

- СММ не обладает различающей способностью, то есть алгоритм обучения, только максимизирует отклик каждого изображения на свою модель, но не минимизирует отклик на другие модели [4];
- Не применяют в коммерческих целях.

4) Метод главных компонент или principal component analysis (PCA) [5]:

Основное преимущество - значительное уменьшение размерности пространства признаков таким образом, чтобы оно как можно лучше описывало «типичные» образы, принадлежащие множеству лиц. Полученный один раз на обучающей выборке изображений лиц набор собственных векторов используется для кодирования всех остальных изображений лиц, которые представляются взвешенной комбинацией этих собственных векторов. Используя ограниченное количество собственных векторов, можно получить сжатую аппроксимацию входному изображению лица, которую затем можно хранить в базе данных в виде вектора коэффициентов, служащего одновременно ключом поиска в базе данных лиц [6].

Основной недостаток - является очень чувствительным к входным данным. Изменение освещенности, угол поворота головы и эмоциональное выражение лица сильно меняют результат распознавания. Это обуславливается тем, что задачей алгоритма является наилучшая аппроксимация входных данных, а не их распределение по классам.

Для большинства современных систем автоматического распознавания лиц основной задачей является задача сравнения данного изображения лица с набором изображений лиц из базы данных. Характеристики систем автоматического распознавания лиц в этом случае, оцениваются путем определения вероятностей ошибочного отказа в распознавании (для изображения лица, присутствующего в базе, принимается решение как о неопознанном лице) и ошибочного распознавания.

Выбор метода идентификации лиц для создания практических систем автоматического распознавания лиц, должен основываться на ограничениях конкретной системы, и определяется непосредственно на этапе проектирования.

Учитывая вышеизложенное, для повышения качества результата, перспективным может являться создание гибридных методов, использующих преимущества и нивелирующих недостатки рассмотренных выше различных частных подходов.

Список использованных источников:

1. L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg, "Face Recognition by Elastic Bunch Graph Matching," Technical Report IR-INI 96-08, Institut für Neuroinformatik, Ruhr-Universität Bochum, D-44780 Bochum, Germany, 1996.
2. Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, Lior Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification" 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014.
3. Гулятьева, Т.А. Скрытые марковские модели с одномерной топологией в задаче распознавания лиц / Т.А. Гулятьева, А.А. Попов; НГТУ. — 2006.
4. Samaria F. S., Face Recognition Using Hidden Markov Models // PhD Thesis, University of Cambridge, 1994.
5. Шерстобитов Александр Иванович, Федосов Валентин Петрович, Приходченко Владислав Александрович, Тимофеев Митрий Витальевич Распознавание лиц на групповых фотографиях с использованием алгоритмов сегментации // Известия ЮФУ. Технические науки. 2013.
6. P.N. Belhumeur ; J.P. Hespanha ; D.J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection", IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997.

АНАЛИЗ МОДИФИКАЦИЙ ГЕНЕТИЧЕСКОГО АЛГОРИТМА

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Игнатъев Н.С.

Ярмолик В.Н. – д.т.н., профессор

Генетические алгоритмы (ГА) - изначально возникли в результате наблюдений и попыток копирования процессов, происходящих в природе, в частности естественного отбора живых организмов. С точки зрения программных и информационных технологий, ГА используются как стратегия решения задач оптимизации, основанная на имитировании биологической эволюции. Программы, построенные на базе таких алгоритмов, дают довольно хорошие результаты и используются в различных сферах науки и инженерии.

Задача классической модели генетического алгоритма формализуется таким образом, чтобы её решение могло быть закодировано в виде генотипа (вектора генов фиксированной длины). Некоторым образом создаётся множество генотипов начальной популяции, которые будут оценены с использованием фитнес функции, в результате чего с каждым генотипом ассоциируется определённое значение приспособленности, которое определяет насколько хорошо фенотип, им описываемый, решает поставленную задачу. С помощью пропорционального отбора формируется промежуточный массив, из которого случайным образом выбираются два родителя. Далее производится одноточечный кроссинговер, и созданные два потомка мутируют (одноточечная мутация) с заданной вероятностью. Мутировавшие потомки занимают места своих родителей. Процесс продолжается до тех пор, пока не будет достигнут критерий окончания алгоритма.

В модификации СНС для кроссинговера, выбирается случайная пара родителей, но не допускается, чтобы между ними было малое хеммингово расстояние. При скрещивании производится однородный кроссинговер: к потомку переходит по половине битов каждого родителя. Для нового поколения выбираются

Н лучших различных особей среди родителей и детей. При этом дублирование генотипов не допускается. В модели СНС размер популяции относительно мал — около 50 особей. Это оправдывает использование однородного кроссинговера и позволяет алгоритму сойтись к решению. СНС противопоставляет агрессивный отбор агрессивному кроссинговеру, однако, малый размер популяции быстро приводит ее к состоянию, когда создаются только более или менее одинаковые генотипы. В таком случае СНС применяет cataclysmic mutation: все генотипы, кроме самого приспособленного, подвергаются сильной мутации (изменяется около трети битов). Так алгоритм перезапускается и далее продолжает работу, применяя только кроссинговер. Таким образом, алгоритм СНС довольно быстро сходится из-за того, что в нем нет мутаций, следующих за оператором кроссинговера, используются популяции небольшого размера, и отбор особей в следующее поколение ведется и между родительскими особями, и между их потомками.

Идея гибридных алгоритмов заключается в сочетании генетического алгоритма с некоторым другим классическим методом поиска, подходящим в данной задаче. В каждом поколении все сгенерированные потомки оптимизируются выбранным методом и затем заносятся в новую популяцию. Тем самым получается, что каждая особь в популяции достигает локального оптимума, вблизи которого она находится. Далее, производятся обычные для ГА действия: отбор родительских пар, кроссинговер и мутации. На практике гибридные алгоритмы оказываются очень удачными. Это связано с тем, что вероятность попадания одной из особей в область глобального максимума обычно велика. После оптимизации такая особь будет являться решением задачи. Известно, что генетический алгоритм способен быстро найти во всей области поиска хорошее решение, но он может испытывать трудности в получении из них наилучших. Обычный оптимизационный метод может быстро достичь локального максимума, но не может найти глобальный. Сочетание двух алгоритмов позволяет использовать преимущества обоих.

В модели генитор используется специфичный способ отбора. Вначале, как и полагается, популяция инициализируется, и ее особи оцениваются. Затем выбираются случайным образом две особи, скрещиваются, причем, получается только один потомок, который оценивается и занимает место менее приспособленной особи в популяции, а не одного из родителей. После этого снова случайным образом выбираются две особи, и их потомок занимает место родительской особи с самой низкой приспособленностью. Таким образом, на каждом шаге в популяции обновляется лишь одна особь. Процесс продолжается до тех пор, пока пригодности хромосом не станут одинаковыми. В данный алгоритм можно добавить мутацию потомка после его создания. Критерий окончания процесса, как и вид кроссинговера и мутации, можно выбрать разными способами.

Для повышения эффективности работы ГА создано множество модификаций. Они связаны с применением иных методов селекции, с модификацией генетических операторов, с преобразованием функции приспособленности, а также, с различными способами кодирования параметров задачи в форме хромосом. Описанные выше примеры не исчерпывают многообразие областей и задач применения ГА. Эта область сейчас является перспективной и требует ещё долгого изучения.

Список использованных источников:

1. Эволюционные вычисления [Электронный ресурс]. – Режим доступа к источнику: <https://www.intuit.ru/studies/courses/14227/1284/info>
2. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы // Под ред. В.М. Курейчика. — 2-е изд., испр. и доп. — Москва: ФИЗМАТЛИТ, 2006. — 320 с.

РЕАЛИЗАЦИЯ ГОЛОВОЛОМКИ «КУБИК РУБИКА»

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Камеко В.В.

Данилова Г. В. – м. т. н., ассистент

В настоящий момент, в связи со спросом на головоломки различных видов, возникла потребность в их симуляторах, позволяющих собирать головоломки на компьютерах. В качестве головоломки для реализации была выбрана самая известная из них – «Кубик Рубика». В программе будут реализованы все функции данной головоломки на языке программирования Delphi.

Известный факт, что головоломки положительно влияют на развитие пространственного и логического мышления, а также, являются отличным антистрессом. Одна из таких головоломок, «Кубик Рубика», не только стала самой знаменитой головоломкой и собрала целое движение «спидкуберов», но и поставила довольно интересную задачу математикам и программистам всего мира. В настоящее время существует достаточно мало приложенных эмулирующих эту замечательную игрушку, как и не существует единого формата хранения состояния «Кубика».

«Кубик Рубика» – механическая головоломка, изобретённая в 1974, представляющая из себя куб 3x3x3 с окрашенными в разные цвета гранями. Задача игрока – собрать «кубик» в первоначальное состояние из перемешанного.