Преодоление ограничений обучения сиамских нейронных сетей в задаче оптического распознавания символов

А. К. Мокин*

Разработка моделей глубокого обучения для задач классификации представляет особую сложность при работе с большим количеством классов в условиях ограниченных данных и вычислительных ресурсов. Метрическое обучение предлагает перспективный подход к решению этой проблемы, хотя его эффективность часто ограничивается присущими недостатками стандартных функций потерь, таких как контрастивная и триплетная потери, а также неоптимальными стратегиями выбора обучающих примеров. В данной работе представлены решения этих проблем: новый автовероятностный метод майнинга для выбора примеров и улучшенная метрическая функция потерь. Предложенный автовероятностный метод майнинга помогает выбирать наиболее информативные пары примеров для обучения сиамских нейронных сетей. В сочетании с ранее разработанным методом автокластеризации этот метод повышает эффективность обучения, максимизируя полезность данных и минимизируя вычислительные затраты. Помимо этого, вводится новая метрическая функция потерь на основе триплетов, учитывающая специфику кластеров и разработанная для преодоления конкретных недостатков традиционных контрастивной и триплетной функций потерь, тем самым улучшая процесс формирования признаковых эмбеддингов. Эффективность предложенных методов была подтверждена в ходе экспериментов по оптическому распознаванию символов на наборах данных PHD08 (корейский алфавит) и Omniglot. Для полного корейского алфавита в наборе данных PHD08 новая функция потерь со случайным майнингом достигла точности классификации $82,6\%$, установив новый эталонный показатель. Используя сокращенный алфавит, был установлен базовый уровень в $88,6\%$ на PHD08. Применение только автовероятностного метода майнинга улучшило точность до $90,6\%$ ($+2,0\%$), а его комбинация с автокластеризацией дополнительно увеличила её до $92,3\%$ ($+3,7\%$). На наборе данных Omniglot предложенный метод майнинга достиг $92,32\%$, а в

* *Мокин Арсений Кириллович* — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: mokinak@my.msu.ru, ORCID: 0009-0005-8044-0245.

Mokin Arseniy Kirillovich — Ph.D. Student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

сочетании с автокластеризацией этот показатель вырос до 93, 17%. Результаты демонстрируют, что предложенные функция потерь и стратегия майнинга представляют собой эффективное решение для задач метрического обучения, особенно в сценариях, характеризующихся большим количеством классов и ограниченными ресурсами.

Ключевые слова: глубокое метрическое обучение, оптическое распознавание символов, сиамские нейронные сети, распознавание паттернов..

1. Введение

Методы глубокого обучения широко используются в последние годы, влияя на такие области, как обработка естественного языка и компьютерное зрение. Сверточные нейронные сети (CNN), способные эффективно строить иерархические представления из исходных данных, стали краеугольным камнем во многих приложениях. Оптическое распознавание символов (OCR) — одно из таких приложений, где CNN показали высокую производительность. Системы OCR необходимы для оцифровки и извлечения текста из документов, изображений и других отсканированных материалов, поскольку они упрощают такие задачи, как текстовый анализ и автоматический ввод данных [1].

Способность CNN автоматически обучаться дискриминативным признакам непосредственно из данных пикселей объясняет их эффективность в задачах OCR [2]. CNN способны улавливать детали, необходимые для точной идентификации символов, используя сверточные слои для извлечения иерархических структур и локальных закономерностей. Например, в простых задачах OCR часто количество нейронов в последнем слое нейронной сети соответствует количеству распознаваемых символов. В таком случае обработка огромного алфавита (Рис. 1) в OCR является нетривиальной задачей.

Рис. 1: Похожие символы Корейского алфавита.

В таких условиях количество нейронов в последнем слое резко возрастает, увеличивая объем данных и требуя большого объема вычислительных мощностей и обучающих данных. В результате обучение этих сетей становится неэффективным.

Метрическое обучение помогает избежать этой проблемы [3]. Метрическое обучение работает путем сопоставления точки в метрическом пространстве с входным вектором, гарантируя, что точка будет расположена как можно ближе к точкам того же класса и как можно дальше от точек других классов [4].

Существует метрический подход, называемый сиамскими нейронными сетями (SNN). SNN, впервые представленные Дж.Бромли и др. [5], привлекли значительное внимание благодаря своей способности обучаться устойчивым представлениям в сценариях с ограниченными размеченными данными. В отличие от традиционных архитектур CNN, которые направлены на непосредственное предсказание классов, SNN изучают метрику сходства между входами, позволяя им улавливать тонкие различия и сходства даже без достаточного количества размеченных данных.

SNN состоит из нескольких ветвей CNN с общими весами. Часто евклидова норма используется для расчета расстояний между выходами ветвей, и полученные векторы затем отправляются в функцию потерь.

Основной вклад данной работы заключается в следующем:

- Разработка метода майнинга для обучения SNN, который улучшает представление данных и снижает вычислительные затраты.
- Предложение новой метрической функции потерь, которая сочетает преимущества контрастивной и триплетной функций потерь, вводя новый механизм для обеспечения компактности кластеров.
- Экспериментальная оценка эффективности предложенных методов в задачах OCR с использованием наборов данных PHD08 и Omniglot, демонстрирующая значительное улучшение точности классификации и вычислительной эффективности.

2. Связанные работы

OCR включает классификацию символов и их сопоставление с соответствующими классами. В то время как для некоторых языков, таких как английский, достигнуты высокие показатели распознавания [6], задача становится все более сложной, когда размер алфавита превышает 10000, требуя новых методологий обучения и распознавания.

2.1. Контрастивная функция потерь

Контрастивная функция потерь [7, 8] (1), первоначально предложенная С.Чопра и др. для задач верификации лиц, стала популярна в метрическом обучении, как показано в исследованиях [9–11].

Контрастивная функция потерь (1) использует попарное расстояние и приближает положительные пары как можно ближе друг к другу и отталкивает отрицательные пары на расстояние друг от друга не меньше выбранного порога (α).

$$L^{\alpha}(x_i, x_j; f) = y_{ij} d_{ij}^2 + (1 - y_{ij}) \cdot max(0, \alpha - d_{ij})^2, \tag{1}$$

где $d_{ij} = \|f(x_i) - f(x_j)\|_2$ — евклидово расстояние между парой, а y_{ij} возвращает 1, если $y_i = y_j$, и 0 в противном случае. x_i, x_j представляют входы SNN (f).

Отсутствие естественного механизма остановки является одним из заметных недостатков контрастивной функции потерь. Эта функция потерь способствует минимизации расстояния между положительной парой до нуля. На практике нейронная сеть может использовать вычислительные ресурсы для уточнения пар, которые уже близки к сходимости, не позволяя ей исследовать и оптимизировать другие примеры.

В работе [12] упоминается другая интерпретация контрастивной функции потерь, которая имеет схожий смысл, но сформулирована иначе. В данной работе используется классическая версия контрастивной функции потерь, предложенная в [7, 8].

2.2. Триплетная функция потерь

Триплетная функция потерь (2), предложенная и описанная в [13], выступает альтернативным подходом к обучению эмбеддингов в задачах метрического обучения. В отличие от контрастивной функции потерь, которая работает с парами образцов, триплетная функция потерь работает с триплетами, состоящими из якоря, положительного примера из того же класса, что и якорь, и отрицательного примера из другого класса.

$$L^{\alpha}(x_a, x_p, x_n, f) = max(0, \|f(x_a) - f(x_p)\|_2 - \|f(x_a) - f(x_n)\|_2 + \alpha) \tag{2}$$

где x_a, x_p, x_n представляют якорь, положительный и отрицательный входы SNN (f).

Цель триплетной функции потерь — обучить сеть уменьшать расстояние между якорями и положительными примерами, одновременно

увеличивая расстояние между якорями и отрицательными примерами как минимум на заданный порог (α).

Однако важно отметить, что триплетная функция потерь вводит кубическую сложность [14], что означает, что возможных триплетов значительно больше, чем пар. Не все триплеты одинаково информативны для обучения, и включение всех возможных триплетов может привести к более медленной сходимости и вычислительной неэффективности. Поэтому выбор «сложных» триплетов, которые особенно информативны [15], эффективен для обучения модели и достижения более быстрой сходимости и улучшенной производительности.

Предлагаемый в этой работе метод майнинга (Раздел 4) представляет стратегию для решения проблемы «сложных» триплетов.

Другая проблема, связанная с триплетной функцией потерь, связана с отсутствием прямого требования близости между якорем и положительными экземплярами. Акцент делается на обеспечении того, чтобы расстояние между якорями и положительными примерами было меньше расстояния между якорями и отрицательными примерами на заданный порог (α).

Предлагаемая метрическая функция потерь (Раздел 4.4) устраняет ключевые ограничения классических функций потерь (контрастивная и триплетная), улучшает способность модели изучать дискриминативные признаки и достигать лучшей производительности в задачах метрического обучения.

2.3. Квадруплетная функция потерь

Существует квадруплетная функция потерь [16], которая расширяет традиционную триплетную функцию потерь за счет введения дополнительного отрицательного примера. Этот подход накладывает более строгое ограничение, гарантируя, что пара якорь-положительный пример не только ближе, чем первый отрицательный пример, но и ближе, чем второй отрицательный пример, тем самым улучшая дискриминацию признаков. Хотя этот метод улучшает обобщаемость в метрическом обучении, он также увеличивает вычислительную сложность и требует более продвинутых стратегий майнинга для эффективного отбора примеров. Несмотря на эти проблемы, квадруплетная функция потерь продемонстрировала улучшенную производительность.

$$\begin{aligned} L^{\alpha,\beta}(x_a, x_p, x_{n_1}, x_{n_2}) = &\max\left(0, \|f(x_a) - f(x_p)\|_2 - \|f(x_a) - f(x_{n_1})\|_2 + \alpha\right) \\ &+ \max\left(0, \|f(x_a) - f(x_p)\|_2 - \|f(x_{n_1}) - f(x_{n_2})\|_2 + \beta\right) \end{aligned} \tag{3}$$

2.4. Декомпозиция символов

Некоторые языки допускают декомпозицию символов, позволяя сегментировать их на отдельные компоненты для распознавания и последующей композиции конечного результата [9, 17]. Этот подход устраняет необходимость в нейронных сетях с десятками и даже сотнями тысяч выходов. Однако он обладает недостатками, такими как уязвимость к искажениям изображения и сильная зависимость от качества сегментации. Некоторые исследователи [18] выбирают глубокие нейронные сети с большим количеством обучаемых параметров, которые предлагают высокое качество, но требуют значительных вычислительных ресурсов и не подходят, например, для мобильных устройств и т.п.

2.5. Методы майнинга

Кроме того, обучение метрических сетей осложнено выбором правил генерации пар/триплетов. Хотя случайный майнинг (random-mining) является обычным явлением [19], в некоторых работах уделяют особое внимание решению этой проблемы, сосредотачиваясь на сходстве. Например, в [20] используется агрессивная стратегия жесткого майнинга (hard-mining), выбирающая пары с наибольшей ошибкой для обратного распространения, чтобы обучать сеть исключительно на сложных примерах. Недостатком метода является то, что возможный шум в данных может затруднить достижение локального минимума.

Другой интересный момент — математическое обоснование [21] эффективности жесткого майнинга в метрическом обучении с использованием теоремы изометрической аппроксимации. Авторы показывают, что жесткий майнинг эквивалентен минимизации расстояния Хаусдорфа между нейронной сетью и ее идеальной функцией, что объясняет эмпирический успех этого подхода.

В [22] авторы предлагают Easy Positive Triplet Mining (EPTM) для решения проблемы нестабильности традиционных методов жесткого майнинга. Hard Positive Mining часто выбирает выбросы или неправильно размеченные примеры, что приводит к зашумленной оптимизации и плохому обобщению. EPTM смягчает эту проблему, выбирая более простые, но информативные положительные примеры, обеспечивая лучшую кластеризацию при сохранении стабильного обучения. В сочетании с Semi-Hard Negative Mining, который избегает экстремальных отрицательных примеров, этот подход приводит к более надежным эмбеддингам и улучшенной сходимости.

Кроме того, методы майнинга пар на основе расстояний, такие как подход, предложенный в [10], демонстрируют эффективные результаты

за счет генерации образцов на основе создания вектора расстояний для всех возможных негативных пар. Хотя этот тип метода перспективен с точки зрения качества обучения, он требует значительных временных затрат.

Метод авто-кластеризации [23] частично решает проблему майнинга в SNN. Этот метод основан на создании кластеров, то есть групп, состоящих из классов, схожих с точки зрения нейросети. Использование кластеров позволяет нейросети уделять больше внимания классам, которые трудно различить. Согласно методу авто-кластеризации [23], отрицательный пример выбирается из того же кластера, что и положительный пример.

В процессе обучения нейронной сети можно выяснить, какие классы находятся далеко от своего кластера в метрическом пространстве, и попытаться вручную увеличить вероятность их выбора при генерации положительного класса. Для такой задачи автоматические методы рекомендуются как наиболее эффективные и удобные решения, не требующие тяжелой настройки.

3. Данные

3.1. Распознавание корейского алфавита

3.1.1. Набор данных PHD08

Корейский алфавит (Хангыль) известен своим большим набором символов и высоким визуальным сходством между многими из них, что делает его особенно сложным для задач OCR. Это хорошо согласуется с целями данного исследования, поскольку создается основа для экспериментов, которые проверяют устойчивость и дискриминативную способность предложенных методов. Таким образом, для оценки предложенных методов был выбран набор данных рукописных символов хангыля под названием PHD08 [24]. Более того, тот же набор данных использовался в предыдущих работах [9, 10, 23]. Набор данных содержит 2350 классов, и каждый класс имеет 2187 изображений корейских символов. Всего имеется 5139450 бинарных изображений разных размеров, с разными поворотами и искажениями. Примеры изображений набора данных показаны на Рис. 2.

Рис. 2: Примеры из набора данных PHD08.

3.1.2. Синтетические обучающие данные

Для оценки объективности предложенного подхода были сгенерированы синтетические данные с использованием того же подхода, что и в [23], для обучения нейросети таким же образом, как и в работах [9,10,23], используя в среднем восемь аналогичных шрифтов для каждого класса. Общее количество классов составило 11172 символа корейского алфавита. Более того, в проведенных экспериментах было решено не генерировать все возможные классы и сделать количество классов равным размеру набора данных для оценки. Это решение было принято для более точной оценки предложенного авто-вероятностного метода майнинга (Раздел 4.2), потому что сеть могла обучаться на некоторых классах, которых нет в наборе данных для оценки. И этот факт в сочетании с рассматриваемым подходом мог значительно увеличить вероятности появления таких ненужных для оценки символов.

3.2. Набор данных Omniglot

Набор данных Omniglot был собран Б.Лейком и его сотрудниками в MIT через Amazon Mechanical Turk для создания стандартного теста для обучения на небольшом количестве примеров в области распознавания рукописных символов [25].

Omniglot содержит 1623 символа из 50 различных алфавитов (Рис. 3). Он состоит не только из международных языков, таких как корейский и латинский, но и из менее известных местных диалектов и вымышленных наборов символов, таких как Futurama и клингонский. Каждый из них был нарисован вручную 20 разными людьми. Количество букв в каждом алфавите значительно варьируется приблизительно от 15 до 40 символов. Набор данных состоит из двух подмножеств: *images_background*, содержащий 19289 изображений, и *images_evaluation*, содержащий 13180 изображений.

3.3. Аугментация

Аугментация данных [23] применялась к изображениям в процессе обучения с вероятностью 0.7 для каждого образца и со следующими искажениями:

- Проективное преобразование — каждая точка изображения преобразуется, и значения смещения по осям x и у выбираются случайным образом в диапазоне $[0.0, 1.0]$, где минимальная и максимальная доли смещения представлены на оси x и оси y от ширины изображения.

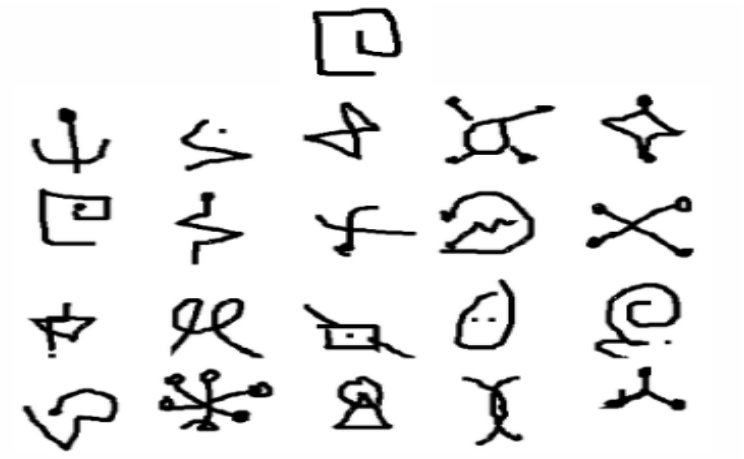

Рис. 3: Примеры набора данных Omniglot из исходной статьи [26].

- Поворот — вращение изображения на угол в диапазоне $[-5, 5]$ градусов.
- Масштабирование — масштабирование изображения до заданного размера, а затем масштабирование результата до исходного размера с коэффициентами масштабирования $min = 0.7$ и $max = 0.9$ от исходного изображения по ширине и высоте, чтобы сделать изображение пикселизированным.

3.4. Настройки для экспериментов

Для всех экспериментов в этой статье изображения изменялись до размера 37×37 пикселей и преобразовывались в оттенки серого, чтобы соответствовать стандартам предобработки предыдущих исследований для объективного сравнительного анализа.

В течение каждой эпохи выполнялось 50 итераций с генерацией 10240 элементов (пар/триплетов) на итерацию. Для контрастивной функции потерь пары динамически выбирались (3072 подлинных пары и 7168 пар-самозванцев), в то время как CATML использовала триплеты. Всего 10240 элементов генерировались как для обучающего, так и для валидационного наборов в каждую эпоху. Эти настройки применялись единообразно ко всем наборам данных, используемым в экспериментах, включая PHD08 и Omniglot, чтобы сохранить условия обучения равными и произвести объективную оценку предложенных методов.

4. Предлагаемый метод

4.1. Авто-вероятностный метод майнинга

Присвоение определенных вероятностей классам во время обучения нейронной сети может улучшить способность модели распознавать определенные классы. Приоритезация некоторых классов с большей вероятностью побуждает нейросеть выбирать их чаще, что улучшает способность нейросети распознавать эти символы.

Присвоение вероятностей вручную вызывает затруднения на практике. Во-первых, когнитивная нагрузка по поддержанию всесторонних знаний о нескольких разнообразных классах выходит за пределы человеческих возможностей, когда количество классов исчисляется тысячами. Кроме того, хотя автоматическое присвоение вероятности возможно для некоторых организованных данных, таких как языки с декомпозицией символов на основе ключей, многие реальные ситуации не имеют такой внутренней структуры, что делает присвоение вручную проблематичным. При этом, автоматизация может быть применена к более широкому кругу объектов, независимо от их типа или сложности.

В результате авто-вероятностный метод майнинга (Auto-Probabilistic Method — APM) предлагает более гибкий подход. Нейросеть приобретает способность динамически адаптироваться и расставлять приоритеты классам в соответствии с их значимостью для обучения путем автоматического вычисления вероятностей для каждого класса во время обучения. Это позволяет нейросети определять, какие классы требуют большего внимания, и выбирать их чаще, что улучшает распознавание и оптимизирует процесс обучения. Следовательно, метод APM предлагает перспективный способ повышения эффективности обучения нейронных сетей. Этот подход отличается от методов равномерного распределения классов и ищет классы с наибольшими средними расстояниями между каждым примером класса и соответствующим ему кластером.

4.2. Автоматическое вычисление вероятностей появления классов

Для улучшения процесса обучения с использованием метода APM вычисляется вектор вероятностей классов P^e на каждой эпохе e. Метод использует средние расстояния между примерами и соответствующими центроидами их классов для определения вероятностей классов.

Пусть $f(x_{i,j})$ — вектор признаков входного изображения $x_{i,j}$ (i-й класс, j-й образец), полученный из модели f. Определим d как расстояние ($L2$

норма разности) между $f(x_{i,j})$ и центром c_i i-го класса, где c_i представляет собой средний вектор признаков примеров класса i. Определим вектор вероятностей классов $\hat{P}^e$ как

$$\hat{P}_i^e = \frac{\left(\frac{1}{M_i}\sum_{j=1}^{M_i} d(f(x_{i,j}), c_i)\right)^\gamma}{\sum_{k=1}^{N}\left(\frac{1}{M_k}\sum_{j=1}^{M_k} d(f(x_{k,j}), c_k)\right)^\gamma}, \quad i = 1, \ldots, N$$

где

- M_i — количество примеров i-го класса;
- $x_{i,j}$ — j-й пример i-го класса;
- N — общее количество классов;
- e — количество эпох;
- γ — коэффициент усиления, контролирующий влияние расстояний для выделения доминирующих классов.

Тогда итоговый вектор вероятностей классов P^e на эпохе e вычисляется как

$$P^e = (1 - w) \cdot \hat{P}^e + w \cdot P^{e-1}, \tag{4}$$

где

- P^{e-1} — вектор вероятностей классов с предыдущей эпохи ($P_i^0 = \frac{1}{N}, \quad i = 1, ..., N$);
- $w \quad (0 \leqslant w \leqslant 1)$ — весовой коэффициент, контролирующий вклад вероятностей предыдущей эпохи;
- γ - степень усиления для обновленных вероятностей для выделения доминирующих классов.

Наконец, P^e нормализуется для формирования дискретного распределения вероятностей, которое используется для майнинга положительных пар во время обучения. Таким образом, это распределение гарантирует, что положительные пары выбираются более эффективно за счет приоритизации классов с большими расстояниями от текущих образцов.

4.3. Улучшения метода авто-кластеризации

Базовый метод авто-кластеризации [23] включает в себя следующие шаги: вычисление всех норм между центроидами (идеальными векторами) каждого класса, их сортировка и, поскольку их может быть много, выбор только некоторых из них.

В предыдущей версии метода авто-кластеризации [23] было много гиперпараметров, необходимых для обучения сети. И такой подход неудобен для экспериментов. На этом этапе было решено упростить, но сохранить основную суть и оставить всего два параметра:

- вероятность выбора класса из кластера для отрицательной пары/триплета (θ);
- количество наименьших норм, рассматриваемых для генерации кластеров (η).

Этот метод используется для майнинга отрицательных примеров в парах/триплетах во время обучения.

4.4. Метрическая функция потерь на основе триплетов с учетом кластеров

Предлагается новая метрическая функция потерь на основе триплетов с учетом кластеров (Cluster-aware triplets-based metric loss - CATML) (5), сочетающая преимущества контрастивной и триплетной функций потерь.

$$CATML = \rho \cdot g_1 + \tau \cdot g_2 + \xi \cdot g_3 \tag{5}$$

где

- $\rho \cdot g_1$ — вклад контрастивной функции потерь, где ρ отвечает за уменьшение расстояния между якорем и положительным примером;
- $\tau \cdot g_2$ — вклад триплетной функции потерь, где τ регулирует принцип, чтобы расстояние между якорем и положительным примером было меньше расстояния между якорем и отрицательным примером на выбранное значение, равное или большее α;
- $\xi \cdot g_3$ — вклад кластеров, где ξ отвечает за стабильность, чтобы кластеры, созданные во время обучения, не распадались и чаще получали градиент, чтобы оставаться на том же месте.

В этой работе экспериментально выбраны значения: $\rho = 0.1$, $\tau = 1.0$, $\xi = 1.0$ и $\alpha = 10$. Компоненты g_1, g_2 и g_3 определяются следующим образом:

- $g_1 = d_{AP} = \|s(f(x_a)) - s(f(x_p))\|_2$;
- $g_2 = s(d_{AP} - d_{AN} + \alpha) =$
  $= s(\|s(f(x_a)) - s(f(x_p))\|_2 - \|s(f(x_a)) - s(f(x_n))\|_2 + \alpha)$, где
  α значение порога;

- $g_3 = \frac{1}{M_i} \sum_{j=1}^{M_i} d(f(x_{i,j}), c_i),\ i = 1, \ldots, N$, где $x_{i,j} \in (x_a, x_p, x_n)$, c_i - идеальный вектор, соответствующий определенному классу (средний вектор примеров данного класса).

Здесь f обозначает модель глубокого обучения, используемую для извлечения признаков, а $s(x) = softrelu(x)$ — функция активации.

5. Эксперименты

5.1. Архитектура

Архитектура (Таб. 1) выбрана из работ [9, 10, 23] для объективности в сравнении.

Таблица 1: Список слоев метрической нейронной сети.

Layer	**Type**	**Parameters**	**Output**
1	conv	16 filters 3×3, stride 1	$35 \times 35 \times 16$
2	conv	16 filters 5×5, stride 2	$18 \times 18 \times 16$
3	conv	16 filters 3×3, stride 1	$18 \times 18 \times 16$
4	conv	24 filters 5×5, stride 2	$9 \times 9 \times 24$
5	conv	24 filters 3×3, stride 1	$9 \times 9 \times 24$
6	conv	24 filters 3×3, stride 1	$9 \times 9 \times 24$
7	fc	25 outputs	$1 \times 1 \times 25$

Функция активации *softsign* (6) подробно описана в работах [27, 28] и рекомендуется как наиболее подходящая для такого рода проблем. *Softsign* — это ограниченная функция активации, которая способствует стабильной сходимости и устраняет любые численные несоответствия. Эта функция также использовалась в работах [9, 10, 23].

$$softsign(x) = \frac{x}{1 + |x|} \tag{6}$$

5.2. Обучение

5.2.1. Синтетические данные

Проведены эксперименты, в которых сеть обучалась на синтетических изображениях корейских символов с новой функцией CATML. Были рассмотрены два размера обучающего алфавита: полный (11172 класса) и

сокращенный (2350 классов). В оцениваемом наборе данных PHD08 всего 2350 классов, поэтому сокращенный подход позволяет методу APM оказывать более объективное влияние и избегает фокусирования на ненужных для оценки классах.

Была обучена нейронная сеть с CATML на полном алфавите с методом случайного майнинга (random-mining). В следующем эксперименте нейросеть обучалась с той же функцией потерь на сокращенном алфавите со случайным майнингом, авто-кластеризацией [23] и авто-вероятностным майнингом. Более того, специфика двух последних методов позволяет использовать их вместе, где метод APM используется для майнинга положительных примеров, а авто-кластеризация [23] отвечает за отрицательные примеры.

В проведенных экспериментах метод APM показал наилучшую точность только при $\gamma = 1$, но для метода APM + Авто-кластеризация [23] наилучший результат показал $\gamma = 2$. Метод авто-кластеризации [23] продемонстрировал высокую точность при $\theta = 0.5$ и $\eta = 1000$, а метод APM достиг наилучшего результата при $w = 0$, что означает отсутствие зависимости от вероятностей с предыдущей эпохи, но этот весовой коэффициент также может быть рассмотрен для будущих экспериментов.

5.2.2. Набор данных Omniglot

Для набора данных Omniglot была обучена нейронная сеть на выделенных 60% от общего объема данных. Тестовая и валидационная части были установлены такими же, как в [2], обе равны 20%. Было зафиксировано единое количество обучающих примеров на алфавит, чтобы каждый алфавит встречался с одинаковой частотой во время обучения, хотя это не гарантируется для отдельных классов символов в каждом алфавите. Все параметры, использованные для предложенных методов, были такими же, как и для экспериментов на корейском алфавите.

5.3. Оценка

Точность (Acc) для классификации определялась как

$$Acc = \frac{N_{correct}}{N_{total}} \cdot 100\%, \tag{7}$$

где N_{total} — размер набора данных, а $N_{correct}$ — количество изображений, правильно классифицированных нейросетью.

6. Результаты

Все эксперименты следовали общему формату оценки с разделением данных на три отдельных набора: обучающий, валидационный и тестовый. Валидационный набор использовался исключительно для настройки гиперпараметров и выбора модели посредством мониторинга сходимости по эпохам, в то время как тестовый набор оставался строго изолированным для окончательной оценки. Итоговые метрики вычислялись путем однократной оценки на тестовом наборе с использованием оптимальной контрольной точки модели, выбранной на основе ошибки валидационной части. Для обеспечения воспроизводимости и предотвращения переобучения никакие обновления параметров или архитектурные решения не предпринимались под влиянием наблюдений тестового набора во время обучения.

Для каждого эксперимента предоставлена таблица результатов, содержащая как обучающую, так и тестовую точность для выбранной модели. Метрики на обучающем наборе данных включены специально для проверки поведения модели; соответствие точности между обучающей и тестовой выборками предоставляет эмпирические доказательства отсутствия переобучения.

6.1. PHD08

Результаты экспериментов с большим количеством классов (11172) представлены в Таб. 2. Результаты случайного майнинга, жесткого майнинга и майнинга на основе расстояний были взяты из работы [10]. Функция CATML показала заметную точность по сравнению со всеми предыдущими результатами.

Таблица 2: Точность классификации для набора данных PHD08 с размером алфавита 11172 классов.

Loss	**Mining Method**	**Train Acc**	**Test Acc**
Contrastive loss	Random-mining [19]	-	63.7%
Contrastive loss	Hard-mining [20]	-	64.5%
Contrastive loss	Distance-based mining [10]	-	69.7%
Contrastive loss	Auto-clustering [23]	79.3%	76.1%
CATML	**Random-mining**	86.2%	**82.6**%

В Таб. 3 видно, что предложенный метод АРМ улучшает точность классификации. Более того, предполагается, что этот метод является

эффективным и подходящим решением независимо от типа распознаваемого объекта. Кроме того, он может успешно применяться как с ранее предложенным методом авто-кластеризации [23], так и без него.

Таблица 3: Точность классификации для набора данных PHD08 с размером алфавита 2350 классов.

Loss	**Mining Method**	**Train Acc**	**Test Acc**
CATML	Random-mining	89.1%	88.6%
CATML	Auto-probabilistic	94.8%	90.6%
CATML	Auto-clustering	93.7%	91.1%
CATML	**Auto-probabilistic + Auto-clustering**	95.1%	**92.3**%

6.2. Omniglot

Таблица 4: Точность классификации для набора данных Omniglot.

Loss	**Mining Method**	**Train Acc**	**Test Acc**
CATML	Random-mining	93.33%	91.25%
CATML	Auto-clustering	95.21%	91.60%
CATML	Auto-probabilistic	94.78%	92.32%
CATML	**Auto-probabilistic + Auto-clustering**	95.89%	**93.17**%

В Таб. 4 можно увидеть, что использование предложенных методов улучшает точность классификации. Также стоит отметить, что метод авто-кластеризации [23] не улучшает точность значительно. Это можно объяснить слишком большим разнообразием набора данных, поскольку он содержит визуально различные алфавиты, в том числе включает корейский.

7. Выводы

В работе предложен авто-вероятностный метод майнинга, который повышает точность распознавания сиамских нейронных сетей, делая их более эффективными для задач классификации, таких как OCR. Этот метод может использоваться как в сочетании с методом авто-кластеризации, так и без него. Он улучшает обучение сети, достигая высокой точности, как продемонстрировано на наборах данных PHD08 и Omniglot. Разработанный метод не требует каких-либо знаний о природе объекта и может использоваться для обучения нейронных сетей для распознавания

любого типа объектов: символы, дескрипторы особых точек, лица и так далее. В дополнение к этому, предложенная метрическая функция потерь на основе триплетов с учетом кластеров сочетает в себе преимущества контрастивной и триплетной функций потерь.

В будущих исследованиях предложенные методы, включая новую функцию потерь и стратегию авто-вероятностного майнинга, могут быть рассмотрены в контексте обучения с одним примером (one-shot learning) и повторной идентификации (re-identification). Эти задачи вызывают сложности в условиях ограниченных данных и необходимости надежного определения схожести, что хорошо согласуется с сильными сторонами метрического обучения. Применяя предложенные методы к этим задачам, можно исследовать их потенциал к повышению обобщающей способности, снижению зависимости от больших наборов данных и достижению конкурентоспособной производительности.

Список литературы

[1] Khaustov P.A., “Algorithms for handwritten character recognition based on constructing structural models”, *Comput. Opt.*, **41** (2017), 67–78. DOI: 10.18287/2412-6179-2017-41-1-67-78.

[2] Koch G., Zemel R., Salakhutdinov R., “Siamese neural networks for one-shot image recognition”, *Proceedings of the 32nd International Conference on Machine Learning*, 2015, 7–9 July 2015; Volume 2.

[3] Hoffer E., Ailon N., “Deep metric learning using triplet network”, *Proceedings of the International Workshop on Similarity-Based Pattern Recognition*, 2015, 84–92.

[4] Oh Song H., Xiang Y., Jegelka S., Savarese S., “Deep metric learning via lifted structured feature embedding”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, 4004–4012.

[5] Bromley J., Guyon I., LeCun Y., Säckinger E., Shah R., “Signature verification using a “siamese” time delay neural network”, *Adv. Neural Inf. Process. Syst.*, **6** (1993), 737–744.

[6] Tafti A.P., Baghaie A., Assefi M., Arabnia H.R., Yu Z., Peissig P., “OCR as a service: An experimental evaluation of Google Docs OCR, Tesseract, ABBYY FineReader, and Transym”, *Proceedings of the 12th International Symposium on Visual Computing*, 2016, 735–746.

[7] Chopra S., Hadsell R., LeCun Y., “Learning a similarity metric discriminatively, with application to face verification”, *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, **1** (2005), 539–546. DOI: 10.1109/CVPR.2005.202.

[8] Hadsell R., Chopra S., LeCun Y., “Dimensionality reduction by learning an invariant mapping”, *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, **2** (2006), 1735–1742. DOI: 10.1109/CVPR.2006.100.

[9] Ilyuhin S.A., Sheshkus A.V., Arlazarov V.L., “Recognition of images of Korean characters using embedded networks”, *Proceedings of the Twelfth International Conference on Machine Vision (ICMV 2019)*, **11433** (2020), 1143311.

[10] Kondrashev I.V., Sheshkus A.V., Arlazarov V.V., “Distance-based online pairs generation method for metric networks training”, *Proceedings of the Thirteenth International Conference on Machine Vision*, **11605** (2021), 1160508.

[11] Wang X., Hua Y., Kodirov E., Hu G., Robertson N.M., “Deep metric learning by online soft mining and class-aware attention”, *Proceedings of the AAAI Conference on Artificial Intelligence*, **33** (2019), 5361–5368. DOI: 10.1609/aaai.v33i01.33015361.

[12] Wang F., Liu H., “Understanding the behaviour of contrastive loss”, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 2495–2504.

[13] Schroff F., Kalenichenko D., Philbin J., “Facenet: A unified embedding for face recognition and clustering”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, 815–823.

[14] Yuan Y., Chen W., Yang Y., Wang Z., “In defense of the triplet loss again: Learning robust person re-identification with fast approximated triplet loss and label distillation”, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, 354–355.

[15] Dalwadi Bijal N., Shah Vatsal, “Deep Embedding Learning for Printed Word Spotting using Triplet CNNs and Transfer Learning”, *Proceedings of the IEEE 5th International Conference on Soft Computing for Security Applications (ICSCSA)*, **73** (2025), 384–389.

[16] Chen W., Chen X., Zhang J., Huang K., “Beyond triplet loss: A deep quadruplet network for person re-identification”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, 403–412.

[17] Franken M., van Gemert J.C., “Automatic Egyptian hieroglyph recognition by retrieving images as texts”, *Proceedings of the 21st ACM international conference on Multimedia*, 2013, 765–768. DOI: 10.1145/2502081.2502199.

[18] Kim Y.g., Cha E.y., “Learning of Large-Scale Korean Character Data through the Convolutional Neural Network”, *Proceedings of the Korean Institute of Information and Commucation Sciences Conference*, 2016, 97–100.

[19] Bell S., Bala K., “Learning visual similarity for product design with convolutional neural networks”, *ACM Trans. Graph. (TOG)*, **34** (2015), 1–10. DOI: 10.1145/2766959.

[20] Simo-Serra E., Trulls E., Ferraz L., Kokkinos I., Fua P., Moreno-Noguer F., “Discriminative learning of deep convolutional feature point descriptors”, *Proceedings of the IEEE International Conference on Computer Vision*, 2015, 118–126.

[21] Xu A., Hsieh J.Y., Vundurthy B., Cohen E., Choset H., Li L., “Mathematical Justification of Hard Negative Mining via Isometric Approximation Theorem”, *arXiv*, 2022.

[22] Xuan H., Stylianou A., Pless R., “Improved embeddings with easy positive triplet mining”, *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, 2474–2482.

[23] Mokin A.K., Gayer A.V., Sheshkus A.V., Arlazarov V.L., “Auto-clustering pairs generation method for Siamese neural networks training”, *Proceedings of the Fourteenth International Conference on Machine Vision (ICMV 2021)*, **12084** (2022), 369–376. DOI: 10.1117/12.2623139.

[24] Ham D.S., Lee D.R., Jung I.S., Oh I.S., “Construction of printed Hangul character database PHD08”, *J. Korea Contents Assoc.*, **8** (2008), 33–40.

[25] Lake B., Salakhutdinov R., Gross J., Tenenbaum J., “One shot learning of simple visual concepts”, *Proceedings of the Annual Meeting of the Cognitive Science Society*, **33** (2011).

[26] Lake B.M., Salakhutdinov R., Tenenbaum J.B., "Human-level concept learning through probabilistic program induction", *Science*, **350** (2015), 1332–1338. DOI: 10.1126/science.aab3050.

[27] Bergstra J., Desjardins G., Lamblin P., Bengio Y., "Quadratic polynomials learn better image features", *Tech. Rep.*, 2009.

[28] Lin G., Shen W., "Research on convolutional neural network based on improved Relu piecewise activation function", *Procedia Comput. Sci.*, **131** (2018), 977–984. DOI: 10.1016/j.procs.2018.04.239.

Статья поступила 9 февраля 2026 г.

Addressing the Training Challenges of Siamese Networks for Optical Character Recognition

A. K. Mokin

The development of deep learning models for classification tasks poses particular challenges when dealing with a large number of classes under conditions of limited data and computational resources. Metric learning offers a promising approach to solving this problem, although its effectiveness is often limited by inherent shortcomings of standard loss functions, such as contrastive and triplet losses, as well as suboptimal training sample selection strategies. This paper presents solutions to these issues: a novel autoprobabilistic mining method for example selection and an improved metric loss function. The proposed autoprobabilistic mining method aids in selecting the most informative example pairs for training Siamese neural networks. In combination with a previously developed autoclustering method, this approach enhances training efficiency by maximizing data utility while minimizing computational costs. Additionally, a new triplet-based metric loss function is introduced, which accounts for cluster-specific characteristics and is designed to overcome specific drawbacks of traditional contrastive and triplet loss functions, thereby improving the process of feature embedding formation. The effectiveness of the proposed methods was validated through experiments on optical character recognition using the PHD08 (Korean alphabet) and Omniglot datasets. For the full Korean alphabet in the PHD08 dataset, the novel loss function with random mining achieved a classification accuracy of 82.6%, establishing a new benchmark. Using a reduced alphabet, a baseline of 88.6% was set on PHD08. The application of the auto-probabilistic mining method alone improved accuracy to 90.6% (+2.0%), and its combination with auto-clustering further increased it to 92.3% (+3.7%). On the Omniglot dataset, the proposed mining method attained 92.32%, which rose to

93.17% when coupled with auto-clustering. The results demonstrate that the proposed loss function and mining strategy offer a robust and effective solution for complex pattern recognition tasks, especially in scenarios characterized by a high number of classes and resource limitations.

Keywords: deep metric learning, optical character recognition, siamese neural networks, pattern recognition.

References

[1] Khaustov P.A., "Algorithms for handwritten character recognition based on constructing structural models", *Comput. Opt.*, **41** (2017), 67–78. DOI: 10.18287/2412-6179-2017-41-1-67-78.

[2] Koch G., Zemel R., Salakhutdinov R., "Siamese neural networks for one-shot image recognition", *Proceedings of the 32nd International Conference on Machine Learning*, 2015, 7–9 July 2015; Volume 2.

[3] Hoffer E., Ailon N., "Deep metric learning using triplet network", *Proceedings of the International Workshop on Similarity-Based Pattern Recognition*, 2015, 84–92.

[4] Oh Song H., Xiang Y., Jegelka S., Savarese S., "Deep metric learning via lifted structured feature embedding", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, 4004–4012.

[5] Bromley J., Guyon I., LeCun Y., Säckinger E., Shah R., "Signature verification using a "siamese" time delay neural network", *Adv. Neural Inf. Process. Syst.*, **6** (1993), 737–744.

[6] Tafti A.P., Baghaie A., Assefi M., Arabnia H.R., Yu Z., Peissig P., "OCR as a service: An experimental evaluation of Google Docs OCR, Tesseract, ABBYY FineReader, and Transym", *Proceedings of the 12th International Symposium on Visual Computing*, 2016, 735–746.

[7] Chopra S., Hadsell R., LeCun Y., "Learning a similarity metric discriminatively, with application to face verification", *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, **1** (2005), 539–546. DOI: 10.1109/CVPR.2005.202.

[8] Hadsell R., Chopra S., LeCun Y., "Dimensionality reduction by learning an invariant mapping", *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, **2** (2006), 1735–1742. DOI: 10.1109/CVPR.2006.100.

[9] Ilyuhin S.A., Sheshkus A.V., Arlazarov V.L., "Recognition of images of Korean characters using embedded networks", *Proceedings of the Twelfth International Conference on Machine Vision (ICMV 2019)*, **11433** (2020), 1143311.

[10] Kondrashev I.V., Sheshkus A.V., Arlazarov V.V., "Distance-based online pairs generation method for metric networks training", *Proceedings of the Thirteenth International Conference on Machine Vision*, **11605** (2021), 1160508.

[11] Wang X., Hua Y., Kodirov E., Hu G., Robertson N.M., "Deep metric learning by online soft mining and class-aware attention", *Proceedings of the AAAI Conference on Artificial Intelligence*, **33** (2019), 5361–5368. DOI: 10.1609/aaai.v33i01.33015361.

[12] Wang F., Liu H., "Understanding the behaviour of contrastive loss", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 2495–2504.

[13] Schroff F., Kalenichenko D., Philbin J., "Facenet: A unified embedding for face recognition and clustering", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, 815–823.

[14] Yuan Y., Chen W., Yang Y., Wang Z., "In defense of the triplet loss again: Learning robust person re-identification with fast approximated triplet loss and label distillation", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, 354–355.

[15] Dalwadi Bijal N., Shah Vatsal, "Deep Embedding Learning for Printed Word Spotting using Triplet CNNs and Transfer Learning", *Proceedings of the IEEE 5th International Conference on Soft Computing for Security Applications (ICSCSA)*, **73** (2025), 384–389.

[16] Chen W., Chen X., Zhang J., Huang K., "Beyond triplet loss: A deep quadruplet network for person re-identification", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, 403–412.

[17] Franken M., van Gemert J.C., "Automatic Egyptian hieroglyph recognition by retrieving images as texts", *Proceedings of the 21st ACM international conference on Multimedia*, 2013, 765–768. DOI: 10.1145/2502081.2502199.

[18] Kim Y.g., Cha E.y., "Learning of Large-Scale Korean Character Data through the Convolutional Neural Network", *Proceedings of the Korean Institute of Information and Commucation Sciences Conference*, 2016, 97–100.

[19] Bell S., Bala K., "Learning visual similarity for product design with convolutional neural networks", *ACM Trans. Graph. (TOG)*, **34** (2015), 1–10. DOI: 10.1145/2766959.

[20] Simo-Serra E., Trulls E., Ferraz L., Kokkinos I., Fua P., Moreno-Noguer F., "Discriminative learning of deep convolutional feature point descriptors", *Proceedings of the IEEE International Conference on Computer Vision*, 2015, 118–126.

[21] Xu A., Hsieh J.Y., Vundurthy B., Cohen E., Choset H., Li L., "Mathematical Justification of Hard Negative Mining via Isometric Approximation Theorem", *arXiv*, 2022.

[22] Xuan H., Stylianou A., Pless R., "Improved embeddings with easy positive triplet mining", *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, 2474–2482.

[23] Mokin A.K., Gayer A.V., Sheshkus A.V., Arlazarov V.L., "Auto-clustering pairs generation method for Siamese neural networks training", *Proceedings of the Fourteenth International Conference on Machine Vision (ICMV 2021)*, **12084** (2022), 369–376. DOI: 10.1117/12.2623139.

[24] Ham D.S., Lee D.R., Jung I.S., Oh I.S., "Construction of printed Hangul character database PHD08", *J. Korea Contents Assoc.*, **8** (2008), 33–40.

[25] Lake B., Salakhutdinov R., Gross J., Tenenbaum J., "One shot learning of simple visual concepts", *Proceedings of the Annual Meeting of the Cognitive Science Society*, **33** (2011).

[26] Lake B.M., Salakhutdinov R., Tenenbaum J.B., "Human-level concept learning through probabilistic program induction", *Science*, **350** (2015), 1332–1338. DOI: 10.1126/science.aab3050.

[27] Bergstra J., Desjardins G., Lamblin P., Bengio Y., "Quadratic polynomials learn better image features", *Tech. Rep.*, 2009.

[28] Lin G., Shen W., "Research on convolutional neural network based on improved Relu piecewise activation function", *Procedia Comput. Sci.*, **131** (2018), 977–984. DOI: 10.1016/j.procs.2018.04.239.

Received on February 9, 2026