Uncertainty Quantification using Variational Inference for Biomedical Image Segmentation

Abhinav Sagar∗
Vellore Institute of Technology
Vellore, Tamil Nadu, India
abhinavsagar4@gmail.com

Abstract

Deep learning motivated by convolutional neural networks has been highly successful in a range of medical imaging problems like image classification, image segmentation, image synthesis etc. However for validation and interpretability, not only do we need the predictions made from the model but also how confident it is while making those predictions. This is important in safety critical applications for the people to accept it. In this work, we used an encoder decoder architecture based on variational inference techniques for segmenting brain tumour images. We compared different backbones architectures like U-Net, V-Net and FCN as sampling data from the conditional distribution for the encoder. We validated our work on BRATS dataset using Dice Similarity Coefficient and Intersection Over Union as the evaluation metrics. Our model achieves state of the art results while making use of a principled way of uncertainty quantification.

1 Introduction

Medical image segmentation is a challenging task for medical practitioners. It is costly, takes time and is prone to error. Hence there is a need to automate the manually done segmentation. Lately Neural Networks have shown great potential on a variety of medical image segmentation problems. The challenge with the approaches used in literature is that it doesn’t predict the uncertainty associated with the model predictions. This is where Bayesian methods come into play as it gives a principled way of measuring uncertainty from the model predictions. Measuring uncertainty in the output predictions made by neural networks is important for interpretation and validation. Rather than learning the point estimates, Bayesian Neural Networks (BNN) learns the distribution over the weights. The training process of BNN involves first initializing the parameters of the neural network. Next the weights are sampled from some distribution (like gaussian with zero mean and unit variance) and both the forward pass and backward pass is done to update the weights using the conventional backpropagation algorithm.

Monte Carlo dropout networks (Kingma et al., 2015) use dropout layers to approximate deep Gaussian processes which still lack theoretical understanding. Bayesian Convolutional Neural Network (Gal et al., 2015) use variational inference to learn the posterior distribution over the weights given the dataset. The problem with this approach is that it requires a lot of computation involving a lot of parameters, making this technique not scalable in practice.

Variational Autoencoder (Kingma and Welling, 2014) which is based on generative models solves the above problems and has been successful in a number of tasks like generating images, texts, recommender systems etc. This approach comes with several challenges in its own right which have been successfully tackled in the literature. A random variable sampled from posterior distribution has

∗Website of author - [https://abhinavsagar.github.io/](https://abhinavsagar.github.io/)

Preprint. Under review.
no gradient so the conventional backpropagation techniques can’t be applied to it. Local Reparameter-
ization Trick (Kingma et al., 2015) was proposed to tackle this by converting the random variable to
a deterministic one for computation. The second challenge was the huge computational requirement
since it required weight updates in every iteration. Bayes by Backprop algorithm (Blundell et al.,
2015) tackled this by calculating gradients in back-propagation using a scale and shift approach by
updating the posterior distribution in the backward pass.

2 Related Work

2.1 Medical Image Segmentation

The problem of segmenting medical images have been successfully tackled in literature using mainly
two techniques, first using a Fully Convolutional Network (FCN) (Long et al., 2014) and second
those which are based on U-Net (Ronneberger et al., 2015). The main characteristic of FCN
architectures is that it doesn’t use fully connected layers at the end which have been used successfully
for image classification problems. U-Net on the other hand uses an encoder-decoder architecture
with pooling layers in the encoder and upsampling layers in the decoder. Skip connections connect
the encoder layers to the decoder layer to create an additional path for the flow of gradients back in
the backpropagation step. This helps in reducing overfitting due to many parameters involved while
training the network.

2.2 Bayesian Neural Network

Lately, there has been a revival of interest in bayesian methods as some of the inherent problems
with deep learning could be solved using it. It is a scalable approach of avoiding overfitting in
neural networks and at the same time gives us a measure of uncertainty. This is very important in
critical applications where not only do we require the predictions made from the model, but also
how confident it is while making its predictions. BNN can be considered as an ensemble of neural
networks (Gal et al., 2016). It has two advantages over the standard neural networks, first it avoids
overfitting and second it gives a measure of uncertainty involved.

Instead of point estimates, the neural network learns posterior distribution over the weights given the
dataset as given in the equation below.

\[
p(\omega|D) = \frac{p(D|\omega) p(\omega)}{p(D)} = \frac{\prod_{i=1}^{N} p(y_i|x_i, \omega) p(\omega)}{p(D)} \tag{1}
\]

The predictive distribution can be calculated by approximating the integral as shown in equation
below.

\[
p(y^*|x^*, D) = \int_{\Omega} p(y^*|x^*, \omega) p(\omega|D) d\omega \tag{2}
\]

The challenge is that the posterior is often intractable in nature. To combat this, (Neal et al., 1993)
used Markov Chain Monte Carlo (MCMC) for learning the weights over the bayesian neural networks.
Also [Graves, 2011, Blundell et al., 2015, Louizos and Welling, 2016, 2017a] proposed independently
a technique using variational inference techniques for approximating the posterior distributions. KL
Divergence between the posterior and the true distribution can be calculated using the equation below.

\[
KL \{q_\theta(\omega)\|p(\omega|D)\} := \int_{\Omega} q_\theta(\omega) \log \frac{q_\theta(\omega)}{p(\omega|D)} d\omega \tag{3}
\]

Alternatively minimizing the KL divergence can be written in another form by maximizing the
Evidence Lower Bound (ELBO) which is tractable. This is shown in the equation below.

\[
-\int_{\Omega} q_\theta(\omega) \log p(y|x, \omega)d\omega + KL \{q_\theta(\omega)\|p(\omega)\} \tag{4}
\]
2.3 Variational Inference

Variational inference finds the parameters of the distribution by maximizing the Evidence Lower Bound. ELBO consists of sum of two terms Kullback-Leibler (KL) divergence between two distributions and the negative log-likelihood (NLL). The KL divergence term which has to be minimized is shown in the equation below.

$$\min_{\theta} \text{KL}(q_{\theta}(w) \| p(w|D))$$

(5)

The KL divergence is defined as shown in the equation below.

$$\text{KL}(q(x) \| p(x)) = - \int q(x) \log \frac{p(x)}{q(x)}$$

(6)

The posterior in the above equation contains an integral which is intractable in nature. The equation can be re written as shown in the equation below.

$$\text{KL}(q_{\theta}(w) \| p(w|D)) = \mathbb{E}_{q_{\theta}(w)} \log \frac{q_{\theta}(w) p(D)}{p(D|w)p(w)} =$$

$$= \log p(D) + \mathbb{E}_{q_{\theta}(w)} \log \frac{q_{\theta}(w)}{p(w)} - \mathbb{E}_{q_{\theta}(w)} \log p(D|w)$$

$$= \log p(D) - \mathcal{L}(\theta)$$

(7)

It can be decomposed into two parts one of which is the KL divergence between the exact posterior and its variational approximation which needs to be minimized and the second is ELBO which needs to be maximized. This is shown in the equation below.

$$\max_{\theta} \log p(D) = \max_{\theta} \left[ \text{KL}(q_{\theta}(w) \| p(w|D)) + \mathcal{L}(\theta) \right]$$

(8)

KL divergence is zero if exact posterior is equal to variational approximation. Since the KL divergence is always greater than zero hence the equation can be approximated by maximizing only the ELBO which needs to be maximized. (Welling et al., 2015) as shown in equation below.

$$\mathcal{L}(\theta) = \mathbb{E}_{q_{\theta}(w)} \log p(D|w) - \mathbb{E}_{q_{\theta}(w)} \log \frac{q_{\theta}(w)}{p(w)} = \mathcal{L}_{D} - \text{KL}(q_{\theta}(w) \| p(w))$$

(9)

2.4 Aleatoric uncertainty and epistemic uncertainty

There are two types of uncertainty - aleatory and epistemic uncertainty where variance is the sum of both these. Bayesian Neural Networks can be considered an ensemble of neural networks initialized randomly which averages the test results in parallel (Gal et al., 2016). For final predictions, single mean and variance can be estimated as shown in the equation below.

$$\mu_c(x) = \frac{1}{M} \sum_{i=1}^{M} \hat{\mu}_i(x)$$

(10)

$$\hat{\sigma}_c^2(x) = \frac{1}{M} \sum_{i=1}^{M} \hat{\sigma}_i^2(x) + \left[ \frac{1}{M} \sum_{i=1}^{M} \hat{\mu}_i^2(x) - \hat{\mu}_c^2(x) \right]$$

(11)

The first term in variance denotes aleatoric uncertainty while the second denotes epistemic uncertainty. Bayesian Neural Network model for uncertainty estimation was done by (Kendall et al., 2017) with
the last layer representing the mean and variance of logits. The predictive distribution approximating
the posterior distribution which gives a measure of uncertainty is shown in the equation below.

\[ q_\theta(y^*|x^*) = \int_\Omega p(y^*|x^*, \omega) q_\theta(\omega) d\omega \quad (12) \]

Aleatoric uncertainty is a measure of the variability of the predictions from the dataset hence it is
inherent in the data present. Epistemic uncertainty on the other hand is a measure of the variability of
predictions from the model which is tied to various metrics used for evaluation like accuracy, loss etc.

### 3 Proposed method

The prior distribution helps to incorporate learning of the weights over the network. Variational
Autoencoder has been used successively as a kind of generative model by sampling from the prior
distribution in the encoder. The decoder uses the mean vector and standard deviation vector from the
latent space to reconstruct the input.

Our model uses a similar encoder decoder architecture as that used in VAEs with the input to the
encoder coming from a pre trained image segmentation architecture. We tried different backbones
which have enjoyed success and found original U-Net gave the best results. The input to the
encoder only needs the mean the standard deviation vectors of the conditional distribution expressing
the confidence with which the pixels are correctly predicted. After passing through the encoder,
the parameters get converted to a latent representation which is again sampled in a mean and
standard deviation vector. The decoder later recovers this back to the original distribution. The
conventional backpropagation algorithm is used for training the model with gradient descent. The
model architecture is shown in Fig 1.

![Figure 1: Our model architecture](image)

Our model uses a similar approach which is trained using Gradient Descent is presented below:

**Algorithm 1:** Uncertainty Quantification using Variational Inference for Biomedical Image
Segmentation

| Input: Dataset \( D = \{(x_i, y_i)\}_{i=1}^N \) |
| Input: Variational approximation of the posterior distribution \( q_\theta(w) \) |
| Input: encoder \( r_\psi(z|w) \) and decoder \( p_\phi(w) \) |
| **while not converged do** |
| Sample minibatch: \( D^* = \{(x_i, y_i)\}_{i=1}^M \) |
| Sample weights with reparametrization: \( \tilde{w}^{(i)} \sim q_\theta(w^{(i)}) \) |
| Sample latent variables with reparametrization: \( z^{(i)} \sim r_\psi(w^{(i)}) \) |
| Compute stochastic gradients of the objective: \( \mathcal{L}_{\text{approx}} = \mathcal{L} + \sum_i [-\log q_\theta(\tilde{w}^{(i)})] - \log r_\psi(z^{(i)}|\tilde{w}^{(i)}) + \log p(z) + \log p_\phi(z^{(i)}|z) + \log p_\phi(x|z) \) |
| Update parameters \( \theta = \theta + \alpha \nabla_\theta \mathcal{L} \) and \( \psi = \psi + \beta \nabla_\psi \mathcal{L} \) |
| \( g_\phi \leftarrow \frac{1}{m} \sum_{k=1}^m \nabla_\phi \log p_\phi(x^{(k)}|z_\theta(x^{(k)}, \epsilon^{(k)})) \) |
| **end** |

Output: \( q_\theta(w) \) — posterior distribution of the model parameters
3.1 Datasets

To validate the performance of our proposed approach to generalization, publicly available datasets were used for brain tumour segmentation BRATS18 (Menze et al., 2015, Bakas et al., 2017). It contains MRI scans of 175 patients with glioblastoma and lower grade glioblastoma. The images were of resolution 240 * 240 * 155 pixels. The ground truth labels were created by expert neuroradiologists. The sample from the dataset is shown in Fig 2.

![Figure 2: Example of MRI slices and ground truth segmentation](image)

3.2 Hyperparameters

The hyperparameters used in our model are specified in Table 1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch Size</td>
<td>16</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adam</td>
</tr>
<tr>
<td>Learning Rate</td>
<td>0.001</td>
</tr>
<tr>
<td>Latent Variable Size</td>
<td>10</td>
</tr>
</tbody>
</table>

In addition to the above hyperparameters, cyclical learning rate schedulers and ReduceLROnPlateau was used. In gradient descent, the value of momentum was taken as 0.9, gamma value of 0.1 and weight decay of 0.0005.

3.3 Evaluation

Evaluation metrics for semantic segmentation problems which have often been used in literature are Dice Similarity Coefficient (DSC) also known as F1-score and Intersection over union (IoU) as proposed by (Clérigues et al., 2018, Kao et al., 2018, Myronenko, 2018, Deniz et al., 2018). The corresponding equations are shown below.

\[
\text{DSC} = \frac{2TP}{2TP + FN + FP} \quad (13)
\]

\[
\text{IoU} = \frac{TP}{TP + FN + FP} \quad (14)
\]

True positive (TR), false negative (FN) and false positive (FP) number of pixels is calculated separately for each image and averaged over the test set. The ground truth is labelled manually by experts which are compared against.

3.4 Loss

A combination of binary cross entropy and dice losses have been used to train the network. The first part binary cross entropy is a commonly used loss function for classification problems as shown by
Every pixel in the image needs to be classified and hence loss function can be written as shown in the equation below.

$$\mathcal{L}_{CE} = -\sum_{i,j} y_{i,j} \log \hat{y}_{i,j} + (1 - y_{i,j}) \log (1 - \hat{y}_{i,j})$$  \hspace{1cm} (15)$$

The problem with binary cross entropy loss is that it doesn’t take into account the class imbalance as the background is the dominant class. This is one of fundamental challenges in semantic segmentation problems. Dice Loss is robust to the aforementioned problem which is based on Dice Similarity Coefficient as defined below.

$$\mathcal{L}_{DICE} = \frac{\sum_{i=1}^{N} FN_i + FP_i}{2TP_i + FN_i + FP_i} = \sum_{i=1}^{N} \left(1 - DSC^{(i)}\right)$$  \hspace{1cm} (16)$$

Both the loss terms were combined in a single term with more weight given to the Dice Loss term since it handles the class imbalance problem better. This is defined using the equation below.

$$\mathcal{L} = 0.9 \cdot \mathcal{L}_{DICE} + 0.1 \cdot \mathcal{L}_{CE}$$  \hspace{1cm} (17)$$

### 4 Results

The Mean Dice Similarity value for various backbone architectures compared against different train size values are shown in Table 2.

<table>
<thead>
<tr>
<th>Train Size</th>
<th>UNet</th>
<th>VNet</th>
<th>FCN</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>53.1</td>
<td>50.6</td>
<td>50.2</td>
</tr>
<tr>
<td>10</td>
<td>56.6</td>
<td>51.3</td>
<td>52.1</td>
</tr>
<tr>
<td>15</td>
<td>60.8</td>
<td>53.8</td>
<td>54.4</td>
</tr>
<tr>
<td>20</td>
<td>64.3</td>
<td>56.5</td>
<td>58.9</td>
</tr>
</tbody>
</table>

The IOU value for various backbone architectures compared against different train size values are shown in Table 3.

<table>
<thead>
<tr>
<th>Train Size</th>
<th>UNet</th>
<th>VNet</th>
<th>FCN</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>48.4</td>
<td>46.7</td>
<td>47.2</td>
</tr>
<tr>
<td>10</td>
<td>50.6</td>
<td>48.8</td>
<td>50.4</td>
</tr>
<tr>
<td>15</td>
<td>53.1</td>
<td>50.8</td>
<td>52.6</td>
</tr>
<tr>
<td>20</td>
<td>55.8</td>
<td>52.8</td>
<td>54.3</td>
</tr>
</tbody>
</table>

The results of uncertainty involved in segmentation is shown in Fig 3.
Conclusions

In this work, we presented a way to quantify uncertainty in the context of medical image segmentation. Our model is based on an encoder decoder framework similar to that used by VAEs. The weights of the network represent distributions instead of point estimates and thus give a principled way of measuring uncertainty at the same time while making the predictions. Our model uses bayesian neural networks for both the encoder and decoder. The inputs to encoder come from backbones like U-Net, V-Net, FCN sampled from conditional distribution representing the confidence with which pixels are labelled correctly. We validated our results on publicly available BRATS dataset with our model achieving state of the art results on DSC and IOU metrics.

Acknowledgments

We would like to thank Dhruv Raj and Shreya Singh for their helpful comments on this work. We would also like to thank Nvidia for providing the GPUs.

References


