Методы, технологии и приложения искусственного интеллекта

УДК 004.93'12:004.032.26
DOI:10.25729/ESI.2025.39.3.001

Генеративные модели на основе пирамидальных нейронных сетей быстрого обучения

Дорогов Александр Юрьевич

Санкт-Петербургский государственный электротехнический университет,
ПАО «Информационные телекоммуникационные технологии»
Россия, Санкт-Петербург, *vaksa2006@yandex.ru*

Аннотация. В статье предложен способ построения генеративных моделей на основе пирамидальных нейронных сетей быстрого обучения (БНС). В основу построения моделей положен вероятностный метод главных компонент (PPCA). Метод PPCA позволяет аналитически построить матрицы оптимальных декодеров, способных восстанавливать образы из случайных латентных переменных малой размерности, распределённых по нормальному вероятностному закону. Реализация PPCA-декодеров в классе БНС позволяет представить декодеры в виде последовательно-параллельных структур, обеспечивающих высокое быстродействие за счёт структурного распараллеливания операций. В работе представлены методы обучения БНС к матрицам декодеров. Обучение выполняется за конечное число шагов и не требует итерационных процедур. Приводятся примеры построения реализующих БНС для набора данных MNIST. Показаны результаты генерации образов, подобных набору MNIST. Выполнено сравнение с классическим вариационным автоэнкодером. Определена область целесообразного использования генеративных моделей PPCA.

Ключевые слова: генеративная модель, вариационный автоэнкодер, вероятностный метод главных компонент, быстрая нейронная сеть, пирамидальная структура

Цитирование: Дорогов А.Ю. Генеративные модели на основе пирамидальных нейронных сетей быстрого обучения / А.Ю. Дорогов // Информационные и математические технологии в науке и управлении, 2025. – № 3 (39). – С. 5-16. – DOI:10.25729/ESI.2025.39.3.001.

Введение. Термин "генеративная модель" используется для описания моделей, которые способны из независимого шума генерировать экземпляры выходных образов подобные образам обучающей выборки. Генеративность заключается в том, что модель, обученная на некоей базе данных, может воспроизводить новые образы, которые раньше не существовали и которых не было в исходной базе данных, но они вполне реалистичны и сохраняют стилевую близость к обучающим данным.

К семейству генеративных моделей относятся вариационные автоэнкодеры (VAE) [1, 2]. Обычный автоэнкодер, состоит из двух основных частей: энкодера, преобразующего входные данные в сжатое представление (латентное или иначе скрытое пространство), и декодера, который восстанавливает данные из латентного пространства. Вариационный автоэнкодер добавляет в эту архитектуру случайный шум, предполагая, что в латентном пространстве данные распределены согласно предварительно выбранному вероятностному распределению, чаще всего нормальному с единичной матрицей ковариаций (рис. 1).

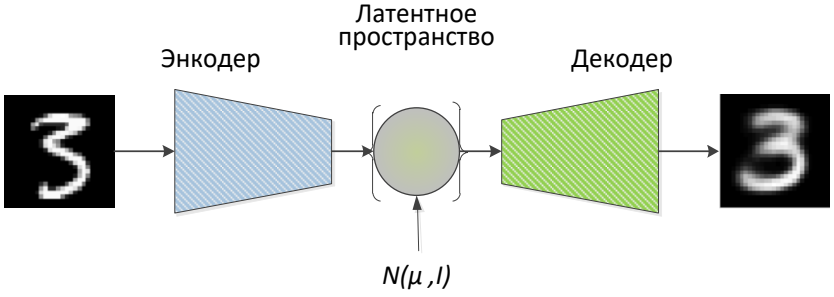

Рис. 1. Архитектура автоэнкодера

При обучении автоэнкодера одновременно решаются две оптимизационные задачи:

– Модель энкодера стремится связать d -мерный вектор обучающей выборки t с некоторым q -мерным случайным вектором латентных переменных с нормальным законом вероятностного распределения:

$$x = f(t, w_E) \to N(\mu, I),$$

где w_E – параметры энкодера, $f(t, w_E)$ – функция отображения целевых образов в латентное пространство, при этом обязательным требованием является обеспечить для значений переменной x, рассматриваемых как случайные, максимальное соответствие предварительно заданному нормальному вероятностному распределению $N(\mu, I)$ с математическим ожиданием μ и единичной ковариационной матрицей. Это требование приводит к тому, что образы обучающей выборки статистически непрерывно покрывают все латентное пространство.

– Модель декодера $t = \varphi(x; w_D)$ с параметрами w_D, напротив, обучается правильно восстанавливать образ t по случайной латентной переменной x при условии, что она соответствует заданному закону распределения $N(\mu, I)$, ожидаемому декодером.

Для реализации энкодера и декодера обычно используются многослойные нейронные сети, которые обучаются методом обратного распространения ошибок с минимизацией одновременно двух критериев: совпадения образов на входе и выходе автоэнкодера и соответствия эмпирического распределения скрытых переменных предварительно заданному вероятностному закону. После того, как обучение завершено, энкодер уже больше не требуется. Для генерации образов на вход декодера достаточно подать значение латентной переменной, выбранной случайным образом из априорно заданного распределения $N(\mu, I)$. Обученный декодер для любого латентного вектора, выбранного из этого распределения, будет возвращать реалистичное цифровое изображение.

Улучшенным вариантом VAE является условный автоэнкодер (Conditional VAE). В этой архитектуре существует дополнительный вход, на который поступает метка класса. По существу, для каждого класса формируется свой вариационный автоэнкодер, который обучается на примерах, принадлежащих только этому классу. Благодаря этому решению реалистичность генерируемых образов улучшается, но приводит к необходимости использования дополнительных ресурсов при реализации архитектуры CVAE. Для больших нейронных сетей процедура обучения VAE и CVAE занимает длительное время и требует большого объёма выборок. Процесс обучения не всегда сходится и может приводить к локальным минимумам.

Вариационные автоэнкодеры имеют линейный прототип− это вероятностный метод главных компонент (англ. Probabilistic Principal Component Analysis – PPCA) [3]. Отличительная особенность метода PPCA состоит в том, что он позволяет найти аналитическое решение для функции отображения из пространства образов в пространство латентных переменных и обратно. Полученное решение можно непосредственно использовать для построения декодера генеративной модели, необходимость в энкодере в этом случае отпадает.

В данной работе будет рассмотрено построение генеративных моделей с использованием метода PPCA на основе многослойных пирамидальных нейронных сетей быстрого обучения [4]. Благодаря самоподобной структуре, сети данного класса способны быстро обучаться к произвольным функциям. Этими функциями могут быть, например, координатные компоненты многомерного отображения векторов из латентного пространства в пространство образов. Алгоритм обучения быстрых нейронных сетей принципиально отличается от метода обратного распространения ошибки отсутствием длительной итерационной процедуры по

минимизации ошибки. Обучение всегда выполняется за конечное число шагов с гарантированной точностью.

1. Вероятностный метод главных компонент. Идея метода PPCA базируется на теоретических разработках конца 90-х годов прошлого века, имеющих общее название Байесовские методы машинного обучения [5, 6, 7]. Следуя работе [5], рассмотрим кратко основные положения данного решения. Согласно методу PPCA, связь между образами обучающей выборки t и латентными переменными x задаётся линейным отображением вида:

$$t = Wx + \mu + \varepsilon, \tag{1}$$

где W – это матрица размерностью $d \times q$, $\varepsilon \sim N\left(0, \sigma^2 I\right)$ – случайный шум, соответствующий нормальному закону с нулевым средним, диагональной ковариационной матрицей и дисперсией σ^2, параметр μ позволяет модели данных иметь ненулевое среднее значение. Для латентных переменных декларируется изотропный нормальный закон распределения с единичной дисперсией:

$$p\left(x\right) = \left(2\pi\right)^{-q/2} \exp\left\{-\frac{1}{2} x^T x\right\}. \tag{2}$$

Выражение (1) подразумевает распределение вероятностей по t-пространству для данного x в виде:

$$p\left(t\middle|x\right) = \left(2\pi\sigma^2\right) \exp\left\{-\frac{1}{2\sigma^2}\left\|t - Wx - \mu\right\|^2\right\}. \tag{3}$$

Используя (2) и (3), получим предельное (маргинальное) распределение для образов t:

$$p\left(t\right) = \int p\left(t\middle|x\right) p\left(x\right) = \left(2\pi\right)^{-d/2} \left|C\right|^{-1/2} \exp\left\{-\frac{1}{2}\left(t-\mu\right)^T C^{-1}\left(t-\mu\right)\right\},$$

где $C = \sigma^2 I + WW^T$, ковариационная матрица (что также можно непосредственно получить из выражения (1)). Используя правило Байеса, можно вычислить апостериорное распределение скрытых переменных x при заданном наблюдаемом t,

$$p\left(x\middle|t\right) = \frac{p\left(t\middle|x\right) p\left(x\right)}{p\left(t\right)} = \left(2\pi\right)^{-q/2} \left|\sigma^2 M\right|^{1/2} \times$$

$$\times \exp\left[-\frac{1}{2}\left\{x - M^{-1}W^T\left(t-\mu\right)\right\}^T \left(\sigma^{-2} M\right)\left\{x - M^{-1}W^T\left(t-\mu\right)\right\}\right],$$

где апостериорная ковариационная матрица задаётся выражением:

$$\sigma^2 M^{-1} = \sigma^2\left(\sigma^2 I + W^T W\right)^{-1}.$$

Размерность матрицы M равна $q \times q$, т.е. определяется размерностью латентного пространства, в то время как размерность ковариационной матрицы C определяется размерностью пространства образов $d \times d$. Надо иметь в виду, что $q \ll d$. Для определения отображения (1) запишем логарифмическую функцию правдоподобия по обучающей выборке размером N:

$$L = \sum_{n=1}^{N} \ln\left\{p\left(t_n\right)\right\} = -\frac{N}{2}\left\{d\ln\left(2\pi\right) + \ln\left|C\right| + tr\left(C^{-1}S\right)\right\},$$

где матрица

$$S = \frac{1}{N}\sum_{n=1}^{N}\left(t_n - \mu\right)\left(t_n - \mu\right)^T,$$

является выборочной ковариационной матрицей обучающей выборки $\{t_n\}$, N – размер обучающей выборки. Для нахождения оптимального отображения функция правдоподобия максимизируется по переменным μ и W, следуя условиям экстремума:

$$\frac{dL}{d\mu}=0, \qquad \frac{dL}{dW}=0.$$

В работе [5] показано, что оптимальные значения определяются выражениями:

$$\mu=\frac{1}{N}\sum_{n=1}^{N}t_n,$$

$$W=U_q\left(\Lambda_q-\sigma^2 I\right)^{1/2}R, \tag{4}$$

где матрица U_q (размером $d\times q$) содержит q векторов-столбцов, которые являются собственными векторами выборочной ковариационной матрицы

$$S=\frac{1}{N}\sum_{n=1}^{N}\left(t_n-\mu\right)\left(t_n-\mu\right)^T$$

с соответствующими собственными значениями $\lambda_1,\lambda_2,\cdots\lambda_q$, размещёнными в диагональной матрице Λ_q в порядке убывания; $R-$произвольная $q\times q$ ортогональная матрица вращения. Также показано, что при оптимальном выборе W оценка максимального правдоподобия для σ^2 в $t-$пространстве определяется как

$$\sigma^2=\frac{1}{d-q}\sum_{j=q+1}^{d}\lambda_j,$$

где $\lambda_{q+1},\lambda_{q+2},\cdots\lambda_d$ – наименьшие собственные значения матрицы S. Для восстановления образа по латентной переменной может быть использовано математического ожидание для модели (1):

$$t=Wx+\mu.$$

В [5] отмечается, что такая оценка возможна, но не является оптимальной. Показано, что оптимальная оценка может быть получена с использованием следующего выражения:

$$t=W\left(W^TW\right)^{-1}Mx+\mu.$$

Таким образом, функция декодера генеративной модели определяется матрицей W или матричным произведением $H=W\left(W^TW\right)^{-1}M$, где $M=\sigma^2 I+W^TW.$ Напомним, что W является прямоугольной матрицей размерности $d\times q$, а M – квадратной матрицей размерности $d\times d$ и $q\ll d$.

2. Пирамидальные нейронные сети быстрого обучения. Теория данного класса сетей представлена в работах [4, 8]. Рассмотрим построение декодера на обратно-ориентированных пирамидальных нейронных сетях. Эти сети имеют глубокие связи с алгоритмом быстрого преобразования Фурье (БПФ). Возьмём за основу топологическую модель «Кули-Тьюки с прореживанием по времени» [9]:

$$U^m=\left\langle u_{n-1}u_{n-2}\cdots u_{m+1}u_m v_{m-1}v_{m-2}\cdots v_1v_0\right\rangle,$$

$$V^m=\left\langle u_{n-1}u_{n-2}\cdots u_{m+1}v_m v_{m-1}v_{m-2}\cdots v_1v_0\right\rangle,$$

$$z^m=\left\langle u_{n-1}u_{n-2}\cdots u_{m+1}v_{m-1}v_{m-2}\cdots v_1v_0\right\rangle.$$

В этой модели кортежи представляют собой поразрядные представления позиционных номеров рецепторов – U^m, аксонов – V^m и нейронных ядер – z^m в пределах нейронного слоя m. В свою очередь, разрядные переменные u_m – это позиционный номер рецептора в

пределах ядра, а v_m – позиционный номер аксона в пределах ядра слоя m (нумерация рецепторов, аксонов и нейронных ядер начинается с нуля). Сеть имеет послойную модульную структуру с числом слоев, равным n. Нейронный модуль – это нейронное ядро (в терминах БПФ нейронное ядро соответствует базовой операции, например, «бабочка»). Все ядра в пределах слоя m имеют размерности p_m – по входу и g_m – по выходу. Нейронное ядро представляет собой матрицу весов размером $p_m \times g_m$, Функции активации и входы смещений отсутствуют. Совокупность чисел $p_0, p_1, \cdots, p_{n-1}$ и $g_0, g_1, \cdots, g_{n-1}$ определяют структурные характеристики сети. Сеть линейна и относится к категории быстрых нейронных сетей (БНС). Для построения пирамидальной сети выберем структурные характеристики сети следующим образом:

– размерности рецепторных полей сети положим равными $p_0 = p_1 = \cdots = p_{n-2} = 1$, но значение $p_{n-1} \neq 1$ может быть выбрано произвольным, в данном случае это значение выбирается равным размерности латентного пространства; в этом случае размерность сети по входу будет равна $N = p_0 p_1 \cdots p_{n-1} = p_{n-1}$;

– размерности аксоновых полей сети зададим произвольными натуральными целыми числами $g_0, g_1, \cdots, g_{n-1}$; размерность сети по выходу в этом случае будет равна $M = g_0 g_1 \cdots g_{n-1}$; это значение выбирается равным размерности $t-$ пространства.

Топологическая модель при данных структурных характеристиках будет иметь вид:

$$U^m = \langle u_{n-1} 0_{n-2} \cdots 0_{m+1} 0_m v_{m-1} v_{m-2} \cdots v_1 v_0 \rangle,$$
$$V^m = \langle u_{n-1} 0_{n-2} \cdots 0_{m+1} v_m v_{m-1} v_{m-2} \cdots v_1 v_0 \rangle,$$
$$z^m = \langle u_{n-1} 0_{n-2} \cdots 0_{m+1} v_{m-1} v_{m-2} \cdots v_1 v_0 \rangle.$$

Появление нулей в представлении модели объясняется тем, что при основании разрядной переменной, равной 1, эта переменная может принимать только нулевое значение. На рис. 2 приведён пример топологической модели трёхслойной пирамидальной нейронной сети для структурных параметров $[p_0 p_1 p_2] = [113]$, $[g_0 g_1 g_2] = [227]$.

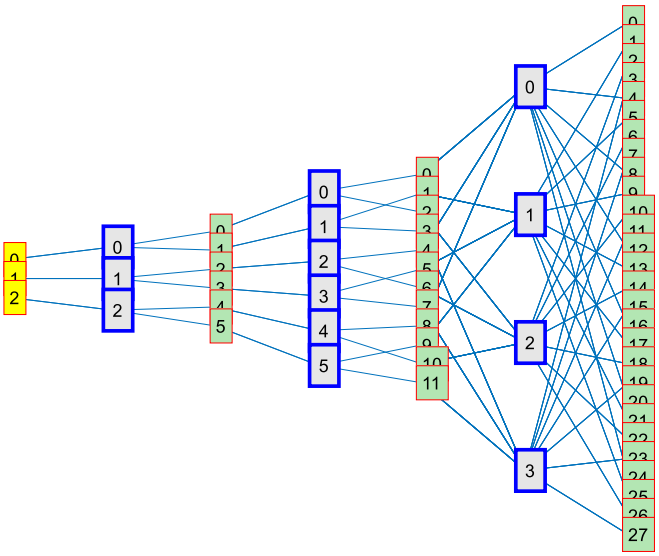

Рис. 2. Топология пирамидальной нейронной сети. Синим цветом выделены нейронные ядра

Сеть имеет размерность, по входу равную 3, а по выходу 28. В [8] показано, что пирамидальная сеть обратной ориентации может хранить произвольные дискретные функции, размерность которых определяется размерностью выхода сети, а их количество –

размерностью входа. Например, сеть, показанная на рис. 2 способна хранить три произвольные вектор-функции размерностью 28. Более того, на её выходе можно получить все линейные комбинации этих функций, т.е. сеть обладает квантовым эффектом.

Следуя [10], рассмотрим кратко способ обучения сети. В соответствии с обобщённой теоремой факторизации элементы матрицы БНС представляются в виде:

$$h(U,V) = w_{z^{n-1}}^{n-1}(u_{n-1}, v_{n-1})\, w_{z^{n-2}}^{n-2}(0_{n-2}, v_{n-2}) \cdots w_{z^0}^{0}(0_0, v_0), \tag{5}$$

где U,V – позиция элемента в матрице БНС, $w_{z^m}^{m}(u_m, v_m)$ – элемент матрицы нейронного ядра с номером z^m в слое m, u_m, v_m – позиция элемента в матрице нейронного ядра. Пусть образы $f^k(V)$ представляют собой набор, состоящий из p_{n-1} дискретных функций, заданных на интервале длинной M. Выполним мультипликативную декомпозицию каждой функции по переменным v_i, начиная со старшего разряда [8], в результате функция образа представляется в виде:

$$f^k(V) = \varphi_{i^{n-1}}^{k}(v_{n-1})\, \varphi_{i^{n-2}}^{k}(v_{n-2}) \cdots \varphi_{i^0}^{k}(v_0), \tag{6}$$

где $i^m = \langle v_{m-1} v_{m-2} \cdots v_1 v_0 \rangle$ – позиционный номер множителя уровня m, k – порядковый номер функции. Сравнивая выражения (5) и (6), непосредственно получим следующее правило обучения нейронной сети:

$$\begin{aligned}
&w_{z^{n-1}}^{n-1}(u_{n-1}, v_{n-1}) = \varphi_{i^{n-1}}^{k}(v_{n-1}) \quad \text{для } m = n-1,\\
&w_{z^m}^{m}(0_m, v_m) = \varphi_{i^m}^{k}(v_m) \quad \text{для } m < n-1,\\
&z^m = \langle u_{n-1} 0_{n-2} \cdots 0_{m+1} v_{m-1} v_{m-2} \cdots v_1 v_0 \rangle,\\
&i^m = \langle v_{m-1} v_{m-2} \cdots v_1 v_0 \rangle.
\end{aligned}$$

Предварительно, для упорядочивания хранимых функций, должно быть установлено взаимно-однозначное соответствие $k \leftrightarrow u_{n-1}$ между порядковым номером функции и разрядной переменной u_{n-1}. Считывание памяти выполняется установкой на входе нейронной сети унарного кода, в котором только один из разрядов равен 1, а остальные нули. Для сети, показанной на рис. 2 – это будут комбинации $[0\,0\,1]$, $[0\,1\,0]$ или $[1\,0\,0]$. При использовании кодов $[1\,1\,0]$ образы 1 и 2 складываются, а при кодах $[1\,-1\,0]$ из первого образа вычитается второй, в общем случае при значениях на входе $[\alpha\ \beta\ \gamma]$ на выходе получим образ $q = \alpha f^1 + \beta f^2 + \gamma f^3$. Заметим, что тот же результат можно получить, если умножить вектор-строку $[\alpha\ \beta\ \gamma]$ на матрицу размерностью 3×28, строки которой совпадают с вектор-функциями f^1, f^2 и f^3:

$$[\alpha\ \beta\ \gamma] \times \begin{bmatrix} f_0^1 & f_1^1 & \cdots & f_{27}^1 \\ f_0^2 & f_1^2 & \cdots & f_{27}^2 \\ f_0^3 & f_1^3 & \cdots & f_{27}^3 \end{bmatrix} = \alpha f^1 + \beta f^2 + \gamma f^3.$$

Отсюда следует, что пирамидальная нейронная сеть является факторизованным представлением прямоугольной матрицы, строками которой являются хранимые образы. Это можно доказать, более строго используя понятие числа степеней свободы [11]. Хранимые образы – это произвольные функции, например, это могут быть строки матрицы декодера. Таким образом, мы показали, что нейронная сеть данного класса может быть использована для построения произвольного декодера генеративной модели, построенной по методу PPCA. Факторизованное представление матрицы состоит из параллельных вычислительных

элементов, организованных, как нейронная сеть, что позволяет обеспечить высокое быстродействие при использовании специализированных процессоров с распараллеливанием операций. В этом состоит главное преимущество нейросетевой реализации.

Пирамидальную нейронную сеть не сложно обобщить на двумерный случай, удобный для обработки изображений. Сохраним для каждой координаты изображения тип топологии рассмотренной выше одномерной модели, тогда топологическая модель пирамидальной сети памяти для двумерного случая примет вид:

$$U_*^m = \left\langle u_{n-1}^* 0_{n-2}^* \cdots 0_{m+1}^* 0_m^* v_{m-1}^* v_{m-2}^* \cdots v_1^* v_0^* \right\rangle,$$
$$V_*^m = \left\langle u_{n-1}^* 0_{n-2}^* \cdots 0_{m+1}^* v_m^* v_{m-1}^* v_{m-2}^* \cdots v_1^* v_0^* \right\rangle,$$
$$z_*^m = \left\langle u_{n-1}^* 0_{n-2}^* \cdots 0_{m+1}^* v_{m-1}^* v_{m-2}^* \cdots v_1^* v_0^* \right\rangle.$$

Символ (*) здесь заменяет символы x и y, означающие принадлежность к координатам изображения. Элементы четырёхмерной матрицы БНС для данной модели также выражаются через элементы нейронных ядер:

$$h\left(U_y U_x, V_y V_x\right) = w^{n-1}_{z_x^{n-1}, z_y^{n-1}}\left(u^y_{n-1}, u^x_{n-1}; v^y_{n-1}, v^x_{n-1}\right) w^{n-2}_{z_x^{n-2}, z_y^{n-2}}\left(0^y_{n-2}, 0^x_{n-2}; v^y_{n-2}, v^x_{n-2}\right) \cdots w^0_{z_x^0, z_y^0}\left(0^y_0, 0^x_0; v^y_0, v^x_0\right),$$

а мультипликативная декомпозиция сохраняемых изображений имеет вид:

$$f^k\left(V_y V_x\right) = \varphi^k_{i_x^{n-1}, i_y^{n-1}}\left(v^y_{n-1}, v^x_{n-1}\right) \varphi^k_{i_x^{n-2}, i_y^{n-2}}\left(v^y_{n-2}, v^x_{n-2}\right) \cdots \varphi^k_{i_x^0, i_y^0}\left(v^y_0, v^x_0\right).$$

Сравнивая два последних выражения, получим правило обучения двумерной пирамидальной нейронной сети:

$$w^{n-1}_{z_x^{n-1}, z_y^{n-1}}\left(u^y_{n-1}, u^x_{n-1}; v^y_{n-1}, v^x_{n-1}\right) = \varphi^k_{i_x^{n-1}, i_y^{n-1}}\left(v^y_{n-1}, v^x_{n-1}\right) \quad для\ m = n-1,$$
$$w^{n-2}_{z_x^{n-2}, z_y^{n-2}}\left(0^y_{n-2}, 0^x_{n-2}; v^y_{n-2}, v^x_{n-2}\right) = \varphi^k_{i_x^{n-2}, i_y^{n-2}}\left(v^y_{n-2}, v^x_{n-2}\right) \quad для\ m < n-1,$$
$$z_*^m = \left\langle u_{n-1}^* 0_{n-2}^* \cdots 0_{m+1}^* v_{m-1}^* v_{m-2}^* \cdots v_1^* v_0^* \right\rangle,$$
$$i_*^m = \left\langle v_{m-1}^* v_{m-2}^* \cdots v_1^* v_0^* \right\rangle.$$

Предварительно, для упорядочивания хранимых функций, должно быть установлено взаимно-однозначное соответствие $k \leftrightarrow \left\langle u^y_{n-1}, u^x_{n-1} \right\rangle$ между порядковым номером функции и значениями переменных u^y_{n-1}, u^x_{n-1}. Считывание памяти выполняется установкой на входе нейронной сети двумерной маски размерностью $p^y_{n-1} \times p^x_{n-1}$, в которой только один из пикселов равен 1, а остальные нули. На рис 3. показана топологическая модель двумерной БНС.

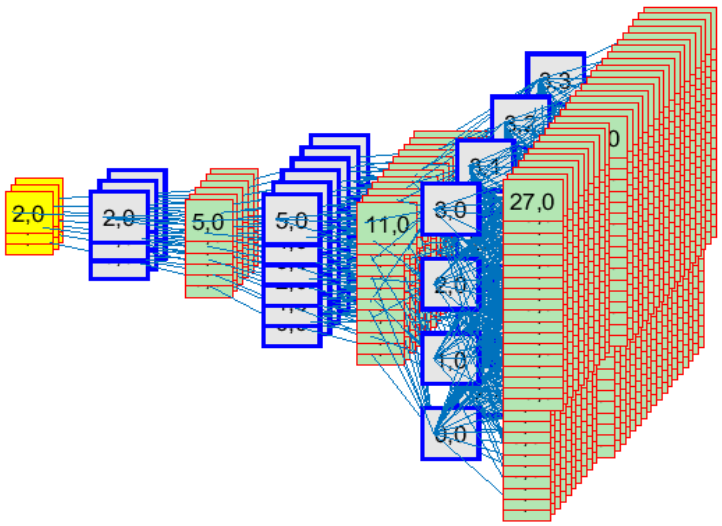

Рис. 3. Топологическая модель двумерной пирамидальной БНС

По входу сеть имеет размерность 3×3, а по выходу 28×28, размерность латентного пространства равна 9. Сеть трёхслойная, структурные характеристики модели определяются размерностями рецепторных и аксоновых полей нейронных ядер по слоям: $p^x = p^y = [1\,1\,3]$, $g^x = g^y = [2\,2\,7]$, символы x и y обозначают пространственные координаты. Сеть представляет собой факторизацию четырёхмерной матрицы размерностью [3×3, 28×28] и способна сохранять 9 произвольных изображений размерностью 28×28.

3. Результаты экспериментов. Эксперименты были проведены на наборе данных MNIST [12]. Набор содержит двумерные образы рукописных цифр от 0 до 9 в виде пиксельных изображений размером 28×28. Объем обучающей выборки равен 60000 образов, тестовая выборка содержит 10000 изображений. На рис. 4 показана выборка с десятью представителями для каждого класса. При проведении экспериментов все примеры выборок предварительно нормировались по энергии к единичному уровню.

Рис. 4. Выборка из набора рукописных цифр

Размерность латентного пространства была выбрана равной 9. Для построения декодера в равной степени можно использовать как одномерную, так и двумерную реализацию сети. Вычисление матрицы декодера $H = \|h(V,U)\|$ удобно производить для одномерного случая, следуя выражению (4). По этой матрице можно непосредственно построить одномерную сеть, используя столбцы, как функции обучающего множества для сети. Одномерная сеть будет иметь размерность 9 по входу и 784=28×28 по выходу. Для перехода к двумерному образу выходной вектор переформатируется к матрице заменой одномерного индекса координат двухразрядным числом $V = \langle V^y, V^x \rangle$, с основаниями разрядов, равными размерностям изображения по осям x и y, в данном случае 28×28.

Для построения двумерной нейронной сети матрицу H следует привести к четырёхмерной форме, это можно сделать представлением номеров строк и столбцов в двухразрядном виде $V = \langle V^y, V^x \rangle$ и $U = \langle u^y_{n-1}, u^x_{n-1} \rangle$. Обучающими образами сети будут являться двумерные плоскости четырёхмерной матрицы, определяемые разрядами u^y_{n-1}, u^x_{n-1}. Реализация двумерной сети показана на рис. 3.

Матрица декодера рассчитывалась по обучающей выборке, состоящей из 30000 примеров в двух вариантах, для оптимального и не оптимального восстановления. На рис. 5a) и 5b) показаны результаты реконструкции образов при подаче на вход декодера случайных векторов латентных переменных. Разница между двумя способами реконструкции визуально не замечена. Для сравнения на рис. 5 (c) показан результат реконструкции, полученной VAE – автоэнкодером, построенным на основе свёрточной нейронной сети, реализованной на платформе МАТЛАБ 2021b [13] для той же обучающей выборки. Очевидно, что генеративная модель на основе свёрточной нейронной сети даёт лучший результат, чем линейный подход

на основе метода PPCA. Тем не менее построенные линейные генеративные модели являются рабочими, а восстанавливаемые образы вполне узнаваемы, при этом время построения генеративной модели PPCA на порядок меньше, чем модели VAE. Увеличение размера обучающей выборки практически не влияет на результаты.

a) не оптимальная реконструкция PPCA

b) оптимальная реконструкция PPCA

c) реконструкция VAE сверточной нейронной сетью

Рис. 5. Результаты реконструкции образов одиночными генеративными моделями

Дальнейшие эксперименты показали, что метод PPCA позволяет получить вполне удовлетворительные результаты при реализации условного множественного декодера (аналог CVAE). В этом случае декодеры PPCA строятся для каждого класса раздельно, при этом используются обучающие примеры конкретного класса. На рис. 6 показаны примеры множественной реконструкции на основе метода PPCA. Каждый столбец в таблице соответствует одной генеративной модели класса, а каждая строка – одной случайной реализации вектора латентных переменных.

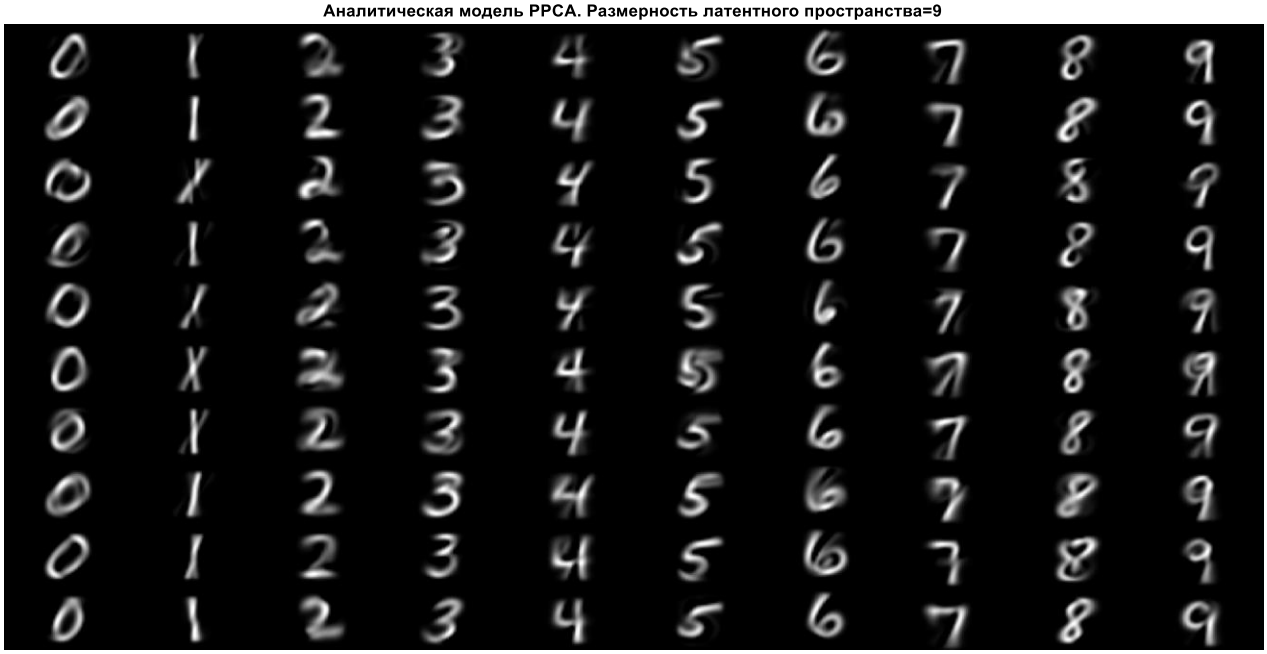

Рис. 6. Результаты реконструкции образов множественными генеративными моделями

Для построения матриц декодеров использовались примерно 6000 примеров для каждого класса.

Заключение. Генеративные модели обладают необычной способностью генерации из случайных векторов реалистических образов, по стилю соответствующих образам обучающей выборки, но не повторяющих их. Эту способность можно использовать, например, для

расширения обучающей выборки при построении нейросетевых классификаторов. Хорошие результаты можно получить при построении генеративных моделей на основе нелинейных нейросетевых VAE и CVAE автоэнкодеров, обучаемых методом обратного распространения ошибки. В данной работе рассмотрен альтернативный вариант построения линейных генеративных моделей на основе вероятностного метода главных компонент (PPCA). В отличие от классических VAE моделей, модель генеративного PPCA-декодера строится аналитически и не требует длительного итерационного обучения, при этом сохраняется приемлемое качество генерации образов. В статье показано, что построенные матрицы декодеров могут быть эффективно реализованы в классе быстрых нейронных сетей. Настройка реализующей БНС выполняется за конечное число шагов, а число вычислительных операций при этом примерно такое же, как и число операции при генерации образа. Топология БНС задаётся аналитической моделью и генерируется автоматически. Генеративная модель для изображений может быть реализована как в одномерном, так и двумерном варианте БНС.

Список источников

1. Doersch C. Tutorial on variational autoencoders, 2016, DOI:10.48550/arXiv.1606.05908.
2. Kingma D.P., Welling M. Auto-encoding variational bayes, 2022, DOI:10.48550/arXiv.1312.6114.
3. Tipping M.E., Bishop Ch.M. Probabilistic principal component analysis. Journal of the Royal statistical society series B: Statistical methodology, 1999, vol. 61, part 3, pp. 611-622, DOI:10.1111/1467-9868.00196
4. Дорогов А.Ю. Самоподобные нейронные сети быстрого обучения / А.Ю. Дорогов. – СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2024. – 188 с. – URL:dorogov.su.
5. Tipping M.E., Bishop C.M. Mixtures of probabilistic principal component analyzers. Neural Computation, no. 11(2), pp. 443-482, available at: http://www.miketipping.com/papers.htm.
6. Christopher M. Bishop. Variational Principal Components. In Proceedings Ninth International Conference on Artificial Neural Networks, ICANN'99, IEE, volume 1, pages 509–514.
7. Бишоп К.М. Распознавание образов и машинное обучение.: Пер. с англ. / К.М. Бишоп. – СПб.: ООО "Диалектика", 2020. – 960 с. – ISBN 978-5-907144-55-2.
8. Дорогов А.Ю. Быстрые преобразования и самоподобные нейронные сети глубокого обучения. Часть 3. Пирамидальные нейронные сети с глубокой степенью обучения / А.Ю. Дорогов // Информационные и математические технологии в науке и управлении, 2024. – № 2(34). – С. 19-32 – DOI:10.25729/ESI.2024.34.2.002.
9. Дорогов А.Ю. Быстрые преобразования и самоподобные нейронные сети глубокого обучения. Часть 1. Стратифицированные модели самоподобных нейронных сетей и быстрых преобразований / А.Ю. Дорогов // Информационные и математические технологии в науке и управлении, 2023. – № 4(32). – С. 5-20. – DOI:10.25729/ESI.2023.32.4.001.
10. Дорогов А.Ю. Быстрые преобразования и самоподобные нейронные сети глубокого обучения. Часть 2. Методы обучения быстрых нейронных сетей / А.Ю. Дорогов // Информационные и математические технологии в науке и управлении. – 2024. – № 1(33). – С. 5-19. – DOI:10.25729/ESI.2024.33.1.001.
11. Дорогов А.Ю. Пластичность самоподобных нейронных сетей / А.Ю. Дорогов // Информационные и математические технологии в науке и управлении, 2024. – № 3(35). – С. 33-43. – DOI:10.25729/ESI.2024.35.3.003.
12. THE MNIST DATABASE of handwritten digits. Available at: http://yann.lecun.com/exdb/mnist/.
13. Документация МАТЛАБ 2021b. – URL: https://docs.exponenta.ru/deeplearning/ug/train-a-variational-autoencoder-vae-to-generate-images.html?searchHighlight=VAE

Дорогов Александр Юрьевич. *Доктор технических наук, доцент, профессор кафедры «Автоматики и процессов управления» Санкт-Петербургского государственного электротехнического университета (СПбГЭТУ) «ЛЭТИ», главный научный сотрудник ПАО «Информационные телекоммуникационные технологии». Основные направления исследований автора связаны с интеллектуальным анализом данных, цифровой обработкой сигналов, проектированием быстрых преобразований и нейронных сетей быстрого обучения, разработкой аналитических платформ, моделированием радио-телекоммуникационных систем. AuthorID: 17611, SPIN: 8645-5873, ORCID: 0000-0002-7596-6761, vaksa2006@yandex.ru, г. Санкт-Петербург, ул. Попова, 5.*

UDC 004.93'12:004.032.26
DOI:10.25729/ESI.2025.39.3.001

Generative models based on pyramid neural networks of fast learning

Alexander Yu. Dorogov

St. Petersburg state electrotechnical university
PJSC "Information telecommunication technologies" ("Inteltech"),
Russia, St. Petersburg, *vaksa2006@yandex.ru*

Abstract. The article proposes a method for constructing generative models based on pyramid neural networks of fast learning (FNN). The model construction is based on the probabilistic principal component method (PPCA). The PPCA method allows us to analytically construct matrices of optimal decoders that are capable to reconstructing images from random latent variables of small dimension distributed according to a normal probability law. The implementation of PPCA- decoders in the FNN class makes it possible to represent decoders in the form of series-parallel structures that provide high performance due to the structural parallelization of operations. The paper presents methods for teaching FNN to decoder matrices. Training is performed in a finite number of steps and does not require iterative procedures. Examples of constructing implementing FNN for the MNIST dataset are given. The results of generating images similar to the MNIST set are shown. A comparison with the classical variational autoencoder has been performed. The field of expedient use of generative models of PPCA is defined.

Keywords: generative model, variational autoencoder, probabilistic principal component method, fast neural network, pyramid structure

References

1. Doersch C. Tutorial on variational autoencoders, 2016, DOI:10.48550/arXiv.1606.05908.
2. Kingma D.P., Welling M. Auto-encoding variational bayes, 2022, DOI:10.48550/arXiv.1312.6114.
3. Tipping M.E., Bishop Ch.M. Probabilistic principal component analysis. Journal of the Royal statistical society series B: Statistical methodology, 1999, vol. 61, part 3, pp. 611-622, DOI:10.1111/1467-9868.00196
4. Dorogov A.Yu. Samopodobnyye neyronnyye seti bystrogo obucheniya [Self-similar neural networks of fast learning]. SPb.: Izd-vo SPbGETU "LETI" [St. Petersburg: Publishing house of ETU "LETI"], 2024, 188 p. available at: dorogov.su.
5. Tipping M.E., Bishop C.M. Mixtures of probabilistic principal component analyzers. Neural Computation, no. 11(2), pp. 443-482, available at: http://www.miketipping.com/papers.htm.
6. Christopher M. Bishop. Variational Principal Components. In Proceedings Ninth International Conference on Artificial Neural Networks, ICANN'99, IEE, volume 1, pages 509–514.
7. Bishop K.M. Raspoznavaniye obrazov i mashinnoye obucheniye.: Per. s angl. [Pattern recognition and machine learning: Trans. from English]. SPb.: OOO "Dialektika" [St. Petersburg: OOO Dialectika], 2020, 960 p., ISBN 978-5-907144-55-2.
8. Dorogov A.Yu. Bystrye preobrazovaniya i samopodobnye neyronnye seti glubokogo obucheniya Chast' 3. Piramidal'nye neyronnye seti s glubokoy stepen'yu obucheniya [Fast transformations and self-similar deep learning neural networks Part 3. Pyramid neural networks with a deep learning degree]. Informacionnye i matematiceskie tehnologii v nauke i upravlenii [Information and mathematical technologies in science and management], 2024, no. 3(35), pp. 33-43, DOI: 10.25729/ESI.2024.34.2.002.
9. Dorogov A.Yu. Bystrye preobrazovaniya i samopodobnye neyronnye seti glubokogo obucheniya. Chast' 1. Stratifitsirovannye modeli samopodobnykh neyronnykh setey i bystrykh preobrazovaniy [Fast transformations and self-similar deep learning neural networks. Part 1. Stratified models of self-similar neural networks and fast transformations Informacionnye i matematiceskie tehnologii v nauke i upravlenii [Information and mathematical technologies in science and management], 2023, no. 4(32), pp. 5-20, DOI: 10.25729/ESI.2023.32.4.001.
10. Dorogov A.Yu. Bystrye preobrazovaniya i samopodobnye neyronnye seti glubokogo obucheniya. Chast' 2. Metody obucheniya bystrykh neyronnykh setey [Fast transformations and self-similar neural networks of deep learning. Part 2. Methods of training fast neural networks]. Informacionnye i matematiceskie tehnologii v nauke i upravlenii [Information and mathematical technologies in science and management], 2024, no. 1(33), pp. 5-19, DOI:10.25729/ESI.2024.33.1.001.
11. Dorogov A.Yu. Plastichnost' samopodobnyh nejronnyh setej [Plasticity of self-similar neural networks]. Informacionnye i matematiceskie tehnologii v nauke i upravlenii [Information and mathematical technologies in science and management], 2024, no 3(35), pp. 33-43. DOI:10.25729/ESI.2024.35.3.003.
12. THE MNIST DATABASE of handwritten digits. Available at: http://yann.lecun.com/exdb/mnist/.

13. MATLAB 2021b documentation. Available at: https://docs.exponenta.ru/deeplearning/ug/train-a-variational-autoencoder-vae-to-generate-images.html?searchHighlight=VAE.

Dorogov Alexander Yurievich. *Doctor of technical sciences, associate professor, professor of the department of automation and control Processes of St. Petersburg state electrotechnical university (SPbETU) "LETI", Chief researcher of PJSC "Information telecommunication technologies". The main directions of the author's research are related to data mining, digital signal processing, the design of fast transformations and neural networks of fast learning, the development of analytical platforms, modeling of radio and telecommunications systems. AuthorID: 17611, SPIN: 8645-5873, ORCID: 0000-0002-7596-6761, vaksa2006@yandex.ru, St. Petersburg, st. Popova, 5.*

Статья поступила в редакцию 28.04.2025; одобрена после рецензирования 02.06.2025; принята к публикации 02.08.2025.

The article was submitted 04/28/2025; approved after reviewing 06/02/2025; accepted for publication 08/02/2025.