AUXILIARY MODELS AND DEPENDENT LEARNING

HYEONGJUN JIN

Abstract. We define what is an auxiliary model for a pre-trained model and present a schema for training such a model in an unsupervised, but dependent manner.

Contents

1. Auxilliary models

Assume $\tilde{\theta}$ is a pre-trained parameter for a parametrized model $\mathcal{M}$ with respect to an admissible loss function λ.

Definition 1.1. *Consider a pair $(\mathcal{N}, g)$ where $\mathcal{N} \in ((Y \models_\Phi Z))$ is a model parametrized by Φ and $g : Z \to \mathbb{R}$ is a function bounded below such that it is smooth on a open dense subset of Z. It is called an* ***auxiliary model for*** $(\mathcal{M}, S, \lambda, \tilde{\theta}, V_{\tilde{\theta}})$ *if the loss function*

$$\begin{aligned} \nu_g : ((X \models Z)) &\to \mathbb{R} \\ \mathcal{L} \qquad &\mapsto \nu_g(\mathcal{L}) = \sum_{(x,y)\in S} g(\mathcal{L}(x)) \end{aligned}$$

is admissible for the composite model $\mathcal{N}_{(-)} \circ \mathcal{M}_{(-)} : V_{\tilde{\theta}} \times \Phi \to ((X \models_{V_{\tilde{\theta}} \times \Phi} Z))$.

Note that the loss ν_g makes the training set S unsupervised.

2. Dependent Learning

The learning scheme of an auxiliary model is as follows:

- Do gradient descent to find a minimizer $\tilde{\phi}$ of the function below while keeping the argument $\tilde{\theta}$ fixed:

$$\begin{aligned} \mu = \nu_g \circ \mathcal{N}_{(-)} \circ \mathcal{M}_{(-)} : V_{\tilde{\theta}} \times \Phi &\to \mathbb{R} \\ (\theta, \phi) &\mapsto \sum_{(x,y)\in S} g(\mathcal{N}_\phi(\mathcal{M}_\theta(x))) \\ (\tilde{\theta}, \phi_{old}) &\mapsto (\tilde{\theta}, \phi_{new}). \end{aligned} \tag{1}$$

  Here we update the variable ϕ only by considering the projected gradient $\nabla\mu(-|\tilde{\theta})$:

$$\begin{aligned} d\mu_{(\theta,\phi)} : T_\theta V_{\tilde{\theta}} \times T_\phi \Phi \to \mathbb{R} \rightsquigarrow \nabla\mu : V_{\tilde{\theta}} \times \Phi \to TV_{\tilde{\theta}} \times T\Phi \\ \Rightarrow \nabla\mu(-|\tilde{\theta}) = \mathrm{proj}_{T\Phi} \nabla\mu(\tilde{\theta}, -) : \Phi \xrightarrow{(\tilde{\theta},-)} V_{\tilde{\theta}} \times \Phi \xrightarrow{\nabla\mu} TV_{\tilde{\theta}} \times T\Phi \to T\Phi \end{aligned} \tag{2}$$

As is $\nabla\mu$, this gradient is carrying information of the direction of steepest change for $\mu(\tilde{\theta},\phi)$ for a given ϕ according to a change $(\tilde{\theta},\phi)\mapsto(\tilde{\theta}+\Delta\theta,\phi+\Delta\phi)$.

(Of course we assume Θ and Φ to have Riemannian metrics so that $d\mu$ can be transformed into the gradient vector field $\nabla\mu$ on $V_{\tilde{\theta}}\times\Phi$.)

Note the dependence of the objective function μ in the direction of θ, as the term $\mathcal{M}_{\tilde{\theta}}$ can vary according to an infinitesimal change on $\tilde{\theta}$, making the update of ϕ to depend on the pre-trained model's current state, whence the name **dependent learning**.

- After the dependent learning of $\mathcal{N}$ with respect to ν_g, we make the composite model learn simultaneously with respect to some loss function $h(\lambda,\nu_g)$, making the model parameters of $\mathcal{M}$ and $\mathcal{N}$ interdependent. We denote simply by $\mathcal{M}\bowtie\mathcal{N}$ a model incorporated in this way.

  (Simplest such h is $h(x,y)=x+y, h(x,y)=xy, h(x,y)=x^y$, etc.)

GitHub of example models(updating): Application to Several Models

(In the GitHub, training of the evaluator is solely done with the reproduced pictures of the VAE. It's remarkable to note that the evaluator seems to value the original data as better photo even in this bias.)

3. Applications

1. Auto-evaluator for any generative model

   1-1. Data selection 1-2. Data augmentation 1-3. Performance improvement

Our auto-evaluator in its simplest form is as follows: We set $Z=\mathbb{R}^k$ for some k and use the exponentially decreasing function on positive end as our loss function:

$$g:\mathbb{R}^k\to\mathbb{R},\quad g(x_1,\dots,x_k)=\sum_{1\le i\le k}\exp(-\mathit{ReLU}(x_i)).$$

The traditional gradient descent with respect to this function makes any parametrized model $\mathcal{N}_{(-)}\in((Y\models_\Phi Z))$ to diverge at the end. As soon as we employ the dependent learning, however, the parameter ϕ stays in a compact region in Φ, meaning its convergence which is the practical part for use.

As our auto-evaluator converges to a trained parameter $\tilde{\phi}$, we let $\mathcal{N}_{\tilde{\phi}}(\mathcal{M}_{\tilde{\theta}}(x))\in Z=\mathbb{R}^k$ the value vector or the values at $x\in X$ for the model $\mathcal{M}_{\tilde{\theta}}$. The sum of its entries or the sum of the log of its entries can be used as the value of $\mathcal{M}$.

2. Hyperparameter adjustor

3. Weight scheduler

4. Everything you want

4. Appendix. Mathematical Definitions

We begin by defining a model in mathematical terms.

Definition 4.1. *Let X and Y be smooth manifolds and $C(X,Y)$ be the set of all continuous maps from X to Y. Then $C(X,Y)$ is a topological space with the compact-open topology.*

*(1) By a **model $\mathcal{M}$ for Y based on X**, we refer an element $\mathcal{M}\in C(X,Y)$ that is smooth on an open dense subset of X. Let $((X\models Y))$ denote the subspace of $C(X,Y)$ consisting of all models for Y based on X.*

(2) For models $\mathcal{M} \in ((X \models Y))$ and $\mathcal{N} \in ((Y \models Z))$ where Z is another smooth manifold, the composite $\mathcal{N} \circ \mathcal{M}$ is again a model. We call it the ***composite model****.*

(3) A ***loss function*** *on $((X \models Y))$ is a function $\lambda : ((X \models Y)) \to \mathbb{R}$ that is continuous and bounded below.*

One typical way to set a loss function is the following.

Definition 4.2. *(1) A* ***(supervised) dataset*** *for $((X \models Y))$ is a finite subset $S \subset X \times Y$.*

(2) Given a dataset S and a function $f : Y \times Y \to \mathbb{R}$ which is bounded below and smooth on an open dense subset of $Y \times Y$, a ***supervised f-loss*** *(or simply an* ***f-loss****) with respect to S is a loss function $\lambda_f : ((X \models Y)) \to \mathbb{R}$ defined as follows:*

$$\lambda_f(\mathcal{M}) = \sum_{(x,y)\in S} f(\mathcal{M}(x), y).$$

In practice, we set λ as a sum discrepancy measures each representing a training objective. Since the compact-open topology on $((X \models Y))$ is too coarse to have a differentiable structure that is necessary for training a machine, we shall consider the following concept.

Definition 4.3. *Let X, Y, and Θ be smooth manifolds and $\lambda : ((X \models Y)) \to \mathbb{R}$ be a loss function.*

(1) A collection of models $\{\mathcal{M}_\theta\}_{\theta\in\Theta}$ for Y based on X is said to be ***parametrized by*** *Θ if (a) $\mathcal{M}_{(-)} : \Theta \to ((X \models Y))$ is continuous and (b) $(\theta, x) \mapsto \mathcal{M}_\theta(x) \in Y$ is smooth on an open dense subset of $\Theta \times X$. Let $((X \models_\Theta Y))$ denote the space of models for Y based on X parametrized by Θ.*

(2) We say λ is ***admissible*** *for the parametrized model $\mathcal{M}_{(-)} \in ((X \models_\Theta Y))$ if the composite continuous function*

$$\begin{aligned} \lambda \circ \mathcal{M}_{(-)} : \Theta &\to \mathbb{R} \\ \theta &\mapsto \lambda(\mathcal{M}_\theta) \end{aligned}$$

is smooth on an open dense subset of Θ.

(3) A parametrized model $\mathcal{M}_{(-)} \in ((X \models_\Theta Y))$ is said to be ***pre-trained*** *with respect to the loss function λ if λ is admissible for $\mathcal{M}_{(-)}$ and there is an observed $\tilde{\theta} \in \Theta$ equipped with a neighborhood $V_{\tilde{\theta}}$ on which $\lambda \circ \mathcal{M}_{(-)} : V_{\tilde{\theta}} \to \mathbb{R}$ is smooth and attains a local minimum at $\tilde{\theta}$.*

Note that, for two parametrized models $\mathcal{M}_{(-)} \in ((X \models_\Theta Y))$ and $\mathcal{N}_{(-)} \in ((Y \models_\Phi Z))$, the composite $\mathcal{N} \circ \mathcal{M} \in ((X \models_{\Theta\times\Phi} Z))$ is also a parametrized model.

Dept. of Mathematics, Yonsei University
Email address: jhj941211@gmail.com