Wake-Sleep Energy Based Models for Continual Learning

Supplementary Material

8. Appendix

8.1. Theorem for the wake-sleep EBM algorithm

Theorem 1: *In WS-EBM, with a dynamic loss function, the optimal model parameters w through gradient optimization can be approximately obtained by minimizing the linear combination of loss function used in wake cycle(αL_{wc}) and sleep cycle(βL_{sc}) as*:

$$w_{k+1} = w_k - \eta(\nabla(\alpha L_{wc} + \beta L_{sc})) \tag{12}$$

where α and β are the number of wake and sleep cycles respectively, and η is the learning rate.

8.1.1 Proof of Theorem 1

Let original model parameters at iteration k be w_k. and assume the same learning rate for both wake and sleep cycles. Moreover, without loss of generality assume the number of wake(α) and sleep cycles(β) to be 1 since these are scalars. after gradient optimization in the wake cycle with loss function L_{wc}, the model parameters would be updated as follows:

$$w'_k = w_k - \eta(\frac{\partial L_{wc}}{\partial w_k}) \tag{13}$$

Feeding these intermediate parameters to sleep cycle optimization, we have

$$w_{k+1} = w'_k - \eta(\frac{\partial L_{sc}}{\partial w'_k}) \tag{14}$$

Combining equations 13 and 14 we have:

$$w_{k+1} = w_k - \eta(\frac{\partial L_{wc}}{\partial w_k} + \frac{\partial L_{sc}}{\partial w'_k}) \tag{15}$$

By change of variables, equation 15 can be written as

$$w_{k+1} = w_k - \eta(\frac{\partial L_{wc}}{\partial w_k} + \frac{\partial L_{sc}}{\partial w_k}\frac{1}{\frac{\partial w'_k}{\partial w_k}}) \tag{16}$$

Now partially differentiating equation 13 wrt to w_k, we have

$$\frac{\partial w'_k}{\partial w_k} = 1 - \eta\frac{\partial^2 L_{wc}}{\partial w_k^2} \tag{17}$$

Substituting this in equation 16 we have

$$w_{k+1} = w_k - \eta(\frac{\partial L_{wc}}{\partial w_k} + \frac{\partial L_{sc}}{\partial w_k}\frac{1}{1-\eta(\frac{\partial^2 L_{wc}}{\partial w_k^2})}) \tag{18}$$

Assuming $\eta << 1$(usually in range of $10e-3$) and $\frac{\partial^2 L_{wc}}{\partial w_k^2} << 1$, we can use the geometric series sum rule for $\frac{1}{1-x}$, which would give us:

$$w_{k+1} = w_k - \eta(\frac{\partial L_{wc}}{\partial w_k} + \frac{\partial L_{sc}}{\partial w_k}(1+\eta\frac{\partial^2 L_{wc}}{\partial w_k^2} + O(\eta\frac{\partial^2 L_{wc}}{\partial w_k^2})^2) \tag{19}$$

Neglecting the smaller terms, we can simplify equation 19 as

$$\begin{aligned} w_{k+1} &= w_k - \eta(\frac{\partial L_{wc}}{\partial w_k} + \frac{\partial L_{sc}}{\partial w_k}) \\ \therefore w_{k+1} &= w_k - \eta(\nabla(L_{wc} + L_{sc})) \end{aligned} \tag{20}$$

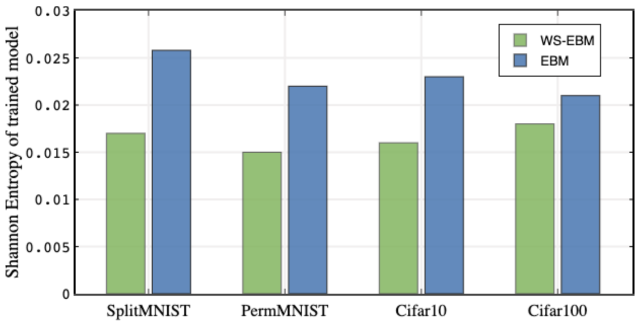

Figure 4. WS-EBM has lower Entropy across all the benchmark datasets. It's desirable for a trained model, to have lower Shannon Entropy, which correlates well with generalization.

8.2. Entropy analysis

In [5], authors empirically confirm that the sharpness of a deep learning loss landscape around local minima indeed correlates with the generalization abilities of the model. The authors performed comparative studies of different existing sharpness measures to guide the optimization process toward the well-generalizing regions of the loss landscape. From a broad family of different measures of sharpness, we consider Shannon entropy [40], decreasing the value of which indicates a flatter optimum. Shannon Entropy can be defined as

$$\mu_{\text{entropy}} = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{\kappa} f_{w^*}(x_i)[j]\log(f_{w^*}(x_i)[j]) \tag{21}$$

where w^* are model parameters, $f_{w^*}(x_i)[j]$ denotes the probability of j_{th} class predicted by the deep learning model f_{w^*} for input data x, and κ be the total number of classes. Figure 4 confirms the lower entropy values for the WS-EBM for all the datasets except, suggesting that its improved accuracy can be linked to the local geometry of the loss.

Method	Test Accuracy(%)	BWT(%)	Shannon Entropy	Cosine Similarity
EBM	8.68	-19.23	0.0379	0.0365
WS-EBM	13.28	-1.50	0.0353	0.0103

Table 4. Evaluation Metrics on synthetic classification dataset. WS-EBM outperforms EBM in every metric.

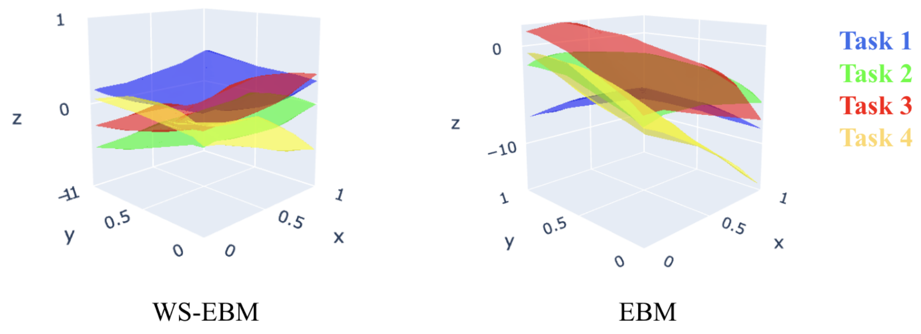

Figure 5. Evolution of Energy Surface for Data in Task 1 when the model is successively trained on task 1,2,3,4. The x and y axis represent the 2-dimensional data. The z-axis represents the energy values. The figure shows that WS-EBM shows less perturbations, smaller range of variations and a smoother energy landscape as compared to EBM in class incremental setting.

8.3. Qualitative analysis of energy surface on a synthetic classification problem

In this section, we attempt to visualize the evolution of energy surface while training on sequential tasks to gain more insight into the effect of wake-sleep cycles in training EBM. Here we perform a qualitative analysis of the training process of EBMs over a simple toy classification problem curated for class incremental setting.

Consider a 2-dimensional x-y plane with 4 quadrants. These quadrants act as 4 disjoint tasks. Each quadrant can be divided into two equal halves by line $y = x$ for quadrants 1 and 3 and $y = -x$ for quadrants 2 and 4. We assign ground-truth classes or labels to each of these octants as shown in Figure 6, such that each quadrant becomes a task of binary classification. Now we sample 1000 random data points from a uniform distribution $\mathcal{U}(0, 1)$ in each quadrant such that 500 points belong to each octant class. In total, we have 4000 uniformly distributed 2-dimensional data values for which we can plot the energy surface during training in class incremental setting. Plotting actual benchmark datasets like SplitMNIST, and Cifar100 is not possible since the data points are high dimensional(> 3). The model is trained in a class incremental setting by sequentially learning on data from task 1 to task 4, such that data from previous tasks is not available while training the current task. The objective of the model at hand is to predict the growing number of classes without forgetting the previous classes. The architecture of the EBM is a 2 layered Neural Network with 400 linear units and is kept constant for both WS-EBM and EBM. Since this dataset is small, we keep the iteration per task as 5 and the number of wake and sleep cycles are kept the same as 2 and 10 respectively, same as described in section Theoretical Analysis.

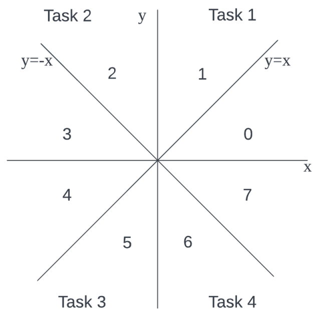

Figure 6. Synthetic data for visualizing energy surfaces

Plotting of Energy Surfaces: To show variations in the energy surface we plot the energy values for data in quadrant 1 as predicted by the model trained on subsequent tasks in a class incremental manner. Figure 5 shows the evolution of energy surface for data points belonging to the first task, as the model learns incrementally on task $1, 2, 3, 4$. It can be seen that the energy surfaces in the WS-EBM strategy are stable as compared to the training regime without wake-sleep cycles. We see sharp changes in the landscape of the energy surface when the model learns new tasks. Further, the range of variations in this regime is larger than the decoupled strategy, demonstrating that the decoupled strategy interferes less with pior tasks while learning new classes. Further, the evaluation metrics in Table 4 demonstrate the superiority of WS-EBM over classical EBMs.

Dataset	Method	Square-Square Loss	Hinge Loss	Log Loss	Sq-Exp. Loss($\gamma = 2.3$)
SplitMNIST	EBM	53.82 ± 0.06	49.17 ± 0.02	51.26 ± 0.06	49.17 ± 0.02
	WS-EBM	56.42 ± 0.31	52.38 ± 0.03	52.67 ± 0.98	54.32 ± 0.31
PermMNIST	EBM	84.97 ± 1.22	86.27 ± 0.22	83.54 ± 0.05	72.27 ± 0.23
	WS-EBM	88.32 ± 0.19	87.51 ± 0.07	87.61 ± 0.23	74.55 ± 0.01
Cifar10	EBM	37.26 ± 0.09	37.13 ± 0.08	38.26 ± 0.09	38.25 ± 0.81
	WS-EBM	40.12 ± 0.17	39.36 ± 0.81	39.11 ± 0.03	39.60 ± 0.23
Cifar100	EBM	28.39 ± 0.42	29.88 ± 0.65	29.88 ± 0.34	30.19 ± 0.03
	WS-EBM	30.71 ± 0.27	30.03 ± 0.08	30.14 ± 0.03	31.08 ± 0.11

Table 5. Comparison of WS-EBM and EBM with General Margin Loss Function. The Average Accuracy(%) is computed over 10 runs. WS-EBM gives a better performance across all the datasets in terms of all the evaluation metrics.

8.4. Analysing WS-EBM with Margin Loss Functions

We applied the WS-EBM algorithm with **Generalised Margin Loss Functions** [28] and compared it with the typical EBM without wake-sleep cycles. These loss forms maintain some sort of a positive margin m to create an energy gap between the correct answer($E(w, y_i, x_i)$) and the most offending incorrect answer($E(w, \bar{y_i}, x_i)$).These loss functions are of the form:

$$\mathcal{L}(w, x_i, y_i) = L((E(w, x_i, y_i), E(w, x_i, \bar{y_i})) \tag{22}$$

where L are different forms of loss functions. Some loss functions falling under margin losses can be described as follows:

Hinge loss: It is defined as $L_{hinge}(w, y_i, x_i) = max(0, m + E(w, y_i, x_i) - E(w, \bar{y_i}, x_i))$ where m is the positive margin. Here the difference between energies of the correct answer and the most offending incorrect answer is penalized linearly when greater than $-m$.

Log loss: $L_{log}(w, y_i, x_i) = log(1 + e^{E(w,y_i,x_i) - E(w,\bar{y_i},x_i)})$. It's a smoother or softer version of hinge loss, with an infinite margin.

Square-Square loss: $L_{sq-sq} = E(w, y_i, x_i)^2 + m - E(w, \bar{y_i}, x_i))^2$. It penalizes large values of energy for correct answers and small values for $E(w, \bar{y_i}, x_i)$ above the margin m. It treats the energy of correct and most offending incorrect answers quadratically and separately.

Square exponential loss: $L_{sq-exp}(w, y_i, x_i) = E(w, y_i, x_i)^2 + \gamma(e^{-E(w,\bar{y_i},x_i)})$. where γ is a positive constant. Although very similar to *square-square loss* the contrastive term is exponential instead of quadratic and pushes the energy of incorrect answers to infinity with a decreasing force.

We empirically found that square-square loss with a margin $m = 1$ gave the best improvement with WS-EBM as seen in Table 5. We observe that the square-square loss pushes the correct answer energy down towards zero and pushes down the incorrect answer energies above m. Therefore, it is only suitable for energy functions that are bounded below by zero, notably in architectures whose output module measures some sort of distance. Since we are using ReLU activation, the energy values are bound by zero. The hyper-parameter m has been empirically chosen as 1.