УДК 004.852:004.855.5

НЕЙРОСЕТЕВОЙ АЛГОРИТМ УТОЧНЕНИЯ ВЕКТОРНЫХ ТОПОГРАФИЧЕСКИХ КАРТ ПО ДАННЫМ ДИСТАНЦИОННОГО ЗОНДИРОВАНИЯ ЗЕМЛИ

Акинин М.В., *аспирант кафедры «Космические технологии» Рязанского государственного радиотехнического университета, e-mail: akinin.m.v@gmail.com*

Ключевые слова: искусственная нейронная сеть, машина опорных векторов, нейронная карта Кохонена, многослойный перцептрон, нейронная сеть прямого распространения грамматика графовой генерации Китано, генетический алгоритм.

Рассмотрен алгоритм уточнения векторных топографических карт по данным дистанционного зондирования Земли. Предложено использование интеллектуальной системы, основанной на машинах опорных векторов, многослойных перцептронах, нейронных сетях прямого распространения, нейронной карте Кохонена. Рассмотрен подход к обучению нейронной сети прямого распространения с помощью генетического алгоритма и грамматик графовой генерации Китано.

Введение. Постановка задачи

Задача оперативного составления и обновления топографических карт является одной из важнейших решаемых задач в различных сферах экономики, природопользования и обеспечении безопасности жизнедеятельности человека.

Данная задача может быть решена посредством анализа данных дистанционного зондирования Земли (ДЗЗ) полу- и полностью автоматическими интеллектуальными системами, что обеспечивает высокую точность результата и низкие временные затраты на его получение.

Топографическая карта подготавливается в геоинформационной системе на основе набора векторных слоев, каждый из которых описывает положение различных объектов местности. Задача уточнения топографической карты сводится, таким образом, к задаче уточнения набора векторных слоев, которая, в свою очередь, сводится к получению векторного слоя V для каждого исходного слоя:

$$V_{old} \overset{f}{\rightarrow} V,$$

где f – функционал уточнения, имеющий вид

$$V = f(V_{old}, I).$$

Здесь I – спутниковый снимок или данные аэрофотосъемки.

I можно представить в виде

$$I = \{i(m,n,k)\}; i(m,n,k) \in \{-1,1\};$$
$$m = \overline{1,M}, n = \overline{1,N}, k = \overline{1,K}; M,N,K \in \mathbb{N},$$

где $i(m,n,k)$ – значение в спектральном канале k пикселя с координатами $(m,\ n)$; M,N,K – количество строк, столбцов и спектральных каналов в изображении I.

Каждому пикселю $i(m,\ n)$ можно поставить в соответствие набор координат $(x,\ y,\ z)$ в системе координат карты: $(x,\ y,\ z) = map(m,\ n)$. Таким образом достигается наложение данных ДЗЗ на векторные слои карты.

Предметом рассмотрения в настоящей статье является функционал f уточнения векторных слоев карты. Он может быть построен разными методами, однако в настоящее время все чаще стали использоваться методы искусственного интеллекта, основанные на теории машинного обучения. В настоящей статье рассматривается только простейший – двухмерный случай. Описываемый алгоритм может применяться также и для уточнения трехмерных моделей местности.

Алгоритм уточнения

Предлагаемый алгоритм уточнения топографических карт состоит из следующих этапов.

Этап 1. Предварительная обработка данных ДЗЗ I_{raw} – коррекция шумов, конвертирование в систему координат карты, проецирование в двухмерную проекцию карты (координата z здесь и далее принимается равной нулю).

Этап 2. Преобразование I_{raw} из исходного формата представления спектральных каналов к формату представления, используемому алгоритмом:

$$i(m,n,k) = spectral\_convert(i_{raw}(m,n,k)) =$$
$$= a * i_{raw}(m,n,k) + b;$$
$$m = \overline{1,M}, n = \overline{1,N}, k = \overline{1,K}; a,b \in \mathbb{R},$$

где a,b – коэффициенты линейного преобразования, выбираемые в зависимости от исходного формата представления спектральных каналов.

Так как чаще всего исходным форматом является беззнаковое целое 8-ми битное число, то коэффициенты a,b принимают значения:

$$a = \frac{2}{255},\ b = -1.$$

Этап 3. Классификация пикселей изображения I с получением на выходе маски C:

$$c(m,n) = classification(i(m,n), V_{old}, I_{old});$$
$$c(m,n) \in P;$$
$$m = \overline{1,M};\ n = \overline{1,N};\ P \subset \mathbb{N},$$

где P – множество индексов классов. Каждый индекс p маркирует один из возможных классов объектов, которые могут присутствовать в векторном слое, I_{old} – данные ДЗЗ, по которым был построен векторный слой V_{old}.

Этап 4. Векторизация маски C с получением на выходе векторного слоя V. Элементы $c(m,n)$ маски C составляют элемент векторного слоя V (точку, линию или полигон – в зависимости от типа слоя) в случае, если они имеют одинаковое значение (принадлежат к одному и тому же классу) и образуют связное множество точек в декартовой системе координат относительно центра маски.

Векторизация может быть выполнена с помощью алгоритмов, описанных в [1] и [2].

Классификатор

Ключевым этапом представленного алгоритма является классификация пикселей изображения I с получением на выходе маски C. Классификация выполня-ется с помощью классификатора $classification$. Сущест-вуют различные способы построения классификатора: ручные, полуавтоматические (с ручным подбором параметров алгоритма, с ручным этапом верификации результатов классификации и прочие) и автоматические. Наибольший интерес представляют автоматические способы построения классификатора из-за возможности существенно снизить временные затраты на классификацию.

В качестве автоматически настраиваемого классификатора $classification$ могут быть использованы решения, основанные на концепциях теории искусственного интеллекта и теории машинного обучения.

В ходе подготовки статьи был разработан классификатор, работающий по алгоритму, состоящему из следующих этапов.

Этап 1. Предварительная классификация пикселей $i(m,\ n)$ с формированием на выходе предварительной маски C_{pre}. Для предварительной классификации пикселей используется нейронная карта Кохонена.

Этап 2. Сопоставление маски C_{pre} и векторного слоя V_{old} с формированием на выходе списка классифицированных и неклассифицированных пикселей $i(m,\ n)$.

Каждый пиксель $i(m,\ n)$ маркируется как классифицированный в случае, если мера расстояния $d(I(m,\ n),\ I_{old}(m,\ n))$ не превосходит некоторого порога $T; T \in \mathbb{R}^+$, выбираемого адаптивно в зависимости от гистограмм изображений I, I_{old} – чем меньше контраст изображений (чем ярче выражены моды гистограммы и чем ближе моды гистограммы друг к другу), тем меньше значение порога T, что необходимо для повышения чувствительности алгоритма на однородных изображениях. Метрика d выбирается в зависимости от решаемой задачи. Как показали эксперименты, для уточнения топографических карт разумным выбором является Евклидова мера расстояния.

Этап 3. Обучение ядра классификатора $kernel$. Ядро классификатора обучается на классифицированных пикселях $i(m,\ n)$.

Этап 4. Классификация пикселей $i(m,\ n)$ с формированием на выходе маски C. Если пиксель $i(m,\ n)$ включен в список классифицированных, то $c(m,n) = c_{pre}(m,n)$, иначе $c(m,n) = kernel(i(m,n))$.

Ядро классификатора

Ядро классификатора представляет собой набор простых классификаторов $classifier\_simple$, организованных в древовидную структуру.

Каждый простой классификатор выполняет разбиение исходного векторного подпространства $Q_{sub} \subset Q$ векторного пространства Q, составленного из векторов $i(m,\ n)$, на два класса. Будучи отнесенным к тому или иному классу, вектор $i(m,\ n)$ спускается на следующий уровень дерева до тех пор, пока не достигнет одного из стоков. Каждый из стоков дерева соответствует одному из результирующих индексов p возможных классов.

Обучение ядра классификатора сводится к обучению простых классификаторов.

Обучение простого классификатора является итеративным процессом, состоящим из следующих этапов:

– подбор параметров обучения и прогона классификатора;

– обучение классификатора по тому или иному алгоритму.

В качестве простых классификаторов могут быть использованы следующие концепции теории машинного обучения:

– машина опорных векторов [4];

– многослойный перцептрон [5];

– нейронная сеть прямого распространения.

Нейронная сеть прямого распространения

Интеллектуальная нейронная сеть (ИНС) прямого распространения может быть представлена как функция $y = ann(v)$ – функция, производящая для каждого входного вектора v соответствующий отклик y.

ИНС прямого распространения состоит из нескольких нейронов, связанных друг с другом синапсами, по которым передаются сигналы, циркулирующие в сети. Каждый нейрон обладает некоторым тормозящим пороговым значением, определяющим уровень реакции нейрона на соответствующее возбуждение. Термин «прямое распространение» означает, что в ИНС отсутствуют циклы. Каждый нейрон сети подвергает свой входной вектор некоторому преобразованию в соответствии со своим пороговым значением и своей функцией активации, передавая свой выход по соответствующим синапсам другим нейронам.

Большинство ИНС прямого распространения относятся к классу нейронных сетей, обучаемых с учителем. Для обучения таковой ИНС используется обучающее

множество V_{teach} векторов, для которого известно множество идеальных откликов d_{teach} . Для тестирования качества обучения ИНС используется тестовое множество V_{test} векторов, для которых также известны идеальные отклики d_{test} .

В общем случае, подбор структуры ИНС прямого распространения представляется сложной задачей. Существует ряд подходов к ее решению, из которых наиболее развитым является подход с использованием генетического алгоритма [6] для поиска оптимальной структуры нейронной сети с точки зрения среднеквадратичного функционала ошибки

$$E(ann, V_{test}) = \frac{\sum_i (ann(v_{test\ i}) - d_{test\ i})^2}{V_{test}}; \quad i = \overline{1, V_{test}}.$$

Важной проблемой, решаемой в процессе поиска оптимальной структуры ИНС с помощью генетического алгоритма, является проблема кодирования структуры ИНС в виде непрерывной последовательности символов – последовательности хромосом, составляющих единственный ген генома каждой особи.

H. Kitano в [7] предложил способ кодирования структуры ИНС, называемый грамматиками графовой генерации Китано. В настоящей статье рассматривается одно из развитий грамматик графовой генерации Китано, которое заключается в введении дополнительного параметра *W*, определяющего глубину разбора начального правила грамматики и позволяющего задавать с помощью грамматик, содержащих одно и то же количество правил, сети произвольного размера.

Структура ИНС прямого распространения может быть представлена в виде ориентированного графа без циклов. Каждая дуга в данном графе соответствует синапсу и имеет собственный вес, устанавливаемый в процессе обучения сети. Каждая вершина графа соответствует одному нейрону; с каждой вершиной связано пороговое значение соответствующего нейрона, настраиваемое в процессе обучения сети.

Имеющиеся в графе дуги могут быть заданы с помощью матрицы смежности G (1) ИНС, в которой единица кодирует наличие ребра, направленного от вершины с номером, равным номеру строки, к вершине с номером, равным номеру столбца, а ноль – соответственно, отсутствие такового ребра.

$$G = \begin{pmatrix} g_{11} & \dots & g_{1H_G} \\ \dots & \dots & \dots \\ g_{H_G 1} & \dots & g_{H_G H_G} \end{pmatrix};$$

$$g_{ij} \in \{0,1\}; \quad i = \overline{1, H_G}; \quad j = \overline{1, H_G}.$$

Матрицу G можно закодировать с помощью набора правил порождающей грамматики A :

$$A = \{T, T_P, T_N, F_T, F_P, F_N, S\};$$

$$T = \{0,1,\omega\}; \quad T_P = \{T_{P\ i}\}; \quad i = \overline{1,16};$$

$$T_N = \{T_{N\ j}\}; \quad j = \overline{1, H_N}; \quad S \in T_N;$$

$$F_T = \{1 \to \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}; \quad 0 \to \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}; \quad \omega \to 0\};$$

$$F_P = \{F_{P1} \to \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \ F_{P2} \to \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \dots, F_{P16} \to \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}\};$$

$$F_N = \{F_{N\ i}\}; \quad F_{N\ i} : T_{N\ i} \to \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix};$$

$$i = \overline{1, H_N}; \quad \alpha, \beta, \gamma, \delta \in (F_N \cup F_P),$$

где: T , T_P , T_N , F_T , F_P , F_N , S – наборы терминальных символов, предтерминальных символов, нетерминальных символов, правил вывода для терминальных символов, правил вывода для предтерминальных символов, правил вывода для нетерминальных символов и стартовый нетерминальный символ соответственно.

Грамматика A называется грамматикой графовой генерации Китано. Одна грамматика A соответствует одной особи популяции генетического алгоритма. Элементы T , T_P , T_N , F_T , F_P фиксированы для любой A , и, следовательно, ген особи кодирует только набор правил F_N . Набор правил F_N может быть закодирован по схеме, приведенной на рис. 1. Каждое правило F_{Ni} отображается в четыре хромосомы α_i , β_i , γ_i , δ_i , следующих друг за другом и за хромосомой $\delta_{i-1}\sigma_X^2$ в единственном гене b особи.

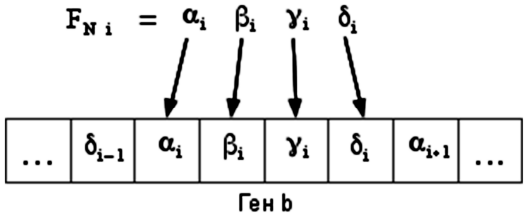

Рис. 1. Схема кодировки набора правил F_N

Алгоритм восстановления матрицы G из грамматики A состоит из следующих этапов.

Этап 1. Создание начальной матрицы $G_1 = S$.

Этап 2. Выполнение $w = 1$ шаг применения правил F_T , F_P и F_N : $G_1 = S \to G_2 = \begin{pmatrix} \alpha_S & \beta_S \\ \gamma_S & \delta_S \end{pmatrix}$.

Этап 3. Повторение этапа 2 для каждого из правил α_S , β_S , γ_S , δ_S : $G_2 = \begin{pmatrix} \alpha_S & \beta_S \\ \gamma_S & \delta_S \end{pmatrix} \to G_3$; размерность G_3 оказывается, таким образом, равной $2^{(3-1)}$ на $2^{(3-1)}$ = 4 на 4 правила.

Этап 4. Повторение этапа 3 $(W-2)$ раз - до получения матрицы G_W .

Этап 5. В матрице G_W любой символ $g \in (T_P \cup T_N)$ рассматривается как символ ω и преобразуется по соответствующему правилу в 0.

Этап 6. Преобразование матрицы G_W в матрицу G по формуле (2) – таким образом, в матрице G гарантированно будут отсутствовать циклы.

$$G = (g_{ij}); \quad G_W = (g_{W\ ij});$$
$$g_{ij} = \begin{cases} 0, & i \geq j \\ g_{W\ ij}, & i < j \end{cases}; \quad i = \overline{1, H_G}; \quad j = \overline{1, H_G} \qquad (2)$$

Параметр W алгоритма восстановления матрицы смежности G определяет максимально возможное количество нейронов, которое может быть в результирующей ИНС – 2^{W-1} нейронов. Учитывая тот факт, что число входов и выходов ИНС фиксировано, то к ИНС, описываемой соответствующим графом, присоединяются несколько нейронов:

– входные нейроны в достаточном количестве – связываются синапсами с истоками графа; данные синапсы направлены от входных нейронов к истокам графа;

– выходные нейроны в достаточном количестве – связываются синапсами со стоками графа; данные синапсы направлены от стоков графа к выходным нейронам.

Мощность H_N множества T_N оказывает, равно как и параметр W, существенное влияние на качество работы генетического алгоритма, поскольку именно оно определяет длину гена (как $4H_N$ хромосом) и, соответственно, степень «разнообразия» правил, кодируемых геном. Выбор H_N осуществляется эвристическим способом – небольшое значение H_N приведет к вырождению популяции и ее схождению к локальному оптимуму, тогда как больше значение H_N, наоборот, увеличит разнообразие особей, но также увеличит время схождения генетического алгоритма к, возможно, глобальному оптимуму.

По сравнению с традиционной версией грамматик графовой генерации Китано, предложенная версия позволяет одинаково компактно описывать нейронные сети с любым максимальным количеством нейронов, что достигается за счет параметра W.

Генетический алгоритм

Генетический алгоритм, используемый для поиска оптимальной структуры ИНС и использующий для кодирования структуры ИНС грамматики графовой генерации Китано, состоит из следующих этапов.

Этап 1. Создание множества $V = \{v_i\}; \quad i = \overline{1, H_V}$ векторов из исследуемой предметной области и множество $d = \{d_i\}; \quad i = \overline{1, H_V}$ идеальных откликов ИНС на соответствующие вектора из V; на основе множеств V и d на каждой итерации генетического алгоритма будут составляться обучающие V_{teach}, d_{teach} и тестовые V_{test}, d_{test} множества векторов и откликов.

Этап 2. Генерация случайной начальной популяции особей $B = \{b_i\}; \quad i = \overline{1, H_B}$.

Этап 3. Выделение из множества V случайным образом подмножеств V_{teach}, V_{test}; для множеств V_{teach} и V_{test} из множества d выделяются соответствующие им подмножества d_{teach} и d_{test}.

Этап 4. Восстановление грамматики A_i для каждой особи $b_i \in B$.

Этап 5. Расчет для каждой особи $b_i \in B$ функции приспособленности

$$Q(b_i) = \frac{1}{E(ann_i, V_{test}) + \varepsilon}; \quad \varepsilon \in \mathbb{R}; \quad \varepsilon < 10^{-8}.$$

Этап 6. Сортировка порядка следования особей $b_i \in B$ по убыванию функции приспособленности $Q(b_i)$.

Этап 7. Особи с индексами $i \in \left[1, H_{B\ best}\right]$ образуют множество особей $B_{best} \subset B$, особи с индексами $i \in (H_{B\ best}, H_{B\ best} + H_{B\ cross}]$ – множество особей $B_{cross} \subset B;;$ особи с индексами $i \in (H_{B\ best} + H_{B\ cross}, H_{B\ best} + H_{B\ cross} + H_{B\ mut}]$ – множество особей $B_{mut} \subset B$, при этом должны выполняться условия:

$$H_{B\ best} \geq 2;$$
$$H_{B\ cross} = \sigma H_{B\ mut}; \quad \sigma \in \mathbb{R}; \quad \sigma > 1;$$
$$H_{B\ best} + H_{B\ cross} + H_{B\ mut} = H_B;$$
$$B_{best} \cap B_{cross} = \varnothing; \ B_{best} \cap B_{mut} = \varnothing; \ B_{cross} \cap B_{mut} = \varnothing.$$

Этап 8. Замена особей $b_i \in B_{cross}$ на особи $b_{cross\ kl} = cross(b_k, b_l); \quad b_k, b_l \in B_{best}$ – то есть на особи – результаты скрещиваний случайных особей из множества $\mathrm{B_{best}}$ лучших особей популяции.

Этап 9. Замена особей $b_i \in B_{mut}$ на особи $b_{mut\ k} = mutation(b_k); \quad b_k \in B_{best}$ – то есть на особи – результаты мутаций случайных особей из множества $\mathrm{B_{best}}$ лучших особей популяции.

Этап 10. Если лучшая особь популяции b_i была лучшей и соответствовала условию останова $Q(b_1) > P$ на менее чем I предыдущих итераций алгоритма подряд, то выполняется переход на шаг 3, иначе выполняется останов алгоритма. Параметр P задает уровень, определяющий достаточное качество обучения ИНС.

По останову выполнения генетического алгоритма ИНС, генерируемая грамматикой A_i, соответствующей особи b_1 окончательной популяции, принимается как лучшая.

Генетический алгоритм обладает следующими параметрами:

– H_V; $H_{V\ teach} = \{V_{teach}\}$; $H_{V\ test} = \{V_{test}\}$ – определяют размеры обучающей и тестовой выборок векторов; значения данных параметров зависят от предметной области и сложности задачи классификации, решаемой искомой ИНС;

– $H_{B\ best}$; $H_{B\ cross}$; $H_{B\ mut}$; σ – определяют количество особей, сохраняемых при переходе к следующей итерации генетического алгоритма, заменяемых на результаты скрещивания и мутации соответственно;

– $cross$, $mutation$ – функционалы скрещивания и мутации соответственно;

– P – определяет верхний уровень качества обуче-

ния лучшей ИНС, достаточный для останова алгоритма;

– I – определяет достаточное для останова алгоритма количество итераций, на которых подряд должны повторяться условия останова для лучшей особи; очевидно, что чем больше I, тем больше вероятность схождения алгоритма к устойчивому оптимуму – к ИНС, дающей одинаковый по качеству обучения результат на произвольных V_{teach} и V_{test}.

Эксперимент

Для проверки состоятельности предлагаемого способа уточнения топографических карт было поставлено несколько экспериментов, один из которых будет рассмотрен далее.

Эксперимент заключается в уточнении и дополнении топографической карты полигона «Малинищи», располо-женного на границе Рязанского и Пронского районов Рязанской области и названного так по названию села Малинищи, расположенного на данном полигоне.

В качестве уточняемых данных ДЗЗ I_{old} был взят спутниковый снимок части полигона «Малинищи», сделанный сенсором «ETM +» КА «Landsat – 7» 22-го мая 2000-го года. На основе I_{old} была создана уточняемая топографическая карта $\{V_{old}\}$. Задача разработанного алгоритма состоит в том, чтобы уточнить исходную топографическую карту части полигона «Малинищи» и дополнить результирующую карту информацией обо всем полигоне. Уточнение и дополнение топографической карты было выполнено по данным I от 23-го мая 2006-го года, полученным тем же сенсором.

По сравнению с топографической картой $\{V_{old}\}$, составленной для I вручную, алгоритм продемонстрировал высокую точность результата, неправильно классифицировав менее 5% пикселей $i(m, n)$.

В среднем, точность алгоритма достигает 93% правильно классифицированных пикселей $i(m, n)$ по сравнению с ручной классификацией.

Разработанный алгоритм показал высокую временную эффективность – спутниковый снимок I с размерами $M = 4000, N = 4000, K = 6$ может быть обработан в среднем за 30 – 40 мс, что достаточно для обработки 25 кадров в секунду (25 FPS (англ. frame per second – количество кадров в секунду) – частота, достаточная для формирования видеопоследовательности, плавной и непрерывной для зрительной системы среднестатистического человека).

Заключение

В ходе проведения экспериментов разработанный алгоритм продемонстрировал следующие достоинства:

– высокая точность результата – не менее 93 % пикселей исходных данных ДЗЗ классифицируются правильно;

– низкие временные затраты – 30–40 мс. на 6-ти канальное изображение размером 4000 на 4000 пикселей (типичный спутниковый снимок), что позволяет достичь 25 FPS;

– высокий уровень автоматизации процесса уточнения.

Недостатками разработанного алгоритма являются высокая сложность его реализации и необходимость предварительной автоматической настройки классификатора для улучшения его временных характеристик.

Литература

1. Шапиро Л., Стокман Д. Компьютерное зрение – М.: БИНОМ. Лаборатория знаний. - 2006.

2. Гонсалес Р., Вудс Р. Цифровая обработка изображений // Пер. с английского под редакцией П.А. Чочиа. – М.: Техносфера. – 2005.

3. Хайкин С. Нейронные сети: Полный курс, 2-е изд., испр.: Пер. с англ. – М.: ООО «И.Д. Вильямс» – 2006.

4. Акинин М.В., Конкин Ю.В. Технология тематического дешифрирования спутниковых изображений на основе машины опорных векторов. // Информатика и прикладная математика: межвузовский сборник научных трудов. – Рязань: Рязанский государственный университет. – 2010.

5. Акинин М.В., Конкин Ю.В. Исследование подходов к обучению многослойного перцептрона. // Методы и средства обработки и хранения информации: межвузовский сборник научных трудов. – Рязань: Рязанский государственный радиотехнический университет. – 2012.

6. Аксенов С.В., Новосельцев В.Б. Организация и использование нейронных сетей (методы и технологии). – Томск: НТЛ. – 2006. – 128 с.

7. Kitano H. Designing neural network using genetic algorithm with graph generation system // Complex Systems. Vol. 4. – 1990.

USING ARTIFICIAL NEURAL NETWORK TO CREATE TOPOGRAPHIC MAP BASED ON REMOTE SENSING DATA

Akinin M.V.

Paper describes algorithm to create topographic vector map based on remote sensing data. Paper describes intellectual system based on support vector machine, multilayer perceptron, feed forward neural network and Kohonen's self-organized map. Paper describs algorithm to teach feed forward neural network using genetic algorithm and Kitano's graph generation grammatic.