Приближение байесовского вывода с помощью DNN

Атанов Андрей, Научный руководитель: Дмитрий Ветров, Куратор: Соколов Евгений
Высшая Школа Экономики, Факультет компьютерных наук

p(B|A)yesgroup.ru

Байесовский вывод

$\mathcal{D}_N = \{(x_i, y_i)\}_{i=1}^N$, $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$ – наблюдаемая выборка, $p(y|x, \theta)$ – параметрическая модель. Большинство методов строят оценку максимального правдоподобия:

$$\theta_{ML} = \arg\max_\theta \prod_{i=1}^N p(y_i|x_i, \theta)$$

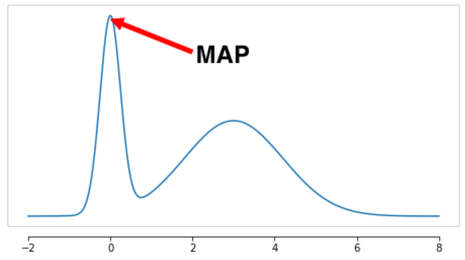
- Легко оптимизируется — в большинстве случаев функционал вогнутый
- Приводит к переобучению при маленьких выборках

Вместо этого будем восстанавливать апостериорную плотность:

$$p(\theta|\mathcal{D}_N) = \frac{p(\theta)\prod_{i=1}^N p(y_i|x_i, \theta)}{p(\mathcal{D}_N)}$$

$$\theta_{MAP} = \arg\max_\theta p(\theta|\mathcal{D}_N)$$

- По-прежнему легко оптимизируется
- Меньше переобучается — $p(\theta)$ играет роль регуляризатора
- Может плохо описывать апостериорное распределение



Воспользуемся всей информацией об апостериорном распределении, проведя взвешенное голосование, чтобы построить распределение на y^* для нового объекта x^*:

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

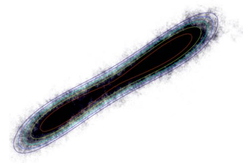
Для нейросетей и других сложных моделей провести полный вывод оказывается невозможным. Обычно его заменяют приближенным выводом:

$$p(y|x, \mathcal{D}_N) \approx \frac{1}{|\Theta|}\sum_{\theta^s \in \Theta} p(y|x, \theta^s)$$

где $\theta^s \sim p(\theta|\mathcal{D}_N)$. Учитываем не все возможные модели, а только конечное число. Данная оценка является достаточно точной при $|\Theta| \gg 1$.

SGLD

Метод Stochstic Gradient Langevin Dynamics является легко масштабируемым методом, строящим случайный процесс, который сходится к апостериорному распределению [1].



$$\theta_{t+1} = \theta_t + \frac{\eta_t}{2}\underbrace{\left(\nabla p(\theta) + \frac{N}{M}\sum_{(x,y)\in\mathcal{D}_M}\nabla p(y|x,\theta)\right)}_{\text{SGD}} + \underbrace{\mathcal{N}(0, \eta_t I)}_{\text{Langevin dynamics}}$$

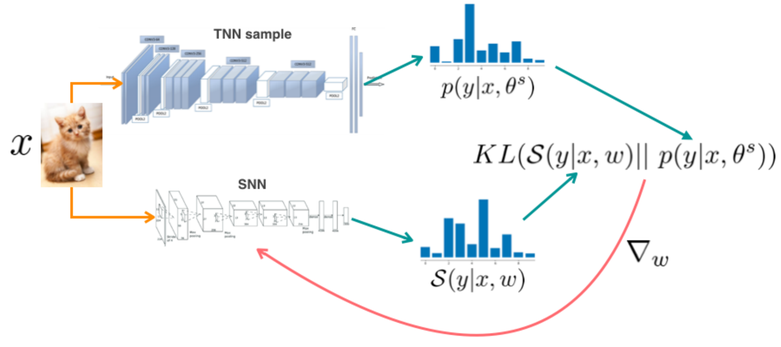
$\mathcal{D}_M$ — минибатч из $\mathcal{D}_N$.

Ensemble distillation

При работе с моделями с большим количеством параметров, сохранять все сэмплы оказывается невозможным, кроме того получение ответа занимает в $|\Theta|$ раз дольше, чем для одной модели. Идея заключается в приближении апостериорного распределения $p(y|x, \mathcal{D}_N)$, которое является ансамблем Teacher NN — $p(y|x, \theta^s)$, одной Student NN — $\mathcal{S}(y|x, w)$.

$$\mathcal{L}(w) = \frac{1}{|\Theta|}\frac{1}{|\mathcal{D}'|}\sum_{x\in\mathcal{D}'}\sum_{\theta^s\in\Theta} KL(\mathcal{S}(y|x,w)||\ p(y|x,\theta^s))$$

Данный функционал можно легко минимизировать с помощью SGD, беря на каждом шаге объект $x \in \mathcal{D}'$ (данные могу быть не размечены) и модель θ^s с помощью SGLD.



Использование температуры (задача классификации):

$$P(y = k|x) = \frac{\exp(logit_k/T)}{\sum_i \exp(logit_i/T)}$$

Строит более вариативное распределение и содержит больше информации об объекте, также может выступать в роли регуляризатора для студента [3].

Результаты

Boston Housing

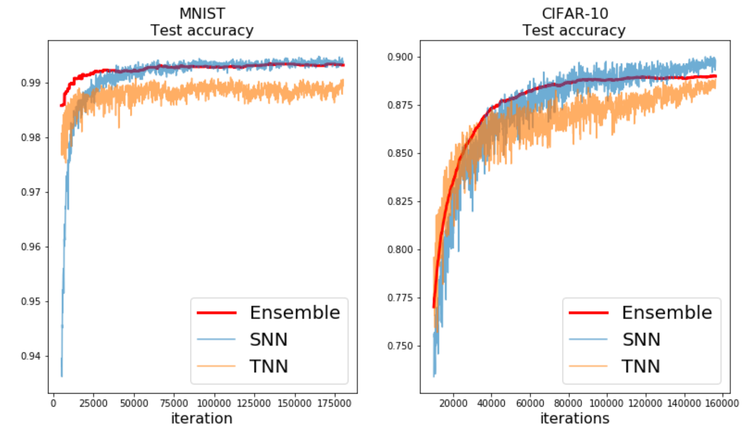
В качестве модели использовалась однослойная нейросеть с 50 скрытыми нейронами для TNN и SNN.

TNN	SNN	Ensemble
3.642	3.201	3.057

Таблица 1: mean test RMSE

MNIST и CIFAR-10

Для задачи MNIST использовалась архитектура Le-Net (TNN и SNN). TNN обучалась с помощью RMSProp в стиле SGLD. SNN обучалась с помощью Adam optimizer.
Для задачи CIFAR-10 использовалась архитектура VGG (TNN и SNN). TNN обучалась с помощью RMSProp, сэмплирование производилось с помощью техники Dropout. SNN обучалась с помощью Adam optimizer.



Источники

[1] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. pages 681–688, 2011.

[2] Anoop Korattikara Balan, Vivek Rathod, Kevin Murphy, and Max Welling. Bayesian dark knowledge. CoRR, abs/1506.04416, 2015.

[3] Jeff Dean Geoffrey Hinton, Oriol Vinyals. Distilling the knowledge in a neural network. 2014.