Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 201 (2022) 771–776

Procedia Computer Science

www.elsevier.com/locate/procedia

The 3rd International Workshop on Statistical Methods and Artificial Intelligence (IWSMAI22)
March 22 - 25, 2022, Porto, Portugal

Bayesian dropout

Tue Herlau[a,*], Mikkel N. Schmidt[a], Morten Mørup[a]

[a]*Technical University of Denmark, Richard Petersens plads 21, 2800 Lyngby, Denmark*

Abstract

In the past decade, Dropout has emerged as a powerful and simple method for training neural networks preventing co-adaptation by stochastically omitting neurons. Dropout is currently not grounded in explicit modelling assumptions which so far has precluded its adoption in Bayesian modeling. Using Bayesian entropic reasoning we show that dropout can be interpreted as optimal inference under constraints. We demonstrate this on an analytically tractable regression model providing a Bayesian interpretation of its mechanism for regularizing and preventing co-adaptation as well as its connection to other Bayesian techniques, and in our experiments we find that dropout can provide robustness under model misspecification. Our framework roots dropout as a theoretically justified and practical tool for statistical modeling allowing Bayesian practitioners to tap into the benefits of dropout training.

Keywords: Dropout; Bayesian learning; Maximum entropy

1. Introduction

Consider a probabilistic model of a dataset parameterized by the parameter vector θ. Often, the dataset will contain complicated structure and require an expressive model. However, a complex model will have many settings of its parameters which are compatible with the training data, and often be *misspecified*, meaning it does not correspond to the underlying generative process. In this case, different settings of the parameters that are compatible with the training data may make different predictions on test data, and there is no guarantee the model will concentrate on parameter values with the better generalization error [10].

In the past decade, Dropout has emerged as a powerful way to address similar problems in neural network training [11]. Dropout stochastically perturbs (typically by setting to zero) parts of the internal representation during training. This prohibit weights to co-adapt to each other, and has been shown to outperform other state-of-the-art regularization methods [11].

* Corresponding author. Tlf.: +45-27830812.
E-mail address: tuhe@dtu.dk

10.1016/j.procs.2022.03.105

The success of dropout has been attributed to a variety of mechanisms, such as the regularizing effect resulting from minimizing appropriately averaged log-likelihood functions [13, 15], bagging or feature corruption where the input data is being perturbed [2, 13, 3], or as maximizing a lower bound on the Bayesian marginal likelihood [14].

However, the most natural interpretation of dropout is that dropout provides a form of Bayesian model averaging, in which the dropout-perturbations induces a distribution over the deterministic neural networks predictions. In this view, the role of an interpretation of dropout is to describe this distribution. One such interpretation was given in the seminal work [6], in which the posterior of a variant of dropout (Monte-Carlo dropout) is shown to be an approximation of a deep Gaussian process.

This work takes a more direct approach, namely to identify the circumstances under which dropout provides the unique optimal assignment of degrees of rational belief. That is, instead of describing the effect of dropout on a model (for instance, as inducing a Bayesian model averaging procedure through the perturbations), we view dropout as a specific model invariance, and the resulting distribution –i.e., Bayesian model with dropout enabled– as simply the maximum entropy distribution under this invariance. For this reason, we dub the method *Bayesian dropout*.

We accomplish this by using that Bayesian inference is a special case of the principle of maximum entropy (ME) [9], which naturally allows us to formulate dropout as a constraint on the space of models within which we are maximizing the entropy. As this view is model-agnostic, dropout can be applied as a principled tool in Bayesian modeling to any probabilistic model for which a dropout-perturbation can be defined. We illustrate the method using a Bayesian linear regression and when maximizing the resulting likelihood function we recover non-Bayesian dropout approaches for linear regression based on loss-functions [14, 13].

2. Methods

The goal of machine learning is to predict, explain and control the environment in a rational manner based on new information and past beliefs. R. T. Cox showed that a theory of degrees of rational beliefs is only consistent if they obey the rules of probability theory [4], and this insight has given rise to Bayesian methods, where the past beliefs are identified with priors on the model parameters, and the available information is the observed data and the relationship between data $\mathbf{x}$ and parameters θ. This leads to Bayes' theorem $p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})}$. However, Bayes' theorem is not the most general way of arriving at rational beliefs. In many situations the relevant information available in the form of constraints, for instance an ideal gas where the relevant constraints include energy conservation. In this case, the method of *maximum entropy*, MaxEnt, allows assignment of rational beliefs under expectation constraints [12].

A more general system of rational inference must be able to handle information in the form of both observed data and arbitrary constraints in an objective manner. If these constraints are available to us in the form of a likelihood (a constraint on the class of posterior functions) and observed data, the method must reduce to Bayesian inference. If the constraint is in the form of expectations, it must reduce to MaxEnt. This can be accomplished by the *extended method of maximum entropy* (ME), which contain both methods as special cases [9].

Extended method of maximum entropy:. Let $\mathbf{z}$ be all the variables under consideration. The goal for a rational learner is to update from a prior distribution $q(\mathbf{z})$ to a posterior distribution $p(\mathbf{z})$ when new information is made available.

The relevant information can come in the form of observed data, priors, the form of the likelihood or expectations, all of which constrain the posterior to belong to a family of distributions $p \in C$. The method of ME makes the assumption that not all distributions $p \in C$ are equally desirable, and assume they can be put in an order of preference represented by the functional $S[p, q]$. Using the further assumptions of locality, coordinate invariance, consistency for independent subsystems, it can be shown that S must have the form [9]

$$S[p, q] = \int d\mathbf{z}\; p(\mathbf{z}) \log \frac{p(\mathbf{z})}{q(\mathbf{z})}. \tag{1}$$

In a machine-learning context, we would consider $\mathbf{z}$ as consisting of n observed data points $\mathbf{x}_i$ and parameters of interest θ. In this case, the relevant information is that we observed an actual value of the data, $\mathbf{x}'$, and this places the constraint on the family of posteriors p that they must belong to

$$C = \{p \mid \textstyle\int d\theta\; p(\mathbf{x}, \theta) = \delta(\mathbf{x} - \mathbf{x}')\}, \tag{2}$$

where δ is the delta-function. Maximizing eq. (1) under this constraint exactly recovers Bayes' theorem [9].

Bayesian Dropout. The neural network formulation of dropout (c.f. [11]) consider a function $\mathbf{y} = f_\theta(\mathbf{x})$ (the neural network) parameterized by a set of weights θ which maps inputs to outputs. The simplest way to train the network is by gradient descent with respect to θ on the loss function averaged over input and output training pairs. Dropout is a simple extension where between each gradient descent step and for each observation $\mathbf{x}_i$ a perturbed set of parameters is generated $\tilde{\theta}_i \sim p(\cdot|\theta)$ and a single gradient update is performed on the modified error function $E(\theta) = \sum_i L(y_i, f_{\tilde{\theta}_i}(x_i)))$.

The perturbed set of parameters are most commonly obtained by simply blanking out a fraction of the hidden units independently [11]. In the limit of low learning rate on θ, dropout will favor weights θ that tend to give good performance when a fraction of the inputs are missing thereby reducing co-adaptation [11].The main difference between dropout and simple model averaging is that dropout does not weight a given set of perturbed parameters by the posterior probability, and it is for this reason we need to specify dropout using the ME framework.

Consider a Bayesian equivalent of a neural network model with joint likelihood $p(\mathbf{x}, \theta) = p(\mathbf{x}|\theta)p(\theta)$ where the likelihood term, for $i = 1, \ldots, n$, is $y_i|\theta \sim \text{Normal}(\cdot\,;\ f_\theta(x_i), \sigma^2)$. It is easy to see that assuming flat priors the MAP solution to this model will be exactly equivalent to the global maximum of the neural-network error. We might consider adding a dropout step to the generative model so it becomes:

$$\theta \sim p(\cdot), \qquad \text{for each } i\text{:} \quad \tilde{\theta}_i \sim p(\cdot|\theta), \qquad y_i|\tilde{\theta} \sim \text{Normal}(\cdot\,;\ f_{\tilde{\theta}_i}(x_i), \sigma^2). \tag{3}$$

Although this generative process bears some similarity to dropout, it will infer which weights are best to blank out to explain each observation, and *not* describe a parameter specification which is robust to *having* parameters blanked out. The effect of this will be twofold. (i) It will create a mixture of models, each model (corresponding to a set of dropped-out parameters) being allowed to co-adapt to the data. (ii) the mixture will be weighted with the posterior probability, thereby reducing the effective number of components.

To implement dropout, we must specify dropout as an external constraint on the learners representation, namely that individual parameters are zeroed stochastically. To put this in words, *Bayesian Dropout* can be defined as *the constraint that the dropout distribution is not inferred from data*. To implement this using ME, we first need to fix the prior measure q and the relevant constraints. As prior distribution q we adopt the same functional form as the naive Bayesian dropout eq. (3). Using $\tilde{\theta} = (\tilde{\theta}_i)_{i=1}^n$, the joint distribution may be written

$$q(\mathbf{x}, \tilde{\theta}, \theta) = q(\mathbf{x}|\tilde{\theta})q(\tilde{\theta}|\theta)q(\theta). \tag{4}$$

Next we specify the constraints: The first constraint is simply the data-constraint that we observed an actual value of $\mathbf{x}$, namely $\mathbf{x}'$ (see eq. (2)). The second constraint is the dropout-condition. To say weights are dropped out stochastically, and that this distribution is not inferred from data, is saying exactly that the dropout weights must depend on θ, but **not** on $\mathbf{x}$. We can therefore identify Bayesian Dropout with the constraint

$$p(\tilde{\theta}|\mathbf{x}, \theta) \equiv q(\tilde{\theta}|\theta) \quad \textit{(Bayesian Dropout)} \tag{5}$$

Accordingly we have $p(\mathbf{x}, \tilde{\theta}, \theta) = p(\tilde{\theta}|\theta)p(\mathbf{x}, \theta)$. The ordering eq. (1) becomes:

$$S[p, q] = \int d\mathbf{x}\mathbf{d}\tilde{\theta}d\theta\ p(\mathbf{x}, \tilde{\theta}, \theta) \log \frac{p(\mathbf{x}, \tilde{\theta}, \theta)}{q(\mathbf{x}, \tilde{\theta}, \theta)} = \int d\mathbf{x}\mathbf{d}\tilde{\theta}d\theta\ p(\mathbf{x}, \theta)p(\tilde{\theta}|\theta) \log \frac{p(\mathbf{x}, \theta)}{q(\mathbf{x}|\tilde{\theta})q(\theta)}. \tag{6}$$

The distribution which uniquely maximizes this functional can be found by taking the functional derivative with respect to $p(\mathbf{x}, \theta)$ while introducing lagrange multipliers $\lambda_\mathbf{x}$ to handle the (infinite) number of data-constraints eq. (2) and α to handle the sum constraint. This leads to the variational problem:

$$0 = \frac{\delta}{\delta p(\mathbf{x}, \theta)} \left\{ S[p, q] + \alpha \left[\int d\mathbf{x}d\tilde{\theta}d\theta\ p(\mathbf{x}, \tilde{\theta}, \theta) - 1 \right] + \int d\mathbf{x}\lambda_\mathbf{x} \left[\int d\tilde{\theta}d\theta\ p(\mathbf{x}, \tilde{\theta}, \theta) - \delta(\mathbf{x} - \mathbf{x}') \right] \right\}. \tag{7}$$

Performing the functional derivative and solving for $p(\mathbf{x}, \theta)$ gives

$$p(\mathbf{x}, \theta) = \frac{1}{z} q(\theta) \exp\left(\int d\tilde{\theta}\ q(\tilde{\theta}|\theta) \log q(\mathbf{x}|\tilde{\theta}) + \lambda_\mathbf{x} \right), \tag{8}$$

where z handles normalization. Recall the requirement that the posterior is consistent with the observed data $\mathbf{x}'$ $\int d\theta\ p(\mathbf{x}, \theta) = \delta(\mathbf{x} - \mathbf{x}')$. Using this identity on the right-hand side of eq. (8) to fix the Lagrange multipliers $\lambda_\mathbf{x}$

$$p(\mathbf{x}, \theta) = \frac{1}{\mathcal{Z}(\mathbf{x})} q(\theta) \exp\left(\int d\tilde{\theta}\ q(\tilde{\theta}|\theta) \log q(\mathbf{x}|\tilde{\theta}) \right) \delta(\mathbf{x} - \mathbf{x}'), \quad \mathcal{Z}(\mathbf{x}) = \int d\theta\ q(\theta) \exp \int d\tilde{\theta}\ p(\tilde{\theta}|\theta) \log q(\mathbf{x}|\tilde{\theta}), \tag{9}$$

such that the posterior distribution of the parameters becomes

$$p(\theta) = \frac{1}{Z(\mathbf{x}')} q(\theta) \exp\left(\int d\tilde{\theta}\, q(\tilde{\theta}|\theta) \log q(\mathbf{x}'|\tilde{\theta}) \right). \tag{10}$$

Note that p refers to our final state of knowledge and is therefore not conditioned on $\mathbf{x}'$. If $p(\tilde{\theta}|\theta) = \delta(\tilde{\theta} - \theta)$ the expression reduces to Bayes' theorem, but otherwise values of θ which are not robust to dropout will be penalized by the log term.

Bayesian linear regression with dropout:. We examine Bayesian dropout by applying it to a conjugated Bayesian linear regression model [7]. The joint prior measure q is defined by the generative process

$$\sigma^2 \sim \text{Gamma}(a_0, b_0), \qquad \mathbf{w}|\sigma^2 \sim \text{Normal}\Big(\mathbf{0}, \frac{\sigma^2}{\lambda_0}\mathbf{I}\Big), \qquad \tilde{\mathbf{w}}_i|\mathbf{w} \sim p_f(\mathbf{w}), \qquad y_i|\tilde{\mathbf{w}}_i, \sigma^2 \sim \text{Normal}(\mathbf{x}_i^T \tilde{\mathbf{w}}_i, \sigma^2), \tag{11}$$

where $\mathbf{y}$ is an n-dimensional vector of responses, and $\mathbf{w}$ and $\mathbf{x}_i$ are p-dimensional vectors of weights and covariates. Compared to eq. (10), $\theta = (\mathbf{w}, \sigma)$. We consider the limit $a_0, b_0 \to 0$ corresponding to the Jeffreys prior $\sigma^2 \sim \sigma^{-2}$. As a dropout distribution $p_f(\mathbf{w})$, we consider independent binary dropout with rate f,

$$p_f(\mathbf{w}) = \prod_{d=1}^{p} [(1-f)\delta(\tilde{w}_{id} - w_d) + f\delta(\tilde{w}_{id})]. \tag{12}$$

Computing the expectation of the log likelihood with respect to the dropout distribution yields

$$\Big\langle \log q(\mathbf{y}|\tilde{\mathbf{w}}) \Big\rangle = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\mathbf{y} - \hat{\mathbf{y}})^\top(\mathbf{y} - \hat{\mathbf{y}}) - f(1-f)\frac{1}{2\sigma^2}\mathbf{w}^T\mathbf{\Lambda}_1\mathbf{w}, \tag{13}$$

where $\hat{\mathbf{y}} = (1-f)\mathbf{X}\mathbf{w}$, $\mathbf{\Lambda}_1 = (\mathbf{X}^\top\mathbf{X}) \circ \mathbf{I}$, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]^T$ is an $n \times p$ matrix of covariates and $\circ$ is the Hadamard product. This is combined with the conjugate prior for $\mathbf{w}$, $p(\mathbf{w}|\sigma^2) = \left(\frac{\lambda_0}{2\pi\sigma^2}\right)^{p/2} \exp\left(-\frac{\lambda_0}{2\sigma^2}\mathbf{w}^\top\mathbf{w}\right)$, gives the posterior

$$p(\mathbf{w}, \sigma|\mathbf{y}) = \frac{1}{(2\pi\sigma^2)^{\frac{n+p}{2}}} \exp\left(-\frac{1}{2\sigma^2}\left[(\mathbf{w} - \mu_n)^T\mathbf{\Lambda}(\mathbf{w} - \mu_n) + \mathbf{y}^T\mathbf{y} - \mu_n^T\mathbf{\Lambda}\mu_n \right] \right), \tag{14}$$

where $\mathbf{\Lambda} = \lambda_0\mathbf{I} + f(1-f)\mathbf{\Lambda}_1 + (1-f)^2\mathbf{X}^T\mathbf{X}$ and $\mu_n = (1-f)\mathbf{\Lambda}^{-1}\mathbf{X}^T\mathbf{y}$. Examining eq. (14), we see this is equal to the familiar Bayesian linear model in the limit $f = 0$, but more generally Bayesian dropout serves as a data dependent prior. The weights of larger features are regularized more heavily by dropout than by ridge regression as also observed by [14]. Also note that removing the prior term λ_0 and taking the maximum likelihood of eq. 14 with respect to $\mathbf{w}$ while keeping σ constant recover the expression for learning with marginalized corrupted features [13].

Dropout as parameter shrinkage. Bayesian priors often understood as improving performance through regularization. From eq. (14) it is evident that in the linear regression model, when the prior is $\lambda_0 = 0$, dropout shrinks parameters towards $\mathbf{\Lambda}_1^{-1}\mathbf{X}^\top\mathbf{y}$, corresponding to the maximum likelihood estimate of each weight in isolation of the other (due to the Hadamard product in $\mathbf{\Lambda}_1$). Thus, dropout provides shrinkage towards a solution with no co-adaptation. This is illustrated in fig. 1 (a) for a quantitative structure property relationship (QSPR) data example [16] with $n = 19$ observation and $p = 7$ highly correlated covariates. The sign of the DGR feature was flipped so that all covariates were positively correlated with the response. In contrast ridge regression and Lasso, dropout shrinks the parameters towards an all positive solution, as would be expected when all covariates are positively correlated with the response.

In fig. 1 (b), we examined the generalization performance of Baysian dropout under model misspecification. We generated $n = 20$ training and test data from the normal linear regression model with $\lambda_0 = \sigma^2 = 1$ and plotted the crossvalidation squared error averaged over 100 000 random data sets. In dropout we used an incorrect prior $\lambda_0 = 10^{-3}$ to examine its performance under model misspecification. The covariates $\mathbf{X}$ were chosen as $\mathbf{X} = \mathbf{RL}$ where $\mathbf{R}$ was a 20×10 standard normal i.i.d. random matrix and $\mathbf{L}$ was a $10 \times p$ random projection matrix where each column had unit length. We considered both the underdetermined ($n > p = 10$), determined ($n = p = 20$), and overdetermined ($n < p = 40$) scenario under three different conditions: In the *default* condition, the response was generated directly from the model. In this condition, ridge regression with the correct prior is optimal, and dropout was found to perform

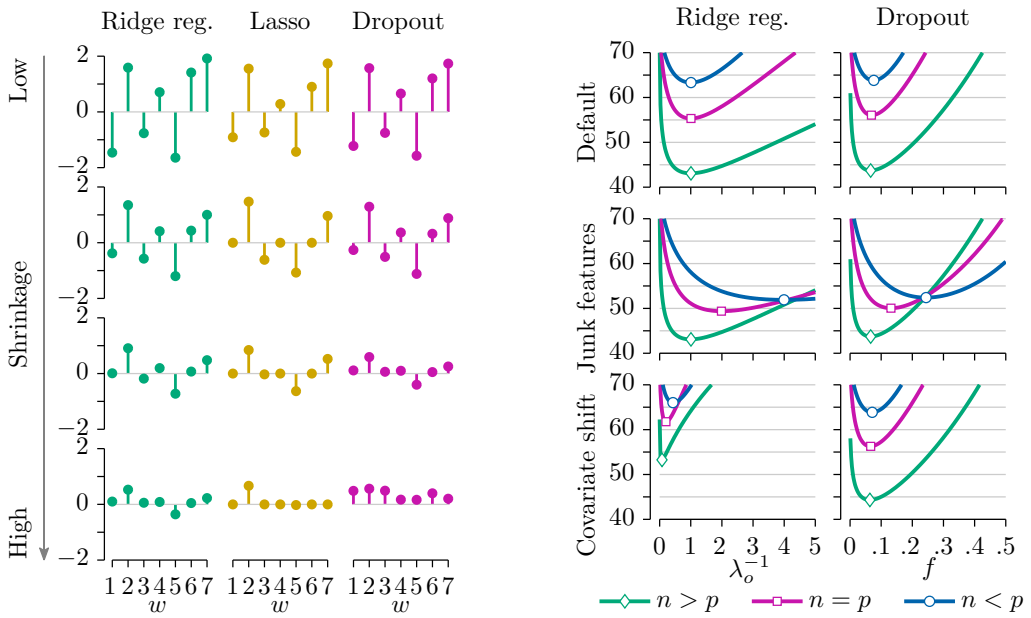

Fig. 1: (a) Comparison of dropout, ridge regression and Lasso for the quantitative structure property relationship (QSPR) data. (b) Performance of ridge regression and dropout in the three scenarios and three conditions.

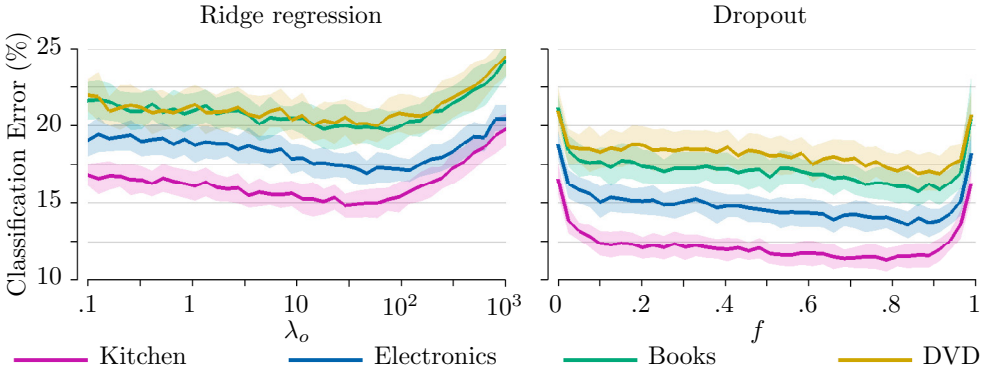

Fig. 2: Performance of Bayesian linear regression with L_2 regularization (ridge regression) and dropout on four binary prediction problems.

on par. In the *junk features* condition, we set all but the first 10 weights to zero when generating the data, corresponding to a misspecified prior or as having $p - 10$ noninformative covariates. Again performing on par, both ridge regression and dropout could counter this by increasing the amount of regularization. Finally, in the *covariate shift* condition we generated the data as in the default condition but multiplied each covariate by a normal random number before fitting the models. We found that ridge regression was quite sensitive to this type of model mismatch whereas dropout was significantly more robust.

Finally, we evaluated Bayesian linear regression with dropout on the Amazon review dataset [1]. The dataset consists of a bag-of-words representation of product reviews in four categories: Books, DVDs, Electronics, and Kitchen, and the task is to predict if the review is positive or negative. The number of observations ranges from 3 587 to 5 946 and the number of covariates from 123 099 to 193 220. We randomly selected 75% of the documents as training data and computed the predictive error on the remaining documents (see Figure 2). We computed the average and standard deviation of the test error over 40 simulations for each data point for Bayesian linear regression with and without dropout for varying values of the scale parameter λ_0. Dropout was found to significantly improved generalization performance for a wide range of parameter settings.

3. Discussion and conclusion

A classical result first proved by J. L. Doob in 1948 is that for finite parameter spaces the Bayesian posterior distribution will almost surely concentrate on the true parameter value provided a consistent estimator exists and the true parameter value is in support of the prior [5]. In this light it might be surprising that additionally restricting the posterior model class can provide any benefits; however, these results must be interpreted in the light of model misspecification.

When the model is misspecified, Bayesian learning will concentrate the posterior around values of θ with the highest likelihood, however, depending on how the model is misspecified, there are no additional guarantees, and the posterior can converge to parameter values which do not give the best prediction [10, 8]. This situation is similar to what motivated dropout in neural networks, and may provide reasons to add constraints such as dropout to make the model less likely to co-adapt.

This also sheds some light on the more general question where constraints come from in the first place. The model and prior probabilities commonly reflect a tradeoff between convenience, scientific knowledge, and tractability [8], and under this view, Bayesian dropout can be well motivated as either reflecting scientific knowledge such as energy conservation, or the simple fact the model is *known* to be misspecified in such a way it is desirable to prevent co-adaptation by adding an appropriate constraint.

Dropout provides a simple yet powerful tool to avoid co-adaptation in neural networks and has been shown to offer tangible benefits; however, its formulation as an algorithm rather than as a set of probabilistic assumptions precludes its use in Bayesian modelling. We have shown how dropout can be interpreted as optimal inference under a particular constraint. This qualifies dropout beyond being a particular optimization procedure, and has the advantage of giving researchers who want to apply dropout to a particular model a principled way to do so.

We have demonstrated Bayesian dropout on an analytically tractable regression model, providing a probabilistic interpretation of its mechanisms for regularizing and preventing co-adaptation as well as its connection to other Bayesian techniques. In our experiments we find that dropout can provide robustness under model misspecification, and offer benefits over ordinary Bayesian linear regression in a real dataset.

References

[1] Blitzer, J., Dredze, M., Pereira, F., . Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification, in: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Association for Computational Linguistics, Prague, Czech Republic. pp. 440–447.

[2] Burges, C.J.C., Schölkopf, B., 1997. Improving the Accuracy and Speed of Support Vector Machines, in: Advances in Neural Information Processing Systems 9, MIT Press. pp. 375–381.

[3] Chen, M., Xu, Z., Weinberger, K.Q., Sha, F., 2012. Marginalized denoising autoencoders for domain adaptation, in: Proceedings of the 29th International Coference on International Conference on Machine Learning, pp. 1627–1634.

[4] Cox, R.T., 1946. Probability, frequency and reasonable expectation. American journal of physics 14, 1–13.

[5] Doob, J.L., 1949. Application of the theory of martingales, in: Le Calcul des Probabilités et ses Applications. Centre National de la Recherche Scientifique, Paris. Colloques Internationaux du Centre National de la Recherche Scientifique, no. 13, pp. 23–27.

[6] Gal, Y., Ghahramani, Z., 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in: international conference on machine learning, PMLR. pp. 1050–1059.

[7] Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B., 2003. Bayesian Data Analysis, (Chapman & Hall/CRC Texts in Statistical Science) .

[8] Gelman, A., Shalizi, R., 2012. Philosophy and the practice of Bayesian statistics. British Journal of Mathematical and Statistical Psychology .

[9] Giffin, A., Caticha, A., Knuth, K.H., Center, J.L., Rodriguez, C.C., 2007. Updating Probabilities with Data and Moments, in: AIP Conference Proceedings, AIP. pp. 74–84.

[10] Grünwald, P., Langford, J., 2007. Suboptimal behavior of Bayes and MDL in classification under misspecification. Machine Learning 66, 119–149.

[11] Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2012. Improving neural networks by preventing co-adaptation of feature detectors. CoRR abs/1207.0.

[12] Jaynes, E.T., 1957. Information theory and statistical mechanics. Physical review 106, 620.

[13] Maaten, L., Chen, M., Tyree, S., Weinberger, K.Q., 2013. Learning with Marginalized Corrupted Features, in: Proceedings of the 30th International Conference on Machine Learning, JMLR Workshop and Conference Proceedings. pp. 410–418.

[14] Wang, S.I., Manning, C.D., 2013. Fast dropout training, in: International Conference on Machine Learning (ICML).

[15] Wei, C., Kakade, S., Ma, T., 2020. The implicit and explicit regularization effects of dropout, in: International Conference on Machine Learning, PMLR. pp. 10181–10192.

[16] Wold, S., 2001. PLS-regression: a basic tool of chemometrics. Chemometrics and Intelligent Laboratory Systems 58, 109–130.