Neural networks, probability, decision theory

P.G.L. Porta Mana

<piero.mana@ntnu.no>

Draft of 22 December 2018 (first drafted 9 August 2018)

***

Note: Dear Reader & Peer, this manuscript is being peer-reviewed by you.

1 Introduction

The purpose of a neural network and some other machine-learning methods is to choose an output y given an input x and data relating the possible inputs and outputs. The way this choice is made and the way it is influenced by already-observed data make intuitive sense. As stated, the task above involves uncertainty and decision; it can therefore also be approached methodically using the plausibility calculus – the 'gold standard' as Srivastava, Hinton, et al. (2014) called it – and the principles of decision theory. It is interesting to see how the intuitive neural-net approach and the methodical one relate to each other.

This relation was first studied by Tishby, Levin, Solla (Tishby et al. 1989; Levin et al. 1990) and then brilliantly explored by MacKay (1992a,b) and others (1996; 1998).

2 ***

The study and grasp of the connection between neural nets and probability is beneficial to both. For neural nets, this connection *might* not lead to any practical improvements – because of scalability etc.*** – but it's very useful for understanding open problems, like the quantification of confidence of neural-network outputs [refs***], overtraining, or the choice of architectural constants. For example, the successful idea of using 'dropout' was also motivated (Hinton et al. 2012; Srivastava et al. 2014) by the 'mixing' of predictions typical of probability. For probability, this connection suggests new, efficient parametric families and approximation methods.

In this note I want to explain the connection between neural nets and some probability models. What's new in this note, compared with the 1990s studies cited above?

- I want to try to show this connection in a *geometric, visual* way.
- I extend those studies in two directions:
    1. including the decision-theoretic character of the problem. This leads to a different probabilistic interpretation of the error function of neural nets;
    2. basing the connection on so-called 'partially exchangeable models', to be discussed later. They give us a better understanding of overtraining, and are also connected with generative models [refs***].
- *Repetita iuvant*: knowledge of those studies seems to be forgotten or much diluted today.

This is what I do not purport to do:

– proving that the connection between neural nets and probability discussed here is unique. Even if it isn't unique, it's still insightful and useful;

– suggest that neural nets should be used 'more in accordance' with the probability calculus. The connection already shows that they are approximate probability calculations; their speed and power come from their approximate nature;

– [add here whatever else you'd like my intentions not to be.]

3 Introduction to partially exchangeable models

Consider a classification problem: given input $x \in \{1, \dots, M\}$, we want to predict output $y \in \{1, \dots, N\}$. Let's consider both discrete.

Two cases must be distinguished: whether every input has several possible outputs, or just one. The second case is a special instance of the first, and could be simply treated as the prediction of a 'function' $x \mapsto y$. But both cases are instances of inference with a *partially exchangeable* model.

Our assumptions are summarized in the proposition I. We assume that the probability of y, for each kind of input x, is exchangeable. Thus the probability for several pairs

$$\mathrm{p}(y^{(T)}, \dots, y^{(1)} \,|\, x^{(T)}, \dots, x^{(1)}, I) \tag{1}$$

must have a partially exchangeable form [refs]: for every kind of input x we consider the N relative frequencies $\boldsymbol{f}_x \coloneqq (f_{1|x}, \dots, f_{N|x})$ of the N possible outputs. The probability above is given by this integral:

$$\mathrm{p}(y^{(T)}, \dots, y^{(1)} \,|\, x^{(T)}, \dots, x^{(1)}, I) =$$

$$\int\!\cdots\!\int \Bigl[\prod_{x=1}^{M} \prod_{t}^{x^{(t)}=x} f_{y^{(t)}|x}\Bigr]\, \mathrm{p}(\boldsymbol{f}_1, \dots, \boldsymbol{f}_M \,|\, I)\, \mathrm{d}\boldsymbol{f}_1 \cdots \mathrm{d}\boldsymbol{f}_M. \tag{2}$$

The complicated look of this formula doesn't do justice to its simple and intuitive interpretation:

Choose a kind of input x. We're judging that the probability for a very large sequence of outputs generated with this input doesn't depend on their order. Suppose we knew the relative frequencies of each kind of output in this sequence, $(f_{1|x}, \dots, f_{N|x}) \eqqcolon \boldsymbol{f}_x$, and knew nothing else. Then out of symmetry reasons we should give a probability $f_{y|x}$ of observing outcome y at the next observation. This is just a 'drawing without replacement' problem. At the subsequent observations with the same input we give again the same probabilities to each y, because the sequence is so large that we approximate 'drawing without replacement' with 'drawing with replacement'. The probability for a sequence of outputs $y^{(1)}, \dots, y^{(T)}$ from the same input x is then

$$\mathrm{p}(y^{(T)}, \dots, y^{(1)} \,|\, \text{same input } x, \boldsymbol{f}_x, I) = f_{y^{(T)}|x} \times \cdots \times f_{y^{(1)}|x}. \tag{3}$$

Within a larger sequence of outputs from all possible inputs, the outputs $y^{(()}t)$ coming from input x are those for which $x^{(()}t) = x$. Thus we can write the formula above more generally as

$$\prod_{t}^{x^{(t)}=x} f_{y^{(t)}|x} \tag{4}$$

The above reasoning applies for each kind of input x. Thus the probability for a sequence of outputs coming from different inputs is the product of the probabilities above for all different x:

$$\mathrm{p}(\text{all outputs} \,|\, \text{inputs}, \boldsymbol{f}_1, \dots, \boldsymbol{f}_M, I) = \prod_{x=1}^{M} \prod_{t}^{x^{(t)}=x} f_{y^{(t)}|x}. \tag{5}$$

Now what if we don't know the relative frequencies $\boldsymbol{f}_x$, for any input? Then we assign a probability distribution over their possible valuesand use the law of total probability:

$$\mathrm{p}(\text{all outputs} \mid \text{inputs}, I) =$$
$$\int\cdots\int \mathrm{p}(\text{all outputs} \mid \text{inputs}, \boldsymbol{f}_1, \dots, \boldsymbol{f}_M, I)\, \mathrm{p}(\boldsymbol{f}_1, \dots, \boldsymbol{f}_M \mid I)\, \mathrm{d}\boldsymbol{f}_1, \dots, \mathrm{d}\boldsymbol{f}_M. \tag{6}$$

Substituting the explicit expression (5) in this formula we obtain formula (6). In summary,

$$\int\cdots\int \Biggl[\underbrace{\prod_{x=1}^{M}}_{\text{product over all inputs}} \prod_{t}^{x^{(t)}=x} \underbrace{f_{y^{(t)}|x}}_{\substack{\text{probability for outputs}\\ \text{from same input}}} \Biggr] \underbrace{\mathrm{p}(\boldsymbol{f}_1, \dots, \boldsymbol{f}_M \mid I)\, \mathrm{d}\boldsymbol{f}_1 \cdots \mathrm{d}\boldsymbol{f}_M}_{\text{uncertainty over all frequencies}} \ . \tag{7}$$

The possible frequencies give one input, $\boldsymbol{f}_x \equiv (f_{1|x}, \dots, f_{N|x})$, belong to the $(N-1)$-dimensional simplex

$$\Delta \coloneqq \bigl\{(f_1, \dots, f_N) \mid f_i \geqslant 0, \textstyle\sum_i f_i = 1\bigr\}, \tag{8}$$

and the collection of possible frequencies $(\boldsymbol{f}_1, \dots, \boldsymbol{f}_M)$ belongs to the M-fold Cartesian product Δ^M. From now on we denote $\boldsymbol{f} \coloneqq (\boldsymbol{f}_1, \dots, \boldsymbol{f}_M)$.

The probability for a new sequence of T' outputs given their inputs and given that we've learned a previous sequence of T input-output pairs is determined by Bayes's theorem:

$$\mathrm{p}(y^{(T')}, \dots, y^{(T+1)} \mid x^{(T')}, \dots, x^{(T+1)},\ y^{(T)}, x^{(T)}, \dots, y^{(1)}, x^{(1)},\ I) =$$
$$\frac{\displaystyle\int \Biggl[\prod_{x=1}^{M} \prod_{t=T+1,\dots,T'}^{x^{(t)}=x} f_{y^{(t)}|x}\Biggr] \mathrm{p}(\boldsymbol{f} \mid I)\, \mathrm{d}\boldsymbol{f}}{\displaystyle\int \Biggl[\prod_{x=1}^{M} \prod_{t=1,\dots,T}^{x^{(t)}=x} f_{y^{(t)}|x}\Biggr] \mathrm{p}(\boldsymbol{f} \mid I)\, \mathrm{d}\boldsymbol{f}}. \tag{9}$$

This formula is equivalent to (2) with and updated distribution for the frequencies:

$$\mathrm{p}(\boldsymbol{f} \,|\, y^{(T)}, x^{(T)}, \dots, y^{(1)}, x^{(1)},\ I)\,\mathrm{d}\boldsymbol{f} = \\ \frac{\bigl[\prod_{x=1}^{M} \prod_{t=1,\dots,T}^{x^{(t)}=x} f_{y^{(t)}|x}\bigr]\, \mathrm{p}(\boldsymbol{f} \,|\, I)}{\int \bigl[\prod_{x=1}^{M} \prod_{t=1,\dots,T}^{x^{(t)}=x} f_{y^{(t)}|x}\bigr]\, \mathrm{p}(\boldsymbol{f} \,|\, I)\,\mathrm{d}\boldsymbol{f}}\,\mathrm{d}\boldsymbol{f}. \tag{10}$$

The form of this updated distribution has important consequences for our learning process.

If the number of learned data is enough large compared with the numbers M, N of possible inputs and outputs and with the magnitude of the initial distribution for the frequencies, and if the latter is strictly positive, then the updated distribution becomes very peaked on the collection of relative frequencies $(\boldsymbol{q}_1, \dots, \boldsymbol{q}_M)$ of the learned outputs for all input values. This can be seen from the asymptotic expression in terms of the relative entropy (Kullback-Leibler divergence) D,

$$\mathrm{p}(\boldsymbol{f} \,|\, y^{(T)}, x^{(T)}, \dots, y^{(1)}, x^{(1)},\ I) \propto \exp\bigl[-\textstyle\sum_x T_x D(\boldsymbol{q}_x \,|\, \boldsymbol{f}_x)\bigr]\, \mathrm{p}(\boldsymbol{f} \,|\, I), \tag{11}$$

where T_x is the number of observations with input x, with $\sum_x T_x = T$.

If the number of learned data is small compared with the dimensions of the input and output spaces, then the initial distribution for the frequencies $\mathrm{p}(\boldsymbol{f} \,|\, I)\,\mathrm{d}\boldsymbol{f}$ greatly influence our inference (9). This distribution determines two important ***

4 Utility functions and probabilities

Utility of behaving as if proposition $B \in X$ is true given that proposition $A \in X$ is true: $c(B \,|\, A)$. Probability for A given D, I: $\mathrm{P}(A \,|\, D, I)$. Optimal decision is B that maximizes

$$\sum_A c(B \,|\, A)\, \mathrm{P}(A \,|\, D, I). \tag{12}$$

Now consider different probabilities for all A given the same data D: $\mathrm{P}(A \,|\, D, I')$. The decision B will still be the same if we use a new utility

$$c'(B \,|\, A) \coloneqq c(B \,|\, A)\, \frac{\mathrm{P}(A \,|\, D, I)}{\mathrm{P}(A \,|\, D, I')}. \tag{13}$$

So the same choice can be made with a different probability, if the utility is appropriately changed, provided $p(A \,|\, D, I') > 0$ for all A.

This leads to a slightly more general view than Tishby et al.'s (1989; 1990) and Mackay's (1992a,b)

✠ Point out that the joint parameter density allows us to make inferences 1. from data about one output to data about the same output; 2. from data about one output to data about a different output – *this* is generalization

✠ Point out that generalization (from one output to another) is *fully* determined by the form of the joint parameter prior

Thanks

. . . to Mari & Miri for continuous encouragement and affection, and to Buster Keaton and Saitama for filling life with awe and inspiration. To the developers and maintainers of LaTeX, Emacs, AUCTeX, Open Science Framework, Python, Inkscape, Sci-Hub for making a free and unfiltered scientific exchange possible.

Bibliography

('de *X*' is listed under D, 'van *X*' under V, and so on, regardless of national conventions.)

Barber, D., Bishop, C. M. (1998): *Ensemble learning in Bayesian neural networks*. In: *Neural networks and machine learning*. Ed. by C. M. Bishop (Springer), 215–237. https://www.microsoft.com/en-us/research/publication/ensemble-learning-in-bayesian-neural-networks/.

Hinton, G., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2012): *Improving neural networks by preventing co-adaptation of feature detectors*. arXiv:1207.0580.

Levin, E., Tishby, N., Solla, S. A. (1990): *A statistical approach to learning and generalization in layered neural networks*. Proc. IEEE **78**^{10}, 1568–1574.

MacKay, D. J. C. (1992a): *Bayesian interpolation*. Neural Comp. **4**^{3}, 415–447. http://www.inference.phy.cam.ac.uk/mackay/PhD.html.

— (1992b): *A practical Bayesian framework for backpropagation networks*. Neural Comp. **4**^{3}, 448–472. http://www.inference.phy.cam.ac.uk/mackay/PhD.html.

Neal, R. M. (1996): *Bayesian Learning for Neural Networks*. (Springer, New York). https://www.cs.toronto.edu/~radford/bnn.book.html.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014): *Dropout: a simple way to prevent neural networks from overfitting*. J. Mach. Learn. Res. **15**, 1929–1958.

Tishby, N., Levin, E., Solla, S. A. (1989): *Consistent inference of probabilities in layered networks: predictions and generalizations*. Int. Joint Conf. Neural Networks **1989**, II-403–II-409.