GoldiProx: Prioritized learning on points that are learnable, worth learning, and not yet learned

Soren Mindermann[1]*, Muhammed Razzak[1]*, Winnie Xu[2, 3]*

Andreas Kirsch[1], Mrinank Sharma[1], Adrien Morisot[3], Aidan N. Gomez[1,2,3], Sebastian Farquhar[1], Jan Brauner[1], Yarin Gal[1]

OATML — [1] University of Oxford — [2] University of Toronto — [3] co:here

Contributions

We developed an information theoretic approach to efficient model training in the large data regime.

Maximal information gain about the labels of a validation set:

- *Pointwise predictive information gain* as acquisition function.
- *GoldiProx Selection*: online batch selection via a proxy.

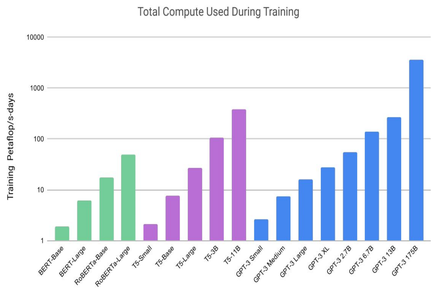
The *reducible held-out cross-entropy loss* as an easily implementable and novel objective function.

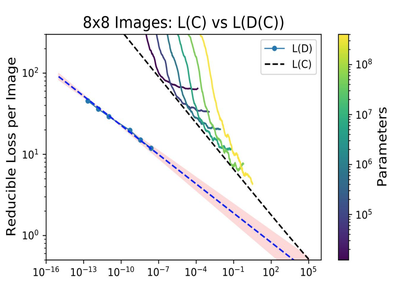
Prioritizing learnable, but informative data relevant to the eval task.

A Transition from Mini-Batch to Mini-Epoch

Training is expensive and most data is <u>uninformative</u>:

1. <u>Redundant:</u> already learned, “easy” points
2. <u>Ambiguous:</u> non-learnable, “noisy” and unpredictable labels
3. <u>Atypical:</u> not worth learning (or prioritizing), “uncertain” labels





Common methods of data selection have shortfalls:

1. <u>Online batch selection</u>**:** train only on “hard” points by sampling without replacement within in epoch to select a training batch

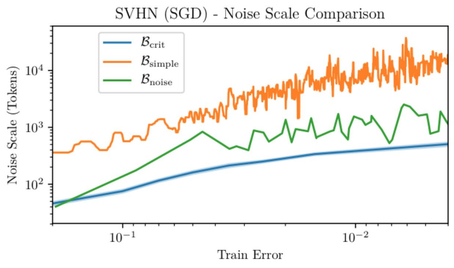
   → Skips redundant points, but expensive forward pass

2. <u>Curriculum learning:</u> prioritize “easy” points with low noise before training on all points with equal weight

   → Better convergence but fails to de-prioritize redundant points

3. <u>Active learning:</u> selects points where model is uncertain of label

   → Selects informative, non-noisy, but test-time irrelevant points



Label-Aware Training Objective

Expected Predictive Information Gain (EPIG): select informative unlabeled points by computing the info gain / mutual information between Y^{val} and Y since labels y for x are unavailable:

$$\mathrm{I}[\mathbf{Y}^{\text{val}}; Y \mid \mathbf{x}^{\text{val}}, x, \mathcal{D}_t] = H[\mathbf{Y}^{\text{val}} | \mathbf{x}^{\text{val}}, \mathcal{D}_t] - H[\mathbf{Y}^{\text{val}} | \mathbf{x}^{\text{val}}, \mathcal{D}_t, x, Y]$$

Pointwise Predictive Information Gain via **pointwise mutual information (pmi):** remove the need to compute expectations over Y and Y^{val} since the labels y^{val} and y are available, i.e. a label-aware mutual info metric. Difference of pointwise conditional entropies (PCE):

$$\mathrm{pmi}[\mathbf{y}^{\text{val}}; y \mid \mathbf{x}^{\text{val}}, x, \mathcal{D}_t] = \underbrace{\mathrm{h}[\mathbf{y}^{\text{val}} \mid \mathbf{x}^{\text{val}}, \mathcal{D}_t]}_{\text{PCE prior to (x, y)}} - \underbrace{\mathrm{h}[\mathbf{y}^{\text{val}} \mid \mathbf{x}^{\text{val}}, \mathcal{D}_t, x, y]}_{\text{PCE after seeing (x, y)}}$$

Reducible Held-Out Cross-Entropy Loss

Measures the uncertainty reduction about the validation labels y^{val} due to observing a point in training (x, y),

$$\underbrace{\mathrm{h}[y \mid x, \mathcal{D}_t]}_{\substack{\text{untrained model} \\ \text{cross-entropy}}} - \underbrace{\mathrm{h}[y \mid x, \mathbf{x}^{\text{val}}, \mathbf{y}^{\text{val}}]}_{\substack{\text{irreducible cross entropy} \\ \text{(loss of proxy and validation} \\ \text{set, i.e. } D_t << D_{val})}}$$

so that a filtered dataset D_t represents a viable sequence of training examples that can be used to train a new model while also maximizing the validation likelihood of the pre-trained proxy model.

GoldiProx Selection: Improved Training Efficiency

Input: Initial parameters θ^0_{small} and θ^0_{big}, learning rate η, small model $\mathrm{p}(y \mid x, \theta_{\text{val}})$ trained on validation set, batch size $|b|$, large batch size $|B| > |b|$.

for i in `training set` **do**
  `IrreducibleLoss[i]` $\leftarrow L(y_i, \mathrm{p}(y_i \mid x_i, \theta_{\text{val}}))$
end for
`Sequence` $\leftarrow$ `[]`

Train a smaller model normally with negative log likelihood objective

for $t = 1, 2, \ldots$ **do** ▷ Train large model
  Load $b_t \leftarrow$ `Sequence[t]`
  $g_t \leftarrow$ mini-batch gradient on b_t for θ^t_{big}
  $\theta^{t+1}_{\text{big}} \leftarrow \theta^t_{\text{big}} - \eta g_t$
end for

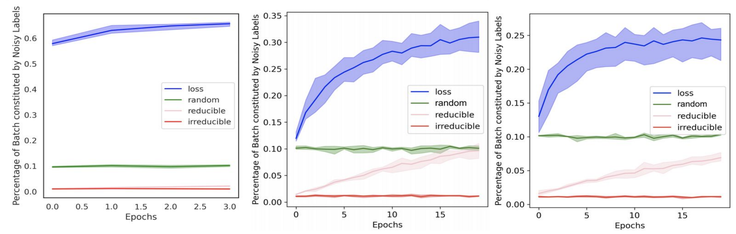
Take a gradient step on a mini-batch, based on reducible loss

for $t = 1, 2, \ldots$ **do** ▷ Select data with small model
  Randomly select a large batch B_t of size $|B|$.
  $\forall i \in B_t$, compute `Loss[i]`, the loss of point i given parameters θ^t_{small}
  $\forall i \in B_t$, compute `ReducibleLoss[i]` $\leftarrow$ `Loss[i]` $-$ `IrreducibleLoss[i]`
  $b_t \leftarrow$ top-$|b|$ samples in B_t in terms of `ReducibleLoss`.
  $g_t \leftarrow$ mini-batch gradient on b_t for θ^t_{small}
  $\theta^{t+1}_{\text{small}} \leftarrow \theta^t_{\text{small}} - \eta g_t$
  Append b_t to `Sequence`.
end for

Select cheaply with a small proxy, train larger model on filtered sequence

Performant Models with Less Data and Iterations

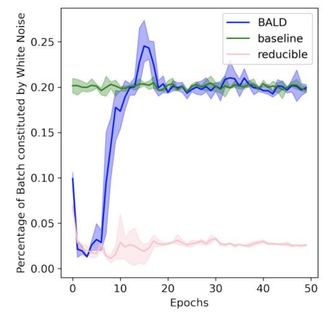
High loss points are the points mislabelled by their noise distribution



(a) QMNIST (b) CIFAR10 (c) CINIC10

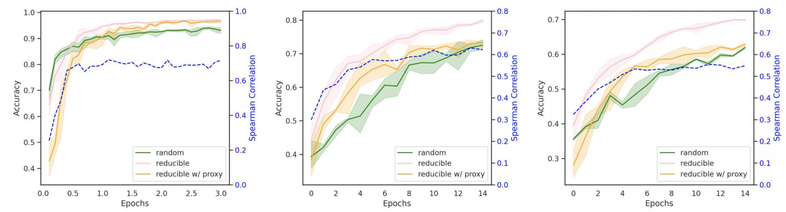
Reduced Uncertainty for the Prediction Task

Bayesian Active Learning with Disagree-ment (*BALD*) selects 20% of noisy samples perhaps since information can be gained from low support regions of input space. This is problematic in high kurtosis input distributions, as data that is unlikely to appear at test time is not worth learning. Reducible loss triumphs.



High Correlation between Proxy and Reducible Model

Effective transfer of reducible loss training with small proxy to larger models of different architectures. See rank correlations.



(a) QMNIST (b) CIFAR10 (c) CINIC10

Model	Num of Params.	FLOPS	Model	Num of Params.	FLOPS
3-layer MLP with 128 hidden units	135 k	136 k	Small CNN	0.538 M	0.019 G
			ResNet-18	11.17 M	0.557 G
3-layer MLP with 512 hidden units	932 k	935 k			

This implies that a point learned by a converged model (i.e. trained on D, the original dataset) may still be informative to a model trained on D_t, the prioritized sequence of examples.

References

[1] Kirsch, A., van Amersfoort, J., & Gal, Y. (2019). BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning. arXiv.org.

[2] Loshchilov, Ilya, and Frank Hutter. "Online batch selection for faster training of neural networks." arXiv preprint arXiv:1511.06343 (2015).

[3] Houlsby, Neil, et al. "Bayesian active learning for classification and preference learning." arXiv preprint arXiv:1112.5745 (2011).

[4] Kirsch, Andreas, Tom Rainforth, and Yarin Gal. "Active Learning under Pool Set Distribution Shift and Noisy Data." arXiv preprint arXiv:2106.11719 (2021).