

Deep Learning

Новый взгляд на нейронные сети

Содержание доклада

1. Ликбез

- нейронные сети

- метод обратного распространения ошибок

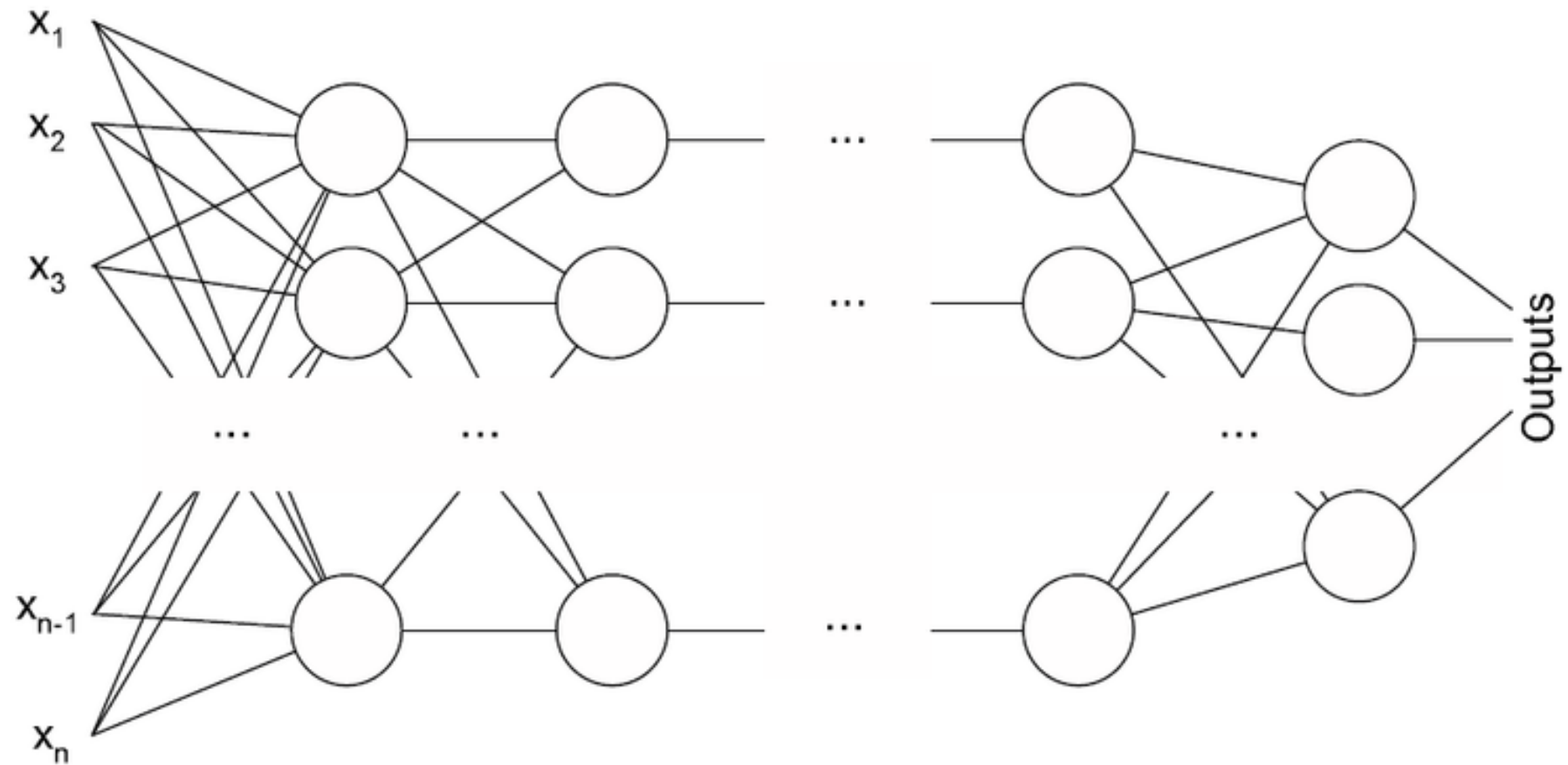
2. Предпосылки deep learning

3. Convolution neural network

- Настройка весов

4. Deep belief nets

Ликбез - нейронные сети



Функция активации $f(s) = \frac{1}{1 + e^{-2\alpha s}}$

Ликбез - нейронные сети

Функционал качества $E(\{w_{i,j}\}) = \frac{1}{2} \sum_{k \in \text{Outputs}} (t_k - o_k)^2$

Стохастический градиентный спуск

$$\Delta w_{i,j} = -\eta \frac{\partial E}{\partial w_{i,j}}$$

Но как же посчитать эту производную?

С помощью метода обратного распространения ошибки

Ликбез - Back Propagation

$$\Delta w_{i,j} = -\eta \frac{\partial E}{\partial w_{i,j}}$$

Для последнего слоя: $\frac{\partial E}{\partial w_{i,j}} = \frac{\partial E}{\partial S_j} \frac{\partial S_j}{\partial w_{i,j}} = x_i \frac{\partial E}{\partial S_j}$

$$\frac{\partial E}{\partial S_j} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial S_j} = \left(\frac{\partial}{\partial o_j} \frac{1}{2} \sum_{k \in \text{Outputs}} (t_k - o_k)^2 \right) \left(\frac{\partial \sigma(S_j)}{\partial S_j} \right) = \left(\frac{1}{2} \frac{\partial}{\partial o_j} (t_j - o_j)^2 \right) (o_j(1-o_j)) = -o_j(1-o_j)(t_j - o_j).$$

Для других слоев: $\frac{\partial E}{\partial S_j} = \sum_{k \in \text{Children}(j)} \frac{\partial E}{\partial S_k} \frac{\partial S_k}{\partial S_j}$

$$\frac{\partial S_k}{\partial S_j} = \frac{\partial S_k}{\partial o_j} \frac{\partial o_j}{\partial S_j} = w_{i,j} \frac{\partial o_j}{\partial S_j} = w_{i,j} o_j(1 - o_j)$$

Ликбез - Back Propagation

Последний слой: $\delta_j = -o_j(1 - o_j)(t_j - o_j)$

Внутренние слои: $\delta_j = -o_j(1 - o_j) \sum_{k \in \text{Outputs}(j)} \delta_k w_{j,k}$

$$\Delta w_{i,j} = -\eta \delta_j x_i$$

Нейронные сети - неудачный пример

1. Пусть у нас есть 10 000 изображений размера 32 на 32, хотим научиться распознавать цифры.
2. Возьмем нейронную сеть, примерно 1000 входов, 10 выходов. И еще возьмем один скрытый слой, 200 узлов.
3. Итого у нас $1000 \cdot 200 + 200 \cdot 10 = 200\,000$ весов.
4. Локальный максимум гарантирован.
5. Обобщающая способность нулевая.
6. По сути нейронная сеть - огромный многочлен с кучей параметров.

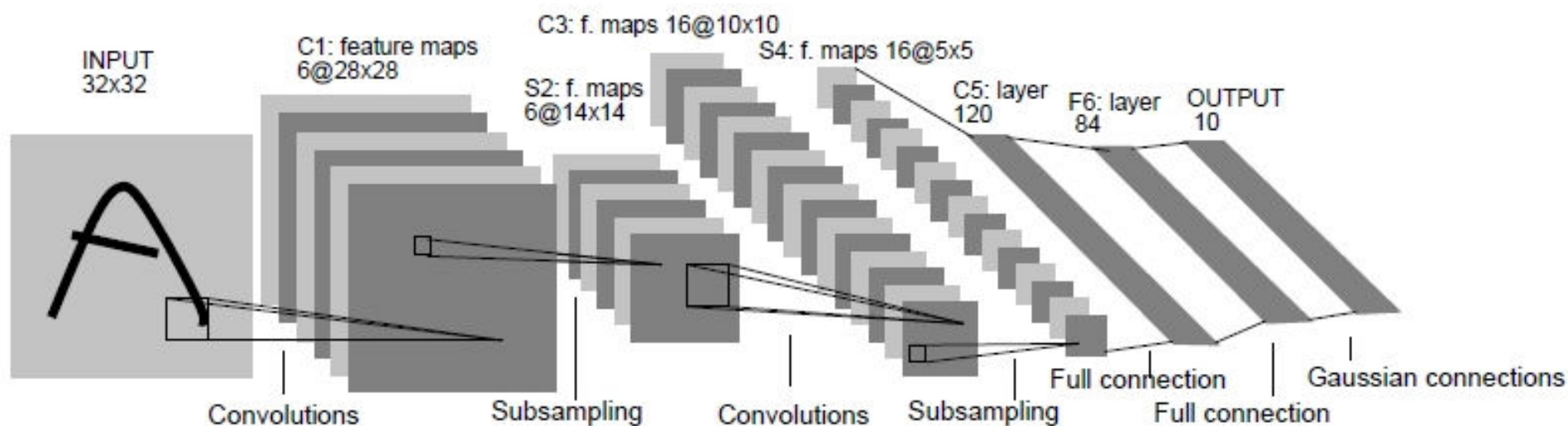
Предпосылки deep learning

- Мозг - не просто куча связанных нейронов
- Cortex (кора головного мозга) пропускает информацию через несколько иерархических слоев предобработки, каждый из которых выполняет свою функцию, и мы хотим строить иерархические модели
- Нейроны, выполняющие сходные функции работают не независимо, и мы хотим сделать, чтобы элементы наших сетей были взаимосвязаны

Convolution neural network

- Распознавание изображений (выделение паттернов)
- Будем учитывать локальную информацию
- Будем использовать иерархическую структуру
- Будем использовать совместную настройку весов
- И получим ...

Convolution neural network

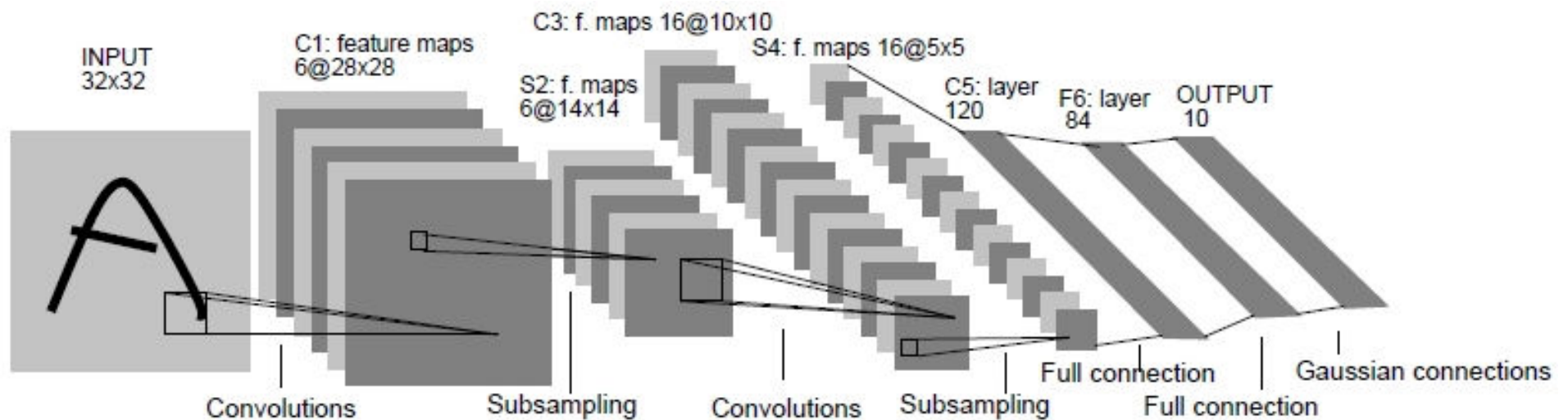


Будем чередовать слои "Convolutions" и "Subsampling", пока не придем к вектору малой размерности.

Convolution: выбираем несколько ядер, делаем свертку

Subsampling: уменьшаем размер изображений

Convolution neural network



- Немного формул на доске
- что такое ядро
 - формула для свертки
 - формула для subsampling'a

Convolution neural network

Свойства:

1. Свертка одна для изображения -> общие веса -> число параметров резко уменьшается
2. На каждом этапе с использованием разных сверток мы получаем изображения, характеризующие разные особенности исходного изображения -> feature extractor
3. Наличие subsampling'a обеспечивает робастность по отношению к поворотам, шумам, сдвигам
4. Практически невозможно переобучиться

Convolution neural network

Как обучать общие веса?

$$w_k \leftarrow w_k - \epsilon_k \frac{\partial E^p}{\partial w_k}$$

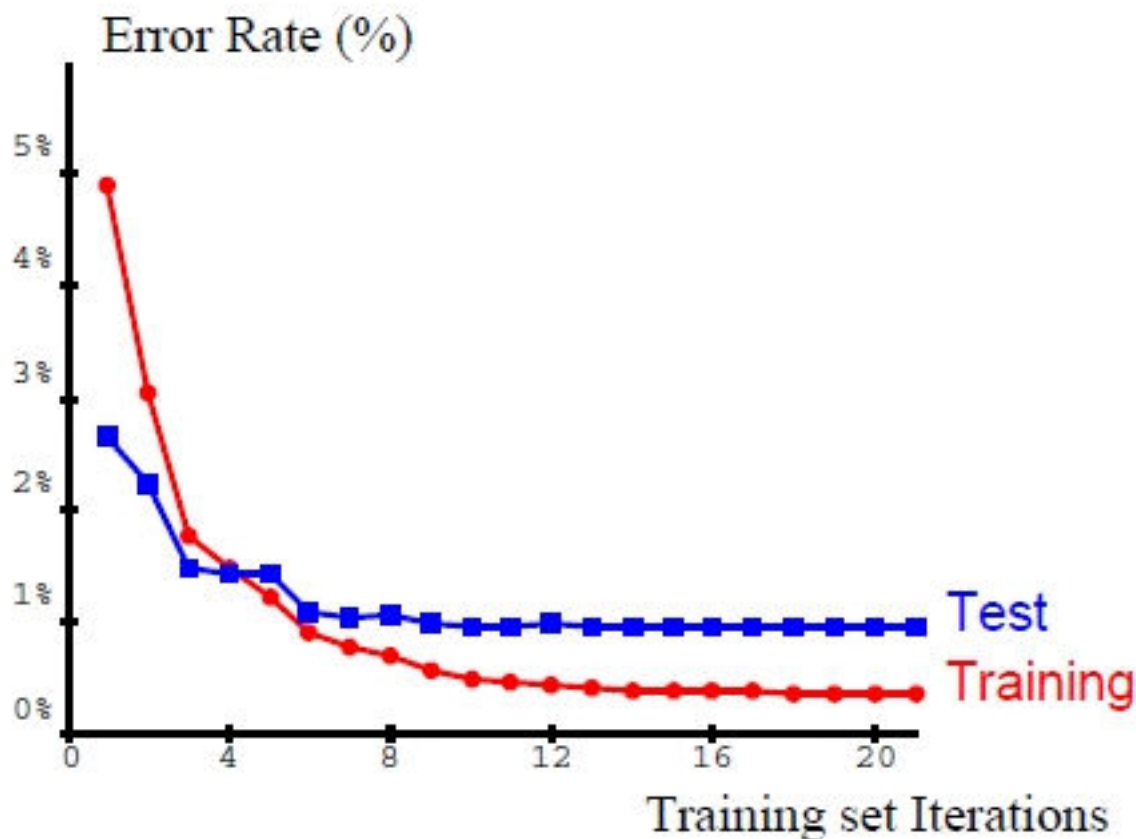
$$\frac{\partial E^p}{\partial w_k} = \sum_{(i,j) \in V_k} \frac{\partial E^p}{\partial u_{ij}}$$

$$u_{ij} = w_k \quad \forall (i,j) \in V_k$$

При фиксированном шаге получим метод обратного распространения ошибки в чистом виде.

Convolution neural network - задачи

1. Распознавание рукописных цифр



Каждая итерация обучения включает в себя обучение на разных подмножествах обучающего множества, этим объясняется низкий процент ошибки после первой итерации.

Convolution neural network - задачи

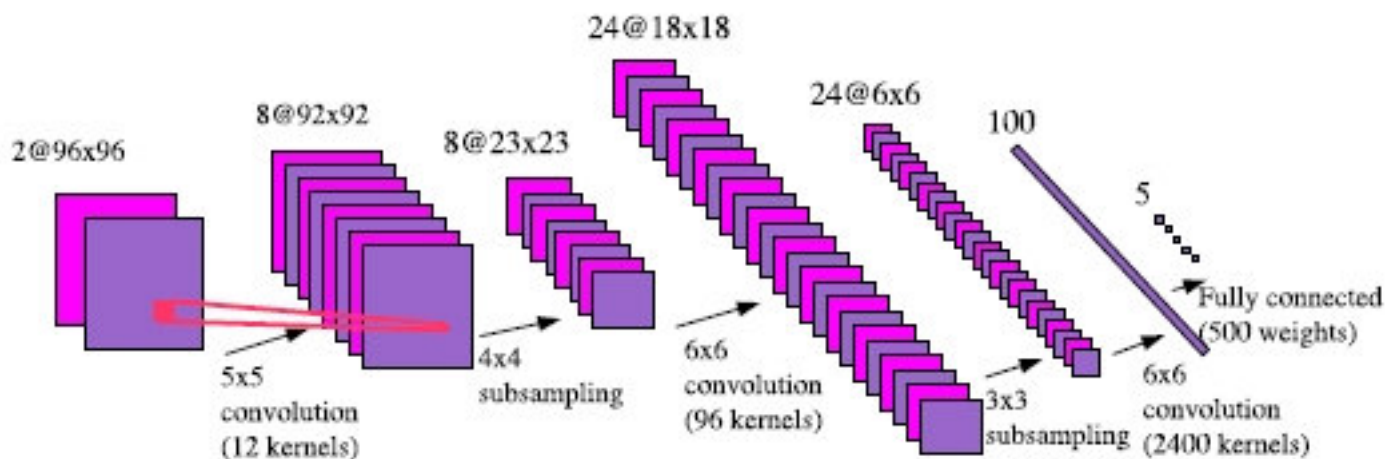
1. Распознавание рукописных цифр



Пример неправильно распознанных цифр

Convolution neural network - задачи

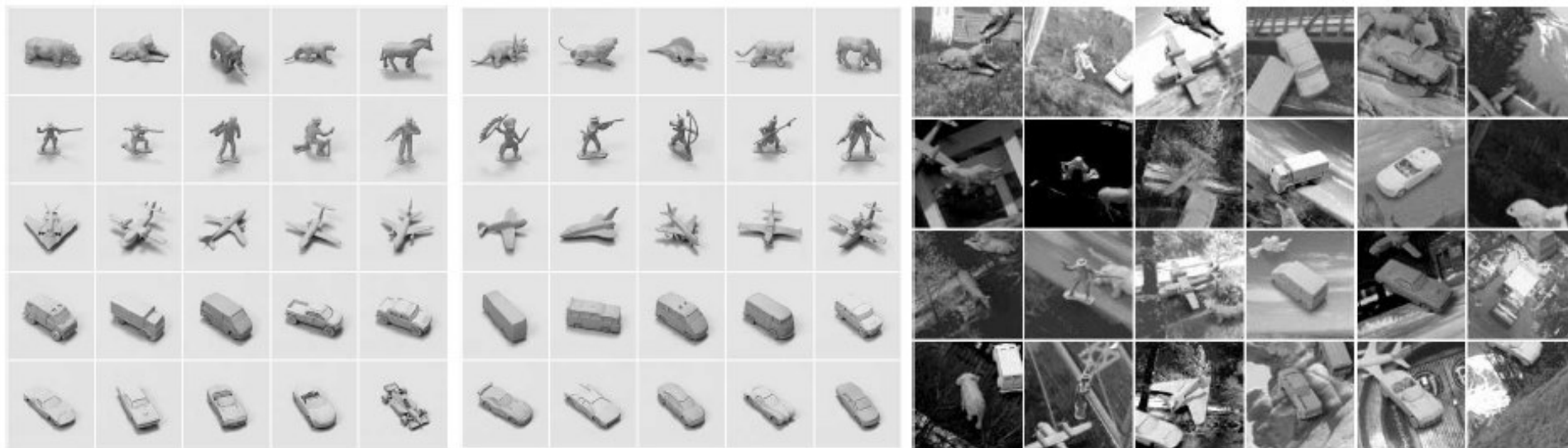
2. Распознавание стереопар



Абсолютно аналогично, за тем лишь исключением, что на вход дается два изображения, которые обрабатываются независимо.

Convolution neural network - задачи

2. Распознавание стереопар



Заявленная ошибка - 7.2%

Deep Belief Nets

RBM (Restricted Boltzman Machines) - для узлов определены вероятности

$$p(v_i^k = 1|\mathbf{h}) = \frac{\exp(b_i^k + \sum_{j=1}^F h_j W_{ij}^k)}{\sum_{l=1}^K \exp(b_i^l + \sum_{j=1}^F h_j W_{ij}^l)}$$

$$p(h_j = 1|\mathbf{V}) = \sigma(b_j + \sum_{i=1}^m \sum_{k=1}^K v_i^k W_{ij}^k)$$

$$p(\mathbf{V}) = \sum_{\mathbf{h}} \frac{\exp(-E(\mathbf{V}, \mathbf{h}))}{\sum_{\mathbf{V}', \mathbf{h}'} \exp(-E(\mathbf{V}', \mathbf{h}'))}$$

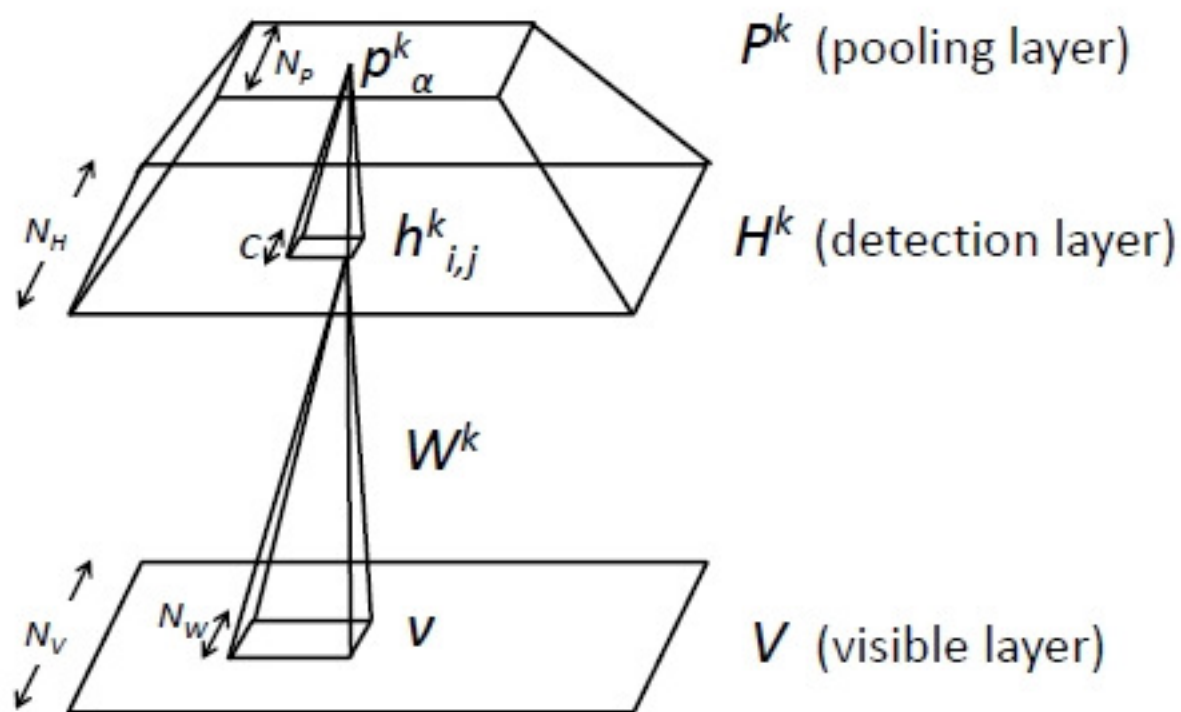
Deep Belief Nets

$$\begin{aligned} E(\mathbf{V}, \mathbf{h}) = & - \sum_{i=1}^m \sum_{j=1}^F \sum_{k=1}^K W_{ij}^k h_j v_i^k + \sum_{i=1}^m \log Z_i \\ & - \sum_{i=1}^m \sum_{k=1}^K v_i^k b_i^k - \sum_{j=1}^F h_j b_j \end{aligned} \quad (4)$$

where $Z_i = \sum_{l=1}^K \exp(b_i^l + \sum_j h_j W_{ij}^l)$ is the normalization term that ensures that $\sum_{l=1}^K p(v_i^l = 1 | \mathbf{h}) = 1$.

Deep Belief Nets

Из подобных последовательных компонент строим сеть.



Как видно, аналогия с CNN практически полная, если заменить в CNN обычные нейронные сети на RBM. Разумеется, при этом меняется также алгоритм обучения.

Литература

CNN

- Gradient Based Learning Applied to Document Recognition. Yann LeCun, Leon Bottou, Yoshua Bengio, Patrick Haffner

- Large-scale Learning with SVM and Convolutional Nets for Object Categorization Fy Jie Huang, Yann LeCun

DBN

- A Fast Learning Algorithm for Deep Belief Nets

- Convolutional Deep Belief Networks

for Scalable Unsupervised Learning of Hierarchical Representations