Hidden Markov Neural Network Paper Review

Anton C. Bagger, Jakob T.O. Danielsen, Jonatan E. Svendsen[1]

[1] *University of Copenhagen, 2100 Copenhagen, Denmark*

INTRODUCTION

Neural Networks (NN) have in recent years grown in prevalence to become the default within artificial intelligence. Adaptation into GPU-computing as well architectural optimization has furthered along this wide spread adaptation and applications include but are not limited to; language processing, image and speech recognition, medical diagnosis, financial modeling and autonomous driving[1–5].

A limitation in conventional NNs is brought about by the way that classical weight update strategies, such as stochastic gradient descent (SGD) work[6]. When new data is introduced, retraining the entire model from scratch is necessary which is computationally expensive. Furthermore, NNs are prone to having newly learned information that overwrites previously acquired knowledge. This is even more so the case when dealing with models training on data that dynamically updates over time, where a conventional NN might be prone to ignore temporal dependencies.

One way to mitigate this weakness is to implement alternative weight updating schemes such as variational inference, which provides a probabilistic framework for learning latent patterns in a way that still allows new data to be processed without requiring complete retraining. One way to implement variational inference is using Hidden Markov Models (HMM), which models the time-sensitive dependencies as transitions through latent states over time. One such Implementation of HMMs in NNs was carried out by Rimella and Whiteley in their paper titled *Hidden Markov Neural Networks*[7]. We will in this review cover and discuss their implementation, reasoning and results.

MODEL

Hidden Markov Neural Networks

Hidden Markov Neural Networks (HMNN)[7] offer a novel continual learning approach by treating network weights as hidden states in a HMM. A HMM is like a normal Markov Chain, but where the true states remain unobserved and the observed data is probabilistically linked to the hidden states. In a HMM we have a transition model that explains how the hidden states change from time t to $t+1$ and a emission model that describes the probability of the hidden states producing the observed data linking the unobservable process to the actual measure. Unlike conventional networks which have static weights, HMNNs uses this HMM approach to update weights from time t to $t+1$ by comparing predictions to the observed data. The transition model governs this update and assuming independent weight evolution using what is called factorial HMM, the joint transition probability is given by:

$$p(W_t \mid W_{t-1}) = \prod_{i=1}^{n} p\Big(W_t^{(i)} \mid W_{t-1}^{(i)}\Big), \tag{1}$$

where $W_t^{(i)}$ denotes the ith weight at time t. The emission model (e.g., using a Gaussian likelihood) is expressed as:

$$p(D_t \mid W_t) \propto \exp\Big(-\frac{1}{2\sigma^2}\|y_t - f(x_t; W_t)\|^2\Big). \tag{2}$$

where x_t is the input, y_t is the true output, $f(x_t; W_t)$ is the network prediction at time t, and σ^2 is the variance.

Bayesian Machine Learning

Bayesian inference provides a way to update a model's weights without completely discarding previous information—a common limitation of standard neural networks, which is the approach used in the HMNN. The core idea is encapsulated by Bayes' theorem:

$$p(W \mid D) \propto p(D \mid W)\, p(W), \tag{3}$$

where the *prior* (via the transition model) and the *likelihood* (via the emission model) yield a posterior that evolves with new data.

This model can be tuned with either a low-variance prior centering the weights near their current values for stability, while a higher variance allows the network to adapt more readily to new data.

Since computing the exact posterior is intractable for complex networks, HMNN's use Variational Inference(VI) to approximate it. *VI* is a filtering algorithm that in the context of a HMNN estimates the hidden state distribution at time t using all observations up to t. This is achieved via a two-step Bayesian update cycle: a prediction step propagates the previous state as a *prior*, and a correction step updates it into a *posterior* using new data.

Instead of calculating the exact posterior, they approximate it using *VI* by choosing a simple distribution $q_\theta(W)$ to approximate $p(W|D)$. Maximizing the Evidence Lower Bound (ELBO) minimizes the Kullback–Leibler divergence, which can be expressed as:

$$\mathrm{KL}\Big(q_\theta(W) \parallel p(W|D)\Big) = \log p(D) - \mathrm{ELBO}(q_\theta; D). \tag{4}$$

The update of $q_\theta(W)$ is implemented via *Bayes by Backprop*, which adapts the parameters as new data arrives.

EXPERIMENTS AND RESULTS

The proposed Hidden Markov Neural Neural Network(HMNN) was tested in various leaning scenarios to see how well the model performed. Five experiments were performed which tested different applications of the model.

The first problem looked at using variational DropConnect with a fully Gaussian HMNN on the MNIST dataset to potentially show a better variational approximation than Bayes by Backprop[8]. For this experiment the neural network was made with the vectorized image as input, two hidden layers with 400 rectified linear units. The training was done on 50 combinations of the parameters(γ^{υ}, ϕ, σ, c) and learning rate. From this tree models were selected depending and their accuracy of the γ^{υ} value compared to the validation set and for these models the accuracy was then found for the test show a higher accuracy for variational DropConnect than Bayes by Backprop.

The next experiment used the two moon dataset from "scikit-learn"[9]. For the experiment two scenarios were used: one in which the two moons are separated and one were they are overlapping. The two moons were rotated at each time t and by doing this the Gaussian HMNN was shown to successfully adapt to the evolving structure, comparing favorable to the chosen baselines. The overlapping case exhibited higher uncertainty in the transition region, highlighting HMNN's ability to quantify confidence in its predictions.

The third experiment looked at how well the HMNN worked with weights changing over time. Here a two dimensional logistic regression described the distribution and the weights follow an Ornstein-Uhlenbeck process. By comparing the predicted weights with the real function it was observed that the model accurately tracked the oscillating behavior of the parameters. demonstrating its capability to handle non-stationary data.

The fourth experiment aimed to test how well the HMNN balances learning new information while preserving previous acquired knowledge. The way this was tested was by using the MNIST data and randomly switching between two different labelers at each time step. So one labeler was the original from MNIST and the other was where everything was shifted one label. The accuracy of the result was compared to other models based on respectively; variational continual learning (VCL), without coreset; Elastic Weight Consolidation (EWC), with tuning parameter chosen with the validation set; Bayes by Backprop trained sequentially on the dataset; Bayes by Backprop on the full dataset; and an adaption of Kurle et al.'s work with the Ornstein–Uhlenbeck process[10]. The result showed that the HMNN outperformed these models in classification accuracy effectively adapting to label shift while maintaining stability over time.

Lastly the HMNN was assessed for how well it performed for time-series forecasting by predicting the next frame in a sequence extracted from a video of a waving flag[11]. The way the HMNN would do this was by using prior known frames to make valid predictions on future frames. The HMNN achieved the best result here making clearer predictions while effectively quantifying uncertainty, which was particularly noticeable in regions of high motion.

DISCUSSION AND CONCLUDING THOUGHTS

In the paper by Rimella and Whiteley, they proposed a novel hybrid model that integrated Bayesian machine learning with Factorial Hidden Markov Models. This model called a Hidden Markov Neural Network utilized Variational Inference via Bayes by Backprop and Varitional DropConnect to create a sequential and online model, that can easily adapt to new data, but also remember past information and without the need for expensive retraining. This model was tested against chosen baselines as MNIST and the Two moon dataset, which are popular benchmark datasets for neural networks. The experiments in the paper all lead to favorable comparisons. While it may be argued that the chosen baselines do not necessarily reflect optimized and polished frameworks, the same can be argued for the proposed HMNN model. In this way it can be concluded that the work done by Rimella and Whiteley serves as a case study on how variational inference through HMM can be effectively applied to mitigate the challenges of time-sensitive learning while maintaining adaptability to new data in Neural Networks. As the proposed model is not optimized and is kept at times simple to better translate as a proof of concept, as in the case of the use of vanilla gradient decent, it instead of polish favors simplicity, opening the door for creativity in further implementation and architecture. Further work on Hidden Markov Neural Networks (HMNNs) could explore several directions. One direction could be to develop a better analysis of the variational approximation quality and to quantify the rate of error accumulation over time. Another important area could to implement smoothing algorithms that refine posterior estimates and to investigate strategies for more robust hyperparameter tuning. Additionally, extending the HMNN framework to more complex network architectures, such as recurrent and convolutional neural networks, could broaden its range of applications, and exploring alternative variational inference techniques and regularization strategies beyond variational DropConnect may further enhance performance and computational efficiency. Finally, replacing or augmenting vanilla gradient descent with more advanced optimization methods could improve training stability and scalability.

[1] Y. Liu and M. Zhang, *Computational Linguistics*, 2018, **44**, 193–195.

[2] Y.-T. Zhou, R. Chellappa, A. Vaid and B. Jenkins, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1988, **36**, 1141–1151.

[3] F. Amato, A. López, E. M. Peña-Méndez, P. Vaňhara, A. Hampl and J. Havel, *Journal of Applied Biomedicine*, 2013, **11**, 47–58.

[4] M. Lam, *Decision Support Systems*, 2004, **37**, 567–581.

[5] D. Feng, L. Rosenbaum and K. Dietmayer, 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 3266–3273.

[6] S. ichi Amari, *Neurocomputing*, 1993, **5**, 185–196.

[7] L. Rimella and N. Whiteley, *arxiv*, 2025.

[8] L. Deng, *IEEE Signal Processing Magazine*, 2012, **29**, 141–142.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, *the Journal of machine Learning research*, 2011, **12**, 2825–2830.

[10] R. Kurle, B. Cseke, A. Klushyn, P. van der Smagt and S. Günnemann, International Conference on Learning Representations, 2020.

[11] A. Basharat and M. Shah, *2009 IEEE 12th International Conference on Computer Vision*, 2009, 1941–1948.