Uncertainty propagation through deep neural networks
Ahmed Hussen Abdelaziz, Shinji Watanabe, John R. Hershey, Emmanuel Vincent, Dorothea Kolossa

To cite this version:
Ahmed Hussen Abdelaziz, Shinji Watanabe, John R. Hershey, Emmanuel Vincent, Dorothea Kolossa. Uncertainty propagation through deep neural networks. Interspeech 2015, Sep 2015, Dresden, Germany. hal-01162550

HAL Id: hal-01162550
https://inria.hal.science/hal-01162550
Submitted on 10 Jun 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Uncertainty Propagation through Deep Neural Networks

Ahmed Hussen Abdelaziz\(^1\)*, Shinji Watanabe\(^2\), John R. Hershey\(^2\), Emmanuel Vincent\(^3\), Dorothea Kolossa\(^1\)

\(^1\)Cognitive Signal Processing Group, Institute of Communication Acoustics, Ruhr-Universität Bochum, Germany
\(^2\)Mitsubishi Electric Research Laboratories, Cambridge, MA, USA
\(^3\)Inria, Villers-lès-Nancy, France

{ahmed.hussenabdelaziz, dorothea.kolossa}@rub.de, \{watanabe, hershey\}@merl.com, emmanuel.vincent@inria.fr

Abstract

In order to improve the ASR performance in noisy environments, distorted speech is typically pre-processed by a speech enhancement algorithm, which usually results in a speech estimate containing residual noise and distortion. We may also have some measures of uncertainty or variance of the estimate. Uncertainty decoding is a framework that utilizes this knowledge of uncertainty in the input features during acoustic model scoring. Such frameworks have been well explored for traditional probabilistic models, but their optimal use for deep neural network (DNN)-based ASR systems is not yet clear. In this paper, we study the propagation of observation uncertainties through the layers of a DNN-based acoustic model. Since this is intractable due to the nonlinearities of the DNN, we employ approximate propagation methods, including Monte Carlo sampling, the unscented transform, and the piecewise exponential approximation of the activation function, to estimate the distribution of acoustic scores. Unfortunately, computing the score distribution requires integrating the DNN over all input values, according to the observation distribution. However, due to the nonlinearities, this integral is intractable for DNNs and requires approximation. The problem of decoding is complicated by its dependency on the approximations that are used to estimate the score distribution. We therefore investigate both problems together. For estimating score distribution, we investigate Monte-Carlo methods, including Monte Carlo sampling, the unscented transform, and the piecewise exponential approximation of the activation function. For decoding, we investigate different methods of integrating over score distributions, where we take two expected values relative to a highly optimized DNN baseline.

Index Terms: Noise-robust ASR, Deep Neural Networks, Observation Uncertainty, Uncertainty Propagation

1. Introduction

Although statistical models like hidden Markov models (HMMs) have shown great success in modeling and recognizing the temporal evolution of the spectral characteristics of speech, wider use of automatic speech recognition (ASR) systems is still precluded by acoustic environmental effects like background noise and reverberation. In order to achieve an acceptable level of recognition robustness against these effects, the distorted speech is usually de-noised using speech enhancement algorithms \([1,2]\). However, the enhanced signals obtained from such algorithms are not perfectly compensated and often contain residual noise, estimation errors, and even artifacts introduced by the speech enhancement algorithms.

Modeling the spectral characteristics of speech signals using statistical models like GMMs has facilitated the development of a range of so-called probabilistic uncertainty-of-observation (UoO) techniques \([3–7]\). Such techniques take into account the residual noise and the estimation errors of the enhancement algorithms by considering the enhanced speech outputs as random variables rather than point estimates. In GMMs, one can easily take the uncertainty into account for Gaussian observations by marginalizing out the random observation variables. In DNN-based systems, in contrast, the observations are not explicitly modeled by statistical distributions. Although there are probabilistic interpretations of DNNs, here we consider standard sigmoid DNNs simply as deterministic nonlinear functions. To incorporate the uncertainty for Gaussian-distributed observations, we have to solve two problems: first, compute the distribution of acoustic scores by integrating out the input random variables, and second, incorporate the score distribution into the decoding algorithm once it has been estimated. Unfortunately, computing the score distribution requires integrating the DNN over all input values, according to the observation distribution. However, due to the nonlinearities, this integral is intractable for DNNs and requires approximation. The problem of decoding is complicated by its dependency on the approximations that are used to estimate the score distribution. We therefore investigate both problems together. For estimating score distribution, we investigate Monte-Carlo methods, including Monte Carlo sampling, the unscented transform, and the piecewise exponential approximation of the activation function. For decoding, we investigate different methods of integrating over score distributions, where we take two expected values relative to the posterior state probabilities.

In \([8]\), uncertainty propagation has been conducted layer by layer and for each node separately by approximating the nonlinear activation function using the so-called piece-wise exponential (PIE) approximation or by approximating the input Gaussian distribution using the unscented transform (UT). These approximations require the hidden layer pre-activations to be statistically independent. In this paper, we show by Monte Carlo simulation that this condition is not always true, especially for deep and wide DNNs. In order to minimize the estimation errors and to avoid the accumulation of the propagated errors from a hidden layer to another, we propose to use Monte Carlo sampling and the UT with the entire DNN. This allows the propagation of uncertainty even through the softmax layer, which is difficult using layer-wise approximation methods.

In \([8]\), the acoustic scores have been modified to match the observation uncertainties in a similar way as used in the uncertainty decoding approach \([4]\), which has first been introduced for GMM-based ASR systems. In this paper, we investigate

\(*\) This work was conducted while the first author was doing an internship at Mitsubishi Electric Research Laboratories.
an alternative approach to exploit the uncertainties for acous-
tic scoring by replacing the DNN pseudo log-likelihoods by
their conditional expectations given the enhanced features. This
new score is reminiscent of the GMM-based modified acoustic
scores in [9]. This approach can be used in conjunction with
the layer-wise uncertainty propagation methods as it does not
require propagation through the softmax layer.

The remaining paper is organized as follows: In Section 2,
Monte Carlo sampling, the unscented transform, and the PIE
approximation are described as possible approaches for uncer-
tainty propagation through DNNs. Next, in Section 3, the two
uncertainty-based acoustic scores are introduced. In Section 4,
all approaches are evaluated using the second track of the sec-
dard CHiME challenge [14]. Finally, in Section 5, the paper is
concluded and an outlook of future work is given.

2. Uncertainty Propagation through DNNs

DNN layers are composed of a linear operation followed by
a nonlinear operation. The typically used nonlinear function is
a sigmoid function in the hidden layers and a softmax function in
the output layer. The question to be addressed in this section is
the following: If the input to a DNN is a multivariate Gaussian
random variable, what is the distribution of the corresponding
output random variable after applying the linear and nonlinear
operations of all neural network layers?

2.1. Uncertainty Propagation through Entire DNN

2.1.1. Monte Carlo Sampling

Monte Carlo sampling is the simplest approach that can be used
to calculate the statistics of random variables that undergo a
nonlinear transformation. This method is based on randomly
drawing a number of samples from the distribution underlying the
random variable. The nonlinear transformation is then applied
to these samples. The first and second order statistics of the
nonlinearly transformed random variable can be estimated as
the mean and the variance of the output samples, respectively.

2.1.2. Unscented Transform

The UT is a similar method to the Monte Carlo approach. How-
ever, in the UT, the samples are not drawn randomly but ac-
cording to a specific criterion. For an \( I \)-dimensional random
variable, \( 2I + 1 \) sample vectors and their associated weights are
computed as introduced in [10]. The nonlinear function, here
the DNN, is then applied to these sample vectors. Using the esti-
minated weights, the first and second order statistics of the out-
put distribution are calculated as a weighted sum of the output
sample vectors and a weighted sum of the squared mean-free
samples, respectively.

2.2. Layer-Wise Uncertainty Propagation

Instead of propagating the uncertainty through the entire DNN
at once, the uncertainties can be propagated layer by layer.
Propagating a multi-variate Gaussian distribution through the
linear part of a neural network layer is simple, as another Gauss-
ian distribution is analytically obtained. On the other hand,
propagating the multi-variate Gaussian distribution through a
sigmoid function results in a very complex distribution as
shown in [11]. This has led the authors of [8] to use simpler
approximations like the PIE approximation [12] and the UT
to determine the first and second order statistics of this complex
distribution for every layer.

2.2.1. PIE Approximation

In the PIE approximation, a sigmoid function \( g(z) \) is approxi-
mated by a sum of two exponential functions as follows:

\[
g(z) = \frac{1}{1 + e^{-z}} \approx 2^{z-1} u(-z) + (1 - 2^{-(z-1)}) u(z),
\]

where \( u(z) \) is the unit step function. If \( z \) is a one-dimensional
Gaussian random variable with mean value \( \mu_z \) and standard
deviation \( \sigma_z \), the first and second order statistics of (1) can be
estimated as follows:

\[
E[g(z)] = 2^{\mu_z + 0.5 \log(2) \sigma_z^2 - 1} \phi \left( -\frac{\mu_z}{\sigma_z} - \log(2) \sigma_z \right)
- 2^{(-\mu_z + 0.5 \log(2) \sigma_z^2 - 1)} \left[ 1 - \phi \left( -\frac{\mu_z}{\sigma_z} + \log(2) \sigma_z \right) \right] + \left[ 1 - \phi \left( -\frac{\mu_z}{\sigma_z} \right) \right].
\]

(2)

\[
\text{Var}[g(z)] = \mathbb{E}[g(z)^2] - (\mathbb{E}[g(z)])^2,
\]

(3)

where

\[
\mathbb{E}[g(z)^2] = 2^{(2\mu_z + 2 \log(2) \sigma_z^2 - 2)} \phi \left( -\frac{\mu_z}{\sigma_z} - 2 \log(2) \sigma_z \right)
- 2^{(-\mu_z + 0.5 \log(2) \sigma_z^2)} \left[ 1 - \phi \left( -\frac{\mu_z}{\sigma_z} + \log(2) \sigma_z \right) \right]
+ 2^{(-2\mu_z + 2 \log(2) \sigma_z^2 - 2)} \left[ 1 - \phi \left( -\frac{\mu_z}{\sigma_z} + 2 \log(2) \sigma_z \right) \right] + 1 - \phi \left( -\frac{\mu_z}{\sigma_z} \right).
\]

(4)

In (2) and (4), \( \phi \) is the cumulative density function of the stan-
dard normal distribution.

2.3. Discussion

For very wide and deep DNNs, propagating a Gaussian distri-
bution through the sigmoid function of a hidden layer using the
UT or the PIE approximation is computationally very expen-
sive. For example, deploying the UT needs 4097 vectors and
their associated weights to be computed, where the dimension of
the hidden layers used in this study is \( I = 2048 \). It is also dif-
ficult to use the PIE approximation, since the computation of the
cumulative density function for multi-variate Gaussian distribu-
tions of such a large dimension is not trivial when the covariance
matrix is not diagonal. Therefore, the off-diagonal components of
the covariance matrix of the hidden layer pre-activations have
been neglected in [8] assuming a weak correlation between their
components. Based on this assumption, the PIE approximation
and the UT have been applied for each neuron separately.

The weak correlation assumption demands a diagonal covari-
ance matrix of the the pre-activations, which can be computed for
the \( n \)-th layer via

\[
[\Sigma_w^n]_{i,j} = \sum_{k=1}^{I} \sum_{l'=1}^{I} [W^n]_{i,k} [W^n]_{j,k'} [\Sigma_{h^{(n-1)}}]_{k,k'}.
\]

(5)

In (5), \( W^n \) and \( \Sigma_{h^{(n-1)}} \) are the weight matrix of the \( n \)-th hid-
den layer and the covariance matrix of the preceding hidden
layer, respectively. From (5), the covariance matrix \( \Sigma_w^n \) be-
comes diagonal if the weight matrix \( W^n \) and the covariance
In order to reduce the estimation errors and avoid their propagation through the DNN, Monte Carlo sampling can be used. In [8], another modified score has been obtained by replacing the log-likelihood in (8) by the pseudo log-likelihood acoustic score, which can be estimated via

\[
\log (p(Y|q_i)) = \log \left( \frac{p(Y|X)p(X)}{p(Y)} \right) - \log \left( \sum_{j=1}^{I} \exp \left( z_j^{L+1} \right) \right). 
\]

(7)

In (6), \( z_i^{L+1} \) is the \( i \)-th pre-activation of the DNN output layer, where \( i \in \{1, \ldots, I\} \), and \( p(q_i) \) is the prior probability of the \( i \)-th state \( q_i \). There are actually two missing terms in (6) that distinguish \( L_{\text{pseudo}} \) from the conventional log-likelihood acoustic score, which can be estimated via

\[
\log (p(X|q_i)) = \log \left( \frac{p(q_i|X)p(X)}{p(q_i)} \right) - \log \left( \sum_{j=1}^{I} \exp \left( z_j^{L+1} \right) \right). 
\]

(8)

In order to estimate the expectation \( \mathbb{E}[z_i^{L+1}|Y] \), the posterior \( p(Y|X) \) should first be propagated up to the pre-activations of the output layer. The mean value of the propagated distribution can then be deployed in (10) as an estimate of \( \mathbb{E}[z_i^{L+1}|Y] \). In [8], another modified score has been obtained by replacing the clean posterior \( p(q_i|X) \) by the enhanced posterior \( p(q_i|Y) \), which can be found by integrating the joint probability \( p(q_i, X|Y) \) over the space of the clean features as follows:

\[
p(q_i|Y) = \int_X p(q_i, X|Y) dX 
\]

(11)

\[
= \mathbb{E}[p(q_i|X)|Y] = \mathbb{E}[h_i^{L+1}|Y]. 
\]

(12)

where \( h_i^{L+1} \) is the output of the softmax layer. The transition from (11) to (12) assumes statistical independence of the state \( q_i \) and the enhanced features \( Y \) given the clean features \( X \). Using the enhanced posterior in (12), the acoustic score can be estimated as follows:

\[
L_{\text{OU2}} = \log (p(q_i|Y)) - \log (p(q_i)). 
\]

(13)
It can be seen in (12) that the posterior \( p(X|Y) \) should be propagated to the DNN output layer in order to estimate the enhanced posteriors \( p(q_i|Y) \) and hence, the new acoustic score \( L_{\text{PIE}}(\text{Baseline}) \). On the other hand, the modified acoustic score in (10) does not need the posterior \( p(X|Y) \) to be propagated through the softmax function of the output layer, which makes it suitable for the layer-wise uncertainty propagation approaches.

### 4. Experiments and Results

#### 4.1. Dataset

Track 2 of the 2nd CHiME Challenge [14] has been used for evaluation. The task is to recognize English sentences read by different male and female speakers taken from the medium vocabulary subset (5,000 words) of the Wall Street Journal (WSJ0) corpus [15]. The training dataset contains 7138 noisy utterances spoken by 84 speakers. The development and the test dataset contain 2454 and 1980 noisy utterances, respectively. The noisy utterances have been created by first convolving the WSJ0 clean utterances with binaural room impulse responses (BRIRs) and then adding background noise signals at six different SNRs: -6, -3, 0, 3, 6, and 9 dB. The BRIRs and the background noise signals have been recorded in a domestic living room using a head and torso simulator (HATS).

#### 4.2. Experimental Setup

The baseline ASR system has been trained as follows [16, 17]. The DNN target state posteriors have been estimated using pre-trained triphone GMM models. Training of the GMM models has been done using the clean signals underlying the noisy training and development utterances. The features used for training are the 13 MFCC features with their corresponding \( \Delta \) and \( \Delta \Delta \) features. The MFCC features have been post-processed using linear discriminant analysis (LDA) [18, 19], maximum likelihood linear regression (MLLR) [20, 21], and speaker adaptive training (SAT) [22].

The DNN input features have been computed as follows. First, the noisy signals have been enhanced using a multichannel NMF pre-processor [23, 24]. From the enhanced signals, 40-dimensional Mel feature vectors have been obtained. Finally, 11 frames (5 previous frames, current frame, and 5 following frames) have been appended to form the DNN input features.

The DNN is composed of a 440-dimensional input layer, seven 2048-dimensional hidden layers, and an output layer of 2004 nodes representing the HMM states. The parameters of the hidden layers have been initialized using restricted Boltzmann machine (RBM) pre-training. Finally, all parameters have been fine-tuned using the back-propagation algorithm [25].

The ASR performance has been evaluated in terms of the word error rate (WER). Training and decoding have been conducted using the Kaldi speech recognition toolkit [26]. The modified acoustic scores (10) and (13) and the uncertainty propagation approaches have been applied using the DNN uncertainty propagation toolbox [27].

#### 4.3. Results

The ground-truth results have been obtained using a Monte Carlo simulation, which has been conducted as follows. Each 440-dimensional DNN input feature vector has been considered as the mean vector of the Gaussian distribution \( p(X|Y) \). Similarly to [28], the covariance matrix of \( p(X|Y) \) has been assumed to be diagonal with diagonal entries defined as the squared difference between the corresponding components of the noisy and the enhanced features weighted by a dimension-, state-, and SNR-independent constant \( \eta \). A grid search with minimum WER criterion has been conducted using the development set to find the appropriate values of \( \eta \). \( \eta = 0.3 \) and \( \eta = 0.4 \) have achieved the best results using the acoustic scores (10) and (13), respectively. The same estimates of the mean vector and the covariance matrix of \( p(X|Y) \) have also been used with the PIE approximation, the layer-wise UT, and the entire-DNN UT. From the Gaussian distribution \( p(X|Y) \), 50 vectors have been sampled and applied to the DNN. The sample mean of the 50 corresponding DNN output vectors \( h^{\text{L}+\text{1}} \) and their pre-activations \( z^{\text{L}+\text{1}} \) have been considered as estimates of the mean vectors \( \mathbb{E}[h^{\text{L}+\text{1}}|Y] \) and \( \mathbb{E}[z^{\text{L}+\text{1}}|Y] \).

For the sake of simplicity and in order to make the entire-DNN UT more practical, we have used just three samples instead of the \((2I + 1)\) required samples. The three samples are simply the mean vector of the posterior \( p(X|Y) \) and the mean vector plus/minus \( \sqrt{3} \) times the covariance matrix diagonal.

In Table 1, we compare the recognition results of the development and test set obtained using the conventional pseudo log-likelihood \( L_{\text{pseudo}}(\text{Baseline}) \) and the uncertainty-based acoustic scores \( L_{\text{OU1}}(\text{OU1}) \) and \( L_{\text{OU2}}(\text{OU2}) \). Since the PIE approximation and the layer-wise UT can only be used for propagating the uncertainty to the output layer pre-activations, these approaches can only be tested with the acoustic score \( L_{\text{OU1}} \). As can be seen, the best results are achieved using Monte Carlo sampling. It can also be noticed that the results obtained using the acoustic score \( L_{\text{OU2}} \) are better than those results achieved using \( L_{\text{OU1}} \). Despite the small number of samples used in the entire-DNN UT, it gives better results than those obtained using the PIE approximation and the unscented transform, while approaching the results achieved using Monte Carlo sampling.

#### 5. Conclusions

In this paper, four possible approaches of uncertainty propagation through DNNs have been investigated: Monte Carlo sampling, entire-DNN unscented transform, PIE approximation, and layer-wise unscented transform. The propagated uncertainties have been deployed in the DNN decoding procedure using two modified acoustic scores. The best results have been achieved using Monte Carlo sampling for uncertainty propagation and the second modified acoustic score, which is similar to the uncertainty decoding approach originally proposed for GMMs in [9]. As an alternative practical approach with a greatly reduced computational effort, the entire-DNN UT can also be used for uncertainty propagation.

This work can be extended by improving the uncertainty estimation using dynamic SNR-dependent weighting constants \( \eta \) instead of the static SNR-independent ones used in this study.

---

**Table 1: WER results obtained using different acoustic scores and different uncertainty propagation (UP) approaches.**

<table>
<thead>
<tr>
<th>Acoustic Score</th>
<th>UP</th>
<th>Dev.</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L_{\text{OU2}}(\text{Baseline}) )</td>
<td>—</td>
<td>27.59</td>
<td>21.67</td>
</tr>
<tr>
<td>( L_{\text{OU1}}(\text{Baseline}) )</td>
<td>PIE Layer-wise UT Entire-DNN UT</td>
<td>31.80</td>
<td>25.85</td>
</tr>
<tr>
<td></td>
<td>Entire-DNN UT</td>
<td>26.88</td>
<td>21.23</td>
</tr>
</tbody>
</table>
6. References


