

УДК 004.896

## ИСПОЛЬЗОВАНИЕ МЕТОДА ПОТЕНЦИАЛЬНЫХ ФУНКЦИЙ ДЛЯ УВЕЛИЧЕНИЯ ОБУЧАЮЩЕЙ ПОСЛЕДОВАТЕЛЬНОСТИ В НЕЙРОСЕТЕВЫХ СИСТЕМАХ

И.Ф. ЯСИНСКИЙ

Ивановская государственная текстильная академия, Иваново, Россия

E-mail: igor2266@yandex.ru

### Авторское резюме

**Состояние вопроса:** Обучение нейронных сетей требует довольно длинных обучающих последовательностей. При этом на практике исследователь часто не имеет последовательностей достаточной длины.

**Материалы и методы:** Предложен способ преодоления этой трудности с помощью привлечения потенциальных функций.

**Результаты:** Удалось применить нейросетевую технологию в задачах с короткими обучающими последовательностями.

**Выводы:** Комбинация потенциальных функций и нейросетевой технологии оказывается эффективным средством распознавания образов и прогнозирования процессов.

**Ключевые слова:** нейронные сети, обучающая последовательность, метод потенциалов.

## POTENTIAL FUNCTIONS METHOD TO INCREASE LEARNING IMAGES AMOUNT IN NEURAL NETWORKS

I.F. YASINSKIY

Ivanovo State Textile Academy, Ivanovo, Russia

E-mail: igor2266@yandex.ru

### Abstract

**Background:** Neural networks learning demands great amounts of learning images. At the same time a scientist often is not provided with such amounts of learning images.

**Materials and methods:** The suggested method allows overcoming this problem of learning images by means of potential functions.

**Results:** Neural networks technology in the problems with the lack of learning images was applied.

**Conclusions:** Potential functions set and neural networks technology turned out to be effective in image recognition and process prediction.

**Key words:** Neural networks, learning images, potentials method.

Известно, что точечный электрический заряд образует поле. В этом случае напряженность поля для каждой точки пространства определится при помощи выражения

$$P = a \frac{q}{r^2} = \quad (1)$$
$$= a \cdot q / \left( (x - x_*)^2 + (y - y_*)^2 + (z - z_*)^2 \right),$$

где  $a$  – коэффициент;  $q$  – величина заряда;  $r$  – расстояние от заданной точки до заряда;  $x_*$ ,  $y_*$ ,  $z_*$  – координаты заряда;  $x$ ,  $y$ ,  $z$  – координаты точки [1].

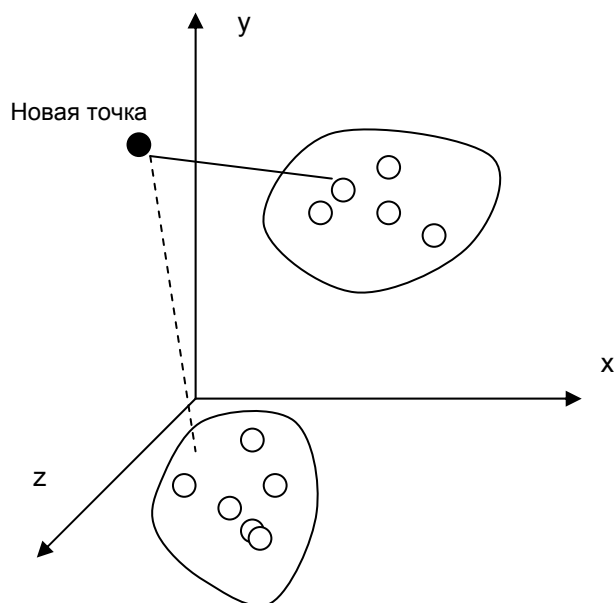
В дальнейшем условно будем называть величину  $P$  потенциалом. Данная формула также позволяет, исходя из величины источника заряда и потенциала в заданной точке, определить расстояние от точки до заряда, т. е. оценить удаленность точки от заряда.

В случае, если имеются несколько зарядов, потенциал в каждой точке пространства будет состоять из потенциалов, образуемых всеми зарядами, и может выступить в качестве

меры удаленности этой точки от совокупности указанных зарядов.

Предположим, что у нас имеются несколько классов образов, представляющих собой наборы определенных параметров. Эти параметры могут выступать в качестве координат зарядов в некотором пространстве параметров. Образы, составляющие класс, часто образуют более плотные обособленные образования в этом пространстве.

Будем считать образы в пространстве параметров единичными зарядами. При этом если некоторая точка, появляющаяся в процессе обучения, окажется ближе к одной из групп зарядов (и следовательно, значение суммарного потенциала, образуемого зарядами (образами) такой группы, в ней больше), то эту точку следует отнести к данной группе (см. рисунок).



Классы образов в пространстве признаков

Заменим потенциал по формуле (1) подобной функцией  $\varphi$ , которая максимальна в точке-источнике и равномерно убывает по мере удаления от нее:

$$\varphi = \frac{1}{1 + \alpha R^2}, \quad (2)$$

где  $\alpha$  – коэффициент, определяющий скорость убывания функции с ростом расстояния;  $R$  – расстояние между точкой-источником и точкой, где вычисляется потенциал. Замена (1) на (2) позволяет избежать нуля в знаменателе.

Нами предлагается метод, который позволяет обучить нейронную сеть в условиях дефицита обучающих данных.

Суть метода заключается в стохастическом размножении обучающих образов для нейронной сети при помощи потенциальных функций.

Имеющиеся в начальный момент обучающие образы представляются как точки-источники потенциалов, группирующиеся по классам. Затем вызывается процедура, которая с учетом специфики задачи, используя случайные числа, генерирует новые образы вблизи исходных. Полученные таким способом образы относятся по методу потенциалов к одному из существующих классов. Таким образом, получены два множества:

- 1) исходное;
- 2) сгенерированное.

Они оба будут использованы для обучения нейросети.

На этом принципе может быть построен алгоритм узнавания.

#### Шаг 1. Подготовка исходных образов.

В начале обучения нормализуются и запоминаются значения параметров всех образов и сохраняется информация – к какому из классов относится каждый образ.

#### Шаг 2. Размножение образов.

Создается массив новых образов с использованием случайных чисел.

#### Шаг 3. Классификация новых образов.

3.1. Для каждого нового образа вычисляются потенциалы для каждого класса:

$$F_a = \frac{1}{n_a} \sum_{i=1}^{n_a} \varphi_{ai}, \quad F_b = \frac{1}{n_b} \sum_{i=1}^{n_b} \varphi_{bi}, \quad F_c = \frac{1}{n_c} \sum_{i=1}^{n_c} \varphi_{ci},$$

$$F_d = \frac{1}{n_d} \sum_{i=1}^{n_d} \varphi_{di},$$

где  $a, b, c, d$  – обозначения распознаваемых классов образов;  $n_a, n_b, n_c, n_d$  – количества образов в каждом классе;  $\varphi_{ai}, \varphi_{bi}, \varphi_{ci}, \varphi_{di}$  – потенциалы, образованные в распознаваемой точке  $i$ -м образом соответствующего класса  $a, b, c, d$ .

3.2. Потенциалы сравниваются, и новый образ относится к тому классу, который создаст в ней больший потенциал.

Ниже предлагается фрагмент программы, реализующий описанные действия.

```
for j:=1 to 50 do //цикл от 1 до числа образов
```

```
begin
```

```
for i:=1 to ni do O[i]:=round(random);
```

```
//формирование нового случайного образа
```

```
//вычисляем потенциал от класса A//////////
```

```
N_obr:=0;//номер класса
```

```
fi:=0; //обнуление потенциала
```

```
N_A:=0; //номер образа в классе
```

```
metka1:
```

```
образ; //вызов процедуры, записывающей в массив O[i] код исходного образа исходя из номера класса и номера образа в классе
```

```
R:=0; //обнуление переменной расстояния
```

```
for i:=1 to ni do R:=R+(O[i]-O[i])*(O[i]-O[i]); //вычисление
```

```
R:=sqrt(R); //расстояние от соответствующего исходного образа класса до нового образа
```

```
fi:=fi+1/(1+alfa*R*R); //вычисление потенциала, образуемого в данной новой точке классом A
```

```
N_A:=N_A+1; //инкремент номера образа в классе
```

```
if N_A<=4 then goto metka1; //если не последний образ в классе, то возврат
```

```
F_A:=fi/5; // деление полученного потенциала группы A на число образов в группе
```

```
//аналогичные действия производим для остальных классов и вычисляем потенциалы, образуемые ими в каждой новой точке (образе)
```

```
//в зависимости от потенциалов относим новый образ к одному из классов
```

```
if (F_A>F_B) and (F_A>F_C) and (F_A>F_D) then class[j]:=0;
```

```
if (F_B>F_A) and (F_B>F_C) and (F_B>F_D) then class[j]:=1;
```

```
if (F_C>F_B) and (F_C>F_A) and (F_C>F_D) then class[j]:=2;
```

```
if (F_D>F_B) and (F_D>F_C) and (F_D>F_A) then class[j]:=3;
```

```
for i:=1 to ni do Oobj[j,i]:=O[i];//сохраняем
новый образ в отдельном массиве
end; //конец цикла
```

Для оценки эффективности изложенного метода нами разработана программа, моделирующая обучение и работу нейронной сети, которая училась распознавать символы (латинские буквы). Обучающее множество делилось на 4 класса (буквы A,B,C,D) по 5 различным способам изображения в каждой группе (всего 20 образов). Рецептное поле такой сети содержало 25 разрядов, скрытый слой – 10 нейронов, на выходном слое – 2 нейрона. Обучение проводилось по алгоритму обратного распространения ошибки [2], [3].

В начале программы массив образов был дополнен новыми случайными образами, которые были затем распределены по классам согласно методу потенциальных функций. При обучении данные на вход подавались из массива исходных образов и из массива новых образов в соотношении 1:1. Численные эксперименты по работе с программой приведены в таблице.

Число добавленных образов, ед.	Средняя обученность нейронной сети за 100 опытов (20 000 шагов обучения для каждого опыта), %	
	с добавлением образов	без добавления образов
20	71,04	71,19
40	66,38	
50	66,35	
100	61,58	

*Ясинский Игорь Федорович*,  
Ивановская государственная текстильная академия,  
кандидат технических наук, доцент кафедры прикладной математики и информационных технологий,  
e-mail: igor2266@yandex.ru

## Заключение

Предложенный способ получения дополнительных образов делает возможным обучение нейронной сети в условиях дефицита обучающей информации при помощи потенциальных функций.

Оптимальным является 2-х кратное увеличение обучающего множества. Согласно полученным результатам численных экспериментов, имеется возможность большего расширения обучающей выборки за счет относительно небольшого понижения процента распознаваемости нейронной сети. По мере эксплуатации нейросети она дообучается.

## Список литературы

1. Айзерман М.А., Браверманн Э.М., Розоноэр Л.И. Теоретические основы метода потенциальных функций в задаче об обучении автоматов распознаванию классов // Автоматика и телемеханика. – 1964. – Т. XXV. – Вып. 6.
2. Ясинский И.Ф. О двух способах настройки нейронных сетей // Вестник ИГТА. – 2006. – № 4. – С.116–120.
3. Хайкин С. Нейронные сети. Полный курс. – М.: Изд-во «Вильямс», 2006. – 1104 с.

## References

1. Ayzerman, M.A., Bravermann, E.M., Rozonoer, L.I. *Avtomatika i telemekhanika*, 1964, vol. XXV, issue 6.
2. Yasinskiy, I.F. *Vestnik IGTA*, 2006, 4, pp.116–120.
3. Khaykin, S. *Neuronnye seti* [Neural networks], Moscow: Vil'yams, 2006, 1104 p.