Kalman Filter Powered Variational Autoencoder for Acoustic Unit Discovery

Anonymous Authors[1]

Abstract

Variational Autoencoder (VAE) empowers identification of dominant latent structure to approximate Bayesian Inference for observation models. Structured Variational Autoencoders (SVAE) have been shown to provide efficient neural network-based approximate inference in the presence of both discrete and continuous latent variables. Inspired by SVAE, a VAE has been developed with Extended Kalman Filter(EKF)s to model latent variables. The contribution of this paper is to introduce the Extended Kalman Filter (EKF) for Variational Autoencoder (VAE) to the task of acoustic unit discovery combining the benefit of EKF for continuous space modeling of latent variables with the power of deep generative models provided by VAE. The EKF-VAE is designed to identify and leverage the latent variable structure. With Extended Kalman Filter in Linear-Gaussian State-Space models, the accuracy of the acoustic unit discovery has been significantly improved by reducing the training loss by 47% and root-mean-square error by more than 50% by the EKF-VAE for acoustic discovery on the TIMIT dataset (Garofolo et al., 1992). The experimental results demonstrated that the EKF-VAE model outperforms Hidden Markov Model (HMM) VAEs (Ebbers et al., 2017) in acoustic discovery.

1. Introduction

Many speech technologies such as automatic speech recognition (ASR) and text-to-speech synthesis (TTS) have been used widely around the world. However, most such systems only cover rich-resource languages. For low-resource languages, using such technologies remains limited due to lack of labeled datasets to achieve good performance.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

It is important to develop unsupervised learning algorithms which can use the unlabelled data. For acoustic discovery where methods of speech processing needs to be learnt from raw speech, the task of acoustic discovery can be recognized. Finding the phone-like subword units as acoustic building blocks to discover semantically meaningful linguistic building blocks is necessary. The Variational Autoencoder which can be used for generative purposes as its latent variable models uses deep neural networks to parameterize flexible probability distributions can be an important tool to perform any recognition task including AUD. An auto-encoder encodes some input into a new and usually more compact representation which can be used to reconstruct the input data again. A VAE makes the assumption that the compact representation follows a probabilistic distribution (usually Gaussian) which makes it possible to sample new points and decode them into new data from a trained variational auto-encoder. There have been lots of work being done using Hidden Markov Model (HMM) (Ebbers et al., 2017) as well as using Latent Variable Models (LVM). One striking difference between these two models is that in the Hidden Markov Model (HMM) the latent variables are discrete while in LVM the latent variables can be continuous. One more difference is that for LVM both the latent and observed variables follow Gaussian Distribution, while for HMM only the observed variables have to be of Gaussian Distribution. LVM is a class of statistical models that seek to model the relationship of observed variables with a set of latent variables to allow for modeling of more complex, generative processes. The inference in these models is often difficult or intractable, motivating a class of variational methods that frame the inference problem as optimization. Variational Autoencoders (Kingma & Welling, 2014), in particular, have seen success in tasks of image generation (Gregor et al., 2015).

On the other hand, the Kalman Filter (Kalman, 1960) has been popular as one of state-of-the-art algorithms for estimating dynamic state for noisy and incomplete measurements in discrete-time for multiple decades. The Extended Kalman Filter (EKF) (Julier & Uhlmann, 1997) handles the system nonlinearities through conversion of nonlinear system equations into linear ones by applying a first order Taylor-series approximation around the current mean error

and covariance, so that the traditional linear Kalman filter can be applied.

The contribution of this work is to apply the Extended Kalman Filter (Chai et al., 2002) to VAE to improvise the prediction of the words by moving in a direction which will reduce the distance between the actual word and predicted word. Applied Kalman Filter at every training step of the VAE to reduce the error and training losses by as much as 48% while incurring a maximum overhead of less than 14% of the epoch duration. Used PyTorch (Paszke et al., 2017),(Paszke et al., 2019) Gated Recurrent Unit (GRU) and Recurrent Neural Network (RNN) (Schuster & Paliwal, 1997) to model the speech recognition aka Acoustic Unit Discovery (AUD). Applied the model to TIMIT (Garofolo et al., 1992) and compared its performance against a Hidden Markov Model (HMM) (Ebbers et al., 2017).

The paper is organized as follows. In Section 2 discusses about the related work in the field of Variational Autoencoder (VAE) and Kalman Filter. The Section 3 recapitulates the core concepts used in building the proposed model, e.g., VAE, Recurrent Neural Network based encoder and decoder. The proposed EKF-VAE model is introduced in Section 4 along with model estimation forward algorithms, Kalman Filter and its extended version. Section 5 describes the AUD experiments being conducted on the TIMIT (Garofolo et al., 1992) database, while Section 6 offers some conclusions.

2. Related Work

There has been substantial exploration on both the acoustic discovery and variational autoencoder fronts. The attention mechanism (Bahdanau et al., 2016) has been extensively used with RNN encoder-decoder models (Wang & Jiang, 2016) to enhance their ability to deal with long source inputs. A basic RNN-based VAE (Bowman et al., 2016) generative model has been used to explicitly model different properties of sentences to propose two workarounds: 1. KL cost annealing and 2. masking parts of the source and target tokens with special symbols in order to improvise inference by weakening the decoder. The intent was to generate coherent novel sentences as opposed to AUD that interpolate between known sentences using RNN-based VAE.

The Kalman variational autoencoder (KVAE) (Fraccaro et al., 2017) extends ideas from the SVAE, modelling latent state using a Linear Gaussian State Space Model (LGSSM). To allow for non-linear dynamics, the KVAE uses a recognition model to produce time-varying parameters for the LGSSM, weighting a set of K constant parameters using weights generated by a neural network. They applied KVAE to learn a recognition and dynamics model from video and used it to impute missing data and perform long-term generation in four different environments.

(Tan & Peharz, 2019) proposed decomposing of the overall learning problem into many smaller problems, which are coordinated by the hierarchical mixture, represented by a sum-product network (SPN) and showed that their model outperform classical VAEs on almost all of their experimental benchmarks.

In a recent work on HMM-VAE (Ebbers et al., 2017), the Hidden Markov Models (HMMs) was being used as latent models to perform acoustic unit discovery (AUD) in a zero resource scenario, and showed significant improvement in the accuracy of the acoustic unit discovery. The kernel Kalman rule has been proven as an improvement over the Kernel Bayes rule (Gebhardt et al., 2017). The Extended Kalman Filter being superior in modeling dynamic state, applied the Extended Kalman Filter with Variational Auto-encoder to bring further improvement in accuracy in acoustic unit discovery.

(Pagnoni et al., 2018) augmented the encoder-decoder NMT paradigm by introducing a continuous latent variable to model features of the translation process by extending this model with a co-attention mechanism motivated by (Parikh et al., 2016) in the inference network to show that the conditional variational model improves upon both discriminating attention-based translation and the conditional variational language model for machine translation presented in (Zhang et al., 2016). (Pagnoni et al., 2018) presented some exploration of the learned latent space to illustrate what the latent variable is capable of capturing by utilizing the latent variable without weakening the translation model.

The contrastive variational autoencoder (cVAE) (Abid & Zou, 2019) was designed to identify and enhance salient latent features by explicitly modeling latent features that are shared between the MNIST dataset (LeCun & Cortes, 2010), as well as those that are enriched in one dataset relative to the other.

(Tjandra et al., 2020) built a Transformer-based Vector Quantised Variational AutoEncoder (VQ-VAE) for unsupervised unit discovery system that addresses two major components such as 1) given speech audio, extract subword units in an unsupervised way and 2) re-synthesize the audio from novel speakers. In VQ-VAE, the focus was on the discrete latent representation as opposed to continuous representations costing it to achieve likelihood close but not as good as the continuous representation.

The work proposed in this paper bears some resemblance to the HMM-VAE (Ebbers et al., 2017) or VQ-VAE (Tjandra et al., 2020) which used VAE for acoustic discovery but it is an improvement over VQ-VAE model as it leveraged the continuous latent space representation along with handling the non-linearities of the system. Also, EKF-VAE is an enhancement over HMM-VAE as it reduces the error and

loss by taking care of the non-linearities of the system by updating the state, weights and covariance effectively.

3. Background

First, let me give a brief overview of Variational Autoencoder (VAE)s along with Recurrent Neural Network (RNN) based Encoder-Decoder.

3.1. Variational Autoencoder

Variational Autoencoders (Kingma & Welling, 2014) (VAEs) got popularity in unsupervised learning (Varadarajan et al., 2008) of complicated distributions.

For every datapoint Z in the dataset, (Doersch, 2021) there is one (or many) settings of the latent variables which causes the model to generate something very similar to Z. The objective is to optimize θ such that we can sample x from p(x) and, with to maximize probability, f(x; θ) with the aim maximize the probability of each Z in the training set under the entire generative process,

$$P(Z) = \int P(Z|x;\theta)P(x)dx \tag{1}$$

Here, $f(x;\theta)$ has been replaced by a distribution $p(Z|x;\theta)$, which allows us to make the dependence of Z on x explicit by using the law of total probability. Based on the principle of "Maximum Likelihood" if the model is likely to produce training set samples, then it is also likely to produce similar samples, and unlikely to produce dissimilar ones.

$$\hat{x}_{ml} = arg\ \max_{x}\ p(z|x) \tag{2}$$

In VAEs, the choice of this output distribution is Gaussian, i.e., $p(Z|x;\theta) = \mathcal{N}(Z|f(x;\theta), \sigma^2 I)$.

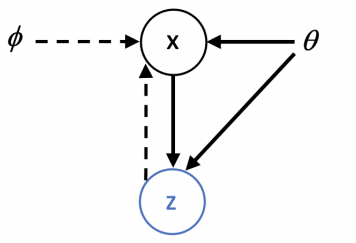

Figure 1. Generative Model in solid lines for $p_\theta(x)p_\theta(z|x)$, dashed lines denote the variational approximation $q_\phi(x|z)$ to intractable posterior $p_\theta(x|z)$.

The Kullback-Leibler divergence (KL divergence) between $p(x|Z)$ and Q(x) for some arbitrary Q, can be written while applying Bayes rule to $p(x|Z)$ as follows:

$$KL[Q(x)\ ||\ p(x|Z)] = E_{x\sim Q}[log\ Q(x) - log\ p(Z|x) - log\ p(x)] + log\ p(Z) \tag{3}$$

Here, log p(Z) comes out of the expectation because it does not depend on latent variable x. The objective is to construct a Q which does depend on Z, and in particular, one which makes $KL[Q(x)\ ||\ p(x|z)]$ small:

$$log\ p(Z) - KL[Q(x|Z)\ ||\ p(x|Z)] = E_{x\sim Q}[log\ p(Z|x)] - KL[Q(x|Z)\ ||\ p(x)] \tag{4}$$

the left hand side of Equation 4 has the quantity which needs to be maximized: log p(Z) (plus an error term, which makes Q produce x's that can reproduce a given Z. The right hand side is optimized by Extended Kalman Filter (or stochastic gradient descent) given the right choice of Q.

The first term in Equation 4 is a bit more tricky. A possible option is to sample to estimate $E_{x\sim Q}[log\ p(Z|x)]$, but getting a good estimate would require passing many samples of x through f , which would be expensive. The full equation to optimize is:

$$E_{Z\sim KL}[log\ p(Z) - KL[Q(x|Z)\ ||\ p(x|Z)] = E_{Z\sim KL}[E_{x\sim Q}[log\ p(Z|x)] - KL[Q(x|Z)\ ||\ p(x)]] \tag{5}$$

Sample a single value of Z and a single value of x from the distribution $Q(x|Z)$, and compute the gradient of the right-hand side. We can then average the gradient of this function over arbitrarily many samples of X and z, and the result converges to the gradient of Equation 5.

3.2. RNN Based Encoder-Decoder

A novel architecture is built leveraging the *RNN Encoder-Decoder* proposed by Cho et al. (2014) and Sutskever et al. (2014) to perform acoustic discovery.

In the Encoder-Decoder framework, an encoder reads the input sentence, a sequence of vectors $x = (x_1, \cdots, x_{T_x})$, into a vector c. The most common approach is to use an RNN such that

$$h_t = f\left(x_t, h_{t-1}\right) \tag{6}$$

and

$$c = q\left(\{h_1, \cdots, h_{T_x}\}\right),$$

where $h_t \in \mathcal{R}^n$ is a hidden state at time t, and c is a vector generated from the sequence of the hidden states. f and q are some nonlinear functions. Sutskever et al. (2014) used an LSTM as f and $q\left(\{h_1, \cdots, h_T\}\right) = h_T$, for instance.

The decoder (Bahdanau et al., 2016) is trained to predict the next word $y_{t'}$ given the context vector c and all the previously predicted words $\{y_1, \cdots, y_{t'-1}\}$. In other words, the decoder defines a probability over the translation by decomposing the joint probability into the ordered conditionals:

$$p(y) = \prod_{t=1}^{T} p(y_t \mid \{y_1, \cdots, y_{t-1}\}, c), \tag{7}$$

where $y = \left(y_1, \cdots, y_{T_y}\right)$. With an RNN, each conditional probability is modeled as

$$p(y_t \mid \{y_1, \cdots, y_{t-1}\}, c) = g(y_{t-1}, x_t, c), \tag{8}$$

where g is a nonlinear, multi-layered, function that outputs the probability of y_t, and x_t is the hidden state of the RNN.

4. Model

Built the model using RNN, GRU Cell of PyTorch and developed Encoder and Decoder RNN as part of the VAE model for comparing the performance of the proposed model EKF-VAE against the HMM-VAE as baseline.

4.1. Kalman Filter & Extended Kalman Filter

The Kalman filter (Newman, 2006) is being popular in predicting the next state of dynamic systems. It uses an iterative approach to tune the model to rectify the error.

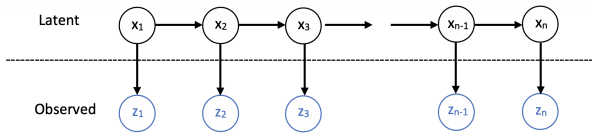

Figure 2. Kalman Filter showing latent and observed variables

The Kalman filter and smoother are based on the following probabilistic model (Miller, 2016).

- Like a discrete-state HMM as shown in the Figure 2, the sequence of observations $z_1, z_2, ..., z_n$ is modeled jointly along with a sequence of hidden latent states $x_1, x_2, ..., x_n$ with the assumption that:

$$p(z_{1:n}, x_{1:n}) = p(x_1)p(z_1|x_1) \prod_{j=2}^{n} p(x_j|x_{j-1})p(z_j|x_j) \tag{9}$$

- Difference from a discrete-state HMM is that each hidden state x_j is modeled as a continuous random variable in $\mathcal{R}^d$ with a multivariate normal distribution.
- The initial distribution $p(x_1)$, the transition distributions $p(x_j|x_{j-1})$ (a.k.a. the "process model") and the emission distributions $p(z_j|x_j)$ (a.k.a. the "measurement model") are assumed to be

$$\begin{aligned} p(x_1) &= \mathcal{N}(x_1|\mu_0, P_0) \\ p(x_j|x_{j-1}) &= \mathcal{N}(x_j|Fx_{j-1}, Q) \\ p(z_j|x_j) &= \mathcal{N}(z_j|Hx_j, R) \end{aligned} \tag{10}$$

where

  - $x_j \in \mathcal{R}^d$ (the state of the system at time step j),
  - $z_j \in \mathcal{R}^d$ (the measurements at time step j),
  - $\mu_0 \in \mathcal{R}^d$ is an arbitrary vector (the initial "best guess" at the initial state),
  - $P_0 \in \mathcal{R}^{d\times d}$ is a symmetric positive definite matrix (the initial covariance matrix, quantifying the uncertainty about the initial state),
  - $F \in \mathcal{R}^{d\times d}$ is an arbitrary matrix (modeling the physics of the process nonlinear vector function, or a linear approximation thereof),
  - $Q \in \mathcal{R}^{d\times d}$ is a symmetric positive definite matrix (quantifying the noise/error in the process that is not captured by F),
  - $H \in \mathcal{R}^{D\times d}$ is an arbitrary matrix (relating the measurements to the state),
  - $R \in \mathcal{R}^{D\times D}$is a symmetric positive definite matrix (quantifying the noise/error of the measurements).

- The model can easily be extended to handle time-dependence in F, Q, H, and R, by simply replacing them with $F_j, Q_j, H_j, and R_j$ in the expressions above.

The Kalman filter (KF) is a method based on recursive Bayesian filtering where the noise in the system is assumed to be Gaussian. The **Extended Kalman Filter** (EKF) is an extension of the classic Kalman Filter for non-linear systems where non-linearity are approximated using the first or second order derivative.

4.2. Extended Kalman Filter: Forward Algorithm

The Extended Kalman filter (EKF) is the nonlinear version of the Kalman filter which linearizes about an estimate of the current mean and covariance.

Model and Observation:
Consider the nonlinear system, described by the difference equation and the observation model with additive noise:

$$x_j = f(x_{j-1}) + w_{j-1} \tag{11}$$

$$z_j = h(x_j) + v_j \tag{12}$$

Initialization:
The initial state x_0 is a random vector with known mean $\mu_0 = E[x_0]$ and covariance $P_0 = E[(x_0 - \mu_0)(x_0 - \mu_0)^T]$.

In the Extended Kalman Filter forward algorithm, Compute $p(x_j|z_{1:j})$ sequentially for j = 1, 2, ..., n in that order. Here is the generalized form for step j (Miller, 2016), $p(x_{j-1}|z_{1:j-1}) = \mathcal{N}(x_{j-1}|\mu_{j-1}, V_{j-1})$

Model Forecast Step/Predictor:
The forecast value for x_j is x_j^f, which can be expressed as:

$$x_j^f \approx f(x_{j-1}^a) \tag{13}$$

Data Assimilation Step/Corrector:
The state-estimate x_j^a can be expressed as:

$$x_j^a \approx x_j^f + K_j(z_j - h(x_j^f)) \tag{14}$$

Kalman Gain at step j K_j can be expressed as:

$$K_j = P_{j-1}H^T(HP_{j-1}H^T + R)^{-1} \tag{15}$$

,

Update the matrix P as it captures the uncertainty about the initial state and improve based on the Kalman Gain K_j

$$P_j = (I - K_jH)P_{j-1} \tag{16}$$

The update of weights of the neural network dW_j can be computed based on the difference of the actual output y and the approximate Jacobian H factored by the Kalman Gain. Approximated Jacobian (H) based on weight, $d\sigma$.

Putting the above equations altogether, the algorithm 1 can be described as follows.

Algorithm 1 Extended Kalman Filter Forward Algorithm

Input: Input state x_j, Observed data z_j, size n and model parameters P_0, F, Q, H, R, step
Initialization:
$K_1 = P_0H^T(HP_0H^T + R)^{-1}$
$P_1 = (I - K_1H)P_0$
for $j = 2$ **to** n **do**
  Approximate Jacobian H
  $K_j = P_{j-1}H^T(HP_{j-1}H^T + R)^{-1}$
  $P_j = (I - K_jH)P_{j-1}$
  $dW_j = stepK_j(z_j - H)$
  $W_j = W_{j-1} + dW_j$
  if Q != 0 **then**
    $P_j = P_j + Q$
  end if
end for

In the above algorithm 1, P is the variance of the state estimation, Q is the variance of the process noise, R captures the variance of the measurement noise.

The Kalman filter is identical to the forward algorithm for discrete-state HMMs, except that it is expressed in terms of μ_j, V_j instead of $s_j(z_j)$ (and the derivation involves an integral instead of a sum).

4.3. EKF-VAE

The Extended Kalman Filter based Variational Autoencoder (EKF-VAE) is built on a Sequence-to-Sequence model made up of an EncoderRNN module and a DecoderRNN module which leverage Extended Kalman Filter.

The forward behavior depends on whether ground-truth is being provided or not. When ground-truth is provided it returns cross-entropy loss, else, it returns predicted word (id).

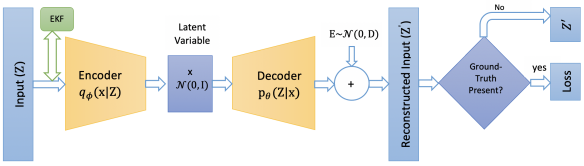

Figure 3. EKF-VAE uses EKF for updating weights and state-estimates. A RNN based Encoder-Decoder to reconstruct by generative process.

The Figure 3 shows a schematic diagram of the EKF based VAE which consists of the following:

- EKF is a neural network to update state-estimation and covariance matrix to provide better prediction.
- RNN Based Encoder-Decoder to perform regenerative actions
- Check against ground-truth (when provided) to compute loss, else, provide reconstructed input.
- Observed and latent variables.

The EKF updates the weight and covariance matrix to improve the state-estimation. The Encoder RNN is a Bi-directional multi-layer gated recurrent unit (GRU) RNN. The Attention decoder is based on **Listen, Attend and Spell (LAS)** (Chan et al., 2016).

The Decoder RNN applies two simple and effective classes of attentional mechanism: a global approach which always attends to all source words and a local one that only looks at a subset of source words at a time similar to the attentional mechanism proposed by (Luong et al., 2015).

The novel EKF-VAE algorithm calls EKF (described in the algorithm 1) to update the weights of the neural network to provide better convergence by predicting the state variables and covariance matrix. Adam optimizer is being used with the $torch.optim.lr\_scheduler.ReduceLROnPlateau$ scheduler as it allows dynamic learning rate reducing based on some validation measurements. The Kalman Filter based

Neural Net (EKF) is a feed-forward neural network (NN). The EKF implements the Extended Kalman Filter algorithm to train. Technically, it could potentially be possible to be trained by stochastic gradient descent (SGD). Trained the NN using the feed-forward function to compute the NN output, and the classify function to round a feed-forward to the nearest class values. Also check-pointed the EKF object in the working directory.

Algorithm 2 EKF Feed Forward Algorithm

Input: Input training data x, activation function σ, current weight W_j of the neural network
$l = \sigma(W_j x)$
$h = W_j l$
return h, l

Updating the weights of the neural network using EKF allows us to predict and update the states. The hypothesis behind weight update using EKF is to capture the non-linearities of the underlying system.

5. Experiments

For experimenting Extended Kalman Filter (EKF) based Variational Autoencoder (VAE), applied Extended Kalman Filter based forward prediction specified in algorithm 1 along with Feed Forward algorithm 2 at each step of the training to improve the prediction of the dynamic state. As the EKF improves the prediction at each step the error (Root Mean Square) reduces as compared to HMM-VAE).

Applied the EKF-VAE model on TIMIT (Garofolo et al., 1992) data and computed the error by computing the distance between the actual and predicted data and learning from it.

5.1. Dataset: TIMIT

The Texas Instruments/Massachusetts Institute of Technology (TIMIT) (Garofolo et al., 1992) corpus of read speech has been designed to provide speech data for the acquisition of acoustic-phonetic knowledge and for the development and evaluation of automatic speech recognition systems. TIMIT contains speech from 630 speakers representing 8 major dialect divisions of American English, each speaking 10 phonetically-rich sentences. The TIMIT corpus includes time-aligned orthographic, phonetic, and word transcriptions, as well as speech waveform data for each spoken sentence.

The acoustic features are 80-dimensional filter banks. They are stacked every 3 consecutive frames, so the time resolution is reduced. Following the standard recipe, used a 462-speaker training set with all SA records removed. Outputs are mapped to 39 phonemes when evaluating.

5.2. Experiment Settings

The EKF-VAE consists of an Encoder Recurrent Neural Network (RNN) and an Decoder Recurrent Neural Network (RNN). The Encoder RNN consists of three encoder layers and the Decoder RNN consists of two decoder layers with Relu activation. The encoder has an input size of 240. Both encoder and decoder have a hidden unit size of 256. The target size is equal to the vocabulary size of the tokenizer.For training purposes, used a batch size of 64 with a dropout percentage of 0.5.

5.3. Results

Trained the EKF-VAE algorithm to perform the acoustic discovery on TIMIT data. The algorithm was trained for 50 iterations and then 100 iterations. Observed that the Root Mean Square Error (RMSE) goes down very fast and but the dev loss plateaued for both EKF-VAE as well as HMM-VAE, but the rate of reduction of RMSE is faster for ELF-VAE and remained even as the number of iterations have been increased.

As shown in Figure 4, the EKF-VAE achieved more than 47% reduction in training loss at the end of 100 iterations (epochs) and experienced a maximum of 49% reduction in dev loss individual epoch as shown in Figure 5. Although both the HMM and EKF are Gaussian based, the training loss reduction is better for EKF-VAE (shown in green) as compared to HMM-VAE as EKF handles the non-linearity and predicts the dynamic state better. As shown in Figure 5, the evaluation (dev) loss has increased in EKF-VAE in later iterations and HMM over-performed by 10%.

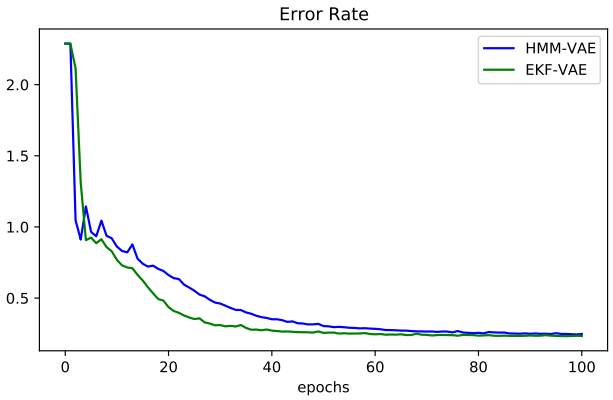

Figure 6. Error Rate

The Kalman filter seems to be effective in modeling language as word utterances can be modeled as discrete occurrences of words. Applying Kalman Filter at each step forced the current state uncertainty P_j to converge faster causing

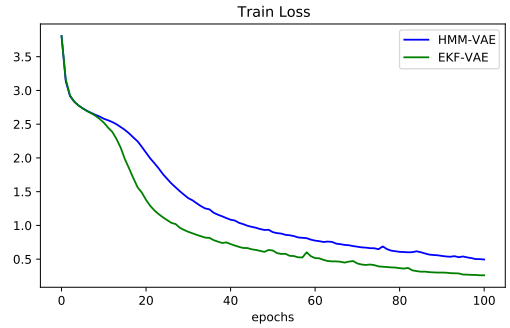

Figure 4. Training Loss

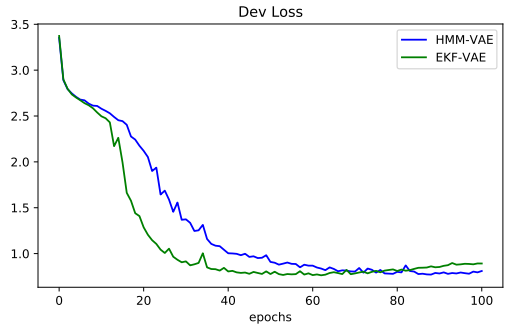

Figure 5. Evaluation/Dev Loss

error rate to go down. For acoustic discovery use-case, the error rate has been reduced up to 37% by Extended Kalman Filter based VAE (EKF-VAE) over HMM VAE as shown in Figure 6.

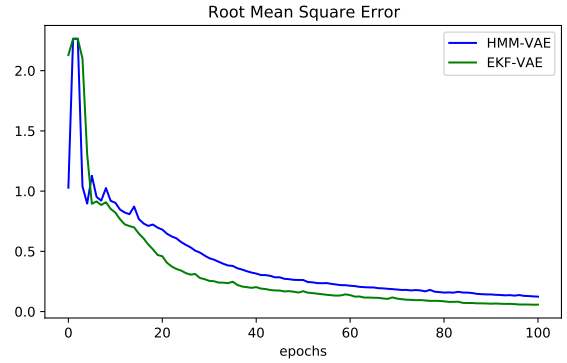

Figure 7. Root Mean Square Error

The error has been computed based on distance of expected and predicted word. The root-mean square error (RMSE) as shown in Figure 7 reduced by 57% at individual epoch in EKF-VAE as compared to its HMM-VAE counterpart.

But, as the EKF is being called in each iteration, it increases the training time. It is essential to understand the trade-off between incurring an additional cost of time for extended kalman filter based prediction to improve convergence and the error reduction rate. The additional Extended Kalman Filter computation increases the training time at every iteration which resulted in less than 14% increase in epoch duration as shown in Figure 8.

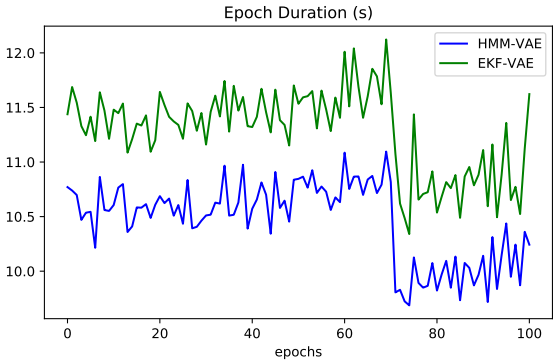

Figure 8. Epoch Duration Comparison between HMM-based and EKF-based VAE

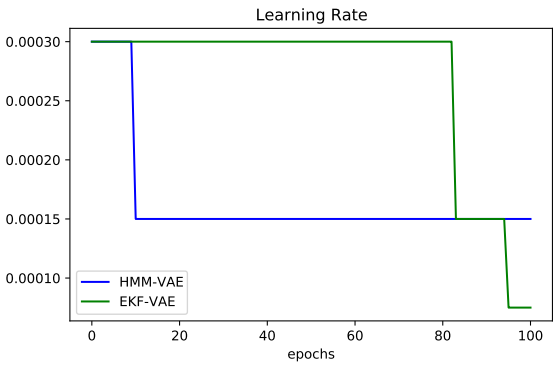

Figure 9. Learning Rate Comparison between HMM-VAE and EKF-VAE

As shown in Figure 9 the learning rate between HMM-VAE and EKF-VAE, the learning rate for HMM-VAE dropped to 0.00005 while the learning rate for EKF-VAE remained at 0.00015. The HMM-VAE kept learning at a higher rate potentially can cause an oscillation in predicting the words. The lower the learning rate, the lesser is the oscillations and

hence can expect to have a better convergence.

The EKF-VAE has predicted the words more accurately than its HMM rivals as it predicted the discrete state more accurately. EKF kept computing the Kalman Gain at each step to decide how much to tune/update the weights to provide a better prediction resulting in superior acoustic discovery.

```
Predict:
w uh kcl k ix dcl b eh r l ix s iy dh ax f iy y ao r dcl z pau f r uw dh ax s n ow f l er ix s h#
Ground-truth:
h# w ax kcl k ix dcl b eh r l ix s iy dh ax f iy y ao r dcl z pau th r uw dh ax s n ow f l er ix s h#
```

Figure 10. Extended Kalman Filter based VAE Prediction

The EKF based predictions as shown in Figure 10 and Figure 11 demonstrates the actual word prediction.

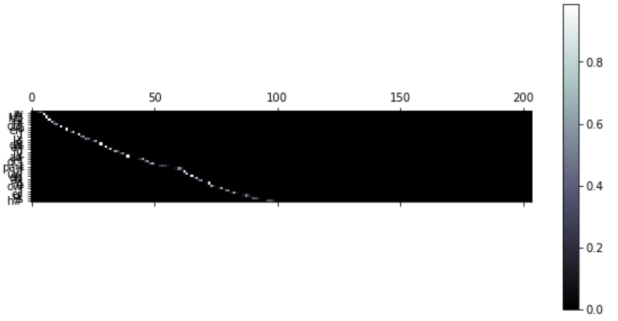

Figure 11. Acoustic Discovery

Using RNN-Based Encoder-Decoder along with the attention model allowed us to have a better word prediction.

6. Conclusions

The proposed an Extended Kalman Filter (EKF) Powered Variational Autoencoder (VAE) for acoustic discovery has achieved a faster convergence as it models the latent space better. From the results, it is evident that the EKF based Variational Model outperforms the HMM based Variational model. The EKF-VAE provides significant improvement over HMM based VAE due to better prediction of the dynamic state by the Kalman filter. Finally, it is shown that the EKF-VAE can reduce the error significantly (49%) to assure superior prediction accuracy.

7. Acknowledgements

Thanks to Prof Bhiksha Raj Ramakrishnan of the Carnegie Mellon University for his immense encouragement and suggestions. Thanks to Nihit Purwar for his support. Thanks to my beautiful wife Mahaswata for motivating me to pursue my passion and buying an NVIDIA RTX2080 based laptop on which all the experiments were performed.

References

Abid, A. and Zou, J. Contrastive variational autoencoder enhances salient features, 2019.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate, 2016.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. Generating sentences from a continuous space, 2016.

Chai, L., Hoff, W. A., and Vincent, T. Three-dimensional motion and structure estimation using inertial sensors and computer vision for augmented reality. *Presence*, 11(5): 474–492, 2002.

Chan, W., Jaitly, N., Le, Q., and Vinyals, O. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4960–4964, 2016.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

Doersch, C. Tutorial on variational autoencoders, 2021.

Ebbers, J., Heymann, J., Drude, L., Glarner, T., Haeb-Umbach, R., and Raj, B. Hidden markov model variational autoencoder for acoustic unit discovery. In *INTERSPEECH 2017, Stockholm, Schweden*, 2017.

Fraccaro, M., Kamronn, S., Paquet, U., and Winther, O. A disentangled recognition and nonlinear dynamics model for unsupervised learning. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *NIPS*, pp. 3601–3610, 2017.

Garofolo, J., Lamel, L., Fisher, W., Fiscus, J., Pallett, D., Dahlgren, N., and Zue, V. Timit acoustic-phonetic continuous speech corpus. *Linguistic Data Consortium*, 11 1992.

Gebhardt, G. H. W., Kupcsik, A., and Neumann, G. The kernel kalman rule - efficient nonparametric inference with recursive least squares. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Gregor, K., Danihelka, I., Graves, A., Rezende, D., and Wierstra, D. Draw: A recurrent neural network for image generation. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1462–1471, Lille, France, 07–09 Jul 2015. PMLR.

Julier, S. J. and Uhlmann, J. K. A new extension of the kalman filter to nonlinear systems. pp. 182–193, 1997.

Kalman, R. E. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1): 35, 1960.

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Luong, M.-T., Pham, H., and Manning, C. D. Effective approaches to attention-based neural machine translation, 2015.

Miller, J. W. Kalman filter and smoother. Lecture Notes on Advanced Stochastic Modeling. Duke University, Durham, NC, 2016.

Newman, P. M. Ekf based navigation and slam. SLAM Summer School 2006, Oxford, July 2006. Background Material, Notes and Example Code.

Pagnoni, A., Liu, K., and Li, S. Conditional variational autoencoder for neural machine translation, 2018.

Parikh, A., Täckström, O., Das, D., and Uszkoreit, J. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2249–2255, Austin, Texas, November 2016. Association for Computational Linguistics.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Schuster, M. and Paliwal, K. K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, 45(11): 2673–2681, 1997.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks, 2014.

Tan, P. L. and Peharz, R. Hierarchical decompositional mixtures of variational autoencoders. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6115–6124, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Tjandra, A., Sakti, S., and Nakamura, S. Transformer vq-vae for unsupervised unit discovery and speech synthesis: Zerospeech 2020 challenge, 2020.

Varadarajan, B., Khudanpur, S., and Dupoux, E. Unsupervised learning of acoustic sub-word units. In *Proceedings of ACL-08: HLT, Short Papers*, pp. 165–168, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Wang, S. and Jiang, J. Learning natural language inference with LSTM. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1442–1451, San Diego, California, June 2016. Association for Computational Linguistics.

Zhang, B., Xiong, D., Su, J., Duan, H., and Zhang, M. Variational neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 521–530, Austin, Texas, November 2016. Association for Computational Linguistics.