Efficient Structured Inference for Stochastic Recurrent Neural Networks

Hao Liu [1] **Lirong He** [2] **Haoli Bai** [1] **Zenglin Xu** [2]

Abstract

Recent advances in sequential data modeling have suggested a class of models that combine recurrent neural networks with state space models. Despite the success, the huge model complexity has brought an important challenge to the corresponding inference methods. This paper introduces an structured inference algorithm to efficiently learn such models, including variants where the emission and transition distributions are modelled by deep neural networks. Our learning algorithm leverages a structured variational approximation parameterized by stochastic models and recurrent neural networks to approximate the posterior distribution. Experimental results on synthetic datasets have demonstrated the promising performance of the proposed method. In addition, our method has significantly outperformed the current state-of-the-art methods on music and speech modeling tasks.

1. Introduction

Sequential data learning is a critical yet challenging research topic in machine learning, which is widely observed in many tasks such as filtering, human activity recognition and segmentation. Considerable research has been devoted to developing probabilistic models for high-dimensional time-series data, such as video and music sequences, motion capture data and text streams. Among them, State Space Models (SSMs) (Roweis & Ghahramani, 1999), such as Hidden Markov Models (HMMs) (Rabiner & Juang, 1986) and Linear Dynamical Systems (LDS) (Kalman, 1963; Krishnan et al., 2015), have been widely studied, but they may be limited in the type of dynamical structures they can model. Recently, some work consider combining neural networks and graphical model (Johnson et al., 2016) or using recurrent neural networks (RNNs) for modelling dependency especially recurrence in sequential data. (Chung et al., 2015; Fabius & van Amersfoort, 2014; Gu et al., 2015a; Gan et al., 2015; Sutskever et al., 2014). Most recently, there is a trend of using stochastic neural network in conjunction with neural network or state space model for the purpose of increasing modelling capacity (Gu et al., 2015b; Fraccaro et al., 2016; Bayer & Osendorfer, 2014a). Specifically, (Fraccaro et al., 2016) introduced a broad class of stochastic sequential neural network (SRNN).

To efficiently and accurately exploit the temporal dependency structure, we introduce an efficient structured inference for stochastic sequential neural network (SRNN), which computes each posterior factor's nonlinear and long-term dependence through a bi-directional inference for SRNN. This can utilize both the past and future information captured by the RNN.

We apply the structured inference for SRNN to both synthetic and real-world datesets on music and speech modeling tasks. Experimental results have demonstrated the structured inference significantly outperforms the state-of-the-art methods.

2. Generative Model

Fristly, throughout this paper, we use z_t, z_t^l and z_t^r to denote hidden variables of state space models, h_t, h_t^l and h_t^r to denote hidden variables of recurrent neural network. a_t denotes the input, and x_t denotes the observation at the time t.

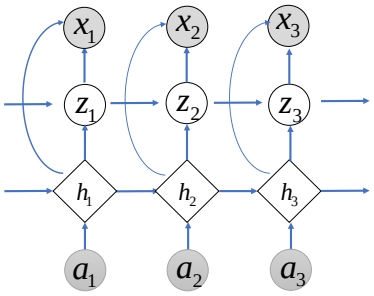

Figure 1. Illustration of the generative model for a single sequence.

As with the SRNN, the generative model interlocks the SSM with the RNN, as illustrated in Figure 1. The joint

[1]SMILE Lab & Yingcai Honors College, University of Electronic Science and Technology of China [2]SMILE Lab & Big Data Research Center School of Computer Science and Engineering, University of Electronic Science and Technology of China. Correspondence to: Zenglin Xu <zenglin@gmail.com>.

ICML 2017 Time Series Workshop, Sydney, Australia, 2017.

probability of an observable sequence and its latent states as follows:

$$\begin{aligned} & p_\theta(x_{1:T}, z_{1:T}, h_{1:T}|a_{1:T}, z_0, h_0) \\ =& p_{\theta_x}(x_{1:T}|z_{1:T}, h_{1:T}) p_{\theta_z}(z_{1:T}|h_{1:T}, z_0) p_{\theta_h}(h_{1:T}|a_{1:T}, h_0) \\ =& \prod_{t=1}^{T} p_{\theta_x}(x_t|z_t, h_t) p_{\theta_z}(z_t|z_{t-1}, h_t) p_{\theta_h}(h_t|h_{t-1}, a_t), \end{aligned} \tag{1}$$

where θ_x, θ_z, and θ_h denote the parameters related to the corresponding conditional distributions. And we set $\theta = \{\theta_x, \theta_z, \theta_h\}$. For a single sequence setting, the log marginal likelihood is

$$\mathcal{L}(\theta) = \log p_\theta(x_{1:T}|a_{1:T}, z_0, h_0). \tag{2}$$

$\mathcal{L}(\theta)$ could be calculated by averaging out the latent states $z_{1:T}$ and $h_{1:T}$ from Equation (1). When there are N sequences in a dynamic system, the whole log marginal likelihood can be written as the summation of each single sequence.

Following Figure 1, the states $h_{1:T}$ are determined by h_0 and $a_{1:T}$ through the recursion $h_t = f_{\theta_h}(h_{t-1}, a_t)$. f_{θ_h} is a GRU network with parameters θ_h.

We assume $p_{\theta_z}(z_i|z_{t-1}, h_t)$ to be a Gaussian distribution with a diagonal covariance structure, namely $p_{\theta_z}(z_t|z_{t-1}, h_t) = \mathcal{N}(z_t; \mu_t, v_t)$. The parameters of the distribution are parameterized by neural networks depending on z_{t-1} and h_t, as follows,

$$\mu_t = f_1(z_{t-1}, h_t), \log v_t = f_2(z_{t-1}, h_t), \tag{3}$$

where $f_i(\cdot)$ denotes a neural network, and $i = 1, 2$.

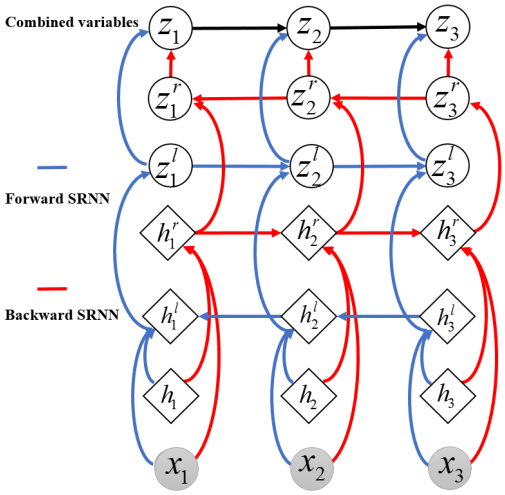

Figure 2. Illustration on the inference network.

3. Structured Inference Network

The posterior is intractable to compute since $z_{1:T}$ cannot be analytically integrated out. Therefore, we need to design an efficient inference network to approximate the true posterior, namely q_ϕ. Instead of maximizing $\mathcal{L}(\theta)$ in Equation (2), we maximize a variational variational evidence lower bound (ELBO). For each sequence, we have $\mathcal{F}(\theta, \phi) \leq \mathcal{L}(\theta)$, and $\mathcal{F}(\theta, \phi)$ is defined as

$$\begin{aligned} \mathcal{F}(\theta, \phi) = \int \int [& q_\phi(z_{1:T}, h_{1:T}|x_{1:T}, \Omega) \\ & \log \frac{p_\theta(x_{1:T}, z_{1:T}, h_{1:T}|\Omega)}{q_\phi(z_{1:T}, h_{1:T}|x_{1:T}, \Omega)}] dh_{1:T} dz_{1:T}, \end{aligned} \tag{4}$$

where $\Omega = \{a_{1:T}, z_0, h_0\}$ is a notation shorthand. When there are multiple sequences in the dynamical system, the ELBO is the sum of the lower bound of each sequence.

Now, we detail the efficient inference network as illustrated in Figure 2. Here the calculation of z_t is not only dependent on the previous state z_{t-1}, but also dependent on the comprehensive information from history and future, denoted by z_t^l and z_t^r, respectively. We assume $z_t \sim \mathcal{N}(z_t; \hat{\mu}_t, \hat{\sigma}_t^2)$, and $\hat{\mu}_t$ and $\hat{\sigma}_t^2$ are parameterized by neural networks as shown below,

$$\begin{aligned} \hat{\mu}_t &= W_\mu \hat{z}_t + b_\mu, \\ \hat{\sigma}_t^2 &= \text{softplus}(W_{\sigma^2} \hat{z}_t + b_{\sigma^2}), \\ \hat{z}_t &= \frac{1}{3}\tanh(W z_{t-1} + b) + z_t^l + z_t^r, \end{aligned} \tag{5}$$

where $\text{softplus}(x) = \ln(1 + e^x), t \in [1, T]$. And we set $\phi_z = \{W_\mu, b_\mu, W_{\sigma^2}, b_{\sigma^2}, W, b\}$.

We assume that z_t^l and z_t^r are respectively subject to the Gaussian distribution. According to the Figure 2, we can get the following formula:

$$\begin{aligned} q_{\phi_{z^l}}(z_t^l|h_{1:T}, x_{1:T}, z_0^l) &= q_{\phi_{z^l}}(z_t^l|z_{t-1}^l, h_t^l), \\ q_{\phi_{z^r}}(z_t^r|h_{1:T}, x_{1:T}, z_0^r) &= q_{\phi_{z^r}}(z_t^r|z_{t+1}^r, h_t^r), \end{aligned}$$

where $h_t^l = g_{\phi_{h^l}}(h_{t+1}^l, [h_t, x_t])$, and $h_t^r = g_{\phi_{h^r}}(h_{t-1}^r, [h_t, x_t])$.

With the knowledge of z_{t-1}^l, the posterior distribution of z_t^l does not depend on the past outputs and deterministic states, but only on the present ones. The advantage of this function is to allow the information propagation from future to the current moment. We model each posterior factor's nonlinear long-term dependence on h_i and $x_{t:T}$ through backward-recurrent function $g_{\phi_{h^l}}$ and $g_{\phi_{h^r}}$, and the inference network can be parameterized by $\phi = \{\phi_z, \phi_{h^l}, \phi_{h^r}, \phi_{z^l}, \phi_{z^r}\}$ and θ_h.

We define $q_{\phi_{z^l}}(z_t^l|z_{t-1}^l, h_t^l) = \mathcal{N}(z_t^l; \mu_t^l, v_t^l)$ and $q_{\phi_{z^r}}(z_t^r|z_{t+1}^r, h_t^r) = \mathcal{N}(z_t^r; \mu_t^r, v_t^r)$, whose mean and the

log-variance are parameterized by neural network, as follows,

$$\mu_t^l = f_3(z_{t-1}, h_t^l), \log v_t^l = f_4(z_{t-1}, h_t^l),$$
$$\mu_t^r = \tilde{f}_3(z_{t+1}, h_t^r), \log v_t^r = \tilde{f}_4(z_{t+1}, h_t^r), \quad (6)$$

where $f_i(\cdot), \tilde{f}_i(\cdot)$ denote a neural network respectively, and $i = 3, 4$.

Following Figure 2, the approximated posterior could be decomposed as

$$\begin{aligned} & q_\phi(z_{1:T}, h_{1:T}|x_{1:T}, \Omega) \\ =& q_\phi(z_{1:T}|h_{1:T}, x_{1:T}, z_0) q(h_{1:T}|x_{1:T}, a_{1:T}, h_0) \\ =& q_\phi(z_{1:T}|h_{1:T}, x_{1:T}, z_0) q(h_{1:T}|a_{1:T}, h_0). \end{aligned} \quad (7)$$

The first factor of Equation (7) could be decomposed as

$$q_\phi(z_{1:T}|h_{1:T}, x_{1:T}, z_0) = \prod_t q_\phi(z_t|z_{t-1}, h_{1:T}, x_{1:T}), \quad (8)$$

which is dependent on $q_{\phi_{z^l}}(z_{1:T}^l|h_{1:T}, x_{1:T}, z_0^l)$ and $q_{\phi_{z^r}}(z_{1:T}^r|h_{1:T}, x_{1:T}, z_0^r)$.

Because $h_{1:T}$ in the second factor of Equation (7) depend on $a_{1:T}$ and h_0, the second factor of Equation (7) is equal to $p_{\theta_h}(h_{1:T}|a_{1:T}, h_0)$ in the generative model. Thus we have the evidence lower bound as follows:

$$\begin{aligned} \mathcal{F}(\theta, \phi) =& E_{q_\phi}[\log p_\theta(x_{1:T}|z_{1:T}, h_{1:T})] \\ & - KL(q_\phi(z_{1:T}|h_{1:T}, x_{1:T}, z_0)||p_\theta(z_{1:T}|h_{1:T}, z_0)). \end{aligned}$$

Finally, according to the Equation (8), the ELBO can be separated as a sum over time steps,

$$\begin{aligned} \mathcal{F}(\theta, \phi) = \sum_t E_{q_\phi^\star(z_{t-1})}[E_{q_\phi(z_t|z_{t-1}, h_{1:T}, x_{1:T})}[\\ \log p_\theta(x_t|z_t, h_t)] - KL(\\ q_\phi(z_t|z_{t-1}, h_{1:T}, x_{1:T})||p_\theta(z_t|z_{t-1}, h_t))], \end{aligned} \quad (9)$$

where $q_\phi^\star(z_{t-1})$ denotes the marginal distribution of z_{t-1} given by

$$\begin{aligned} q_\phi^\star(z_{t-1}) &= \int q_\phi(z_{1:t-1}|h_{1:T}, x_{1:T}, z_0) dz_{1:t-2} \\ &= E_{q_\phi^\star(z_{t-2})}[q_\phi(z_{t-1}|z_{t-2}, h_{1:T}, x_{1:T})]. \end{aligned} \quad (10)$$

Maximizing $\mathcal{F}(\theta, \phi)$ over parameters θ and ϕ can be done by the stochastic gradient ascent algorithm. Our proposed inference for SRNN is summarized in Algorithm.1.

4. Experiment

In this section, we evaluate our proposed method on two synthetic datasets and three real world dataset: TIMIT, Blizzard and Polyphonic Music.

Algorithm 1 Strucutured Inference Algorithm for SRNN

inputs: Observed sequences $\{x^{(n)}\}_{n=1}^N$,z_0,h_0,z_0^r,z_0^l;
Randomly initialized $\phi^{(0)}$ and $\theta^{(0)}$;
Inference Model: $q_\phi(z_{1:T}, h_{1:T}|x_{1:T})$;
Generative Model: $p_\theta(x_{1:T}, z_{1:T}, h_{1:T})$;
outputs:Model parameters θ and ϕ;
for $i = 1$ to Iter **do**
1. Sample data points.
2. Estimate parameters of Eq.3.
3. Estimate posterior parameters of Eq.5 and Eq.6.
5. Evaluate ELBO Eq.4 and estimate MC approx. to $\nabla_\theta F$ and $\nabla_\phi F$.
5. Update $\theta^{(i)}$, $\phi^{(i)}$ using the ADAM.

end for

4.1. Synthetic Datasets

Synthetic Sequential Data We first test our model on a synthetic sequential data generated from a two-dimensional and non-linear GSSM with $N = 5000, T = 25$ as follows:

$$z_{1:T} \sim \mathcal{N}([0.2z_{t-1}^0 + \tanh(\alpha z_{t-1}^1); 0.2z_{t-1}^1 + \sin(\beta z_{t-1}^0)], \mathbf{I})$$
$$x_{1:T} \sim \mathcal{N}(0.5z_{1:T}, \mathbf{I}),$$

where $[\cdot;\cdot]$ denotes a concatenation and $\mathbf{I}$ denotes the identity matrix. The objective is to recover the ground-truth generation parameters $\alpha^\star = 0.50$ and $\beta^\star = -0.10$ for the dataset. Results shown in Figure 3 illustrate that our model could quickly find the true parameters.

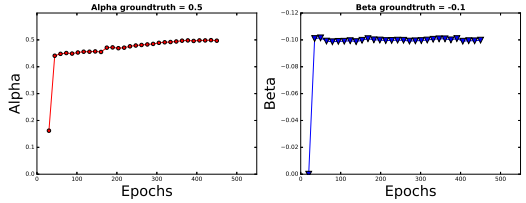

Figure 3. Parameter estimation: Learning parameters α; β in a synthetic non-linear GSSM .

Linear Dynamic System We also evaluate our model on a dynamic system. The data is generated by

$$x_t \sim N(x_{t-1} + 0.05, 4), \quad y_t \sim N(0.5y_{t-1}, 25x_t^2).$$

Here y_t is the observation and x_t is the latent variable. Our goal is to discover the latent space given the observation. The results are shown in Figure 4.

Bouncing Balls The dataset simulates three balls rolling and bouncing within a bounding box on a plane. The dynamic movement of balls are highly dependent on the positions and velocities. We follow the procedure in (Sutskever et al., 2009; Gan et al., 2015), and generate 4000 videos for training, and 200 videos for testing. Each video is of

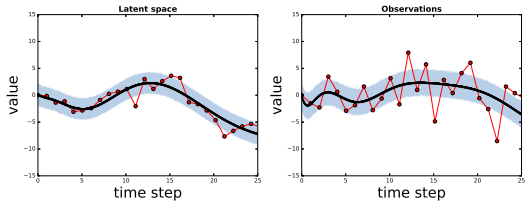

Figure 4. Visualization of the latent space and observations. Shading areas denote the standard deviations.

length 100 and of resolution 30×30. Our goal is to predict the next movement of the balls, and we visualize 8 consecutive frames of the videos, as reported in Figure 5. As can be seen, our model captures the movement of the balls accurately, demonstrating its advantages of long-term prediction under uncertainty.

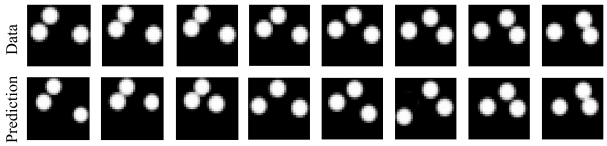

Figure 5. Visulization of 8 consecutive frames.

4.2. Real World Datasets

Blizzard & TIMIT We evaluate our model on the modeling of two speech data, i.e., Blizzard and TIMIT datasets. Blizzard records 300 hours English speech data by a single female speaker and TIMIT consists of 6300 English sentences read by 630 speakers. The preprocessing of the data and the performance measures are identical to those reported in (Chung et al., 2015; Fraccaro et al., 2016). For Blizzard we report the average log-likelihood for half-second sequences, and for TIMIT we report the average log likelihood per sequence for the test data. For the raw audio datasets, we use a fully factorized Gaussian output distribution. The learning rate is set 0.002 and batch size is 128 for Blizzard, for TIMIT they are 0.001 and 64 respectively.

We report results in Table 1. It can be found that our method outperforms most state of the art baselines and shows superior ability in modeling complex dependency in sequential data.

Polyphonic Music: Additionally, we test our method for modeling sequences of polyphonic music (Boulanger-Lewandowski et al., 2012), Each dataset contains more than 7 hours of polyphonic music with varying complexity: folk tunes (Folk), the four-part chorales by J. S. Bach (JSB), orchestral music (Muse) and classical piano music (Piano).

Table 2 compares the average log-likelihood on the test data with multiple state of the art baselines. The results of HMSBN, TSBN are from (Gan et al., 2015), NASMC from (Gu et al., 2015a), STORN from (Bayer & Osendorfer, 2014b), RNN-NADE and RNN from (Boulanger-Lewandowski et al., 2012), SRNN from (Fraccaro et al., 2016), and DMM from (Krishnan et al., 2016).

It can be seen that our method achieves competitive or much better results than other methods.

Table 1. Average log-likelihood per sequence on the test sets.

MODELS	BLIZZARD	TIMIT
SRNN(SMOOTH+Res_q)	$\geq$ 11991	$\geq$ 60550
SRNN(SMOOTH)	$\geq$10991	$\geq$59269
SRNN(FILT)	$\geq$10846	50524
VRNN-GMM	$\geq$ 9107	$\geq$ 28982
VRNN-GAUSS	$\geq$ 9223	$\geq$ 28805
VRNN-I-GAUSS	$\geq$ 9223	$\geq$28805
RNN-GMM	7413	26643
RNN-GAUSS	3539	-1900
OUR METHOD	$\geq$ 13756	$\geq$ 60899

Table 2. Test negative log-likelihood on Polyphonic Music Generation data, the lower the better.

MODELS	FOLK	JSB	MUSE	PIANO
SRNN (SMOOTH+Res_q)	$\geq$ -2.94	$\geq$ -4.74	$\geq$ -6.28	$\geq$ -8.20
TSBN	$\geq$ -3.67	$\geq$-7.48	$\geq$-6.83	$\geq$ -7.94
NASMC	$\approx$ -2.71	$\approx$ -3.98	$\approx$ -6.88	$\approx$ -7.62
STORN	$\approx$ -2.85	$\approx$ -6.93	$\approx$ -6.17	$\approx$ -7.15
RNN-NADE	$\approx$ -2.31	$\approx$ -5.19	$\approx$ -5.60	$\approx$ -7.05
RNN	$\approx$ -4.45	$\approx$ -8.72	$\approx$ -8.11	$\approx$ -8.34
DMM	$\approx$ - 2.77	$\approx$ -6.39	$\approx$ - 6.83	$\approx$ - 7.84
HMSBN	$\geq$ -7.98	$\geq$ -5.13	$\geq$ -9.79	$\geq$ -8.90
OURS	$\approx$- 2.70	$\approx$ -6.71	$\approx$ - 7.91	$\approx$ -7.91

5. Conclusion

Recurrent neural networks with stochastic variables has became very popular for modelling sequential data. We propose an efficient structured inference method, which explicitly combines both past and future information by approximating the posterior through RNNs and graphical model. Experimental results in both synthetic and real-world sequential benchmarks have fully demonstrated the advantages of our proposed inference scheme.

References

Bayer, Justin and Osendorfer, Christian. Learning stochastic recurrent networks. *CoRR*, abs/1411.7610, 2014a.

Bayer, Justin and Osendorfer, Christian. Learning stochastic recurrent networks. *arXiv preprint arXiv:1411.7610*, 2014b.

Boulanger-Lewandowski, Nicolas, Bengio, Yoshua, and Vincent, Pascal. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. *arXiv preprint arXiv:1206.6392*, 2012.

Chung, Junyoung, Kastner, Kyle, Dinh, Laurent, Goel, Kratarth, Courville, Aaron C, and Bengio, Yoshua. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pp. 2980–2988, 2015.

Fabius, Otto and van Amersfoort, Joost R. Variational recurrent auto-encoders. *arXiv preprint arXiv:1412.6581*, 2014.

Fraccaro, Marco, Sønderby, Søren Kaae, Paquet, Ulrich, and Winther, Ole. Sequential neural models with stochastic layers. *arXiv preprint arXiv:1605.07571*, 2016.

Gan, Zhe, Li, Chunyuan, Henao, Ricardo, Carlson, David E, and Carin, Lawrence. Deep temporal sigmoid belief networks for sequence modeling. In *Advances in Neural Information Processing Systems*, pp. 2467–2475, 2015.

Gu, Shixiang, Ghahramani, Zoubin, and Turner, Richard E. Neural adaptive sequential monte carlo. In *Advances in Neural Information Processing Systems*, pp. 2629–2637, 2015a.

Gu, Shixiang, Levine, Sergey, Sutskever, Ilya, and Mnih, Andriy. Muprop: Unbiased backpropagation for stochastic neural networks. *CoRR*, abs/1511.05176, 2015b.

Johnson, Matthew, Duvenaud, David K, Wiltschko, Alex, Adams, Ryan P, and Datta, Sandeep R. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in Neural Information Processing Systems*, pp. 2946–2954, 2016.

Kalman, Rudolf Emil. Mathematical description of linear dynamical systems. *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, 1(2):152–192, 1963.

Krishnan, Rahul G, Shalit, Uri, and Sontag, David. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

Krishnan, Rahul G, Shalit, Uri, and Sontag, David. Structured inference networks for nonlinear state space models. *arXiv preprint arXiv:1609.09869*, 2016.

Rabiner, Lawrence and Juang, B. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

Roweis, Sam and Ghahramani, Zoubin. A unifying review of linear gaussian models. *Neural computation*, 11(2):305–345, 1999.

Sutskever, Ilya, Hinton, Geoffrey E, and Taylor, Graham W. The recurrent temporal restricted boltzmann machine. In *Advances in Neural Information Processing Systems*, pp. 1601–1608, 2009.

Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.