Latent Space Evolution under Incremental Learning with Concept Drift (Student Abstract)

Charles Bourbeau[1], **Audrey Durand**[1,2,3]

[1]Département d'informatique et de génie logiciel, Université Laval, Québec (QC), Canada
[2]Département de génie électrique et de génie informatique, Université Laval, Québec (QC), Canada
[3]Canada-CIFAR AI Chair, Mila, Québec (QC), Canada
charles.bourbeau.1@ulaval.ca

Abstract

This work investigates the evolution of latent space when deep learning models are trained incrementally in non-stationary environments that stem from concept drift. We propose a methodology for visualizing the incurred change in latent representations. We further show that classes not targeted by concept drift can be negatively affected, suggesting that the observation of all classes during learning may regularize the latent space.

Introduction

Supervised deep learning requires vast and rich data sources to capture the inherent diversity of a target domain and thus achieve good generalization. However, the creation such large datasets is often very expensive, motivating a more practical alternative: deploy a model trained on an smaller initial dataset, and keep collecting data to improve the model over its lifetime. For this strategy to be successful, it is crucial to enable the integration of incoming data samples into the model's existing knowledge base, thus continually increasing its performance over the entire observed domain.

Incremental Learning consists in pursuing the training of a model on new data without accessing previous data. Such sequential learning raises the famous stability-plasticity dilemma, wherein stability refers to the retainment of previous knowledge and plasticity the acquirement of new knowledge (Elwell and Polikar 2011). Deep neural networks lie on the plastic end of the spectrum, as the distributed nature of learned features renders them very sensitive to the integration of new information, while also enabling their impressive generalization power. Furthermore, when the data distribution is non-stationary, continually integrating new information interferes with previously acquired knowledge, which leads to forgetting (French 1997).

In real-world scenarios, one cannot assume stationarity between the distributions encountered during training and at deployment. For instance, predictive models used in decision systems can interact in intricate ways with their environment through their predictions. As an example from the medical setting, being able to predict and prevent some disease before it occurs would make the label frequency of said disease decrease overtime. Such alterations to a data distribution are captured under the phenomenon of *concept drift*, defined as a change in the statistical properties of a target domain over time in an arbitrary way, i.e. $\exists t : P_t(X, y) \neq P_{t+1}(X, y)$, where (X, y) denote input/output pairs. In this work, we focus on virtual concept drift, a type of drift relevant to many real-world applications, where distribution shift occurs only in the input distribution $P(X)$ without affecting the input/output relationship $P(y|X)$ (Lu et al. 2019). Hence, virtual concept drift does not affect the decision boundary, but only latent representations.

In the following section, we investigate the performance impact of incrementally learning from an environment which undergoes virtual concept drift. Using the well-known MNIST dataset (Lecun et al. 1998), we artificially create a virtual concept drift problem. We first show that in a non-stationary environment, the overall performance is not an appropriate indicator for monitoring the quality of model updates. More importantly, we show that even classes not targeted by concept drift are negatively affected, suggesting that some classes may serve as regularizers when learning the representation of other classes, especially if they share common features. These results are supported by our final qualitative analysis of the latent representation evolution.

Incremental Learning Under Concept Drift

Given an initial target domain D, we artificially introduce virtual concept drift by splitting D using two custom joint probability distributions $P_1(X, y)$ and $P_2(X, y)$, ensuring that $P_1(X) \neq P_2(X)$ and $P_1(y|X) = P_2(y|X)$. We then sample from D using both distributions to create two disjoint and equally-sized domains D_1 and D_2, with $D_i \sim P_i(X, y)$. To simulate a scenario akin to many real-world applications, we craft the first domain D_1 as a balanced pre-training dataset, and the second domain D_2 as new observations of the target domain D that become available through time. To ensure that D_2's input distribution differs from that of D_1, we introduce class imbalance in D_2 by specifying an under-sampling factor μ and under-sampled class y_-. Formally, the two label distributions are defined as $P_1(y_i) = \frac{1}{K}\forall y_i$ and $P_2(y_i) = \frac{\alpha}{K}\forall y_i \neq y_-, P_2(y_-) = \frac{\alpha}{\mu K}$, with α being the normalizing constant. *Note that* $\mu = 1$ *represents the absence of concept drift.* In our experiments, datasets for domains D_1 and D_2 each contain 25000 samples. The bal-

μ	y_-	Accuracy evolution (%)	
		Overall	Class-Wise (y_-)
1	-	$+0.32 \pm 0.45$	N/A
10	0	-0.32 ± 0.88	-4.06 ± 3.22
10	1	-0.31 ± 0.46	-4.78 ± 1.65
10	2	-1.12 ± 1.14	-16.53 ± 6.73
10	3	-1.02 ± 0.58	-16.80 ± 5.48
10	4	-1.14 ± 0.51	-20.66 ± 7.36
10	5	-0.53 ± 0.79	-14.20 ± 6.29
10	6	-0.24 ± 0.87	-8.27 ± 4.51
10	7	-0.99 ± 1.20	-13.79 ± 5.84
10	8	-0.90 ± 1.09	-19.55 ± 9.31
10	9	-1.09 ± 1.00	-20.36 ± 7.04

Table 1: Evolution of accuracy during incremental learning with virtual concept drift (averaged over 10 runs).

anced dataset for domain D_1 is fixed for all experiments. In total, we generate 11 configurations of D_2: one with $\mu = 1$ (no concept drift) as a baseline, and 10 with $\mu = 10$ using each digit once as the under-sampled class y_-. For each configuration, training is first performed on D_1, then continued on D_2 without access to previous data. Given D_i, we train a deep neural network f_{θ_i}, where $\hat{y} = f_{\theta_i}(x) = C_{\theta_i} \circ E_{\theta_i}(x)$ denotes the prediction for input x. The intermediate layers E_{θ_i} encode the input into a latent space of dimensionality L, extracting task-specific features: $x \in \mathbb{R}^{H \times W \times C} \mapsto z \in \mathbb{R}^L := E_{\theta_i}(x)$. The final classification layer C_{θ_i} specifies the decision boundary: $z \in \mathbb{R}^L \mapsto \hat{y} \in \mathbb{R}^K := C_{\theta_i}(z)$.

We use a multi-layer perceptron with two hidden layers of width 20, optimized over a cross entropy loss using SGD and learning rate of 0.05. We learn over D_1 for 5 epochs before observing D_2 in batches of size 32. Each batch in D_2 is seen only once and the model is updated using a single optimization step per batch. Table 1 shows the results of the incremental learning experiments. We observe that the overall performance appears to remain stable, while the accuracy of the under-represented class decreases drastically. These results demonstrate that simply measuring the metrics of interest on the whole label space is not sufficient when incrementally learning after deployment.

Qualitative Analysis

In order to visualise the change in latent representations learned by E_{θ_1} and E_{θ_2}, we propose a new decoder-based method. Optimizing over a reconstruction task from latent space Z back to the input space, we train a decoder network g which mirrors the encoder's architecture, with $z \in \mathbb{R}^L \mapsto \hat{x} \in \mathbb{R}^{H \times W \times C} := g(z)$. The decoder's training inputs are encoded using E_{θ_1}, which is freezed during the reconstruction task. By uncoupling the reconstruction task from the target task, we force g to use only task-specific features in order to generate an approximation of the input. The visualization set for class y_k contains of the set of inputs X_k with label y_k. Then, we project the set of samples X_k into latent space using both encoders and compute the mean representation for each one: $z_{k_i} = \mathrm{mean}(E_{\theta_i}(X_k))$. The final reconstruction is then generated as: $\hat{x}_i = \mathrm{normalize}(g(z_{k_i}))$. To assess the evolution due to incremental learning, we visualise *i)* the difference between both reconstruction: $\hat{x}_{diff} = \mathrm{normalize}(\hat{x}_2 - \hat{x}_1)$, and *ii)* the difference blended over the first reconstruction to extrapolate the representation's long term evolution: $\hat{x}_{proj} = \mathrm{blend}(\hat{x}_1, \hat{x}_{diff}, \alpha = 0.5)$.

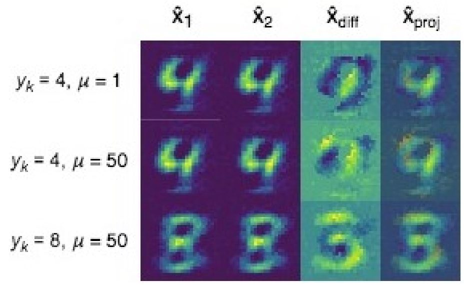

Figure 1: Examples of representation evolution ($y_- = 4$).

Figure 1 shows examples of changes in representation for an input of class 4 when there is no concept drift (top), the same input under concept drift of $y_- = 4$ (middle), and an input of class 8 under concept drift of $y_- = 4$ (bottom). We observe that the representation of class 4 input shifts towards the representation of class 9 under concept drift. While this is expected, the behaviour highlighted on the bottom row is more surprising: we observe a representation shift for the input corresponding to digit 8 towards a representation resembling digit 3. Errors on class 8 increase from 1.3% for 8.2%. We posit that the latent representations learned through a balanced pre-training should be regarded as a fragile ecosystem, as learned features are distributed throughout the entire network and shared between different classes.

Conclusion

Our results confirm that we can indeed expect confusion to be introduced within the latent representations of the under-sampled class. More importantly, we discovered this confusion to be generalized over the whole label space, and not limited to the under-sampled class. This motivates further investigations to ensure a safe usage of models undergoing incremental learning in non-stationary environments.

References

Elwell, R.; and Polikar, R. 2011. Incremental Learning of Concept Drift in Nonstationary Environments. *IEEE Transactions on Neural Networks*, 22(10): 1517–1531.

French, R. 1997. Pseudo-recurrent Connectionist Networks: An Approach to the 'Sensitivity-Stability' Dilemma. *Connection Science*, 9(4): 353–380.

Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Lu, J.; Liu, A.; Dong, F.; Gu, F.; Gama, J.; and Zhang, G. 2019. Learning under Concept Drift: A Review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12): 2346–2363.