IMPROVED AUTO-MARKING CONFIDENCE FOR SPOKEN LANGUAGE ASSESSMENT

M. Del Vecchio, A. Malinin, M. J. F. Gales

University of Cambridge, Department of Engineering Trumpington St., Cambridge CB2 1PZ, UK
{md772 am969, mjfg}@eng.cam.ac.uk

ABSTRACT

Automatic assessment of spoken language proficiency is a sought-after technology. These systems often need to handle the operating scenario where candidates have a skill level or first language which was not encountered during the training stage. For high stakes tests it is necessary for those systems to have good grading performance when the candidate is from the same population as those contained in the training set, and they should know when they are likely to perform badly in the case when the candidate is not from the same population as the ones contained in training set. This paper focuses on using Deep Density Networks to yield auto-marking confidence. Firstly, we explore the benefits of parametrising either a predictive distribution or a posterior distribution over the parameters of the model likelihood and obtaining the predictive distribution via marginalisation. Secondly, we investigate how it is possible to act on the parametrised density in order to explicitly teach the model to have low confidence in areas of the observation space where there is no training data by assigning confidence scores to artificially generated data. Lastly, we compare the capabilities of Factor Analysis, Variational Auto-Encodes, and Wasserstein Generative Adversarial Networks to generate artificial data.

Index Terms— Auto-Marking, Confidence, Deep Neural Networks.

1. INTRODUCTION

Systems for automatic assessment of spontaneous spoken language proficiency are becoming increasingly important due to the rise in the demand for English as a second language learning. Previous work [1], [2], [3] and [4] looked at filtering-out non-scorable candidates, effectively performing anomaly detection. One major disadvantage of those systems is that they reject candidates based on whether they can be scored at all, rather than providing a confidence measure over their predictions. To overcome this, [5] successfully used Gaussian Processes (GP) [6] to yield state-of-the-art grading performance whilst leveraging GP's probabilist framework to provide meaningful confidence estimates. However, GPs are known for being computational prohibitive due to their cubic cost in the number of observations.

Deep Neural Networks (DNNs) have achieved state-of-the-art performance on a wide variety of machine learning tasks. However, despite their impressive performance, DNNs are known for not being able to adequately quantify confidence in their predictions, and tend to produce overconfident predictions [7]. Recently, Bayesian Neural Networks (BNNs) [8], [9] have generated increasing attention as a principled framework to provide confidence estimation for deep learning models. BNNs introduce confidence to deep learning models from a Bayesian perspective. By giving a prior to the network parameters, Bayes Theorem is applied to find the posterior distribution of network weights given the training data, instead of a point estimate. Whether the resulting predictive distribution is meaningful depends on the choice of prior distribution, and one should be aware of the fact that inappropriate priors can give rise to arbitrarily bad predictive distributions. Further, due to the complicated non-linearity and non-conjugacy in deep models, exact posterior inference is rarely available. As a consequence, the confidence estimates will not only be affected by the choice of prior but also by the nature of the approximation used in order to make inference in these models tractable. Several inference schemes exists with the state-of-the-art being Hamiltonian Monte Carlo [10] however, those schemes are an active area of research as, at the moment, they cannot be scaled to the millions of parameters found in modern DNNs. Recently, [11] developed a new theoretical framework, Monte Carlo Drop-Out (MCD), casting drop-out training in DNNs as approximate Bayesian inference in Deep Gaussian Processes. In so doing, they were able to derive practical confidence estimates mitigating the problem of sacrificing either computational complexity or test accuracy. Unfortunately, the nature of the confidence estimates greatly depends on the activation functions being used and the drop-out rate chosen.

Lately, [12] proposed a novel method to yield confidence in auto-markers in which a Deep Density Network (DDN) [13] is used to model a predictive distribution and it is trained in a two-stage fashion to yield a high confidence data distribution on observations coming from the training set and a low confidence artificial data distribution on artificial data representing candidates with unseen characteristics. In so

doing, they were able to obtain grading performance comparable to the one obtained with GPs and MCD and confidence estimates which surpassed those of GPs and MCD in the task of deciding which candidates should be backed-off to human graders.

In this work, we adopt and extend the training procedure introduced in [12] as follows. Firstly, we explore the impact that different form of predictive distributions parametrised by a DDN have on confidence. Secondly, we investigate how it is possible to act on the parametrised densities in order to explicitly teach the model to have low confidence in areas of the observation space where there is no training data by assigning confidence scores to artificially generated data. Lastly, we compare the capabilities of Factor Analysis, Variational Auto-Encodes, and Wasserstein Generative Adversarial Networks to generate artificial data.

2. PREDICTIVE DISTRIBUTIONS

In this section, we will explore how different predictive distributions can be parametrised via a DDN. Throughout it, $f_\theta(\boldsymbol{x})$ will denote the output of a DDN, $f(\cdot)$, evaluated at $\boldsymbol{x}$, used to yield a parameter θ of a predictive distribution $p(y|\boldsymbol{x},\theta)$ over the overall grade y. So for example, $p(y|\mu,\sigma^2) = \mathcal{N}(y|f_\mu(\boldsymbol{x}), f_{\sigma^2}(\boldsymbol{x}))$ is a Gaussian whose mean, μ, and variance, σ^2, are modelled by a DDN, $f(\cdot)$. Training in this kind of models will be done via the classical maximum likelihood criterion with the $i.i.d.$ assumption: given training data $\mathcal{D} = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$, we solve

$$\max_{\boldsymbol{\theta}} \sum_{n=1}^{N} \log p(y_n|\boldsymbol{x}_n,\theta). \tag{1}$$

Our ultimate goal is to have a model which can not only predict a grade given a feature representation of the candidate performance, but also provide an estimate of its confidence. A standard way, and the way adopted in this work, is to use the mean and the variance of the predictive distribution as the model prediction of the grade and its confidence in it, respectively

$$\mu_y = \mathbb{E}_{p(y|\boldsymbol{x})}(y) = \int y p(y|\boldsymbol{x}) d\boldsymbol{x}, \tag{2}$$

$$\sigma_y^2 = \mathrm{Var}_{p(y|\boldsymbol{x})}(y) = \int y^2 p(y|\boldsymbol{x}) d\boldsymbol{x} - \mu_y^2. \tag{3}$$

2.1. Form of Predictive Distribution

One of the simplest form of predictive distribution that can be used is a Gaussian with empirical variance

$$p(y|\boldsymbol{x},\mu) = \mathcal{N}(y|f_\mu(\boldsymbol{x}), \hat{\sigma}^2), \tag{4}$$

where $\hat{\sigma}^2$ denotes the empirical variance of the training set. In so doing, the model is effectively predicting a grade and attaching to it a confidence score equal to $\hat{\sigma}^2$ irrespective of what the input was. A natural extension is to model both the mean and the variance of the Gaussian

$$p(y|\boldsymbol{x},\mu,\sigma^2) = \mathcal{N}(y|f_\mu(\boldsymbol{x}), f_{\sigma^2}(\boldsymbol{x})). \tag{5}$$

This allows the model to provide an estimate of the confidence in its predictions by taking into account candidate-specific information.

The Generalised Gaussian (GG), a.k.a the exponential power distribution, represents an extension of the standard Gaussian distribution and can be parametrised as follows

$$\begin{aligned} p(y|\boldsymbol{x},\mu,\alpha,\beta) &= \mathrm{GG}(y|f_{\boldsymbol{\mu}}(\boldsymbol{x}), f_\alpha(\boldsymbol{x}), f_\beta(\boldsymbol{x})) \\ &= \frac{f_\beta(\boldsymbol{x})}{2 f_\alpha(\boldsymbol{x}) \Gamma\left(\frac{1}{f_\beta(\boldsymbol{x})}\right)} \exp\left(- \left(\frac{|y - f_\mu(\boldsymbol{x})|}{f_\alpha(\boldsymbol{x})} \right)^{f_\beta(\boldsymbol{x})} \right). \end{aligned} \tag{6}$$

This family is much more flexible than a standard Gaussian as it allows for tails that are either heavier than a Gaussian (when $\beta < 2$) or lighter than a Gaussian (when $\beta > 2$). Further, it is equivalent to a Gaussian and a Laplace when β is set to 2 and 1, respectively.

So far, we have considered predictive densities which fall under the exponential family of distributions. A widely used distribution which is not from the exponential family is the location-scale t-distribution. This distribution can be parametrised as

$$\begin{aligned} p(y|\boldsymbol{x},\alpha,\mu,\sigma^2) &= t(y|(f_\alpha(\boldsymbol{x}), f_\mu(\boldsymbol{x}), f_{\sigma^2}(\boldsymbol{x})) \\ &= \frac{\Gamma\left(\frac{f_\alpha(\boldsymbol{x})+1}{2}\right)}{\sqrt{\pi f_{\sigma^2}(\boldsymbol{x}) f_\alpha(\boldsymbol{x})} \Gamma\left(\frac{f_\alpha(\boldsymbol{x})}{2}\right)} \\ &\times \left(\frac{f_\alpha(\boldsymbol{x}) + \frac{(y - f_\mu(\boldsymbol{x}))^2}{f_{\sigma^2}(\boldsymbol{x})}}{f_\alpha(\boldsymbol{x})} \right)^{-\frac{f_\alpha(\boldsymbol{x})+1}{2}}. \end{aligned} \tag{7}$$

Interestingly, a predictive t-distribution can also be obtained by using a Gaussian likelihood and assuming a Normal-Inverse-Wishart (NIW) distribution over its parameters. In this case, A DDN is used to parametrise a posterior NIW distribution

$$\begin{aligned} p(\mu,\sigma^2|\boldsymbol{x}) &= NIW(\mu,\sigma^2|f_m(\boldsymbol{x}), f_\lambda(\boldsymbol{x}), f_{\psi^2}(\boldsymbol{x}), f_\rho(\boldsymbol{x})) \\ &= \mathcal{N}\left(y \middle| f_\mu(\boldsymbol{x}), \frac{f_{\sigma^2}(\boldsymbol{x})}{f_\lambda(\boldsymbol{x})}\right) \mathrm{IW}(\sigma^2|f_{\psi^2}(\boldsymbol{x}), f_\rho(\boldsymbol{x}))) \end{aligned} \tag{8}$$

where

$$\mathrm{IW}(\sigma^2|f_{\psi^2}(\boldsymbol{x}), f_\rho(\boldsymbol{x}))) = \frac{\left(f_{\psi^2}(\boldsymbol{x}) \frac{f_\rho(\boldsymbol{x})}{2}\right)^{\frac{f_\rho(\boldsymbol{x})}{2}}}{\Gamma\left(\frac{f_\rho(\boldsymbol{x})}{2}\right)} \frac{e^{\frac{f_\rho(\boldsymbol{x}) f_{\psi^2}(\boldsymbol{x})}{2y}}}{y^{1+\frac{f_\rho(\boldsymbol{x})}{2}}}. \tag{9}$$

Given a Gaussian likelihood over the grade y and a NIW distribution over its parameters, the predictive distribution, which is obtained via marginalisation, is a t-distribution

$$p(y|\boldsymbol{x}) = \int\int p(y|\boldsymbol{x}, \mu, \sigma^2)p(\mu, \sigma^2|\boldsymbol{x})d\mu d\sigma^2. \quad (10)$$

We refer to this parametrisation of the t-distribution as a t-distribution obtained via marginalisation and we denote it as t-distribution†. Although using a DNN to parametrise a posterior NIW over a Gaussian likelihood parameters, yields a t-distribution, doing so provides more flexibility when it comes to estimating the predictive variance, i.e. confidence, Under a t-distribution, the predictive variance is equal to $\frac{\alpha}{\alpha-2}\sigma$, whilst in a t-distribution† it is equal to $\frac{\lambda+1}{\lambda(\rho-2)}\psi^2$. That is, we effectively have one more parameter, λ, that can help in modelling confidence more accurately.

3. EXPLICIT CONFIDENCE MODELLING

DNNs are extremely complex models and guarantees on their behaviour once asked to make predictions in region of the observation space where there is no training data very rarely, if never, exists. In this section we outline how the models presented above can be explicitly trained in a two-stage fashion to have low confidence in region of the observation space where there is no training data via artificially generated data to which we attach confidence scores.

3.1. Training Criterion

In order to teach a model to have low confidence in particular regions of the observation space, we train a DDN in a two-stage fashion. Firstly, we train a DDN $\mathcal{M}$ to parametrise a predictive density $p(y|\boldsymbol{x})$ via maximum likelihood as outlined in Section 2. Then, we initialise another DDN, $\mathcal{M}_1$, from the trained model and we train it to minimise

$$\begin{aligned}\mathcal{L} =&\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}(\mathrm{KL}(q(y|\boldsymbol{x})||p(y|\boldsymbol{x}, \mathcal{M}_1))) \\ &+ \eta\mathbb{E}_{\boldsymbol{x}\sim \tilde{q}(\boldsymbol{x})}(\mathrm{KL}(\tilde{q}(y|\boldsymbol{x})||p(y|\boldsymbol{x}, \mathcal{M}_1))),\end{aligned} \quad (11)$$

In the above equation, $q(\boldsymbol{x})$ and $q(y|\boldsymbol{x})$ are the real training data distribution of $\boldsymbol{x}$ and the target distribution of $\boldsymbol{x}|\boldsymbol{y}$ when $\boldsymbol{x}$ is a training data point. $\tilde{q}(\boldsymbol{x})$ and $\tilde{q}(y|\boldsymbol{x})$ are the distribution of the artificial data and the target distribution over $y|\boldsymbol{x}$ when $\boldsymbol{x}$ is an artificial data point. Therefore, we are essentially asking the model to override what it might have learnt during the ML training stage with the behaviour that we want it to exhibit in particular locations of the observation space. Here, we set $q(y|\boldsymbol{x}) = p(y|\boldsymbol{x}, \mathcal{M})$ and the constant $\eta \in \mathbb{R}$, is used to scale the effect of the second KL divergence.

In the next section, we will go over how one might choose $\tilde{q}(y|\boldsymbol{x})$ and specify the confidence scores of the artificial data depending on what from of predictive distribution has been chosen, and how to generate artificial data. We will do so by focusing on the Gaussian and t-distribution leaving the Generalised Gaussian aside.

3.2. Form of Predictive Distribution over Artificial Data

When the predictive distribution is a Gaussian, we can take the target distribution over the artificial data to be

$$\tilde{q}(y|\boldsymbol{x}) = \mathcal{N}(y|f_{\boldsymbol{\mu}}(\boldsymbol{x}|\mathcal{M}), \nu_{\sigma^2}||\boldsymbol{x} - \bar{\boldsymbol{x}}||^2), \quad (12)$$

i.e. we use of the mean of the Gaussian predictive distribution parametrised by $\mathcal{M}$ as the target mean and the scaled euclidean distance between $\boldsymbol{x}$ and the training data mean $\bar{\boldsymbol{x}}$ as the target variance. In so doing, we are explicitly asking the model to have low confidence away from the mean of the training data. We refer to $\nu_{\sigma^2}||\boldsymbol{x} - \bar{\boldsymbol{x}}||^2$ as the confidence score associated to an artificial data point $\boldsymbol{x}$.

When the predictive distribution is a t-distribution we have two choices when it comes to enforcing low confidence away from the training data. On the one hand, we can act on the predictive t-distribution directly and choose the target distribution over the artificial data to be

$$\tilde{q}(y|\boldsymbol{x}) = t(y|f_\alpha(\boldsymbol{x}|\mathcal{M}), f_\mu(\boldsymbol{x}|\mathcal{M}), \nu_{\sigma^2}||\boldsymbol{x} - \bar{\boldsymbol{x}}||^2). \quad (13)$$

i.e. we use of the degrees of freedom and location of the t-distribution parametrised by $\mathcal{M}$ as the target number of degrees of freedom and location, and the scaled euclidean distance between $\boldsymbol{x}$ and the training data mean $\bar{\boldsymbol{x}}$ as the target scale. Unfortunately, a closed form solution for the KL divergence between two t-distributions distributions does not exist [14]. To overcome this issue, a popular approach, and the approach used in this work, is to approximate the t-distribution with a Gaussian

$$t(y|\alpha, \mu, \sigma^2) \approx \mathcal{N}(y|\mathbb{E}_{t(y|\alpha,\mu,\sigma^2)}(y), \mathrm{Var}_{t(y|\alpha,\mu,\sigma^2)}(y)), \quad (14)$$

where

$$\mathbb{E}_{t(y|\alpha,\mu,\sigma^2)}(y) = \mu \qquad \text{if } \alpha > 1, \text{ else undefined}, \quad (15)$$

$$\mathrm{Var}_{t(y|\alpha,\mu,\sigma^2)}(y) = \frac{\alpha}{\alpha-2}\sigma^2 \quad \text{if } \alpha > 2, \text{ else undefined}. \quad (16)$$

Note that this approximation is only used for KL divergences computations, we do make predictions at test time using this approximate form.

On the other hand, we can act on the parameters of the NIW posterior in which case all the KL diverges are in terms of the distribution over (μ, σ^2), and we can set

$$\begin{aligned}\tilde{q}(\mu, \sigma^2|\boldsymbol{x}) =&NIW(\mu, \sigma^2|f_m(\boldsymbol{x}|\mathcal{M}), f_\lambda(\boldsymbol{x}|\mathcal{M}), \\ &\nu_\rho||\boldsymbol{x} - \bar{\boldsymbol{x}}||^2, \nu_{\psi^2}||\boldsymbol{x} - \bar{\boldsymbol{x}}||^2),\end{aligned} \quad (17)$$

i.e. we use the confidence in the posterior mean λ and the posterior mean itself estimated by $\mathcal{M}$ as the target values and the scaled euclidean distance between x and the training data mean $\bar{\boldsymbol{x}}$ as the target confidence in the posterior scale, ρ, and the posterior scale itself. This way, we are effectively asking the NIW to increase its certainty in a high posterior scale as the distance from the mean of the training data increases.

3.3. Artificial Data Generation

In order to generate artificial observations $\tilde{\boldsymbol{X}}$, we model $\tilde{q}(\boldsymbol{x})$ using a FA model, a VAE, and a WGAN. In this section we give a brief overview all of three models.

FA [15] is possibly the simplest Latent Variable Model (LVM) where the conditional distribution $p(\boldsymbol{x}|\boldsymbol{z})$ is expressed in terms of a linear mapping from latent variables to data variables $\boldsymbol{x} = \boldsymbol{W}\boldsymbol{z}+\boldsymbol{\mu}+\boldsymbol{\epsilon}$, where $\boldsymbol{W}$ and $\boldsymbol{\mu}$ parametrise the linear mapping between the observation space and the latent space, and $\boldsymbol{\epsilon}$ is a noise process independent of $\boldsymbol{x}$. In a FA model, the prior distribution over $\boldsymbol{z}$ is given by $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, \boldsymbol{I})$, and the noise $\boldsymbol{\epsilon}$ is distributed according to $p(\boldsymbol{\epsilon}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, \boldsymbol{\Psi})$, where $\boldsymbol{\Psi}$ is a $d \times d$ diagonal matrix. Because of the linearity of the mapping and the Gaussianity of $\boldsymbol{\epsilon}$, the joint distribution of $\boldsymbol{x}$ and $\boldsymbol{z}$ is Gaussian and as a result, the marginal distribution of the observations and the conditional distribution of the observations given the latent variables are analytical tractable

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{W}\boldsymbol{W}^T + \boldsymbol{\Psi}), \tag{18}$$

$$p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu}, \boldsymbol{\Psi}). \tag{19}$$

We can determine the parameters $\boldsymbol{W}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Psi}$ by maximum likelihood. The solution for $\boldsymbol{\mu}$ is given by the training data mean. However, there is no closed-form solution for $\boldsymbol{W}$ and $\boldsymbol{\Psi}$, which must be found iteratively. Because FA is a LVM, this can be done using the EM algorithm [15].

A VAE [16] is a Latent Variable Model (LVM) where the conditional distribution $p(\boldsymbol{x}|\boldsymbol{z})$, assumed to be Gaussian in the case of a continuous variable, is parametrised by a DDN

$$p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}|f_{\boldsymbol{\mu}}(\boldsymbol{z}|\boldsymbol{\theta}), f_{\boldsymbol{\Sigma}}(\boldsymbol{z}|\boldsymbol{\theta})), \tag{20}$$

where $f_{\boldsymbol{\mu}}(\boldsymbol{z}|\boldsymbol{\theta})$ and $f_{\boldsymbol{\Sigma}}(\boldsymbol{z}|\boldsymbol{\theta})$ are the outputs of the DDN, which is parametrised by $\boldsymbol{\theta}$ and also called the decoder network, providing the mean vector and the diagonal covariance matrix of the Gaussian distribution, respectively. Because of the non-linearities introduced by the DDN however, the true conditional $p(\boldsymbol{z}|\boldsymbol{x})$ is not analytically tractable. To remedy this, an inference network is introduced

$$q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\nu}) = \mathcal{N}(\boldsymbol{z}|f_{\boldsymbol{\mu}}(\boldsymbol{x}|\boldsymbol{\nu}), f_{\boldsymbol{\Sigma}}(\boldsymbol{x}|\boldsymbol{\nu})), \tag{21}$$

where $f_{\boldsymbol{\mu}}(\boldsymbol{x}|\boldsymbol{\nu})$ and $f_{\boldsymbol{\Sigma}}(\boldsymbol{x}|\boldsymbol{\nu})$ are the outputs of the DNN, which is parametrised by $\boldsymbol{\nu}$ and also called the encoder network, providing the mean vector and the diagonal covariance matrix of the variational approximation to the true conditional, respectively. Like in the FA model, we assume that the prior distribution over the latent variables is given by $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. The model and inference network can then be jointly trained using the Auto-Encoding Variational Bayes (AEVB) Estimator [16].

Traditional GANs, [17] argues, typically minimise divergences which are potentially not continuous with respect to the generators parameters, leading to training difficulty. They propose instead using the Earth-Mover, which is also called Wasserstein-1 divergence, and constructing the WGAN value function using the Kantorovich-Rubinstein duality [18]

$$\min_G \max_{C \in \mathcal{C}} \mathbb{E}_{\boldsymbol{x} \sim p_{data}}(C(\boldsymbol{x})) + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}}(C(G(\boldsymbol{z}))), \tag{22}$$

where $\mathcal{C}$ is the set of 1-Lipschitz functions, $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ is the latent space from which the generator $G(\boldsymbol{z}) : \mathcal{Z} \to \mathcal{X}$ creates samples and $C(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$ is the critic. Originally, to enforce the Lipschitz constraint on the critic [17] proposed to clip the weights of the critic to lie within a compact space $[-c, c]$. However, the set of functions satisfying this constraint is a subset of the k-Lipschitz functions for some k which depends on c and the critic architecture. [19] proposed to solve this issue by introducing a gradient penalty in the WGAN loss

$$\begin{aligned} &\min_G \max_{C \in \mathcal{C}} \mathbb{E}_{\boldsymbol{x} \sim p_{data}}(C(\boldsymbol{x})) + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}}(C(G(\boldsymbol{z}))) \\ &+ \gamma \mathbb{E}_{\hat{\boldsymbol{x}} \sim p_{\hat{\boldsymbol{x}}}}(||\nabla_{\hat{\boldsymbol{x}}} C(\hat{\boldsymbol{x}}) - 1||)^2, \end{aligned} \tag{23}$$

where γ is the gradient penalty coefficient and $p_{\hat{\boldsymbol{x}}}$ is explicitly defined by sampling uniformly along straight lines between pairs of points sampled from the data distribution p_{data}, i.e. the training samples, and the generator distribution which is implicitly defined via $p_{\boldsymbol{z}}$. In this work, both the generator and the critic are parametrised by a DNN.

4. EXPERIMENTAL RESULTS

4.1. Data, Assessment and Experimental Procedure

All experiments were performed using 33-dimensional pronunciation, fluency and acoustic features derived from audio and ASR transcriptions of responses to questions from the BULATS exam [20]. The ASR system has a Word Error Rate (WER) of 32% on a development set. The training and test sets have 4300 and 224 candidates, respectively. Each candidate provided a response to 21 questions, and the features used are aggregated over all questions into a single vector. The target variable is 1-dimensional and it represents the average grade obtained by each candidate. The test data was graded by expert graders at Cambridge English. These experts have inter-grader Pearson correlation coefficients (PCCs) in the range 0.95-0.97. Candidates are equally distributed across Common European Framework of Reference

for Languages (CEFR) grade levels [21]. The input features were normalised by subtracting the mean and dividing by the standard deviations for each dimension computed on all the training observations.

Assessing confidence is challenging as the ground truth confidence estimates are usually not available. The operating scenario is to use a model's estimate of confidence in its prediction to decide what candidates should be assessed by human graders for high-stakes tests a decision which we call back-off. As the back-off fraction is increased, model predictions are replaced with true targets according to: (i) random ordering, yielding the expected random back-off curve, (ii) order of decreasing mean squared error MSE relative to true targets, yielding the optimal back-off curve, (iii) order of decreasing predictive variance yielding the model back-off curve. Given an ordering, the model performance is summarised by either the PCC or the mean squared error (MSE) w.r.t. the true targets. In order to asses confidence estimates, we employ the following metric

$$\text{AUC}_r = \frac{\text{AUC}_{model} - \text{AUC}_{radom}}{\text{AUC}_{optimal} - \text{AUC}_{radom}}. \tag{24}$$

where AUC_{radom}, $\text{AUC}_{optimal}$ and AUC_{model} represent the area under the random, optimal and model back-off curves respectively. When the PCC is used to summarise the model performance, we refer to $\text{AUC}_r^{\text{PCC}}$, and when the MSE is used we refer to $\text{AUC}_r^{\text{MSE}}$. It must be noted that this assessment criterion is not completely independent of the baseline performance of the model (the performance when 0% of the candidates are backed-off) so care must be taken when we want to compare the confidence estimates of two models that have very different baseline performance in terms of PCC or MSE.

All the models where tuned on a validation test consisting of a random 10% of the training data. The DDNs used all had 2 hidden layers with 180 units within each layer. The networks weights were initialised from $\mathcal{N}(0, 0.05)$ whilst the biases were initialised at zero. For those DDNs trained using the maximum likelihood criterion, they were trained using Adam [22] with a learning rate of $1e-5$ and batch size of 50 for 800 epochs. For those trained using the two-stage training procedure, vanilla Stochastic Gradient Descent (SGD) with a learning rate of $1e-6$ and batch size of 50 and 1000 epochs of training were used. A drop-out rate of 40% and leaky ReLu activation functions were used. In order to test the sensitivity to the random seed, all the experiments were repeated 5 times and the mean performance together with the standard deviation recorded. However, it was found that the performance metrics used PCC, MSE, $\text{AUC}_r^{\text{PCC}}$, and $\text{AUC}_r^{\text{MSE}}$ had all standard deviations less than 0.001, hence it was chosen not to report them.

The FA model used in this work had a 2-dimensional latent space and it is trained using 100 iterations of the EM algorithm. The VAE had itself a 2-dimensional latent space, and its architecture comprised of an encoder and a decoder each having 2 hidden layers with 180 units within each layer. The networks weights were initialised from $\mathcal{N}(0, 0.01)$ whilst the biases were initialised at zero. The WGAN used had a 2-dimensional latent space, and its architecture comprised of a critic and a generator each having 2 hidden layers with 180 units within each layer. A gradient penalty of 5.0 was used and for each generator update the critic was updated 5 times. Both the VAE and the WGAN were trained using Adam with a learning rate of $1e-05$, for 2000 epochs. A drop-out rate of 40%, and leaky ReLu activation functions were used.

4.2. Results

Table 1 shows that a predictive t-distribution seems to be better suited to the dataset at hand as it yields better uncertainty estimates when compared to a Gaussian and Generalised Gaussian. We also observe that using a Gaussian with empirical variance (Empirical Gaussian) yields slightly better PCC and MSE performance but it does not yield meaningful uncertainty estimates as it makes every prediction with the same level of uncertainty, i.e. the variance of the training data. Unsurprisingly, the Generalised Gaussian performs slightly better in terms of both $\text{AUC}_r^{\text{PCC}}$ and $\text{AUC}_r^{\text{MSE}}$ than a Gaussian since the latter is a special case of the former. We also see a gain in performance over the t-distribution in terms of AUC_r when the t-distribution is obtained via marginalisation. This behaviour is certainly linked to the higher modelling capability of the predictive variance under the t-distribution† as showed by the fact that when fixing the value of λ to 1, the t-distribution† achieves a PCC of 0.870 and $\text{AUC}_r^{\text{PCC}}$ of 0.255. The same trend in the confidence estimates is observed w.r.t. both the PCC and the MSE, so, since the same is true for the subsequent experiments, we report the PCC results.

Table 1. Performance of different forms of predictive distributions.

	Grade		AUC_r	
Pred. Dist.	PCC	MSE	PCC	MSE
Emp. Gaussian	0.874	8.385	-	-
Gaussian	0.870	9.024	0.232	0.185
Gen. Gaussian	0.872	8.834	0.248	0.204
t-distribution	0.870	8.939	0.259	0.223
t-distribution†	0.870	8.996	0.311	0.274

For all three generative models presented above in order to generate artificial observations which are different from the training data but still lie on the training data manifold after training we increase the variance on the prior over the latent variables, i.e. we control β in $p(\boldsymbol{z}) = \mathcal{N}(\mathbf{0}, \beta \boldsymbol{I})$. The VAE achieves much better average negative log likelihood

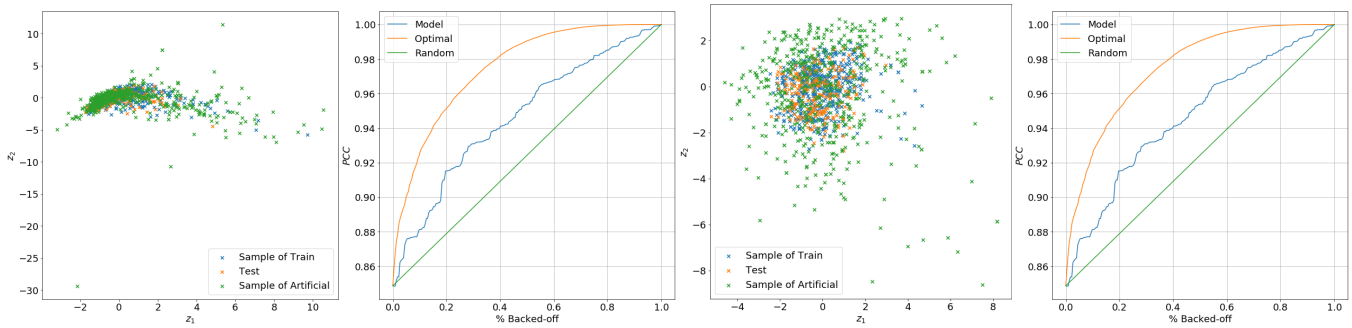

Fig. 1. Two-stage training of a Gaussian: (a)-(b) FA latent space and AUC curves (c)-(d) VAE latent space and AUC curves.

(NLL) on the test set when compared to the FA model (89.952 vs. 186.103) showing the ability of the non-linear mapping between observations and latent variables to model the data manifold much better. In their work [12] used FA-generated artificial data to train a Gaussian predictive distribution to have low confidence in region of the space where there is no training data. Table 2 shows that it is possible to improve on the prediction confidence by using generative models that can model the data manifold more effectively. As we can see, using a VAE and a WGAN over a FA model to generate artificial data, whilst fixing $\eta = 1.0$ and $\nu_{\sigma^2} = 5.0$, yields better confidence estimates with data generated with the VAE yielding the best performance. The difference in performance between using a FA model and a VAE can be explained by looking at the latent space learned by the two models. Figure 1 shows that the FA latent space does not seem to be Gaussian distributed as the linear mapping assumed by the model is simply too restrictive. As a consequence, generating observations by sampling the latent variables from a Gaussian prior might not yield observations which lie on the data manifold.

Table 2. Performance of the Gaussian predictive distribution under the two-stage training criterion.

Pred. Dist.	$\tilde{\boldsymbol{X}}$	PCC	$\text{AUC}_r^{\text{PCC}}$
Gaussian	-	0.870	0.232
”	FA	0.870	0.281
”	VAE	0.870	0.348
”	WGAN	0.870	0.346

Finally, having observed that the artificial data generated using a VAE improved confidence estimates under the Gaussian predictive distribution the most, we moved onto looking at the effect of using the two-stage training criterion with VAE-generated data on the other forms of predictive distribution whilst fixing $\eta = 1.0$, $\nu_{\sigma^2} = 5.0$, and $\nu_\rho = \nu_{\psi^2} = 1e3$. Table 3 shows that once again the training procedure improves the confidence estimates with the t-distribution obtained via marginalisation yielding the best performance. It is also interesting to note that the significant differences observed in terms of AUC_r due to the different form of predictive distribution used tend to disappear once the two-stage training procedure is used.

Table 3. Performance of different predictive distributions under the two-stage training criterion with VAE-generated artificial data.

Pred. Dist.	$\tilde{\boldsymbol{X}}$	PCC	$\text{AUC}_r^{\text{PCC}}$
Gaussian	VAE	0.870	0.348
t-distribution	”	0.870	0.344
t-distribution†	”	0.869	0.368

5. CONCLUSION AND FUTURE WORK

We explored how a deep-learning-based auto-marker estimate of confidence can be improved. We find that parametrising a t-distribution via marginalisation yields better confidence estimates compared to parametrising a Gaussian, Generalised Gaussian, or t-distribution. In [12], it was found that a two-stage training criterion which enforces low confidence away from the training data under a Gaussian predictive distribution could improve an auto-marker estimate of confidence. Our work agrees with this finding and it complements it by showing that using more complex generative models, in this work VAEs, to obtain the artificial data can help improve confidence estimates across different forms of predictive distributions. Further, it was found that differences observed in terms of confidence estimates due to the different form of predictive distribution used tend to disappear once the two-stage training procedure is used. Future work should asses the performance of these models on other tasks and datasets.

6. ACKNOWLEDGEMENTS

Thanks to Cambridge English, and The University of Cambridge for support and access to the BULATS data.

7. REFERENCES

[1] Su-Youn Yoon and Shasha Xie, "Similarity-based non-scorable response detection for automated speech scoring," in *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications*, 2014, pp. 116–123.

[2] Klaus Zechner, Derrick Higgins, Xiaoming Xi, and David M Williamson, "Automatic scoring of non-native spontaneous speech in tests of spoken English," *Speech Communication*, vol. 51, no. 10, pp. 883–895, 2009.

[3] Derrick Higgins, Xiaoming Xi, Klaus Zechner, and David Williamson, "A three-stage approach to the automated scoring of spontaneous spoken responses," *Computer Speech & Language*, vol. 25, no. 2, pp. 282–306, 2011.

[4] Shasha Xie, Keelan Evanini, and Klaus Zechner, "Exploring content features for automated speech scoring," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2012, pp. 103–111.

[5] R. C. van Dalen et al., "Automatically grading learners' English using a Gaussian process," in *Proc of ISCA International Workshop on Speech and Language Technology in Education (SLaTE)*. ISCA, Aug 2015, pp. 7–12.

[6] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[7] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, 2017, pp. 6405–6416.

[8] David JC MacKay, "Probable networks and plausible predictions?a review of practical Bayesian methods for supervised neural networks," *Network: Computation in Neural Systems*, vol. 6, no. 3, pp. 469–505, 1995.

[9] Radford M Neal, *Bayesian learning for neural networks*, vol. 118, Springer Science & Business Media, 2012.

[10] Radford M Neal et al., "MCMC using Hamiltonian Dynamics," *Handbook of Markov Chain Monte Carlo*, vol. 2, no. 11, pp. 2, 2011.

[11] Yarin Gal and Zoubin Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[12] Andrey Malinin, Anton Ragni, Kate Knill, and Mark Gales, "Incorporating Uncertainty into Deep Learning for Spoken Language Assessment," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2017, vol. 2, pp. 45–50.

[13] Christopher M Bishop, "Mixture density networks," *Technical Report NCRG 4288*.

[14] Sengupta Ashis, Samanta Tapas, and Basu Ayanendranath, *Statistical paradigms: recent advances and reconciliations*, vol. 14, World Scientific, 2014.

[15] Kevin P Murphy, "Machine Learning: A Probabilistic Perspective. Adaptive Computation and Machine Learning," 2012.

[16] D.P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd Int. Conf. Learning Representations (ICLR)*, 2013.

[17] Martin Arjovsky, Soumith Chintala, and Léon Bottou, "Wasserstein GAN," *arXiv preprint arXiv:1701.07875*, 2017.

[18] Cédric Villani, *Optimal transport: old and new*, vol. 338, Springer Science & Business Media, 2008.

[19] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville, "Improved training of Wasserstein GANs," in *Advances in Neural Information Processing Systems*, 2017, pp. 5769–5779.

[20] Lucy Chambers and Kate Ingham, "The BULATS Online Speaking Test," *Research Notes*, vol. 43, pp. 21–25, 2011.

[21] Council of Europe. Council for Cultural Co-operation. Education Committee. Modern Languages Division, *Common European Framework of Reference for Languages: learning, teaching, assessment*, Cambridge University Press, 2001.

[22] D.P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *3rd Int. Conf. Learn. Representations (ICLR)*, 2014.