LEARNING HARD ALIGNMENTS WITH VARIATIONAL INFERENCE

*Dieterich Lawson**, *Chung-Cheng Chiu**, *George Tucker**, *Colin Raffel, Kevin Swersky, Navdeep Jaitly*

Google Brain
{dieterichl,chungchengc,gjt,craffel,kswersky,ndjaitly}@google.com

ABSTRACT

There has recently been significant interest in hard attention models for tasks such as object recognition, visual captioning and speech recognition. Hard attention offers benefits over soft attention such as decreased computational cost, but training hard attention models can be difficult because of the discrete latent variables they introduce. Previous work used REINFORCE to approach these issues, however, it suffers from high-variance gradient estimates, resulting in slow convergence. In this paper, we tackle the problem of learning hard attention for a sequential task using variational inference methods, specifically the recently introduced Variational Inference for Monte Carlo Objectives (VIMCO) and Neural Variational Inference (NVIL). Furthermore, we propose a novel baseline that adapts VIMCO to this setting. We demonstrate our method on a phoneme recognition task in clean and noisy environments and show that our method outperforms REINFORCE, with the difference being greater for a more complicated task.

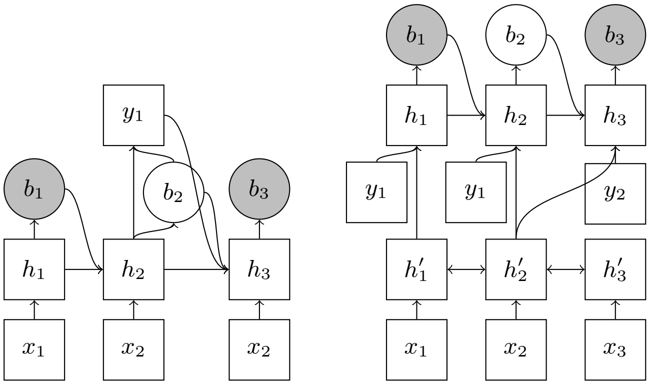
Index Terms— Variational inference, online, sequence-to-sequence, end-to-end, LAS

1. INTRODUCTION

Attention models have gained widespread traction from their successful use in tasks such as object recognition, machine translation, speech recognition where they are used to integrate information from different parts of the input before producing outputs. Soft attention does this by weighting and combining all input elements into a context vector while hard attention selects specific inputs and discards others, leading to computational gains and greater interpretability. While soft attention models are differentiable end-to-end and thus straightforward to train, hard attention models introduce discrete latent variables that often require reinforcement learning style approaches.

Classic reinforcement learning methods such as REINFORCE [1] and Q-learning [2] have been used to train hard attention models, but these methods can provide high-variance gradient estimates, making training slow and providing inferior solutions. An alternative to reinforcement learning is variational inference, which trains a second model, called the approximate posterior, to be close to the true posterior over the latent variables. The approximate posterior uses information about both the input and its labels to produce settings of the latent variables used to train the original model. This can provide lower-variance gradient estimates and better solutions.

In this paper, we leverage recent developments in variational inference such as NVIL [3] and VIMCO [4] to fit hard attention models in a sequential setting. We specialize these methods to sequences and develop a model for the approximate posterior. In response to issues applying variational inference techniques to long sequences, we develop new variance control methods. Finally, we show experimentally that our approach improves performance and substantially reduces training time for speech recognition on the TIMIT dataset as well as a noisy, multi-speaker version of TIMIT that we call Multi-TIMIT.



(a) Model, $p(y, b|x)$

(b) Approximate Posterior, $q(b|x, y)$

Fig. 1. A diagram of our models. bs denote the Bernoulli emission decision variables, xs are inputs, ys are targets, and hs and h's are the hidden states of the recurrent neural networks (RNNs) that parameterize the conditional distributions of the models. Square nodes are deterministic, round nodes are stochastic. A shaded b_i indicates that the model chose to consume an input and not emit an output while an unshaded b_i mean that the model chose to produce an output and not consume an input. For example, in (a) note that b_1 is shaded, so the model did not produce an output on timestep 1 and instead consumes the input x_2 on the next timestep. b_2 is unshaded, so on the second timestep the model produced output y_1.

2. METHODS

2.1. Model

In this paper, we use the online sequence-to-sequence model described in [5] to demonstrate our methods. We model $p(y, b|x)$ where $y = y_1, \dots, y_n$ is a sequence of observed target tokens and $x = x_1, \dots, x_m$ is a sequence of observed inputs. The Bernoulli latent variables $b = b_1, \dots, b_{m+n}$ define when the model outputs tokens, i.e. $b_t = 1$ implies the model emitted a token at timestep t, and $b_t = 0$ implies the model did not emit a token at timestep t. If $b_t = 1$, the model is forced to dwell on the same input at the next time step, i.e. the observation fed in at timestep t is fed in again at timestep $t + 1$ when $b_t = 1$. Let n be the number of target tokens, m the number of inputs, and $T = m + n$ the number of steps the model

*Equal contribution.

is run for. Our model assumes $p(y, b|x)$ factorizes as

$$p(y, b|x) = \prod_{t=1}^{T} p(y_{O(t)}|b_{1:t}, x_{1:I(t)}, y_{1:O(t-1)})^{b_t} \times \\ p(b_t|b_{1:t-1}, x_{1:I(t)}, y_{1:O(t-1)}) \quad (1)$$

where $O(t) = \sum_{i=1}^{t} b_i$ is the position in the output at time t and $I(t) = 1 + \sum_{i=1}^{t-1}(1 - b_i)$ is the input position at time t. Intuitively, this expression is the product over time of the probability assigned to the current ground truth given that the model emitted, multiplied by the probability that the model emitted. When $b_t = 0$ the model did not emit at time t, so there is no probability assigned to the ground truth on that timestep. For brevity, we will use y_t to implicitly mean $y_{O(t)}$ (i.e., the target at step t). Similarly, we will refer to $x_{I(t)}$ as x_t and similarly for ranges over time for these variables.

2.2. Learning

To fit the model (1) with maximum likelihood we are concerned with maximizing the probability of the observed variables y. However, (1) is written in terms of the unobserved latents, b, so we must marginalize over them. The marginal likelihood is intractable, so the authors of [5] maximize a lower bound

$$\mathbf{E}_b\left[\sum_{t=1}^{T} b_t \log p(y_t|s_t, b_t)\right] = \mathbf{E}_b\left[\log \prod_{t=1}^{T} p(y_t|s_t, b_t)^{b_t}\right] \\ \leq \log p(y|x),$$

where $s_t = \{b_{1:t-1}, x_{1:t}, y_{1:t-1}\}$ is the state at time t and the expectations are over $p(b_t|s_t)$. Maximizing this lower bound will hopefully increase the likelihood of the observed data. Differentiating the lower bound with respect to the model parameters gives

$$\mathbf{E}_b\left[\nabla \sum_{t=1}^{T} b_t \log p(y_t|s_t, b_t)\right] + \sum_{t=1}^{T} \mathbf{E}_b\left[R_t \nabla \log p(b_t|s_t)\right], \quad (2)$$

where $R_t = \sum_{t' \geq t}^{T} \log p(y_{t'}|s_{t'}, b_{t'})$ is known as the *return* at timestep t and is the log probability the model assigns to observed data for a given series of emission decisions. To reduce variance in the Monte Carlo estimate of the gradient, [5] subtracts a learned baseline $c(b_{1:t-1}, x_{1:T}, y_{1:T})$ from the return, which does not change the expectation as long as it is independent of b_t [1].

Performing stochastic gradient ascent with this gradient estimator is the standard REINFORCE algorithm where the reward is the log-likelihood. Unfortunately, this requires sampling b_t from $p(b_t|s_t)$ during training, which can lead to gradient estimates with high variance when settings of b that assign high likelihood to y are rare [4]. Variational inference is a family of techniques that use importance sampling to instead sample b from a different model, called the approximate posterior or q, which approximates the true posterior over b, $p(b|x, y)$. The approximate posterior (see Figure 1b) factorizes as

$$q(b|x, y) = \prod_{t=1}^{T} q(b_t|b_{1:t-1}, x_{1:T}, y_{1:T}). \quad (3)$$

The approximate posterior has access to all past and future x and y, as well as past b, and leverages this information to assign high probability to b that produce large values of $p(y|b, x)$. Intuitively, in speech recognition, knowing the token the model must emit is helpful in deciding when to emit.

Using q and an importance sampling identity we obtain a lower bound on the log-likelihood

$$\log p(y|x) = \log \mathbf{E}_{b\sim q}\left[\frac{p(y, b|x)}{q(b|x, y)}\right] \geq \mathbf{E}_{b\sim q}\left[\log \frac{p(y, b|x)}{q(b|x, y)}\right] \quad (4)$$

where we can simultaneously optimize q and the parameters of the model to improve the lower bound. Optimizing this bound via stochastic gradient ascent can be thought of as training p with maximum likelihood to reproduce bs sampled from q. q is then updated with REINFORCE-style gradients where the reward is the log-probability p assigns to y given b, similar to (2), see [4] for details. Setting $q(b|x, y) = \prod_t p(b_t|s_t)$ recovers the REINFORCE objective.

2.2.1. *Multi-sample Objectives*

Both the REINFORCE and the variational inference objectives admit multi-sample versions that give tighter bounds on the log-likelihood [6]. In particular, the multi-sample variational lower bound is

$$\mathcal{L} = \mathbf{E}_{b^{(1:k)}\sim q}\left[\log\left(\frac{1}{k}\sum_{i=1}^{k} \frac{p(y, b^{(i)}|x)}{q(b^{(i)}|x, y)}\right)\right] \quad (5)$$

where k is the number of samples and $b^{(i)}$ denotes the ith sample of the latent variables. Setting $q(b|x, y) = \prod_t p(b_t|s_t)$ recovers the multi-sample analogue to REINFORCE.

The gradient of (5) takes a similar form to (2), with one low-variance term and one REINFORCE-style term with high variance, for details see [4]. Similarly to the REINFORCE objective, we can use a baseline $c(b_{1:t-1}^{(i)}, x_{1:T}, y_{1:T}, b_{1:T}^{(-i)})$ to reduce the variance of the gradient as long as it does not depend on $b_t^{(i)}$. Notably, the baseline for trajectory i is allowed to depend on all timesteps of other trajectories, i.e. $b_{1:T}^{(-i)}$.

2.3. Variance Reduction

Training these models is challenging due to high variance gradient estimates. We can reduce the variance of the estimators by using information from multiple trajectories to construct baselines. In particular, for REINFORCE, we can write the gradient update as

$$\mathbf{E}_{b^{(i)}}\left[\sum_{t=1}^{T}\left(R_t - c(s_{t-1}^{(i)}, \{R_{1:T}^{(j)}\}_{j\neq i})\right)\nabla \log p(b_t^{(i)}|s_{t-1}^{(i)})\right],$$

where c is a baseline for sample i that is a function of the ith trajectory's state up to time $t - 1$ as well as the returns produced by all other trajectories. The goal is to pick a c that is a good estimate of the return, and a straightforward choice is the average return from the other samples

$$c = \frac{1}{k-1}\sum_{j\neq i} R_t^{(j)}.$$

This ignores the fact that $s_t^{(i)} \neq s_t^{(j)}$, which can make this standard baseline unusable. For example, in our setting different trajectories may have emitted different numbers of tokens on a given timestep, resulting in substantial differences in return between trajectories that do not indicate the relative merit of those trajectories. Ideally, we would average over multiple trajectories starting from $s_t^{(i)}$, but this is computationally expensive. In [4] the authors propose the following baseline which adds a residual term to address this. Let

$r_t = \log p(y_t|s_t, b_t)$ be the instantaneous reward at timestep t, then the baseline at timestep t can be written

$$c = \frac{1}{k-1}\sum_{j\neq i} R_t^{(j)} + \frac{1}{k-1}\sum_{j\neq i}\sum_{t'<t} r_{t'}^{(j)} - r_{t'}^{(i)}. \tag{6}$$

This baseline results in a learning signal that is the same across all timesteps, potentially increasing variance as all decisions in a trajectory are rewarded or punished together. We will call this the *leave-one-out* (LOO) baseline because the baseline for a given sample is constructed using an average of the return of the other $k-1$ samples. Note that VIMCO optimizes the multisample variational lower bound in equation (5) with the leave-one-out baseline, and NVIL optimizes the single sample variational lower bound in equation (4) with a baseline that can be learned or computed from averages [3].

As the return strongly depends on the number of emitted tokens at time t, we can instead average the return of the other samples from when they have emitted the same number of tokens as sample i. Let $e_t^{(j)} = \min_{t'} O^{(j)}(t') \geq O^{(i)}(t)$ be the first timestep when sample j has emitted the same number of tokens as sample i at timestep t, then

$$c = \frac{1}{k-1}\sum_{j\neq i}\sum_{t'>e_{t-1}^{(j)}}^{T} r_{t'}^{(j)}. \tag{7}$$

We call this new baseline the *temporal leave-one-out* baseline because it takes into account the temporal reward structure of our setting. This baseline can be combined with the parametric baseline, and is applicable to both variational inference and REINFORCE objectives in single- and multi-sample settings. We explore the performance of these baselines empirically in the experiments section.

3. RELATED WORK

In this section we first highlight the relationship between our model and other models for attention. Tang et. al. [7] proposed visual attention within the context of generative models, while Mnih et. al. [8] proposed using recurrent models of visual attention for discriminative tasks. Subsequently, visual attention was used in an image captioning model [9]. These forms of attention use discrete variables for attention location. Recently, 'soft-attention' models were proposed for neural machine translation and speech recognition [10, 11]. Unlike the earlier mentioned, hard-attention models, these models pay attention to the entire input and compute features by blending spatial features with an attention vector that is normalized over the entire input. Our paper is most similar to the hard attention models in that features at discrete locations are used to compute predictions. However it is different from the above models in the training method: While the hard attention models use REINFORCE for training, we follow variational techniques. We are also different from the above models in the specific application – attention in our models is over temporal locations only, rather than visual and temporal locations. As a result, we additionally propose the temporal leave-one-out baseline.

Because the attention model we use is hard-attention, the model we use has parallels to prior work on online sequence-to-sequence models [12, 5]. The neural transducer model [12] can use either hard attention, or a combination of hard attention with local soft attention. However it explicitly splits the input sequences into chunks, and it is trained with an approximate maximum likelihood procedure that is similar to a policy search. The model of Luo et. al. [5] is most similar to our model. Both models use the same architecture; however, while they use REINFORCE for training, we explore VIMCO for training the attention model. We also propose the novel temporal LOO baseline. A similar model with REINFORCE has also been used for training an online translation model [13] and for training Neural Turing Machines [14]. Our work would be equally valid for these domains, which we leave for future work.

There has also been work using reweighted wake sleep to train sequential models. In [15], Ba et. al. optimize a variational lower bound with the prior instead of using a variational posterior. In this work, we refer to this as REINFORCE to distinguish it from variational inference with an inference network. In [16] the authors revisit this topic, using reweighted wake sleep to train similar models. Their algorithm makes use of an inference network but does not optimize a variational lower bound. Instead they optimize separate objectives for the model and the inference network that produce a biased estimate of the gradient of the log marginal likelihood.

4. EXPERIMENTS

For our experiments we used the standard TIMIT phoneme recognition task. The TIMIT dataset has 3696 training utterances, 400 validation utterances, and 182 test utterances. The audio waveforms were processed into frames of log mel filterbank spectrograms every 25ms with a stride of 10ms. Each frame had 40 mel frequency channels and one energy channel; deltas and accelerations of the features were append to each frame. As a result each frame was a 123 dimensional input. The targets for each utterance were the sequence of phonemes. We used the 61 phoneme labels provided with TIMIT for training and decoding. To compute the phone error rate (PER) we collapsed the 61 phonemes to 39 as is standard on this task [17].

To model p we used a 2-layer LSTM with 256 units in each layer. For the variational posterior q we first processed the inputs $x_{1:T}$ with a 4-layer bidirectional LSTM and then fed the final layer's hidden state h' into a 2-layer unidirectional LSTM along with the current target y_t and the previous emission decision b_{t-1}. Each layer had 256 units. Note that in this case the approximate posterior does not have access to $y_{t+1:T}$ at timestep t — in practice we found giving q access to y far in the future did not improve performance.

We regularized the models with variational noise [18] and performed a grid search over the values $\{0.075, 0.1, 0.15\}$ for the standard deviation of the noise. We also used L2 regularization and grid searched over the values $\{1\times10^{-5}, 1\times10^{-4}, 1\times10^{-3}\}$ for the weight of the regularization.

4.1. Multi-TIMIT

In addition to TIMIT, we also trained models on a noisy, multispeaker dataset called Multi-TIMIT. In noisy settings an approximate posterior with access to all x and y could disambiguate between possible decodings of the same sequence better than the prior, which only has access to x and past y.

To create Multi-TIMIT we mixed male and female voices from TIMIT. Each utterance in the original TIMIT dataset was paired with an utterance from the opposite gender. The waveform of both utterances was first scaled to lie within the same range, and then the scale of the second utterance was reduced to a smaller volume before mixing the two utterances. We used three different scales for the second utterance: 50%, 25%, and 10%. The new raw utterances were processed in the same manner as the original TIMIT utterances, resulting in a 123 dimensional input per frame. The transcript of the speaker 1 was used as the ground truth transcript for this new utterance. Multi-TIMIT has the same number of train, dev, and test utterances as the original TIMIT, as well as the same target phonemes.

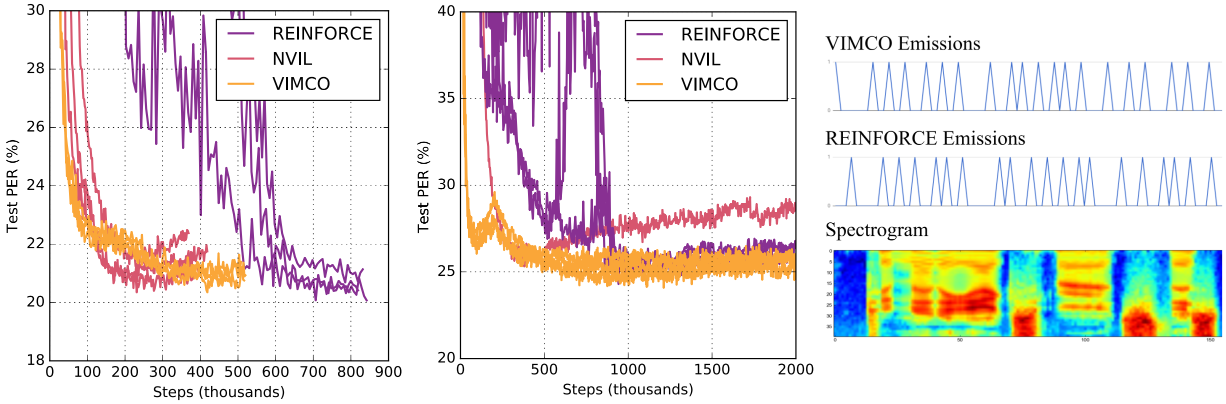


Fig. 2. Test set phoneme error rate (PER) curves for models trained with REINFORCE, NVIL, and VIMCO on the TIMIT dataset (left), the Multi-TIMIT 10% mixing proportion dataset (middle), and sample emission decisions for different methods on a TIMIT utterance (right). We evaluated three independent trials for each method. VIMCO converged more quickly than REINFORCE on both datasets. Furthermore, the performance gap between REINFORCE and VIMCO increases with Multi-TIMIT. We hypothesize that because Multi-TIMIT is a more challenging task, having a strong approximation to the posterior lets the model draw attention to the correct positions. NVIL performed well on TIMIT, but struggled with the more challenging Multi-TIMIT (note that only a single trial performs reasonably).

Table 1. PER results on TIMIT test set for various models. This shows that REINFORCE performs comparably to the variational inference methods and that our novel baselines improve training for REINFORCE. It also shows that our baselines improve performance over [5] which uses the same model with parametric baselines. Each number is the average of three runs. Our methods are above the horizontal line, while methods from the literature are listed below it.

Method	PER
REINFORCE with leave-one-out (LOO) baseline	20.5
NVIL with LOO baseline	21.1
VIMCO with LOO baseline	**20.0**
REINFORCE with temporal LOO baseline	**20.0**
NVIL with temporal LOO baseline	21.4
VIMCO with temporal LOO baseline	**20.0**
Online Alignment RNN (stacked LSTM) [5]	21.5
Neural Transducer with unsupervised alignments [12]	20.8
Online Alignment RNN (grid LSTM) [5]	20.5
Monotonic Alignment Decoder [19]	20.4
Neural Transducer with supervised alignments [12]	19.8
Connectionist Temporal Classification [20]	**19.6**

Table 2. PER results on Multi-TIMIT for various algorithms. It can be seen that for this task VIMCO outperforms REINFORCE, and both VIMCO and REINFORCE outperforms RNN trained with Connectionist Temporal Classification significantly. The benefit of VIMCO increases as the second speaker's volume increases.

Method	Mixing Proportion		
	0.50	0.25	0.1
Connectionist Temporal Classification	43.8	33.3	27.3
RNN Transducer	48.9	32.2	25.7
REINFORCE with LOO baseline	42.9	32.5	25.9
NVIL with LOO baseline	70.1	71.8	55.2
VIMCO with LOO baseline	**41.7**	**30.7**	25.4
REINFORCE with temporal LOO baseline	43.5	31.6	25.6
NVIL with temporal LOO baseline	74.3	71.9	54.9
VIMCO with temporal LOO baseline	**41.7**	30.75	**25.2**

We trained models with the same configuration described above on the 3 different mixing scales, and also trained 2-layer unidirectional LSTM models with Connectionist Temporal Classification for comparison. The results are shown in Table 2.

5. RESULTS

Figure 2 shows training curves for different training methods. The variational methods (VIMCO and NVIL) require many fewer training steps compared to REINFORCE on both datasets. All methods used the same batch size and number of samples, so training steps are comparable. NVIL performs adequately on TIMIT, but struggles with Multi-TIMIT. It can be seen that the gap between REINFORCE and VIMCO increases on Multi-TIMIT (also see table 2).

The right panel of Figure 2 shows that REINFORCE attempts to wait to emit outputs until more information has come in, compared to VIMCO. This is presumably because it requires more information during learning. VIMCO, on the other hand, leverages the variational posterior which can access future y and find the optimal place to emit.

In our experiments the difference between the performance of VIMCO and REINFORCE was larger for the more complicated task of Multi-TIMIT than for TIMIT. This can be explained by considering the samples that the models learn from. In the simpler problem of single speaker TIMIT, Monte-Carlo samples generated by REINFORCE have very high likelihood under $p(b|x)$ – there are only a small number of samples that explain the entire probability mass, and these are sampled easily by a left to right ancestral pass (in time) of the model. These are very similar to the samples generated by the approximate posterior from VIMCO. As a result both methods perform approximately the same. In the case of Multi-TIMIT, however, in the ancestral pass the probabilities for individual emissions are much lower. Thus the likelihood is less 'peaked', and a large diversity of samples is chosen, leading to higher variance and poor learning. VIMCO, on the other hand does not face this problem because it samples from the approximate posterior, which is close to the true posterior and so very peaked around the 'correct' samples of experience.

6. CONCLUSION

We demonstrated how VIMCO can be adapted to sequential hard attention problems, and introduced a new variance-reducing baseline. Our method outperforms other methods of training online sequence-to-sequence models, and the improvements are greater for more difficult problems such as noisy mixed speech. In the future we will apply these techniques to other domains such as visual attention.

7. REFERENCES

[1] Ronald J Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[2] Christopher John Cornish Hellaby Watkins, *Learning from delayed rewards*, Ph.D. thesis, King's College, Cambridge, 1989.

[3] Andriy Mnih and Karol Gregor, "Neural variational inference and learning in belief networks," *CoRR*, vol. abs/1402.0030, 2014.

[4] Andriy Mnih and Danilo Jimenez Rezende, "Variational inference for monte carlo objectives," *CoRR*, vol. abs/1602.06725, 2016.

[5] Yuping Luo, Chung-Cheng Chiu, Navdeep Jaitly, and Ilya Sutskever, "Learning online alignments with continuous rewards policy gradient," *CoRR*, vol. abs/1608.01281, 2016.

[6] Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov, "Importance weighted autoencoders," *CoRR*, vol. abs/1509.00519, 2015.

[7] Yichuan Tang, Nitish Srivastava, and Ruslan R Salakhutdinov, "Learning generative models with visual attention," in *Advances in Neural Information Processing Systems*, 2014, pp. 1808–1816.

[8] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al., "Recurrent models of visual attention," in *Advances in neural information processing systems*, 2014, pp. 2204–2212.

[9] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio, "Show, attend and tell: Neural image caption generation with visual attention.," in *ICML*, 2015, vol. 14, pp. 77–81.

[10] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014.

[11] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., pp. 577–585. Curran Associates, Inc., 2015.

[12] Navdeep Jaitly, David Sussillo, Quoc V. Le, Oriol Vinyals, Ilya Sutskever, and Samy Bengio, "An online sequence-to-sequence model using partial conditioning," *CoRR*, vol. abs/1511.04868, 2015.

[13] Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O. K. Li, "Learning to translate in real-time with neural machine translation," *CoRR*, vol. abs/1610.00388, 2016.

[14] Wojciech Zaremba and Ilya Sutskever, "Reinforcement learning neural turing machines-revised," *arXiv preprint arXiv:1505.00521*, 2015.

[15] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu, "Multiple object recognition with visual attention," *arXiv preprint arXiv:1412.7755*, 2014.

[16] Jimmy Ba, Ruslan R Salakhutdinov, Roger B Grosse, and Brendan J Frey, "Learning wake-sleep recurrent attention models," in *Advances in Neural Information Processing Systems*, 2015, pp. 2593–2601.

[17] K-F Lee and H-W Hon, "Speaker-independent phone recognition using hidden markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, 1989.

[18] Alex Graves, "Practical variational inference for neural networks," in *Advances in Neural Information Processing Systems*, 2011, pp. 2348–2356.

[19] Colin Raffel, Thang Luong, Peter J Liu, Ron J Weiss, and Douglas Eck, "Online and linear-time attention by enforcing monotonic alignments," *arXiv preprint arXiv:1704.00784*, 2017.

[20] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.