The Memory Perturbation Equation: Understanding Model's Sensitivity to Data

Peter Nickl†, Lu Xu†*, Dharmesh Tailor‡*, Thomas Möllenhoff†, Mohammad Emtiyaz Khan†

†RIKEN Center for AI Project, Japan, ‡University of Amsterdam, Netherlands, *equal contribution

NEURAL INFORMATION PROCESSING SYSTEMS

Derived using Bayesian Principles

$$\text{MPE:}\quad \hat{\boldsymbol{\lambda}}_t^{\setminus\mathcal{M}} - \boldsymbol{\lambda}_t = \rho \sum_{j\in\mathcal{M}} \tilde{\mathbf{g}}_j(\boldsymbol{\lambda}_t)$$

Deviation in natural parameters

Sum over natural gradients of removed examples

We propose to estimate sensitivity by taking an **opposite** step of the Bayesian Learning Rule [1]

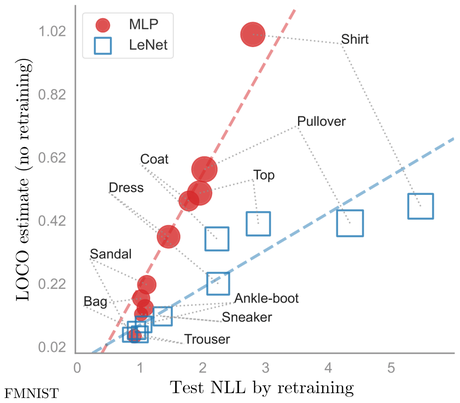
The opposite step is equivalent to simply dividing the posterior of a conjugate model by the removed examples, giving rise to MPE

Sensitivity estimation for a wide-variety of algorithms

Algorithm	Update	Sensitivity
Newton's method	$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \mathbf{H}_{t-1}^{-1}\nabla\mathcal{L}(\boldsymbol{\theta}_{t-1})$	$\mathbf{H}_{t-1}^{-1}\nabla\ell_i(\boldsymbol{\theta}_t)$
Online Newton (ON) [1]	$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \rho\,\mathbf{S}_t^{-1}\nabla\mathcal{L}(\boldsymbol{\theta}_{t-1})$	$\mathbf{S}_t^{-1}\nabla\ell_i(\boldsymbol{\theta}_t)$
ON (diag.+minibatch) [1]	$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \rho\,\mathbf{s}_t^{-1}\cdot\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}_{t-1})$	$\mathbf{s}_t^{-1}\cdot\nabla\ell_i(\boldsymbol{\theta}_t)$
iBLR (diag.+minibatch) [2]	$\mathbf{m}_t \leftarrow \mathbf{m}_{t-1} - \rho\,\mathbf{s}_t^{-1}\cdot\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}_{t-1})$	$\mathbf{s}_t^{-1}\cdot\nabla\ell_i(\boldsymbol{\theta}_t)$
RMSprop/Adam	$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \rho\,\mathbf{s}_t^{-\frac{1}{2}}\cdot\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}_{t-1})$	$\mathbf{s}_t^{-\frac{1}{2}}\cdot\nabla\ell_i(\boldsymbol{\theta}_t)$
SGD	$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \rho\,\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}_{t-1})$	$\nabla\ell_i(\boldsymbol{\theta}_t)$

$\boldsymbol{\theta}_t$ is the parameter at iteration t, ρ is the learning rate, $\mathcal{L}$ is the loss, ℓ_i is the loss of example i, $\mathbf{H}$ is the Hessian, $\mathbf{s}$ and $\mathbf{S}$ are scale vector and matrix, and $\mathbf{m}$ is the mean of the posterior.

Predicting the effect of class removal

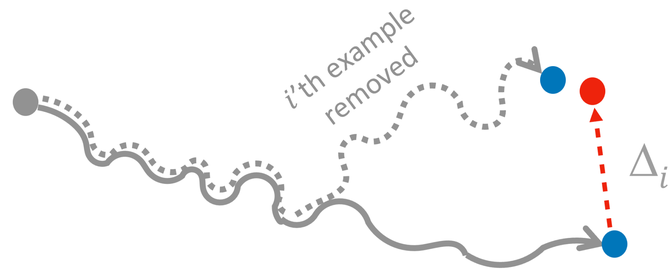


FMNIST

Leave-one-class-out estimation (LOCO) can predict the decrease in test performance when a class C is removed

$$\sum_{i\in\mathcal{C}} \ell\left(y_i, f_i(\theta_t^{\setminus\mathcal{C}})\right) \approx \sum_{i\in\mathcal{C}} \ell\left(y_i,\, f_i(\theta_t) + v_{it}e_{it}\right)$$

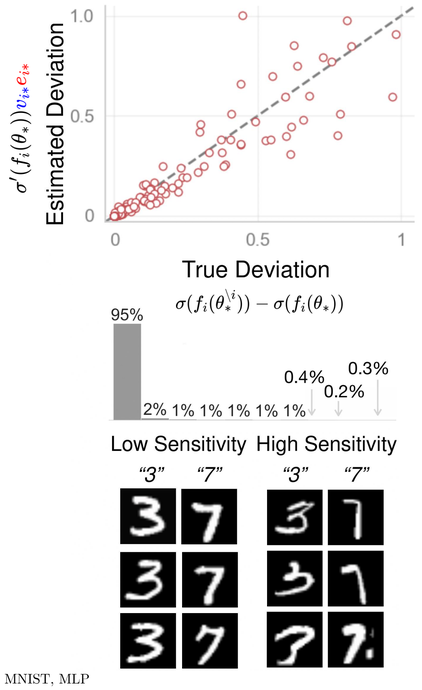
The proposed MPE generalizes influence measures [3] to a wide-variety of algorithms (including their iterations)



Sensitivity $\propto$ **Prediction variance** $\times$ **Prediction error**

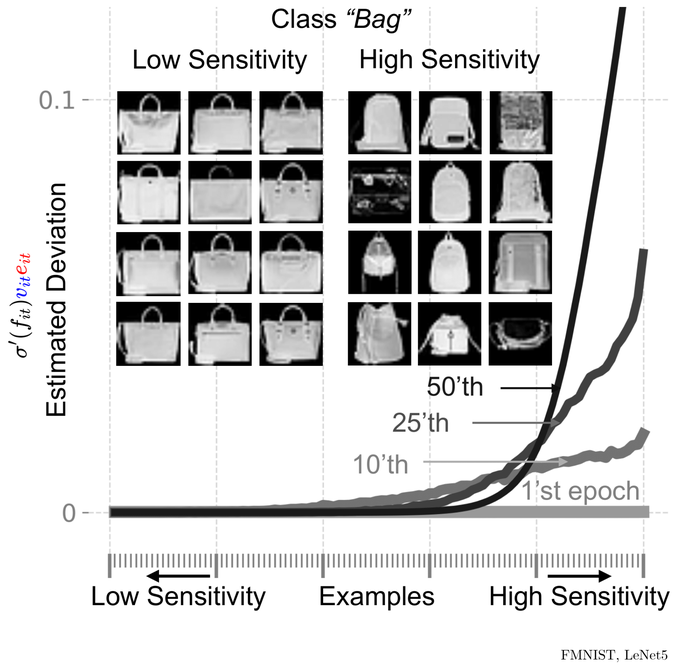
$\boldsymbol{\Delta_i}$ $\qquad$ $\boldsymbol{v_i}$ $\qquad$ $\boldsymbol{e_i}$

The model is highly sensitive to only a small fraction of examples



MNIST, MLP

Such high-sensitivity examples can be used to characterize the model's memory

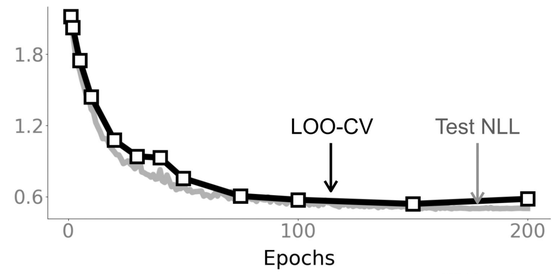


FMNIST, LeNet5

Sensitivities can be used to accurately predict generalization

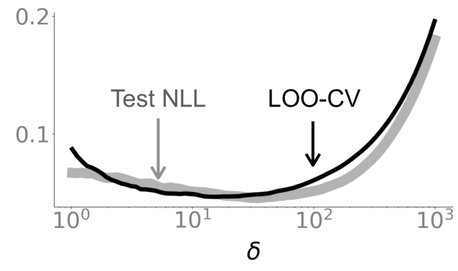
- Predicting generalization on unseen test data **during training** based on training data alone
- Evaluation of a LOO cross-validation (CV) loss
- No retraining required

$$\sum_{i=1}^{N} \ell\left(y_i, f_i(\theta_t^{\setminus i})\right) \approx \sum_{i=1}^{N} \ell\left(y_i,\, f_{it} + v_{it}e_{it}\right)$$



CIFAR-10, ResNet20

- **Model selection** of L2-regularization param. δ
- Evaluating LOO cross-validation **after training**
- MPE avoids retraining of N models per δ



MNIST, MLP

References:

[1] Khan and Rue, The Bayesian Learning Rule. JMLR 2023.
[2] Lin et al., Handling the Positive-Definite Constraint in the Bayesian Learning Rule. ICML 2020.
[3] Koh and Liang, Understanding Black-box Predictions via Influence Functions. ICML 2017.