

Московский государственный технический университет им. Баумана

Отчёт  
по лабораторной работе  
«Нейронные сети»  
по курсу «Интеллектуальные системы»

Выполнил: Миляев Н.А.,  
Группа ИУ5-81.

Москва  
2010

# 1. Аннотация

В данной работе представлена программа, решающая задачу распознавания гласных звуков с использованием однослойной нейронной сети.

## 2. Условие задачи

Программа принимает от пользователя звуковой файл в несжатом формате (WAV PCM, 8 kHz, 16 bit, mono), и на основании имеющихся данных о характеристиках звуков даёт ответ, на какой из четырёх гласных звуков (А, О, И, У) больше всего похож звук, загруженный пользователем. Данные о характеристиках звуков собираются в режиме обучения.

Программа работает в режиме обучения (пользователь загружает файл и указывает, какой это звук, программа сохраняет информацию в виде весовых коэффициентов входа нейрона), и в режиме распознавания (пользователь загружает файл, программа сообщает, на какой звук из четырёх больше всего похож тот, который был загружен пользователем).

## 3. Используемые технологии

Программа написана на языке PHP и построена на основе Web-технологий. Для её работы требуется следующее ПО:

- На сервере — ОС Linux, Web-сервер Apache, PHP 5-й версии (возможна работа с другими ОС и Web-серверами, тестирование не проводилось).
- На клиенте — любой Web-браузер (на клиентской стороне используется только HTML).

Так как программа построена на основе Web-технологий, она может быть доступной через сеть Интернет.

Программа использует алгоритм быстрого преобразования Фурье (БПФ), приведённый в статье Wikipedia ([http://ru.wikipedia.org/wiki/Быстрое\\_преобразование\\_Фурье](http://ru.wikipedia.org/wiki/Быстрое_преобразование_Фурье)). Этот алгоритм используется с целью разложения сигнала на гармоники с целью последующего анализа.

## 4. Этапы работы программы

Интерфейс пользователя программы имеет вид, представленный на рис. 1. Основные элементы интерфейса:

1. Поле выбора файла. Нажав кнопку «Обзор...», пользователь может выбрать звуковой файл для анализа. Файл должен быть в формате несжатого WAV-звука, с частотой дискретизации 8 кГц, 16 бит на отсчёт и одним каналом (моно-запись). Размер файла должен быть от 8236 до 100000 байт. В начале файла желательно обрезать тишину, так как для анализа берутся первые 4096 отсчётов (0,512 сек.) звукового файла. В случае загрузки файла неправильного формата или размера, пользователь получает сообщение об ошибке.
2. Выбор звука. Пользователь указывает, какой звук содержится в заданном файле (для обучения программы), или выбирает пункт «Надо узнать, какой это звук!» для распознавания.
3. Чувствительность. Этот параметр задаёт чувствительность рецептора. Чем больше значение этого параметра, тем больше входов нейрона перейдут в активное состояние. Число входов нейронов в активном состоянии должно быть не менее

$$N_{\text{акт.мин.}} = 3C + 20, \text{ где } C\text{-параметр чувствительности.}$$

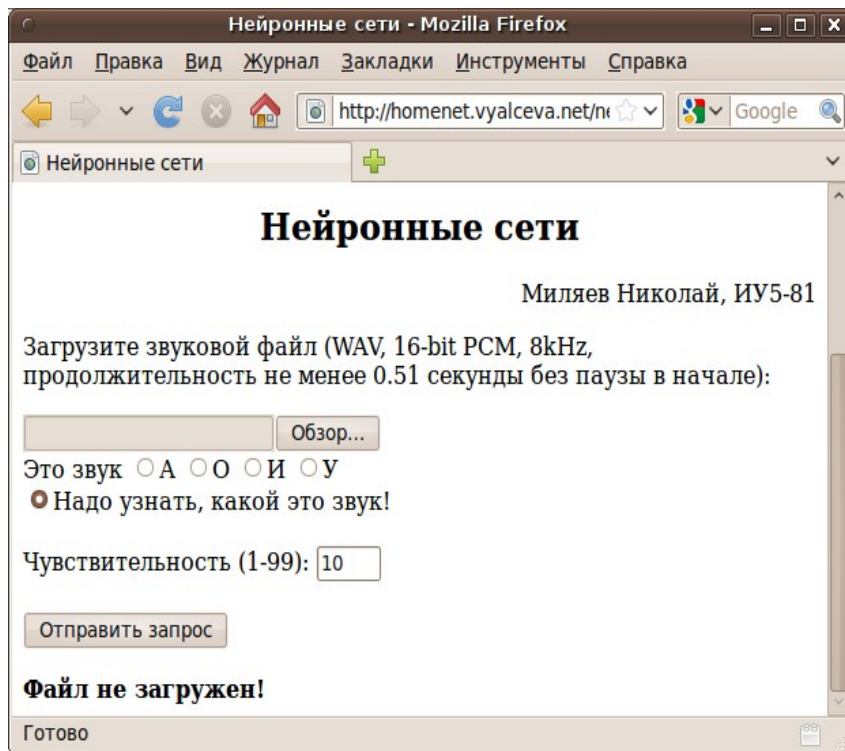


Рис. 1. Интерфейс пользователя программы.

Приняв от пользователя файл, программа проверяет его на корректность (по размеру и содержимому заголовка файла). Если файл не соответствует по размеру или по формату, программа выдаёт сообщение об ошибке, иначе она считывает 8192 байта в массив из 4096 двухбайтовых слов и подаёт на вход алгоритма БПФ. С выхода алгоритма БПФ снимается массив из 2048 вещественных чисел  $A$ , которые представляют собой амплитуду каждой гармоники сигнала:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn}, k=0, \dots, N/2-1$$

$$A_k = \sqrt{\Re X_k^2 + \Im X_k^2}$$

Затем производится поднятие верхних гармоник по формуле

$$\hat{A}_k = A_k \cdot \left(1 + \frac{k}{100}\right)$$

В реальных звуковых сигналах низкочастотные гармоники имеют большую амплитуду, чем высокочастотные, но высокочастотные содержат большую часть полезной информации (по этой причине, например, при нарушениях слуха с ограничением верхних частот воспринимаемая громкость сигнала практически не изменяется, а разборчивость речи заметно снижается).

После этого значения амплитуд нормализуются путём деления на максимальное значение приведённой амплитуды:

$$\hat{A}_{kn} = \frac{\hat{A}_k}{\hat{A}_{max}}$$

Далее производится подбор чувствительности рецептора. Для каждого  $K$ , начиная с 2, ищется число таких входов нейрона (нормализованных амплитуд), при которых

$$K \cdot \hat{A}_{kn} > 1$$

Если число таких гармоник меньше  $N_{акт.мин.} = 3C + 20$  (где  $C$  — заданное пользователем значение чувствительности), то значение  $K$  увеличивается на 1 и процесс повторяется.

После этого формируются входные сигналы нейрона по формуле сигмоида:

$$P_k = \frac{1}{1 + e^{M_1 \cdot (1 - K \cdot \hat{A}_{kn})}}$$

Параметр  $M_1$  определяет крутизну характеристики. Чем он больше, тем ближе характеристика к пороговой. Сигмоидальная характеристика используется для того, чтобы учесть не только максимальные значения гармоник, но и близкие к ним.

Слишком маленькое значение  $M_1$  приводит к размыванию разницы между значениями гармоник, и тоже нежелательно. В данном примере используется значение  $M_1=5$ .

После преобразования пользователю передаются номера активных нейронов (т. е. таких, что  $K \cdot \hat{A}_{kn} > 1$  или, что то же самое,  $P_k > 0,5$ ).

Затем производится считывание таблицы весов входов нейрона, которая хранится в текстовом файле. Файлы с таблицами весов (по одному файлу для каждого звука) хранятся в каталоге, задаваемом переменной \$dir, например, «/var/www/neuro/». Если файл не найден, значения коэффициентов полагаются равными нулю. После чего выполняется непосредственно обучение нейрона. Если  $P_k$  — значение сигнала на входе нейрона,  $S$  — значение сигнала на выходе сумматора, где

$$S = \sum_{k=0}^{N-1} c_k P_k, \text{ то значение выхода нейрона } Q \text{ вычисляется по следующей формуле:}$$

$$Q = \frac{1}{1 + e^{-S \cdot M_2}}.$$

Значение коэффициента  $M_2$  определяет крутизну выходной характеристики нейрона. Чем он больше, тем ближе характеристика к пороговой. В данной программе выбрано значение 0.2.

Новый вес входа  $c_{k,c}$  вычисляется по формуле:

$$\bar{c}_k = c_k + \frac{(1-Q)}{2} \cdot P_k; c_{k,c} = \bar{c}_k - \frac{\sum_{i=0}^{N-1} \bar{c}_i}{N},$$

таким образом, сумма весов равняется нулю. После этого новые значения весов записываются в тот же файл, из которого они были прочитаны.

Если пользователь работает с программой в режиме распознавания, программа аналогичным образом анализирует звук, читает каждую таблицу весов входов нейрона, вычисляет значение «степени уверенности»  $Q$  по приведённой выше формуле, и выбирает звук с наибольшим значением  $Q$ .

## 5. Результаты проведённых экспериментов

В ходе отладки программы была проведена запись нескольких звуков из числа распознаваемых программой, которые затем загружались для обучения и распознавания. Сначала программа обучалась на всех четырёх звуках из набора звуков для обучения, потом делалась попытка провести распознавание звуков из набора для распознавания. Звуки для обучения и распознавания записывались в разное время. Результаты экспериментов приведены в таблице. Значение чувствительности равнялось 10.

Правильная последовательность звуков: А О У И О А О, последний звук представляет собой не очень чёткое О.

Число циклов обучения и распознавания	Результаты по номерам звуков							Число правильно распознанных звуков
	1	2	3	4	5	6	7	
1	О	О	И	И	О	А	А	4
2	О	У	И	И	О	А	А	3
3	О	У	И	И	О	А	А	3

Видно, что обучение проходит не очень эффективно. Попробуем увеличить значение чувствительности до 30 (число активных входов нейронов должно возрасти):

Число циклов обучения и распознавания	Результаты по номерам звуков							Число правильно распознанных звуков
	1	2	3	4	5	6	7	
1	А	У	У	И	О	А	А	5
2	А	У	У	И	О	А	А	5
3	А	У	У	И	О	А	А	5

Качество обучения возрастает.

Увеличим чувствительность до максимального значения (99):

Число циклов обучения и распознавания	Результаты по номерам звуков							Число правильно распознанных звуков
	1	2	3	4	5	6	7	
1	А	У	У	И	У	А	А	4
2	А	У	У	И	У	А	А	4
3	А	У	У	И	У	А	А	4

Качество распознавания снижается за счёт того, что звуки О и У сильнее перекрываются по активным нейронам.

## 6. Выводы

1. Программа хуже всего распознаёт звук О, путая его с У и А. Это связано с тем, что звук О является по своему произношению промежуточным между А и У.
2. Как слишком малое, так и слишком большое число активных входов нейрона ухудшает эффективность распознавания.