ГОЛОСОВАЯ ИДЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЯ В СИСТЕМАХ КОНТРОЛЯ ДОСТУПА

П. А. Меньшаков

Учреждение образования «Гомельский государственный технический университет имени П. О. Сухого, Беларусь

Научный руководитель И. А. Мурашко

Принцип голосовой идентификации

Сам процесс голосовой идентификации не требователен к ресурсам и состоит из двух этапов. Сначала необходимо получить голосовой отпечаток пользователя и преобразовать к виду, в котором его можно будет сравнить с другими. Вторым шагом является сравнение голосовых отпечатков при помощи обученной нейронной сети. Для реализации процесса преобразования необходимо произвести определенный порядок действий.

При помощи микрофона получается запись голоса идентифицируемого и отправляется на ЭВМ. Наиболее оптимальным является получение WAV файла ввиду простоты работы с ним.

Полученную запись голоса необходимо разделить на кадры. Разделение на кадры представлено на рис. 1. Данное действие необходимо для более простой работы с записанной звуковой дорожкой.

Далее все вычисления будут производиться с каждым кадром в отдельности.

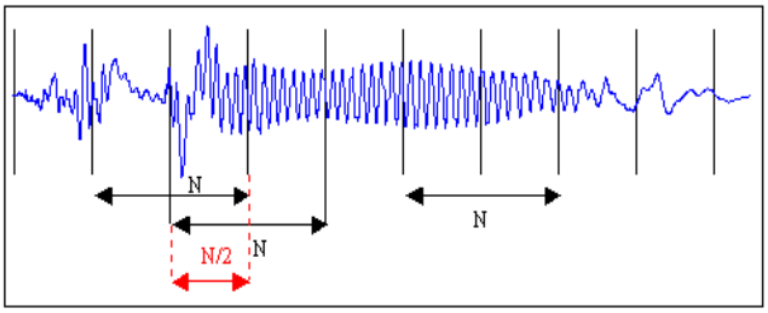

Рис. 1. График звуковой волны

Следующим этапом является устранение нежелательных эффектов и шумов. Это необходимо для того, чтобы записи, полученные в разное время, соответствовали друг другу независимо от сторонних факторов. Существует множество способов, при помощи которых можно уменьшить шумовые эффекты. Нами использовалось умножение каждого кадра на особую весомую функцию «Окно Хемминга»:

$$\omega(n) = 0{,}53836 - 046164 \cdot \cos\left(\frac{2\pi n}{N-1}\right), \quad (1)$$

где n – порядковый номер элемента в кадре, для которого вычисляется новое значение амплитуды; N – длина кадра (количество значений сигнала, измеренных за период).

Полученные кадры преобразуются в их частотную характеристику при помощи прогонки через быстрое преобразование Фурье:

$$X_k = \sum_{i=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn}, \quad (2)$$

где N – длина кадра (количество значений сигнала, измеренных за период); xn – амплитуда n-го сигнала; Xk – N-комплексных амплитуд синусоидальных сигналов, слагающих исходный сигнал.

На сегодняшний день наиболее успешными являются системы распознавания голоса, использующие знания об устройстве слухового аппарата. Они базируются на том, что ухо интерпретирует звуки нелинейно, а в логарифмическом масштабе. Ввиду данных особенностей необходимо привести частотную характеристику каждого кадра к «мелам». Зависимость представлена на рис. 2.

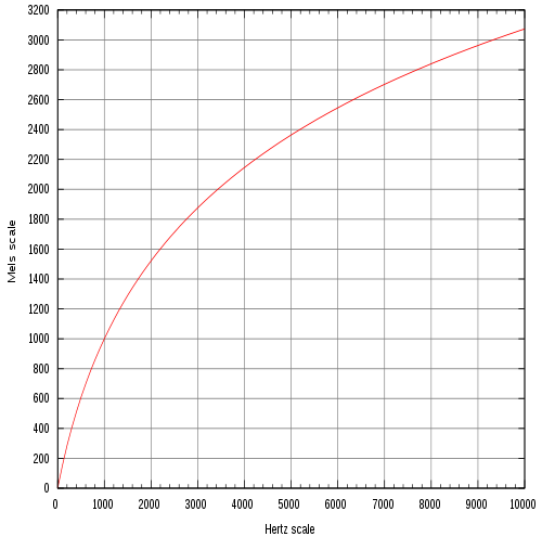

Рис. 2. График зависимости высоты звука (в мелах) от его частоты

Для перехода к «мел» характеристике используется следующая зависимость:

$$m = 1127 \log_e \left(1 + \frac{f}{700}\right), \quad (3)$$

где m – частота в мелах; f – частота в герцах.

Это последнее действие, необходимое для последующего преобразования в вектор характеристики, который впоследствии сравнивается с базой голосовых записей. Вектор будет состоять из мел-кепстральных коэффициентов, получить которые можно по следующей формуле:

$$c_n = \sum_{k=1}^{K} (\log S_k) \left[n \left(k - \frac{1}{2} \right) \frac{\pi}{K} \right], \quad (4)$$

где c_n – мел-кепстральный коэффициент под номером n; S_k – амплитуда k-го значения в кадре в мелах; K – наперед заданное количество мел-кепстральных коэффициентов $n \in [1, K]$.

Полученный вектор характеристик добавляется в базу данных для последующего сравнения с ним.

Однако более оптимальным вариантом является использование нескольких записей одного и того же голоса. Заранее определенное количество образцов голоса можно использовать для обучения нейронной сети.

В работе использовалось обучение безучителя, так как оно является намного более правдоподобной моделью обучения в биологической системе. Развитая Кохоненом и многими другими, она не нуждается в целевом векторе для выходов и, следовательно, не требует сравнения с предопределенными идеальными ответами, а обучающее множество состоит лишь из входных векторов. Обучающий алгоритм подстраивает веса сети так, чтобы получались согласованные выходные векторы, т. е. чтобы предъявление достаточно близких входных векторов давало одинаковые выходы. Процесс обучения, следовательно, выделяет статистические свойства обучающего множества и группирует сходные векторы в классы. Предъявление на вход вектора из данного класса даст определенный выходной вектор [3].

Распространение сигнала в такой сети происходит следующим образом: входной вектор нормируется на 1.0 и подается на вход, который распределяет его дальше через матрицу весов *W*. Каждый нейрон в слое Кохонена вычисляет сумму на своем входе и в зависимости от состояния окружающих нейронов этого слоя становится активным или неактивным (1.0 и 0.0). Нейроны этого слоя функционируют по принципу конкуренции, т. е. в результате определенного количества итераций активным остается один нейрон или небольшая группа. Этот механизм называется латеральным. Так как отработка этого механизма требует значительных вычислительных ресурсов, в моей модели он заменен нахождением нейрона с максимальной активностью и присвоением ему активности 1.0, а всем остальным нейронам 0.0. Таким образом, срабатывает нейрон, для которого вектор входа ближе всего к вектору весов связей.

Если сеть находится в режиме обучения, то для выигравшего нейрона происходит коррекция весов матрицы связи по формуле

$$w_{\text{н}} = w_{\text{м}} + \alpha(x - w_{\text{м}}), \tag{5}$$

где $w_{\text{н}}$ – новое значение веса; $w_{\text{м}}$ – старое значение; α – скорость обучения; x – величина входа.

Так как входной вектор x нормирован, т. е. расположен на гиперсфере единичного радиуса в пространстве весов, то при коррекции весов по этому правилу происходит поворот вектора весов в сторону входного сигнала. Постепенное уменьшение скорости поворота позволяет произвести статистическое усреднение входных векторов, на которые реагирует данный нейрон.

Однако имеется несколько проблем. Первая – выбор начальных значений весов. Так как в конце обучения вектора весов будут располагаться на единичной окружности, то в начале их также желательно нормировать на 1.0. В нашей модели вектора весов выбираются случайным образом на окружности единичного радиуса.

Вторая – если весовой вектор окажется далеко от области входных сигналов, он никогда не даст наилучшего соответствия, всегда будет иметь нулевой выход, следовательно, не будет корректироваться и окажется бесполезным. Оставшихся нейронов может не хватить для разделения входного пространства сигналов на классы. Для

решения той проблемы предлагается много алгоритмов, в работе применяется правило «работать»: если какой либо нейрон долго не находится в активном состоянии, он повышает веса связей до тех пор, пока не станет активным и не начнет подвергаться обучению. Этот метод позволяет также решить проблему тонкой классификации: если образуется группа входных сигналов, расположенныхблизко друг к другу, с этой группой ассоциируется и большое число нейронов Кохонена, которые разбивают ее на классы.

Литература

1. Bosi, M. Introduction to digital audio coding and standards / M. Bosi, R. E. Goldberg – Springer Science + Business, Media USA, 2003. – 434 p.
2. You, Y. AudioCoding: Theory and Applications / Y. You. – NY : Springer, 2010. – 349 p.
3. Загуменнов, А. П. Компьютернаяобработказвука / А. П. Загуменнов. – М. : ДМК, 1999. – 384 с.