

Лекция №10

Введение в машинное обучение

Спасёнов Алексей

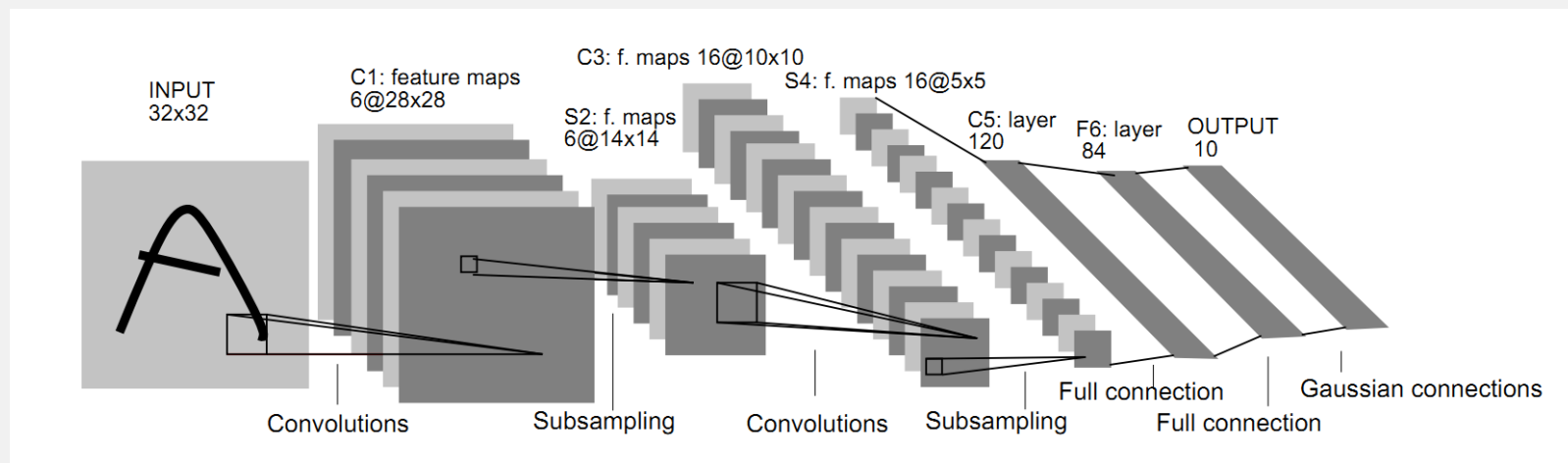


Содержание лекции

1. Transfer learning - Fine tuning
2. Автоэнкодеры
 1. MLP – Автоэнкодеры
 2. CNN – Автоэнкодеры
 3. Denoising – Автоэнкодеры
3. Сиамские сети
 1. Contrastive loss
 2. Triplet loss
 3. Center loss



LeNet-5, convolutional neural networks



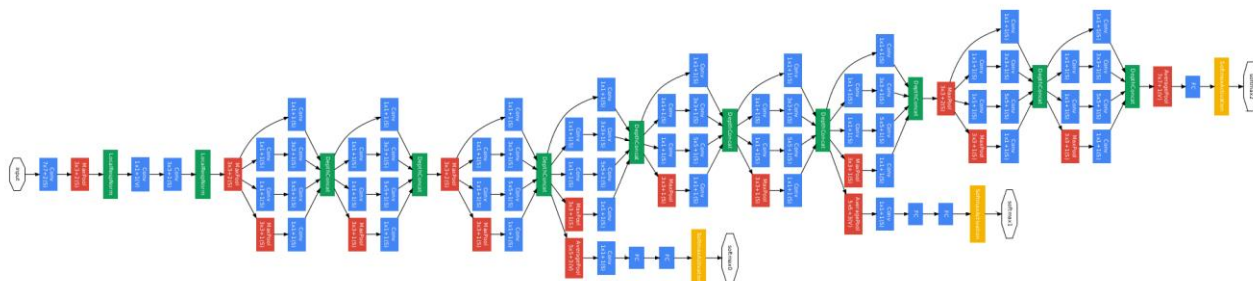
Карты признаков

<http://yann.lecun.com/exdb/lenet/>

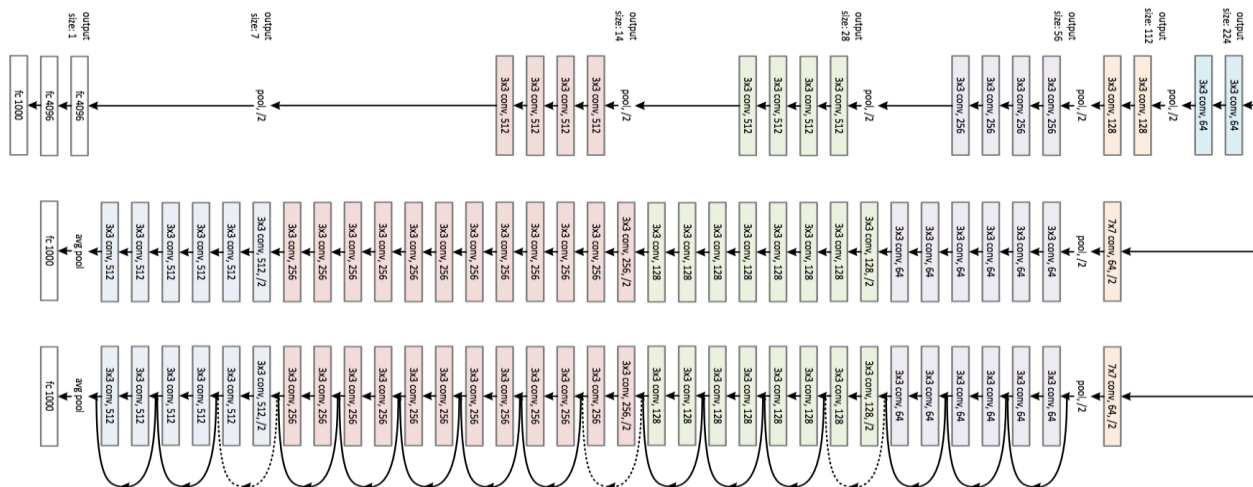
Нейронные сети. CNN



GoogLeNet



VGG 16



ResNet



Convolution

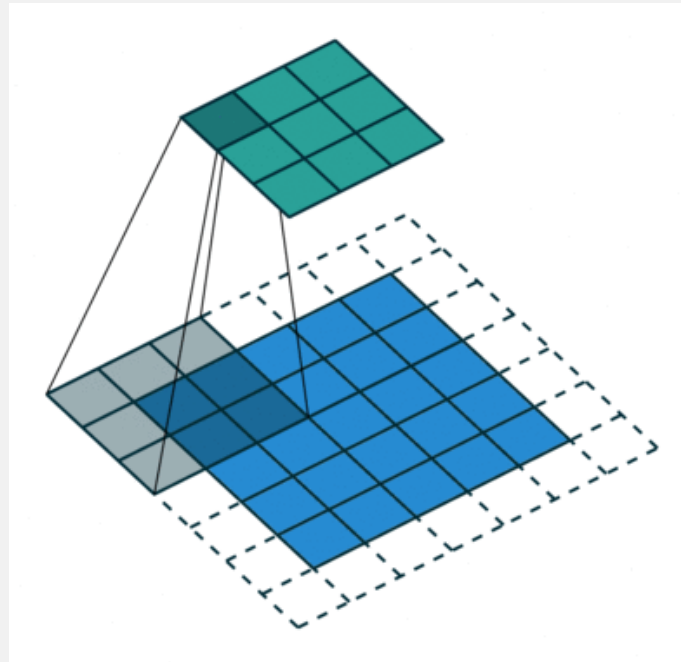
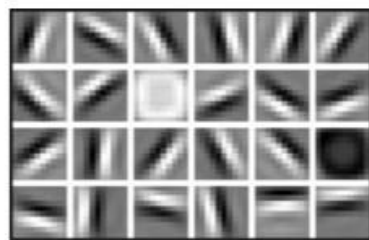
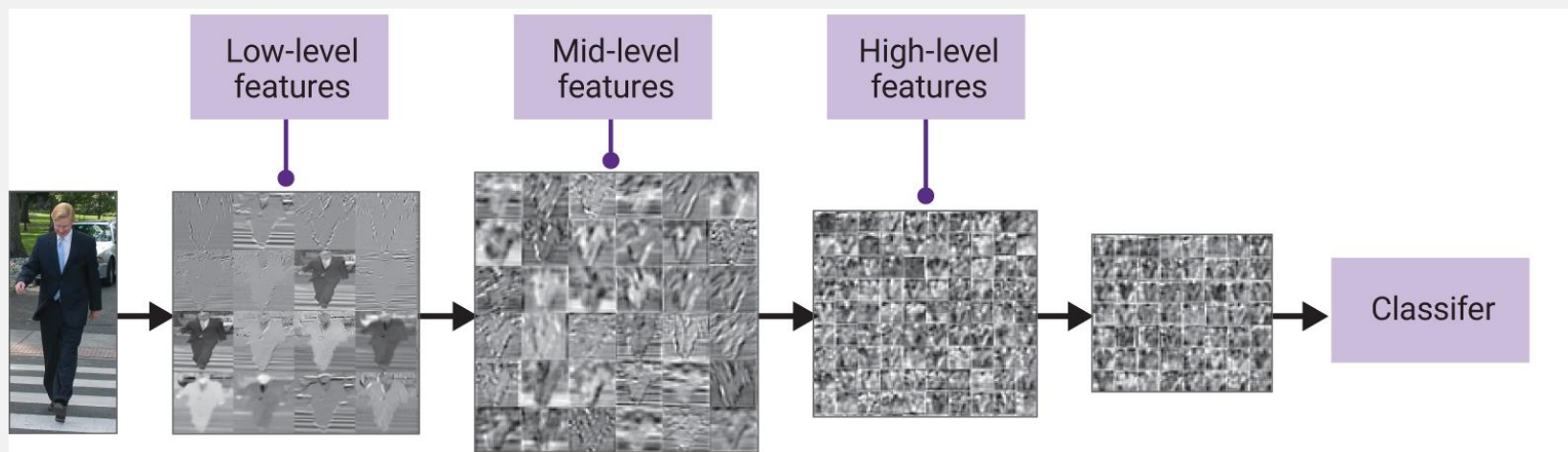




Image Embedding



First Layer Representation



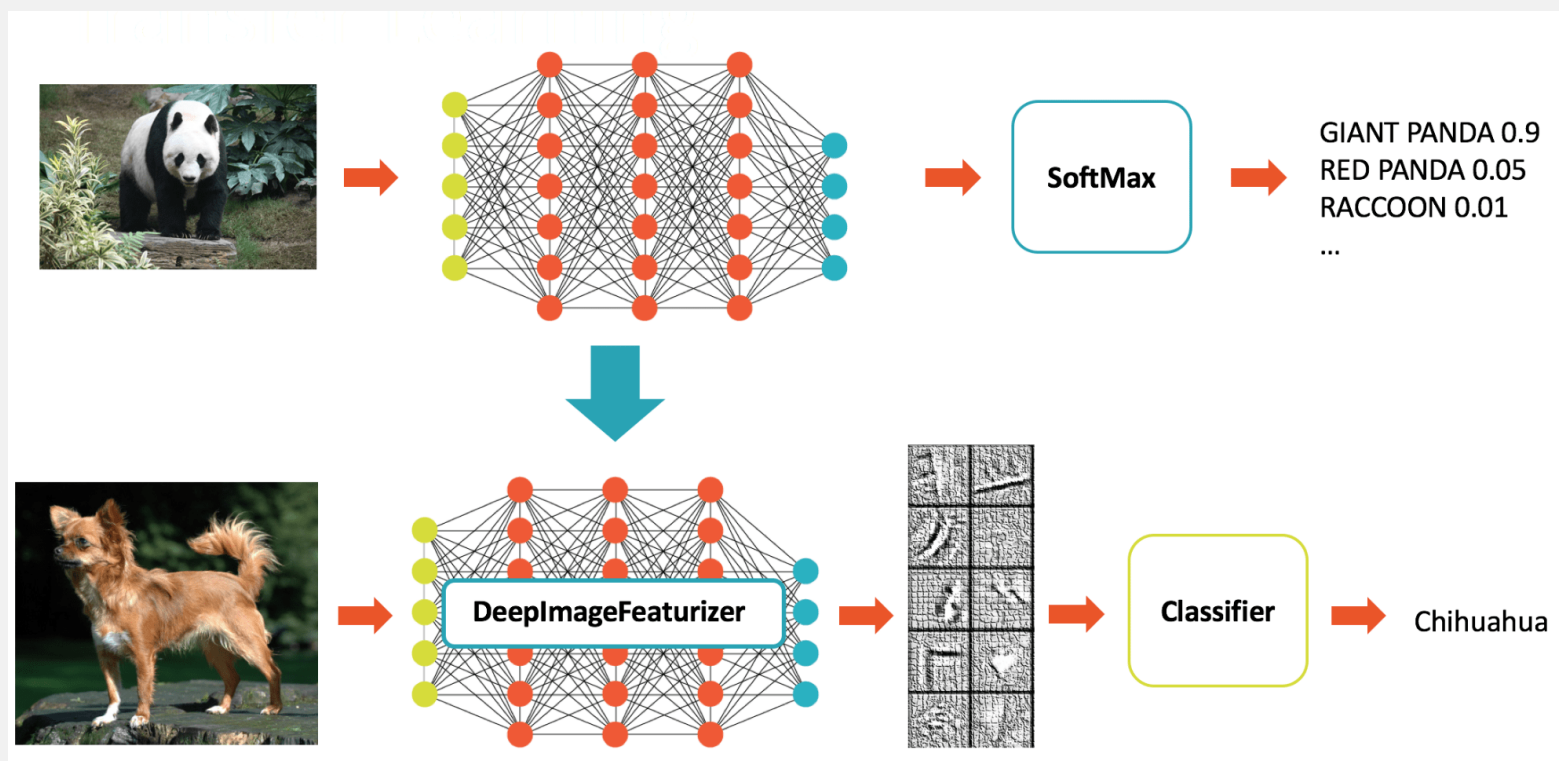
Second Layer Representation



Third Layer Representation



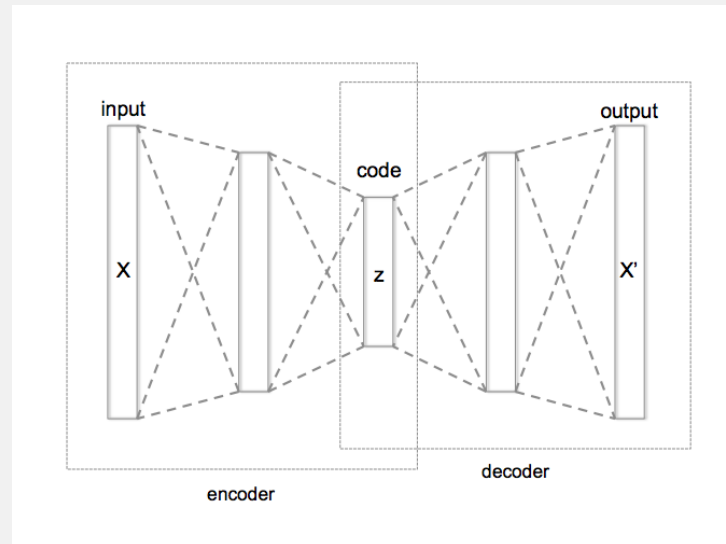
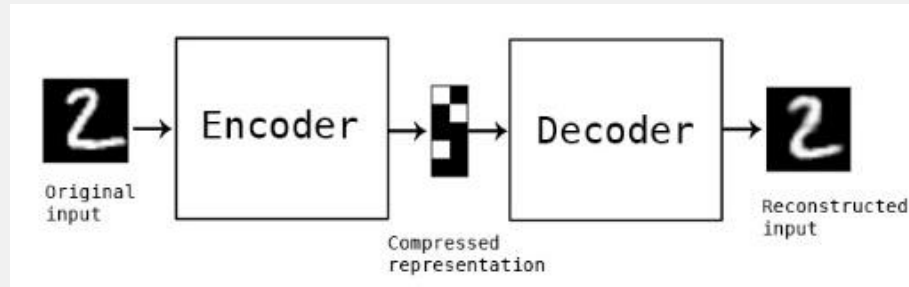
Transfer learning - Fine tuning



Нейронные сети. Автоэнкодеры



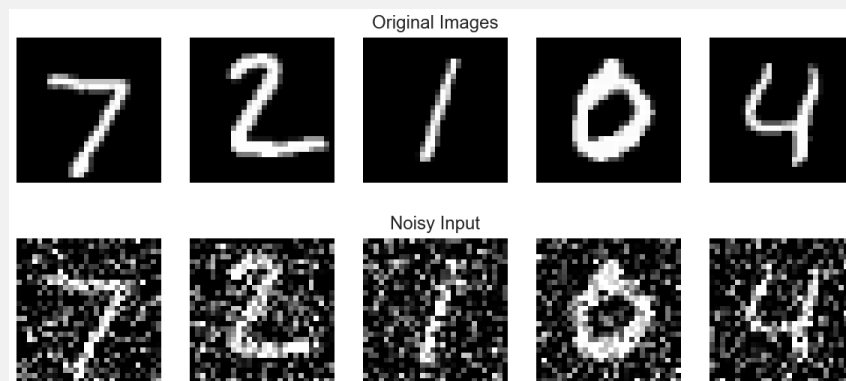
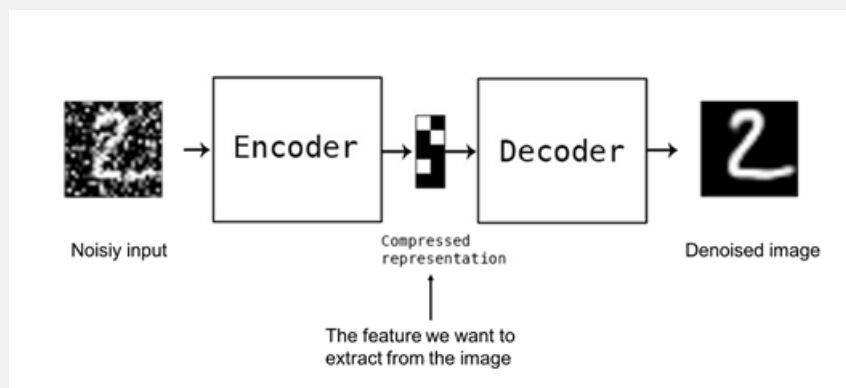
Autoencoder на основе MLP



Нейронные сети. Автоэнкодеры

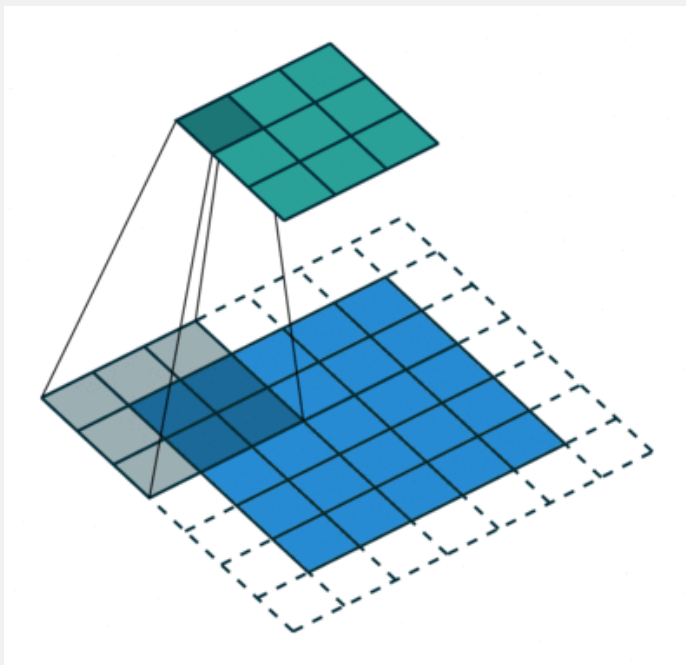


Denoising Autoencoder

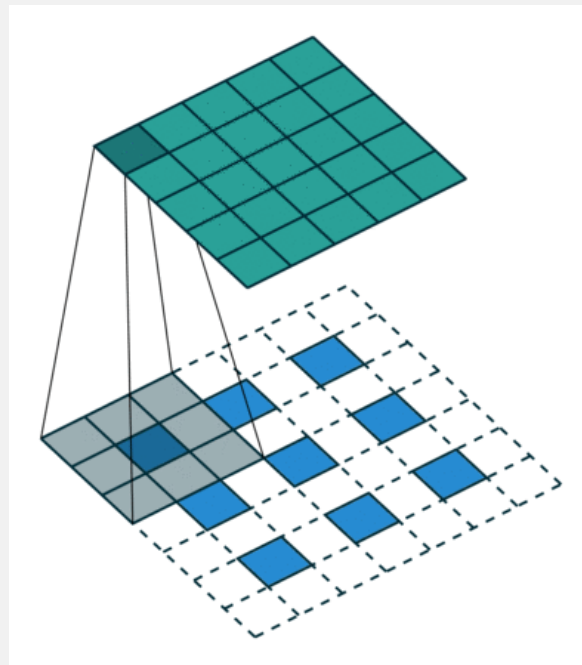




Convolution



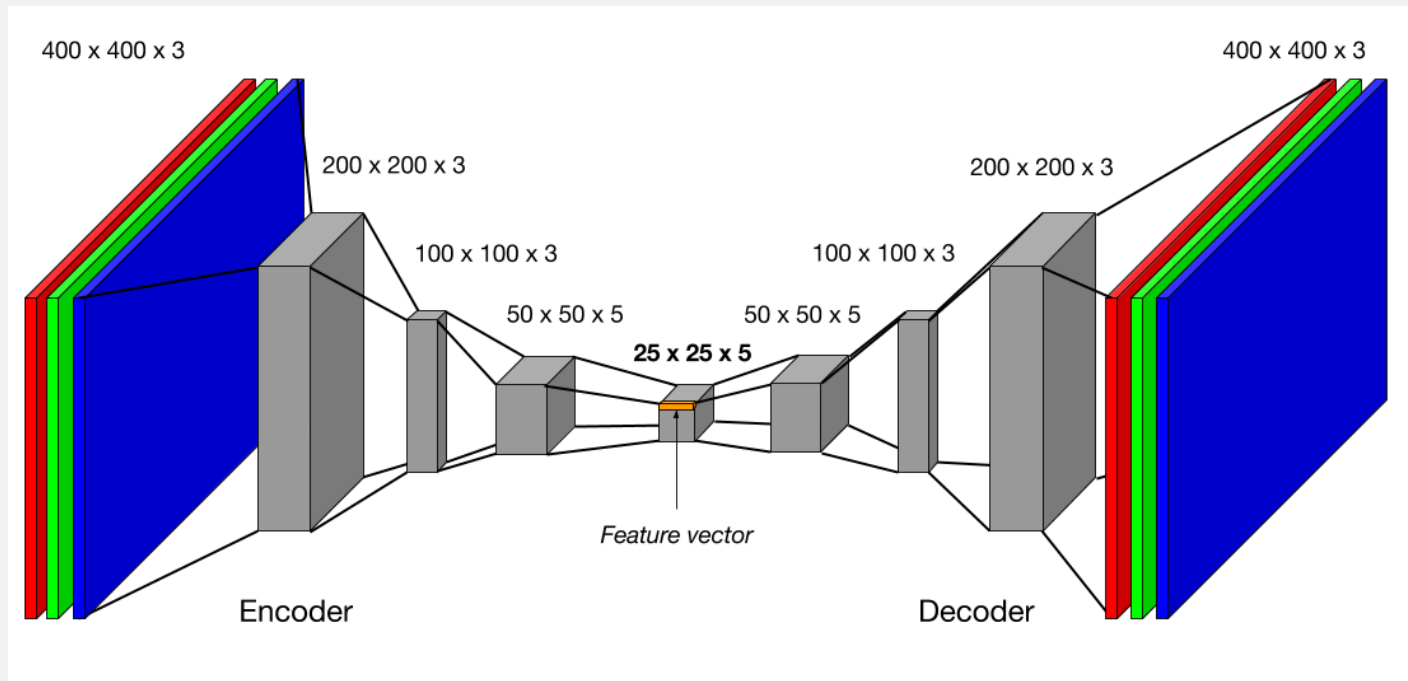
Deconvolution



Нейронные сети. Автоэнкодеры



Autoencoder на основе CNN



Deep Metric Learning



1. При решении задач компьютерного зрения достаточно редки ситуации, когда имеется много изображений, характеризующих каждый интересующий класс объектов
2. Чаще всего имеется достаточно сильная несбалансированность классов

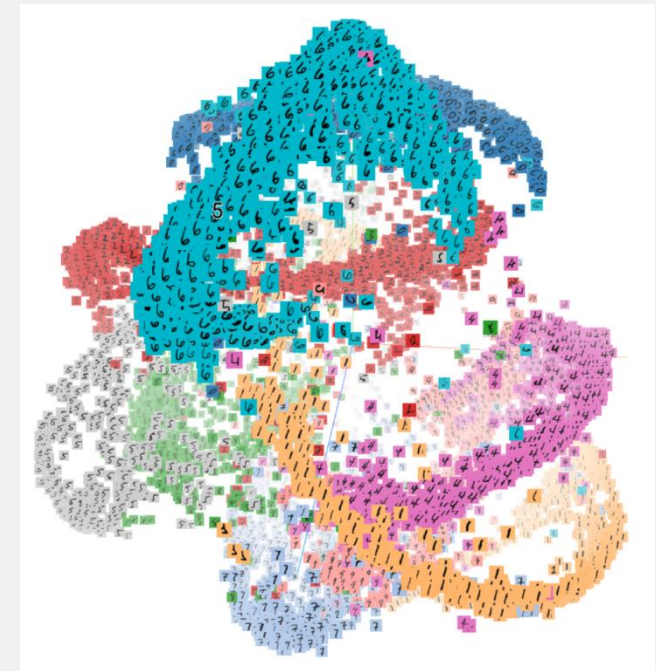
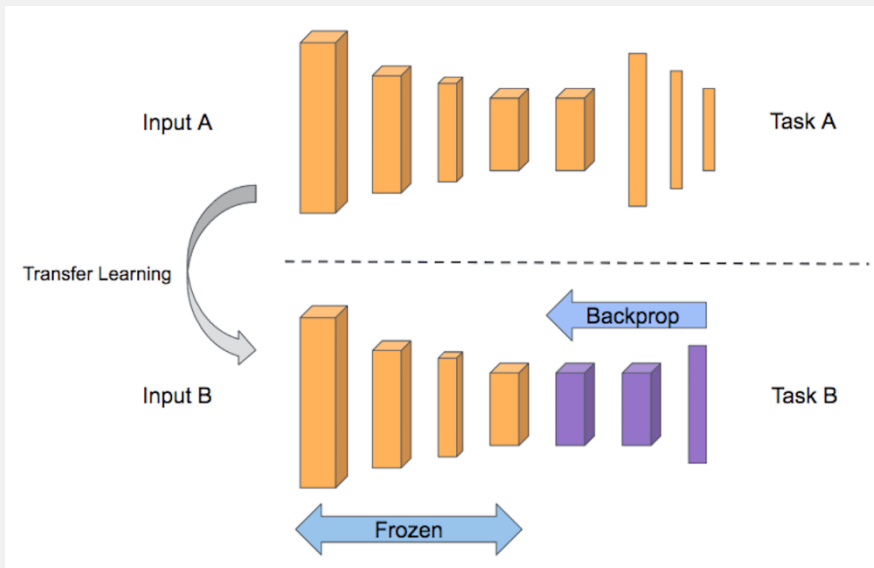


Deep Metric Learning



1. Создание модели, хорошо описывающей изображения
2. Обучение модели таким образом, чтобы сохранялась «близость» между embedding'ами изображений, имеющих семантическое сходство
3. Возможность использовать полученную модель для предсказания классов, которые не были представлены в обучающей выборке

Deep Metric Learning





1. Имеется некоторое отображение для изображений в embedding'и

$$f(x_i): R^L \rightarrow R^D$$

2. Метрика схожести объектов

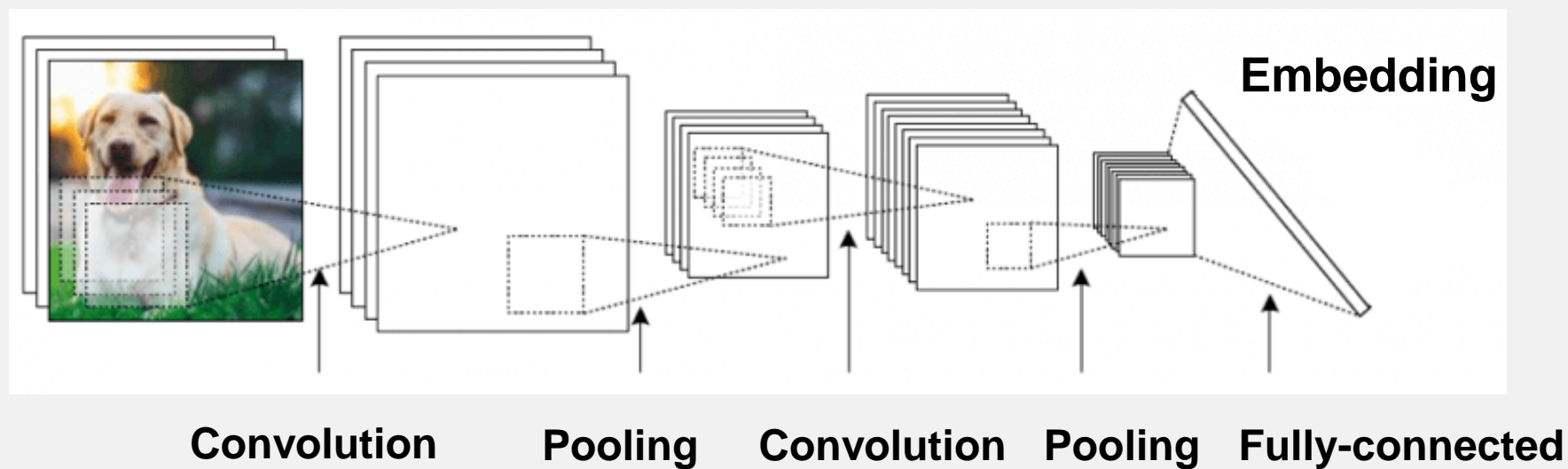
$$\phi(x_i, x_j) = \text{distance}(f(x_i), f(x_j)),$$

где *distance* может быть евклидовым, косинусным расстоянием и т.д.

$$\phi \left(\begin{bmatrix} \text{car1} & \text{car2} \end{bmatrix} \right) < \phi \left(\begin{bmatrix} \text{car1} & \text{mower} \end{bmatrix} \right) < \phi \left(\begin{bmatrix} \text{car1} & \text{airplane} \end{bmatrix} \right)$$

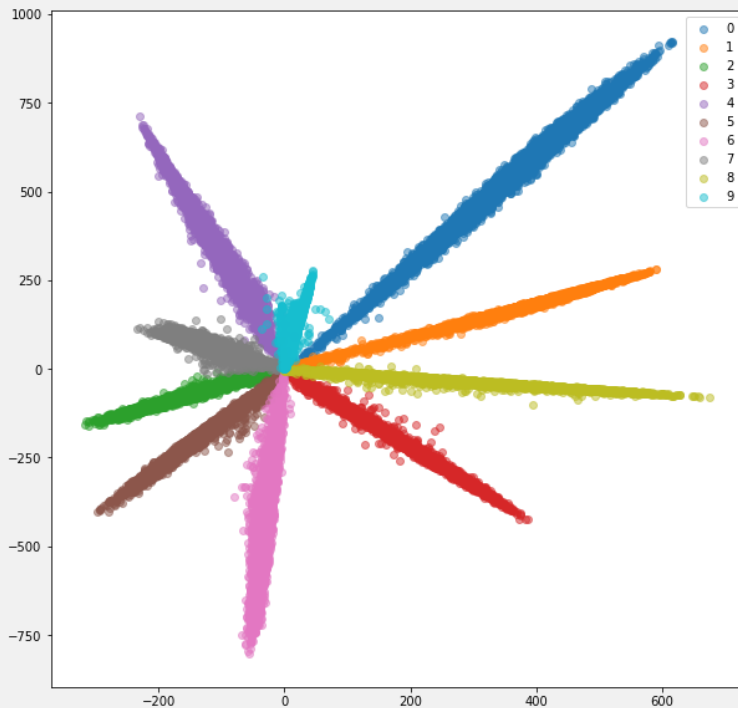


Получение embedding'ов





Softmax Loss



Преимущества:

1. Простая реализация
2. Можно получить компактный embedding

Недостатки:

1. Отсутствует «близость»



Contrastive Loss

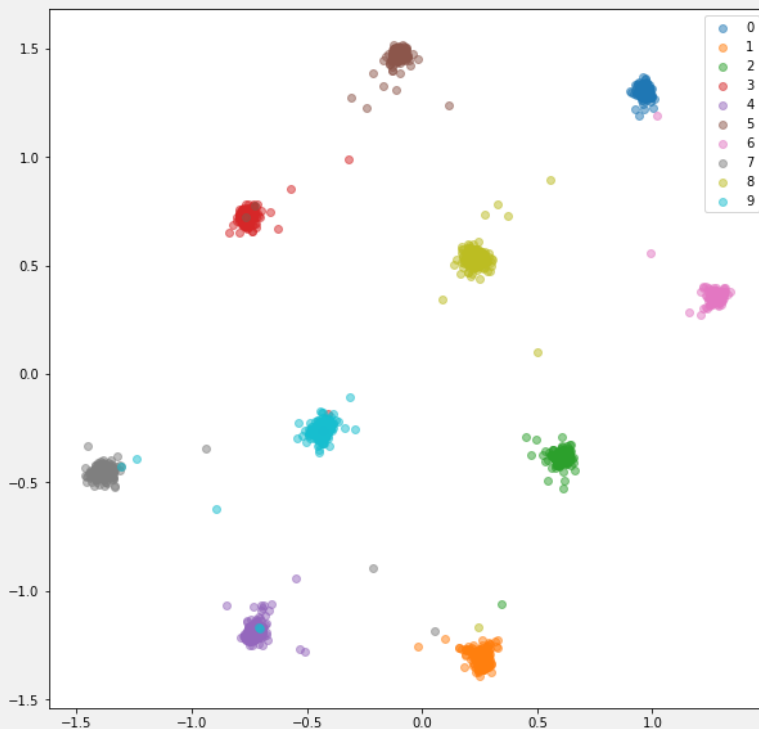
Разделяем выборку на positive и negative пары

Ставим метку $y_{ij} = 1$ для positive пар и $y_{ij} = -1$ для negative пар

$$L(X, \Theta) = \frac{1}{2} \sum_{i=1}^m y_{ij} \|f(x_i) - f(x_j)\|_2^2 + (1 - y_{ij}) \max(0, [\alpha - \|f(x_i) - f(x_j)\|_2]^2)$$



Contrastive Loss



Преимущества:

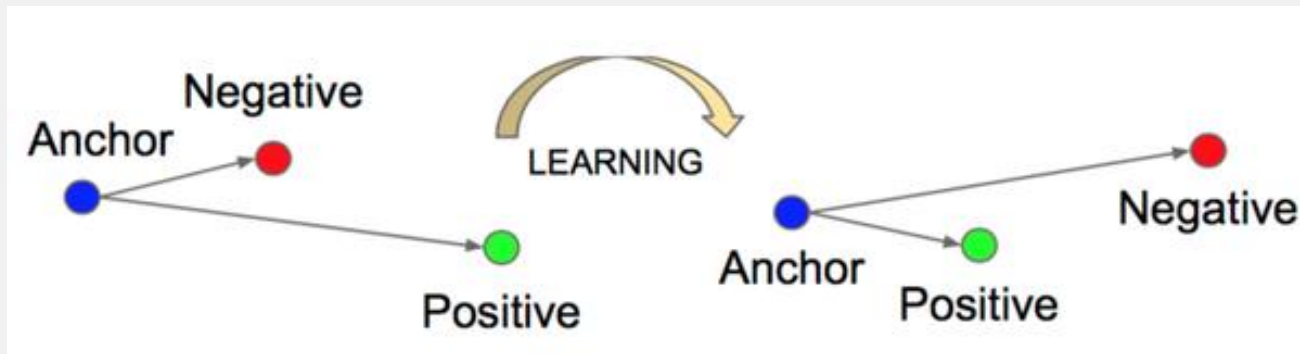
1. Простая реализация
2. Присутствует «близость» в представлении
3. Возможность комбинировать с другими loss'ами

Недостатки:

1. Медленная сходимость
2. Необходимость подбора гиперпараметра α
3. Формирование батчей



Triplet Loss

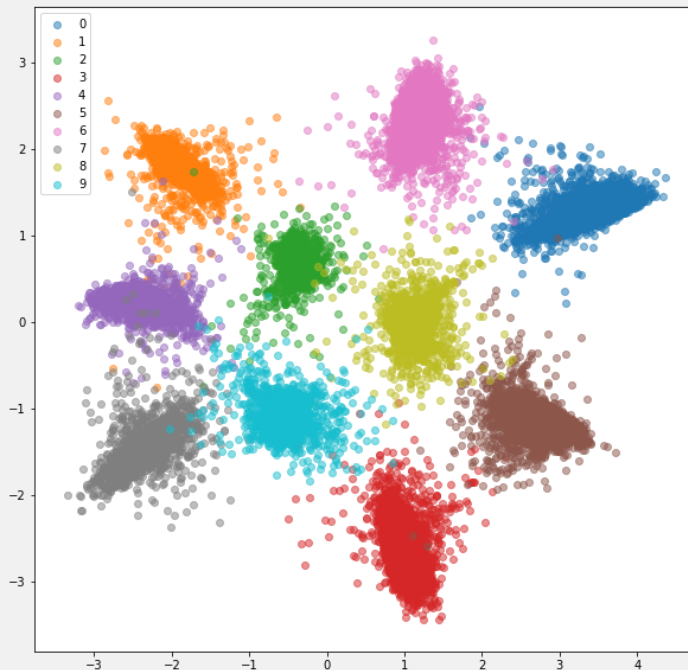


$$L(X, \Theta) = \frac{1}{2} \sum_{i=1}^m \max(0, \|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha)$$

Deep metric learning using triplet network,
Elad Hoffer, Nir Ailon



Triplet Loss



Преимущества:

1. + Гибкость формирования внутреннего представления

Недостатки:

1. Частое попадание в локальные оптимумы
2. Необходимость подбора гиперпараметра α
3. Формирование батчей



Center Loss

Добавление представления центров классов C_{y_i}

$$L_c(X, \Theta) = \frac{1}{2} \sum_{i=1}^m \|x_i - C_{y_i}\|_2^2$$

Обновление положения центров классов происходит по правилу:

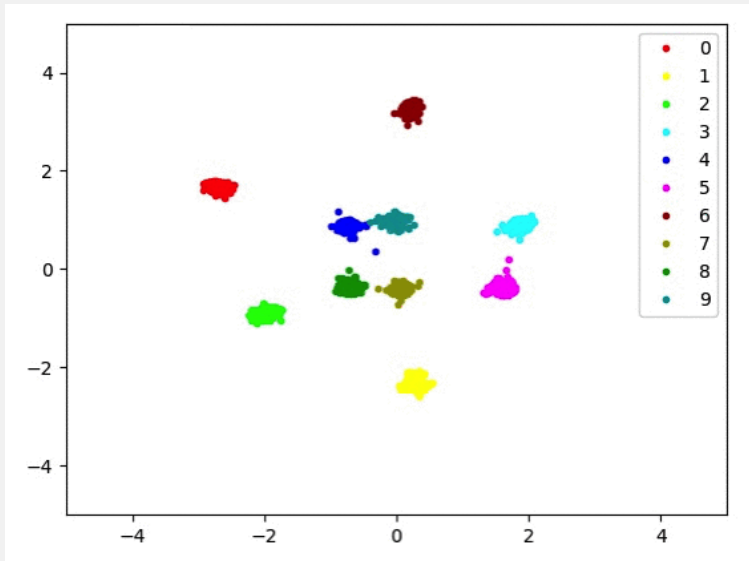
$$\Delta C_j = \frac{\sum_{i=1}^m \delta(y_i = j)(C_j - x_i)}{1 + \sum_{i=1}^m \delta(y_i = j)}$$

Комбинированный Loss:

$$L(X, \Theta) = - \sum_{i=1}^m \log \frac{e^{x^T w_j}}{\sum_{k=1}^K e^{x^T w_k}} + \lambda L_c$$



Center Loss



Преимущества:

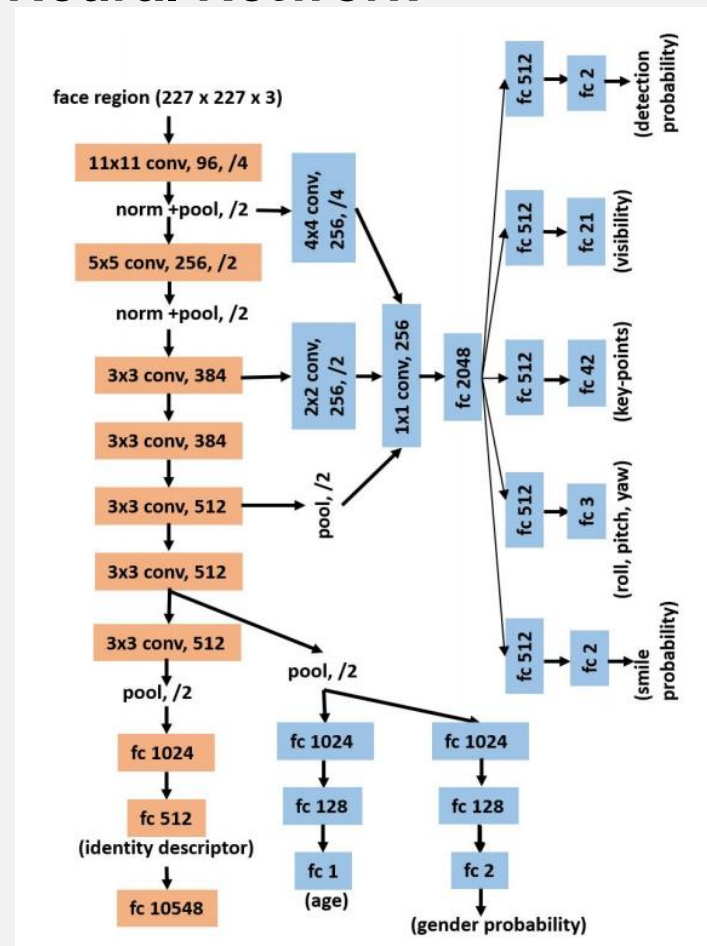
1. Быстрее обучается
2. Позволяет улучшить интерпретируемость

Недостатки:

1. Необходимость подбора гиперпараметра λ (обновление центроидов)
2. Добавление softmax

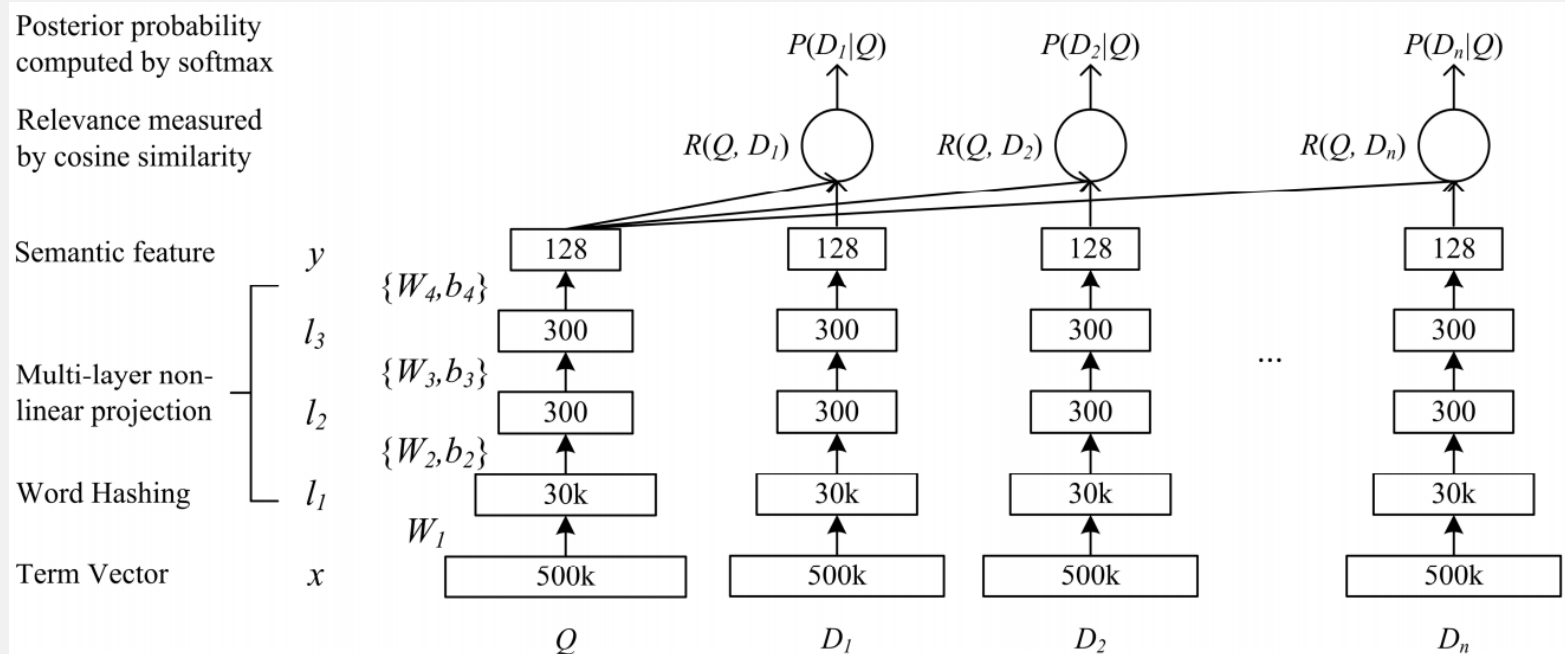


An All-In-One Convolutional Neural Network





Deep Semantic Similarity Model



Нейронные сети



Не забываем отмечаться и оставлять отзывы!

Спасибо!



**Спасибо за
внимание!**

Спасёнов Алексей

a.spasenov@corp.mail.ru