Fisher Information and Online Continual Learning

The approach of Online Curvature-Aware Replay

Edoardo Urettini

Unversity of Pisa and Scuola Normale Superiore

Table of contents

Backgorund

Relative Entropy

Relative Entropy, also called **KL divergence**, is a statistical distance between two probability distributions. It measures the **excess entropy of assuming the distribution Q when the true one is P**.

$$D_{KL}(P||Q) = \mathrm{E}\left[log(\frac{P(x)}{Q(X)})\right] = -\sum_{x \in \mathcal{X}} P(x) log \frac{Q(x)}{P(X)} \tag{1}$$

In machine learning, we commonly use it for classification problems. The **cross-entropy** is defined as:

$$H(P, Q) = H(P) + D_{KL}(P||Q) \tag{2}$$

But The KL divergence is also perfectly valid for regression problems with continuous distributions.

Relative Entropy - 2

Relative Entropy represents the **expected value of the log-likelihood ratio statistic** when the real distribution is P.

Usually, in machine learning, we assume a **parameterized distribution** p_θ and we want to find the "real" parameters θ^* given by the data.

This optimization process is done in the parameter space Θ. In this space, each point corresponds to a different vector of parameters that defines **a different distribution**. So the distance between two points in the parameter space is defined as the **statistical distance between the two distributions: the KL-divergence**.

$$D(\theta, \theta^{'}) = D_{KL}(p_\theta(x)||p_{\theta'}(x)) = \sum_{x \in \mathcal{X}} p_\theta(x) \frac{p_\theta(x)}{p_{\theta'}(x)}$$

Fisher Information as Curvature

We can now see how our distance changes under infinitesimal perturbations using **Taylor Expansion of our distance**. Assume we are at a minimum of the KL: our θ are equal to the optimal θ^*.

For ease the computation, we take the continuous version of the KL divergence:

$$D(\theta^*, \theta) = \frac{1}{2}(\theta - \theta^*)(\frac{\partial^2}{\partial\theta_i \partial\theta_j} D(\theta^*, \theta))_{\theta=\theta^*}(\theta - \theta^*) + o((\theta - \theta^*)^2)$$

Note that the **KL-divergence at the minimum is 0** and **the first derivative at the same point is also 0**.

Fisher Information as Curvature - 2

We then see that **the Hessian of the divergence** has the information on how the KL-divergence changes for infinitesimal perturbations.

$$\frac{\partial^2}{\partial\theta_i\partial\theta_j} D_{KL}(p(x;\theta^*), p(x;\theta))_{\theta=\theta^*} = \tag{3}$$

$$= \frac{\partial^2}{\partial\theta_i\partial\theta_j}(\int p(x;\theta^*) log \frac{p(x;\theta^*)}{p(x;\theta)} dx)_{\theta=\theta^*} = \tag{4}$$

$$= -(\int p(x;\theta^*) \frac{\partial^2}{\partial\theta_i\partial\theta_j} log p(x;\theta) dx)_{\theta=\theta^*} \tag{5}$$

We computed this at convergence because to compute the KL-divergence **we need to know the real parameter θ^***. This is true only at optimum.

Fisher Information as Curvature - 3

When everything is evaluated at $\theta^* = \theta$ we obtain that **the curvature of the KL** is equal to:

$$\mathbf{F}(\theta) = -\mathbf{E}[\frac{\partial^2}{\partial\theta_i \partial\theta_j} logp(x;\theta)|\theta]$$

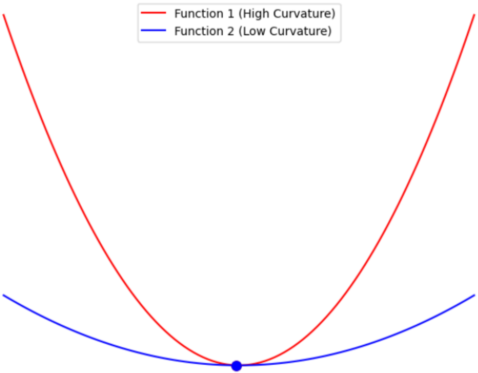
This is defined **FISHER INFORMATION**. It describes the **amount of information that a random variable carries about the parameters**.

A higher Fisher Information means that I will need fewer observations to learn the right parameter.

Finally, it is a measure of the **sensibility of the KL-divergence** to perturbations in the parameters.

Fisher Information as Curvature - 4

When at optimum, the gradient is zero. The curvature information tells us **how much the KL changes when we move**. High curvature means a high KL increase.



Online Continual Learning (OCL)

Additional constraints and desiderata from CL:

1. **Online Training** No access to whole task data. Only a small mini-batch can be processed for a limited time;
2. **Anytime Inference** The model should be ready for inference at any point in time;
3. **Continual Stability** The model must be stable at any point in time, instead of only at the task boundaries;
4. **Fast Adaptation** The model must be able to learn quickly from new data.

Stability Gap

The standard approach for OCL is **Replay**.

But, when continually evaluated, Replay methods suffer from **Stability gap**.

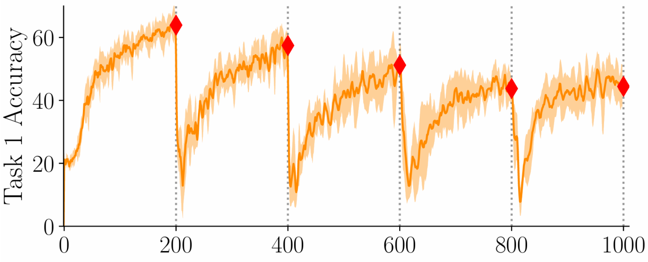


Figure 1: Accuracy on first task when using ER in task-incremental scenario

Online Curvature-Aware Replay

Optimization problem

We approach OCL as a sequence of **local optimization problems** using **Replay data**.

At every step in time, we receive a few observations N_t from the current data distribution and a few samples B_t from our limited buffer $\mathcal{B}$. The optimization problem for time t is:

$$\begin{aligned} \min_{\delta_t} \quad & \hat{KL}(y_{N_t} \,||\, f_{w_t}(x_{N_t})) + \hat{KL}(y_{B_t} \,||\, f_{w_t}(x_{B_t})) \\ \text{subject to} \quad & \frac{1}{2}||\delta||_2^2 \leq \epsilon, \end{aligned} \tag{6}$$

First-Order Solution

The KL-divergence can be approximated by Taylor-expansion:

$$\hat{KL}(y_D \,||\, f_w(x_D)) \approx \hat{KL}(y_D \,||\, f_{w=w_0}(x_D)) + \nabla_t^T \delta_t + \delta_t^T \mathbf{H}_t \delta_t$$

Note

∇_t is equivalent to the "usual" loss gradient when using cross-entropy, MSE, or negative log-likelihood losses.

Solving the problem with a first-order approximation:

$$\delta_t^* = -\frac{1}{\lambda}(\nabla_{N_t} + \nabla_{B_t}) \tag{7}$$

This is the **standard Experience Replay**. **At task boundary, the first-order information of the previous tasks is very small**.

Second-Order Solution and Stability Constraint

If at task boundary $\nabla_{B_t} \approx 0$ and $\nabla_{N_t} \gg \nabla_{B_t}$, we can instead use the second-order approximation:

$$\delta_t^* = -(\mathbf{H}_{N_t} + \mathbf{H}_{B_t} + \lambda \mathbf{I})^{-1}(\nabla_{N_t} + \nabla_{B_t}),$$

This is equivalent to Newton optimization (with a damping term). It improves optimization, but we can put an **explicit stability constraint**. The new problem becomes:

$$\begin{aligned} \min_{\delta} \quad & \hat{KL}(y_{N_t} \,||\, f_{w_t}(x_{N_t})) + \hat{KL}(y_{B_t} \,||\, f_{w_t}(x_{B_t})) \\ \text{subject to} \quad & \hat{KL}(f_{w_{t-1}}(x_{B_t}) \,||\, f_{w_t}(x_{B_t})) \leq \rho \\ & \frac{1}{2}||\delta||_2^2 \leq \epsilon. \end{aligned}$$

The New solution

The terms of the Taylor expansion of the stability constraint around w_{t-1}:

- $\hat{KL}(f_{w_{t-1}}(x_{B_t}) \,||\, f_{w_{t-1}}(x_{B_t})) = 0$
- $\nabla_{w=w_{t-1}} \hat{KL}(f_{w_{t-1}}(x_{B_t}) \,||\, f_{w_t}(x_{B_t})) = 0$
- $\mathbf{H}_{w=w_0} \hat{KL}(f_{w_{t-1}}(x_{B_t}) \,||\, f_{w_t}(x_{B_t})) = \mathbf{F}_{B_t}$

Hence, the new solution is:

$$\delta_t^* = -(\mathbf{H}_{N_t} + \mathbf{H}_{B_t} + \lambda \mathbf{F}_{\mathbf{B_t}} + \tau \mathbf{I})^{-1} (\nabla_{N_t} + \nabla_{B_t}),$$

λ depends on ρ and controls the **importance of the stability**. τ depends on ϵ and act as a **Tikhonov damping**.

Approximations

Computing and inverting two Hessians and a Fisher matrix is not feasible. Note that

$$\mathbf{H}_t = \mathbf{H}_{w=w_{t-1}} \hat{KL}(y_{D_t} \,||\, f_w(x_{D_t})) \tag{8}$$

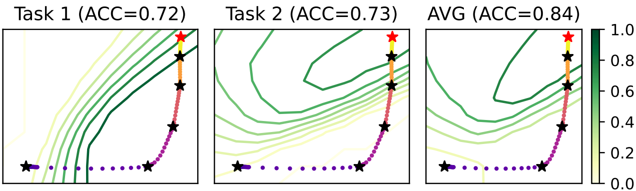
both for $D_t = N_t$ and $D_t = B_t$. If we assume that our current model $f_{w_{t-1}}$ is a **good representation of the new and buffer data**, we get $\mathbf{H}_{N_t} = \mathbf{F}_{N_t}$ and $\mathbf{H}_{B_t} = \mathbf{F}_{B_t}$ and a **a single Fisher matrix is required**. We hence obtain the **Online Curvature-Aware Replay** update:

$$\delta_t^* = -\alpha(\mathbf{F}_{N_t} + (1+\lambda)\mathbf{F}_{\mathbf{B_t}} + \tau\mathbf{I})^{-1}(\nabla_{N_t} + \nabla_{B_t}).$$

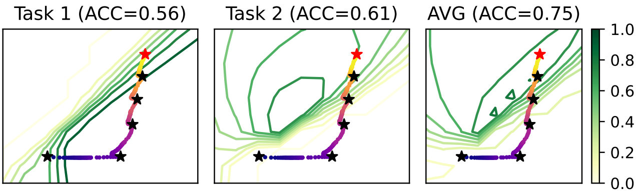
The Fisher matrix is obtained **weighting more the buffer data** and is approximated by **Kronecker-factored Approximate Curvature**.

Results

Training trajectories



(a) OCAR 2D projection of the learning trajectory.



(b) ER 2D projection of the learning trajectory.

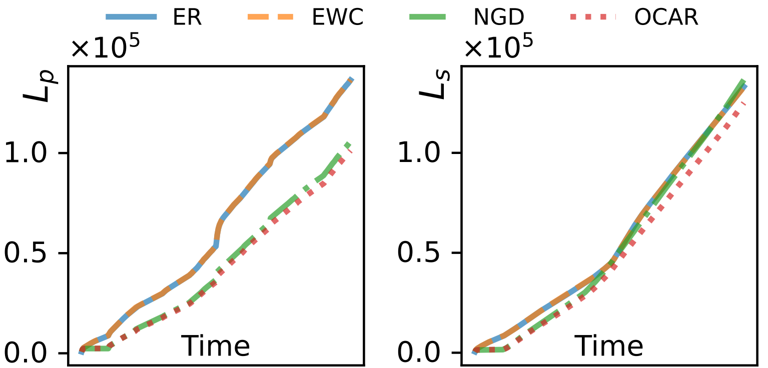


Figure 3: *Left*: L_p Cumulative loss of single batches. *Right*: L_s Cumulative loss measured on all previous data of the stream. The model is linear and the problem a linear regression.

Split-Cifar 100

Method	Acc ↑	AAA^{val} ↑	WC-Acc$^{\text{val}}$ ↑	Probed Acc ↑
i.i.d	35.3 ± 1.5	-	-	45.8 ± 0.6
ER	28.2 ± 1.2	36.6 ± 2.0	12.5 ± 0.6	44.9 ± 0.9
GDumb	18.5 ± 0.5	-	-	-
AGEM	3.1 ± 0.2	10.4 ± 0.6	2.9 ± 0.3	18.7 ± 0.8
ER+LwF	30.4 ± 0.8	39.2 ± 2.0	15.3 ± 0.9	44.4 ± 0.8
MIR	29.4 ± 1.9	33.1 ± 3.2	11.6 ± 1.6	43.4 ± 0.7
RAR	28.2 ± 1.4	38.2 ± 1.6	14.9 ± 0.7	42.3 ± 0.9
DER++	29.3 ± 0.9	37.5 ± 2.5	13.4 ± 0.7	44.0 ± 0.8
ER-ACE	29.9 ± 0.6	38.5 ± 1.8	14.9 ± 0.9	42.4 ± 0.6
SCR	28.3 ± 0.8	42.1 ± 2.1	20.3 ± 0.4	37.0 ± 0.3
OnPro	31.7 ± 1.2	36.6 ± 2.5	12.2 ± 1.1	-
OCM	30.9 ± 0.7	33.3 ± 1.9	14.9 ± 0.4	-
LPR	33.3 ± 0.6	42.5 ± 0.5	19.3 ± 0.3	-
OCAR	$\mathbf{34.9} \pm 0.6$	$\mathbf{48.2} \pm 1.2$	$\mathbf{25.0} \pm 1.1$	$\mathbf{46.2} \pm 0.6$
OCAR-DER++	34.3 ± 1.1	46.8 ± 1.7	25.4 ± 0.8	46.0 ± 0.8
OCAR-ACE	$\underline{35.6} \pm 1.2$	$\underline{48.7} \pm 1.7$	$\underline{26.5} \pm 0.4$	44.1 ± 0.7

Split-TinyImagenet

Method	Acc ↑	AAA^{val} ↑	WC-Acc$^{\text{val}}$ ↑	Probed Acc ↑
i.i.d	26.5 ± 0.6	-	-	34.3 ± 0.5
ER	21.2 ± 0.6	33.9 ± 1.7	15.2 ± 0.5	35.6 ± 0.6
GDumb	13.1 ± 0.4	-	-	-
AGEM	2.6 ± 0.2	7.3 ± 0.5	2.6 ± 0.2	23.3 ± 0.6
ER+LwF	22.7 ± 1.1	34.4 ± 2.4	17.0 ± 0.7	33.8 ± 0.9
MIR	21.3 ± 0.8	31.0 ± 1.8	15.2 ± 0.5	33.0 ± 0.4
RAR	15.7 ± 0.9	27.8 ± 2.8	10.1 ± 0.9	29.8 ± 0.9
DER++	22.9 ± 0.5	34.2 ± 4.0	16.3 ± 0.3	31.5 ± 0.9
ER-ACE	$\mathbf{23.6} \pm 0.7$	35.0 ± 1.5	16.8 ± 0.7	34.2 ± 0.3
SCR	16.9 ± 0.4	30.7 ± 1.5	12.3 ± 0.5	22.5 ± 0.4
OnPro	17.1 ± 1.5	24.2 ± 0.4	8.00 ± 0.8	-
OCM	20.6 ± 0.6	24.8 ± 1.1	10.9 ± 0.5	-
LPR	23.1 ± 0.2	34.9 ± 0.4	16.2 ± 0.2	-
OCAR	21.7 ± 1.0	$\mathbf{38.3} \pm 1.4$	$\mathbf{17.4} \pm 0.6$	$\mathbf{38.3} \pm 0.6$
OCAR-ACE	$\underline{25.6} \pm 0.4$	$\underline{39.8} \pm 2.0$	$\underline{21.5} \pm 0.9$	34.7 ± 0.3

Online CLEAR

Table 1: Results on Online CLEAR (10 Tasks) domain incremental setting. 2000 Buffer size. Best in bold.

Method	Online CLEAR (10 Tasks)		
	Acc $\uparrow$	AAA^{val} $\uparrow$	WC-Acc $^{\text{val}}$ $\uparrow$
ER	63.1 ± 0.7	58.9 ± 0.8	47.7 ± 1.6
LPR	65.2 ± 0.9	63.5 ± 1.0	62.6 ± 0.7
OCAR(Ours)	$\mathbf{75.3} \pm 0.8$	$\mathbf{73.9} \pm 0.5$	$\mathbf{70.3} \pm 0.5$

Single task Training Times

Table 2: Training Time for the First Task on Split-CIFAR-100.

Method	Training Time (seconds)
ER	14
ER + LWF	15
MIR	31
ER-ACE	17
DER	17
RAR	72
SCR	131
LPR	213
OCAR(Ours)	38

Conclusions

- We approach OCL as a **sequence of local and independent optimization problems**;
- At minimum, the **alert of current task tends to zero** while **second-order information is valid**;
- Using an explicit **stability constraint** introduce the use of the Fisher Information;
- Using approximations, we need to compute **a single Fisher**, with theoretical similarities to **Natural Gradient Descent**;
- **OCAR** has solid empirical results while keeping training times acceptable.

S.-i. Amari.
***Information geometry and its applications*, volume 194.**
Springer, 2016.

T. M. Cover.
***Elements of information theory*.**
John Wiley & Sons, 1999.

M. De Lange, G. van de Ven, and T. Tuytelaars.
Continual evaluation for lifelong learning: Identifying the stability gap.
arXiv preprint arXiv:2205.13452, 2022.

R. Grosse and J. Martens.
A kronecker-factored approximate fisher matrix for convolution layers.
In *International Conference on Machine Learning*, pages 573–582. PMLR, 2016.

F. Kunstner, P. Hennig, and L. Balles.
Limitations of the empirical fisher approximation for natural gradient descent.
Advances in neural information processing systems, 32, 2019.

R. Pascanu and Y. Bengio.
Revisiting natural gradient for deep networks.
arXiv preprint arXiv:1301.3584, 2013.