CONDITIONING SEQUENCE-TO-SEQUENCE MODELS WITH LEARNED ACTIVATION FUNCTIONS

Alberto Gil C. P. Ramos[1,*], **Abhinav Mehrotra**[1,*], **Nicholas D. Lane**[1,2], **Sourav Bhattacharya**[1]
[1]Samsung AI Centre, Cambridge, UK [2]University of Cambridge, UK
{a.gilramos,a.mehrotra1,nic.lane,sourav.b1}@samsung.com

A APPENDIX

A.1 BASELINE MODEL FOR PSE

Figure 5 presents a pictorial representation of the PSE models introduced in §4.2.

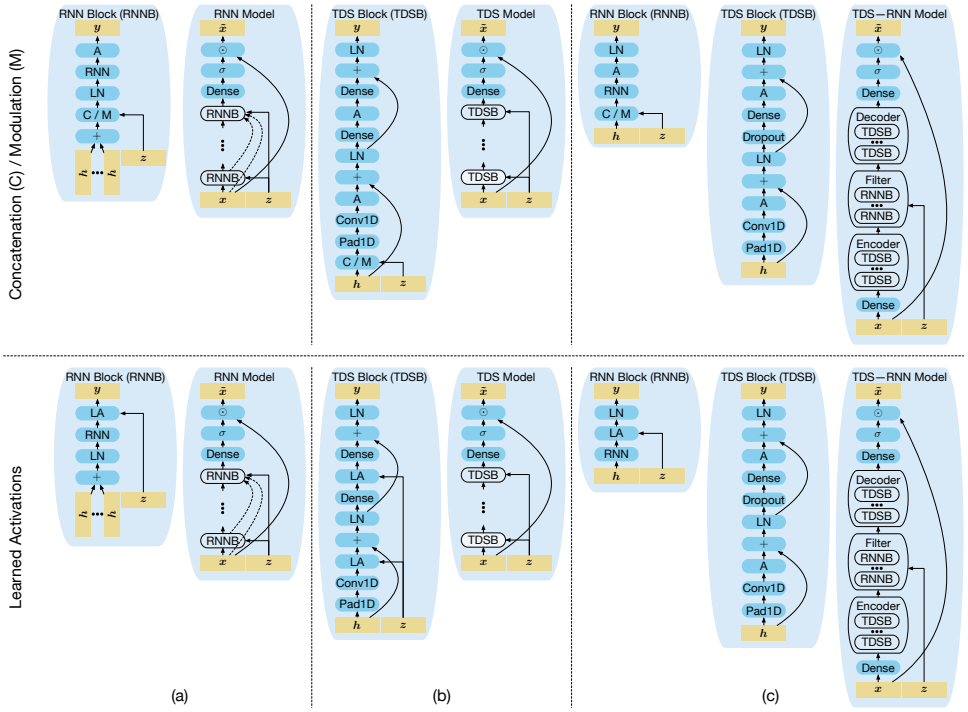


Figure 5: A pictorial representation of the RNN (a), TDS (b) and TDS–RNN (c) models used in the evaluation of conditioning based on concatenation (top, C), modulation (top, M) and based on the proposed learned activations (LAs) (bottom).

A.2 REGULARIZERS

We also investigate the use of regularizers during training for learning activation functions, in order to boost deployment and conditioning efficiency. We consider two regularizers: entropy-based and t-SNE (Van der Maaten & Hinton, 2008). Through entropy regularization we encourage learned activations to approximately select one element of a family of predefined activations (with applications to efficiency). Whereas, with t-SNE regularization we promote that similar/dissimilar conditioning vectors yield similar/dissimilar learned activations, thereby instilling the geometry of the speaker embedding space in the conditioning mechanism (for enhanced conditioning quality).

[*]Equal contribution

A.2.1 APPROXIMATELY ONE-HOT ASSIGNMENTS AND GEOMETRY

We explore mainly two regularization techniques to promote certain qualitative properties in the learned activations during training. The first technique encourages the learned activations to approximately select one element of $\{\mathrm{A}_i : \mathbb{R} \to \mathbb{R}\}_{i=1}^{a}$, based on the conditioning vector. Different non-linear functions require different amount of computational resources, thus biasing the learned activations towards a single known form can improve efficiently, in comparison to their weighted combinations. The second technique guides training so that similar/dissimilar conditioning vectors yield similar/dissimilar learned activations, thereby preserving the geometry of the speaker embedding space in the conditioning mechanism. Considering a particular use-case of PSE, there might be some variation in the conditioning vectors (i.e., speech samples collected during enrolment). This turns into a challenge of selecting or aggregating these vectors to properly condition the network. The use of our second technique for regularization during training could reduce the impact of such variations in conditioning vectors at deployment phase by increasing the stability of the network against subtle variations in the conditioning vectors.

Learning approximately one-hot assignments Sparse assignments can be encouraged by adding known regularizers, such as entroypy or ℓ_1 to the loss during training, scaled by a hyper-parameter α, i.e., $\alpha \mathrm{H}(\boldsymbol{s})$ or $\alpha \|\boldsymbol{s}\|_1$. Alternatively, sparsity can be promoted by considering a temperature parameter $t \leq 1$. For example, a value lower than one makes the model more confident, whereas higher than one makes it less so. Thus sparse activation can be learned as: $\boldsymbol{s} \coloneqq \mathrm{softmax}_{\mathrm{rowwise}}(t^{-1}(\boldsymbol{z} \times \boldsymbol{w} + \boldsymbol{b}))$.

Approximately preserving conditioning vectors geometry Learned activations are built based on conditioning vectors, and we may expect these functions to have a discriminative property for the conditioning vectors they are based on. In other words, clusters in $\boldsymbol{z}_j$ space would be correlated to clusters in the space of learnt conditioned activations weights $\boldsymbol{s}_j$ for similar/dissimilar conditioning vectors. However, strong correlation may not happen automatically due to: i) the dimension of the conditioning vectors is often much larger than the number of basic activation functions a, i.e., $d \gg a$, ii) the manner in which $\boldsymbol{z}_j$ is used to produce $\boldsymbol{s}_j$ may be affected by the fact that $\mathrm{softmax}$ is invariant to translation, i.e., $\mathrm{softmax}(\boldsymbol{v} + c) = \mathrm{softmax}(\boldsymbol{v})$, and iii) the similarity/dissimilarity of the basic activations $\{\mathrm{A}_i : \mathbb{R} \to \mathbb{R}\}_{i=1}^{a}$. High correlation can be promoted though by geometry aware high- to low-dimensional embeddings like t-SNE (Van der Maaten & Hinton, 2008). Although typically used for data visualization, we instead leverage the t-SNE objective function as a regularizer in our loss function. Specifically, we take t-SNE probabilities $\boldsymbol{p}, \boldsymbol{q} \in [0, 1]^{b \times b}$ defined by $\boldsymbol{p}_{i,i} \coloneqq \boldsymbol{q}_{i,i} \coloneqq 0$ and for $i \neq j$:

$$p_{j|i} \coloneqq \frac{e^{-\|\boldsymbol{z}_i - \boldsymbol{z}_j\|^2 / 2\sigma^2}}{\sum_{k \neq i} e^{-\|\boldsymbol{z}_i - \boldsymbol{z}_k\|^2 / 2\sigma^2}}, \quad \boldsymbol{p}_{i,j} \coloneqq \frac{p_{j|i} + p_{i|j}}{2b}, \quad \boldsymbol{q}_{i,j} \coloneqq \frac{(1 + \|\boldsymbol{s}_i - \boldsymbol{s}_j\|^2)^{-1}}{\sum_k \sum_{l \neq k} (1 + \|\boldsymbol{s}_k - \boldsymbol{s}_l\|^2)^{-1}},$$

and add $\beta \mathrm{KL}\left(\boldsymbol{p} \parallel \boldsymbol{q}\right)$ to our loss function, where β is a hyper-parameter and KL denotes the Kullback–Leibler divergence. The minimization of the KL regularizer may be useful in applications, where conditioning vectors are naturally clustered such as in audio applications involving speaker embedding vectors, such as PSE, TTS, and speaker-dependent ASR. Also in applications with discrete latent vectors (Oord et al., 2017), where the preservation of the geometry of the conditioning vectors in the learned activations may offer superior results or better insights into the modelling internals.

A.2.2 IMPACT OF REGULARIZERS

We now investigate the impact of using regularizers for training PSE models. We trained all models with two regularizes (c.f §A.2), which are used individually to analyze their unique benefits (as discussed earlier). Tables 5–6 present the performance of all models by using each regularizer with models trained on two datasets. To show the effectiveness of the regularizers, we compare these results against those in Tables 1–2 for models learned without regularizers.

The results clearly demonstrate the improvement in performance of RNN and TDS models when trained with either of the regularizers. The performance gain is consistent across both datasets, but there is a trade off with the impact on clean speech. Moreover, we observe that entropy regularizer is outperformed by t-SNE regularizer. This could be due to the fact that t-SNE regularizer is less restrictive allowing a wider range of combinations of the basic activations. Whereas, entropy regularizer promotes the selection of only one of the basic activations.

Interestingly, the TDS-RNN models, which performed best without regularizers, get worse performance with regularizers. This could be due to limited number of filter layers (i.e., only 2, compared with 8 and 42 for RNN and TDS models respectively) present in the model architecture. Given that LA was given only 11 basic activation to learn from, there could be merely 121 (i.e., 11^2) unique separations learned by LA. This makes it difficult for these models to successfully differentiate between all unique conditioning vectors which might contain way more separations.

Note, we considered as hyper-parameters only the values in $\alpha, \beta \in \{1\mathrm{e}{-5}, 1\mathrm{e}{-4}, 1\mathrm{e}{-3}, 1\mathrm{e}{-2}\}$, $\sigma \in \{1\}$ and used the same learning rates as for training models without regularizers. Due to limited computational budget, we could not perform an exhaustive search on these hyper-parameters.

A.2.3 CORRELATION ANALYSIS

To understand the manner in which the models leverage the conditioning vectors together with inputs to predict outputs, we perform a correlation analysis between distance matrices of softmax values (of LA) and of conditioning vectors. For a random selection of embedding vectors $\{\boldsymbol{z}_j\}_{j=1}^{100}$, we computed the weights $\boldsymbol{s}_j^{(l)}$ with $j \in \{1, \ldots, 100\}$ of the l LAs of a model. Now, we compute the distance matrices for $\boldsymbol{z}_j$ and $\boldsymbol{s}_j^{(l)}$, which we refer to as $\boldsymbol{Z}$ and $\boldsymbol{S}^{(l)}$. Here, the distance matrix is computed by taking the pair-wise cosine distances. To get insights about how LAs perform conditioning, we compute the Spearman correlation between $\mathrm{vec}(\boldsymbol{Z})$ and $\mathrm{vec}(\boldsymbol{S}^{(l)})$. This correlation is computed for all LAs (i.e., 8, 42 and 2 for RNN, TDS, TDS–RNN models). This allows to better measure the impact of the conditioning vector across the depth (i.e., position of the LA) in the network.

In Figure 6 (bottom rows of each sub-figure) we present the correlation values for each LA in all three models. We observe there is a large variability in correlation values as the number of layers increases. We do not see very strong correlation with a specific LA, which might be due to the fact that conditioning is performed at different stages.

In addition, we also present (shown in top rows of each sub-figure of Figure 6) the distance matrices $\boldsymbol{S}^{(l)}$ (for l which has the highest correlation), as well as $\boldsymbol{Z}$ (in the last column). The overall structure of $\boldsymbol{Z}$ is approximately block diagonal with three blocks, is not well preserved when learning without regularizers given the blocks collapse to two, but is well preserved when learning with regularizers. This is especially significant given that conditioning vectors are high dimensional (in this example 256) but the weight vectors in LAs are low dimensional (in this case 11). We also observe that the models with less number of LAs preserve much similar structure between $\boldsymbol{S}^{(l)}$ and $\boldsymbol{Z}$. Given the networks have similar performance, this indicates that as the number of LAs decrease, the networks enforce higher degree of conditioning per LA.

A.3 IMPACT OF ENROLLMENT DATA ON AMBIENT NOISE

Tables 5 and 6 show the performance of various PSE models based on two seconds of enrollment data in terms of babble and ambient noise. Although babble noise, i.e., other people talking, cannot be expected to be suppressed without knowledge of the owner's voice profile, clean and ambient noise on the other hand does not in principle require such enrollment data. To investigate the performance of the various PSE models in the absence of enrolment data we have replaced it with vectors of random, zeros and ones, and measured the quality of PSE on ambient noise in Tables 7–8, which demonstrate that all architectures work well (compared to concatenation based conditioning approach) even without enrolment data for clean and ambient noise when using ones instead of conditioning vectors. We also see that for the best architecture (TDS–RNN) that is the case not just with replacing by ones but also by random and zeros.

REFERENCES

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In NeurIPS, 2017.

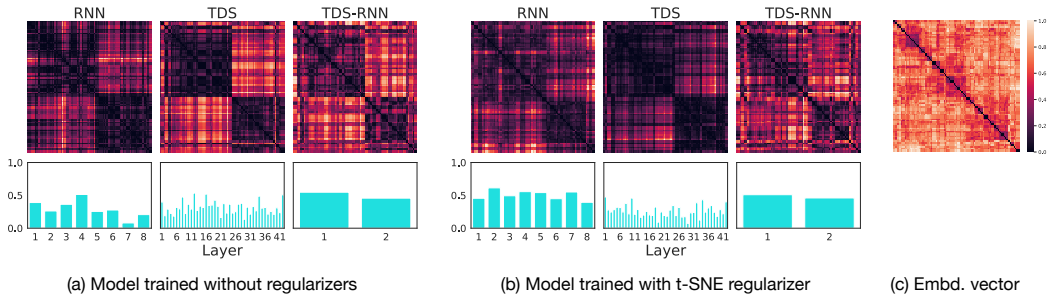
Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. Journal of machine learning research, 9, 2008.

Table 5: SDRi and WER results on LibriSpeech (English) when using learned activations (LA) trained with entropy (α) and t-SNE ($\{\beta, \sigma\}$) regularizers.

Model	Regularizers		SDRi		WER		
	Entropy	t-SNE	Babble	Ambient	Clean	Babble	Ambient
RNN	1e−4	{0, 1}	8.53	11.90	8.5	24.5	15.4
TDS	1e−3	{0, 1}	7.27	11.55	8.3	33.0	16.0
TDS–RNN	1e−5	{0, 1}	7.95	12.14	8.2	28.8	15.2
RNN	0	{1e−4, 1}	8.46	12.08	8.4	24.3	15.0
TDS	0	{1e−5, 1}	7.42	11.42	8.5	33.0	16.1
TDS–RNN	0	{1e−5, 1}	8.51	12.50	8.2	25.4	14.9

Table 6: SDRi and WER results on VoxForge (Spanish) when using learned activations (LA) trained with entropy (α) and t-SNE ($\{\beta, \sigma\}$) regularizers.

Model	Regularizers		SDRi		WER		
	Entropy	t-SNE	Babble	Ambient	Clean	Babble	Ambient
RNN	1e−5	{0, 1}	7.08	9.74	1.6	18.6	5.9
TDS	1e−5	{0, 1}	5.85	8.61	2.1	27.0	7.1
TDS–RNN	1e−5	{0, 1}	5.51	9.34	1.3	33.3	6.8
RNN	0	{1e−5, 1}	7.36	10.12	2.6	16.9	5.9
TDS	0	{1e−2, 1}	5.68	8.62	1.8	29.0	7.1
TDS–RNN	0	{1e−5, 1}	6.33	9.92	1.4	25.2	6.5



(a) Model trained without regularizers (b) Model trained with t-SNE regularizer (c) Embd. vector

Figure 6: Correlation analysis of conditioning vectors with softmax values of LA for models trained on Librispeech. In sub-figure (a) and (b), the bottom row shows the overall correlation of all LAs in model with conditioning vectors, and the top row depicts the distance matrix corresponding to LA with the highest overall correlation for each model as well as for conditioning vectors (fig c).

Table 7: Results for models trained on Librispeech (English) data with proposed LA and baseline concatenation and evaluated with random, zeros and ones as speaker embedding vectors.

Model	LA	SE	SDRi		WER		
			Babble	Ambient	Clean	Babble	Ambient
Identity	-	-	-	-	8.2	89.4	17.3
RNN	LA	Random	3.08	9.24	10.6	48.8	15.7
RNN	LA	Zeros	0.47	8.41	25.9	60.2	21.8
RNN	LA	Ones	4.61	11.64	9.7	44.4	15.6
RNN	Concat.	Random	1.17	6.32	28.2	55.2	20.0
RNN	Concat.	Zeros	3.70	10.76	20.2	43.8	17.3
RNN	Concat.	Ones	2.20	11.44	10.8	51.1	15.8
TDS	LA	Random	-3.48	4.36	11.3	79.6	19.7
TDS	LA	Zeros	2.65	4.58	14.4	85.5	25.5
TDS	LA	Ones	-0.59	10.59	8.4	74.0	16.1
TDS	Concat.	Random	-2.90	5.63	32.7	74.6	24.1
TDS	Concat.	Zeros	-0.68	8.38	30.9	68.5	21.7
TDS	Concat.	Ones	-1.13	9.06	24.8	67.4	19.5
TDS–RNN	LA	Random	1.99	12.23	14.0	52.2	14.4
TDS–RNN	LA	Zeros	1.91	12.19	17.0	52.8	14.5
TDS–RNN	LA	Ones	3.90	12.33	9.5	44.8	14.4
TDS–RNN	Concat.	Random	1.79	7.42	58.2	46.1	21.0
TDS–RNN	Concat.	Zeros	7.33	11.84	25.4	25.6	15.2
TDS–RNN	Concat.	Ones	3.66	11.75	23.1	39.8	15.4

Table 8: Results for models trained on VoxForge (Spanish) data with proposed LA and baseline concatenation and evaluated with random, zeros and ones as speaker embedding vectors.

Model	LA	SE	SDRi		WER		
			Babble	Ambient	Clean	Babble	Ambient
Identity	-	-	-	-	1.2	89.1	9.2
RNN	LA	Random	-3.23	-3.00	76.6	77.8	52.7
RNN	LA	Zeros	-1.60	2.45	48.7	64.9	33.3
RNN	LA	Ones	-3.53	1.37	61.5	69.7	37.7
RNN	Concat.	Random	-1.77	-0.44	56.9	73.6	41.0
RNN	Concat.	Zeros	-1.26	2.34	42.9	63.4	30.7
RNN	Concat.	Ones	-2.09	1.02	57.9	67.8	39.2
TDS	LA	Random	-3.91	-3.21	49.8	85.8	40.3
TDS	LA	Zeros	-3.91	-3.21	49.8	85.8	40.3
TDS	LA	Ones	-3.84	0.18	33.4	78.7	30.4
TDS	Concat.	Random	-3.80	-2.48	34.1	84.0	33.6
TDS	Concat.	Zeros	-4.05	-1.06	39.9	80.2	39.5
TDS	Concat.	Ones	-4.06	-1.51	47.2	78.1	42.2
TDS–RNN	LA	Random	-1.07	8.37	13.0	56.8	10.3
TDS–RNN	LA	Zeros	-1.13	8.26	14.0	56.9	10.9
TDS–RNN	LA	Ones	-0.85	8.58	10.7	56.4	9.4
TDS–RNN	Concat.	Random	-0.10	0.88	86.6	56.3	39.9
TDS–RNN	Concat.	Zeros	1.59	6.09	56.2	42.7	19.9
TDS–RNN	Concat.	Ones	0.47	3.75	66.3	51.1	31.7