Exploring Bayesian Deep Learning for Urgent Instructor Intervention Need in MOOC Forums

Jialin Yu, Laila Alrajhi, Anoushka Harit, Zhongtian Sun, Alexandra I. Cristea, and Lei Shi

Department of Computer Science, Durham University, Durham, UK
{jialin.yu,laila.m.alrajhi,anoushka.harit,
zhongtian.sun,alexandra.i.cristea,lei.shi}@durham.ac.uk

Abstract. Massive Open Online Courses (MOOCs) have become a popular choice for e-learning thanks to their great flexibility. However, due to large numbers of learners and their diverse backgrounds, it is taxing to offer real-time support. Learners may post their feelings of confusion and struggle in the respective MOOC forums, but with the large volume of posts and high workloads for MOOC instructors, it is unlikely that the instructors can identify all learners requiring intervention. This problem has been studied as a Natural Language Processing (NLP) problem recently, and is known to be challenging, due to the imbalance of the data and the complex nature of the task. In this paper, we explore for the first time Bayesian deep learning on learner-based text posts with two methods: Monte Carlo Dropout and Variational Inference, as a new solution to assessing the need of instructor interventions for a learner's post. We compare models based on our proposed methods with probabilistic modelling to its baseline non-Bayesian models under similar circumstances, for different cases of applying prediction. The results suggest that Bayesian deep learning offers a critical uncertainty measure that is not supplied by traditional neural networks. This adds more explainability, trust and robustness to AI, which is crucial in education-based applications. Additionally, it can achieve similar or better performance compared to non-probabilistic neural networks, as well as grant lower variance.

Keywords: Deep Learning · Artificial Intelligence in Education · Educational Data Mining · Bayesian Modelling · Urgent Instructor Intervention · Natural Language Processing.

1   Introduction

MOOCs are well-known for their high dropout rates [2][3]. Whilst learners may discuss their problems in the forums before actually dropping out, the sheer volume of posts renders it almost impossible for instructors to address them. Thus, many of these urgent posts are overlooked or discarded. Hence, a few researchers proposed [4][9] automated machine learning models for need prediction based on learners' posts in MOOC forums. Such an approach would allow instructors to

identify learners who require urgent intervention, in order to, ultimately, prevent potential dropouts (see our recent research, where we have shown only 13% of learners passing urgent intervention messages complete the course [27]).

More recently, techniques for applying deep neural networks to interpret texts from the educational field have emerged [17], including identifying learners' needs based on their posts in forums [5][16][36]. Despite their success, standard deep learning models have limited capability to incorporate uncertainty. One other challenge is that post data is notoriously imbalanced, with urgent posts representing a very low percentage of the overall body of posts - the proverbial 'needle in the haystack'. This tends to make a neural network overfit and ignore the urgent posts, resulting in large variance in model predictions.

To address the above two challenges, we apply Bayesian probabilistic modelling to standard neural networks. Recent advances in Bayesian deep learning offer a new theory-grounded methodology to apply probabilistic modelling using neural networks. This important approach is yet to be introduced in the Learning Analytics (LA) field.

Thus, the main contributions of this work are:

1. We present the first research on how Bayesian deep learning can be applied to text-based LA. Here, the aim is to predict instructor intervention need in the educational domain.
2. Hence, we explore, for the first time, not only one, but two Bayesian deep learning methods, on the task of classifying learners' posts based on their urgency, namely Monte Carlo Dropout and Variational Inference.
3. We show empirically the benefits of Bayesian deep learning for this task and we discuss the differences in our two Bayesian approaches.
4. We achieve competitive results in the task and obtain a lower variance when training with small size data samples.
5. We apply this approach to text-based processing on posts in MOOCs - a source generally available across all MOOC providers. Thus our approach is widely applicable - generalisable to foresee instructor's intervention need in MOOCs in general and to support the elusive problem of MOOC dropout.

2   Related Work

2.1   Urgent Intervention Need in MOOCs

Detection of the need for urgent instructor intervention is arguably one of the most important challenges in MOOC environments. The problem was first proposed and tackled [10] as a binary prediction task on instructor's intervention histories based on statistical machine learning. A follow-up study [9] proposed the use of $L1$ regularisation techniques during the training and used an additional feature about the type of forum (thread), besides the linguistic features of posts. Another study [4] tried to build a generalised model, using different shallow ML models with linguistic features extracted by NLP tools, metadata and term frequency. In general, this problem was attempted based on two types

of data format: text-only [6][11][16][36][39] or a mixture of text and post features [4][9]. From a machine learning perspective, both traditional machine learning methods [4][6][9] and deep learning based methods [11][16][36][39] were proposed and explored; with more recent studies being in favour of deep neural network-based approaches [17]. However, one critical problem for deep neural networks is that they do not offer a robust estimation over the prediction values. Also, we can not perform efficient learning on small sample size data. Thus, in this paper, we explore the benefits of Bayesian deep learning to predict learners who require urgent interventions from an instructor. We use text only features in our study, as it is the first study to explore the benefits of this new approach, and we leave future optimisation for further work.

To the best of our knowledge, this is the first study of Bayesian deep learning methods for learners' urgent intervention need classification. Our research sheds light on a new direction for other researchers in the fields of Educational Data Mining (EDM) and Learning Analytics (LA).

2.2 Bayesian Neural Networks

Modern neural networks (NNs) are self-adaptive models with a learnable parameter set W. In a supervised learning setting, given data $D = (x_i, y_i)_{i=1}^{N}$, we aim to learn a function through the neural network $y = f_{NN}(x)$ that maps the inputs x to y. A point estimation of the model parameter set W^* is obtained through a gradient based optimisation technique and with a respective cost function.

Bayesian neural networks (BNNs) [29][30][32], alternatively, consider the probability of the distribution over the parameter set W and introduce a prior over the neural network parameter set $P(W)$. The posterior probability distribution $P(W \mid D)$ is learnt in a data-driven fashion through Bayesian inference. This grants us a distribution over the parameter set W other than a static point estimation, which allows us to model uncertainty in the neural network prediction. In the prediction phase, we sample model parameters from the posterior distribution i.e. $w \sim P(W \mid D)$ and predict results with $f_{NN}^{w}(x)$ for the corresponding y. We marginalise the w samples and obtain an expected prediction. Due to the complexity and non-linearity of neural networks, an exact inference for BNNs is rarely possible, hence various approximation inference methods have been developed [12][15][18][19]. The most widely adopted approximation method is the Monte Carlo Dropout [12], with applications in natural language processing, data analytics and computer vision [13][22][23][42][40]. In this paper, we adopt the same idea and use Monte Carlo Dropout [12] to approximate the neural network as a BNN.

2.3 Variational Inference

Variational inference (VI) [7][21][38] is a general framework for Bayesian statistical modelling and inference under a maximum likelihood learning scheme. It introduces an unobserved random variable as the generative component to

model the probabilistic uncertainty. Given fully observed data $D = (x_i, y_i)_{i=1}^{N}$, we consider them as random variables and use capital letters X and Y to represent them. The unobserved random variable introduced with VI is denoted as Z and passes the information from X to Y. It can be marginalised out with Bayes' rule as:

$$P(X, Y; \theta) = \sum_Z P(X, Y, Z; \theta) = P(Y \mid Z; \theta) P(Z \mid X; \theta) \qquad (1)$$

Under a mean-field assumption [37] over the unobserved random variable Z, we can factorise it as:

$$P(Z; \theta) = P(z_1, ..., z_N; \theta) = \prod_{i=1}^{N} P(z_i; \theta) \qquad (2)$$

Hence for each pair of data x and y, the maximum likelihood learning delivers the following objective with respect to θ:

$$\log P(y|x; \theta) = \log \int_z P(y|z; \theta) P(z|x; \theta) dz \qquad (3)$$

Given observed data $D = (x_i, y_i)_{i=1}^{N}$, we can not directly model the distribution of unobserved z and hence the probability distribution $P(y|z; \theta)$ is intractable for data-driven models, such as neural networks. With VI, an additional variational parameter ϕ with its associated variational family distribution $q(z; \phi)$ is introduced, to approximate the real probability $P(y|z; \theta)$. During the learning process, we minimise the distance between $q(z; \phi)$ and $P(y|z; \theta)$ through the Kullback–Leibler divergence, a term that measures the distance between two probability distributions. Hence, the learning of the intractable probability distribution problem is converted to an optimisation problem over the evidence lower bound (ELBO), where $\mathbb{D}_{KL}$ refers to the Kullback–Leibler divergence:

$$\log P(y|x; \theta) \geq \mathcal{L}(ELBO) = \mathbb{E}_{q(z; \phi)}[\log P(y|z; \theta)] - \mathbb{D}_{KL}[q(z; \phi)||p(z|x; \theta)] \quad (4)$$

VI was initially developed to solve a specific class of modelling problems where conditional conjugacy is presumed, and variational parameter ϕ is updated through closed-form coordinate ascent [14]. However, conditional conjugacy is not practical in most of the real world problems; thus further advancements [8][20][25][34][41] extend VI to large scale datasets and non-conjugate models.

3   Methodology

In this section, we first introduce the baseline model built based on recurrent neural networks (RNNs) and an attention mechanism. Then we present our two approaches for applying Bayesian deep learning with our baseline model: 1) Monte Carlo Dropout and 2) Variational Inference.

3.1 Baseline Deep Learning Model

In this section, we first introduce our non-Bayesian model, which serves as our baseline model. The model consists of three different components: an embedding layer, a two-layer recurrent neural network (RNN), and a prediction layer. We use attention based on the output of the RNNs to create a contextual representation over the RNN hidden outputs and then concatenate it with the last layer RNN outputs. The model architecture is presented in Figure 1.

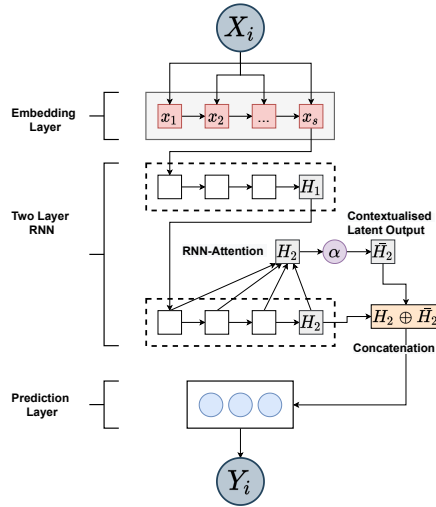Given the data $D = (x_i, y_i)_{i=1}^{N}$, where each sentence x_i consists of a sequence of

Fig. 1. Model architecture for baseline model (operation $\oplus$ refers to the concatenation).

tokens $x_i^1, x_i^2, ..., x_i^s$ where s denotes the sequence length. For our baseline model, given a sentence x_i, we first pass it through the embedding layer and obtain a sequence of word embeddings:

$$E = (emb(x_i^1), emb(x_i^2), ..., emb(x_i^s)) \tag{5}$$

where emb is the embedding function we used for our experiment with d dimensions. Here, x_i^m denotes the mth word in the sentence x_i. For the initial sentence $x_i \in \mathbb{R}^{s \times 1}$, we derive a sentence $x_i \in \mathbb{R}^{s \times d}$ after the embedding layer. Then we feed this as a sequence input through a two-layer long-short-term memory (LSTM) model as in [36]. The initial hidden state h_0 is set to 0 and we calculate the sequence of hidden states as:

$$h_m = LSTM(h_{m-1}, x_i^m) \tag{6}$$

where we have $m = 1, ..., s$. The last layer of hidden states provides a sequence output $H \in \mathbb{R}^{s \times h}$, where h here represents the hidden dimension size. In order to utilise the contextual information through the LSTM encoding process, we calculate the attention score α based on the last hidden state outputs H_2 ($H_2 =$

h_s) and each hidden state in the sequence of H, as:

$$\alpha_m = \frac{H_2 * h_m}{\sum_{m=1}^{s} H_2 * h_m} \tag{7}$$

Then we calculate the contextual $\bar{H}_2$ as:

$$\bar{H}_2 = \sum_{m=1}^{s} \alpha_m h_m \tag{8}$$

Finally, we concatenate them and feed them through a fully connected layer with the output dimension equal to the number of classes for our task. This fully connected layer is represented as a prediction layer in Figure 1.

3.2 Model Uncertainty with Monte Carlo Dropout

In this section, we present how to convert our baseline model into a Bayesian neural network. With Monte Carlo Dropout [12], we only need to use the dropout technique [35] before each layers containing the parameter set W. In our case, we add a dropout layer after the first and second LSTM layers, as well as after the fully connected layer, which takes the input as the concatenation of $\bar{h_2}$ and h_2.

Compared with the standard dropout technique, which works as a regularisation

Fig. 2. A demonstration of the Monte Carlo Dropout in the test phase. We run the model for M times for M different prediction results and then calculate their average as the prediction layer output.

technique in the training phase only, the Monte Carlo Dropout technique requires the dropout layer to be activated in both training and testing phases. This allows the standard neural network model to work as a BNN [12]. Each dropout works

as a sample of w from its probabilistic distribution space and hence allows us to measure the uncertainty of the model, as shown in Figure 2. In the testing phase, we predict the output through sampling M times [12] and the expectation of y can be calculated as:

$$\mathbb{E}(y \mid x) \approx \frac{1}{M} \sum_{i=1}^{M} f_{NN}^{w_i}(x) \tag{9}$$

We use this expectation as the final logits value and in our experiment we use a total sample M of 50 as in [23].

3.3 Model Uncertainty with Variational Inference

As discussed in the section 2.3, Variational Inference (VI) introduces an additional random variable z with probability distribution $q(z; \phi)$ to the original model. This variational family $q(z; \phi)$ here approximates the posterior distribution $P(z \mid x; \theta)$ as $q(z|x, y; \phi)$. The model architecture is presented in Figure 3. Following [31], we define $q_\phi(z|x, y)$ as:

$$q(z|x, y; \phi) = \mathcal{N}(z|\mu_\phi(x, y), \mathrm{diag}(\sigma_\phi^2(x, y))) \tag{10}$$

we have:

$$\mu_\phi(x, y) = l_1(\pi_\phi) \tag{11}$$

and:

$$\log \sigma_\phi(x, y) = l_2(\pi_\phi) \tag{12}$$

where:

$$\pi_\phi = g_\phi(H_2, f_y(y)) \tag{13}$$

where $f_y(y)$ is an affine transformation from output $y \in \mathbb{R}^1$ to a vector space size $s_y \in \mathbb{R}^z$. The H_2 is the final latent state output of the second LSTM network layer as stated in section 3.1. The latent variable $z \in \mathbb{R}^h$ can be reparameterised as $z = \mu + \sigma \cdot \epsilon$, known as the "reparameterisation trick" [26] with sample $\epsilon \sim \mathcal{N}(0, \mathbb{I})$. For the conditional distribution $p_\theta(z|x)$, we can model it as:

$$p(z|x; \theta) = \mathcal{N}(z|\mu_\theta(x), \mathrm{diag}(\sigma_\theta^2(x))) \tag{14}$$

where we have:

$$\mu_\theta(x) = l_3(\pi_\theta) \tag{15}$$

$$\log \sigma_\theta(x) = l_4(\pi_\theta) \tag{16}$$

and:

$$\pi_\theta = g_\theta(H_2) \tag{17}$$

where l_1, l_2, l_3 and l_4 are four affine transformation functions. Since both $p(z|x; \theta)$ and $q(z|x, y; \phi)$ are multivariate Gaussian distributions, this allows us to have a closed-form solution for the Kullback–Leibler (KL) divergence term [25]. For the reconstruction term $\log p(y \mid z; \theta)$ with Monte Carlo approximation [31], the final reconstruction loss can be calculated as:

$$\mathbb{E}_{q(z)}[\log p(y|z;\theta)] \approx \frac{1}{M} \sum_{m=1}^{M} \log p(y|z_m \oplus H_2 \oplus \bar{H}_2; \theta) \tag{18}$$

where $\oplus$ denotes the concatenation operation and M is the number of samples from the posterior distribution z. We use a single sample of $M = 1$ during training based on [24] and $M = 20$ during testing based on [31]. In the training phase, z is sampled from $q(z|x, y; \phi)$ and in the test phase, from $p(z|x; \theta)$.
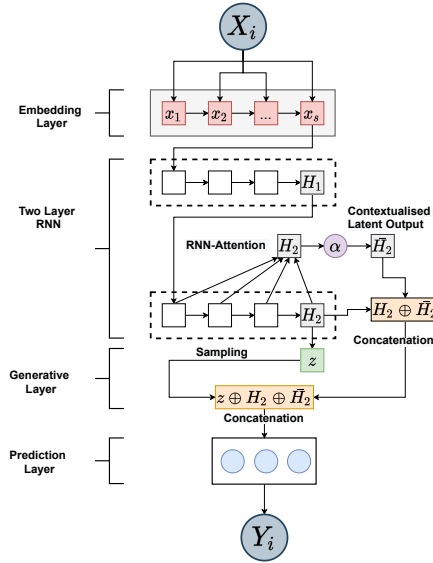

Fig. 3. Model architecture for the VI model.

4   Experiments

4.1   Dataset

Here, we used the benchmark posts dataset from the Stanford MOOC forum [1], containing 29604 anonymised posts collected from 11 different courses. Each post is manually labelled by three independent human experts and with agreements for the gold label. Apart from the text content, each post is evaluated based on six categories, amongst which urgency, which is the one we used here. Its range is 1 to 7; with 1 meaning no reason to read the post and 7 meaning extremely urgent for instructor interventions. An example urgent message is "I hope any course staff member can help us to solve this confusion asap!!!"; whilst a non-urgent would be "Good luck to everyone.". See more details on their website[1]. Similar to [16], we convert the problem of detecting urgent posts to a binary classification task. A threshold of 4 is used as in [16] to create two need categories as: 1) *Need for urgent intervention (value > 4) with label 1*; and 2) *No need for intervention (values ≤ 4) with label 0*. This allows us to obtain a total of 29,597 posts, with

---

[1] https://datastage.stanford.edu/StanfordMoocPosts/

23,991 labelled as 0 and 5,606 labelled as 1. We tokenise the text and create a vocabulary based on a frequency-based cutoff [28] and use the special token $< pad >$ for padding and the unknown token $< unk >$ for out-of-vocabulary words. We initialise the embedding layer with a 300-dimensional GloVe vector [33] if found in the pre-trained token list.

4.2   Experiment Setup and Evaluation

In this paper we have implemented 3 different models: a baseline model ($Base$), as shown in Figure 1; a baseline model converted to a Bayesian neural network through Monte Carlo Dropout (MCD), as shown in Figure 2; and a baseline model with variational inference (VI), as shown in Figure 3. For the evaluation, we report mean accuracy; F1 score, Precision score, Recall score for all three models under each class (the higher the better); and entropy based on the prediction layer [23] [40] (the lower the better).

We conduct two sets of experiments. For the first set, we follow the setup in [5]. At each run of the experiment, we randomly split this data into training and testing sets each with a ratio of 80% and 20%, respectively, with stratified sampling on a random state. In the second set of experiments, we use less training examples, since the intervention case is rare compared with non-intervention, and we compare the robustness of our model given smaller size samples and we use a split of 40%, 60% for training and testing. The results for the two experiments are reported in Table 1 and Table 2, respectively, and we run both experiments 10 times. In Table 1, we report the best run of the model and in Table 2, we report the mean and variance. All the evaluation metrics results reported here in this paper are based on test dataset only. In the first table, we use bold text to denote the results that outperform results in [5] and in the second table, we use bold to denote results outperforming the ($Base$) model.

5   Results and Discussions

The results are presented in Table 1. The baseline model (Base) performs competitively against a strong model [5], especially in the recall and F1 score for the 'urgent' class and the precision score for the 'non-urgent' class. For the Monte Carlo Dropout (MCD) and Variational Inference (VI) models, we achieve better performance in these measurements against the baseline model (Base). Importantly, as an indication of the uncertainty measurement, we note that the entropy dropped for the MCD model. In Table 2, we can see that Bayesian deep learning methods generally achieve similar or better performance compared to the non-Bayesian base model, but hold lower variance and lower entropy against small sample size data. This is often the case in real life scenarios, where the label 'need intervention' is scarce. A probabilistic approach works as a natural regularisation technique, when neural network models are generally over-parameterised. We can conclude that Bayesian deep learning mitigates this issue of over-parametrisation with lower variance and entropy. This is especially clear for the VI methods. The

Table 1. Results compare baseline model and Bayesian deep learning approach in accuracy, precision, recall and F1 score.

			Non-urgent (0)			Urgent (1)		
	Accuracy	**Entropy**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Text [5]	.878	-	.90	.95	.93	.73	.56	.64
Base	**.883**	.095	**.937**	.918	.927	.677	**.738**	**.697**
MCD	.883	**.085**	**.939**	.915	.926	.675	**.742**	**.698**
VI	.873	.103	**.940**	.901	.919	.644	**.752**	.687

Table 2. Results compare mean and variance of Deep learning and Bayesian deep learning approach based on 10 runs.

			Non-urgent (0)			Urgent (1)		
	Accuracy	**Entropy**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Base	.870+-.0039	.1126+-.0041	.930+-.0039	.908+-.0088	.918+-.0030	.645+-.0159	.707+-.0215	.664+-.0052
MCD	.869+-**.0013**	**0.101**+-.0326	.929+-.0128	.908+-.0319	.917+-.0104	.652+-**.0574**	.703+-.0693	.660+-**.0042**
VI	.867+-**.0019**	**0.078**+-.0296	.924+-**.0028**	**.910**+-**.0058**	.916+-**.0017**	**.642**+-**.0093**	.680+-**.0164**	.649+-**.0034**

result from a Wilcoxon test shows that, compared with the Base model, the experiment results of the VI model are statistically significant at the *.05* level, with $p=.022$ for the entropy value and with $p=.007$ for the recall, in the 'urgent' case. Comparing MCD and VI models, the latter achieves better performance in most metrics, as shown in both tables, especially with a higher recall score. The recall score is preferable to precision in this task, where we have a comparatively small number of positive examples. However, the implementation of MCD models is more accessible to researchers interested in introducing uncertainty into their neural networks. This should be considered in using them in practice.

6   Conclusion

Identifying the need of learner interventions for instructors is an extremely important issue in MOOC environments. In this paper, we have explored the benefits of a Bayesian deep learning approach to this problem for the first time. We have implemented two different approaches to Bayesian deep learning, namely Monte Carlo dropout and variational inference. Both offer a critical probabilistic measurement in neural networks. We have demonstrated the effectiveness of both approaches in decreasing the epistemic uncertainty of the original neural network and granting equivalent or even better performance. We have thus provided guidelines for researchers interested in building safer, more statistically sound neural network-based models in the Learning Analytics (LA) field. Entropy measures a classifiers' confidence level. In intelligent tutoring systems, high confidence (thus low entropy) is essential. With Bayesian deep learning, we turn NN models into probabilistic models, allowing more explanability and trust. For future research, these can be extended and applied in more areas.

References

1. Agrawal, A., Venkatraman, J., Leonard, S., Paepcke, A.: Youedu: addressing confusion in mooc discussion forums by recommending instructional video clips (2015)
2. Alamri, A., Alshehri, M., Cristea, A., Pereira, F.D., Oliveira, E., Shi, L., Stewart, C.: Predicting moocs dropout using only two easily obtainable features from the first week's activities. In: International Conference on Intelligent Tutoring Systems. pp. 163–173. Springer (2019)
3. Alamri, A., Sun, Z., Cristea, A.I., Senthilnathan, G., Shi, L., Stewart, C.: Is mooc learning different for dropouts? a visually-driven, multi-granularity explanatory ml approach. In: International Conference on Intelligent Tutoring Systems. pp. 353–363. Springer (2020)
4. Almatrafi, O., Johri, A., Rangwala, H.: Needle in a haystack: Identifying learner posts that require urgent response in mooc discussion forums. Computers & Education **118**, 1–9 (2018)
5. Alrajhi, L., Alharbi, K., Cristea, A.I.: A multidimensional deep learner model of urgent instructor intervention need in mooc forum posts. In: International Conference on Intelligent Tutoring Systems. pp. 226–236. Springer (2020)
6. Bakharia, A.: Towards cross-domain mooc forum post classification. In: Proceedings of the Third (2016) ACM Conference on Learning@ Scale. pp. 253–256 (2016)
7. Bishop, C.M.: Pattern recognition and machine learning. springer (2006)
8. Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: A review for statisticians. Journal of the American statistical Association **112**(518), 859–877 (2017)
9. Chandrasekaran, M.K., Kan, M.Y., Tan, B.C., Ragupathi, K.: Learning instructor intervention from mooc forums: Early results and issues. arXiv preprint arXiv:1504.07206 (2015)
10. Chaturvedi, S., Goldwasser, D., Daumé III, H.: Predicting instructor's intervention in mooc forums. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1501–1511 (2014)
11. Clavié, B., Gal, K.: Edubert: Pretrained deep language models for learning analytics. arXiv preprint arXiv:1912.00690 (2019)
12. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning. pp. 1050–1059 (2016)
13. Gal, Y., Ghahramani, Z.: A theoretically grounded application of dropout in recurrent neural networks. In: Advances in neural information processing systems. pp. 1019–1027 (2016)
14. Ghahramani, Z., Beal, M.: Propagation algorithms for variational bayesian learning. Advances in neural information processing systems **13**, 507–513 (2000)
15. Graves, A.: Practical variational inference for neural networks. Advances in neural information processing systems **24**, 2348–2356 (2011)
16. Guo, S.X., Sun, X., Wang, S.X., Gao, Y., Feng, J.: Attention-based character-word hybrid neural networks with semantic and structural information for identifying of urgent posts in mooc discussion forums. IEEE Access **7**, 120522–120532 (2019)
17. Hernández-Blanco, A., Herrera-Flores, B., Tomás, D., Navarro-Colorado, B.: A systematic review of deep learning approaches to educational data mining. Complexity **2019** (2019)
18. Hernández-Lobato, J.M., Adams, R.: Probabilistic backpropagation for scalable learning of bayesian neural networks. In: International Conference on Machine Learning. pp. 1861–1869 (2015)

19. Hernández-Lobato, J.M., Gelbart, M., Hoffman, M., Adams, R., Ghahramani, Z.: Predictive entropy search for bayesian optimization with unknown constraints. In: International conference on machine learning. pp. 1699–1707. PMLR (2015)
20. Hoffman, M.D., Blei, D.M., Wang, C., Paisley, J.: Stochastic variational inference. The Journal of Machine Learning Research **14**(1), 1303–1347 (2013)
21. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. Machine learning **37**(2), 183–233 (1999)
22. Kendall, A., Badrinarayanan, V., Cipolla, R.: Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. arXiv preprint arXiv:1511.02680 (2015)
23. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: Advances in neural information processing systems. pp. 5574–5584 (2017)
24. Kim, Y., Wiseman, S., Rush, A.M.: A tutorial on deep latent variable models of natural language. arXiv preprint arXiv:1812.06834 (2018)
25. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
26. Kingma, D.P., Salimans, T., Welling, M.: Variational dropout and the local reparameterization trick. In: Advances in neural information processing systems. pp. 2575–2583 (2015)
27. Laila, A., Ahmed, A., Filipe, D.P., Alexandra, I.C.: Urgency analysis of learners' comments: an automated intervention priority model for mooc. International Conference on Intelligent Tutoring Systems (to appear) (2021)
28. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025 (2015)
29. MacKay, D.J.: A practical bayesian framework for backpropagation networks. Neural computation **4**(3), 448–472 (1992)
30. MacKay, D.J.: Probable networks and plausible predictions—a review of practical bayesian methods for supervised neural networks. Network: computation in neural systems **6**(3), 469–505 (1995)
31. Miao, Y., Yu, L., Blunsom, P.: Neural variational inference for text processing. In: International conference on machine learning. pp. 1727–1736 (2016)
32. Neal, R.M.: Bayesian learning for neural networks, vol. 118. Springer Science & Business Media (2012)
33. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
34. Ranganath, R., Gerrish, S., Blei, D.: Black box variational inference. In: Artificial Intelligence and Statistics. pp. 814–822. PMLR (2014)
35. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014)
36. Sun, X., Guo, S., Gao, Y., Zhang, J., Xiao, X., Feng, J.: Identification of urgent posts in mooc discussion forums using an improved rcnn. In: 2019 IEEE World Conference on Engineering Education (EDUNINE). pp. 1–5. IEEE (2019)
37. Tanaka, T.: A theory of mean field approximation. In: Advances in Neural Information Processing Systems. pp. 351–360 (1999)
38. Wainwright, M.J., Jordan, M.I.: Graphical models, exponential families, and variational inference. Now Publishers Inc (2008)
39. Wei, X., Lin, H., Yang, L., Yu, Y.: A convolution-lstm-based deep neural network for cross-domain mooc forum post classification. Information **8**(3), 92 (2017)

40. Xiao, Y., Wang, W.Y.: Quantifying uncertainties in natural language processing tasks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 7322–7329 (2019)
41. Zhang, C., Bütepage, J., Kjellström, H., Mandt, S.: Advances in variational inference. IEEE transactions on pattern analysis and machine intelligence **41**(8), 2008–2026 (2018)
42. Zhu, L., Laptev, N.: Deep and confident prediction for time series at uber. In: 2017 IEEE International Conference on Data Mining Workshops (ICDMW). pp. 103–110. IEEE (2017)