МОДЕЛЬ МЕТА-ОБУЧЕНИЯ С РАСШИРЕНИЕМ ПРИЗНАКОВ

META-LEARNING MODEL WITH AUGMENTATION FEATURES

УДК 004.8

Ерискин Виталий Владимирович, магистрант, Московский государственный технический университет им. Н. Э. Баумана, г. Москва

Eriskin Vitaly Vladimirovich, Graduate Student, The Moscow State Technical University, Moscow. e-mail: eriskinv@gmail.com

Аннотация

В статье рассматривается модель мета-обучения на малой выборке. Глубокое обучение в области компьютерного зрения в последние годы демонстрирует большие достижения. Компьютерное зрение уже близко к уровню человеческого. Но чтобы продемонстрировать всю мощь технологии необходимы большие массивы данных для обучения. Если провести обучение на небольшом наборе данных, то результат будет не столь впечатляющий, и модель с высокой степенью вероятности продемонстрирует переобучение. Такие проблемы решает мета-обучение[5]. Большинство исследований в области распознавания изображений сосредоточены на алгоритме классификации, игнорируя важность признаков. Представляется, что каждый класс имеет свои уникальные особенности, которые делают его более узнаваемым. В статье предлагается расширить признаки класса, и использовать это для решения задачи классификации на малой выборке. В предложенной модели используется сеть Prototype Network, для тестирования модели используется набор данных Mini-ImageNet.

Annotation

The paper discusses a small sample meta-learning model. Deep learning in computer vision has shown great advances in recent years. Computer vision is already close to the human level. But to demonstrate the full power of the technology, large datasets are needed for training. If you train on a small dataset, the result will not be so impressive, and the model will demonstrate overfitting with a high degree of probability. Meta-learning solves such problems. Most of the research in image recognition focuses on the classification algorithm, ignoring the importance of features. Each class appears to have its own unique characteristics that make it more recognizable. The paper proposes to expand the features of the class and use it to solve the problem of classification on a small sample. The proposed model uses the Prototype Network, and the Mini-ImageNet dataset is used to test the model.

Ключевые слова: машинное обучение, мета-обучение, глубокие нейронные сети, компьютерное зрение

Keywords: machine learning, meta-learning, deep neural networks, computer vision

1 Введение

Глубокое обучение успешно применяется в различных задачах распознавания изображений[1], но его результаты не идеальны, когда данных мало или когда их необходимо адаптировать к новым задачам в короткие сроки[3]. В то же время, взрослому человеку достаточно увидеть предмет один или несколько раз, чтобы понять его природу и четко выделять его среди множества других предметов.

Самое простое решение этой задачи - использовать обучение с миграцией[6] - тонкую настройку заранее обученной модели. Однако из-за недостатка данных это может привести к переобучению и как следствие к ошибкам в работе модели. Мета-обучение [2] может лучше справляться с задачами подобного рода. Мета-обучение стало основным методом решения проблемы обучения на малых выборках. Для решения задач обучения на небольших выборках в мета-обучении применяют один из трех основных подходов.

Метрический подход к мета-обучению [2]. Цель обучения состоит в подборе функции расстояния, которая подбирается в процессе обучения в соответствии с расстоянием между векторами признаков изображений. Сеть Prototype Network[7] основана на представлении, что существует точка-прототип для каждого класса в пространстве признаков, которая может наилучшим образом представлять все данные этого класса, для предсказания класса объекта вычисляется расстояние между точками-прототипами и неразмеченным изображением. Сеть Matching Network[10] вычисляет сходство между неразмеченными и размеченными данными. Сеть Relationship Network[7] вычисляет расстояние, сравнивает его с евклидовым расстоянием, либо косинусным расстоянием, и таким образом можно получить более точное представление о расстоянии между вектором признаком и отнести изображение к тому или иному классу.

Мета-обучение на основе инициализации[16]. Этот метод реализуется через обучение ранее обученной сети, когда перед ней ставится новая задача, происходит операция обновления градиентного спуска существующей модели для адаптации к сценарию мета-обучения.

В MAML, например, обучающий процесс обычно состоит из двух частей: внутреннего и внешнего цикла. Внутренний цикл имитирует различные сценарии задач с небольшими выборками и получает средние потери по всем задачам. Внешний контур оптимизирует средние потери, чтобы улучшить возможность обобщения всей модели.

Методы мета-обучения, основанные на улучшении данных[13-15], добавляют дополнительные данные для решения фундаментальной проблемы дефицита данных в задачах с небольшими наборами данных. MetaGAN[13] генерирует некоторые «поддельные» изображения на основе текущей задачи, и модель может найти более четкие границы различий в процессе идентификации реального изображения и сгенерированного изображения. Сеть FeatureHallucination[9] имитирует “аналоговую” способность человека передавать информацию об окружающей среде для генерации изображений целевого класса

в соответствующей сцене. Ван[11] считает, что изображение содержит не только информацию о метке, но и информацию о сцене, такую как освещение, позы на фотографии и окружающая среда, эти факторы сцены могут быть переданы целевому классу путем генерации модели.

Большинство вышеперечисленных методов имеют недостаток: базовый набор данных используется только для определения параметров модели, игнорируя связь между данными и новой задачей. Даже после тонкой настройки модели нет никакой гарантии, что она будет хорошо работать на целевых данных.

Для решения обозначенного ограничения предлагается случайным образом выбрать ряд дополнительных изображений из начального набора данных в качестве основного набора в процессе обучения вместе с целевыми данными обучения. Такой подход может легко масштабироваться до более сложных задач, где пространство меток для тестовых данных охватывает базовый класс и целевой класс. Например, изображения с крыльями, скорее всего, будут птицами, изображения с черными и оранжевыми полосами, скорее всего, будут тиграми. Мы определяем крылья и полосы как классовые характеристики птиц и тигров. Очевидно, что изображения с четкими классовыми признаками, легко классифицировать, а изображения без четкого классификационного признака, классифицировать сложнее. Как показано на рисунке 1, изображение слева, скорее всего, будет классифицировано правильно, а вот с изображениями справа могут возникнуть трудности при классификации.



Рисунок 1. Изображения с разными признаками класса

Используемые методы улучшения данных при мета-обучении различны, современное направление исследований в данной области фокусируется на получении максимально распределенного набора данных целевого класса, игнорируя при этом признаки, содержащиеся в самом изображении. Если эффект генерации изображения оставляет желать лучшего, то это снижает общую точность модели. С другой стороны, очень сложная структура генерации дополнительных изображений будет сложна с точки зрения обучения модели.

Представляется, что реальное изображение более важно, чем изображение, полученное генератором. Сделать реальное изображение более легким для идентификации или сделать признаки класса изображений более очевидными — это более эффективный способ решения проблемы обучения на малых выборках.

В данной работе предлагается CFA (Class Feature Augmentation) в качестве общей, гибкой структуры мета-обучения. Объединим CFA с наиболее часто используемым методом мета-обучения Prototype Network[1], и покажем, что CFA может улучшить работу Prototype Network.

2. Предварительная подготовка

2.1 Обучение на малых выборках и мета-обучение

Для обучающей задачи T с небольшой выборкой, алгоритм обучения требует нахождения совместного распределения набора признаков X и набора меток Y, $p^{T}_{X\times Y}(x,y)$. Обычно задача $T = (S_T, Q_T)$ состоит из вспомогательного набора S_T и набора запросов Q_T. Вспомогательный набор определяет параметры модели, а набор запросов определяет классификационный эффект модели на эту задачу. Вспомогательный набор содержит N различных типов, и каждый класс содержит K изображений (K обычно составляет 1, 5 или 20), поэтому это также называют N-образной задачей классификации малых выборок из K снимков.

Алгоритмы мета-обучения состоят из двух этапов. На первом этапе определяется параметр w алгоритма классификации посредством обучения. На этапе мета-обучения размеченный набор данных S_{meta} содержит большое количество изображений. В ходе каждого цикла обучения случайным образом

выбирается несколько изображений из S_{meta} в качестве задачи для классификации. Далее случайным образом выбирают подмножество m категорий из набора меток C, затем случайным образом выбирают несколько различных изображений из этой категории m для формирования вспомогательного набора и набора запросов, используемых для имитации задач классификации с небольшой выборкой. Во время обучения вспомогательный набор используется как априорного знания, а через мета-обучение получаем функцию потерь и оптимизируем модель. Второй этап — это мета-тестирование, обученную на предыдущем этапе модель проверяют на выполнении задачи классификации на целевой выборке.

2.2 Сеть Prototype Network

Модель сети Prototype Network основана на предположении, что в наборе данных для каждого класса существует точка-прототип, которая отражает общие характеристики всего класса[1]. Аналогично идеи алгоритма KNN[4], чем ближе объект к точке-прототипу, тем больше вероятность того, что объект и точка-прототип принадлежат к одному классу. Основная идея алгоритма в том, чтобы сопоставить D-мерный набор данных с M-мерным пространством признаков с помощью глубокой нейронной сети, а затем вычислить среднее значение p_k для каждого типа выборки в качестве точки-прототипа в новом пространстве.

3 Описание модели

Популярные методы мета-обучения используют только вспомогательный набор и набор запросов в каждой итерации, в предлагаемой в данной статье модели есть три набора: вспомогательный набор (Support Classes) S, набор запросов (Query Classes) Q и базовый набор (Base Classes) β. Если пространство меток вспомогательного набора и набора запросов одно и то же, определяемое как Y_s, то тип пространства меток называется целевым классом.

$$p_k = \frac{1}{|S_k|} \sum_{(x_i, y_i) \in S_k} f_\phi(x_i) \quad (1)$$

Модель после получения всех точек прототипа на тестовой фазе выбирает из набора запросов для классификации некоторый вектор x, вычисляет расстояние между x и всеми точками-прототипами в пространстве признаков и с помощью функции softmax классифицируется вектор x – это происходит путем вычисления расстояния между x и всеми точками прототипа в пространстве признаков. При этом расстояние нормализуется для получения вероятности x, соответствующей типу k.

$$p_\phi\ (y\ =\ k \mid x) = \frac{exp\left(-d\left(f_\phi(x), p_k\right)\right)}{\sum_{k'} exp\left(-d\left(f_\phi(x), p_{k'}\right)\right)} \qquad (2)$$

Метка в наборе данных и наборе запросов определяется как Y_β, то есть $Y_s \cap Y_\beta = \emptyset$,а типы в пространстве меток называются базовыми классами. Вспомогательный набор используется для имитации задачи распознавания на небольшой выборке, данные набора запросов используются для получения функции потерь и оценки эффективности классификации модели, а базовый набор служит для получения характеристик класса для каждого вспомогательного набора. Чтобы обеспечить объективность экспериментальных результатов, на этапе мета-обучения и мета-тестирования базовые наборы выбирались из набора данных мета-обучения и не использовались в мета-тесте. Как показано на рисунке 2, предложенная модель состоит из трех частей: модуль извлечения признаков, модуль улучшения признаков класса (CFA) и модуль мета-обучения.

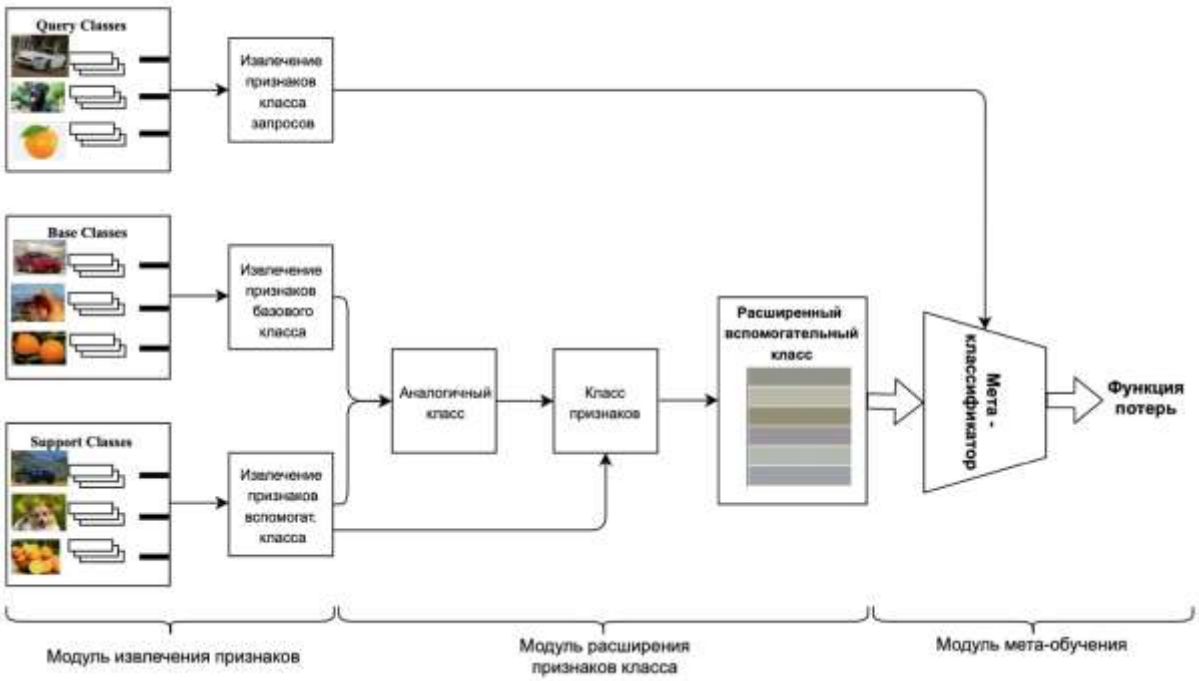


Рисунок 2. Структура модели.

Модуль извлечения признаков сопоставляет изображения во вспомогательном наборе, наборе запросов, и базовом наборе с пространством признаков. Далее вычисляется базовый класс и все точки-прототипы, что поддерживают целевой класс в пространстве признаков, и выбираются «аналогичный класс» соответствующий целевому классу из базового набора по расстоянию между точками-прототипами. Затем изображения в аналогичных классах берут в качестве входных данных, получаем характеристики для каждого класса, и перемещаем изображения вспомогательного набора в пространстве признаков, чтобы сделать характеристики класса более очевидными. И наконец, использование набора запросов Q и вспомогательного набора S^{aug} после CFA модулем мета-обучения, позволяет классифицировать изображение из набора запросов, получить функцию потерь и оптимизировать модель в общем.

Чтобы оценить эффективность модели, предложенной в данном исследовании, в качестве модуля мета-обучения используется сеть Prototype Network (PN). В разделе 3.1 описывается алгоритм генерации признаков класса, в разделе 3.2 описывается алгоритм улучшения признаков класса, подробное описание двухэтапного процесса реализации модуля CFA, в разделе 3.3 описываются детали реализации CFA-PN.

3.1 Алгоритм генерации признаков «аналогичного класса»

Одна из причин, по которой люди могут учиться на небольших выборках, заключается в том, что для новых предметов, которые мы не видели ранее, мы можем найти некоторые другие предметы, которые близки по признакам к новому предмету, а также благодаря способности объединять предметы на основании их признаков в классы «аналогичных» предметов (как показано на рисунке 3.)

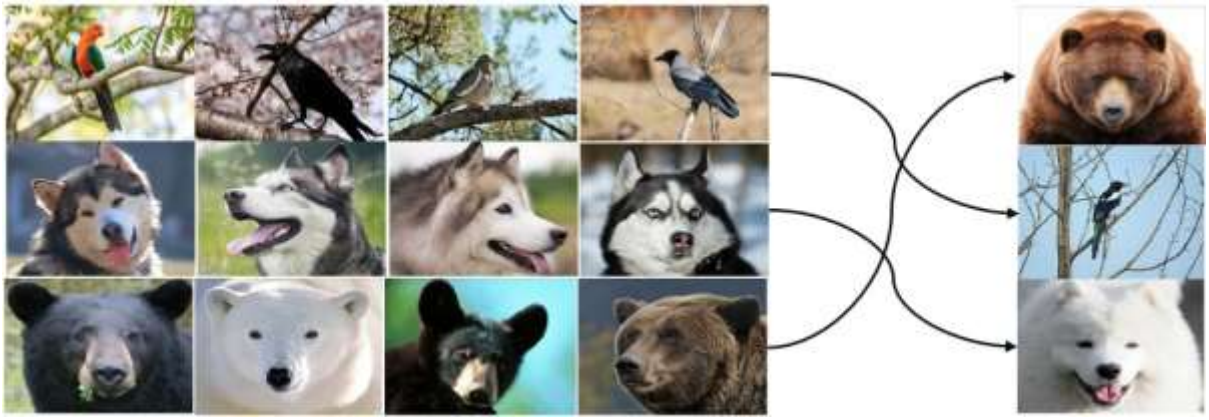


Рисунок 3. Аналогичные классы

Несмотря на то, что информации о новом изображении мало, характеристики нового изображения могут быть четко идентифицированы с помощью изображений аналогичного класса, содержащих большой объем информации об аналогичных изображениях. Например, даже если вы никогда не видели медведя и тигра, но у вас будет достаточно знаний о кошках, то вы сможете правильно классифицировать их (медведя и тигра), после просмотра нескольких изображений, потому что тигры и кошки внешне более похожи, чем медведи и кошки. Опираясь на этот феномен, мы используем «аналогичные классы» для генерации признаков целевого класса.

Исследования[10] показывают, что сопоставление многомерных данных с низкоразмерным пространством признаков с помощью метрического обучения может упростить распознавание изображений различного типа. Чем ближе объекты в пространстве признаков друг к другу, тем выше вероятность того, что объекты принадлежат к одному и тому же классу. Согласно этому принципу, мы определяем изображения, которые близки в пространстве признаков, но не

являются одинаковыми, как аналогичные классы. С другой стороны, цель метода улучшения признаков класса состоит в том, чтобы использовать аналогичные классы для усиления признаков целевого класса, чтобы сделать признаки класса более очевидными, что требует наличия сходных признаков между целевым классом и аналогичным классом. Таким образом, аналогичные классы получаются из пространства признаков.

Сначала вычисляется среднее значение всех изображений целевого и базового класса в пространстве признаков как класс признаков для каждого класса, затем вычисляется расстояние между классами признаков, после сортировки по расстоянию, выбирается ближайший в качестве соответствующего аналогичный класс.

$$p_{y_b} = \frac{1}{|\beta_{y_b}|} \sum_{(x_i, y_i) \in \beta_{y_b}} f_\phi(x_i) \quad (3)$$

$$p_{y_n} = \frac{1}{|S_{y_n}|} \sum_{(x_i, y_i) \in S_{y_n}} f_\phi(x_i) \quad (4)$$

$$similar(y_n) = \underset{y_b \in Y_\beta}{argmin}\, Distance\left(p_{y_b}, p_{y_n}\right) \quad (5)$$

где Y_β пространство меток базового набора β,

y_b и y_n метки в базовом наборе β и вспомогательном наборе S,

f_ϕ – функция извлечения признаков (реализованная с помощью нейронной сети), которая отображает все изображения в низкоразмерное пространство объектов,

$Distance$ — это функция вычисления расстояния, обычно косинусное или евклидово расстояние.

В задачах с небольшой выборкой исходный целевой класс содержит недостаточное количество изображений, чтобы получить характеристики класса. Кроме того, для задач типа 1-shot целевой класс содержит только одно изображение, и если использовать это изображение для создания признаков класса, тогда невозможно изменить исходное изображение с помощью признаков класса. Поэтому в данной работе для получения признаков класса используются аналогичные классы, а не сам целевой класс.

В соответствии с определением и методом расчета сходных классов, мы рассматриваем классовые признаки аналогичных классов как класс признаков целевого класса. Размерность вектора признаков уменьшена до более низкоразмерного пространства признаков, чтобы сделать представление признаков более четким. Процесс извлечения признаков класса для каждого изображения должен учитывать все изображения в аналогичных классах, а не напрямую извлекать признаки из одного изображения.

Более интуитивно понятный метод состоит в том, чтобы «соединить» все векторы изображений, а затем обработать их соответствующим образом. Однако, в отличие от варианта с обработкой некоторого небольшого количества изображений, при одновременной обработке большого количество изображений путем объединения векторов их признаков, размер результирующего вектора будет слишком большим, что может отрицательно сказаться на процессе обучения модели. Эта статья опирается на идею использования сети LSTM, все изображения аналогичных классов последовательно подаются на вход сети, чтобы получить признаки класса каждого изображения.

Общий выход сети LSTM зависит от порядка входной последовательности, и двунаправленная сеть LSTM может ослабить это влияние. Извлечение признаков класса, очевидно, не зависит от порядка ввода изображения, поэтому в этой статье используется двусторонняя сеть LSTM для получения признаков класса каждого изображения x_i в аналогичном классе. На результат выхода сети LSTM влияет порядок входной последовательности, двусторонняя сеть LSTM может уменьшить этот эффект. Извлечение признаков класса, очевидно, не имеет ничего общего с порядком ввода изображения, поэтому в этой статье используется двусторонняя сеть LSTM для получения характеристик класса каждого изображения в аналогичном классе.

$$\overrightarrow{h_k}, \overrightarrow{c_k} = LSTM(g_1(f_\phi(x_i)), h_{k-1}, c_{k-1}) \quad (6)$$

$$\overleftarrow{h_k}, \overleftarrow{c_k} = LSTM(g_1(f_\phi(x_i)), h_{k-1}, c_{k-1}) \quad (7)$$

$$X_i = \overrightarrow{h_k} + \overleftarrow{h_k} + g_1(f_\phi(x_i)) \quad (8)$$

где f_ϕ - функция извлечения признаков,

g_1 - функция уменьшения размерности (реализованная нейронной сетью),

k - длина последовательности LSTM.

$$feature(y_n) = g_2\left(\frac{1}{k}\sum_{i=1}^{k} X_i\right) \quad (9)$$

где g_2 - функция увеличения разрядности (реализована нейронной сетью),

$similar\ (y_n)$ - категория подобия.

Для удобства последующего выражения LSTM, g_1, g_2 объединены в функцию генерации признаков класса g_ψ, а именно:

$$feature(y_n) = g_\psi(similar(y_n)) \quad (10)$$

3.2 Алгоритм получения признаков класса

С помощью структуры извлечения признаков можно получить признаки класса вспомогательного набора в общем пространстве признаков. Подобно метрическому обучению, чем ближе расстояние в пространстве признаков, тем ближе отношение объекта изображения к объекту класса. Поэтому сокращая расстояние между ними, мы добиваемся улучшения характеристик. При этом, положение признака фиксируется, а положение, представленное изображением, перемещается.

Возможно выполнить один и тот же масштаб движения для каждого изображения из вспомогательного набора, но начальное расстояние между различными изображениями и признаками класса может быть очень разным, поэтому для масштабирования расстояния используют переменный коэффициент. Для изображений, значительно отдаленных от признаков особенности изображения не очевидны, и требуют перемещения на большее расстояние. Если расстояния между признаками небольшие, то признаки изображения относительно очевидны, и его следует переместить на небольшое расстояние, чтобы предотвратить отклонение от его собственного распределения признаков. Сохраняются некоторые различия между признаками аналогичного и

целевого класса. Поэтому предлагается использовать модуль масштабирования для вывода подходящего коэффициента для каждого вспомогательного набора изображений s_k.

Мы объединили вспомогательный набор изображений и соответствующие признаки класса в качестве входных данных для модуля масштабирования. Затем использовали выходные данные в качестве веса для перемещения изображения вспомогательного набора.

$$s_k = h([x_i, feature(y_n)]) \quad (11)$$

$$x_i^{aug} = s_k \cdot feature(y_n) + (1 - s_k) \cdot x_i \quad (12)$$

где h - функция масштабирования (реализована с помощью нейронной сети), s_k - значение от 0 до 1.

Реализация CFA показана в алгоритме 1.

Алгоритм 1. Алгоритм получения признаков класса

Input: количество объектов вспомогательного набора – N_s, количество изображений в каждой категории – K_s; количество объектов общего набора - N_b , количество изображений в каждой категории - K_b; количество изображений каждой категории в наборе запросов - K_q; функция извлечения признаков - f_ϕ, функция вычисления расстояния - Distance, функция генерации признаков класса - g_ψ, функция масштабирования - h.

Output: Вспомогательный набор данных

Begin

1. Случайным образом выберите N_s уникальных объектов из вспомогательного набора данных и выберите N_b уникальных объектов из целевого набора данных.

2. Случайным образом выберите K_s и K_q различных изображений из категории N_s, чтобы сформировать вспомогательный набор S и набор запросов Q. Случайным образом выберите K_b различных изображений из категории N_b, чтобы сформировать базовый набор β.

3.**for** y_n in 1~N_s

4. $p_{y_n} = \frac{1}{S_{y_n}} \sum_{(x_i, y_i) \in S_{y_n}} f_\phi(x_i)$

5. **for** y_b in 1~N_b
6. $p_{y_n} = \frac{1}{S_{y_n}} \sum_{(x_i, y_i) \in S_{y_n}} f_\phi(x_i)$
7. $similar(y_n) = argmin\ Distance(p_{y_b}, p_{y_n})$
8. **end for**
9. $feature(y_n) = g_\psi(similar(y_n))$
10. **for** $x_i \in S_{y_n}$ **do**
11. $s_k = h([x_i, feature(y_n)])$
12. $x_i^{aug} = s_k \cdot feature(y_n) + (1 - s_k) \cdot x_i$
13. **end for**
14. **end for**

End

3.3 Особенности реализации CFA-PN

Объединим CFA с сетью Prototype Network(PN), которая широко используется для моделей мета-обучения. Сначала применим CFA в пространстве признаков для выполнения обработки с целью улучшения признаков на всех изображениях во вспомогательном наборе для получения нового вспомогательного набора S^{aug}, а затем оптимизируем общую модель CFA-PN в соответствии со структурой сети PN.

$$S^{aug} = CFA\left(f_\phi(x_i), f_\phi(x_j)\right)\ x_i \in S, x_j \in \beta \tag{13}$$

$$p_k = \frac{1}{\left|S_k^{aug}\right|} \sum_{(x_i, y_i) \in S_k} f_\phi(x_i) \tag{14}$$

$$p_\phi(y = k \mid x) = \frac{exp\left(-d\left(f_\phi(x), p_k\right)\right)}{\sum_{k'} exp\left(-d\left(f_\phi(x), p_{k'}\right)\right)} \tag{15}$$

Следует отметить, что точки-прототипы в процессе обучения рассчитываются дважды. В первый раз рассчитывается точка-прототипы, которые используются для получения аналогичных классов для улучшения признаков. Во

второй раз точки - прототипы находятся для расчета дистанции, которая используются для предсказания класса изображения.

4 Эксперимент и анализ результатов

4.1 Набор данных

Для тестирования модели CFA-PN был использован набор данных Mini-ImageNet. Это подмножество ILSVRC-12, первоначально предложенного Виньялсом[10]. Полный набор данных содержит 100 классов, и каждый класс имеет 600 цветных изображений размером 84 × 84. Для обеспечения сопоставимости результатов был использован метод разделения набора данных, предложенный Рави и Ларошелем[16]. Использованный набор данных состоит из 64 классов для обучения, 20 классов для тестирования и 16 классов для верификации.

4.2 Структура модели

Для реализации алгоритма CFA-PN использована глубокая нейронная сеть. Модель состоит из трех частей: модуль извлечения признаков, модуль расширения признаков класса (CFA) и модуля мета-обучения. Модуль извлечения признаков сопоставляет исходное изображение с пространством признаков. Модуль CFA использует аналогичные классы для создания признаков класса и улучшает вспомогательные изображения на уровне признаков. Модуль мета-обучения классифицирует изображения из набора запросов в соответствии с комбинированным методом мета-обучения.

Модуль извлечения признаков использует четырехуровневую свертку (ConvNet) либо ResNet12[13] - две разные сетевые архитектуры нейронной сети. В 4-слойной структуре свертки каждый слой имеет одинаковую структуру, размер ядра свертки составляет 3 × 3, а количество фильтров - 64. Вход слоя отображает изображение 84 × 84 × 3 в 1600-мерное пространство признаков. Для структуры resnet12 он состоит из 4 остаточных блоков, каждый из которых содержит 3 сверточных слоя и 1 слой быстрого доступа. Между каждыми двумя остаточными

блоками существует максимальный уровень объединения 2 × 2. Число фильтров инициализируется до 64 и удваивается после каждого остаточного блока. Финальный слой отображает изображение 84 × 84 × 3 в 512-мерное пространство признаков.

Модуль CFA состоит из двустороннего LSTM, структуры уменьшения размерности g_1, структуры увеличения размерности g_2 и структуры масштабирования h. g_1 - это полносвязный слой, выходные данные которого составляют 128 измерений, размер двунаправленного скрытого слоя LSTM - 128, а g2 - это полносвязный слой, выходные данные которого составляют 512 измерений (структура извлечения признаков - resnet12) или 1600 измерений (структура извлечения признаков - 4-х слойная свертка). h содержит 2 полносвязных слоя, окончательный результат представляет собой одномерный коэффициент масштабирования, скрытый слой содержит 1024 узла, и используется функция активации ReLu. Поскольку коэффициент масштабирования имеет значение от 0 до 1, выходной слой использует сигмоид в качестве функции активации.

В качестве функции потерь модель использует кросс-энтропию, в качестве оптимизатора используется SGD. Весь процесс обучения состоит в общей сложности из 100 000 итераций, 2000 задач генерируются случайным образом для тестирования каждые 5000 раундов. Скорость обучения установлена на 0,005. Для ResNet12 коэффициент регуляризации L2 равен 0,0005.

4.3 Результаты эксперимента

Результаты эксперимента приведены в таблице 1. Проведено сравнение точности классификации для сетей Matching Network(MN)[10], Prototype Network (PN)[7], Relational Network (RN)[8], MAML, а также предложенной нами модели CFA-PN при использовании 4-х слойной ConvNet. Модель CFA-PN показала лучшие результаты, по сравнению с базовой моделью Prototype Network, уровень точности был увеличен с 49,42% до 54,34% в тесте с 1 изображением и с 68,20% до 70,98% в тесте с 5 изображениями. Также в таблице приведены результаты

тестов с использованием архитектуры ResNet12, в этом варианте предложенная модель (CFA-PN) сравнивалась с Meta-GAN[13], SNAIL[12] и TADAM[14], при этом модель CFA-PN показала лучшие результаты. Эксперимент показал, что эффект повышения точности классификации от применения CFA зависит от применяемой архитектуры, и при 4-х слойной ConvNet выше, чем при ResNet12. Это объясняется тем, что когда уровень структуры выделения признаков является глубоким, то могут быть выделены менее очевидные признаки классов в исходном изображении. Но когда количество слоев в сети извлечения признаков меньше, предложенная модель может значительно улучшить результаты.

Таблица 1 Результаты экспериментов

Модель	**Архитектура**	**1-изображение**	**5-изображений**
MN	ConvNet	43.56±0.84	55.31±0.73
PN	ConvNet	49.42±0.78	68.20±0.66
RN	ConvNet	50.44±0.82	65.32±0.70
MAML	ConvNet	48.70±1.84	55.31±0.73
CFA-PN	**ConvNet**	**54.34±0.62**	**70.98±0.45**
MetaGAN	ResNet12	52.71±0.64	68.63±0.67
SNAIL	ResNet13	55.71±0.99	68.88±0.92
TADAM	ResNet14	58.50±0.30	76.70±0.30
CFA-PN	**ResNet15**	**60.47±0.61**	**77.82±0.44**

4.4 Сравнение параметров

В этом разделе показано влияние различных параметров на процесс обучения. Так же, как и у вспомогательного набора, параметры базового набора разделены на две части: N_b и K_b, то есть базовый набор содержит в общей сложности N_b категорий, и каждая категория содержит K_b изображений. Экспериментальные результаты по 4-х слойной ConvNet показаны в таблице 2. Было обнаружено, что наилучшие экспериментальные результаты продемонстрированы при $N_b = 10$ и $K_b = 5$. Если N_b очень мало, полученные

признаки класса ненадежны; если K_b слишком мал, точные характеристики класса не могут быть сгенерированы; если N_b или K_b слишком велико, то модель больше сосредоточится на извлечении признаков базового класса и проигнорирует важные характеристики целевого класса.

Таблица 2. Сравнение производительности различных параметров

Модель	**N_b**	**K_b**	**1-изображение**	**5-изображений**
CFA-PN	5	5	53.23	70.47
CFA-PN	10	5	**54.34**	**70.98**
CFA-PN	15	5	53.87	70.45
CFA-PN	20	5	53.26	70.38
CFA-PN	25	5	53.45	70.11
CFA-PN	10	5	**54.34**	**70.98**
CFA-PN	10	10	53.76	70.52
CFA-PN	10	15	53.34	70.61
CFA-PN	10	20	53.26	70.40
CFA-PN	**10**	**25**	**52.98**	**70.31**

5. Выводы

В настоящее время у большинства методов мета-обучения есть две проблемы: они игнорируют связь между обучающим набором данных и данными для классификации; основное внимания уделяется разработке подходящего алгоритма классификации, а важность признаков игнорируется.

Чтобы решить обозначенные проблемы, предлагаем использовать фреймворк мета-обучения CFA. В процессе обучения несколько классов случайным образом выбираются из базового набора данных и формируются характеристики класса, которые позволяют распознавать изображения с более высокой точностью. В статье показано, что расширение признаков класса — это эффективный метод мета-обучения.

Литература:

1. Дэвид Форсайт, Джин Понс. Компьютерное зрение. Современный подход. // М.: Вильямс. 2018. – 960 с.
2. Саймон Хайкин. Нейронные сети. Полный курс. // М.: Вильямс. 2017. – 1104 с.
3. С. Николенко, А. Кадурин, Е. Архангельская. Глубокое обучение. Погружение в мир нейронных сетей. // СПб.: Питер. 2018. – 480 с.
4. Джош Паттерсон, Адам Гибсон. Глубокое обучение с точки зрения практика. // М.: ДМК Пресс. 2018. – 418 с.
5. В. К. Финн. Искусственный интеллект. Методология, применения, философия. // М.: Красанд. 2018. – 448 с.
6. А. И. Галушкин. Нейронные сети. Основы теории. // М.: Горячая Линия – Телеком. 2012. – 496 с.
7. Snell J., Swersky K., Zemel R. Prototypical networks for few-shot learning. // Advances in Neural Information Processing Systems.2017.
8. Sung F., Yang Y., Zhang L., et al. Learning to compare relation network for few-shot learning. // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.
9. Sun Q., Liu Y., Chua T. S., et al. Meta-transfer learning for few-shot learning. // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019.
10. Vinyals O., Blundell C., Lillicrap T., et al. Matching networks for one shot learning. // Advances in Neural Information Processing Systems.2016.
11. Girshick R., Hebert M., et al. Low-shot learning from imaginary data. // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.
12. Mishra N., Rohaninejad M., Chen X., et al. A simple neural attentive meta-learner. arXiv preprint arXiv:1707.03141. 2017.

13. ZhangR, CheT, GhahramaniZ, etal. Metagan: an adversarial approach to few-shot learning. // Advances in Neural Information Processing Systems. 2018.
14. Oreshkin B., López P. R., Lacoste A. Tadam: task dependent adaptive metric for improved few-shot learning. // Advances in Neural Information Processing Systems. 2018.
15. Finn C., Abbeel P., Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. // International Conference on Machine Learning. 2017.
16. Ravi S., Larochelle H. Optimization as a model for few-shot learning. // International Conference on Learning Representations, 2017.

Literature

1. David Forsyth, Jean Pons. Computer vision. Modern approach. // M.: Williams. 2018. – 960 p.
2. Simon Khaikin. Neural networks. Complete course. // M.: Williams. 2017. – 1104 p.
3. S. Nikolenko, A. Kadurin, E. Arkhangelskaya. Deep Learning. Dive into the world of neural networks. // SPb.: Peter. 2018. – 480 p.
4. Josh Patterson, Adam Gibson. Deep learning from a practice perspective. // M.: DMK Press. 2018. – 418 p.
5. V. K. Finn. Artificial Intelligence. Methodology, applications, philosophy. // M.: Krasand. 2018. – 448 p.
6. A. I. Galushkin. Neural networks. Foundations of the theory. // M.: Hot Line - Telecom. 2012. – 496 p.
7. Snell J., Swersky K., Zemel R. Prototypical networks for few-shot learning. // Advances in Neural Information Processing Systems. 2017.
8. Sung F., Yang Y., Zhang L., et al. Learning to compare relation network for few-shot learning. // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.
9. Sun Q., Liu Y., Chua T.S., et al. Meta-transfer learning for few-shot learning. // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019.

10. Vinyals O., Blundell C., Lillicrap T., et al. Matching networks for one shot learning. // Advances in Neural Information Processing Systems. 2016.
11. Girshick R., Hebert M., et al. Low-shot learning from imaginary data // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.
12. Mishra N., Rohaninejad M., Chen X., et al. A simple neural attentive meta-learner. arXiv preprint arXiv: 1707.03141. 2017.
13. ZhangR, CheT, GhahramaniZ, etal. Metagan: an adversarial approach to few-shot learning // Advances in Neural Information Processing Systems. 2018.
14. Oreshkin B., López P. R., Lacoste A. Tadam: task dependent adaptive metric for improved few-shot learning. // Advances in Neural Information Processing Systems. 2018.
15. Finn C., Abbeel P., Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. // International Conference on Machine Learning. 2017.
16. Ravi S., Larochelle H. Optimization as a model for few-shot learning. // International Conference on Learning Representations. 2017.