On the Interpretability of Attention Networks

Lakshmi Narayan Pandey* LNPANDEY.IITM@GMAIL.COM
Department of CSE, IIT Madras, Chennai, India.

Rahul Vashisht* RAHUL@CSE.IITM.AC.IN
Department of CSE, IIT Madras, Chennai, India.

Harish G. Ramaswamy HARIGURU@CSE.IITM.AC.IN
Department of CSE, IIT Madras, Chennai, India.

Editors: Emtiyaz Khan and Mehmet Gönen

Abstract

Attention mechanisms form a core component of several successful deep learning architectures, and are based on one key idea: "The output depends only on a small (but unknown) segment of the input." In several practical applications like image captioning and language translation, this is mostly true. In trained models with an attention mechanism, the outputs of an intermediate module that encodes the segment of input responsible for the output is often used as a way to peek into the 'reasoning' of the network. We make such a notion more precise for a variant of the classification problem that we term selective dependence classification (SDC) when used with attention model architectures. Under such a setting, we demonstrate various error modes where an attention model can be accurate but fail to be interpretable, and show that such models do occur as a result of training. We illustrate various situations that can accentuate and mitigate this behaviour. Finally, we use our objective definition of interpretability for SDC tasks to evaluate a few attention model learning algorithms designed to encourage sparsity and demonstrate that these algorithms help improve interpretability.

Keywords: Interpretability; Attention models; Deep Learning

1. Introduction

Attention mechanisms Devlin et al. (2019); Seo et al. (2017); Vaswani et al. (2017); Xu et al. (2015) have had a phenomenal success in deep learning, and are used in almost every state of the art model for several tasks. In particular, machine translation, handwriting synthesis, image to text generation, audio classification, text summarization and audio synthesis tasks have shown state of the art performances using attention mechanisms Bahdanau et al. (2015); Chorowski et al. (2014); Graves (2013); Xu et al. (2015).

The intuitive idea of attention is very simple and appealing and hence easily extensible: "The output depends only on a small (but unknown) segment of the input". There have been several works diving deep into attention models proposing variants and analysing performance. A particularly appealing (but debatable) property of attention models is that the part of the network that 'focuses' on a segment of the input is often easily interpretable

* The first two authors contributed equally to this work.

in the case of both correct and wrong predictions Xu et al. (2015); Wang et al. (2016); Jain and Wallace (2019); Wiegreffe and Pinter (2019).

While there have been several subjective studies/statements on how attention is interpretable Jain and Wallace (2019); Wiegreffe and Pinter (2019), to the best of our knowledge there has been no quantitative study demonstrating the interpretability of attention. In this paper, we take a step towards defining a framework for such a quantitative study.

1.1. Contributions

The contributions of this paper are listed below.

1. We define crude but quantifiable measures of interpretability on a standard image captioning task and show that standard well known attention models are indeed more interpretable than a random network.

2. We define and introduce a new type of machine learning problem/task, which we call the *selective dependence classification* (SDC) problem. It captures the essence of problems where attention mechanisms are motivated as a reasonable solution.

3. We define a simplified attention mechanism called the *Focus-Classify Attention Model* (FCAM) that is apt for SDC problems. Using FCAM for SDC tasks allows for a natural and objective notion of interpretability.

4. We describe and illustrate various modes of operation of an FCAM on SDC tasks, and example conditions under which the attention model trains well, generalises well, but has poor interpretability.

5. We experimentally analyse some variants of attention models that are designed to improve performance and interpretability via sparsity.

1.2. Related Works

Jain and Wallace (2019) and Wiegreffe and Pinter (2019) study the relationship between the attention vector $\boldsymbol{\alpha}$ and real world explanations.

There have been links made between multiple instance learning (MIL) Maron and Lozano-Pérez (1998); Sabato and Tishby (2012) and attention models Ilse et al. (2018). Attention models are shown to be a good tool to solve MIL problems, but MIL problems do not capture the essence of attention models used in practical architectures.

Latent variable alignment Deng et al. (2018), is standard problem used in the analysis of attention models, and is used as the theoretical basis for deriving the various loss functions used in attention models via approximations of maximum likelihood. The SDC task defined in this paper can be viewed as a special case of latent variable alignment with a few extra assumptions that make the measurement of interpretability easier.

1.3. Interpretability and Attention models

In numerous NLP tasks and other domains, neural attention has emerged as a critical component of many state-of-the-art results. One primary reason for such results is often

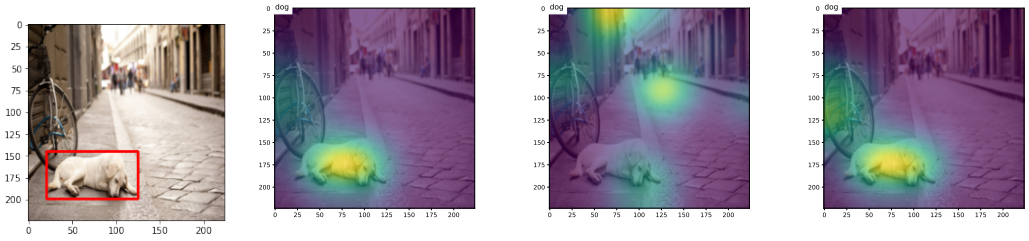

Figure 1: Attention Analysis showing (a) Original image and bounding box for the 'dog' object, (b) Heatmap overlay visualisation of attention vector $\boldsymbol{\alpha}$ when the word 'dog' is output (c) Heatmap overlay of a random sparse attention vector $\boldsymbol{\alpha}'$ (d) Heatmap overlay of a sparse version of the model output $\boldsymbol{\alpha}$.

attributed to the ability of these models to attend to a specific part of the input for the downstream task. To this part, we raise the following questions

1. How can we compare the performance of attention models on a particular task?
2. How can we quantitatively measure the interpretability of attention models?

The main character in the play of the attention mechanism is the attention vector. Intuitively, the attention vector specifies which part of the input is responsible for the downstream task. In this paper we analyse the attention vector by comparing it with the 'ideal' selection that chooses only the relevant input segment.

2. Interpretability in Image Captioning: A Case Study

Let us consider the image captioning task based on the sequence-to-sequence encoder decoder with the attention model in Xu et al. (2015). A convolutional neural network (CNN) is used to extract feature vectors, referred to as annotation vectors, $\mathbf{a}_i \in \mathbb{R}^d$, where $i \in [m]$ represents a segment/patch of the image. A decoder that outputs one word at a time then selectively focuses on certain patches of the image by selecting a subset of annotation vectors. The size of extracted feature vector is $14 \times 14 \times 512$, which corresponds to $m = 196$ annotation vectors, each of which is a $d = 512$ dimensional representation of a patch of an image. For each $1 \leq i \leq 196$, the attention mechanism generates a positive weight $\boldsymbol{\alpha}_i$, which can be interpreted as the probability that location i is the right place to focus to generate the current word. Thus each generated word in the caption, has a 196 dimensional 'attention vector' $\boldsymbol{\alpha} \in \mathbb{R}_+^{196}$, whose values represent the part of the image the model is currently focused on.

We use the following method to measure the 'interpretability' of the model trained on the COCO dataset Lin et al. (2015). The image captioning model is trained only using the image and captions, but the COCO dataset also has additional metadata in the form of 80 'objects' and bounding box information for every occurrence of these objects in the images. This additional information can be used for quantifying interpretability of the trained model

Table 1: Quantified Attention on Validation Data

Attention Weight	full	random 20	top 20
Average Attention	0.539	0.24	0.429

in an objective way. All the 80 objects are manually associated with a few words that are commonly used to refer to them (e.g. we associated the *people* class with the words *man, woman, guy, boy, girl, people*. Tables **??**, **??** and **??** in the appendix gives words associated with different objects). Each image has typically 2 to 3 objects, and the bounding box of these objects are encoded by a $224*224 = 50176$-dimensional $\{0,1\}$ vector $\mathbf{v}$. The attention vector $\boldsymbol{\alpha} \in [0,1]^{196}$ when the captioning model predicts any of the words associated with any of the objects in the image is also noted. Ideally, (an up-sampled version of) the vector $\boldsymbol{\alpha}$ should align with the vector $\mathbf{v}$ associated with the bounding box of the corresponding object for good interpretability (refer appendix section **??** for an example). The cosine of the angle between $\boldsymbol{\alpha}$ (upsampled) and $\mathbf{v}$, averaged over all occurrences of 'object words' output by the model on the validation set was observed to be 0.54. For comparison, the average cosine of the vector $\mathbf{v}$ with a random vector $\boldsymbol{\alpha}'$ that set 20 of the 196 co-ordinates at random to 1 and the rest to 0 was observed to be 0.24. The average cosine of the vector $\mathbf{v}$ with a vector $\boldsymbol{\alpha}''$ that sets the top 20 co-ordinates of the $\boldsymbol{\alpha}$ vector to 1 and the rest to 0 was observed to be 0.43. These numbers in Table 1 support the idea that the attention vector $\boldsymbol{\alpha}$ aligns significantly with the ground truth location of the relevant object, despite the training data for the model containing no location cues like bounding boxes. See Figure 1 for an illustrated example.

3. Selective Dependence Classification and Focus-Classify Attention Models

In the example analysis in Section 2, there are several issues that make measuring the 'interpretability' or 'goodness' of the attention model ill-defined. To this end, we consider a new task motivated by the basic philosophy of attention.

3.1. Selective Dependence Classification (SDC)

A selective dependence classification (SDC) problem is a k-class classification problem with a special structure on the true labelling function. The instance $\mathbf{x} \in \mathbb{R}^{d \times m}$, contains m parts or segments, each of which is represented by a vector in $\mathbb{R}^d$. The instance $\mathbf{x}$ is called a mosaic instance/image to capture the idea that it is made of many parts. One out of these m segments is called a 'foreground' segment, and the rest are called 'background' segments. A labelling function $L : \mathbb{R}^{d \times m} \to [k]$ labels each mosaic instance into one of k-classes. However, the output of this labelling function only uses the 'foreground' segment of the input.

More formally,

$$L([\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m]) = g^*(\mathbf{x}_{i^*})$$

where i^* is the index of the foreground segment, and $g^* : \mathbb{R}^d \to [k]$ is a function that gives the true label and takes the foreground segment as input. The training data for the task is simply the collection of pairs $\mathbf{x}, \mathbf{y}$ *without* the knowledge of foreground segment index i^*.

The labelling function has a specific structure that uses only a portion of the input for labelling any given instance. This toy problem is analogous to an image classification problem where each image is labeled only based on a small object occupying only a fraction of the pixels. The key difficulty in this problem is that the identity of the foreground segment is not known in the training data. In principle, this extra structure can be ignored and SDC can be viewed simply as a multiclass classification problem over an input domain of dimension dm, but this is clearly inefficient and we observed that any model that tries to treat this as a direct classification problem has a good training performance but perform close to random on the test set (possibly because the inherent invariance of the class label when the segments of a mosaic instance are permuted is hard to learn without encoding explicitly).

For concreteness, and to make the problem well-defined, the segments of any given mosaic instance are assumed to be independently distributed, conditioned on whether the segment is foreground or background.

We let distributions $D_1, D_2, \ldots, D_k$ over $\mathbb{R}^d$ denote the class-conditional distribution of the foreground segments for class label $y = 1, \ldots, k$, and let D_0 denote the class-conditional distribution of the background segments. The generative model of a *mosaic instance-label* pair (X, Y) is described as follows. Some additional details and an illustration of the SDC problem is given in the appendix.

Algorithm 1 Generative Model for an Instance-Label Pair in the Selective Dependence Classification Problem

Input: Number of segments m, Base Distributions $D_1, \ldots, D_k$ and D_0.
$i^* = \text{Random}(\{1, 2, \ldots, m\})$
$y = \text{Random}(\{1, 2, \ldots, k\})$
For $i = 1$ **to** m, $i \neq i^*$:
  $\mathbf{x}_i = \text{Independent-Draw}(D_0)$
$\mathbf{x}_{i^*} = \text{Independent-Draw}(D_y)$
Return $([\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m], y)$

The SDC task is cartoon-like in nature and it is unrealistic to expect real data (e.g. image patches) to satisfy assumptions such as full label dependence on a single foreground part/segment or independence of background parts/segments. However, we argue that the SDC task is the right tool to study the various moving parts of the attention mechanism. The SDC task can enable the understanding of crucial aspects such as optimisation, generalisation and interpretability of attention models on real world data.

3.2. Focus-Classify Attention Models (FCAM)

The SDC problem is a prime candidate for the application of the attention mechanism. The Focus-Classify Attention Model (FCAM) is an extremely simple attention model defined as follows. A 'focus network' scores all the segments on its chance for being a foreground

segment. Ideally, only one of the segments would score high in such a model. A linear combination of the segments, weighted based on the focus network's score is then fed to a 'classification network', which just attempts to classify a single d-dimensional feature vector into one of k-possible classes. One can easily see that, such a model is much simpler than a direct model that takes a full mosaic data as input, and outputs one of the classes.

More concretely, a FCAM, consists of two functions f and $\mathbf{g}$, where $f : \mathbb{R}^d \to \mathbb{R}$ is the focus model and $\mathbf{g} : \mathbb{R}^d \to \mathbb{R}^k$ is the classification model. The focus model f and the classification model $\mathbf{g}$ are from a family of functions $\mathcal{F}$ and $\mathcal{G}$. In practice, $\mathcal{F}$ and $\mathcal{G}$ correspond to neural architectures. The output of the focus model forms a natural intermediate output

$$\boldsymbol{\alpha}(\mathbf{x}) = \text{Softmax}([f(\mathbf{x}_1), \ldots, f(\mathbf{x}_m)])$$

which is called the focus or attention vector. The entire FCAM $h : \mathbb{R}^{d \times m} \to \Delta_k$ is given by:

$$h(\mathbf{x}) = \text{Softmax}\left(\mathbf{g}\left(\sum_{j=1}^{m} \alpha_j(\mathbf{x})\mathbf{x}_j\right)\right)$$

The weighted combination of the segments $\widetilde{x} = \sum_{j=1}^{m} \alpha_j(\mathbf{x})\mathbf{x}_j \in \mathbb{R}^d$ is called the attended/aggregated input/data point.

If the FCAM is well-trained for a SDC problem, a reasonable and intuitive focus vector $\boldsymbol{\alpha}(\mathbf{x})$ would have a 1 in the position corresponding to the foreground segment, and 0 everywhere else. Such a focus model would be a part of an interpretable FCAM as the focus vector would essentially point to the part of the input responsible for the output. Hence, a natural measure of interpretability is given by the fraction of mosaic instances where the focus network f scores the foreground segment higher than all the background segments,

$$\text{FT}[f] = \mathbf{E}[\mathbf{1}(f(X_m) > \max(f(X_1), \ldots, f(X_{m-1}))]$$

where the segments $X_1, \ldots X_{m-1}$ are drawn i.i.d. from D_0, X_m is drawn from D_Y with Y being drawn from the prior label distribution. FT$[f]$ can be estimated when the side information of the foreground index is available for a held-out data set. FT$[f]$ is an objective measure of interpretability free of subjective biases. However it is limited in scope and applies only to SDC tasks learnt using FCAM.

A FCAM with high FT gives confidence to the end-user in using $\boldsymbol{\alpha}(\mathbf{x})$ as an explanation for the decision made by the model, and also indicate that the final model decision is truly made based on the relevant foreground component. Other measures of interpretability, including those based on the gradient (such as sensitivity, GradCAM Selvaraju et al. (2017) etc.) are based on local heuristics and do not have this guarantee.

In the rest of the paper we will restrict our attention to SDC problems learnt using a FCAM model unless mentioned otherwise.

4. Accuracy Does Not Imply Interpretability

The FCAM is trained only for maximizing accuracy (the SDC training data does not even contain the foreground segment index i^*) and hence we can only reasonably expect a well-trained FCAM to be accurate. In practice, however, trained attention models are also

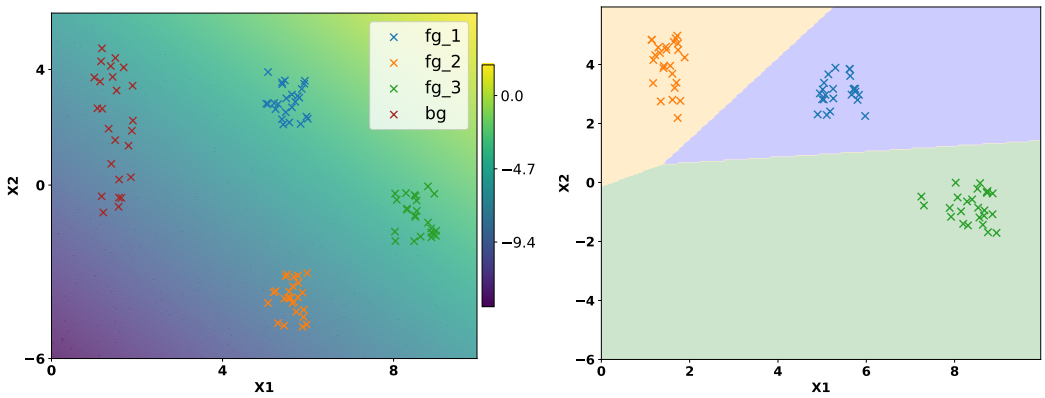

Figure 2: First Error mode illustrations. (**a**) Trained focus model f overlaid as a heat map over a scatter plot of points from the base distributions D_0, D_1, D_2 and D_3. (**b**) Decision boundaries of the trained classification model $\mathbf{g}$ overlaid with a scatter plot of the attended input $\widetilde{\mathbf{x}}$ with the focus model f illustrated in part (a).

assumed to be interpretable, and the output of the focus module (or an equivalent object) is often used as an 'explanation' for the class output.

In this section, we ask a natural question "Do accurate FCAMs focus correctly?". We show three distinct modes (which we call error modes) in which accurate FCAMs fail to do so. All the examples in this section are constructed using synthetic examples of $D_0, D_1, \ldots, D_k$ with base dimension $d = 2$ for easy visualization of the failure modes.

4.1. First Error Mode

Figure 2(a) shows an example of scatter plot samples from base distributions D_0, D_1, D_2 and D_3 with base dimension $d = 2$ and number of foreground classes $k = 3$. Mosaic instances are created using Algorithm 1 as discussed in section 3.1. Consider a focus network in FCAM given by $f(x_1, x_2) = x_1 + x_2$. This focus network gives the foreground segment a higher score than a background segment when the class label is $y = 1$ or $y = 3$ (corresponding to the blue and green points scoring higher than the red points in the focus model f). However, when $y = 2$, the background segment is likely to get a higher score (corresponding to the observation that several orange points score lower than some red points in the focus model f). This results in the distribution of attended input $\sum_{j=1}^{m} \alpha_j(\mathbf{x})\mathbf{x}_j$ to be similar to D_1 or D_3 when $y = 1, 3$, but when $y = 2$ the attended input distribution is skewed significantly. The focus network effectively mistakes one of the foreground classes as the background, but there still exists a simple linear multi-class classifier which can classify the attended input with full accuracy. (See Figure 2(b)). This FCAM model would thus have 100% accuracy but only about 67% interpretability or FT (discussed in section 3.2).

The focus model f and classification model $\mathbf{g}$ illustrated in Figures 2(a) and 2(b) are obtained by training an FCAM corresponding to linear models for $\mathcal{F}$ and $\mathcal{G}$ on an SDC task with 200 mosaic training points, each having $m = 9$ segments. This shows that, while there clearly exist linear functions $f, \mathbf{g}$, that achieve 100% FT and 100% accuracy, SGD

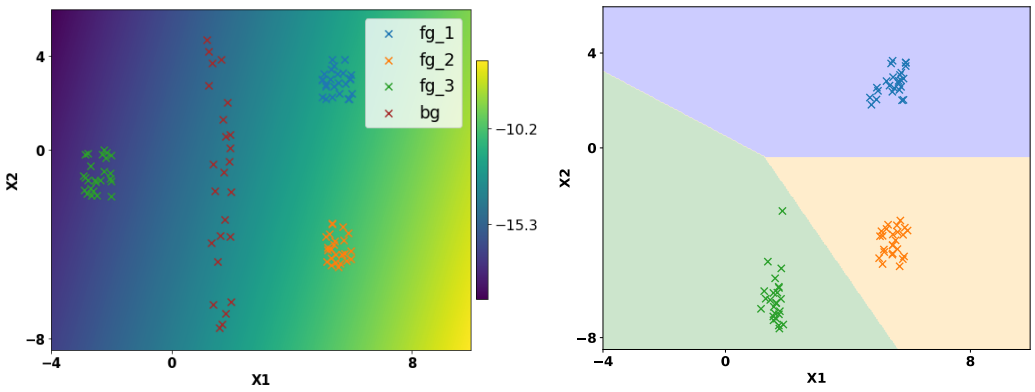

Figure 3: Second Error mode illustration. (**a**) Trained focus model f overlaid on scatter plot of the base distributions. (**b**) Decision boundary of the trained classification model $\mathbf{g}$ overlaid on a scatter plot of the attended data $\widetilde{\mathbf{x}}$.

optimisation does not guarantee that this solution is reached. Using the trained f as an insight into why a mosaic instance $\mathbf{x}$ is labelled as a certain class by the trained FCAM using $\boldsymbol{\alpha}(\mathbf{x})$ is thus not always helpful. (e.g. it might say the reason an image is classified as dog is because of a patch of blue sky.)

4.2. Second Error Mode

With simple hypotheses class $\mathcal{F}$ and $\mathcal{G}$ for the focus and classification network, it is possible there exists no good focus model or classification model individually, but an accurate FCAM model can be achieved by a 'wrong' f and 'wrong' $\mathbf{g}$, thus effectively the two wrongs righting each other.

Figure 3 illustrates such an example (with $d = 2$ and $k = 3$) where both $\mathcal{F}$ and $\mathcal{G}$ are linear models. It can be clearly seen that there exists no linear separator separating the background from the foreground. However the focus model f and classification model $\mathbf{g}$ illustrated in Figure 3 constitute an accurate attention model.

The FCAM illustrated in the Figure 3(a,b) (got by training on 200 mosaic training points with $m = 9$ segments) achieves 100% accuracy but only about 67% FT. The reason however is different from that of the first error mode: a focus net with 100% FT simply cannot be represented using a linear architecture $\mathcal{F}$.

4.3. Third Error Mode

One other potential error mode that can result in an accurate but non-interpretable attention model is when the classification network is powerful enough to classify the attended input $\widetilde{\mathbf{x}} = \sum_{j=1}^{m} \alpha_j(\mathbf{x})\mathbf{x}_j$ even when the focus model is close to its initial parameter, where it is effectively a constant, resulting in $\alpha_j(\mathbf{x}) = \frac{1}{m}$ for all j and $\mathbf{x}$. In Figure 4(a), we give such an example, with $d = 2, k = 6$. There clearly exists a good focus and classification network, even if $\mathcal{F}$ and $\mathcal{G}$ are linear models. But there is no strict necessity for the focus model to be particularly good, as even with a focus function f that is identically zero (which results

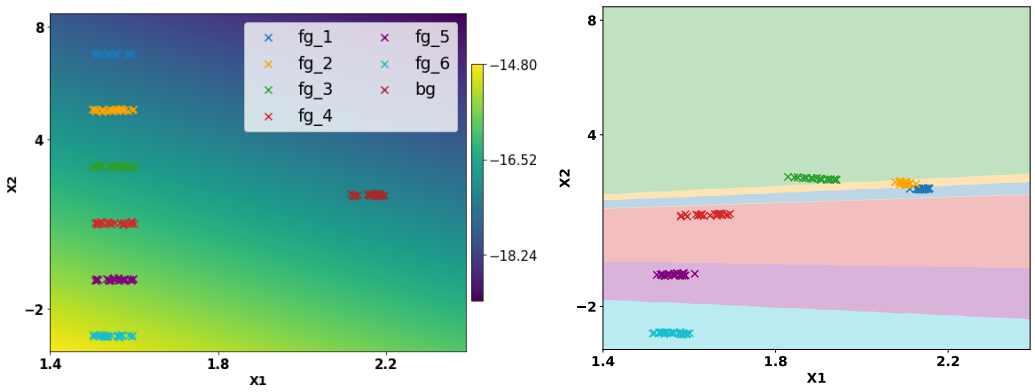

Figure 4: Third Error mode illustration. (**a**) Trained focus model f overlaid on a scatter plot of the base distributions. (**b**) Decision boundary of the trained classification model $\mathbf{g}$ with scatter plot of the attended data $\widetilde{\mathbf{x}}$.

in $\alpha_j(\mathbf{x}) = 1/m$ for all $\mathbf{x}, j$), the resulting attended/aggregated input can be separated by a classification network $\mathbf{g}$.

Figure 4(a,b) illustrate the trained focus (with scatter plot samples from the base data) and classification net (with a scatter plot of the attended data using the trained focus net), on a simulated SDC task with 200 mosaic training points, each having 9 segments. The FCAM used a linear model for $\mathcal{F}$ and a 2-layer ReLU network for $\mathcal{G}$. The FCAM model as a whole is accurate, but the trained focus model chooses to focus on a background segment over foreground segment when the class label is $y = 1, 2, 3$ (corresponding to blue, orange and green points in Figure 4(a)). This happens even though there exists a simple good focus model (e.g $f(x_1, x_2) = -x_1$), because the classification model $\mathbf{g}$ learns to distinguish the attended input $\widetilde{\mathbf{x}}$ based on the class label y, even before a good focus model is learnt.

The optimisation/training dynamics of the three illustrations for the error modes are given in Figure 5(a,b,c). In all three cases the FCAM eventually converges to a model with 100% accuracy and about 70% FT. The convergence under the first error mode was observed to be much slower than the other two modes. We believe these three modes (or a combination of these) to be the main error modes that cause an accurate attention model to still be non-interpretable. Investigating other modes under which this can happen is a direction of future work.

5. Architectures And Loss functions for Interpretability

We have seen various error modes where an FCAM model is accurate, but not interpretable. This brings us to a practical question of 'what changes to the architecture/model/objective can be made to improve the interpretability without sacrificing the accuracy of the model?'. As the identity of the foreground segment is not known in the training data, any such modification will have to be based on other signals.

One common approach studied in attention mechanisms, usually with a view towards increasing accuracy Zhang et al. (2019), is to enforce sparsity constraints on the attention

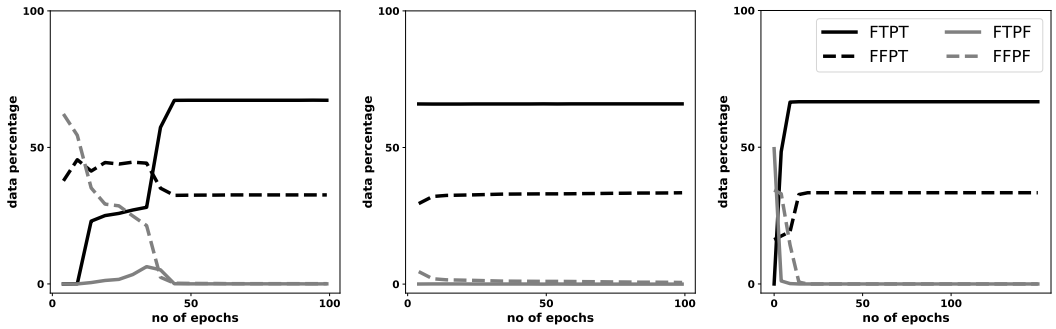

Figure 5: Training dynamics illustrating the fraction of mosaic instance that are being focussed correctly/wrongly and classified correctly/wrongly. (FTPT stands for Focus True Predicted True, FFPT stands for Focus False Predicted True, FTPF stands for Focus True Predicted False, FFPF stands for Focus False Predicted False) (**a**) First Error mode, (**b**) Second Error mode, (**c**) Third Error mode

vector $\boldsymbol{\alpha}$. This seems like an intuitive choice, as forcing the network to always choose only one (or a few) out of the m segments could force the focus model to give maximum score to the foreground segment. Indeed, in our preliminary experiments we discovered the following. The attention vector $\boldsymbol{\alpha}(\mathbf{x})$ is non-sparse in most situations where the focus model is incorrect, but the FCAM model as a whole is accurate.

We test the impact of the following modification to the loss function, where we add an entropy regulariser $\lambda \text{Ent}(\boldsymbol{\alpha}(\mathbf{x}))$ term to the standard cross-entropy loss. The entropy function $\text{Ent}(\boldsymbol{\alpha}(\mathbf{x}))$ has the least value when the attention vector is sparse. The hyperparameter λ is chosen to balance the sparsity regulariser and the cross entropy loss function.

Another approach that can be followed to introduce sparsity in the attention vector $\boldsymbol{\alpha}$ is to use activation functions other than softmax which results in sparse probability distribution such as sparsemax Duchi et al. (2008); Martins and Astudillo (2016); Laha et al. (2018), spherical softmax Ollivier (2013); Vincent et al. (2015); de Brébisson and Vincent (2016); Laha et al. (2018).

Sparsemax function gives the Euclidean projection of input vector $\mathbf{z}$ on to the probability simplex, thus resulting in sparse posterior distribution. Sparsemax is defined as $\rho(z) = \text{argmin}_{p \in \Delta_k} ||p - z||^2$ where Δ_k is the probability simplex.

Spherical softmax is a spherical alternative to softmax activation and also produces sparse probability distribution when most of its components z_i are close to zero. Spherical softmax is defined as $\rho_i(z) = \dfrac{z_i^2}{\sum_{i=1}^{m} z_i^2}$.

We also consider the hard attention mechanism Mnih et al. (2014); Xu et al. (2015), which chooses a single patch j at random with probability $\alpha_j(\mathbf{x})$ instead of an additive combination. In training, this corresponds to weighting the log loss of $\mathbf{g}$ on $(\mathbf{x}_j, y)$ by $\alpha_j(\mathbf{x})$. In the prediction phase, the patch $j^* = \text{argmax}_j\, \alpha_j(\mathbf{x})$ is fed to the classification network $\mathbf{g}$. Training the classification network is about a factor m costlier with hard attention than soft attention, as each mosaic data point $(\mathbf{X}, y)$ corresponds to m data points $(\mathbf{x}_1, y), \ldots, (\mathbf{x}_m, y)$.

Algorithm	averaging layer	attention mechanism	accuracy	FT	NNZ($\boldsymbol{\alpha}$)	Dist($\boldsymbol{\alpha}$)	Ent($\boldsymbol{\alpha}$)
SM-0	zeroth	Softmax (SM)	95.04	79.77	4.2315	0.2594	1.0272
ER-0	zeroth	Entropy reg.	94.76	79.97	3.4927	0.2113	0.8111
SpMax-0	zeroth	Sparsemax	95.8	91.43	1.91	0.181	0.4866
SSM-0	zeroth	Spherical SM	95.17	91.07	2.613	0.2539	1.064
HA-0	zeroth	Hard attention	92.13	87.63	1.311	0.0259	0.070
SM-6	sixth	Softmax (SM)	95.12	86.21	4.7424	0.2776	1.1298
ER-6	sixth	Entropy reg.	95.39	83.64	3.6073	0.2175	0.8466
SpMax-6	sixth	Sparsemax	94.23	85.54	2.46	0.2466	0.7146
SSM-6	sixth	Spherical SM	95.07	91.34	4.7521	0.3005	1.2693
HA-6	sixth	Hard attention	74.45	86.5	1.5031	0.0378	0.1150

Table 2: Performance on CIFAR-SDC Dataset: Standard FCAM and variants.

Another architectural effect that we study has to do with the layers at which the input is averaged. In practical attention models, the input fed to the classification model is $\sum_{i=1}^{m} \alpha_i(\mathbf{x})\boldsymbol{\phi}(\mathbf{x}_i)$, where $\boldsymbol{\phi}$ is the feature mapping corresponding to the last layer (or the last convolutional layer Xu et al. (2015)) of the focus network. In our analysis till now, we had just set $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$ (or the zeroth layer output) for simplicity, but most of the analysis can be extended to the version where $\boldsymbol{\phi}(\mathbf{x})$ is the output of a hidden layer in the focus network.

6. Experiments

We performed experiments on one synthetic SDC dataset and one semi-synthetic dataset based on CIFAR-10. The details of the synthetic data experiment can be found in the appendix.

6.1. CIFAR-SDC Dataset and Architecture Used

The CIFAR-SDC is a semi-synthetic dataset derived from CIFAR10 Krizhevsky and Hinton (2009), with each segment being an image with $d = 3072$, and number of foreground classes $k = 3$. The foreground segments are drawn from the first three classes (`car`, `plane` and `bird`) and the background segments are drawn from the other seven classes. The number of segments per mosaic instance is set as $m = 9$. We sample 40000 such mosaic instances using Algorithm 1 discussed in section 3.1, and set aside 10000 points for testing.

We use a convolutional neural network (CNN) based architecture with both f and $\mathbf{g}$ having 6 convolutional layers followed by a two fully connected layers. An illustration of this dataset and architecture is given in Figure **??** in the appendix.

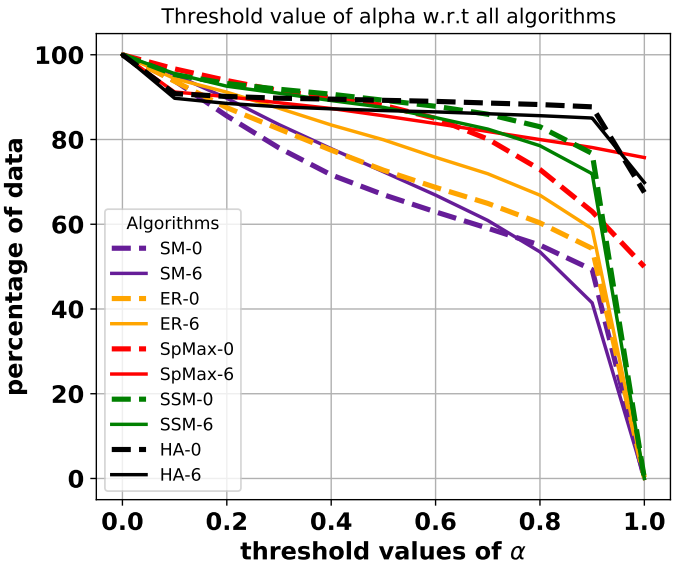

Figure 6: Fraction of test data for which the focus score α_{j^*} for the true foreground index j^* is above a threshold, plotted as function of the threshold for the algorithms mentioned in Table 2

6.2. Experimental Results

The results of the standard FCAM and the variants encouraging sparsity on the two SDC problems are given in Table 2 and Table **??** (refer Appendix). The results in Table 2 are averaged over 3 runs. Figure 6 shows the fraction of instances for which the attention vector $\boldsymbol{\alpha}$ scores the true foreground index above a threshold. We also report the average sparsity of the $\boldsymbol{\alpha}$ vector for each of the methods using following metrics.

1. $\text{NNZ}(\boldsymbol{\alpha}) = \frac{1}{m}\sum_{j=1}^{m} \mathbf{1}(\alpha_j > 0.01)$
2. $\text{Dist}(\boldsymbol{\alpha}) = \min_{j\in[m]} \|\boldsymbol{\alpha} - \mathbf{e}_j\|_2$ where $\mathbf{e}_j \in [0,1]^m$ is the j^{th} co-ordinate vector.
3. $\text{Ent}(\boldsymbol{\alpha}) = \sum_{j=1}^{m} -\alpha_j \log(\alpha_j)$

In the CIFAR-SDC dataset results in Table 2 we observe that all the algorithms achieve high test accuracy (of about 95%) and even the vanilla FCAM model with softmax achieves fairly high FT (of about 80%). FCAM with other variants designed to increase sparsity, do have an effect in terms of lower NNZ and entropy values, but the effect in terms of the FT numbers is varied. The sparsemax and spherical softmax activation functions show a more than 10% improvement in FT while the entropy regularisation methods do not register any improvement.

Some of the attention mechanism variants studied, e.g. the sparsemax Martins and Astudillo (2016), have been shown to increase performance over the vanilla attention mechanism on large data sets. This improvement has been attributed to increased interpretability,

i.e. the focus network gives higher score to the 'true' foreground segment. Our experiments demonstrate that the sparsemax and spherical softmax activation functions can improve interpretability even in situations where the accuracy improvement is not significant.

One noticeable (if small) improvement in the interpretability (FT) numbers in Table 2 is achieved by using the last hidden layer of the focus network instead of the input in the attended data point $\widetilde{\mathbf{x}}$ – especially for the vanilla softmax and entropy regularisation methods. This effect is more pronounced in the low-dimensional synthetic dataset (See Table **??** in the Appendix). We hypothesize that in datasets where the patches corresponding to background and foreground are similar, there is a significant advantage to aggregating the final layer of the focus network over aggregating the input directly.

The hard attention paradigm would seem to have a natural advantage in terms of interpretability as the input fed to the classification network is always one of the patches. However, the improvement of the FT values in Table 2 for hard attention comes at a cost of lower accuracy. The hard attention algorithm performs much worse in the synthetic dataset results (Table **??** in appendix) and hence is not a good candidate for most practical applications. In addition, the training of hard attention models is significantly more complex computationally.

More details about the experiments are discussed in the appendix.

7. Conclusion

In this paper we present a type of classification problem that is suitable for the analysis of attention models, and enables an objective way to measure an aspect of interpretability in attention models. We analysed various error modes that can cause an accurate attention model to be non-interpretable. We then performed a benchmark empirical study of the interpretability of attention models learnt using various algorithms, including those that purportedly improve performance and interpretability via sparsity encouraging modifications.

Acknowledgments

LNP, RV, and HGR thank the support of the Robert Bosch Centre for Data Science and Artificial Intelligence at IIT Madras. LNP acknowledges the support of Samsung IITM PRAVARTAK Fellowship.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. End-to-end continuous speech recognition using attention-based recurrent nn: First results. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

Alexandre de Brébisson and Pascal Vincent. An exploration of softmax alternatives belonging to the spherical loss family. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.05042.

Yuntian Deng, Yoon Kim, Justin Chiu, Demi Guo, and Alexander M. Rush. Latent alignment and variational attention. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 9735–9747, Red Hook, NY, USA, 2018. Curran Associates Inc.

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)*, pages 4171–4186, Minneapolis, Minnesota, jun 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the $l1-$ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 272–279, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390191. URL https://doi.org/10.1145/1390156.1390191.

Alex Graves. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850, 2013. URL http://arxiv.org/abs/1308.0850.

Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2127–2136. PMLR, 10–15 Jul 2018. URL http://proceedings.mlr.press/v80/ilse18a.html.

Sarthak Jain and Byron C. Wallace. Attention is not explanation. 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)*, pages 3543–3556, Minneapolis, Minnesota, jun 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1357. URL https://www.aclweb.org/anthology/N19-1357.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.

Anirban Laha, Saneem A. Chemmengath, Priyanka Agrawal, Mitesh M. Khapra, Karthik Sankaranarayanan, and Harish G. Ramaswamy. On controllable sparse alternatives to softmax. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 6423–6433, Red Hook, NY, USA, 2018. Curran Associates Inc.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. URL http://arxiv.org/abs/1405.0312.

Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. MIT Press, 1998. URL https://proceedings.neurips.cc/paper/1997/file/82965d4ed8150294d4330ace00821d77-Paper.pdf.

Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1614–1623, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/martins16.html.

Volodymyr Mnih, Nicolas Heess, Alex Graves, and koray kavukcuoglu. Recurrent models of visual attention. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, pages 2204–2212. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/09c6c3783b4a70054da74f2538ed47c6-Paper.pdf.

Y. Ollivier. Riemannian metrics for neural networks i: feedforward networks. *arXiv: Neural and Evolutionary Computing*, 2013.

Sivan Sabato and Naftali Tishby. Multi-instance learning with any hypothesis class. *J. Mach. Learn. Res.*, 13:2999–3039, 2012.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017. doi: 10.1109/ICCV.2017.74.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=HJ0UKP9ge.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Pascal Vincent, Alexandre de Brébisson, and Xavier Bouthillier. Efficient exact gradient update for training deep networks with very large sparse targets. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1108–1116, Cambridge, MA, USA, 2015. MIT Press.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615, Austin, Texas, nov 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1058. URL https://www.aclweb.org/anthology/D16-1058.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China, nov 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1002. URL https://www.aclweb.org/anthology/D19-1002.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2015. URL http://proceedings.mlr.press/v37/xuc15.html.

Jiajun Zhang, Yang Zhao, Haoran Li, and Chengqing Zong. Attention with sparsity regularization for neural machine translation and summarization. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 27(3):507–518, mar 2019. ISSN 2329-9290. doi: 10.1109/TASLP.2018.2883740. URL https://doi.org/10.1109/TASLP.2018.2883740.