton et al., 2018).

The Maximum Likelihood Estimate (MLE) solution of the reversed GMM model leads to a very similar attention update as the original, except for the attention weight normalization. The original attention scheme only normalizes the attention weights once, over the lower layer of every upper-layer neuron. By contrast, the doubly-normalized attention mechanism requires a two-step attention weight normalization: the first normalizes over the upper layer for each lower-layer neuron, and the second normalizes over the lower layer for each upper-layer neuron. In the rest of this paper, we denote the original, lower normalized attention scheme as LNAS, and the doubly-normalized attention scheme as DNAS.

In addition, our analysis shows that the two normalization schemes can also be connected as optimizing two closely related constrained objective functions, with one additional normalization constraint for DNAS. We also show that DNAS updates correspond to one iteration of the Sinkhorn algorithm in optimal transport (Peyré & Cuturi, 2019).

We mathematically analyze the statistical properties of these two attention schemes, finding that the DNAS proposal possesses two advantages over LNAS. First, it avoids the "explaining away" effect that the LNAS scheme suffers from, in which information present in the input is filtered out too early and cannot be recovered at a later stage when needed. Second, it alleviates the mode-collapsing phenomenon that also plagues the LNAS formulation, which fails to accommodate multiple modes and concentrates its entire mass in a single mode.

We also formulate a hybrid scheme, HNAS, that dynamically combines both attention schemes, and can provide a handle on the preference betewen LNAS and DNAS, as resulting from the optimization algorithm. We perform empirical studies that quantify the numerical advantages of the doubly-normalized attention model, as well as the ones for the hybrid model. We obtain clear numerical improvements using the HNAS formulation over several well-known benchmarks, including a new state-of-the-art result on the Gigaword bechmark for text summarization.

2 ATTENTION FROM THE PROBABILISTIC PERSPECTIVE

In this section, we review the Transformer self-attention mechanism and analyze how it relates to the Gaussian Mixture Model formulation from the probabilistic perspective.

The Transformer attention mechanism involves two layers of neurons. In what follows, we denote the lower-layer neurons as $\mathbf{x}_j$, and the upper-layer neurons as $\mathbf{y}_i$. The self-attention mechanism first transforms the input features to a query and a key by applying the transformations $\mathbf{q}_i = \mathbf{Q}\,\mathbf{x}_i$ and $\mathbf{k}_j = \mathbf{K}\,\mathbf{x}_j$, where $\mathbf{Q}$ and $\mathbf{K}$ are trainable transformation matrices. The value of an upper-layer neuron $\mathbf{y}_i$ is then computed as the weighted sum over the lower-layer neurons $\mathbf{x}_j$, using the attention update equation,

$$\mathbf{y}_i = \sum_j \frac{\exp(\mathbf{q}_i^\top \mathbf{k}_j)}{\sum_j \exp(\mathbf{q}_i^\top \mathbf{k}_j)} \mathbf{x}_j, \tag{1}$$

which is then followed by an additional output-value transformation.

The Transformer attention updates from Eq. (1) can be regarded as the solution of the most likely data generation of a Gaussian mixture model (GMM). To make this connection clear, let us consider the log-likelihood function of a GMM. We denote the Gaussian cluster centers as $\mathbf{k}_j$, the priors of the clusters as α_j satisfying $\sum_j \alpha_j = 1$, and the generated data as $\mathbf{q}_i$. If we assume the variance of the Gaussian distributions to be equal to 1, then the log-likelihood of the GMM is:

$$\sum_i \log p(\mathbf{q}_i) = \sum_i \log \left(\sum_j \alpha_j \, \mathcal{N}(\mathbf{q}_i | \mathbf{k}_j, 1) \right). \tag{2}$$

To find the most likely data generated by the Gaussian mixture model, we take the derivative of $\mathbf{q}_i$,

$$\frac{\partial}{\partial \mathbf{q}_i} \sum_i \log p(\mathbf{q}_i) = \frac{\partial}{\partial \mathbf{q}_i} \log p(\mathbf{q}_i) = \frac{\sum_j \alpha_j \, \mathcal{N}(\mathbf{q}_i | \mathbf{k}_j, 1) \frac{\partial \log \mathcal{N}(\mathbf{q}_i | \mathbf{k}_j, 1)}{\partial \mathbf{q}_i}}{\sum_j \alpha_j \, \mathcal{N}(\mathbf{q}_i | \mathbf{k}_j, 1)}.$$

We introduce the abbreviation

$$\pi_{ij} \triangleq \frac{\alpha_j \, \mathcal{N}(\mathbf{q}_i | \mathbf{k}_j, 1)}{\sum_j \alpha_j \, \mathcal{N}(\mathbf{q}_i | \mathbf{k}_j, 1)} = \frac{\alpha_j \exp(\mathbf{q}_i^\top \mathbf{k}_j - \frac{1}{2} \mathbf{k}_j^\top \mathbf{k}_j)}{\sum_j \alpha_j \exp(\mathbf{q}_i^\top \mathbf{k}_j - \frac{1}{2} \mathbf{k}_j^\top \mathbf{k}_j)}, \tag{3}$$

where $\sum_j \pi_{ij} = 1$. We approximate the cluster priors as $\alpha_j \propto \exp(\frac{1}{2} \mathbf{k}_j^\top \mathbf{k}_j)$, and therefore

$$\pi_{ij} = \frac{\exp(\mathbf{q}_i^\top \mathbf{k}_j)}{\sum_j \exp(\mathbf{q}_i^\top \mathbf{k}_j)}.$$

Also, since $\frac{\partial \log \mathcal{N}(\mathbf{q}_i | \mathbf{k}_j, 1)}{\partial \, \mathbf{q}_i} = \mathbf{k}_j - \mathbf{q}_i$, then,

$$\frac{\partial}{\partial \, \mathbf{q}_i} \sum_i \log p(\mathbf{q}_i) = \sum_j \pi_{ij} (\mathbf{k}_j - \mathbf{q}_i) = \sum_j \pi_{ij} \, \mathbf{k}_j - \mathbf{q}_i \, .$$

Let $\frac{\partial}{\partial \, \mathbf{q}_i} \log p(\mathbf{q}_i) = 0$, then we have:

$$\mathbf{q}_i^{new} = \sum_j \pi_{ij} \, \mathbf{k}_j = \sum_j \frac{\exp(\mathbf{q}_i^\top \mathbf{k}_j)}{\sum_j \exp(\mathbf{q}_i^\top \mathbf{k}_j)} \, \mathbf{k}_j \, . \tag{4}$$

If we compare Eq. (4) with Eq. (1), the two equations are equivalent modulo a transformation matrix $\mathbf{K}$, which can be incorporated in the additional output-value transformation that follows.

3 Doubly-normalized Attention

As we have shown, in the original Transformer self-attention mechanism LNAS, the lower layer corresponds to the Gaussian centers, while the upper layer corresponds to the data generated from these centers. This flow of information aligns well with the goal of a decoding mechanism, in which the upper layer generates the decoded outputs. However, for an encoding mechanism, the information flows in the other direction: data comes from the lower layer, while the role of the upper layer is to represent the signal from the lower-layer. As such, the current design of the Transformer self-attention mechanism, in which the LNAS scheme is applied universally for both the encoder and the decoder networks, does not align well with the information flow of the encoding process.

To resolve this discrepancy, we propose a new attention mechanism, DNAS, resulting from reversing the role of the upper and lower layers in the GMM. Under this new scheme, the lower layer neurons $\mathbf{k}_j$ are treated as data, and the upper layer neurons $\mathbf{q}_i$ are the Gaussian centers. The log-likelihood function becomes:

$$\sum_j \log p(\mathbf{k}_j) = \sum_j \log \left(\sum_i \beta_i \, \mathcal{N}(\mathbf{k}_j | \mathbf{q}_i, 1) \right), \tag{5}$$

with priors β_i satisfying $\sum_i \beta_i = 1$. We take the gradient with respect to $\mathbf{q}_i$,

$$\frac{\partial}{\partial \, \mathbf{q}_i} \sum_j \log p(\mathbf{k}_j) = \sum_j \frac{\beta_i \frac{\partial}{\partial \, \mathbf{q}_i} \mathcal{N}(\mathbf{k}_j | \mathbf{q}_i, 1)}{\sum_i \beta_i \, \mathcal{N}(\mathbf{k}_j | \mathbf{q}_i, 1)} = \sum_j \frac{\beta_i \, \mathcal{N}(\mathbf{k}_j | \mathbf{q}_i, 1) \frac{\partial}{\partial \, \mathbf{q}_i} \log \mathcal{N}(\mathbf{k}_j | \mathbf{q}_i, 1)}{\sum_i \beta_i \, \mathcal{N}(\mathbf{k}_j | \mathbf{q}_i, 1)}.$$

We introduce the abbreviation

$$\xi_{ji} \triangleq \frac{\beta_i \, \mathcal{N}(\mathbf{k}_j | \mathbf{q}_i, 1)}{\sum_i \beta_i \, \mathcal{N}(\mathbf{k}_j | \mathbf{q}_i, 1)} = \frac{\beta_i \exp(\mathbf{q}_i^\top \mathbf{k}_j - \frac{1}{2} \mathbf{q}_i^\top \mathbf{q}_i)}{\sum_i \beta_i \exp(\mathbf{q}_i^\top \mathbf{k}_j - \frac{1}{2} \mathbf{q}_i^\top \mathbf{q}_i)}, \tag{6}$$

where $\sum_i \xi_{ji} = 1$. If we approximate $\beta_i \propto \exp(\frac{1}{2} \mathbf{q}_i^\top \mathbf{q}_i)$, then

$$\xi_{ji} = \frac{\exp(\mathbf{q}_i^\top \mathbf{k}_j)}{\sum_i \exp(\mathbf{q}_i^\top \mathbf{k}_j)}. \tag{7}$$

Let $\frac{\partial}{\partial \, \mathbf{q}_i} \sum_j \log p(\mathbf{k}_j) = 0$, we have $0 = \sum_j \xi_{ji} (\mathbf{q}_i - \mathbf{k}_j)$, which results in:

$$\mathbf{q}_i^{new} = \sum_j \frac{\xi_{ji}}{\sum_j \xi_{ji}} \, \mathbf{k}_j \, . \tag{8}$$

Comparing (4) and (8), we notice that the only difference between the two updates is the normalization process of the attention weights. Under the Transformer attention scheme LNAS from Eq. (3), its weights π_{ij} are normalized just once over the lower layer neurons j. For DNAS in Eq. (6), the attention weights are computed in two steps: first, an upper-layer normalization is applied over the upper-layer neurons i according to Eq. (7), for every lower-layer neuron j; second, a lower-layer normalization is applied over the lower-layer neurons j according to Eq. (8), for every upper-layer neuron i. The name *doubly-normalized attention* scheme, DNAS, is chosen to reflect this normalization mechanism. It is worth noting that DNAS is closely related to the EM routing algorithm in the capsule networks (Hinton et al., 2018), which is also derived from GMM. Due to space limitations, this connection is discussed in more detail in Appendix A.

4 Attention from the Optimization Perspective

The Transformer LNAS scheme can also be understood from an optimization perspective. Consider the following constrained optimization problem that characterizes π_{ij},

$$\min_{\pi} \sum_{ij} \pi_{ij} D(\mathbf{q}_i, \mathbf{k}_j) + \pi_{ij} \log \pi_{ij} \qquad \text{s.t. } \sum_j \pi_{ij} = 1. \tag{9}$$

Introducing the Lagrange multipliers λ_i, this formulation is equivalent to optimizing the Lagrangian, whose gradient with respect to π_{ij} gives

$$\frac{\partial L(\pi_{ij}, \lambda_i)}{\partial \pi_{ij}} = D(\mathbf{q}_i, \mathbf{k}_j) + 1 + \log \pi_{ij} + \lambda_i,$$

and leads to the same solution as (3) when $D(\mathbf{q}_i, \mathbf{k}_j) := -\,\mathbf{q}_i^\top \mathbf{k}_j$.

The doubly-normalized attention scheme DNAS can be derived from a very similar constrained optimization except that an additional normalization constraint is added to the lower layer neurons:

$$\min_{\pi} \sum_{ij} \pi_{ij} D(\mathbf{q}_i, \mathbf{k}_j) + \pi_{ij} \log \pi_{ij} \qquad \text{s.t. } \sum_i \pi_{ij} = 1, \ \sum_j \pi_{ij} = 1. \tag{10}$$

The above objective function is well-known in the optimal transport literature, and the classical iterative algorithm for finding the solution is called the Sinkhorn algorithm (Peyré & Cuturi, 2019). This algorithm uses the initial condition $\pi_{ij}^0 = \exp(-D(\mathbf{q}_i, \mathbf{k}_j))$, and iterates

$$\xi_{ji}^t = \frac{\pi_{ij}^{t-1}}{\sum_i \pi_{ij}^{t-1}}, \quad \pi_{ij}^t = \frac{\xi_{ji}^t}{\sum_j \xi_{ji}^t}.$$

If we write $D(\mathbf{q}_i, \mathbf{k}_j) := -\,\mathbf{q}_i^\top \mathbf{k}_j$ then the doubly-normalized attention weights in Eq. (8) correspond exactly to the updates of the Sinkhorn algorithm for one iteration.

Comparing the two constrained optimization problems in (9) and (10), the only difference is that the constraint on the attention-weight sum for the lower neurons is removed in (9). The removal of the constraint allows solutions in which a lower-layer neuron j has an arbitrary contribution to the upper layer. This is the reason for which the LNAS scheme suffers from the so-called "explaining-away" effect, as we discuss in the next section.

5 Properties of Doubly-Normalized Attention

5.1 Doubly-Normalized Attention Avoids Explaining Away

Under the original Transformer normalization scheme, LNAS, the mechanism allows the nodes in the higher layer to not attend to some of the nodes in the lower layer. In particular, for a certain lower-layer neuron j, the sum of its attention assignments to the upper layer is $\sum_i \pi_{ij}$, where i denotes the location of the indices over the upper-layer neurons. Since the only restriction under LNAS is $\sum_j \pi_{ij} = 1$, the summation $\sum_i \pi_{ij}$ can be as low as 0, which means the contribution of the lower neuron j is effectively filtered out in the upper layer. This is the very definition of the "explaining-away" phenomenon, which can potentially have negative effects for an encoder: if early encoder layers filter out the contribution of certain inputs that could be useful at later stages in the encoder, that information can no longer be recovered. In contrast to LNAS, the DNAS scheme avoids completely the "explaining-away" phenomenon. The following lemma formalizes this property by showing that each lower-layer neuron gets to contribute with a weight assignment of at least $1/L$, where L is the layer size.

Lemma 1 *For any lower-layer neuron j, the sum of the doubly-normalized attention weights over the upper layer neurons $\sum_i \pi_{ij} = \sum_i \frac{\xi_{ji}}{\sum_j \xi_{ji}}$ is lower bounded by $1/L$.*

The proof of the lemma is provided in Appendix B. To illustrate the difference between the two attention schemes and how different they behave in practice with respect to the "explaining-away" phenomenon, we use the Visual Question Answering setup described in more detail in Sec. 7.1. Fig. 1 shows the minimum attention weight sum values, $\min_j \sum_i \pi_{ij}$, achieved by both LNAS and DNAS during training. As predicted by our analysis, the LNAS scheme has the minimum attention weight sum close to 0 (meaning that it explains-away at least some of its inputs), while the DNAS scheme maintains a minimum attention weight sum above its lower-bound value of $1/L = 0.01$.

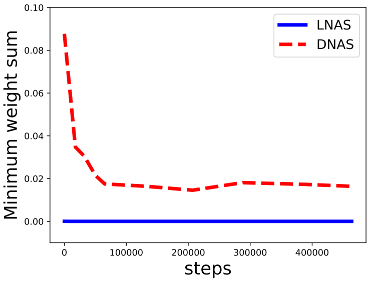


Figure 1: The minimum of attention weight sum in LNAS and DNAS.

5.2 Doubly-normalized Attention Alleviates Mode Collapse

Another shortcoming of the LNAS scheme is its tendency to collapse modes[*]. To study the mode-collapsing effect, we analyze the speed of two clusters approaching each other in a 1D scenario using the two attention schemes. Let us assume we have two sets of data, one containing N_0 data points centered at value a, and another containing N_1 data points centered at value $-a$. The distance between the two centers is $2a$. Assuming the relative distance between the data points within each set is negligible compared to $2a$, the unnormalized attention weights between one center and the data from the other set is $s = \exp(-(2a)^2/2) = \exp(-2a^2)$, and the weights between one center and the data within that set is $t = \exp(0) = 1$[†].

Due to space limitations, the details of the derivations are provided in Appendix C and only the main results are presented here. If $r := N_0/N_1$, after one LNAS update, the new center distance is:

$$c_0^L - c_1^L = \frac{2r(1-s^2)a}{(1+rs)(r+s)}. \tag{11}$$

For DNAS, the new center distance is

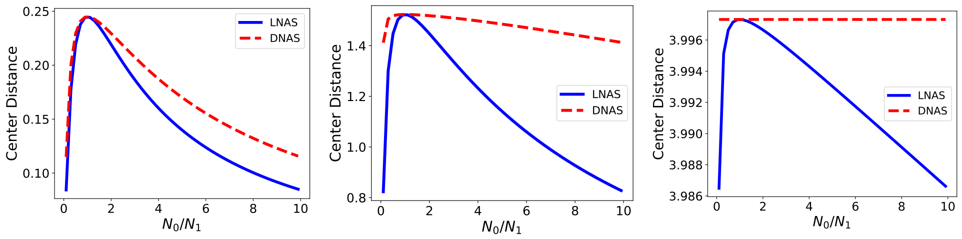
$$c_0^D - c_1^D = \frac{2qr(1-s^2)a}{(q+rs)(r+sq)}. \tag{12}$$

where $q = \frac{r+s}{rs+1}$. To better understand the difference between the values of Eq. (11) and Eq. (12), we plot them on the y-axis against that of $r = N_0/N_1$ on the x-axis, for several different a values, see Fig. 2. We see that in both cases the distance between the two centers decays after the attention updates. However, the center distance of DNAS always upper bounds the one of LNAS, with the gap getting larger as the cluster sizes get more unbalanced ($r \neq 1$).

The mode collapse effect is even more obvious in multi-layer attention. In Appendix C, we show that when the two clusters are unbalanced (Fig. 6), the LNAS collapses to a single cluster after 4 steps, while the DNAS maintains two separate clusters.

[*] Note that most multi-layer attention models such as the Transformer avoid such collapsing effect by adding a residual layer after attention.

[†] The attention weights are computed with a Gaussian. But the same result holds with dot product attention, where the inter-attention weight is $s = \exp(\langle -a, a \rangle) = \exp(-a^2)$ and the intra-attention weight is $t = \exp(\langle a, a \rangle) = \exp(a^2)$. The ratio $s/t = \exp(-2a^2)$ is identical to the Gaussian case.



(a) $a = 0.5$ (b) $a = 1$ (c) $a = 2$

Figure 2: Center distance values after lower-normalized attention (LNAS, blue solid curve) and doubly-normalized attention (DNAS, red dashed curve), as a function of cluster mass ratio $r = N_0/N_1$ with different a values (initial distance between centers is $2a$).

6 Hybrid Attention

In practice, we can also combine the LNAS and DNAS formulations. We investigated doing so with a trainable variable $u \in [0, 1]$ that controls the contribution of the attention weights of the two normalization schemes:

$$\pi_{ij}^H = u\, \pi_{ij}^D + (1 - u)\pi_{ij}^L,$$

where π^D denotes the DNAS weights and π^L denotes the LNAS weights. We call this combination form to be the hybrid normalized attention scheme (HNAS). The HNAS is attractive because it allows the model to learn, at different layers l, which of the two normalization schemes achieves better results; the u_l parameter is trained jointly with the other parameters to improve the representation power of the model and better fit the data. As a side effect, this approach also allows one to visualize how the values of the u_l parameters change as the model is training, and therefore provides direct evidence of how much and where the different normalization schemes lead to better training performance. We provide examples of such visualizations in Sec. 7.

7 Numerical Experiments

7.1 Visual Question Answering

The goal of a visual question answering model is to provide an answer to a natural language question relevant to the contents of a given image. For our first experimental setup, we use a model similar to the one proposed in (Yu et al., 2019a), which uses one attention layer to combine multi-view features, i.e., visual features produced by different image processing modules.

Experiment Setup. We conduct experiments on the most commonly used VQA benchmark dataset, VQA-v2 (Goyal et al., 2017). Our core VQA model uses as a backbone the Pythia architecture (Jiang et al., 2018). Aside from the backbone network, a crucial factor in the performance of a good VQA system is its visual feature extraction. Currently, virtually all high-performing model use bounding-box visual features extracted by object-detector models trained on the Visual Genome Dataset (Krishna et al., 2017). Additionally, Yu et al. (2019a) showed that it is beneficial to use bounding-box features from multiple object detectors. In our experiments, we use three object detection models, where each detector generates 100 bounding-box features. All three object detection models are trained over the Visual Genome dataset. The difference between detection models is in the backbone network: the first uses a ResNet-101 network (He et al., 2016), the second a ResNet-200 network, and the third an Inception-ResNetV2 network (Szegedy et al., 2016).

Multi-view features can be used in a VQA model in a straightforward manner by concatenating them all together before feeding them into the Pythia model; we call this approach the 3x100-boxes baseline. The proposal from (Yu et al., 2019a) combines the multi-view features using a one-layer attention mechanism, as follows: one object-detector model is designated as primary, and its corresponding features are used as queries after transformation; the second and third object detection models are designated as secondary, and their corresponding features are used to obtain keys and values. The resulting output feature is a weighted sum of the features according to the attention weights. With this attention scheme, the original 300 bounding-box features from the three object-detection models are transformed into 100 features, which are then fed into a Pythia model. We experiment with two versions of this scheme: one with an attention mechanism using lower-normalized attention (LNAS) and one using doubly-normalized attention (DNAS).

Results and analysis. The results are summarized in Table 1.

Method	Test-dev	Test-std
100-boxes Pythia (Jiang et al., 2018)	68.31	-
100-boxes (our baseline)	68.33	-
3x100-boxes (our baseline)	68.79	69.22
3x100-boxes LNAS	69.14	69.50
3x100-boxes DNAS	**69.70**	**70.01**

Table 1: Test Accuracy on VQA v2.0, over Test-dev and Test-std splits.

Confirming the findings from (Yu et al., 2019a), we see that using visual features from three object detectors improves performance over using the one from a single object detector (+0.46 on Test-dev). Furthermore, using an attention mechanism over the 3x100 boxes further improves the accuracy over the 3x100-boxes baseline, but the DNAS mechanism achieves a better utilization of the signal provided by the three object detectors compared to the LNAS mechanism (+0.56 on Test-dev and +0.51 on Test-std). Moreover, using the HNAS formulation (Sec. 6) allows us to both visually and empirically confirm the superiority of the DNAS mechanism: as we plot the hybrid weight u in Fig. 3, the hybrid weight rapidly converges to 1.0 in this scenario.

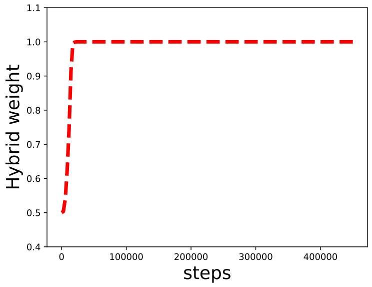


Figure 3: The hybrid weight heavily favors DNAS for the VQA task.

7.2 Headline Generation

The goal of a headline generation model is to abstractively (as opposed to extractively) generate a short, headline-like summary given a text document. Similar to recent work that achieves state-of-the-art results on the task (Goodman et al., 2019), we use an encoder-decoder Transformer architecture with 12 layers of attention.

Experiment Setup. The standard benchmark for the headline generation task is the Gigaword dataset (Graff & Cieri, 2003), which consists of about 4M $\langle article, headline \rangle$ pairs. We pre-process this dataset as in (Rush et al., 2015), which results in an average $article$ length of 31.4 words, and an average $headline$ length of 8.5 words. We further tokenize the words into word-pieces (Devlin et al., 2019), which results in a vocabulary size of 30,522 word-piece types. We use a 10k dataset for validation, and the standard 2k test set (Rush et al., 2015) as the evaluation test.

The backbone model is a Transformer encoder-decoder containing 12 layers, each with a hidden size of 768 and 12 attention heads. We use an Adam optimizer (Kingma & Ba, 2015) and a learning rate of $2e^{-5}$. We truncate (or pad) the input and output sequences to a fixed number of word-piece positions, namely 128 encoder positions and 64 decoder positions, to accommodate hardware and model-architecture limitations. Similar to (Goodman et al., 2019), we run experiments starting from both random initializations and BERT-checkpoint initializations.

Results Analysis. The results are summarized in Table 2. Our Transformer LNAS model reproduces the configuration from (Goodman et al., 2019), and obtains 35.23 ROUGE-L F1 score with random initialization and 36.24 with BERT initialization. Consistent with the previous results, the Transformer HNAS model performs better compared to the LNAS version by +0.5 ROUGE-L F1 points with random intialization and +0.3 ROUGE-L F1 points with BERT initialization. To the best of our knowledge, the ROUGE scores of the Transformer HNAS$\langle$BERT, BERT$\rangle$ configuration establish a new state-of-art on the Gigaword headline generation benchmark.

Method	ROUGE-1	ROUGE-2	ROUGE-L
SEASS (Zhou et al., 2017)	36.15	17.54	33.63
Base+E2Tcnn+sd (Amplayo et al., 2018)	37.04	16.66	34.93
Transformer $\langle \mathrm{Random}, \mathrm{Random} \rangle$ (Goodman et al., 2019)	38.05	18.95	35.26
Transformer $\langle \mathrm{BERT}, \mathrm{BERT} \rangle$ (Goodman et al., 2019)	38.96	19.55	36.22
Transformer LNAS$\langle \mathrm{Random}, \mathrm{Random} \rangle$	38.15	18.73	35.23
Transformer HNAS$\langle \mathrm{Random}, \mathrm{Random} \rangle$	**38.47**	**19.11**	**35.71**
Transformer LNAS$\langle \mathrm{BERT}, \mathrm{BERT} \rangle$	38.90	19.65	36.24
Transformer HNAS$\langle \mathrm{BERT}, \mathrm{BERT} \rangle$	**39.06**	**20.14**	**36.51**

Table 2: ROUGE F1 scores for headline generation on the Gigaword benchmark.

Method	SQuAD 1.1	SQuAD 2.0	MNLI
BERTbase	90.5/83.3	80.3/77.3	84.1
LNAS	90.5/83.6	80.3/77.0	84.3
HNAS	**90.9/84.0**	**80.5/77.4**	**84.7**

Table 3: Pretraining with HNAS and finetuning on SQuAD 1.1 and SQuAD 2.0 (F1/EM) and MNLI.

7.3 Language Representation Learning

The goal of language representation learning is to pretrain textual representations that are useful for solving natural language understanding (NLU) tasks like entailment or question answering.

Experiment Setup. BERT (Devlin et al., 2019) established itself as a high-performing contextual representation model. We report here experiments done with the base setting for BERT: a Transformer network with 12 layers of attention, the hidden and embedding size set to 768, and 12 attention heads. Following recent work (Joshi et al., 2019), we use n-gram masking ($n \leq 3$), with the length of each n-gram mask randomly selected.

We use the BookCorpus (Zhu et al., 2015) and English Wikipedia (Devlin et al., 2019) to pretrain two contextual representation models, one using LNAS and one using HNAS. We evaluate the resulting representations by using them as a starting point to finetune for the SQuAD task (Rajpurkar et al., 2018) and the MNLI task (Williams et al., 2018).

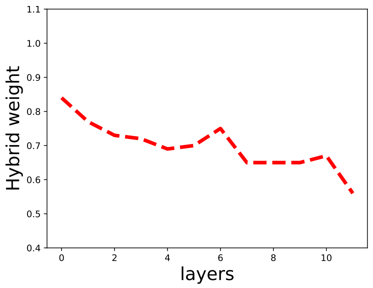


Figure 4: The hybrid weight favors DNAS in all layers of the encoder ($u \geq .5$); LNAS gains more weight for closer-to-output layers.

Results Analysis. The results are summarized in Table 3. The HNAS experimental condition improves over the LNAS condition on all tasks considered. Aside from the numerical improvements when finetuning on the task, we also inspect what happens to the hybrid weight u during HNAS pretraining. In Fig. 4, we plot the hybrid weight for all 12 layers and find that they are always larger than 0.5, meaning that the DNAS method is preferred during the pretraining optimization. In agreement with the GMM view, the plot shows that the LNAS method starts to have more weight for the layers that are closer to the output layer, approaching a point where LNAS and DNAS are similarly useful. This suggests that the resulting network parameters encode their language representations by making use of the advantages of the DNAS method, positively contributing to the resulting empirical advantage of the HNAS-driven language representations.

8 Discussion

The attention mechanism of the Transformer, here called LNAS, can be framed under both a probabilistic perspective and an optimization perspective. The new attention scheme we propose based on these perspectives, called DNAS, compensates for certain shortcomings of the LNAS scheme. Together, these two attention normalization schemes can be combined into a hybrid one, HNAS, which we show here to be empirically superior to the original normalization mechanism. Further research needs to be conducted to thoroughly analyze the interplay of the attention scheme with other Transformer components, such as the residual layer.

References

Reinald Kim Amplayo, Seonjae Lim, and Seung-won Hwang. Entity commonsense representation for neural abstractive summarization. In *Proceedings of NAACL-HLT 2018*, pp. 697–707, 2018.

Hao Cheng, Hao Fang, and Mari Ostendorf. A dynamic speaker model for conversational interactions. In *NAACL-HLT*, 2019.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *NAACL-HLT*, 2018.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. *ArXiv*, abs/1807.03819, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Sebastian Goodman, Zhenzhong Lan, and Radu Soricut. Multi-stage pretraining for abstractive summarization. *CoRR*, 2019.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6904–6913, 2017.

David Graff and Christopher Cieri. English Gigaword Fifth Edition LDC2003T05. In *Linguistic Data Consortium*, Philadelphia, 2003.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of CVPR*, 2016.

Geoffrey Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. 2018. URL https://openreview.net/pdf?id=HJWLfGWRb.

Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Pythia v0. 1: the winning entry to the vqa challenge 2018. *arXiv preprint arXiv:1807.09956*, 2018.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*, 2019.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. Training millions of personalized dialogue agents. In *EMNLP*, 2018.

Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2124. URL https://www.aclweb.org/anthology/P18-2124.

Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of EMNLP*, pp. 379–389, 2015.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual Captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*, 2018.

Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016.

Hao Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. *ArXiv*, abs/1908.07490, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of NeurIPS*, 2017.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL `https://www.aclweb.org/anthology/N18-1101`.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. XLNet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

Jun Yu, Jing Li, Zhou Yu, and Qingming Huang. Multimodal transformer with multi-view visual representation for image captioning. 05 2019a.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *CVPR*, 2019b.

Sanqiang Zhao, Piyush Sharma, Tomer Levinboim, and Radu Soricut. Informative image captioning with external sources of information. In *ACL*, 2019.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. Selective encoding for abstractive sentence summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *ICCV*, 2015.

A Connection to Capsule Networks with EM Routing

DNAS is closely related to the EM routing algorithm in the capsule networks (Hinton et al., 2018). In particular, the vote matrix V_{ij} in (Hinton et al., 2018) is similar to $\mathbf{k}_j$ in Eq. (5); the new pose matrix μ_j in (Hinton et al., 2018) is similar to $\mathbf{q}_i$ in Eq. (5). The difference is that in DNAS, the $\mathbf{q}_i$ is only updated for one iteration, while the pose matrix μ_j is updated until it converges. There is no variance σ_i^2 and β_i estimation in DNAS, as we assume $\sigma_i^2 = 1$ and $\beta_i \propto \exp(\frac{1}{2}\,\mathbf{q}_i^\top\,\mathbf{q}_i)$.

One interesting question is whether DNAS could potentially perform better with more iterations for the updates in Eq. (6) and Eq. (8), as well as updates for σ_i^2 and β_i. In our experiments, we observed that adding more update iterations increases computational time but does not improve the performance significantly. Furthermore, trying to estimate σ_i^2 tends to hurt the empirical performance of the algorithm.

B Proof of Lemma 1

Lemma 1 For any lower-layer neuron j, the sum of the doubly-normalized attention weights over the upper layer neurons $\sum_i \frac{\xi_{ji}}{\sum_j \xi_{ji}}$ is lower bounded by $1/L$.

Proof Since $\sum_i \xi_{ji} = 1$,

$$\begin{aligned}\sum_i \frac{\xi_{ji}}{\sum_j \xi_{ji}} &\geq \sum_i \frac{\xi_{ji}}{\max_i(\sum_j \xi_{ji})} = \frac{\sum_i \xi_{ji}}{\max_i(\sum_j \xi_{ji})} \\ &= \frac{1}{\max_i(\sum_j \xi_{ji})} \geq \frac{1}{\sum_j \max_i(\xi_{ji})} \geq \frac{1}{L}\end{aligned}$$

■

C Details about the mode collapsing analysis

To study the mode-collapsing effect, we analyze the speed of two clusters approaching each other in a 1D scenario using the LNAS and DNAS attention schemes. Let us assume we have two sets of data, one containing N_0 data points centered at value a, and another containing N_1 data points centered at value $-a$. The distance between the two centers is $2a$. Assuming the relative distance between the data points within each set is negligible compared to $2a$, the unnormalized attention weights between one center and the data from the other set is $s = \exp(-(2a)^2/2) = \exp(-2a^2)$, and the weights between one center and the data within that set is $t = \exp(0) = 1$.

Applying Eq. (4) for the LNAS scheme, the new center distance of the lower-normalized attention scheme are:

$$\begin{aligned}c_0^L &= \left(\frac{N_0 t}{N_0 t + N_1 s} - \frac{N_1 s}{N_0 t + N_1 s}\right) a = \frac{N_0 t - N_1 s}{N_0 t + N_1 s} a \\ c_1^L &= \left(\frac{N_0 t}{N_0 t + N_1 s} - \frac{N_1 s}{N_0 t + N_1 s}\right) a = \frac{N_0 s - N_1 t}{N_1 t + N_0 s} a\end{aligned}$$

and the distance between the two updated centers is:

$$c_0^L - c_1^L = \frac{2N_0 N_1 (t^2 - s^2) a}{(N_1 t + N_0 s)(N_0 t + N_1 s)}.$$

Since we have that $t = 1$, defining $r = N_0/N_1$ then gives

$$c_0^L - c_1^L = \frac{2r(1 - s^2)a}{(1 + rs)(r + s)}. \tag{13}$$

By contrast, if we apply the Eq. (8) updates for the DNAS scheme, the new center distance of the doubly-normalized attention scheme are:

$$c_0^D = \left(\frac{\frac{N_0 t}{N_0 t + N_1 s}}{\frac{N_0 t}{N_0 t + N_1 s} + \frac{N_1 s}{N_0 s + N_1 t}} - \frac{\frac{N_1 s}{N_0 s + N_1 t}}{\frac{N_0 t}{N_0 t + N_1 s} + \frac{N_1 s}{N_0 s + N_1 t}} \right) a = \frac{N_0 t (N_0 s + N_1 t) - N_1 s (N_0 t + N_1 s)}{N_0 t (N_0 s + N_1 t) + N_1 s (N_0 t + N_1 s)} a$$

$$c_1^D = \left(\frac{\frac{N_0 s}{N_0 t + N_1 s}}{\frac{N_0 s}{N_0 t + N_1 s} + \frac{N_1 t}{N_0 s + N_1 t}} - \frac{\frac{N_1 t}{N_0 s + N_1 t}}{\frac{N_0 s}{N_0 t + N_1 s} + \frac{N_1 t}{N_0 s + N_1 t}} \right) a = \frac{N_0 s (N_0 s + N_1 t) - N_1 t (N_0 t + N_1 s)}{N_0 s (N_0 s + N_1 t) + N_1 t (N_0 t + N_1 s)} a,$$

and the distance between the two updated centers is:

$$c_0^D - c_1^D = 2 N_1 a \left(\frac{t (N_0 t + N_1 s)}{N_0 s (N_0 s + N_1 t) + N_1 t (N_0 t + N_1 s)} - \frac{s (N_0 t + N_1 s)}{N_0 t (N_0 s + N_1 t) + N_1 s (N_0 t + N_1 s)} \right).$$

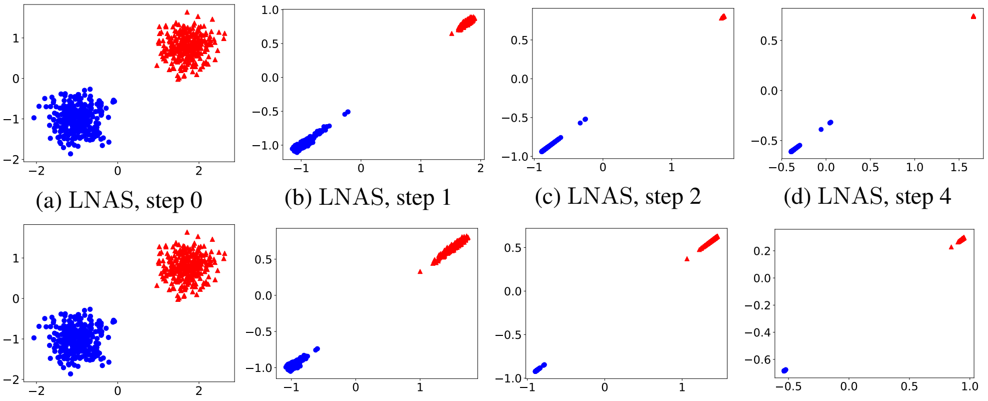
Since again $t = 1$, defining $r = N_0 / N_1$ and $q = \frac{N_0 t + N_1 s}{N_0 s + N_1 t} = \frac{r+s}{rs+1}$ then yields

$$c_0^D - c_1^D = \frac{2qr(1 - s^2)a}{(q + rs)(r + sq)}. \tag{14}$$

We plot them on the y-axis against that of $r = N_0/N_1$ on the x-axis, for several different a values, see Fig. 2.

We see that in both cases the distance between the two centers decays after one attention updates. However, the center distance of the doubly-normalized attention mechanism always upper bounds the one of the lower normalization. For $r = 1$, the reduction is minimized and the center distances of both normalization schemes decay at the same rate. When r is greater than or less than 1, the center distance of the lower layer normalization decays much faster than that of the double normalization. The latter maintains a slow decay rate until r is extremely small or large.

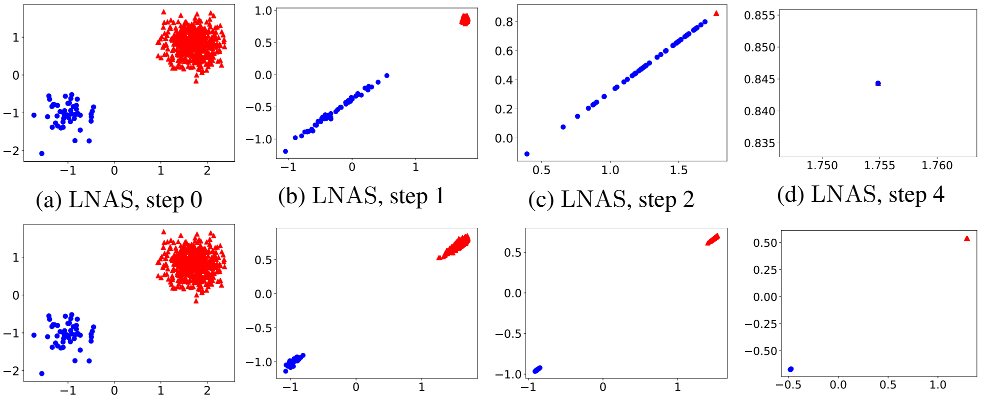
The mode collapse effect is even more obvious in multi-layer attention. In Fig. 5, when the two clusters are balanced, both normalization schemes yield similar results. However, when the two clusters are unbalanced (Fig. 6), the lower normalization collapses to a single cluster after 4 steps, while the doubly-normalized scheme maintains two separate clusters.



(a) LNAS, step 0 (b) LNAS, step 1 (c) LNAS, step 2 (d) LNAS, step 4

(e) DNAS, step 0 (f) DNAS, step 1 (g) DNAS, step 2 (h) DNAS, step 4

Figure 5: Mode-collapsing behavior on balanced mixture of Gaussian data: LNAS and DNAS behave similarly without mode collapsing after 4 steps.



(a) LNAS, step 0 (b) LNAS, step 1 (c) LNAS, step 2 (d) LNAS, step 4

(e) DNAS, step 0 (f) DNAS, step 1 (g) DNAS, step 2 (h) DNAS, step 4

Figure 6: Mode-collapsing behavior on unbalanced mixture of Gaussian data: LNAS collapses to one cluster after 4 steps, while DNAS maintains 2 clusters.