Course Project Presentation

Deven Joshi (210326)
Lakshvant Balachandran (210557)
Rishabh Arijeet (210842)
Udhav Varma (211120)

IIT Kanpur

Table of Contents

Introduction

- Machine learning models often exhibit unexpected failures when deployed due to deficiencies in the training data, such as:
  - Incorrect labels
  - Outliers
  - Selection biases
- The paper proposes a technique to repair a model on-the-fly as new failures are encountered.
- The technique has two main steps:
  1. Identify the training examples (failure causes) that contributed most to the observed failures.
  2. Adapt the model to fix the failures by erasing the memories of the identified failure causes.

Introduction

- The problem is formulated as a Bayesian continual learning problem, estimating the counterfactual posterior without the failure causes.
- The framework is general, handling failures due to issues in both input data and labels.

Table of Contents

The model repairment problem

- Consider a prediction model $p(y|x,\theta)$ - the probability distribution of output y given an input x. Suppose $p(\theta|D)$ is the posterior distribution given the training data D (which is assumed to be i.i.d).
- For test inputs x^* we use the PPD to infer the predicted label y^*

$$p(y^*|x^*,D) = \int p(y^*|x^*,\theta)p(\theta|D)d\theta$$

- Suppose the set of incorrect predictions from the test set is denoted by the failure set $\mathcal{F} = \{(x_f^{(n)}, y_f^{(n)})\}_{n=1}^{N_f}$.
- Our goal is to repair the model by removing bad examples (caused by low quality inputs).

The model repairment problem

- The plan is to do the model repairment in two steps:
  1. **Cause Identification:** We identify a set of poor data points $\mathcal{C}$, which are believed to have contributed to $\mathcal{F}$ the most.
  2. **Treatment:** Adapt the model to perform better on the failure set, without affecting its accuracy on the remaining test data (Using continual unlearning).
- In the cause identification step, we need to find out how much each subset of training samples is affecting the result of $\mathcal{F}$. An effective measure for this value would be the change in the posterior predictive distribution of $\mathcal{F}$

$$r(\mathcal{C}) = \log p(\mathcal{F}|\mathcal{D} \setminus \mathcal{C}) - \log p(\mathcal{F} \setminus \mathcal{D})$$

Table of Contents

Cause Identification

- Since the naive way of finding optimal $\mathcal{C}$ would be computationally prohibitive, we use the following predictive approach.
- Using Taylor expansion and approximation, we can approximately represent $r(\mathcal{C})$ as

$$\hat{r}(\mathcal{C}) = \mathbb{E}_{p(\theta|\mathcal{D})}[\log p(\mathcal{C}|\theta)] - \mathbb{E}_{p(\theta|\mathcal{D},\mathcal{F})}[\log p(\mathcal{C}|\theta)]$$

- Using the i.i.d. assumption, we can represent this expression as sum of $\hat{r}(z)$ over individual points z in $\mathcal{C}$, where $\hat{r}(z)$ is given by

$$\hat{r}(z) = \mathbb{E}_{p(\theta|\mathcal{D})}[\log p(z|\theta)] - \mathbb{E}_{p(\theta|\mathcal{D},\mathcal{F})}[\log p(z|\theta)]$$

- Therefore, using this approximation, the optimal set $\mathcal{C}$ to remove is the top K points z with the highest value of $\hat{r}(z)$.

Cause Identification

There are a couple of ways to evaluate the metric $\hat{r}(z)$ for a data point z.

- **Linear Influence Function:** A point estimate of $p(\theta|\mathcal{D}, \mathcal{F})$, can be found using a single update of gradient descent on log likelihood of $\mathcal{F}$. This gives

$$\hat{r}(z) = -\gamma \nabla_{\hat{\theta}} \log p(\mathcal{F}|\hat{\theta})^T \hat{\mathbb{F}}_{\hat{\theta}} \nabla_{\hat{\theta}} \log p(\mathcal{F}|\hat{\theta})$$

- **Elastic Weight Consolidation:** In this technique, we find a point estimate of θ (MAP solution), then we use SGD to maximise the objective function:

$$-\log p(\mathcal{F}|\theta) - \frac{N}{2}(\theta - \hat{\theta})^T \hat{\mathbb{F}}_{\hat{\theta}}(\theta - \hat{\theta}) - \frac{\lambda}{2}\left\|\theta - \hat{\theta}\right\|_2^2$$

Table of Contents

Model Treatment

- In this step, we use the current posterior $p(\theta|\mathcal{D})$ as the prior and continually remove the data points $\mathcal{C}$ to obtain the required posterior $p(\theta|\mathcal{D} \setminus \mathcal{C})$.
- If we use an approximate posterior $q(\theta)$, then the updated posterior will be

$$q_{-\mathcal{C}}(\theta) \propto \frac{q(\theta)}{p(\mathcal{C}|\theta)} \approx p(\theta|\mathcal{D} \setminus \mathcal{C})$$

Model Treatment

There are a number of ways to find the approximate posterior for the corrected data.

- One simple way would be to learn the new MLE/MAP estimate of $p(\theta|\mathcal{D} \setminus \mathcal{C})$ by using SGD to maximise $\log p(\mathcal{D} \setminus \mathcal{C}|\theta)$ starting from $\hat{\theta}$.
- We can use a certified removal technique developed in *Guo et al.* which applies a newton step on the model parameters to remove the influence of the deleted data point. We apply newton update removal mechanism by using a single step gradient descent:

$$\hat{\theta}_{-\mathcal{C}} = \hat{\theta} - \gamma \hat{\mathbb{F}}_{\hat{\theta}}^{-1} \nabla_{\hat{\theta}} \log p(\mathcal{C}|\hat{\theta})$$

- However, this technique requires the expensive computation of the Fisher Information matrix.

Model Treatment

- Another technique that could be used to update the parameters is to use EWC. We maximise the following objective function:

$$-\log p(\mathcal{C}|\theta) - \frac{N}{2}(\theta - \hat{\theta})^T \hat{\mathbb{F}}_{\hat{\theta}}(\theta - \hat{\theta}) - \frac{\lambda}{2}\left\|\theta - \hat{\theta}\right\|_2^2$$

- The first term tries to reduce the influence of $\mathcal{C}$ whereas the remaining two terms form a regulariser, which makes sure θ doesn't deviate too much from the original value.
- This method also requires the computation of the Fisher Information matrix, however this method is a arguably better than Newton update removal, since it is a more general method.

Table of Contents

Strengths

- The proposed method is general and can be applied to a wide range of architectures and applications.
- The identification of failure causes leads to an extra generation of useful data, which can later be assessed to check possible sources of detrimental training data.

Table of Contents

Limitations

- This method requires the original training data to be available at the time of update.
- To successfully implement this model, we would need to store the training data during the deployment, this is a potential privacy concern, and some regulations enforce deletion of data after a specific time bound.

Table of Contents

Our Proposal

- The detrimental data may contain some useful information as well.
- We aim to extract the useful features from detrimental data by selectively "aligning" the mislabeled data with the correct data.
- This can be done by defining an energy function for each data point. Minimising the energy for an input aligns it with correct data.

$$E_{\hat{\theta}_{+\mathcal{F}}}(x) = -\log \sum_{y'} \exp\left(f_{\hat{\theta}_{+\mathcal{F}}}(x)\left[y'\right]\right)$$

$$\tilde{x} = x - \eta \nabla_x E_{\hat{\theta}_{+\mathcal{F}}}(x),$$

$$\tilde{y} = p_{\hat{\theta}_{+\mathcal{F}}}(y \mid \tilde{x}) = \frac{\exp\left(f_{\hat{\theta}_{+\mathcal{F}}}(\tilde{x})\right)}{\sum_{y'} \exp\left(f_{\hat{\theta}_{+\mathcal{F}}}(\tilde{x})\left[y'\right]\right)},$$

Our Proposal (Contd)

- The updated data may then be incorporated in the learned Neural Network.
- Only last few layers of the Neural Network may be updated, ie, the model may be fine tuned with this new useful data.

For defective data identification, we identified an alternative way of calculating a score for each data point.

- **Data Shapley** value may be used in place of $r(z)$ for assessing the contribution of each input i the model. A negative Shapley value implies that the data point has a detrimental effect on the model.
- The strength of this method is that we get a quantitative estimate of how much defective an input is, which would help us in better demarcating mislabeled vs true data.

Table of Contents

Conclusion

- The seed paper explains how continual unlearning can be used to improve accuracy of neural networks.
- We explored other methods of finding the score function $r(z)$ for cause identification and better techniques for repairing the model after identifying the detrimental data from related papers.
- We attempted to implement the framework developed in this paper and test it on real data.

Table of Contents

References

- Repairing Neural Networks by Leaving the Right Past Behind
- Retaining Beneficial Information from Detrimental Data for Deep Neural Network Repair
- Data Shapley: Equitable Valuation of Data for Machine Learning