Московский государственный университет
имени М. В. Ломоносова
Факультет Вычислительной Математики и Кибернетики
Кафедра Математических Методов Прогнозирования

# Отчет о выполнении задания №5 по практикуму на ЭВМ.

Выполнил студент 317 группы
*Гарипов Тимур Исмагилевич*

Москва, 24 февраля 2016 г.

# Содержание

# 1 Описание задания

Цель задания состоит в том, чтобы реализовать обучение разреженного автокодировщика и показать, как он обнаруживает, что границы объектов и цветовые переходы одно из лучших представлений для естественных изображений.

# 2 Нейросетевой разреженный автокодировщик

Рассмотрим устройство исследуемого автокодировщика и выведем формулы для вычисления градиентов в методе обратного распространения ошибки. Так как для ускорения работы программ требуется использовать векторизацию и матричные вычисления, все приведенные ниже формулы будут записаны в матричном виде.

## 2.1 Архитектура сети

На приведенном ниже рисунке схематично изображен автокодировщик, имеющий $n$ скрытых слоёв с номерами $0, 1, \ldots, n-1$ (предполагается, что $n$ нечётно).

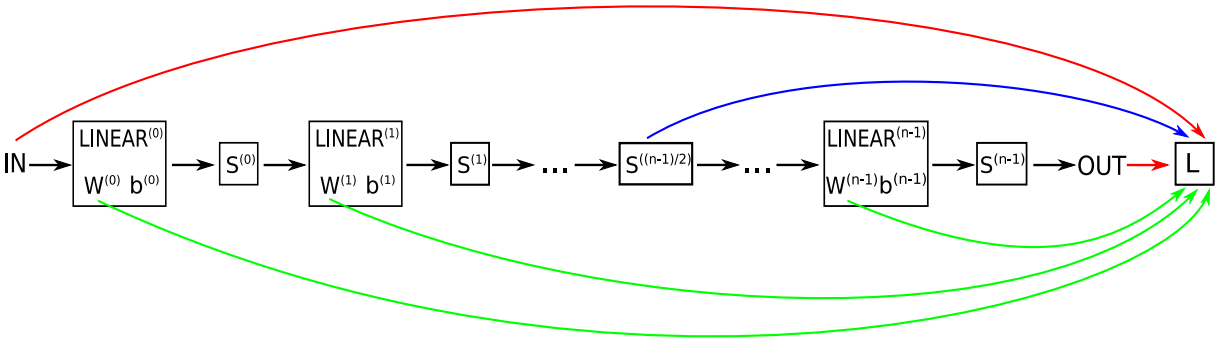



**Рис. 1:** Схема автокодировщика

На вход нейросети ($IN$) подается набор из $N$ изображений размера $d \times d \times 3$ в виде матрицы размера $N \times D$, где $D = d \times d \times 3$. Каждая строка этой матрицы представляет собой очередное изображение из набора, "вытянутое" вектор длины $D$.

Выходом сети ($OUT$) также является набор из $N$ изображений с такими же размерами. Ожидается, что выход обученного автокодировщика будет приблизительно равен входу.

Преобразование входной матрицы в выходную осуществляется последовательным применением линейных (LINEAR) и нелинейных (S) преобразований. То есть нейросеть можно рассматривать как последовательность слоёв, каждый из которых выполняет некоторое преобразование.

Пусть на вход линейному слою подается матрица $X \in \mathbb{R}^{N \times D_{in}}$, тогда его выходом является матрица $Y \in \mathbb{R}^{N \times D_{out}}$, которая может быть вычислена по формуле: $Y = XW + b$, где $W \in \mathbb{R}^{D_{in} \times D_{out}}$ и $b \in \mathbb{R}^{D_{out}}$ — параметры линейного слоя. Под сложением матрицы и вектора понимается поэлементное сложение после предварительного broadcating'а вектора $b$ к размерности $N \times D_{out}$.

Нелинейный слой поэлементно применяет заданую функцию активации ко всем элементам входной матрицы. $Y_{ij} = f(X_{ij})$. В качестве функции активаци рассматривается сигмоидальная функция $f(z) = \dfrac{1}{1+e^{-z}}$.

По выходу среднего нелинейного слоя в обученном автокодировщике может быть приблизительно восстановлен вход, это свойство позволяет использовать этот выход как признаковое описание входных изображений.

## 2.2 Функция потерь

Обучение автокодировщика осуществляется путем минимизации функции потерь $L$, состоящей из трёх слагаемых: $L = J + R_\lambda + R_{\rho,\beta}$.

Первое слагаемое $J$ штрафует отклонение выходных изображений от входных и вычисляется следующим образом:

$$J = \frac{1}{2}\sum_{i=1}^{N} ||IN_i - OUT_i||^2 = \frac{1}{2}||IN - OUT||^2,$$

в последней записи под нормой матрицы подразумевается норма фробениуса.

Второе слагаемое представляет собой L2 регуляризатор на веса (элемнты матриц $W^{(i)}$):

$$R_\lambda = \frac{\lambda}{2}\sum_{i=0}^{n-1} ||W^{(i)}||^2,$$

число $\lambda$— параметр регуляризации.

Третье слагаемое есть регуляризатор разреженности активаций среднего скрытого слоя.

$$R_{\rho,\beta} = \beta \sum_j \left[\rho \log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j}\right],$$

здесь числа $\rho$ и $\beta$ являются параметрами регуляризации, $\hat{\rho}_j$ вычисляется как среднее значении активации $j$-ого нейрона в среднем слое по всем объектам.

## 2.3 Вычисление градиентов

Рассмотрим произвольный слой нейросети, который вычилсяет выходную матрицу $Y$ по входной матрице $X$. Для применения метода обратного распространения ошибки необходимо вычислить градиент функции потерь $\dfrac{\partial L}{\partial X}$, зная градиент $\dfrac{\partial L}{\partial Y}$. Кроме того требуется вычислить градиенты по параметрам слоя.

#### Линейное преобразование

Получим требуемые формулы для линейного слоя.

$$Y_{ij} = \sum_{k=1}^{D_{in}} X_{ik} W_{kj} + b_j$$

$$\frac{\partial L}{\partial X_{st}} = \sum_{i=1}^{N} \sum_{j=1}^{D_{out}} \frac{\partial L}{\partial Y_{ij}} \frac{\partial Y_{ij}}{\partial X_{st}} = \left\{ \frac{\partial Y_{ij}}{\partial X_{st}} = 0 \ \text{ при i} \neq \text{s} \right\} = \sum_{j=1}^{D_{out}} \frac{\partial L}{\partial Y_{sj}} \frac{\partial Y_{sj}}{\partial X_{st}} = \sum_{j=1}^{D_{out}} \frac{\partial L}{\partial Y_{sj}} W_{tj} = \left( \frac{\partial L}{\partial Y} W^T \right)_{st}.$$

Далее вычислим градиенты функции потерь по параметрам $W$ и $b$:

$$\frac{\partial L}{\partial W_{st}} = \sum_{i=1}^{N} \sum_{j=1}^{D_{out}} \frac{\partial L}{\partial Y_{ij}} \frac{\partial Y_{ij}}{\partial W_{st}} = \left\{ \frac{\partial Y_{ij}}{\partial X_{st}} = 0 \ \text{ при j} \neq \text{t} \right\} = \sum_{i=1}^{N} \frac{\partial L}{\partial Y_{it}} \frac{\partial Y_{it}}{\partial W_{st}} = \sum_{i=1}^{N} \frac{\partial L}{\partial Y_{it}} X_{is} = \left( X^T \frac{\partial L}{\partial Y} \right)_{st},$$

$$\frac{\partial L}{\partial b_s} = \sum_{i=1}^{N} \sum_{j=1}^{D_{out}} \frac{\partial L}{\partial Y_{ij}} \frac{\partial Y_{ij}}{\partial b_s} = \left\{ \frac{\partial Y_{ij}}{\partial b_s} = 0 \ \text{ при j} \neq \text{s} \right\} = \sum_{i=1}^{N} \frac{\partial L}{\partial Y_{is}} \frac{\partial Y_{is}}{\partial b_s} = \sum_{i=1}^{N} \frac{\partial L}{\partial Y_{is}}.$$

**Функция активации**

Для нелинейного слоя матрицы $X$ и $Y$ связаны соотношением.

$$Y_{ij} = f(X_{ij})$$

А формулы для пересчета градиентов принимают следующий вид:

$$\frac{\partial L}{\partial X_{ij}} = \frac{\partial L}{\partial Y_{ij}} \frac{\partial Y_{ij}}{\partial X_{ij}} = \frac{\partial L}{\partial Y_{ij}} f'(X_{ij})$$

В случае сигмоидальной функции $f$ имеем:

$$\begin{aligned} \frac{\partial L}{\partial X_{ij}} &= \frac{\partial L}{\partial Y_{ij}} \left( \frac{1}{1+e^{-X_{ij}}} \right)' = \frac{\partial L}{\partial Y_{ij}} \frac{-1}{(1+e^{-X_{ij}})^2} e^{-X_{ij}} (-1) = \\ &= \left\{ Y_{ij} = \frac{1}{1+e^{-X_{ij}}}, \quad e^{-X_{ij}} = \frac{1}{Y_{ij}} - 1 \right\} = \frac{\partial L}{\partial Y_{ij}} Y_{ij}^2 (\frac{1}{Y_{ij}} - 1) = \frac{\partial L}{\partial Y_{ij}} Y_{ij}(1 - Y_{ij}) \end{aligned}$$

**Обратное распространение ошибки**

Мы получили формулы, которые позволяют для любого слоя вычилсить градиент функции потерь по входу этого слоя $\dfrac{\partial L}{\partial X}$ при известном градиенте по выходу слоя $\dfrac{\partial L}{\partial Y}$. При этом также можно вычислить градиент $L$ по параметрам слоя (если они имеются). Осталось вычилсить градиент $L$ по выходу последнего слоя, после чего можно вычилсять градиенты для предыдущих слоёв.

$$L = L(IN, OUT, W^{(0)}, W^{(1)}, \dots, W^{(n-1)}, S^{(\frac{n-1}{2})}) =$$
$$= J(IN, OUT) + R_\lambda(W^{(0)}, W^{(1)}, \dots, W^{(n-1)}) + R_{\rho,\beta}(S^{(\frac{n-1}{2})})$$

$$J(IN, OUT) = \frac{1}{2}||IN - OUT||^2,$$

$$R_\lambda(W^{(0)}, W^{(1)}, \dots, W^{(n-1)}) = \frac{\lambda}{2}\sum_{i=0}^{n-1}||W^{(i)}||^2,$$

$$R_{\rho,\beta}(S^{(\frac{n-1}{2})}) = \beta\sum_j\left[\rho\log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j}\right], \qquad \hat{\rho}_j = \frac{1}{N}\sum_{i=1}^{N}S_{ij}^{(\frac{n-1}{2})}$$

$$\frac{\partial L}{\partial OUT} = \frac{\partial J}{\partial OUT} = OUT - IN$$

После того как градиент по $OUT$ вычислен, можно вычислять градиент для предыдущих слоёв по описанным выше формулам.

Но кроме того надо учесть градиенты, которые появляются из-за регуляризаторов. Рассмотрим, как матрица линейного преобразования $W^{(i)}$ влияет на $L$. Во-первых от этой матрицы зависит значения выхода сети $OUT = OUT(IN, W^{(i)}, \dots)$, во-вторых эта матрица вносит вклад в регуляризатор $R_\lambda$. Эти соображения приводят нас к следующей формуле:

$$\frac{\partial L}{\partial W^{(i)}} = \frac{\partial L}{\partial OUT}\frac{\partial OUT}{\partial W^{(i)}} + \frac{\partial R_\lambda}{\partial W^{(i)}},$$

здесь первое слагаемое может быть вычислено при обратном распространении ошибки (с использованием уже упоминавшихся формул), а второе слагаемое вычилсяется следующим образом:

$$\frac{\partial R_\lambda}{\partial W^{(i)}} = \lambda W^{(i)}$$

.

Перейдем ко второму регуляризатору, введем обозначение $k = \frac{n-1}{2}$. С помощью аналогичных рассуждений можно прийти к формуле:

$$\frac{\partial L}{\partial S^{(k)}} = \frac{\partial L}{\partial OUT}\frac{\partial OUT}{\partial S^{(k)}} + \frac{\partial R_{\rho,\beta}}{\partial S^{(k)}}.$$

Вычисление первого слагаемого опять может быть выполнено при обратном распространении ошибки, получим выражение для второго слагаемого:

$$\frac{\partial R_{\rho,\beta}}{\partial S_{st}^{(k)}} = \sum_j \frac{\partial R_{\rho,\beta}}{\partial \hat{\rho}_j} \frac{\partial \hat{\rho}_j}{S_{st}^{(k)}} = \left\{ \frac{\partial \hat{\rho}_j}{S_{st}^{(k)}} = 0, \text{ при } j \neq t \right\} = \frac{\partial R_{\rho,\beta}}{\partial \hat{\rho}_t} \frac{\partial \hat{\rho}_t}{S_{st}^{(k)}} = \frac{1}{N} \frac{\partial R_{\rho,\beta}}{\partial \hat{\rho}_t} = \frac{\beta}{N} \left( -\frac{\hat{\rho}_t}{\rho} + \frac{1-\hat{\rho}_t}{1-\rho} \right)$$

# 3 Результаты

## 3.1 Обученные фильтры

На рис. (2) изображены обученые фильтры. Изображение (2(a)) соответствует подобранным гиперпараметрам, полученные таким образом фильтры как и ожидалось детектируют границы объектов и цветовые переходы. Остальные изображения показывают, что при изменении гиперпараметров визульное качество фильтров ухудшается (особенно это заметно при изменении параметра $\lambda$).

## 3.2 Обучение классификаторов на данных сокращённой размерности

Для классификации изображений из датасета STL-10 использовались алгоритмы из библиотеки scikit-learn: RandomForestClassifier (**RF**) и LogisticRegression (**LR**). В таблице (1) показано качество этих кслассификаторов на валидационной выборке.

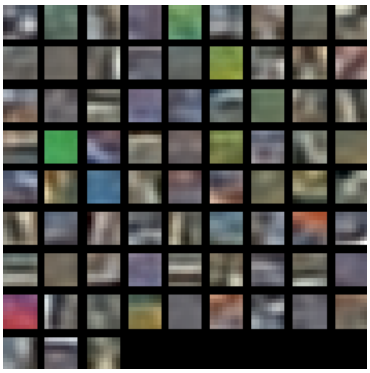
| Метод генерации признаков | **RF** | **LR** |
|---|---|---|
| Интенсиности цветовых каналов | **43.76** | 34.01 |
| Однослойных автокодировщик (выбор патчей с шагом 8) | 42.41 | 47.56 |
| Однослойных автокодировщик (выбор патчей с шагом 4) | 43.43 | **50.58** |
| Трёхслойный автокодировщик (выбор патчей с шагом 8) | 40.02 | 41.28 |

**Таблица 1:** Точность обученных классификаторов (%)

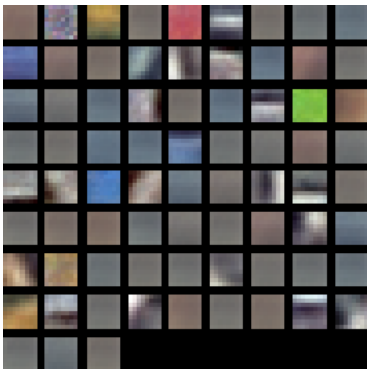
Полученные результаты позволяют заключить, что признаки, сгенерированные с помощью автокодировщика позволяют повысить качесвто классификации. Об этом свидетельствует прирост качества классификации с помощью логистической регрессии, при использовании автокодировщика. Случайные леса по каким-то причинам не прибавили в качестве. Автокодировщик с тремя скрытыми слоями работает хуже, чем однослойный, возможно требуется более точный подюор параметров сети.

# 4 Выводы

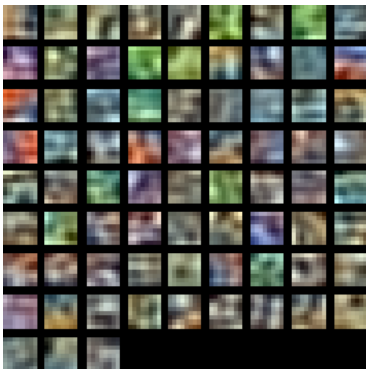
В ходе выполнения работы реализовано обучение разреженного автокодировщика и показано, как он обнаруживает, что границы объектов и цветовые переходы одно из лучших представлений для естественных изображений. Проведена классификация изображений из датасета STL-10 с использованием парадигмы предобучения без учителя.



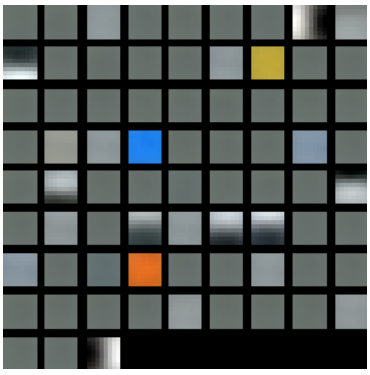
**(a)** $\rho = 0.06,\ \lambda = 10^{-4},\ \beta = 3$



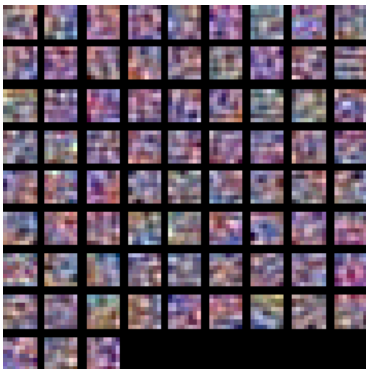
**(b)** $\rho = 0.01,\ \lambda = 10^{-4},\ \beta = 3$



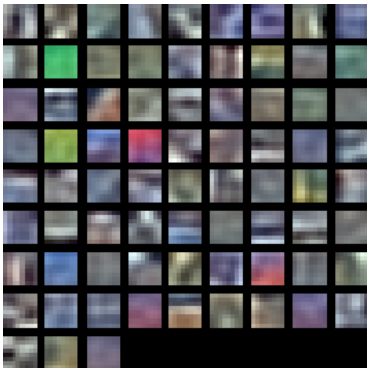
**(c)** $\rho = 0.12,\ \lambda = 10^{-4},\ \beta = 3$



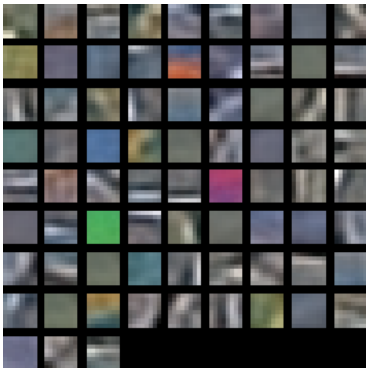
**(d)** $\rho = 0.06,\ \lambda = 10^{-3},\ \beta = 3$



**(e)** $\rho = 0.06,\ \lambda = 10^{-5},\ \beta = 3$



**(f)** $\rho = 0.06,\ \lambda = 10^{-4},\ \beta = 6$



**(g)** $\rho = 0.06,\ \lambda = 10^{-4},\ \beta = 0.5$

**Рис. 2:** Обученные фильтры, полуенные при различных значениях гиперпараметров