

РАЗРАБОТКА ОБЪЕКТНО-ОРИЕНТИРОВАННОЙ МОДЕЛИ ЭМУЛЯТОРА НЕЙРОННОЙ СЕТИ С ВРЕМЕННОЙ ЗАДЕРЖКОЙ СИГНАЛА

Гусак Е.А., Бондаренко И.Ю.

Донецкий национальный технический университет

Кафедра прикладной математики и информатики

E-mail: kristallina@mail.ru

Аннотация:

Гусак Е.А., Бондаренко И.Ю. Разработка объектно-ориентированной модели эмулятора нейронной сети с временной задержкой сигнала. Предложен нейроалгоритм распознавания устной речи, основанный на использовании нейронной сети с временной задержкой сигнала. Разработана объектно-ориентированная UML-модель (диаграмма классов) для построения программного эмулятора данной нейросети.

Общая постановка проблемы

В последние годы значительный научный интерес представляет проблема распознавания речевых сигналов. Создание системы автоматического распознавания речи позволит решить следующие проблемы: во-первых, увеличение информационной нагрузки на пользователя и наличие в широком распространении единственного канала общения с техникой; во-вторых, создание каких-либо альтернативных систем управления техникой для людей с ограниченными возможностями; в-третьих, построение на основе распознавания голоса защитных систем с голосовым доступом.

Сравнительный анализ методов распознавания речи показал, что одним из наиболее эффективных для решения подобной задачи является нейросетевой подход. Для того, чтобы иметь возможность представлять такой класс задач, нейронная сеть должна обладать следующими свойствами:

- наличие нескольких слоев для построения сложных разделяющих поверхностей;
- достаточное количество взаимосвязей между слоями для того, чтобы достичь желаемой обучаемости сети и в то же время избежать появления слишком большого количества настраиваемых параметров;
- возможность представлять зависимость между сигналами, протяженными во времени;
- эффективная процедура обучения.[1]

Наиболее подходящей для решения подобной задачи и обладающей всеми вышеперечисленными свойствами является нейронная сеть с временной задержкой сигнала – некоторая модификация классического многослойного персептрона, обеспечивающая более быстрое обучение сети и лучшее представление зависимости сигналов во времени.

Математическая модель нейронной сети

Определим входной сигнал для данной нейросети. Он будет представлять собой спектрально-временной образ — матрицу спектральных коэффициентов входного речевого сигнала, взятых с заданным интервалом. Эти коэффициенты поступают на вход нейронов входного слоя и затем подвергаются скользящему оконному анализу. Вычисляется взвешенная сумма выходов нейронов входного слоя с учетом рецептивных полей нейронов и подается на вход соответствующим нейронам первого скрытого слоя. Благодаря наличию рецептивных полей нейронная сеть становится неполносвязной, соответственно, количество весов в сети значительно уменьшается. Это позволяет настроить обучение нейросети так, чтобы снизить вероятность запоминания ею нежелательных признаков вроде шума. [2] Для извлечения структурных признаков из спектрально-временного образа, являющегося

двумерным сигналом, нейроны входного и всех скрытых слоев также организованы в виде двумерных матриц.

В полносвязном многослойном персептроне входной сигнал нейрона первого скрытого слоя представлял бы собой такую взвешенную сумму:

$$z_{j,l} = \sum_{n=1}^N \sum_{i=1}^I w_{n,i,j,l} * f_{n,i} \quad (1)$$

где $z_{j,l}$ – номер нейрона 1-го скрытого слоя,
 $w_{n,i,j,l}$ – вес связи между нейроном (n,i) входного слоя и нейроном (j,l) первого скрытого слоя,

N – количество нейронов во входном слое по частоте,

I – количество нейронов во входном слое по времени.

С учетом рецептивных полей размера g вышеуказанная формула изменяется так:

$$z_{j,l} = \sum_{n=1}^N \sum_{i=1}^{I-r+1} w_{n,i,j,l} * f_{n,i} \quad (2)$$

где $z_{j,l}$ – номер нейрона 1-го скрытого слоя,
 $w_{n,i,j,l}$ – вес связи между нейроном (n,i) входного слоя и нейроном (j,l) первого скрытого слоя,

N – количество нейронов во входном слое по частоте,

I – количество нейронов во входном слое по времени,

r – размер рецептивного поля для нейрона (j,l).

Функция активации чаще всего используется нелинейная, например, сигмоида:

$$y = F(G) = \frac{1}{1 + e^{-G}} \quad (3)$$

Сигмоида чаще всего выбирается в качестве функции активации из-за своих удобных математических свойств, таких, как: наличие простой производной и обеспечение нелинейного сжатия входного сигнала ($g \in (-\infty; +\infty)$, а $y \in (1,0)$). [3]

На втором скрытом слое нейроны первого скрытого слоя просматривает большее скользящее окно и выходы формируются так же, как и для первого скрытого слоя.

Выходной слой представляет собой вектор, в котором каждый нейрон соответствует одному распознаваемому классу. Каждый элемент выходного слоя получает входные сигналы из соответствующей ему по номеру строки, и выходной сигнал нейросети вычисляется по формуле:

$$y_m = \sum_{v=1}^V f_{m,v}^2 \quad (4)$$

где m – номер нейрона выходного слоя/номер нейрона второго скрытого слоя по времени, $m=1..M$;

v – номер нейрона второго скрытого слоя по времени, $v=1..V$.

Выход нейросети формируется нахождением максимума из всех выходов в последнем слое. [2]

Для обучения нейросети используется алгоритм обратного распространения ошибки, модифицированный для учета рецептивных полей нейрона. В частности, введена функция «соседства», которая помогает определить, есть ли связь между нейроном (n,i) и (j,k). Тогда, к примеру, вычисление корректирующей величины Δ для изменения весов при обратном проходе по сети будет производиться по следующей формуле:

$$\Delta = \alpha * \delta_{j,l} * y_{n,i}^{k-1} * h(l,i,R^{k-1}) \quad (5)$$

где $h(i, R^{k-1})$ – функция "соседства",

$$h(i, R^{k-1}) = \begin{cases} 1, & i-1 < R_{k-1} \\ 0, & \text{иначе} \end{cases} \quad (6)$$

Благодаря вышеуказанной функции достигается неполносвязность сети и обеспечивается ее обучаемость с учетом неполносвязности.

Объектно-ориентированная модель нейросети

Для более удобного представления структуры проектируемой системы используется объектно-ориентированная модель. Такой подход к описанию системы предполагает выделение различных классов объектов классов, играющих важную роль в формировании и функционировании системы. [4]

Для нейросети с временной задержкой сигнала можно выделить такие классы, как:

- нейрон;
- слой:
 - входной;
 - выходной;
 - скрытый;
- нейросеть;
- алгоритм обучения;
- обучающее множество.

Атрибуты классов и операции, выполняемые ими, представлены в Таблицах 1-8, структура объектно ориентированной модели представлена на рис. 1.

Таблица 1. Класс «Нейрон»

Класс	Атрибут	Операция
Neuron	input: float - входной сигнал нейрона	create() - создание нейрона
	output: float - выходной сигнал нейрона	delete() - удаление нейрона
	activation: string - функция активации нейрона	create_output() - вычисление выходного сигнала нейрона
	weight_previous: array [] [] of float - матрица весов связей нейрона с предыдущим слоем	SetWeights() - задать веса связей
	weight_next: array [] [] of float - матрица весов связей нейрона со следующим слоем	GetWeights() - получить веса связей

Таблица 2. Класс «Слой»

Класс	Атрибут	Операция
Layer	Kol_neurons_freq: int - количество нейронов в слое по частоте	create() - создать слой
	Kol_neurons_time: int - количество нейронов в слое по времени	delete() - удалить слой
	receptive_field: int - рецептивное поле нейрона следующего слоя (принимается одинаковым для всех нейронов следующего слоя)	connect_prev() - соединить слой с предыдущим connect_next() - соединить слой с предыдущим

Таблица 3. Класс «Входной слой»

Класс	Атрибут	Операция
Input_layer	Kol_neurons_freq: int - количество нейронов в слое по частоте	create() - создать слой
	Kol_neurons_time: int - количество нейронов в слое по времени	delete() - удалить слой connect_next() - соединить слой с предыдущим

Таблица 4. Класс «Скрытый слой»

Класс	Атрибут	Операция
Hidden_layer	Kol_neurons_freq: int - количество нейронов в слое по частоте	create() - создать слой
	Kol_neurons_time: int - количество нейронов в слое по времени	delete() - удалить слой
	receptive_field: int - рецептивное поле нейрона следующего слоя (принимается одинаковым для всех нейронов следующего слоя)	connect_prev() - соединить слой с предыдущим connect_next() - соединить слой с предыдущим

Таблица 5. Класс «Выходной слой»

Класс	Атрибут	Операция
Output_layer	Kol_neurons_freq: int - количество нейронов в слое по частоте	create() - создать слой
	Neurons_activation - функция активации нейронов	delete() - удалить слой connect_prev() - соединить слой с предыдущим

Таблица 6. Класс «Нейросеть»

Класс	Атрибут	Операция
Neurnet	Kol_layers: int - количество нейронов в слое по частоте	create() - создать нейросеть
		delete() - удалить нейросеть
		edit() - изменить структуру сети
		training() - обучить нейросеть
		save_results() - сохранить результаты распознавания в файл

Таблица 7. Класс «Алгоритм обучения»

Класс	Атрибут	Операция
Teaching_algorithm	Weight_old: array [] [] [] offloat - матрица необновленных весов связей нейросети	get_weights_old() - получить текущие значения весов
	Weight_new: array [] [] [] offloat - матрица новых весов связей нейросети	teach() - выполнить обучение
	K_teach_speed: float - коэффициент скорости обучения	correct_weights() - корректировать веса
		SetTeachSpeed() - задать коэффициент скорости обучения

Таблица 8. Класс «Обучающее множество»

Класс	Атрибут	Операция
Teaching_array	Teaching_input: array [] [] offloat - матрица входных коэффициентов обучающего множества	SetTeachingArray() - задать обучающее множество
	Teaching_output: array [] offloat - вектор выходных сигналов обучающего множества	GetTeachingArray() - получить обучающее множество
	Verification_input: array [] [] offloat - матрица входных коэффициентов тестового множества	SetVerificationArray() - задать тестовое множество
	Verification_output: array [] offloat - вектор выходных сигналов тестового множества	GetVerificationArray() - получить тестовое множество

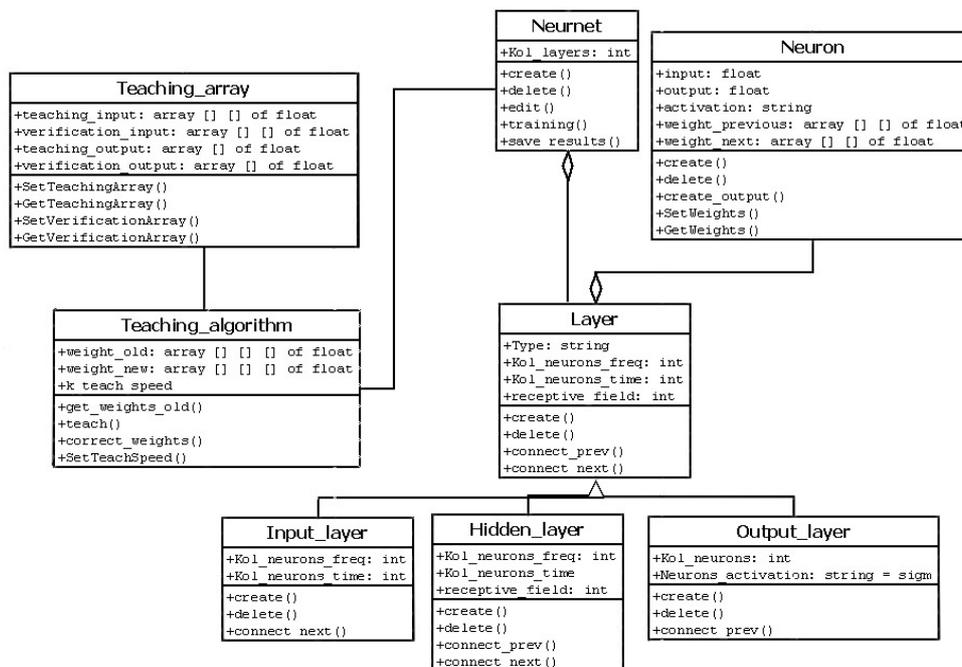


Рис.1 – Диаграмма классов.

Выводы

На основе проведенного анализа существующих подходов к решению задачи распознавания устной речи в качестве одного из самых перспективных (по критериям точности и быстродействия) был выбран нейросетевой подход. Определено, что, поскольку речевой сигнал характеризуется существенными нелинейными искажениями во временной области, то для его распознавания наиболее пригодной является специализированная нейросетевая архитектура — нейронная сеть с временной задержкой сигнала.

Описан нейроалгоритм распознавания устной речи, основанный на применении нейронной сети с временной задержкой для классификацию последовательности фрагментов речевого сигнала.

Проведён объектно-ориентированный анализ предметной области (самой нейросети, обучающего множества, алгоритма обучения и т. п.), на основе результатов которого создана UML-модель программной системы, эмулирующей работу нейронной сети с временной задержкой на персональном компьютере.

В дальнейшем предполагается применение разработанного программного эмулятора для проведения экспериментов по распознаванию фонем и слов устной речи.

Список использованной литературы.

1. Гусак Е.А., Бондаренко И.Ю. Применение специализированной нейросетевой архитектуры TDNN для распознавания речевых сигналов// Сб. тр. междунар. научно-техн. конференции «Информатика и компьютерные технологии 2010». – Донецк: ДонНТУ. – 2010.
2. A. Waibel, T. Hanazawa, G. Hinton, K.Shikano, K.J. Lang. Phoneme Recognition Using Time-Delay Neural Networks – IEEE Transaction on acoustic, speech, and signal processing. Vol. 37 No. 3 March 1989 – p. 328-339.
3. Руденко О.Г., Бодянский С.В. Штучні нейронні мережі. – Харків: ТОВ «Компанія СМІТ», 2006. – 404 с.
4. Гради Буч и др. Объектно-ориентированный анализ и проектирование с примерами приложений (UML 2). 3-е изд.: Пер. с англ. – М.: Издательский дом «Вильямс», 2010. – 720 с.