# Как я провел лето. Deep Learning.

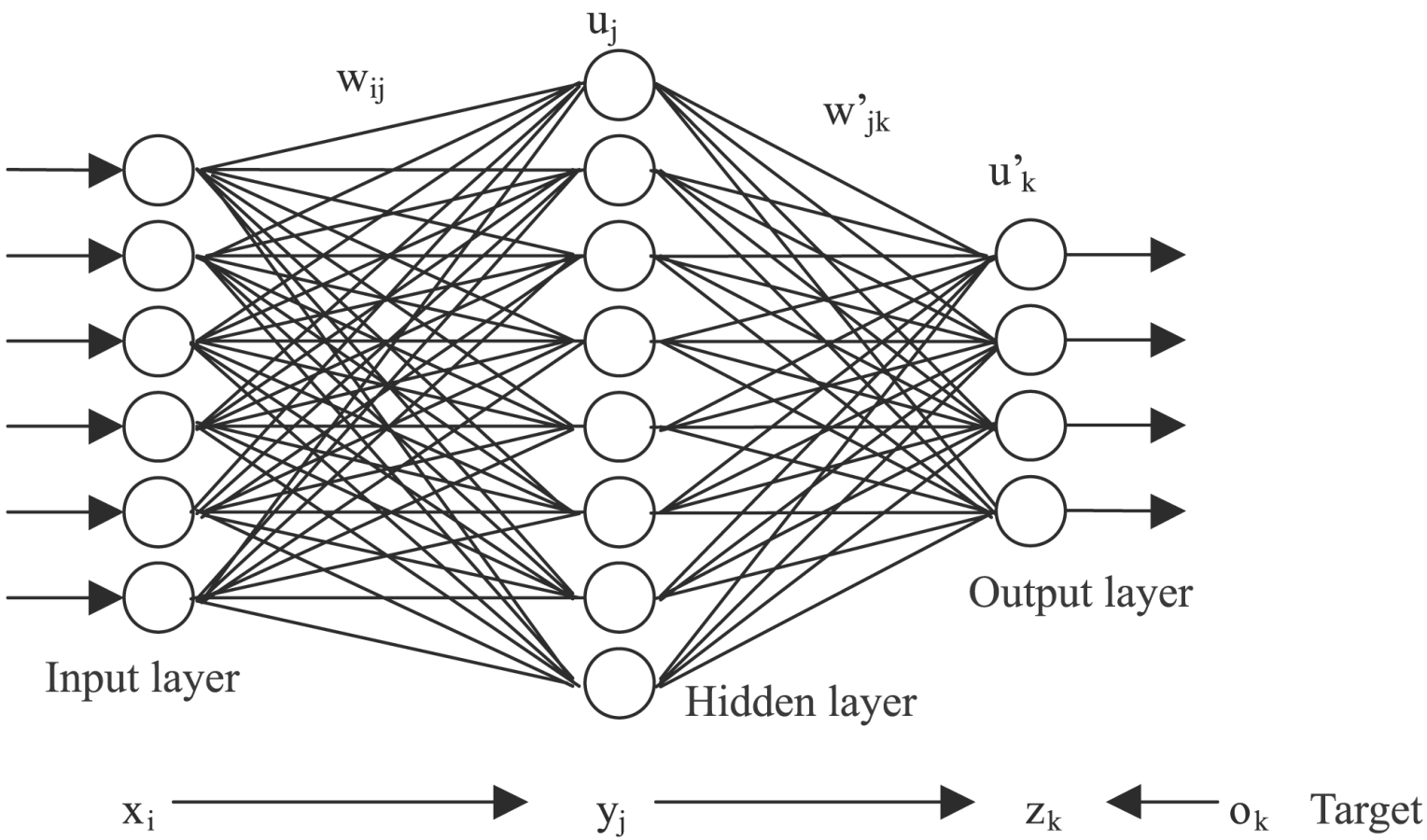
Олег Харациди

ВМК МГУ

28 октября 2013

# Содержание

# Backpropagation (Rumelhart, Hinton, Williams, 1985)



## Почему deep?

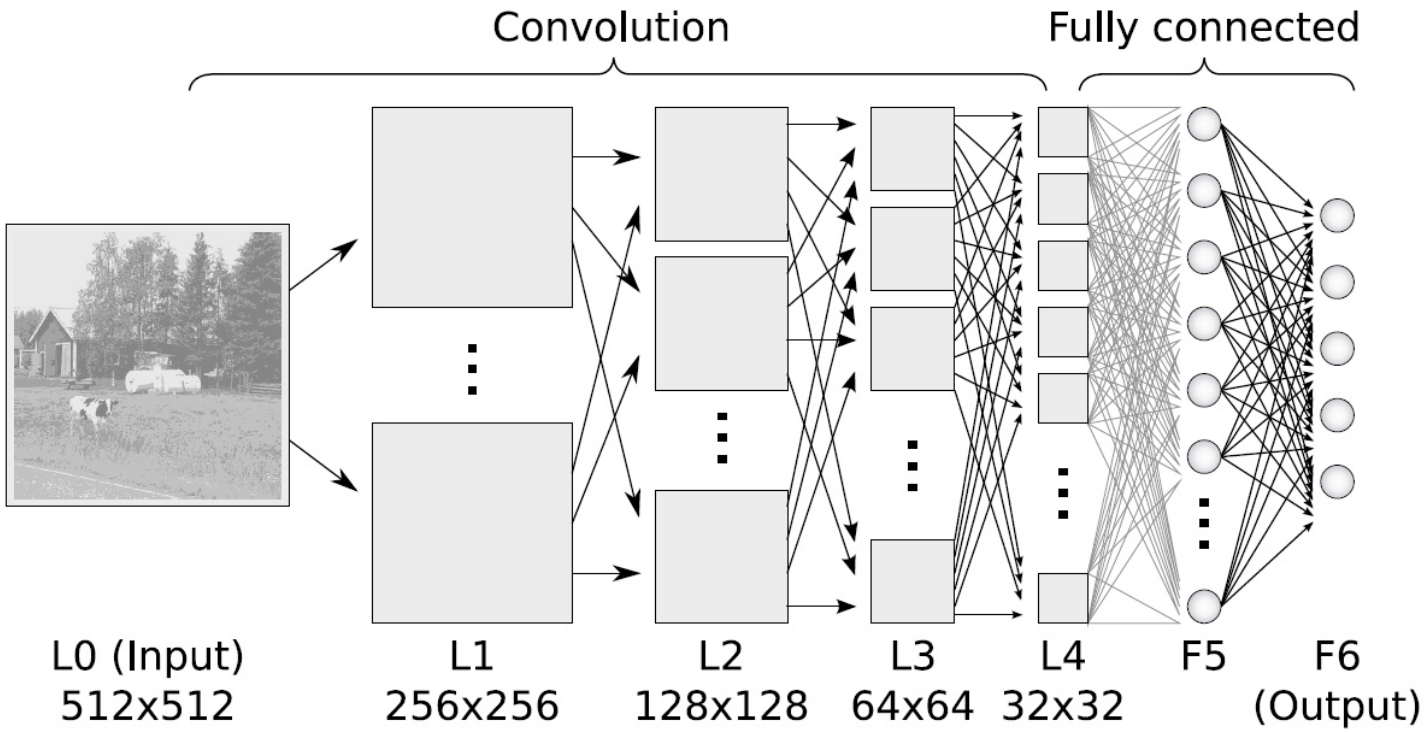
- Построение сложных функций зависимости (разделяющих поверхностей)
- Построение сложных признаков (в т.ч. feature learning)

## Проблемы нейросетей

- Большое количество совместно настраиваемых параметров
- Невыпуклая задача оптимизации
- «Паралич» сети
- Сложность вычислений
- Медленное обучение первого слоя
- Сильная чувствительность к инициализации весов

# Сверточные нейросети

Соединяем каждый нейрон только с «соседними» нейронами предыдущего слоя.




*Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, 1989*

## Особенности сверточного слоя

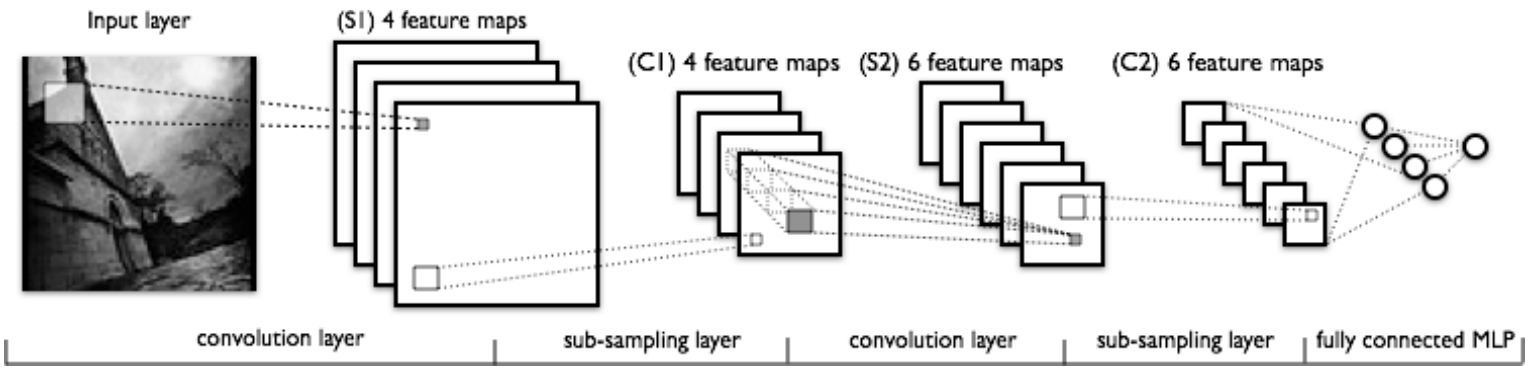
- Каждый нейрон на выходе сверточного слоя – фильтр.
- Случайная свертка – обычно детектор границ.
- Допускается эвристика: сделать все фильтры равными.

# Архитектура LeNet

Чередование сверточных и max-pooling (subsampling, downsampling) слоев.

## Max-pooling layer

Разобьем текущий слой на несколько *непересекающихся* блоков и для каждого выберем максимальное значение.



## Особенности LeNet

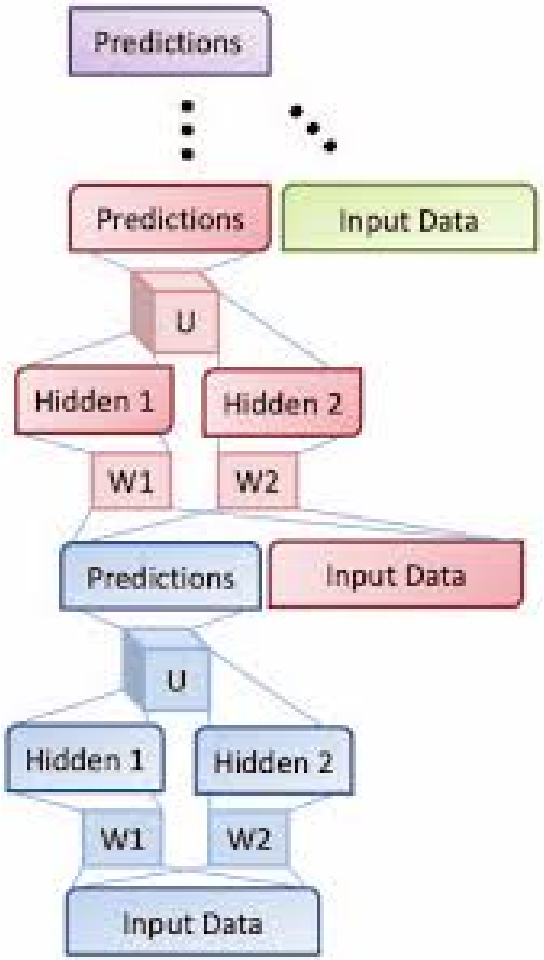
- Сравнительно небольшое количество параметров
- Относительно устойчивы к переобучению
- Подходит для распознавания объектов на картинке, инварианты относительно его сдвига.
- Рекорд на MNIST – 23 ошибки из 10000.
  Композиция из 35 LeNet с архитектурой
  $29x29 - 20C4 - MP2 - 40C5 - MP3 - 150N - 10N$
- *Не инвариантна относительно перестановки признаков*

# Другие эвристики

## Optimal Brain Damage (LeCun, 1990)

Ускорение стадии вывода (inference) сети.
После попадания в локальный минимум удаляем ребра с наименьшей по модулю второй производной и продолжаем оптимизацию.

# Deep Stacking Network (Deng, Yu, Platt, 2012)




Обучение DSN:

- Двухслойные блоки обучаются последовательно.
- Каждый из двух слоев обучается отдельно.
- После этого запускается backpropagation

Преимущества:

- Выпуклая оптимизация.
- Низкая склонность к переобучению.
- Хорошо распараллеливается.

А также:

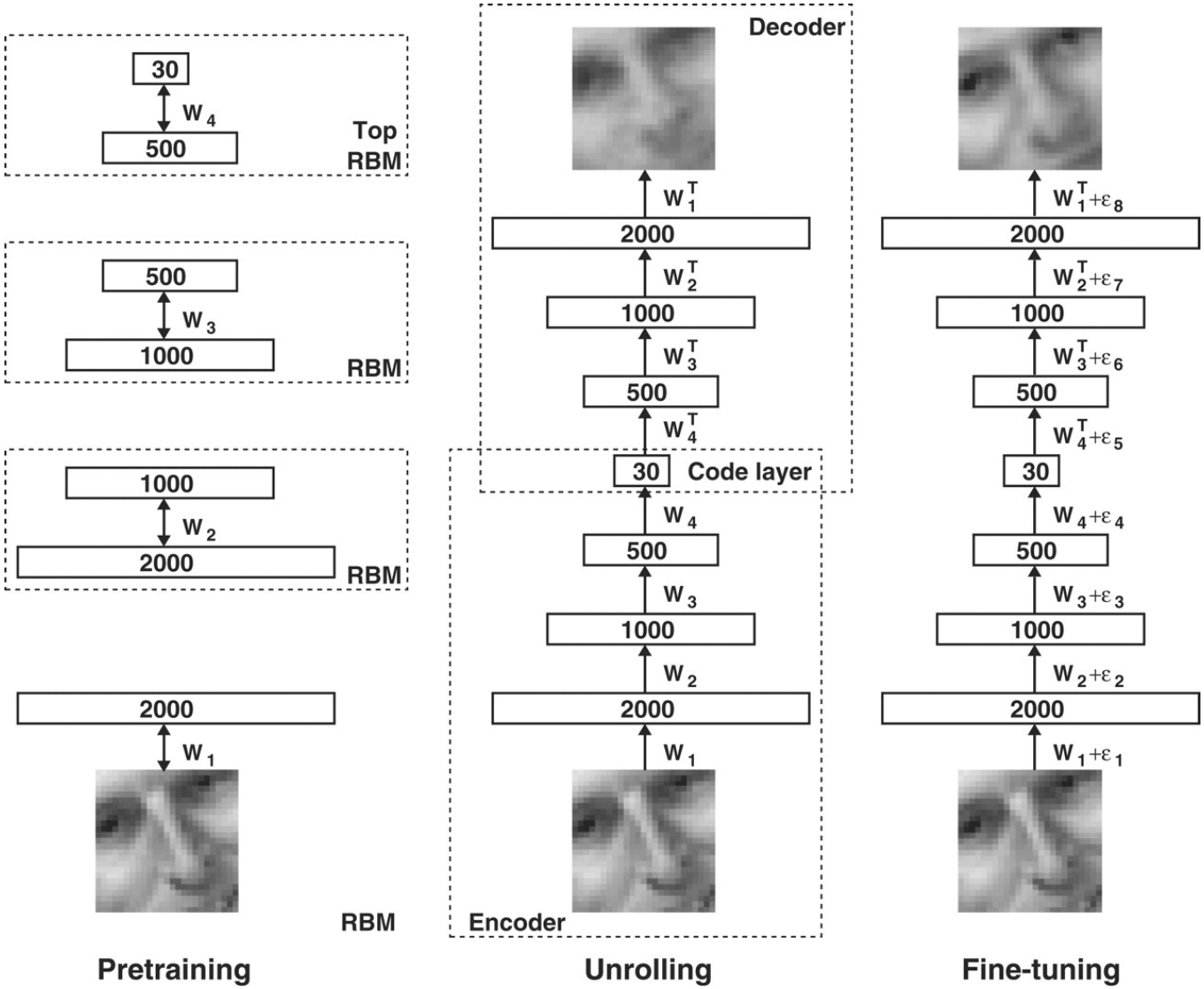
- Прореживание на стадии предобучения
- Нормализация входящих весов каждого нейрона

# Deep autoencoder

Алгоритм:

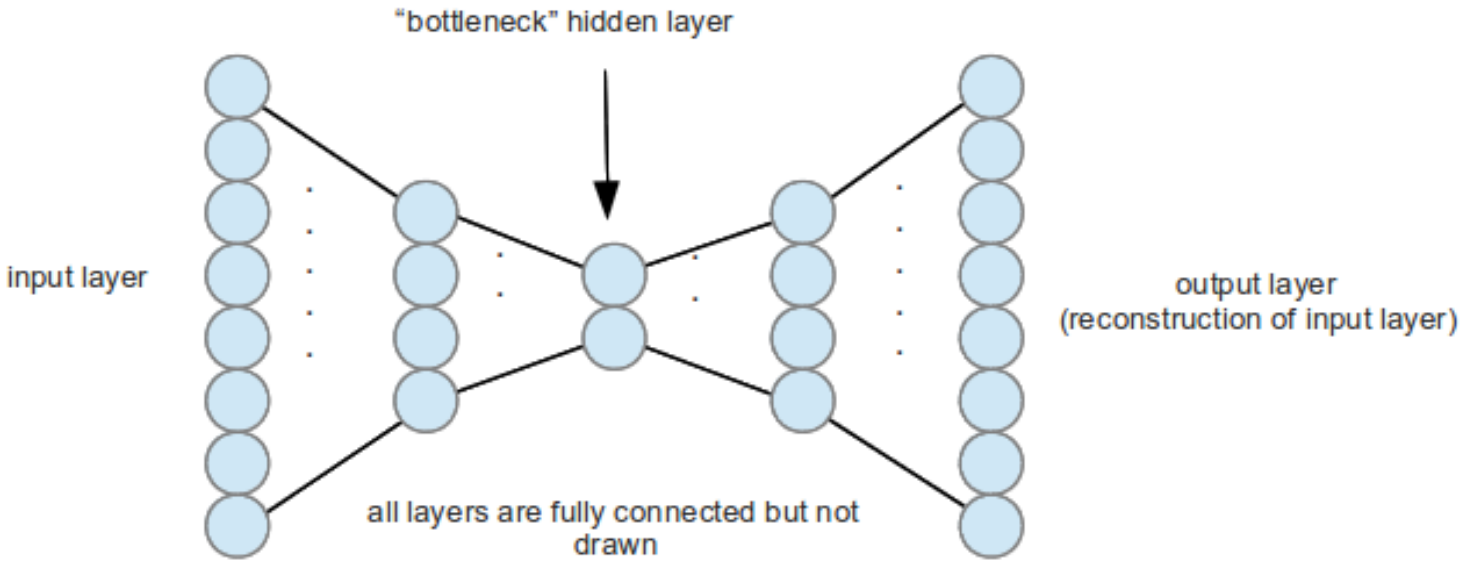
- Предобучение по слоям с помощью RBM
- Симметричное отражение полученной сети
- Обучение без учителя с помощью backpropagation (симметрия весов нарушается)

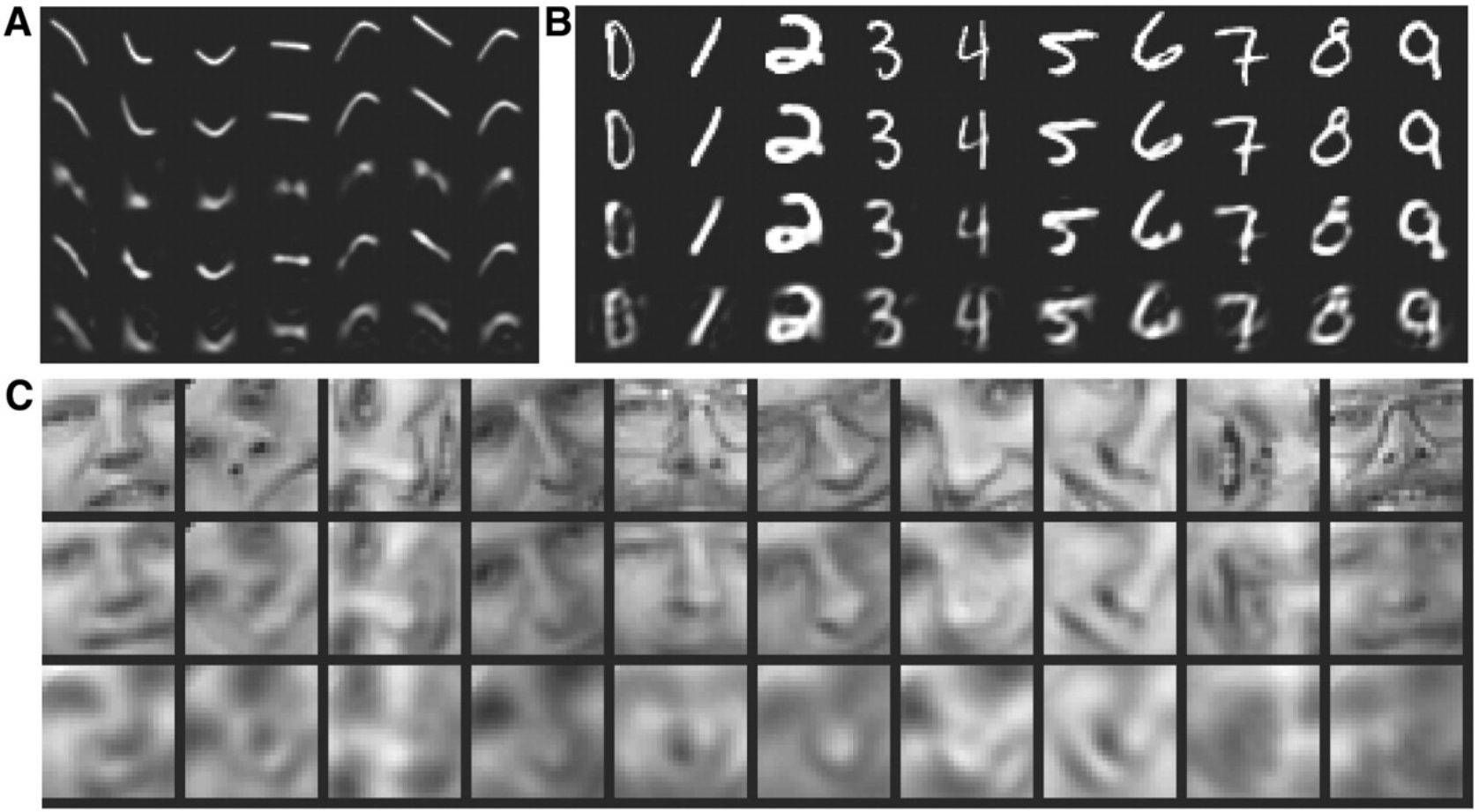
*Hinton, Salakhutdinov, 2006*


30
W 4
500
Top RBM
500
W 3
1000
RBM
1000
W 2
2000
RBM
2000
W 1
RBM
Pretraining
Decoder
W T 1
2000
W T 2
1000
W T 3
500
W T 4
30
Code layer
W 4
500
W 3
1000
W 2
2000
W 1
Encoder
Unrolling
W T 1+ε8
2000
W T 2+ε7
1000
W T 3+ε6
500
W T 4+ε5
30
W 4+ε4
500
W 3+ε3
1000
W 2+ε2
2000
W 1+ε1
Fine-tuning

## Особенности autoencoder-a

- Нелинейное сжатие данных
- Генерация сложных высокоуровневых признаков (feature learning)






- (1) исходные объекты
- (2) сжатие с помощью Autoencoder-a
- (3-4) сжатие с помощью PCA

## Deep Belief Net

- Снова удаляем скопированную часть сети
- Добавляем выходной слой
- Запускаем обучение с учителем (снова backpropagation)

## Deep Boltzman Machine (Salakhutdinov, Hinton, 2009)

DBM иногда используется на стадии предобучения вместо Autoencoder-a.

Основное отличие от RBM – многослойная архитектура.

# Dropout

Простая эвристика: на каждой итерации backpropagation «выкидываем» половину нейронов скрытых слоев, вместе с их входящими и исходящими весами, а после завершения итерации – возвращаем.
После окончания обучения делим все веса пополам.

*Hinton et al., 2012*

## MNIST

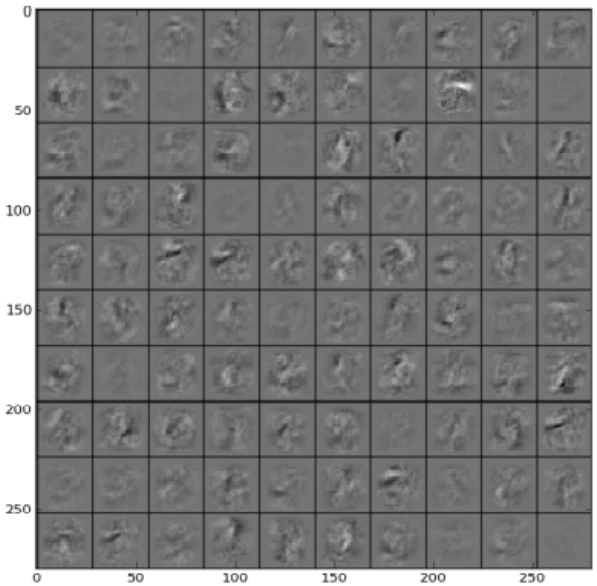
Предыдущий результат: 160 ошибок

DBN: 118 ошибок

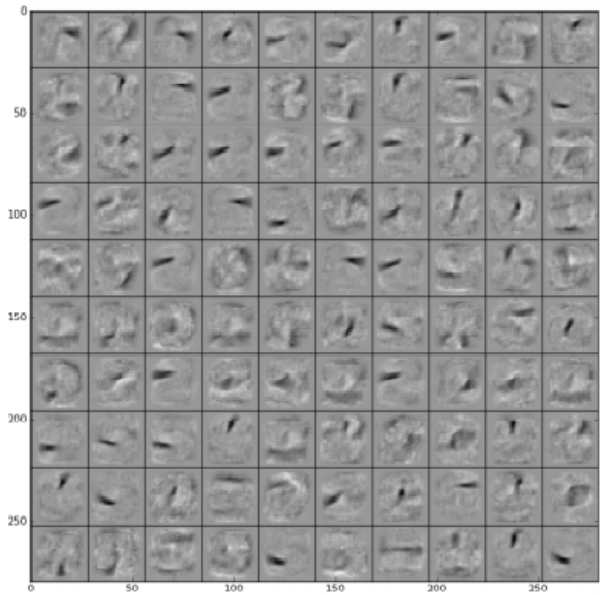
Dropout DBN: 92 ошибки

DBM + backpropagation: 94 ошибки

DBM + dropout backpropagation: 79 ошибок




(a)



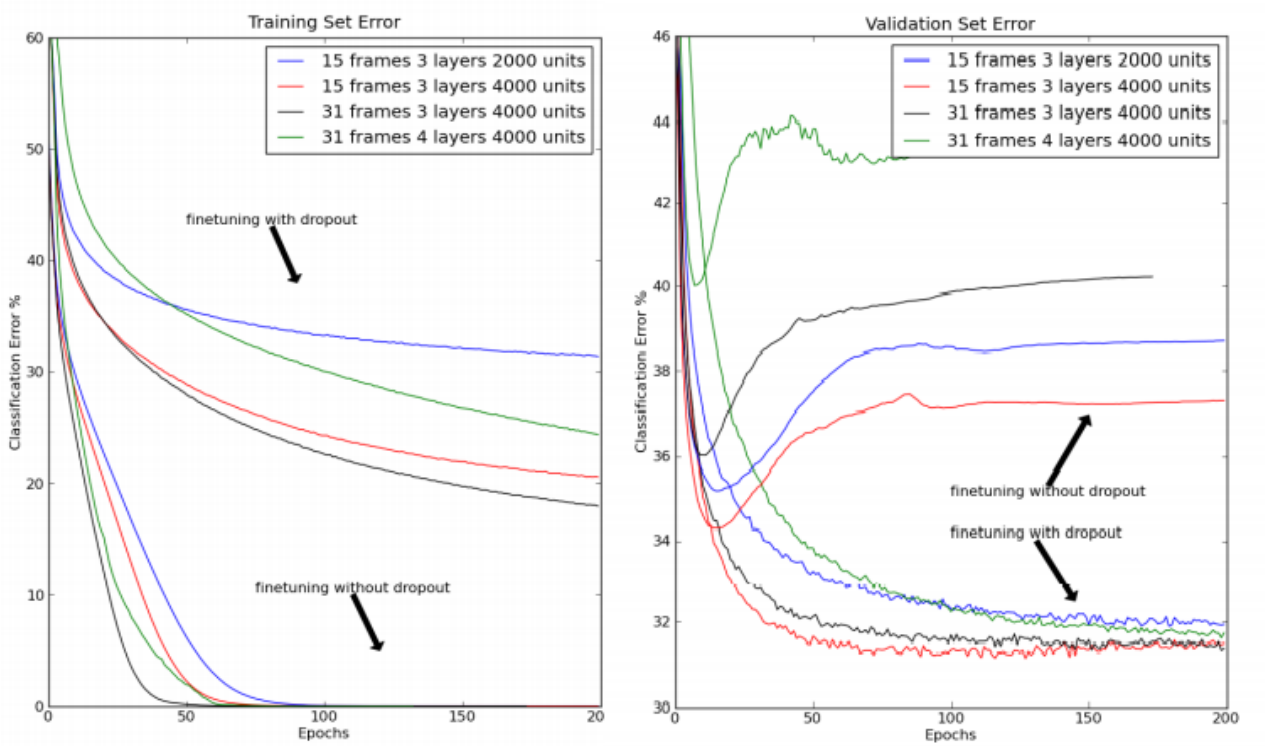

(b)

Признаки первого скрытого слоя для MNIST: (a) без dropout (b) с dropout

# TIMIT

TIMIT – бенчмарк по распознаванию речи.

Dropout уменьшает долю ошибок классификации с 22.7% до 19.7%




TIMIT

## ImageNet

ImageNet – набор из $\sim$ 1.3млн картинок из интернета и 1000 тэгов.

Dropout снижает долю ошибок с 48.6% до 42.4%.

Архитектура сети

$$(224x224) - C - MP - C - MP - C - C - C - MP - \underbrace{F(4096) - F(4096)}_{dropout} - (1000)$$

# Kaggle Job Salary Prediction

1 и 2 место – нейросети.
Лучший результат: нейросеть с архитектурой

$$15000 - 4000 - 1000 - 1$$

и использованием dropout.

# Kaggle Merck Molecular Activity Challenge

1-й результат – нейросеть с использованием dropout. Без препроцессинга данных :)

| # | Δ1w | Team Name * in the money | Score | Entries |
|---|---|---|---|---|
| 1 | - | gggg * | 0.49410 | 20 |
| 2 | ↑3 | DataRobot * | 0.48811 | 37 |
| 3 | ↑10 | . * | 0.48209 | 88 |
| 4 | ↓2 | Gangnam Style | 0.48158 | 43 |
| 5 | ↑56 | Luxtorpeda | 0.48154 | 35 |

# Google Image Search

*DNNResearch* – стартап, созданный Geoffrey Hinton, Alex Krizhevsky и Ilya Sutskever (2011).
В 2012 приобретен компанией Google.
Разработан алгоритм поиска по изображениям Google, аналогичный методу, использованному для ImageNet. Основан на глубокой нейронной сети типа LeNet.

# Другие применения

- Перевод (английский $\rightarrow$ китайский, Microsoft)
- Распознавание речи (Google Search, Bing)
- Новостная лента в Fabebook (в разработке с сентября 2013)

# Пишем нейросеть

## Необходимые требования

- Стандартная задача регрессии или классификации
- Большая обучающая выборка
- Много признаков

## Инструменты

- Библиотека Theano (Python + CUDA)
- Библиотека Deepnet (Python + CUDA)
- Написать руками в MATLAB

## Архитектура сети

- Если объекты обучающей выборки – изображения, то рекомендуется использовать LeNet
(Рекомендации по настройке LeNet есть здесь)
- Иначе – обычная нейронная сеть с fully-connected слоями

## Нейросеть с fully-connected слоями

- Количество слоев – обычно 3-5
- На скрытых слоях обычно от 500 до 2000 нейронов.
- Предобучение Autoencoder-ом (опционально)

## Функции активации

- $\sigma(x) = \frac{1}{1+e^{-x}}$ (стандартный вариант)
- $tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- $max(0, x)$ (ReLU-function, обучается быстрее)

## Выходной слой

- В случае регрессии функции активации на выходном слое может не быть
- Для классификации – softmax:

$$output_i(x_1, .., x_n) = \frac{exp(x_i)}{\sum_j exp(x_j)}$$

функция потерь – кросс-энтропия выходного распределения с целевым

## Инициализация весов

Инициализируем все веса случайными величинами из нормального распределения с $\mu = 0$ и $\sigma \approx 10^{-3}$.

## Предобработка данных

Признаки центрируются и нормируются. Имеет смысл поэкспериментировать.

## Dropout rate

Dropout rate – вероятность «выкидывания» узла. Для скрытых слоев выставляется dropout rate, равный $p = 0.5$, для входного слоя $0 \leq p < 0.5$.

## Обучение

- Обучение производится «эпохами» – проходами по всей выборке в произвольном порядке
- Объекты группируются в блоки (minibatch), по 10-100 объектов в каждом. После обработки каждого блока градиент усредняется, обновляются веса нейросети.
- Коэффициент обучения (learning rate) подбирается экспериментально

## Что почитать

- DeepLearning.net – много разной информации по Deep Learning-y
- Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. Science 313.5786 (2006): 504-507. – статья про Autoencoder
- Hinton G. E. et al. Improving neural networks by preventing co-adaptation of feature detectors. – оригинальная статья по Dropout
- Srivastava, Nitish. Improving neural networks with dropout. Diss. University of Toronto, 2013. – подробный анализ Dropout

# В С Ё !

Geoffrey Hinton