

Нобелевская премия по физике — 2024

Часть II

Искусственные нейронные сети

Воронцов Константин Вячеславович

д.ф.-м.н., проф. РАН • кафедра интеллектуальных систем МФТИ
кафедра машинного обучения и цифровой гуманитаристики МФТИ
ФИЦ ИУ РАН • Институт ИИ МГУ • кафедра ММП ВМК МГУ

• МФТИ • 21 октября 2024 •

1 Машинное обучение и нейронные сети

- Эмпирическая индукция
- Нейронные сети
- Вехи и этапы развития

2 Архитектуры и методы обучения

- Сети ассоциативной памяти
- От машин Больцмана к генеративным моделям
- Метод обратного распространения ошибок

3 Глубокие нейронные сети

- Глубина важнее ширины
- Свёрточные нейронные сети
- Капсульные нейронные сети

Принцип эмпирической индукции

«Не следует полагаться на сформулированные аксиомы и формальные базовые понятия, какими бы привлекательными и справедливыми они не казались. Законы природы нужно «расшифровывать» из фактов опыта.

Следует искать правильный метод анализа и обобщения опытных данных;

здесь логика Аристотеля не подходит в силу её абстрактности, оторванности от реальных процессов и явлений.»



Фрэнсис Бэкон
(1561–1626)

Таблица открытия: множество объектов $\{x_i; i = 1, \dots, \ell\}$

- $f(x_i)$ — измеряемое значение признака объекта x_i
- $y_i \in \mathbb{R}$ — измеряемое значение целевого свойства x_i , либо $y_i \in \{0, 1\}$ — отсутствие или наличие целевого свойства

Фрэнсис Бэкон. Новый органон. 1620.

Восстановление зависимостей по эмпирическим данным

Дано:

объекты $x_i = (f_1(x_i), \dots, f_n(x_i))$ и ответы $y_i = y(x_i)$, $i = 1, \dots, \ell$
 $f_j(x)$ — признаки объекта x , $j = 1, \dots, n$

Найти:

функцию $a(x, w)$, восстанавливающую зависимость $y(x)$

Критерий: минимум эмпирического риска

$$\sum_{i=1}^{\ell} \mathcal{L}(a(x_i, w), y_i) \rightarrow \min_w,$$

где $\mathcal{L}(a, y)$ — функция потерь от ошибки a при ответе y .

Основные типы задач обучения с учителем:

- регрессия: $y_i \in \mathbb{R}$, $\mathcal{L}(a, y) = (a - y)^2$
- классификация: $y_i \in \{-1, +1\}$, $\mathcal{L}(a, y) = [a \neq y]$

Искусственный нейрон — линейная модель классификации

Линейная модель нейрона (1943):

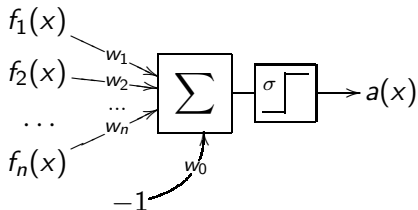
$$a(x, w) = \sigma \left(\sum_{j=1}^n w_j f_j(x) - w_0 \right)$$

$f_j(x)$ — признаки объекта x

w_j — веса признаков

w_0 — порог активации

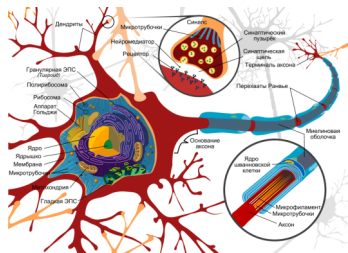
$\sigma(z)$ — функция активации



Уоррен
МакКаллок
(1898–1969)



Вальтер
Питтс
(1923–1969)

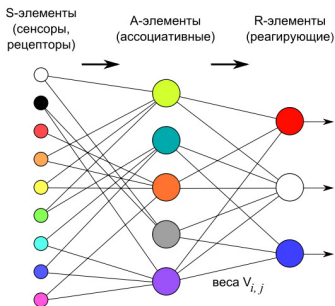


Персептрон Розенблатта (1957) и теорема Новикова (1960)

Mark-1 — первый нейрокомпьютер (1960)

Обучение — метод коррекции ошибки

Архитектура — двухслойная сеть



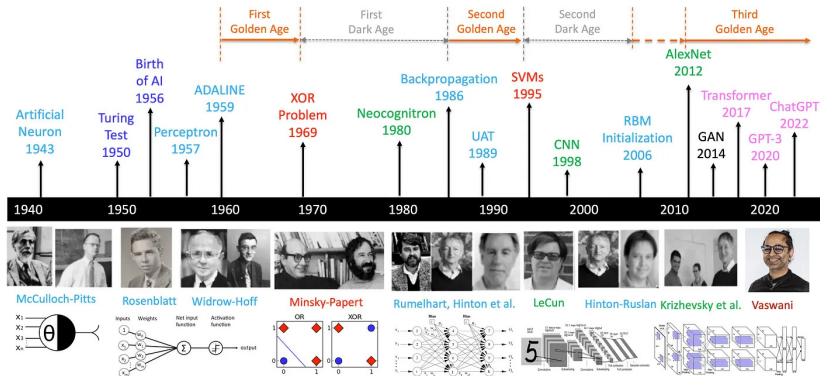
Фрэнк Розенблатт
(1928–1971)



Розенблатт Ф. Принципы нейродинамики. Перцептроны и теория механизмов мозга. 1965 (1962).

Novikoff A. B. J. On convergence proofs on perceptrons. 1962.

Основные вехи развития нейронных сетей



— Развитие нейронных сетей затормозилось на десятилетие

Минский М., Паперт С. Перцептроны. 1971 (1969).

— Рекуррентные сети

Hopfield J. Neural networks and physical systems with emergent collective computational abilities. 1982.

Ackley D.H., Hinton G.E., Sejnowski T.J. A learning algorithm for Boltzmann machines. 1985.

Hochreiter S., Schmidhuber J. Long short-term memory. 1997.

Основные вехи развития теории нейронных сетей

— BackProp — метод обратного распространения ошибок

Галушкин А. И. Синтез многослойных систем распознавания образов. 1974.
Werbos P. J. Beyond regression: new tools for prediction and analysis in the behavioral sciences. 1974.
LeCun Y. Une procedure d'apprentissage pour reseau a seuil assymetrique. 1985.
Parker D. B. Learning-logic: Casting the cortex of the human brain in silicon. 1985.
Rummelhart D., Hinton G., Williams R. Learning internal representations by error propagation. 1986.

— Глубокие сети (сети с большим числом слоёв)

Ивахненко А. Г., Лапа В. Г. Кибернетические предсказывающие устройства. 1965.
Rina Dechter. Learning while searching in constraint-satisfaction problems. 1986.
Hinton G.E. Learning multiple layers of representation. 2007.

— Свёрточные сети

LeCun, Bottou, Bengio, Haffner. Gradient-based learning applied to document recognition. 1998.
Krizhevsky, Sutskever, Hinton. ImageNet classification with deep convolutional neural networks. 2012.

— Капсульные сети

Sabour S., Frosst N., Hinton G.E. Dynamic Routing Between Capsules. 2017.

— Генеративные сети

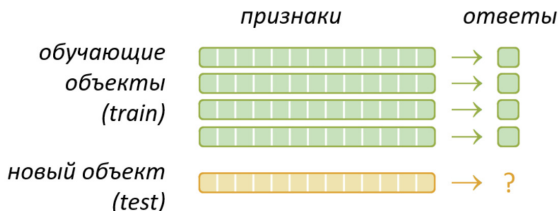
Hinton G.E., Osindero S., Teh Y.W. A fast learning algorithm for deep belief nets. 2006.
Kingma D.P., Welling M. Auto-Encoding Variational Bayes. 2013.
Goodfellow I. et al. Generative Adversarial Nets. 2014.
Vaswani A. et al. Attention is all you need. 2017.

Три основных этапа. Этап 1: вектор \rightarrow скаляр

Предсказательное моделирование векторных данных

Вход: векторные признаковые описания объектов

Выход: скалярные ответы (предсказания, прогнозы)



Приложения: медицинская диагностика,
геологическое прогнозирование, кредитный скоринг,...

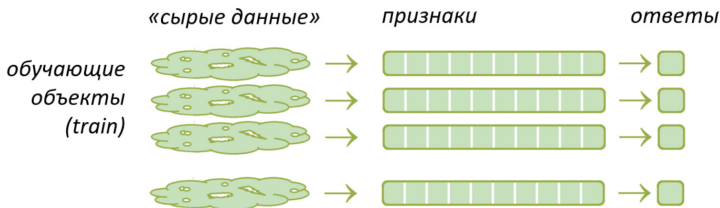
Модели: SVM, LR, MVR, RBF, MLP, ID3, CART, RF, GBM,...

Три основных этапа. Этап 2: структура → вектор → скаляр

Обучаемая векторизация сложно структурированных данных

Вход: сложно структурированные «сырые» данные объектов

Выход: векторные представления объектов, затем ответы



Приложения: классификация изображений, текстов, сигналов, голосовых команд, биометрическая идентификация личности,...

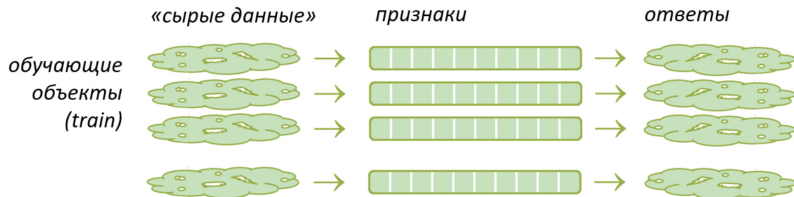
Модели: CNN, AlexNet, ResNet, word2vec, FastText,...

Три основных этапа. Этап 3: структура → вектор → структура

Обуаемая генерация сложно структурированных данных

Вход: сложно структурированные объекты

Выход: сложно структурированные ответы



Приложения: синтез изображений, перенос стиля,
распознавание речи, машинный перевод, суммаризация
текстов, чат-боты,...

Модели: seq2seq, CNN, RNN, LSTM, GAN, VAE, BERT, GPT,...

Сеть Хопфилда: постановка задачи

Дано: обучающая выборка объектов

$$X^\ell = \{x_1, \dots, x_\ell\} \subset \{-1, +1\}^n$$

Найти: модель ассоциативной памяти $a(x, w)$, способную выдавать объект x_i из X^ℓ , ближайший к заданному x

Критерий: минимум «энергии»

$$E(x) = -\frac{1}{2} \sum_{i=1}^{\ell} \langle x_i, x \rangle^2 = -\frac{1}{2} x^T \left(\sum_{i=1}^{\ell} x_i x_i^T \right) x = -\frac{1}{2} x^T W x \rightarrow \min_x$$

что эквивалентно поиску наиболее вероятного состояния x с минимальной энергией в модели Изинга без внешнего поля

$$p(x|W) = \frac{1}{Z(W)} \exp\left(-\frac{1}{2} x^T W x - x^T a\right) \rightarrow \max_x$$

Hopfield J. Neural networks and physical systems with emergent collective computational abilities. 1982.

Сеть Хопфилда: решение и «архитектура сети»

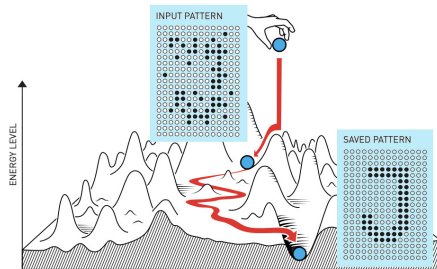
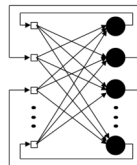
Метод внешнего произведения (Хопфилд):

$$W = \frac{1}{n} \sum_{i=1}^{\ell} x_i x_i^T - I_n$$

W — $n \times n$ -матрица, $w_{jk} = w_{kj}$, $w_{jj} = 0$

Модель ассоциативной памяти $a: x^{(0)} \mapsto x$ реализуется сходящимся итерационным процессом

$$x^{(t+1)} = \text{sign}(Wx^{(t)})$$



Hopfield J. Neural networks and physical systems with emergent collective computational abilities. 1982.

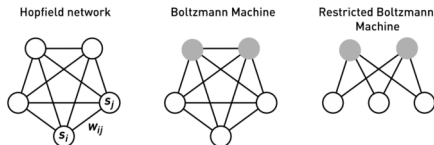
От машин Больцмана к глубоким сетям доверия

Критерий: минимум энергии (Boltzmann Machine)

$$E(x) = -\frac{1}{2}x^T W x - x^T a \rightarrow \min_x$$

Критерий: минимум энергии (Restricted Boltzmann Machine)

$$E(x, h) = -\frac{1}{2}x^T W h - x^T a - h^T b \rightarrow \min_{x, h}$$



Ackley, *Hinton*, Sejnowski. A learning algorithm for Boltzmann machines. 1985.

Hinton G.E., Osindero S., Teh Y.W. A fast learning algorithm for deep belief nets. 2006.

Hinton G.E., Salakhutdinov R. A better way to pretrain deep Boltzmann machines. 2012.

Построение автокодировщика — задача обучения без учителя

$X^\ell = \{x_1, \dots, x_\ell\}$ — обучающая выборка

$f: X \rightarrow Z$ — кодировщик (encoder), кодовый вектор $z = f(x, \alpha)$

$g: Z \rightarrow X$ — декодировщик (decoder), реконструкция $\hat{x} = g(z, \beta)$

Суперпозиция $\hat{x} = g(f(x))$ должна восстанавливать исходные x_i :

$$\mathcal{L}_{AE}(\alpha, \beta) = \sum_{i=1}^{\ell} \mathcal{L}(g(f(x_i, \alpha), \beta), x_i) \rightarrow \min_{\alpha, \beta}$$

Квадратичная функция потерь: $\mathcal{L}(\hat{x}, x) = \|\hat{x} - x\|^2$

Пример 1. Линейный автокодировщик: $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$

$$f(x, A) = \underset{m \times n}{A} x, \quad g(z, B) = \underset{n \times m}{B} z$$

Пример 2. Двухслойная сеть с функциями активации σ_f, σ_g :

$$f(x, A) = \sigma_f(Ax + a), \quad g(z, B) = \sigma_g(Bz + b)$$

Способы использования автокодировщиков

- Генерация признаков (feature generation)
- Снижение размерности (dimensionality reduction)
- Сжатие данных с минимальными потерями точности
- Более эффективное решение задач обучения с учителем в новом признаковом пространстве
- Обучаемая векторизация объектов, встраиваемая в более глубокие нейросетевые архитектуры
- Послойное предобучение многослойных сетей
- Генерация синтетических объектов, похожих на реальные

Rumelhart, Hinton, Williams. Learning Internal Representations by Error Propagation. 1986.

David Charlet et al. A practical tutorial on autoencoders for nonlinear feature fusion: taxonomy, models, software and guidelines. 2018.

Вариационный автокодировщик (Variational AE)

Строится генеративная модель, способная породить новые объекты x , похожие на объекты выборки $X^\ell = \{x_1, \dots, x_\ell\}$

$q_\alpha(z|x)$ — вероятностный кодировщик с параметром α

$p_\beta(\hat{x}|z)$ — вероятностный декодировщик с параметром β

Максимизация нижней оценки лог-правдоподобия:

$$\begin{aligned} \mathcal{L}_{VAE}(\alpha, \beta) &= \sum_{i=1}^{\ell} \log p(x_i) = \sum_{i=1}^{\ell} \log \int q_\alpha(z|x_i) \frac{p_\beta(x_i|z)p(z)}{q_\alpha(z|x_i)} dz \geq \\ &\geq \sum_{i=1}^{\ell} \int q_\alpha(z|x_i) \log \frac{p_\beta(x_i|z)p(z)}{q_\alpha(z|x_i)} dz = \\ &= \sum_{i=1}^{\ell} \int q_\alpha(z|x_i) \log p_\beta(x_i|z) dz - \text{KL}(q_\alpha(z|x_i) \parallel p(z)) \rightarrow \max_{\alpha, \beta} \end{aligned}$$

D.P.Kingma, M.Welling. Auto-encoding Variational Bayes. 2013.

C.Doersch. Tutorial on variational autoencoders. 2016.

Вариационный автокодировщик (Variational AE)

Оптимизационная задача для вариационного автокодировщика:

$$\sum_{i=1}^{\ell} \underbrace{E_{z \sim q_{\alpha}(z|x_i)} \log p_{\beta}(x_i|z)}_{\substack{\text{качество реконструкции} \\ \approx \log p_{\beta}(x_i|z), z \sim q_{\alpha}(z|x_i)}} - \underbrace{\text{KL}(q_{\alpha}(z|x_i) \parallel p(z))}_{\text{регуляризатор по } \alpha} \rightarrow \max_{\alpha, \beta}$$

где $p(z)$ — априорное распределение, обычно $\mathcal{N}(0, \sigma^2 I)$

Репараметризация $q_{\alpha}(z|x_i)$: $z = f(x_i, \alpha, \varepsilon)$, $\varepsilon \sim \mathcal{N}(0, I)$

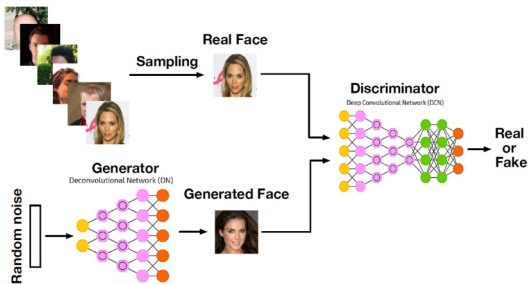
Метод стохастического градиента:

- сэмплировать $x_i \sim X^{\ell}$, $\varepsilon \sim \mathcal{N}(0, I)$, $z = f(x_i, \alpha, \varepsilon)$
- градиентный шаг:
 $\alpha := \alpha + h \nabla_{\alpha} [\log p_{\beta}(x_i|f(x_i, \alpha, \varepsilon)) - \text{KL}(q_{\alpha}(z|x_i) \parallel p(z))];$
 $\beta := \beta + h \nabla_{\beta} [\log p_{\beta}(x_i|z)];$

Генерация похожих объектов: $x \sim p_{\beta}(x|f(x_i, \alpha, \varepsilon))$, $\varepsilon \sim \mathcal{N}(0, I)$

Генеративная состязательная сеть (Generative Adversarial Net)

Генератор $G(z)$ учится порождать объекты x из шума z
Дискриминатор $D(x)$ учится отличать их от реальных объектов



Antonia Creswell et al. Generative Adversarial Networks: an overview. 2017.
Zhengwei Wang, Qi She, Tomas Ward. Generative Adversarial Networks: a survey and taxonomy. 2019.
Chris Nicholson. A Beginner's Guide to Generative Adversarial Networks.
<https://pathmind.com/wiki/generative-adversarial-network-gan>. 2019.

Постановка задачи GAN

Дано: выборка объектов $\{x_i\}_{i=1}^m$ из X

Найти:

вероятностную генеративную модель $G(z, \alpha): x \sim p(x|z, \alpha)$

вероятностную дискриминативную модель $D(x, \beta) = p(1|x, \beta)$

Критерий:

обучение дискриминативной модели D :

$$\sum_{i=1}^m \ln D(x_i, \beta) + \ln(1 - D(G(z_i, \alpha), \beta)) \rightarrow \max_{\beta}$$

обучение генеративной модели G по случайному шуму $\{z_i\}_{i=1}^m$:

$$\sum_{i=1}^m \ln(1 - D(G(z_i, \alpha), \beta)) \rightarrow \min_{\alpha}$$

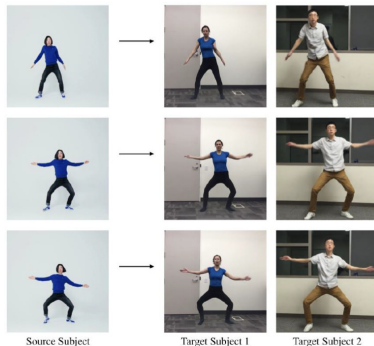
Примеры GAN для синтеза изображений и видео



(d) input image

(e) output 3d face

(f) textured 3d face



Source Subject

Target Subject 1

Target Subject 2

Chuan Li, Michael Wand. Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks. 2016.

Xiaoxing Zeng, Xiaojiang Peng, Yu Qiao. DF2Net: A Dense Fine Finer Network for Detailed 3D Face Reconstruction. ICCV-2019.

Caroline Chan, Shiry Ginosar, Tinghui Zhou, Alexei A. Efros. Everybody Dance Now. ICCV-2019.

Полносвязная нейронная сеть с L слоями

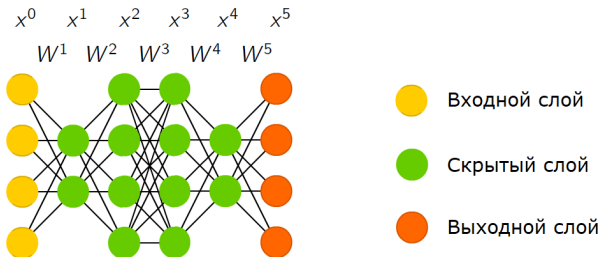
Архитектура сети: H_l — число нейронов в l -м слое, $l = 1, \dots, L$

$x^0 = x = (f_j(x))_{j=0}^n$ — вектор признаков на входе сети, $H_0 = n$

$x^l = (x_h^l)_{h=0}^{H_l}$ — вектор признаков на выходе l -го слоя, $x_0^l = -1$

$x^L = a(x) = (a_m(x))_{m=1}^M$ — выходной вектор сети, $H_L = M$

$W^l = (w_{kh}^l)$ — матрица весов l -го слоя, размера $(H_{l-1} + 1) \times H_l$



Метод стохастического градиента SG (Stochastic Gradient)

Критерий: минимум средних потерь на обучающей выборке:

$$Q(w) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathcal{L}(w, x_i) \rightarrow \min_w.$$

Вход: выборка $(x_i, y_i)_{i=1}^{\ell}$; темп обучения η ; параметр λ ;

Выход: вектор весов всех слоёв $w = (W^1, \dots, W^L)$;

инициализировать веса w и текущую оценку $Q(w)$;

повторять

выбрать объект x_i из X^{ℓ} (например, случайно);

вычислить потерю $\mathcal{L}_i := \mathcal{L}(w, x_i)$;

градиентный шаг: $w := w - \eta \nabla \mathcal{L}(w, x_i)$;

оценить значение функционала: $Q := (1 - \lambda)Q + \lambda \mathcal{L}_i$;

пока значение Q и/или веса w не стабилизируются;

Задача дифференцирования суперпозиции функций

Вычисление сети по входному вектору x , рекуррентно по слоям:

$$x_h^l = \sigma_h^l(S_h^l), \quad S_h^l = \sum_{k=0}^{H_{l-1}} w_{kh}^l x_k^{l-1}, \quad h = 1, \dots, H_l, \quad l = 1, \dots, L,$$

то же самое в матричной записи: $x^l = \sigma^l(W^l x^{l-1})$.

Функция потерь на объекте x_i (например, квадратичная):

$$\mathcal{L}_i(w) = \sum_{m=1}^M \frac{1}{2} (a_m(x_i, w) - y_{im})^2$$

По формуле дифференцирования суперпозиции функций:

$$\frac{\partial \mathcal{L}_i(w)}{\partial w_{kh}^l} = \frac{\partial \mathcal{L}_i(w)}{\partial x_h^l} \frac{\partial x_h^l}{\partial w_{kh}^l}, \quad k = 0, \dots, H_{l-1}, \quad h = 1, \dots, H_l$$

Rumelhart, *Hinton*, Williams. Learning internal representations by error propagation. 1986.

Рекуррентное вычисление частных производных

Найдём сначала частные производные $\mathcal{L}_i(w)$ по $x_m^L \equiv a_m(x_i, w)$:

$$\frac{\partial \mathcal{L}_i(w)}{\partial x_m^L} = a_m(x_i, w) - y_{im} \equiv \varepsilon_{im}^L;$$

для квадратичной функции потерь это *ошибка выходного слоя*.

Частные производные по x_h^l будем вычислять рекуррентно, по уровням справа налево, $l = L, \dots, 2$:

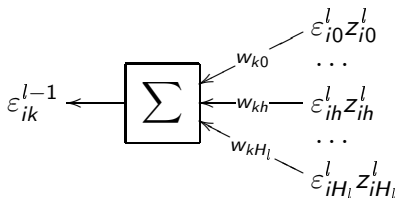
$$\frac{\partial \mathcal{L}_i(w)}{\partial x_k^{l-1}} = \sum_{h=0}^{H_l} \frac{\partial \mathcal{L}_i(w)}{\partial x_h^l} \underbrace{(\sigma_h^l)'(S_{ih}^l)}_{z_{ih}^l} w_{kh}^l = \sum_{h=0}^{H_l} \varepsilon_{ih}^l z_{ih}^l w_{kh}^l = \varepsilon_{ik}^{l-1}$$

— формально назовём это *ошибкой скрытого слоя*.

Замечание: функция активации σ_h^l и её производная $(\sigma_h^l)'$ вычисляются в одной и той же точке $S_{ih}^l = \sum_{k=0}^{H_{l-1}} w_{kh}^l x_{ik}^{l-1}$

Быстрое вычисление градиента

Рекуррентная формула записана так, будто сеть запускается «задом наперёд», чтобы вычислять ε_{ik}^{l-1} по ε_{ih}^l :



Теперь, имея частные производные $\mathcal{L}_i(w)$ по всем x_h^l , легко найти градиент $\mathcal{L}_i(w)$ по вектору весов w :

$$\frac{\partial \mathcal{L}_i(w)}{\partial w_{kh}^l} = \frac{\partial \mathcal{L}_i(w)}{\partial x_h^l} \frac{\partial x_h^l}{\partial w_{kh}^l} = \varepsilon_{ih}^l z_{ih}^l x_{ik}^{l-1}$$

Алгоритм обратного распространения ошибки BackProp

Вход: выборка $(x_i, y_i)_{i=1}^{\ell}$, архитектура $(H_l)_{l=1}^L$, параметры η, λ ;

Выход: вектор весов всех слоёв $w = (W^1, \dots, W^L)$;

инициализировать веса w ;

повторять

выбрать объект x_i из X^{ℓ} (например, случайно);

прямой ход: **для всех** $l = 1..L, h = 1..H_l$

$$S_{ih}^l := \sum_{k=0}^{H_{l-1}} w_{kh}^l x_{ik}^{l-1}; \quad x_{ih}^l := \sigma_h^l(S_{ih}^l); \quad z_{ih}^l := (\sigma_h^l)'(S_{ih}^l);$$

$$\varepsilon_{hi}^L := \frac{\partial \mathcal{L}_i(w)}{\partial x_h^L}, \quad h = 1..H_L;$$

обратный ход: **для всех** $l = L..2, k = 0..H_{l-1}$

$$\varepsilon_{ik}^{l-1} = \sum_{h=0}^{H_l} \varepsilon_{ih}^l z_{ih}^l w_{kh}^l;$$

градиентный шаг: **для всех** $l = 1..L, k = 0..H_{l-1}, h = 1..H_l$

$$w_{kh}^l := w_{kh}^l - \eta \varepsilon_{ih}^l z_{ih}^l x_{ik}^{l-1};$$

пока значения Q и/или веса w не стабилизируются;

Алгоритм BackProp: преимущества и недостатки

Преимущества:

- время вычисления градиента $O(\dim w)$ вместо $O(\dim^2 w)$
- обобщение на любые σ , \mathcal{L} , любые архитектуры
- возможность динамического (поточкового) обучения
- возможно сублинейное обучение на больших выборках (когда части объектов x_i уже достаточно для обучения)
- возможно распараллеливание

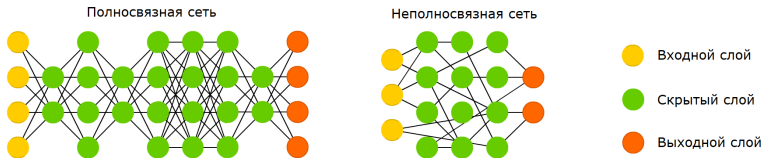
Недостатки:

- медленная сходимость, застревание в локальных экстремумах
- возможны «затухание» и «взрывы» градиентов
- возможно переобучение
- подбор комплекса эвристик является искусством

Глубокие нейронные сети (Deep Neural Network, DNN)

1965: первые глубокие нейронные сети

2012: свёрточная сеть для классификации изображений AlexNet



- *Архитектура сети* — структура слоёв и связей между ними, позволяющая наделять DNN нужными свойствами
- DNN позволяют принимать на входе и генерировать на выходе *сложно структурированные данные*

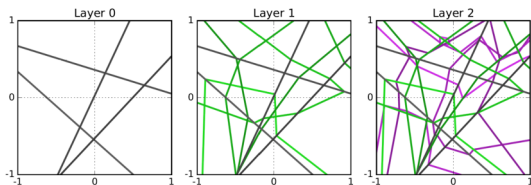
Ива́хненко А. Г., Лапа В. Г. Кибернетические предсказывающие устройства. 1965
Krizhevsky A., Sutskever I., *Hinton G.* ImageNet classification with deep convolutional neural networks. 2012.

Глубина важнее ширины

A_{LH}^n — семейство полносвязных многослойных сетей $a(x, w)$:
 n признаков, L слоёв, H нейронов в каждом слое, $x \in \mathbb{R}^n$,
 функции активации кусочно-линейные: ReLU, hard-tanh и т.п.

Мера разнообразия семейства A_{LH}^n — максимальное число
 участков линейности $a(x, w)$ — выпуклых многогранников в \mathbb{R}^n .

Пример. Участки линейности, $n = 2$, $L = 3$, $H = 4$:



Теорема. Разнообразие семейства A_{LH}^n растёт как $O(H^{nL})$.

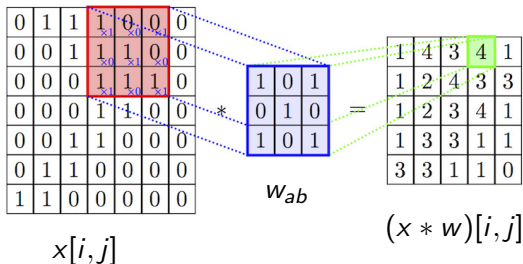
M. Raghu et al. On the Expressive Power of Deep Neural Networks, 2016.

Свёрточный слой нейронов (convolution layer)

$x[i, j]$ — исходные признаки, пиксели $n \times m$ -изображения
 w_{ab} — ядро свёртки, $a = -A, \dots, +A$, $b = -B, \dots, +B$

Неполносвязный свёрточный нейрон с $(2A + 1)(2B + 1)$ весами:

$$(x * w)[i, j] = \sum_{a=-A}^A \sum_{b=-B}^B w_{ab} x[i + a, j + b]$$



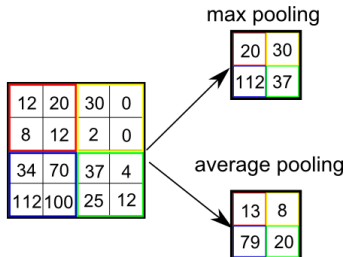
Объединяющий слой нейронов (pooling layer)

Объединяющий нейрон — это необучаемая свёртка с шагом $h > 1$, агрегирующая данные прямоугольной области $h \times h$:

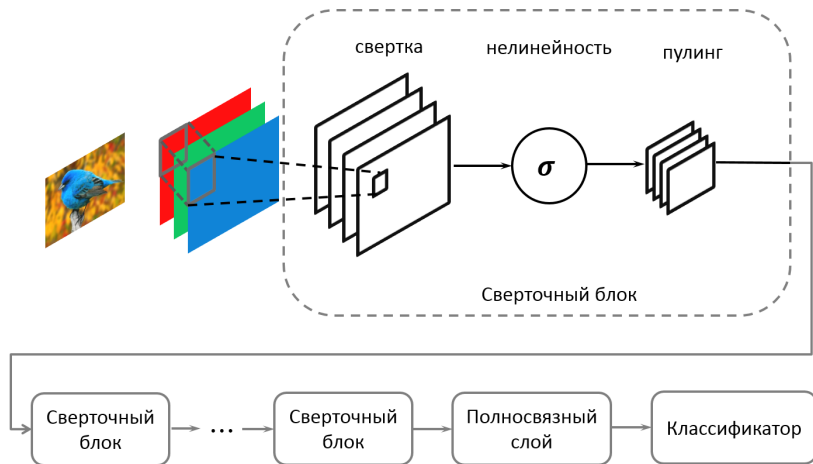
$$y[i, j] = F(x[hi, hj], \dots, x[hi + h - 1, hj + h - 1]),$$

где F — агрегирующая функция: max, average и т.п.

max-pooling позволяет обнаружить элемент в любой из ячеек



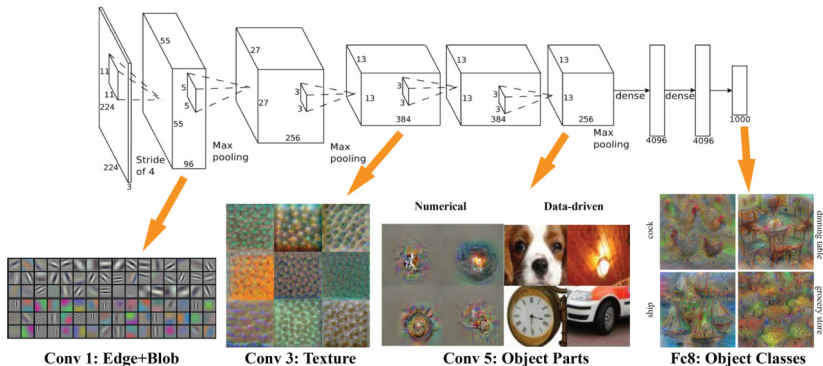
Стандартная схема сверточной сети (Convolutional NN)



Yann LeCun et al. Learning algorithms for classification: A comparison on handwritten digit recognition. 1995

Свёрточная сеть обучается извлечению признаков

Чем выше слой, тем более крупные и сложные элементы изображений он способен распознавать



Krizhevsky A., Sutskever I., Hinton G. ImageNet classification with deep convolutional neural networks. 2012.

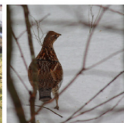
ImageNet — большая выборка размеченных изображений



flamingo



cock



ruffed grouse



quail



partridge

..



Egyptian cat



Persian cat



Siamese cat

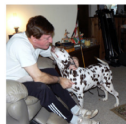


tabby



lynx

..



dalmatian



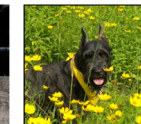
keeshond



miniature schnauzer



standard schnauzer

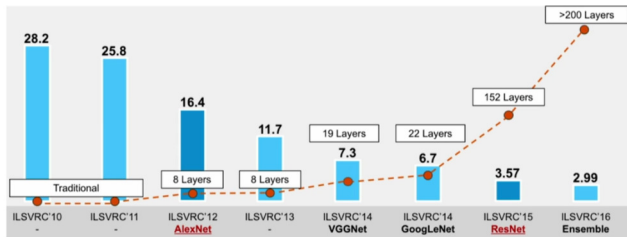


giant schnauzer

Li Fei-Fei et al. ImageNet: A large-scale hierarchical image database. 2009.

Li Fei-Fei et al. Construction and analysis of a large scale image ontology. 2009.

Глубокие свёрточные сети для классификации изображений



Старт в 2009. Человеческий уровень ошибок 5% пройден в 2015

Свёрточная сеть **AlexNet**:

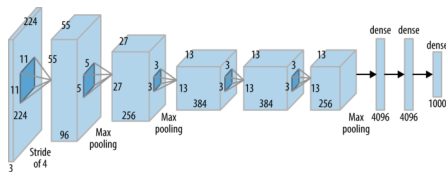
+ ReLU + Dropout

+ 60M параметров

+ пополнение выборки

+ подбор размеров слоёв

+ GPU



Krizhevsky A. et al. ImageNet classification with deep convolutional neural networks. 2012.

Сеть со сквозными связями ResNet (Residual Neural Network)

Сквозная связь (skip connection)

слоя ℓ с предшествующим слоем $\ell - d$:

$$x_\ell = \sigma(Wx_{\ell-1}) + x_{\ell-d}$$

Регуляризирующее воздействие:

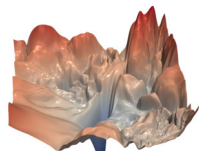
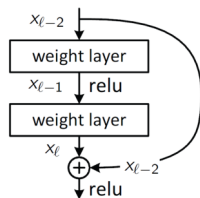
слой ℓ строит не новый вектор x_ℓ ,
а его малое приращение $x_\ell - x_{\ell-d}$

Нетривиальный результат:

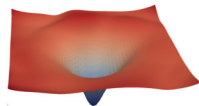
упрощается ландшафт оптимизируемого
критерия, устраняются локальные
экстремумы и седловые точки

Kaiming He et al. Deep residual learning for image
recognition. 2015

Hao Li et al. Visualizing the loss landscape of
neural nets. 2018



без сквозных связей



со сквозными связями

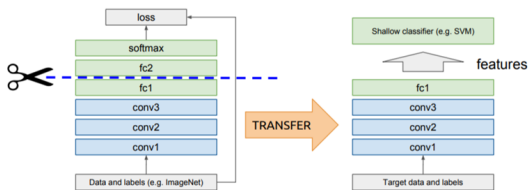
Предобучение (pre-training), перенос обучения (transfer learning)

Обучение модели векторизации $z = f(x, \alpha)$ на выборке $\{x_i\}_{i=1}^{\ell}$:

$$\sum_{i=1}^{\ell} \mathcal{L}_i(g(f(x_i, \alpha), \beta)) \rightarrow \min_{\alpha, \beta}$$

Обучение целевой модели $y = g(z, \beta)$ на малых данных:

$$\sum_{i=1}^m \mathcal{L}'_i(g'(f(x'_i, \alpha), \beta')) \rightarrow \min_{\beta'}$$

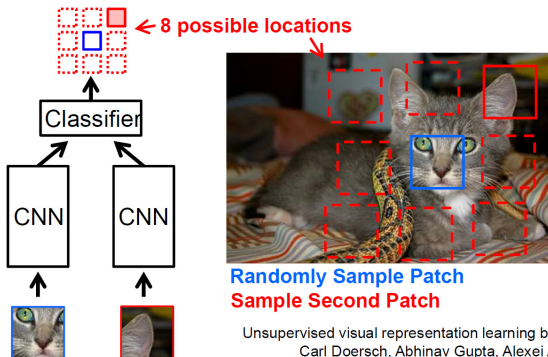


Sinno Jialin Pan, Qiang Yang. A Survey on Transfer Learning. 2009

J. Yosinski et al. How transferable are features in deep neural networks? 2014.

Самостоятельное обучение (self-supervised learning)

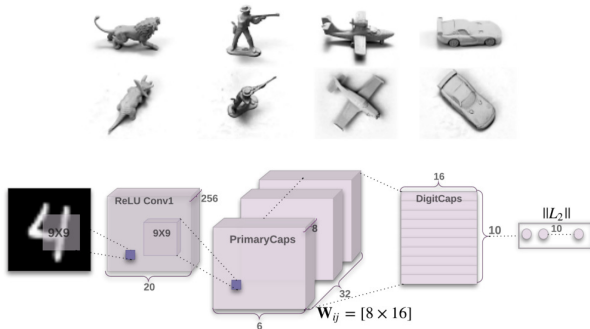
Модель векторизации $z = f(x, \alpha)$ обучается предсказывать взаимное расположение пар фрагментов одного изображения



Преимущество: сеть выучивает векторные представления объектов без размеченной обучающей выборки (без ImageNet).

Капсульная нейронная сеть Хинтона

Мотивация: микроколоники в коре мозга способны узнавать объекты независимо от расположения, освещения. Однако max-pooling на это не способен.



Sabour S., Frosst N., **Hinton G.E.** Dynamic Routing Between Capsules. 2017.
Hinton G.E., Sabour S., Frosst N. Matrix capsules with EM routing. 2018.

Три этапа развития технологий машинного обучения:

- 1 вектор \rightarrow скаляр
- 2 структура \rightarrow вектор \rightarrow скаляр
- 3 структура \rightarrow вектор \rightarrow структура

Достижения нобелевских лауреатов Джона Хопфилда и Джеффри Хинтона, способствовали переходу к этапам 2 и 3:

- сети ассоциативной памяти
- метод обратного распространения ошибок
- генеративные сети и автокодировщики
- обучаемая векторизация данных, сеть AlexNet
- свёрточные и капсульные нейронные сети

- *Кевин Мэрфи*. Вероятностное машинное обучение. Введение. 2022.
- *Визильтер Ю. В.* От слабого ИИ к общему универсальному интеллекту (обзор тенденций 2020-2023). Семинар РАИИ и ФИЦ ИУ РАН «Проблемы искусственного интеллекта» 31-01-2024
<https://rutube.ru/video/2aad53ec833f19918c1593398a2a1b88>
- Не пропустите открытие тысячелетия! Vital Math, 13-01-2024,
<https://www.youtube.com/watch?v=JZjH0it9Jyg>
- Report: AI Decrypted: A Guide for Navigating AI Developments in 2024, Навигатор по ИИ-ландшафту от Dentons Global Advisors, 24-01-2024
<https://www.albrightstonebridge.com/news/report-ai-decrypted-guide-navigating-ai-developments-2024>
- *Воронцов К. В.* Лекции по машинному обучению. МФТИ, МГУ. www.MachineLearning.ru, User:Vokov, 2004–2024.
<https://bit.ly/ML-Vorontsov>
- *Гарбук С. В., Губинский А. М.* Искусственный интеллект в ведущих странах мира: стратегии развития и военное применение. 2020.
- *Шумский С. А.* Машинный интеллект. РИОР ИНФРА-М, 2020.