Мета-оптимизация для многозадачности

Бишук Антон

Введение

Предлагается алгоритм метаобучения, который совместим с любой моделью, обученной с помощью градиентного спуска, и применим к множеству различных задач обучения, включая классификацию, регрессию и обучение с подкреплением.

Цель метаобучения — обучить модель множеству учебных задач, чтобы она могла решать новые учебные задачи, используя лишь небольшое количество обучающих выборок.

По сути, предложенный метод обучает модель, чтобы ее можно было легко настроить.

Алгоритм **не увеличивает** количество изученных **параметров** и не накладывает **ограничений** на **архитектуру** модели (например, требуя рекуррентной модели или сиамской сети), и его можно легко комбинировать с полносвязными, сверточными или рекуррентными нейронными сетями. Его также можно использовать с различными функциями потерь.

Идея

Процесс обучения параметров модели таким образом, что несколько шагов градиента или даже один шаг градиента могут дать хорошие результаты в новой задаче, можно рассматривать с точки зрения изучения признаков как **построение внутреннего представления**, которое в целом подходит для многих задач.

Если внутреннее представление подходит для многих задач, простая точная настройка параметров (например, в первую очередь путем изменения весов верхнего слоя в модели с прямой связью) может дать хорошие результаты.

По сути, процедура оптимизирует модели, которые легко и быстро настроить, позволяя адаптироваться в нужном месте для быстрого обучения.

С точки зрения динамических систем процесс обучения можно рассматривать как максимизацию чувствительности функций потерь новых задач по отношению к параметрам: когда чувствительность высока, небольшие локальные изменения параметров могут привести к значительному улучшению потерь задачи.

Интуиция, лежащая в основе этого подхода, заключается в том, что одни внутренние представления легче переносятся, чем другие.

Постановка

Рассмотрим модель **f**, которая отображает наблюдения **x** в выходные данные **a**.

Во время метаобучения модель обучается, чтобы она могла адаптироваться к большому или бесконечному количеству задач.

Поскольку необходимо применить структуру к множеству задач обучения – введем формально определение задачи:

$$T = \{L(x_1, a_1, \ldots, x_H, a_H)\ , q(x_1), q(x_{t+1}|x_t, a_t), H\}$$

состоит из функции потерь **L**, распределения по начальные наблюдения $\mathbf{q(x_1)}$, распределение переходов $\mathbf{q(x_{t+1}|x_t, a_t)}$ и длина эпизода **H**.

В i.i.d. задачи обучения с учителем, длина **H** = 1.

Модель может генерировать образцы длины **H**, выбирая выход в каждый момент времени **t**.

Рассматривается распределение по задачам **p(T)**, к которым мы хотим, чтобы наша модель могла адаптироваться. В настройке обучения **K**-выстрелов модель обучается для изучения новой задачи $\mathbf{T_i}$, полученной из **p(T)**, только из **K** выборок, полученных из $\mathbf{q_i}$ и обратной связи $\mathbf{L_{Ti}}$, сгенерированной $\mathbf{T_i}$, а затем модель тестируется на новых выборках из $\mathbf{T_i}$.

Затем модель **f** улучшается путем рассмотрения того, как ошибка теста на новых данных из $\mathbf{q_i}$ изменяется по отношению к параметрам. По сути, ошибка теста на выборочных задачах $\mathbf{T_i}$ служит ошибкой обучения процесса метаобучения.

Алгоритм

Поскольку модель будет точно настроена для новой задачи, мы будем стремиться изучить модель таким образом, чтобы правило обучения на основе градиента могло быстро сделать прогресс в новых задачах, взятых из **p(T)**, без переобучения. По сути, мы будем стремиться найти параметры модели, чувствительные к изменениям в задаче, так что небольшие изменения параметров будут приводить к значительным улучшениям функции потерь любой задачи, взятой из **p(T)**, при изменении в сторону увеличения градиент этой потери.

Мы не делаем никаких предположений о форме модели, кроме предположения, что она параметризована некоторым вектором параметров $\boldsymbol{\theta}$ и что функция потерь является достаточно гладкой по $\boldsymbol{\theta}$. Формально мы рассматриваем модель, представленную параметризованной функцией $\mathbf{f}_{\boldsymbol{\theta}}$ с параметрами $\boldsymbol{\theta}$. При адаптации к новой задаче $\mathbf{T}_i$ параметры модели $\boldsymbol{\theta}$ становятся $\boldsymbol{\theta}_i'$.

В нашем методе обновленный вектор параметров $\boldsymbol{\theta}_i'$ вычисляется используя обновления градиентного спуска для задачи $\mathbf{T}_i$. Например, при использовании одного обновления градиента:

$$\theta_i' = \theta - \alpha \nabla_\theta L(T_i(f_\theta)).$$

Параметры модели обучаются путем оптимизации производительности $\mathbf{f}_{\boldsymbol{\theta}_i'}$ по отношению к $\boldsymbol{\theta}$ для задач, выбранных из **p(T)**. Более конкретно, метацель выглядит следующим образом:

$$\min_\theta \Sigma_{\ Ti\sim p(T)}\ L_{Ti}(f_{\theta i'}) = \Sigma_{\ Ti\sim p(T)}\ L_{Ti}\,(f_{\theta-\alpha\nabla_\theta L(Ti\ (f_\theta))})$$

Мета-оптимизация выполняется по параметрам модели $\boldsymbol{\theta}$, тогда как цель вычисляется с использованием обновленных параметров модели $\boldsymbol{\theta}'$. Метод направлен на оптимизацию параметров модели таким образом, чтобы один или небольшое количество шагов градиента в новой задаче обеспечивали максимально эффективное поведение в этой задаче. Мета-оптимизация задач выполняется с помощью стохастического градиентного спуска (SGD):

$$\theta \leftarrow \theta - \beta \nabla_\theta\, \Sigma_{Ti\sim p(T)}\ L(T_i\ (f_{\theta i'})),$$ где β — размер меташага.