ARTIFICIAL BANDWIDTH EXTENSION USING
CONDITIONAL VARIATIONAL AUTO-ENCODERS AND ADVERSARIAL LEARNING

Pramod Bachhav, Massimiliano Todisco and Nicholas Evans

EURECOM, Sophia Antipolis, France
{bachhav, todisco, evans}@eurecom.fr

ABSTRACT

Artificial bandwidth extension (ABE) algorithms have been developed to estimate missing highband frequency components (4-8kHz) to improve quality of narrowband (0-4kHz) telephone calls. Most ABE solutions employ deep neural networks (DNNs) due to their well-known ability to model highly complex, non-linear relationship between narrowband and highband features. Generative models such as conditional variational auto-encoders (CVAEs) are capable of modelling complex data distributions via latent representation learning. This paper reports their application to ABE. CVAEs, form of directed, graphical models, are exploited to model the probability distribution of highband features conditioned on narrowband features. While CVAEs are trained with the standard mean square criterion (MSE), their combination with adversarial learning give further improvements. When compared to results obtained with the baseline approach, the wideband PESQ is improved significantly by 0.21 points. The performance is also compared on an automatic speech recognition (ASR) task on the TIMIT dataset where word error rate (WER) is decreased by an absolute value of 0.3%.

Index Terms— variational auto-encoder, generative adversarial network, latent variable, artificial bandwidth extension, speech quality

1. INTRODUCTION

Legacy narrowband (NB) networks and devices typically support bandwidths of 0-4kHz. Today’s wideband (WB) networks support bandwidths of 50Hz-8kHz and thus provide improved speech quality. While the transition from NB to WB networks will require significant investments and time [1], artificial bandwidth extension (ABE) algorithms have been developed to improve speech quality when WB devices are used with NB devices or infrastructure. ABE methods estimate missing highband (HB) frequency components above 4kHz from available NB features, typically using a regression model learned from WB training data.

ABE algorithms use either a classical source-filter model [2, 3] or operate directly on complex short-term spectral estimates [4, 5]. Estimation is usually performed via a regression approach using conventional Gaussian mixture [2, 6, 7] and hidden Markov models [8, 9]. Several other approaches exploit superiority of deep neural networks (DNNs) to model non-linear relationship between NB and HB components using Gaussian Bernoulli restricted boltzmann machines (GBRBMs), deep recurrent-neural networks (RNNs) with Long short-term memory (LSTM) cells [10, 11], recurrent temporal restricted Boltzmann machines (RTRBMs) [12]. Some approaches perform ABE via direct modelling and generation of time-domain waveforms [13, 14].

Probabilistic deep generative models such as variational auto-encoders (VAEs) and their conditional variant (CVAEs) are capable of modeling complex data distributions. In contrast to bottleneck features learned by stacked auto-encoders (SAEs), the latent representation is probabilistic and can be used to generate new data. Inspired by their successful use in image processing [15, 16], they have become increasingly popular in numerous fields of speech processing (e.g., speech modelling and transformation [17], voice conversion [18]) and neural machine translation (NMT) [19, 20]. The performance of CVAEs can further be improved by their combination with generative adversarial networks (GANs) [21]. Despite their capability of modelling data distributions, CVAEs have not been investigated in regression tasks such as ABE.

Inspired by the approaches presented in [21,22] for image generation task, the work reported in this paper aims to explore the use of generative modelling techniques to further improve performance of a baseline DNN. In particular, we exploit CVAEs to model distribution of HB features where the conditioning variable of the CVAE is derived from NB features via an auxiliary neural network. The performance of the proposed CVAE architecture is further improved via its combination with a GAN. The novel contributions of this work are; (i) the application of CVAEs to ABE for estimation of missing HB features from available NB features; (ii) the combination of CVAE with a probabilistic encoder in the form of an auxiliary neural network and their joint optimisation; (iii) adversarial training of the proposed CVAE architecture to further improve the ABE performance.

The remainder of this paper is organised as follows. Section 2 describes a baseline ABE algorithm. Section 3 explains the proposed CVAE scheme and its combination with GAN for ABE. Experimental setup and results are described in Section 4 and conclusions are presented in Section 5.

2. BASELINE ABE ALGORITHM

Fig. 1 illustrates the baseline ABE system. It is identical to the source-filter model based approach presented in [23]. The algorithm is described in brief in two blocks: estimation and resynthesis.

During estimation, a NB speech frame $s_{NB}$ of 30 ms duration with a sampling rate of 16kHz is processed using a 512-point FFT in order to extract 128-dimensional NB log power spectrum (LPSNB) coefficients $x_{NB}$. Mean and variance normalisation (mvn) is then applied to obtain $x_{mvn}$. After concatenation with the coefficients obtained from 2 neighbouring frames, the resulting 640-dimensional concatenated vector $x_{mvn,c}$ is then fed to a DNN to estimate 10-dimensional normalised HB features $y_{mvn}^{HB}$, consisting of first 10 linear prediction cepstral coefficients (LPCCs). Inverse mean and variance normalisation (mvn$^{-1}$) is then applied, giving HB features $\hat{y}_{HB}$. The
HB LP coefficients $\hat{g}^{\text{NB}}$, $\hat{a}^{\text{NB}}$ are then calculated from the estimated HB LPCCs ($\mathbf{s}^{\text{NB}}$) via recursion.

**Resynthesis** is performed in three steps. First (box in Fig. 1), LP parameters $\hat{a}^{\text{NB}}$, $\hat{g}^{\text{WB}}$ are obtained from speech frame $\mathbf{s}^{\text{NB}}$ via selective linear prediction (SLP$_{\text{NB}}$) to get the NB power spectrum PS$_{\text{NB}}$. This is then concatenated with the HB power spectrum PS$_{\text{NB}}$, and hence estimated WB LP parameters $\hat{g}^{\text{WB}}$, $\hat{a}^{\text{WB}}$. Second (box 2), the HB excitation $\hat{u}^{\text{NB}}$ is estimated from the spectral translation of the NB excitation $\hat{u}^{\text{NB}}$ with $f_m = 8$ kHz (which corresponds to spectral folding around 4kHz). NB and HB excitation components are then combined to obtain the extended WB excitation $\hat{u}^{\text{WB}}$. Finally (box 3), $\hat{u}^{\text{WB}}$ is filtered using a synthesis filter defined by $\hat{g}^{\text{WB}}$ and $\hat{a}^{\text{WB}}$ in order to resynthesise speech frame $\mathbf{s}^{\text{WB}}$. A conventional overlap and add (OLA) technique is used to produce extended WB speech.

### 3. APPLICATION OF CVAE AND GAN FOR ABE

In this section we describe how CVAEs can be used for estimation of HB features from input NB features in an ABE task. The CVAEs are trained in an adversarial fashion to deliver improvements in ABE performance.

#### 3.1. Conditional variational auto-encoders (CVAEs)

A conditional variational auto-encoder (CVAE) is a conditional, generative model of the form $p_{\theta}(y, z|x) = p_{\theta}(z)p_{\theta}(y|x, z)$. For a given input observation $x$, a latent variable $z$ is drawn from a prior distribution $p_{\phi}(z)$ from which the posterior distribution $p_{\theta}(y|x, z)$ generates the output $y$ [15, 16].

CVAE maximises the conditional likelihood $p_{\theta}(y|x, z)$ via the use of recognition/inference model $q_{\phi}(z|y)$ (also referred to as probabilistic encoder) which estimates the parameters $\phi$ of the posterior distribution over all possible values of the latent variables $z$ that may have generated the given datapoint $y$. The probabilistic decoder $p_{\theta}(y|x, z)$ then produces a distribution with parameters $\theta$ over all possible values of $y$ for given $z$ and $x$. For simplicity, it is assumed that the approximate $q_{\phi}(z|y)$ and true posteriors ($p_{\theta}(z|y)$) are diagonal multivariate Gaussian distributions whose respective parameters $\phi$ and $\theta$ are computed using two different DNNs.

The variational lower bound on the conditional likelihood $p_{\theta}(y|x)$ is then given by:

$$\log p_{\theta}(y|x) \geq \mathcal{L}(\theta, \phi; x, y) = -D_{KL}[q_{\phi}(z|y) || p_{\theta}(z)] + E_{q_{\phi}(z|y)}[\log p_{\theta}(y|x, z)] \tag{1}$$

where $D_{KL}(\cdot)$ acts as a regulariser which can be computed analytically. In practice, the prior $p(z)$ is assumed to be a centred isotropic multivariate Gaussian $\mathcal{N}(z; 0, I)$ with no free parameters. The second term is approximated via sampling by $\frac{1}{L} \sum_{l=1}^{L} \log p_{\theta}(y|x, z^{(l)})$ using $L$ samples drawn from the recognition network $q_{\phi}(z|y)$. Sampling is performed using a differentiable deterministic mapping such that $z^{(l)} = g_{\phi}(y, \epsilon^{(l)}) = \mu_{z} + \epsilon^{(l)} \odot \sigma_{z}$ where $\epsilon^{(l)} \sim \mathcal{N}(0, I)$. $\mu_{z} = \mu(y; \phi)$ and $\sigma_{z} = \sigma(y; \phi)$ are outputs of the recognition network $q_{\phi}(z|y)$. This is called the reparameterization trick.

The output distribution $p_{\theta}(y|x, z)$ in Eq. 1 is chosen to be Gaussian with mean $\mu_{x}$ and covariance matrix $\sigma_{x}^{2} + I$, i.e., $p_{\theta}(y|x, z) = \mathcal{N}(\mu_{x}, \sigma_{x}^{2} + I)$ where $\mu_{x}$ is output of the decoder DNN $\mu(x; z; \theta)$. Therefore second term in Eq. 1 can be re-written as:

$$E_{q_{\phi}(z|y)}[\log p_{\theta}(y|x, z)] = C - \frac{1}{L} \sum_{l=1}^{L} \frac{||y - \mu(x, z^{(l)}; \theta)||^{2}}{\alpha} \tag{2}$$

where $C$ is a constant that can be ignored during optimisation. The scalar $\alpha = 2\sigma^{2}$ can be seen as a weighting factor between the KL-divergence and the reconstruction term. In practice, $L = 1$ samples are used per datapoint [24]. The lower bound $\mathcal{L}(\cdot)$ forms the objective function which can be optimized with respect to parameters $\theta$ and $\phi$ using a stochastic gradient descent algorithm.

#### 3.2. Generative adversarial networks

In regression framework, a GAN consists of two adversarial networks, a generator $G$ and a discriminator $D$. Generator maps an input sample $x$ to an output sample $y$. $D$ takes form of a binary classifier which predicts the probability $D(x)$ that a given sample $x$ belongs to the training distribution and not the distribution modelled by $G$. $D$ is thus trained to maximise the probability of assigning a correct label to both training samples and samples from $G$ [25]. This adversarial learning process is formulated as a minimax game between $G$ and $D$ given according to the following objective function:

$$\min_{G} \max_{D} V(G, D) = E_{x}[\log(D(x))] + E_{y}[\log(1 - D(G(y)))] \tag{3}$$

During training, $G$ tries to fool $D$ by generating samples close to the training data so that $D$ classifies $G$’s output as real. $D$ then updates its parameters in order to classify samples generated by $G$ as fake. Both $G$ and $D$ are trained iteratively until the GAN converges to a good estimator of the true data distribution [25].

#### 3.3. Application to ABE

This section describes the proposed scheme for optimisation of CVAEs specifically tailored to ABE. The scheme is illustrated in Fig. 2(a). Parallel training data consisting of NB and WB utterances is processed in frames of 30ms duration with 15ms overlap. Input data $x = x_{\text{NB} \times \text{WB}}$ consists of NB LPS coefficients with memory (as described in Section 2). The output data $y = y_{\text{NB} \times \text{WB}}$ consists of first 10 LPCCs extracted from parallel HB data via SLP.

The CVAE is then trained to model the distribution of the output $y$ conditioned on the input $x$ as follows. The HB data $y$ is fed
Fig. 2. (a) The proposed CVAE-DNN scheme during training (or reconstruction) phase and (b) testing (or prediction) phase.

to the encoder \( q_{\theta_y}(z_y|x) \) (top-left network in Fig. 2(a)) in order to predict the mean \( \mu_{z_y} \) and log-variance \( \log(\sigma^2_{z_y}) \) of the approximate posterior distribution \( q_{\theta_y}(z_y|y) \). The predicted parameters are then used to obtain the latent representation \( z_y \sim q_{\theta_y}(z_y|y) \) of the output variable \( y \) via the reparameterization trick (see Section 3.1). The encoder \( q_{\theta_x}(z_x|x) \) (bottom of Fig. 2(a)) is fed with input data \( x \) in order to predict the mean \( \mu_{z_x} \) and log-variance \( \log(\sigma^2_{z_x}) \) that represent the posterior distribution \( q_{\theta_x}(z_x|x) \). The latent variable \( z_x \sim q_{\theta_x}(z_x|x) \) is then used as the CVAE conditioning variable.

After concatenation, \( z_x \) and \( z_y \) are fed to the decoder \( p_{\theta_y}(y|z_x, z_y) \) (top-right network) in order to predict the mean \( \mu_y = \mu(z_x, z_y; \theta_y) \) of the output variable \( y \). Finally, the entire network is trained to learn parameters \( \phi_x \), \( \phi_y \) and \( \theta_y \) jointly. From Eqs. 1 and 2, the equivalent variational lower bound under optimisation is given by:

\[
\log p_{\theta_y}(y|z_x) \geq L(\theta_y, \phi_y, \phi_x; z_x, y) = -\mathbb{D}_{KL}[q_{\phi_y}(y|z_x)||p_{\theta_y}(y)] - \frac{1}{L} \sum_{l=1}^{L} \left\| \mu(z_x, z_y(l); \theta_y) \right\|^2_c
\]

It is expected that, during optimisation of Eq. 4, parameters \( \phi_x \), \( \phi_y \) and \( \theta_y \) are jointly updated so that the framework learns to reconstruct the data \( y \) from the input data \( x \).

Finally after training (i.e. reconstruction phase), the encoder \( q_{\theta_x}(z_x|x) \) and the decoder \( p_{\theta_y}(y|z_x, z_y) \) (signified by the red components in Fig. 2(a)) networks are used to form a DNN (denoted as CVAE-DNN) with two stochastic layers \( z_x \) and \( z_y \). The proposed CVAE-DNN scheme is illustrated in Fig. 2(b). It can be used for estimation of \( y \) where \( z_y \) is sampled from the prior distribution \( p_{\theta_y}(z_y) = \mathcal{N}(0, I) \) during testing (or estimation phase). It can be noted that there exists a discrepancy during reconstruction and estimation phases of CVAEs since \( y \) is not available during estimation.

3.4. Combining CVAE with GAN

In ABE framework, the reconstruction error term in the objective function given in Eq. 4 can be replaced by GAN based objective loss function (described in Section 3.2) [1, 26]. In this work, this is achieved by forming a generator network \( G \) using the proposed CVAE scheme (which is described in Section 3.3 and shown in Fig. 2(a)). As shown in Fig. 3, \( G \) then trained to reconstruct \( y \) from the inputs \( x \) and \( y \). \( D \) is then optimised to classify the generated \( \hat{y} \) and real \( y \) samples correctly. Both \( G \) and \( D \) are trained simultaneously.

The objective function of the CVAE-GAN scheme is thus given according to:

\[
\mathcal{L}(\theta_y, \phi_y, \phi_x; z_x, y) = -\mathbb{D}_{KL}[q_{\phi_y}(z_y|y)||p_{\theta_y}(z_y)] - \left[ \min_{G} \max_{D} V(G, D) + \lambda \mathcal{L}_1(G) \right]
\]

The error term \( \mathcal{L}_1(G) = \mathbb{E}_y[y - \hat{y}] \) is added in order to minimise the distance between generated \( \hat{y} \) and real \( y \) samples, the trick that helps to stabilise GAN training [27]. After adversarial training of the CVAE, a DNN (denoted as CVAE-GAN) is formed from \( G \) network in a similar fashion as described in 3.3 and shown in Fig. 2(b).

4. EXPERIMENTAL SETUP AND RESULTS

This section describes the datasets used for ABE experiments, baseline algorithm, configuration details of CVAE and GAN architectures, and results. Experiments are designed to compare the performance of ABE systems that use CVAE-DNN and CVAE-GAN with that uses a DNN trained with conventional MSE criterion.

4.1. Database

The dataset by Valentini et al. [28] was used for training and validation. The training set consists of 11572 sentences spoken by 28 English speakers at a sampling rate of 48kHz. Parallel NB and WB speech signals were created by downsampling the original files to 8 and 16kHz respectively. While 80% of feature vectors extracted from the training set were used for training DNN models, remaining 20% samples were used for validation. The acoustically-different TIMIT core test subset [29] was used for testing.

4.2. CVAE configuration and training

The CVAE architecture\(^2\) is implemented using the Keras toolkit [30]. Encoders \( q_{\theta_x}(z_x|x) \) and \( q_{\theta_y}(z_y|y) \) consist of one hidden layer with

\(^2\)Implementation is available at https://github.com/bachhavpramod/bandwidth_extension
Conditional variational auto-encoders (CVAEs) are directed graphical models that are used for generative modelling. This paper reports their application to ABE for modelling distribution of high-band features. The conditioning variable of the CV AE is derived from narrowband features via an auxiliary neural network. CV AEs are also combined with GAN framework via adversarial training in order to seek further improvements. The proposed approaches produce speech of substantially better quality which is confirmed via improvements in WB-PESQ, segSNR estimates. The merit of the proposed approach is further assessed via improvements in ASR performance. Crucially the improvements are achieved without augmenting complexity of the baseline regression model. Future work should investigate why spectral distance measure did not show correlation with the other performance metrics. Better CVAE training strategies in order to reduce the discrepancy during reconstruction and estimation phases should bring further improvements to ABE performance.

4.4. Word error rate (WER) assessments on ASR task

Objective metrics may not provide accurate estimates, subjective listening tests are thus necessary in order to assess ABE performance.
6. REFERENCES


