Training-Free Neural Active Learning with Initialization-Robustness Guarantees

ICML International Conference On Machine Learning

Apivich Hemachandra [1] Zhongxiang Dai [1] Jasraj Singh [2] See-Kiong Ng [1] Bryan Kian Hsiang Low [1]

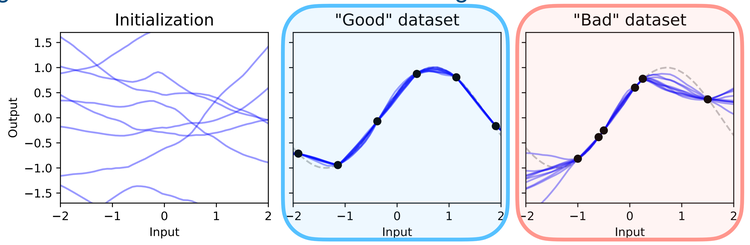
[1] Department of Computer Science, National University of Singapore [2] School of Computer Science and Engineering, Nanyang Technological University

NUS National University of Singapore

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Initialization-Robustness

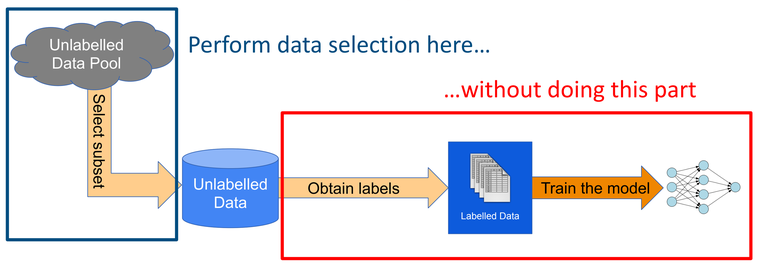
- Parameters of neural networks are initialized randomly
- As a result, the **neural network output after training with gradient descent will vary**
- A good dataset makes the neural network output vary less with respect to its (random) parameter initialization - a dataset that guarantees this is better for model training



- Initialization-robustness is important in **safety-critical applications**
- This is because if a different parameter initialization leads to very different model outputs, then we could trust the model less

Active Learning

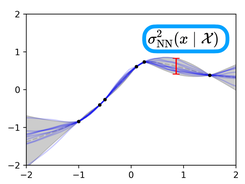
- Constructing a training set is expensive, especially obtaining labels
  - An expert may be asked to provide data labels, which can require a high cost and can take a long time to obtain the labels
- Can we select a training set in a way that we can still guarantee **initialization-robustness** and **low generalization error**?



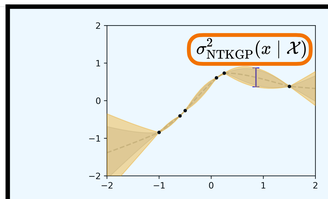
The algorithm should:

- Require little or no initial labelled data in the beginning
- Be able to select unlabelled points in batches (batched active learning regime)
- Not require training of any actual neural networks
- Not impose any requirements on the neural network training or architecture (e.g. not require us to use Bayesian neural networks)

Neural Network Behaviors



True neural network output variance requires model training and so it is expensive to compute



Gaussian Process with NTK as the kernel function (NTKGP)
Efficient to compute and approximate!

We can show that:

- The difference between **neural network output variance** and the **NTKGP variance** is **bounded**
  - This means we can use NTKGP to approximate the true neural network behaviors

$$\exists\, \alpha, \beta > 0 \quad \text{s.t.} \quad \left|\sigma^2_{\mathrm{NN}}(x \mid \mathcal{X}) - \alpha \cdot \sigma^2_{\mathrm{NTKGP}}(x \mid \mathcal{X})\right| \leq \beta$$

Depends on **size of training data**, **maximum NTK value**, **minimum NTK eigenvalue**, and **neural network architecture**

- A **low NTKGP variance** implies a **low generalization error**
  - This means training points that result in a low NTKGP variance would also result in trained models that is more accurate

$$|f^*(x) - f(x; \mathrm{train}(\theta_0))| \leq \zeta \cdot \sigma_{\mathrm{NTKGP}}(x \mid \mathcal{X})$$

Depends on **label noise**, and **how "easy" to learn the underlying function**

EV-GP Criterion

$$\alpha_{\mathrm{EV}}(\mathcal{X}) = \frac{1}{|\mathcal{X}_T|} \sum_{x \in \mathcal{X}_T} \left[\sigma^2_{\mathrm{NTKGP}}(x \mid \emptyset) - \sigma^2_{\mathrm{NTKGP}}(x \mid \mathcal{X})\right]$$

"Average model output variance w.r.t. model initialization"

The criterion is:
- Label-independent (can be used on large batches)
- Submodular (can approx. w/ sequential greedy)

Model Selection

- In reality, we don't know which NN architecture fits data best
- We propose to **find the best NN architecture while doing AL**, based on the expected MSE loss by bootstrapping, where we select the best model after each AL batch

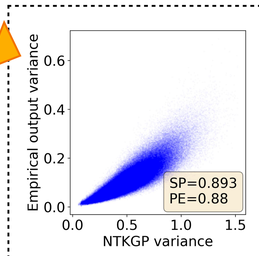
$$\hat{\alpha}_{M,\mathcal{D}_T}(f; \mathcal{D}) \triangleq \mathop{\mathbb{E}}_{\theta_0 \sim \mathrm{init}(\theta)} \left[\ell(\mathcal{D}_T, \mathrm{train}(\theta_0))\right]$$

$$\alpha_M(f; \mathcal{D}) = - \mathop{\mathbb{E}}_{\mathcal{D}_T \subset \mathcal{D}; |\mathcal{D}_T| = \kappa} \left[\hat{\alpha}_{M,\mathcal{D}_T}(f; \mathcal{D} \setminus \mathcal{D}_T)\right]$$

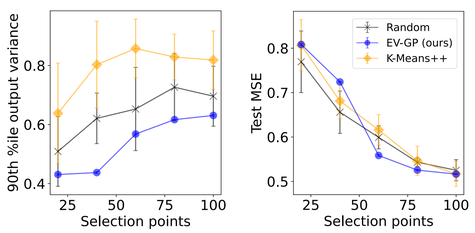
EV-GP+MS Algorithm

$(\mathcal{X}_\mathcal{L}, \mathbf{y}_\mathcal{L}) \leftarrow (\mathcal{X}_0, \mathbf{y}_0)$
Pick an initial model $f^* \in \mathcal{M}$
repeat
  // Phase 1: Data selection
  for b iterations **do**
    $x^* \leftarrow \arg\max_{x \in \mathcal{X}_U \setminus \mathcal{X}_\mathcal{L}} \alpha_{\mathrm{EV}}(\mathcal{X}_\mathcal{L} \cup \{x\}; f^*)$
    $\mathcal{X}_\mathcal{L} \leftarrow \mathcal{X}_\mathcal{L} \cup \{x^*\}$
  end for
  Select active set with sequential greedy
  Query the unlabelled points in $\mathcal{X}_\mathcal{L}$ for the labels $\mathbf{y}_\mathcal{L}$
  // Phase 2: Model selection
  $f^* \leftarrow \arg\max_{f \in \mathcal{M}} \alpha_M(f; (\mathcal{X}_\mathcal{L}, \mathbf{y}_\mathcal{L}))$
  Select best model given queried points
Repeat until done
until budget exhausted
return $(\mathcal{X}_\mathcal{L}, \mathbf{y}_\mathcal{L}), f^*$

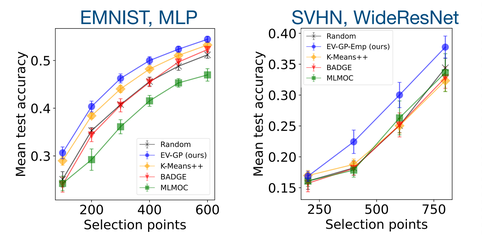
Experiments



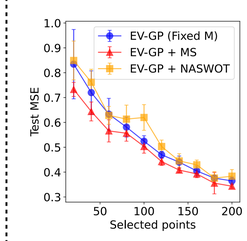
NTKGP vs NN variance



AL on regression problems (Robot Kin. w/ MLP)



AL on classification problems



Model selection

Acknowledgements

This research/project is supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-018) and by A*STAR under its RIE2020 Advanced Manufacturing and Engineering (AME) Programmatic Funds (Award A20H6b0151). We would like to thank Yao Shu for his valuable inputs to our paper.