Why you don’t overfit, and don’t need Bayes if you only train for one epoch

Laurence Aitchison
University of Bristol
laurence.aitchison@gmail.com

Abstract

Here, we show that in the data-rich setting where you only train on each datapoint once (or equivalently, you only train for one epoch), standard “maximum likelihood” training optimizes the true data generating process (DGP) loss, which is equivalent to the test loss. Further, we show that the Bayesian model average optimizes the same objective, albeit while taking the expectation over uncertainty induced by finite data. As standard maximum likelihood training in the single-epoch setting optimizes the same objective as Bayesian inference, we argue that we do not expect Bayesian inference to offer any advantages in terms of overfitting or calibration in these settings. This explains the diminishing importance of Bayes in areas such as LLMs, which are often trained with one (or very few) epochs.

1 Introduction

In the early days of deep learning, datasets were small, and we trained for many epochs (Krizhevsky et al., 2012; He et al., 2016). This led to some degree of overfitting, as measured by poor calibration, which could be mitigated by Bayesian neural networks (Graves, 2011; Blundell et al., 2015; Gal & Ghahramani, 2016; Sun et al., 2019; Papamarkou et al., 2024; Unlu & Aitchison, 2020; Ober & Aitchison, 2021; Fortuin et al., 2021) and related methods such as ensembles (Lakshminarayanan et al., 2017; Fort et al., 2019; D’Angelo & Fortuin, 2021), dropout (Srivastava et al., 2014; Gal & Ghahramani, 2016; Folgoc et al., 2021), or methods that encourage the optimizer to find broad modes (Keskar et al., 2016; Dziugaite & Roy, 2017; Jiang et al., 2019; Foret et al., 2020; Zheng et al., 2021). However, more recently two trends have emerged. First, we have much larger datasets (such as multi-trillion token text datasets for LLMs e.g. Gao et al., 2020; Dodge et al., 2021; Elazar et al., 2023; Dubey et al., 2024). With such large datasets, we often only have enough compute for a single pass over the data, or one epoch of training (Touvron et al., 2023; Dubey et al., 2024). Second, overfitting in models such as LLMs appears to be far less of an issue; for instance, see Fig. 8 in Achiam et al., 2023, which shows that the pre-trained GPT-4 is well-calibrated in terms of next-token prediction probablities. As such, in practice Bayesian or related methods such as ensembles are rarely, if ever, used in LLM pre-training, though they are sometimes used in LLM post-training (Wang et al., 2023; Yang et al., 2024b,a, e.g.).

Why is this? Bayesian neural networks and related methods were motivated by the need to mitigate overfitting (Domingos, 2000; Cawley & Talbot, 2007; Watanabe, 2009; Izmailov et al., 2021). Overfitting can be understood in terms of calibration: the tendency of neural networks to become overly certain as training proceeds through many epochs (Cawley & Talbot, 2007; Izmailov et al., 2021). Here, we argue that we do not expect such overfitting, as measured by poor calibration, to occur in modern data rich training pipelines, in which data is not repeated. In particular, we show that Bayesian inference can be understood as minimizing the expected log-likelihood under data drawn from the true data generating process, i.e. minimizing the expected test-loss. Critically, we show that in the data-rich setting where we only have one epoch, we can equivalently optimize the exact same objective, simply using standard “maximum likelihood” training.

2 Results

The first question is why does standard maximum likelihood training on finite data overfit? To make the problem concrete, consider training a neural network with weights w which outputs a probability distribution, $\mathrm{Q}_w(y|x)$. For instance in classification, $\mathrm{Q}_w(y|x)$ would be the distribution over labels as judged by the model, or for an LLM, $\mathrm{Q}_w(y|x)$ would be the distribution over the next token. The training loss is,

$$\mathcal{L}_{\text{empirical}}(w;\theta^*) = -\,\mathrm{E}_{\mathrm{P}_{\text{empirical}}(y|x)\,\mathrm{P}_{\text{empirical}}(x)}\left[\log \mathrm{Q}_w(y|x)\right]. \tag{1}$$

Here, we have $\mathrm{P}_{\text{empirical}}(y|x)$ and $\mathrm{P}_{\text{empirical}}(x)$ because we are sampling data from a finite dataset, $\{x_i, y_i\}_{i=1}^N$. Formally, this distribution is,

$$\mathrm{P}_{\text{empirical}}(x) = \frac{1}{N}\sum_{i=1}^{N}\delta(x - x_i), \tag{2}$$

Then taking the observed value for y at that datapoint (assuming there was only one input at each location)

$$\mathrm{P}_{\text{empirical}}(y|x) = \delta(y - y_i). \tag{3}$$

Now, the optimal neural network $\mathrm{Q}_{w^*}(y|x)$, when evaluated on the training points is,

$$\mathrm{Q}_{w^*}(y|x{=}x_i) = \delta(y - y_i). \tag{4}$$

Thus, all uncertainty vanishes on the input points, and this represents overfitting.

Bayes allows you mitigate overfitting in this setting by allowing you to work with the loss under the true data generating process,

$$\mathcal{L}(w;\theta^*) = -\,\mathrm{E}_{\mathrm{P}(y|x,\theta^*)\,\mathrm{P}(x)}\left[\log \mathrm{Q}_w(y|x)\right]. \tag{5}$$

Here, $\mathrm{P}(y|x,\theta^*)\,\mathrm{P}(x)$ is the true data generating process, with parameters, θ^*. We call this the true data generating process (DGP) loss. It is equivalent to the test loss, but we don't call it the test loss to avoid confusion later. Of course, we don't know the true DGP, so we can't sample from $\mathrm{P}(y|x,\theta^*)\,\mathrm{P}(x)$, and hence we can't evaluate this objective. Nonetheless, Bayes decision theory tells us how to handle this setting: we should minimize the expected loss under the posterior, $\mathrm{P}(\theta^*|\text{data})$, given finite data,

$$\mathcal{L}(w) = \mathrm{E}_{\mathrm{P}(\theta^*|\text{data})}\left[\mathcal{L}(w;\theta^*)\right] \tag{6}$$

$$\mathcal{L}(w) = -\,\mathrm{E}_{\mathrm{P}(\theta^*|\text{data})}\left[\mathrm{E}_{\mathrm{P}(y|x,\theta^*)\,\mathrm{P}(x)}\left[\log \mathrm{Q}_w(y|x)\right]\right] \tag{7}$$

This expected loss can be understood as maximum likelihood on $\mathrm{Q}_w(y|x)$, where we generate y's by:

$$x \sim \mathrm{P}(x) \tag{8}$$

$$\theta^* \sim \mathrm{P}(\theta^*|\text{data}) \tag{9}$$

$$y \sim \mathrm{P}(y|x,\theta^*) \tag{10}$$

Thus, the value of $\mathrm{Q}_w(y|x)$ that optimizes the expected test loss is the Bayesian model average,

$$\mathrm{Q}^*(y|x) = \int d\theta^*\, \mathrm{P}(y|x,\theta^*)\,\mathrm{P}(\theta^*|\text{data}). \tag{11}$$

Importantly, this argument isn't trying to justify Bayes using Bayes decision theory: that would be silly. Instead, we're trying to understand the Bayesian model average as the solution to an optimization problem, namely optimizing the expected test loss.

Now this raises a question: in the data-rich, single-epoch setting where we run only one epoch, can we avoid Bayes and instead do something simpler to optimize the true DGP loss Eq. (5)? Remarkably, in the single epoch setting the answer is yes: the data itself is sampled from the true DGP, $\mathrm{P}(y|x,\theta^*)\,\mathrm{P}(x)$. Thus, you can obtain unbiased estimates of the true DGP loss (Eq. 5) stochastic gradient descent, on a maximum likelihood objective, with a minibatch of data of size B,

$$\tilde{\mathcal{L}}(w) = -\frac{1}{B}\sum_{i=1}^{B}\log \mathrm{Q}_w(y_i|x_i). \tag{12}$$

This is just the standard objective used e.g. in LLM pretraining. As this estimator is unbiased, gradient descent with sufficiently small learning rates will find a (local) optimum of the true DGP objective, (Eq. 5) which is also, remember, the objective optimized by Bayesian inference. Thus, in the single-epoch, data-rich setting, there is no good reason to believe that Bayesian inference will give any improvements over standard maximum-likelihood pretraining in terms of overfitting or calibration.

You may be asking why we can't apply the same argument in the multi-epoch setting. After all, the x_i's and y_i's in the data were ultimately generated from $\mathrm{P}\left(y|x, \theta^*\right) \mathrm{P}\left(x\right)$, whether you do one epoch or multiple epochs. There are two alternative ways of seeing why this is the case. First, in the multi-epoch setting, the data we are actually training on are not sampled from $\mathrm{P}\left(y|x_i, \theta^*\right) \mathrm{P}\left(x_i\right)$. Instead they are sampled from the empirical data distribution, $\mathrm{P}_{\text{empirical}}\left(x, y\right)$. These are different distributions, and this is especially evident if you think about repeated data. In the multi-epoch setting, with distribution $\mathrm{P}_{\text{empirical}}\left(x, y\right)$, datapoints are repeated frequently. In contrast, if you sample from the true DGP, you usually do not expect to see repeated data. Second, we take the weights, w_t to be random variables that depend on the previous datapoints, (i.e. $(x_1, y_1), \dots, (x_{t-1}, y_{t-1})$. This makes clear that the precise notion of unbiasedness we need is,

$$\mathcal{L}(w_t) = \mathrm{E}\left[\tilde{\mathcal{L}}(w_t)|w_t\right] = -\frac{1}{B}\sum_{i=1}^{B} \mathrm{E}\left[\log \mathrm{Q}_{w_t}\left(y_t|x_t\right)|w_t\right]. \tag{13}$$

i.e. we need to condition on the current value of the weights, w_t. This is important because it emphasises that in order to get unbiased estimates, we need samples of (x_t, y_t) conditioned on w_t to be drawn from the true DGP,

$$\mathrm{P}\left(x_t, y_t|\theta^*, w_t\right) = \mathrm{P}\left(y_t, x_t|\theta^*\right), \tag{14}$$

or alternatively, we need (x_t, y_t) to be independent of w_t, conditioned on the true DGP parameters, θ^*,

$$x_t, y_t \perp\!\!\!\perp w_t \;\mid\; \theta^*. \tag{15}$$

We have this in the single-epoch setting, where x_t, y_t are drawn from the true data generating process, and are independent of w_t. But we do not have this in the multi-epoch setting, where we may have trained on the current datapoint previously, and thus there may be dependencies between (x_t, y_t) and w_t.

3 Conclusions

There are several conclusions from this line of argument.

First, when we are training for a single epoch, we are in effect training on the true DGP loss, which is equivalent to the test loss. We therefore do not expect to see overfitting. Technically, we do not expect to see training increase the degree of overfitting, though of course if you start with an overfitted model and train very little, you may still have an overfitted model. In agreement with this conclusion we indeed, do not see overfitting in practice when pre-training modern LLMs on large datasets.

Second, if the key benefits of e.g. Bayesian neural networks are in mitigating overfitting, and if we do not expect LLM pre-training to overfit, then we would not expect explicitly Bayesian methods (e.g. variational Bayes etc.) to give any benefits in LLM pretraining. Given that explicitly Bayesian methods involve some additional cost, this likely means that they should not be used in LLM pretraining.

Third, of course, these arguments do not affect the usefulness of Bayesian methods in mitigating overfitting in settings where multi-epoch training is necessary to achieve acceptable performance. However, as the field increasingly moves towards foundation models that are trained on increasingly larger datasets, (even in post-training Dubey et al., 2024), we expect the usefulness of Bayesian neural networks for the field to diminish generally.

Fourth, in this context, fast (potentially Bayes-inspired) optimizers may become even more useful (Aitchison, 2020; Shen et al., 2024). In addition to the obvious benefits of a faster optimizer, they also allow you to loose less performance if you allow yourself to only go over each datapoint once.

Finally, while the importance of Bayesian neural networks may diminish in machine learning, Bayes will of course remain essential in scientific and statistical settings, where understanding our uncertainty conditioned on finite data is precisely the point.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Aitchison, L. Bayesian filtering unifies adaptive and non-adaptive neural network optimization methods. *Advances in Neural Information Processing Systems*, 33:18173–18182, 2020.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks. In *International Conference on Machine Learning*, 2015.

Cawley, G. C. and Talbot, N. L. Preventing over-fitting during model selection via bayesian regularisation of the hyper-parameters. *Journal of Machine Learning Research*, 8(4), 2007.

D'Angelo, F. and Fortuin, V. Repulsive deep ensembles are bayesian. *Advances in Neural Information Processing Systems*, 34:3451–3465, 2021.

Dodge, J., Sap, M., Marasović, A., Agnew, W., Ilharco, G., Groeneveld, D., Mitchell, M., and Gardner, M. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. *arXiv preprint arXiv:2104.08758*, 2021.

Domingos, P. Bayesian averaging of classifiers and the overfitting problem. In *ICML*, volume 747, pp. 223–230, 2000.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Dziugaite, G. K. and Roy, D. M. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

Elazar, Y., Bhagia, A., Magnusson, I., Ravichander, A., Schwenk, D., Suhr, A., Walsh, P., Groeneveld, D., Soldaini, L., Singh, S., et al. What's in my big data? *arXiv preprint arXiv:2310.20707*, 2023.

Folgoc, L. L., Baltatzis, V., Desai, S., Devaraj, A., Ellis, S., Manzanera, O. E. M., Nair, A., Qiu, H., Schnabel, J., and Glocker, B. Is mc dropout bayesian? *arXiv preprint arXiv:2110.04286*, 2021.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Fort, S., Hu, H., and Lakshminarayanan, B. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

Fortuin, V., Garriga-Alonso, A., Ober, S. W., Wenzel, F., Ratsch, G., Turner, R. E., van der Wilk, M., and Aitchison, L. Bayesian neural network priors revisited. In *International Conference on Learning Representations*, 2021.

Gal, Y. and Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, 2016.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Graves, A. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems*, 2011.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Izmailov, P., Vikram, S., Hoffman, M. D., and Wilson, A. G. G. What are bayesian neural network posteriors really like? In *International conference on machine learning*, pp. 4629–4640. PMLR, 2021.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 2017.

Ober, S. W. and Aitchison, L. Global inducing point variational posteriors for Bayesian neural networks and deep Gaussian processes. In *International Conference on Machine Learning*, 2021.

Papamarkou, T., Skoularidou, M., Palla, K., Aitchison, L., Arbel, J., Dunson, D., Filippone, M., Fortuin, V., Hennig, P., Hubin, A., et al. Position paper: Bayesian deep learning in the age of large-scale ai. *arXiv preprint arXiv:2402.00809*, 2024.

Shen, Y., Daheim, N., Cong, B., Nickl, P., Marconi, G. M., Bazan, C., Yokota, R., Gurevych, I., Cremers, D., and Khan, M. E. Variational learning is effective for large deep networks. In *International Conference on Machine Learning*, 2024.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958, 2014.

Sun, S., Zhang, G., Shi, J., and Grosse, R. Functional variational Bayesian neural networks. In *International Conference on Learning Representations*, 2019.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Unlu, A. and Aitchison, L. Variational laplace for bayesian neural networks. *arXiv preprint arXiv:2011.10443*, 2020.

Wang, X., Aitchison, L., and Rudolph, M. Lora ensembles for large language model fine-tuning. *arXiv preprint arXiv:2310.00035*, 2023.

Watanabe, S. *Algebraic geometry and statistical learning theory*, volume 25. Cambridge university press, 2009.

Yang, A. X., Robeyns, M., Coste, T., Wang, J., Bou-Ammar, H., and Aitchison, L. Bayesian reward models for LLM alignment. *arXiv preprint arXiv:2402.13210*, 2024a.

Yang, A. X., Robeyns, M., Wang, X., and Aitchison, L. Bayesian low-rank adaptation for large language models. *International Conference on Learning Representations*, 2024b.

Zheng, Y., Zhang, R., and Mao, Y. Regularizing neural networks via adversarial model perturbation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8156–8165, 2021.