Deep Probabilistic Neural Networks for Geoscience

Introduction

Imaging of faults within seismic datasets is a key step in hydrocarbon exploration, carbon capture and sequestration, and geothermal applications. Incorporating predictive models into probabilistic decision making workflows such as reserve estimation requires well-calibrated models that are not only able to make good predictions but which also allow us to associate calibrated probabilities to their predictions. As an example, while Deep Convolutional Neural Networks (CNNs) have become the state-of-the-art for making high-quality predictions of fault probability cubes from 3D seismic data (Zhang et al., 2019), it has been shown that CNNs can have poorly calibrated outputs that may over- or under-estimate the probability of an event (e.g. Guo et al. 2017). This is an inherent shortcoming and can occur in other geoscience applications where CNNs are being deployed thus making their incorporation into probabilistic decision making workflows possibly unreliable. We present a comparison between three different approaches to produce calibrated predictive models following a Bayesian interpretation of training neural networks (Mosser et al., 2020). Although the example shown here is for detecting faults in seismic data, the methods described are equally applicable to other applications.

Figure 1 The posterior of the CNN weights given the training data is multi-modal. When marginalizing to obtain the posterior-predictive distribution, different methods represent different regions of the posterior. Deep Ensembles only consider MAP solutions close to single modes, while techniques such as Stochastic Weight Averaging-Gaussian or Concrete Dropout explore the vicinity of a single mode. Figure after Wilson and Izmailov (2020).

Methodology

A key difference between traditional neural network training and a Bayesian approach to obtaining probabilistic models is the process of marginalization. For a test image $x^*$ we aim to characterize the predictive distribution over the possible labels $y^*$ (e.g. fault or non-fault in the example in this paper). Our probabilistic model is a function of the training images $x$, and their respective labels $y$ which form the training dataset $D$, as well as the model weights $\omega$. To obtain the posterior-predictive distribution, we weight the likelihood of the labels given the test image by the posterior probability of the weights given the training data, and integrate over the entire space of weights $\Omega$

$$p(y^*|x^*, D) = \int_{\Omega} p(y^*|x^*, \omega)p(\omega|x^*, D)d\omega$$  

(1)

Our probabilistic model is a CNN with millions of parameters (Mosser et al., 2020), thus integrating over such a high-dimensional space is intractable. We therefore have to resort to approximate methods to characterize the predictive distribution (Eq. 1).
A simplistic characterization of the predictive distribution (Fig. 1 blue) can be obtained by first training multiple models with different initial weights and biases to obtain individual maximum a-posteriori estimates (MAP) of the weights given the training data $\mathcal{D}$ and subsequently averaging the predictions made by each individual model $i$

$$\mathbb{E}[p(y^*|x^*, \mathcal{D})] = \frac{1}{M} \sum_{i=1}^{M} p_i(y^*|x^*, \mathcal{D})$$  \hspace{1cm} (2)

This is known as a Deep Ensemble (Lakshminarayanan et al., 2017) where in our case we train three CNNs ($M = 3$) with different weight initializations. We apply a binary cross-entropy loss and $L_2$ regularization to represent a MAP objective function that honors the likelihood and a Gaussian weight prior. While Deep Ensembles allow us to consider different modes of the posterior, they come at a high computational cost, since each ensemble member requires training a CNN to full convergence.

To alleviate this computational cost, Gal and Ghahramani (2016) propose to use Stochastic Variational Inference (SVI) (Hoffman et al., 2013) to characterize the posterior distribution of a Bayesian neural network. Representing the true posterior by a simpler so-called variational distribution $\omega \sim q_\theta(\omega)$ and minimizing the Kullback-Leibler (KL) divergence between the variational distribution and the true posterior

$$\arg\min_{\theta} \text{KL} \{q_\theta(\omega)||p(\omega|\mathcal{D})\}$$  \hspace{1cm} (3)

we obtain an approximate representation of the posterior weight distribution. Such minimization can be formulated as an optimization problem which in the case of CNNs, can be solved by mini-batch stochastic-gradient-descent (SGD). They showed that applying Dropout regularization at test-time is equivalent to an approximate SVI approach. The variational distribution in this case is the Dropout distribution, a combination of weights sampled from a Gaussian prior distribution, which are multiplied by random samples from a Bernoulli-distribution. To obtain a well-calibrated predictive model one should perform a grid-search over the optimal values of the dropout probabilities $p_i$.

To minimize computational cost, Gal et al., (2017) propose to use a continuous relaxation of the Bernoulli distribution known as the Concrete distribution, to automatically optimize for the optimal dropout probabilities in the training process of the Bayesian neural network. The Concrete distribution reparameterizes a random variable sampled from a uniform distribution and depends on a parameter $t$ which determines probability mass allocation to its end-members $[0,1]$. Gradients with respect to the dropout probability $p_i$ can be obtained by automatic differentiation.

$$q(\omega) \sim W_t \cdot \text{diag}(\{z_{i,j}\}_{i=1}^{K_i}), \text{ for } i = 1, \ldots, L \text{ and } j = 1, \ldots, K_i$$  \hspace{1cm} (4a)

$$z_{i,j} = \sigma \left( \frac{1}{t} \cdot (\log p_i - \log (1 - p_i) + \log u - \log (1 - u)) \right), \text{ where } u \sim \text{Uniform}(0,1)$$  \hspace{1cm} (4b)

where $K_i$ is the number of kernels in layer $i$ of a neural network with $L$ layers, and $\sigma(\cdot)$ is the sigmoid function. We train a CNN with Concrete dropout, where convolutional kernels are randomly zeroed out, with initial Dropout probabilities randomly sampled from a uniform distribution $p_{init} \sim \text{Uniform}(0,0.5)$ and a MAP objective to minimize Equation (3). After the parameters of the Bayesian neural network have been obtained, we characterize the predictive distribution by a Monte-Carlo approximation of the mean using samples $\omega_t \sim q_\theta(\omega)$

$$p(y^*|x^t) = \int_\Omega p(y^*|x^t, \omega)q_\theta(\omega)d\omega \approx \frac{1}{T} \sum_{t=1}^{T} p(y^*|x^t, \omega_t), \text{ where } \omega_t \sim q_\theta(\omega)$$  \hspace{1cm} (5)

which is equivalent to predicting $t$-times on the same input test image $x^t$ with Dropout. We typically average $t = 30$ predictions for synthetic and $t = 10$ predictions for real datasets. Compared to the Deep Ensemble, the Concrete Dropout method induces changes to weights of the neural network while functionally maximizing a MAP objective. Therefore we assume that Concrete Dropout explores the vicinity of a single mode of the posterior weight distribution (Fig. 1 orange).
Finally, we consider a method that explores a connection between the optimization of the weights of a neural network through SGD and stochastic random processes to characterize the posterior weight distribution. Mandt et al. (2017) show that under certain assumptions, mini-batch SGD is equivalent to an Ornstein-Uhlenbeck (OU) process, a type of stochastic partial differential equation. The analytical stationary solutions of this stochastic process take on the form of Gaussian probability distributions

$$q(\theta) \propto \exp\{-\theta^T \Sigma^{-1} - \theta\} \tag{6}$$

Maddox et al. (2019) use this relationship to the OU-process to justify approximating the distribution of iterates of SGD-based training of a CNN with a low-rank multivariate Gaussian distribution

$$\tilde{\theta} \sim \mathcal{N}\left(\theta_{SWA}, \frac{1}{2}(\Sigma_{\text{diag}} + \Sigma_{\text{low}})\right) \tag{7}$$

where the covariance $\Sigma$ is split into a diagonal $\Sigma_{\text{diag}}$ and a low-rank approximation $\Sigma_{\text{low}}$ of rank $K$ which is computed using only the last $K$ SGD-iterates. Their approach is called Stochastic Weight Averaging - Gaussian (SWA-G), since we determine the mean of the Gaussian by averaging the CNN weights from subsequent SGD-iterates. This is different to Deep Ensembles where the mean is formed over the output predictions of the ensemble members. After training we sample from the approximate posterior distribution (Eq. 7) of the weights to obtain a Monte-Carlo estimate of the mean of the predictive distribution. We sample $t = 30$ sets of weights for synthetic and $t = 10$ samples of weights for real seismic data. Following the training process outlined by Maddox et al. (2019), we use SGD with momentum and $L_2$ regularization, with an initial learning rate $lr_{\text{init}} = 10^{-2}$ which is linearly decayed at the end of training to the sampling learning rate $lr_{SWA} = 10^{-3}$ during which we form the Gaussian approximation (Eq. 7) from $K = 20$ SGD-iterates. The approximate posterior obtained by SWA-G is represented by samples from a multivariate Gaussian in the vicinity of a single mode of the weight posterior due to the MAP objective and the theoretical connection to the OU-process (Fig. 1-red).

To evaluate the three methods proposed, we apply them on a fault imaging exercise where the synthetic training, validation, and test datasets were created based on the technique presented by Zhang et al. (2019) and Mosser et al. (2020).

### Results

To evaluate the resulting probabilistic neural networks we propose three criteria as measures of the quality of the uncertainty representation and model calibration which we compute on a held out test set of 400 synthetic seismic images: the negative log-likelihood (NLL), equivalent in our case to the binary cross-entropy loss, the Brier-score, and the expected calibration error (ECE). Additionally we evaluate the quality of the predicted fault masks by computing the Intersection over Union (IoU). As can be observed in Table 1, while the Deep Ensemble performs best for many metrics, it comes at a very high computational cost. The SWA-G approach shows the lowest NLL, second best Brier and ECE score, and is on par with Concrete Dropout in terms of IoU while at a much lower computational cost.

![Table 1](image)

<table>
<thead>
<tr>
<th>Method</th>
<th>NLL ↓</th>
<th>Brier Score ↓</th>
<th>ECE ↓</th>
<th>IoU ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep Ensemble</td>
<td>2.85e-03</td>
<td>3.28e-02</td>
<td>9.01e-03</td>
<td>0.55</td>
</tr>
<tr>
<td>Concrete Dropout</td>
<td>3.73e-03</td>
<td>3.65e-02</td>
<td>9.95e-03</td>
<td>0.51</td>
</tr>
<tr>
<td>SWA-Gaussian</td>
<td>2.74e-03</td>
<td>3.53e-02</td>
<td>9.66e-03</td>
<td>0.51</td>
</tr>
</tbody>
</table>

Table 1 Model performance on test dataset of synthetic seismic images. Best result in bold.

We evaluate the pixel-wise total, aleatoric, and epistemic uncertainties (Kendall and Gal 2017) on a synthetic dataset (Fig. 2) where all three methods are able to identify the fault within the seismic image but with low probabilities and high uncertainties at the fault tips. The main difference between the three methods can be seen for the epistemic uncertainty, where SWA-G shows considerably lower uncertainty. These differences are related to the individual approximation methods and their ability to represent the posterior distribution of the weights (Eq. 1 and Fig. 1). We have also conducted such evaluation on real datasets from the North Sea, Offshore Netherlands, and Western Australia where we had similar observation to those of the synthetic case.
Conclusions

We have evaluated three methods to obtain calibrated CNNs that allow prediction of faults within seismic data and their associated uncertainties. Deep Ensembles outperform Concrete Dropout and SWA-G in terms of quantitative metrics (Table 1), but this comes at a high computational cost. Based on our evaluation, both SWA-G and Concrete Dropout represent computationally feasible methods with competitive quantitative and qualitative performance. These presented approaches can be readily applied to other subsurface related workflows, such as property inversion.

References