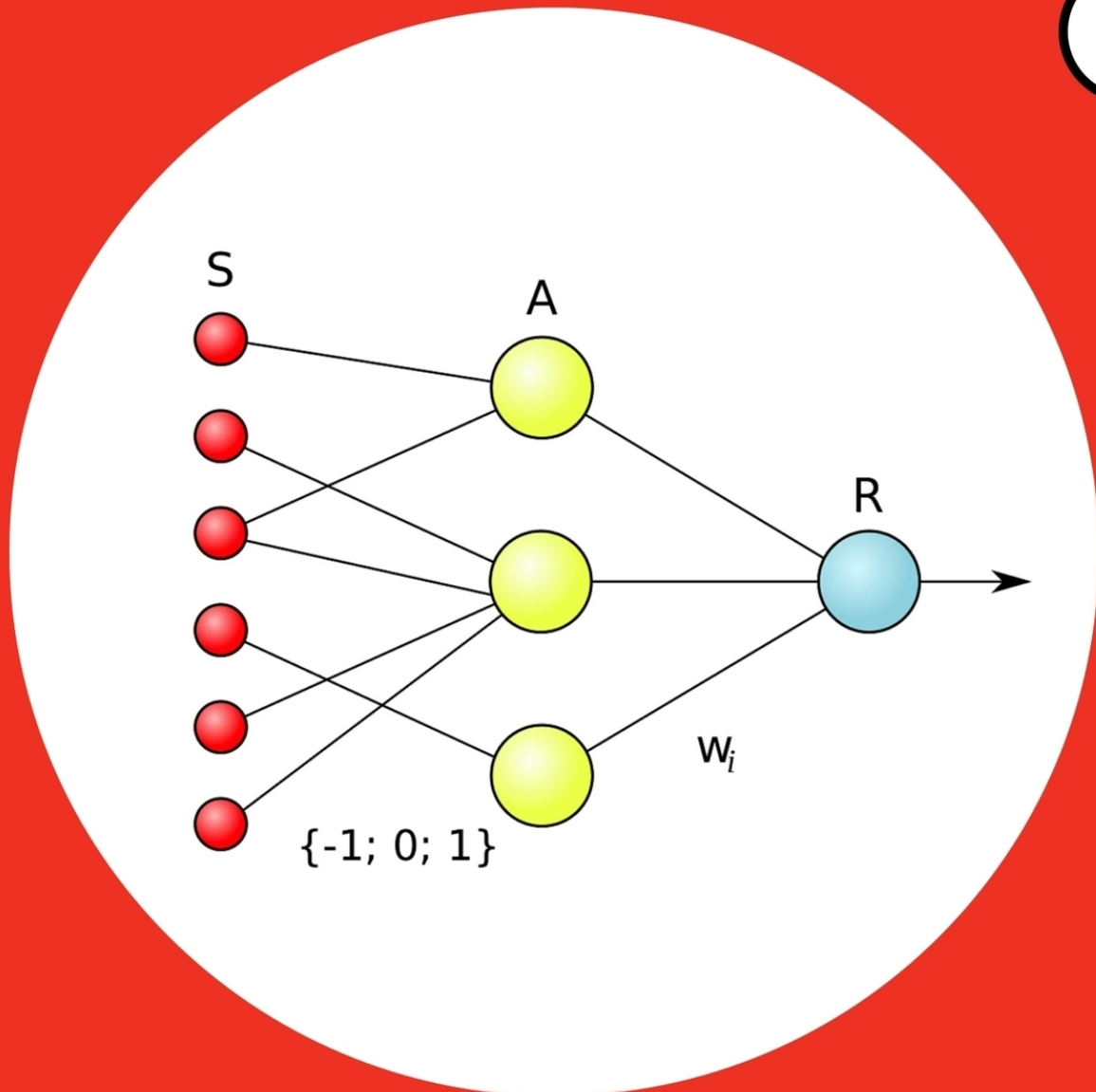


Константин Берлинский

12+



# Основы нейросетей

С примерами на C#, Keras.NET, Numpy.NET и Python

Константин Берлинский

**Основы нейросетей**

«ЛитРес: Самиздат»

2020

**Берлинский К. К.**

Основы нейросетей / К. К. Берлинский — «ЛитРес: Самиздат»,  
2020

ISBN 978-5-532-06726-4

Цель этой книги - максимально быстро научиться применять нейросети (НС) для решения задач. Вы научитесь: создавать и обучать НС; обрабатывать изображения, цифры и тексты; встраивать НС в реальное приложение через REST API; выбирать архитектуру НС: число слоев, нейронов, функции активации, оптимизаторы, коэффициенты скорости обучения, функции ошибки, эпохи, батчи; работать с НС на основе многослойного перцептрона, свёрточными и рекуррентными; и даже сделать НС без ML-библиотек на чистом C#!

ISBN 978-5-532-06726-4

© Берлинский К. К., 2020

© ЛитРес: Самиздат, 2020

# Содержание

1 Где можно поучиться AI?	5
2 Венчурная аналитика AI-проектов	12
3 Погружение в ИИ	19
4 Распознавание цифр без нейросетей	23
5 Нейросеть Matt Mazur	31
Конец ознакомительного фрагмента.	34

## 1 Где можно поучиться AI?

Чтож. Теперь, когда пассивный доход от издания книг (это еще не сарказм) сделал избыточной работу в найме (а вот и он), приступаю ко 2-му этапу диджитал-трансформации.

Посоветуйте, где лучше поучиться AI, ML, нейронным сетям, обработке BigData, AR/VR, компьютерному зрению и всему такому. Самостоятельно изучать тему неэффективно – быстро выгораю, становится скучно, нужен надсмотрщик с плеткой.

Интересуют практические знания. Чтобы после курса уметь писать приложения вида PRISM или MSQRD. Ну или которые смогут отличить кошку от собаки, написать новую шутку или предсказать курс \$ на неделю вперед.

В любом случае интересны проекты, основанные на легко доступных больших данных. Это данные в личном смартфоне (звонки, микрофон, геопозиция, камера), на своем ПК (файлы, почта, интересы), Интернет (вики, карты, анекдоты, ютуб, соцсети?), массовая литература. Какие еще открытые бигдата есть?

Что точно не интересно – делать проект, заточенный под данные или API определенной компании. Это телекомы, банки, логистика, магазины, интернет-холдинги или соцсети.

Бигдата у них, конечно, есть, но:

1) Фиг они ими поделятся (если это вообще законно – выдавать наружу данные, даже обезличенные).

2) Фиг они что-то у тебя купят. Как-то не принято это здесь. Да и бизнес-смысла в этом нет. Проще тупо украсть наработки, сделать свой велосипед или максимум предложить работу команде. Даже в случае успешного пилота с корпорацией это довольно слабая переговорная позиция – когда данные у компании, а у тебя лишь обученная на их основе нейросеть (алгоритм обработки данных) и проведенный кастдев. Они без тебя жить смогут, а ты без них нет.

Есть конечно кейс – знать, что нужно большой компании, сделать это на стороне и продать наработки компании. Бобук говорил, что он так сделал механизм распознавания людей на фотках и продал это VK/OK. Им это нужно чтобы автоматически тэгать людей. Это повышает вовлечение юзеров и время зависания на сайте. Но это скользкий путь:

1) Неясно, что нужно корпам.

2) Не факт, что они прямо сейчас не делают такую штуку.

3) Не факт, что эту штуку сможешь сделать ты.

4) Не факт, что готовое решение купит корп. Они любят тянуть время (приходите через год, на сегодня все бюджеты освоены).

Думаю, по деньгам у Григория получилось так. Команда 5 чел, ЗП 4 тыс \$/мес, делать такую штуку ~10 мес. Себестоимость 200 тыс \$. Продать корпорации с премией 3-5x можно за 0.6-1 млн \$.

Какие курсы по data science и нейронным сетям науглил и рассматриваю:

1) Машинное обучение и анализ данных – Яндекс, МФТИ, Курсера, 5500 руб/мес, 250 часов.

2) Data science – SkillFactory, 120 тыс руб, 12 мес.

- 3) Data scientist – Нетология, 200 тыс руб, 8 мес.
- 4) Введение в машинное обучение – Яндекс, ВШЭ, Курсера, бесплатно, 35 часов.
- 5) Introduction to Machine Learning – Google, бесплатно, 15 часов.  
<https://developers.google.com/machine-learning/crash-course>
- 6) Ваш вариант?

Месяца 4 назад начинал проходить курс #5 от Гугл, но через 2 недели бросил. Все-таки, мне нужен кто-то, кто будет пинать и мотивировать продолжать обучение.

Что посоветуете?

## Какой курс по AI выбрать?

Анонимный опрос

- 33% 1) Яндекс, МФТИ, Курсера, 5500руб/мес, 250ч  

- 0% 2) SkillFactory, 120тыс руб, 12 мес  

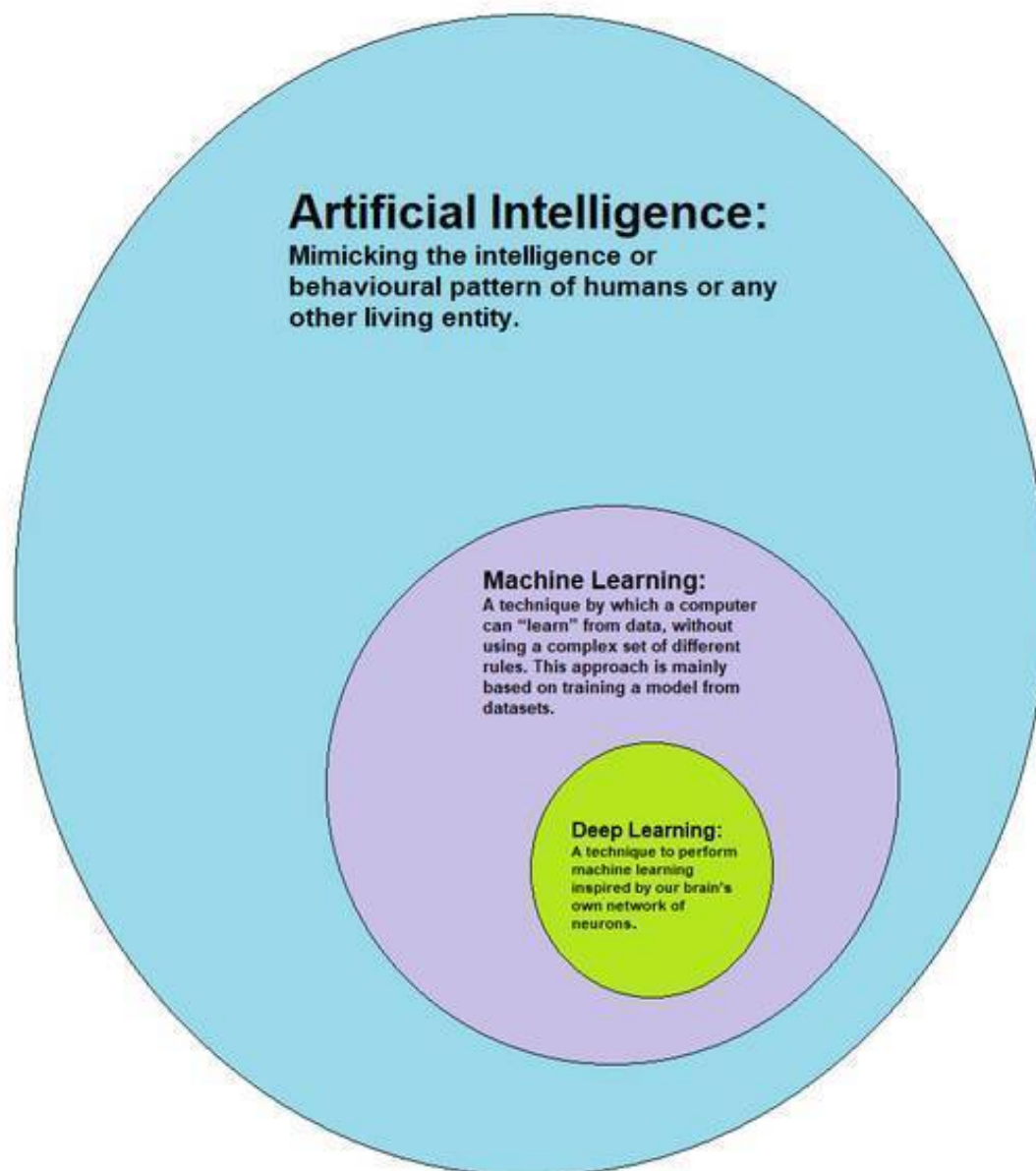
- 0% 3) Нетология, 200тыс руб, 8 мес  

- 33% 4) Яндекс, ВШЭ, Курсера, бесплатно, 35ч  

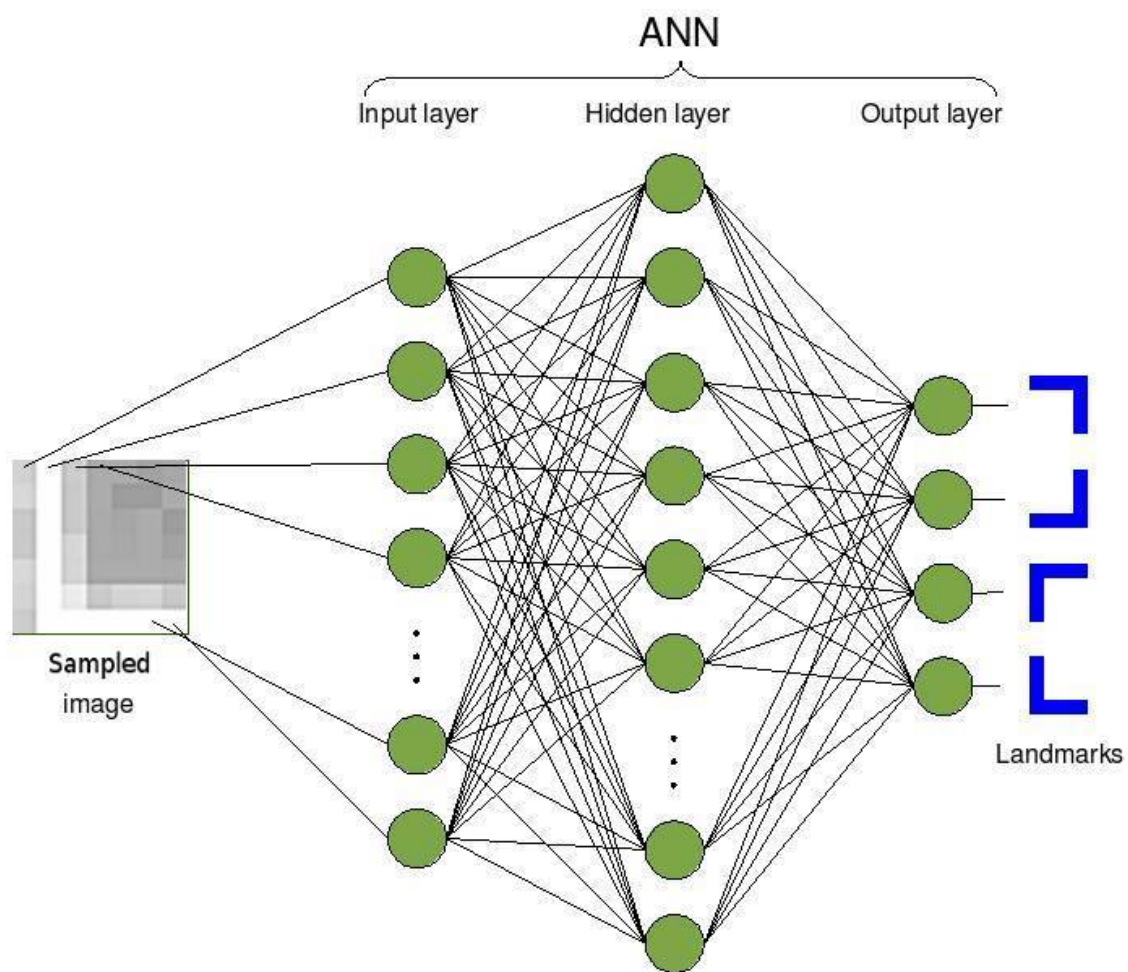
- 33% 5) Гугл, бесплатно, 15ч  
 
- 0% 6) Ваш вариант (напишите в комментариях)  


3 голоса

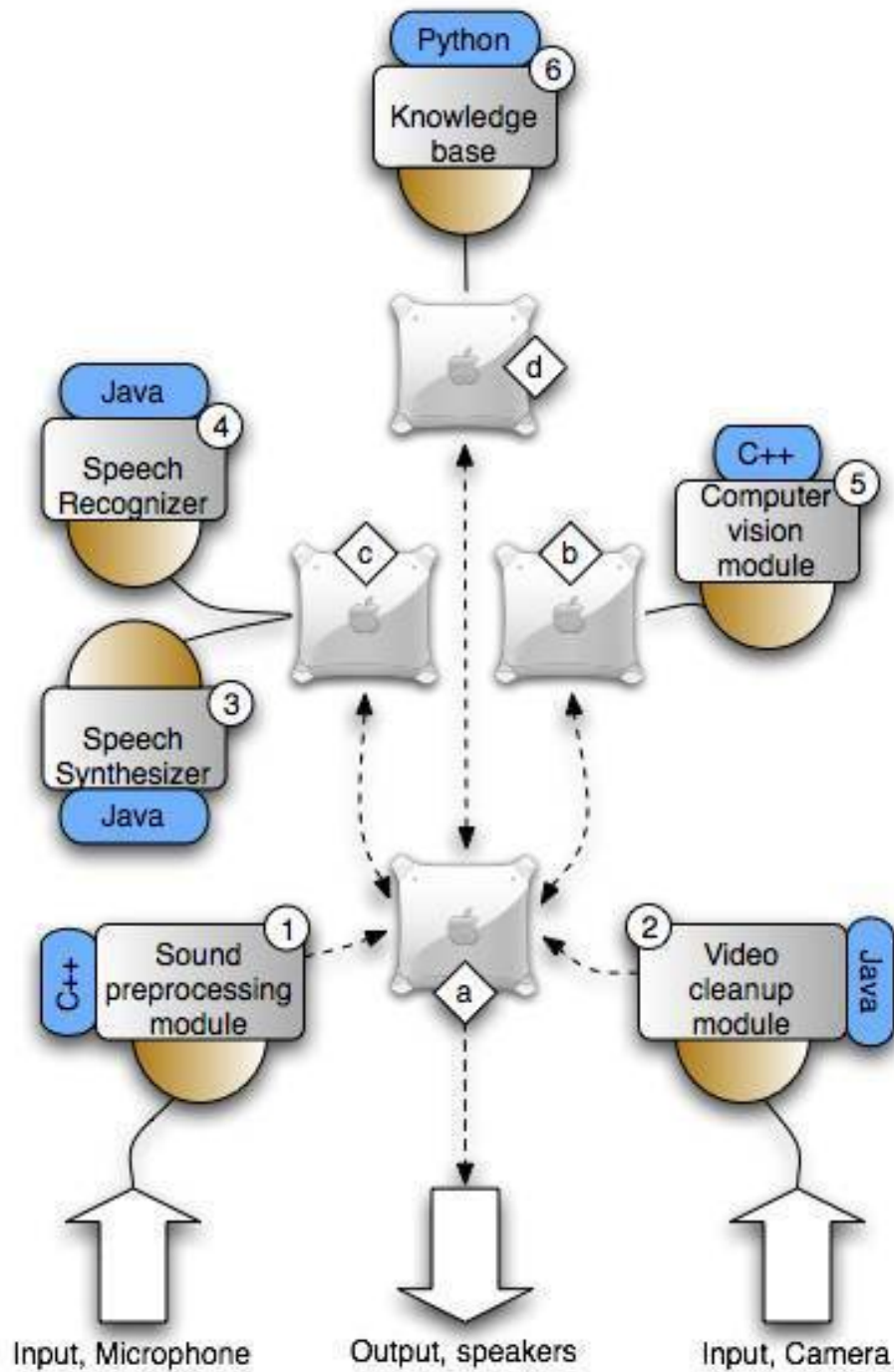
 23 15:53



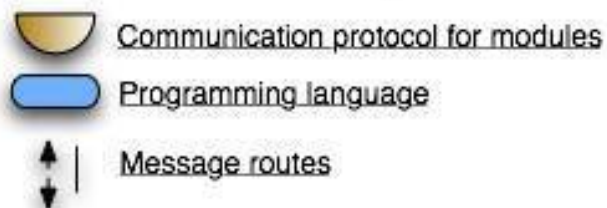
[Источник фото](#)



[Источник фото](#)

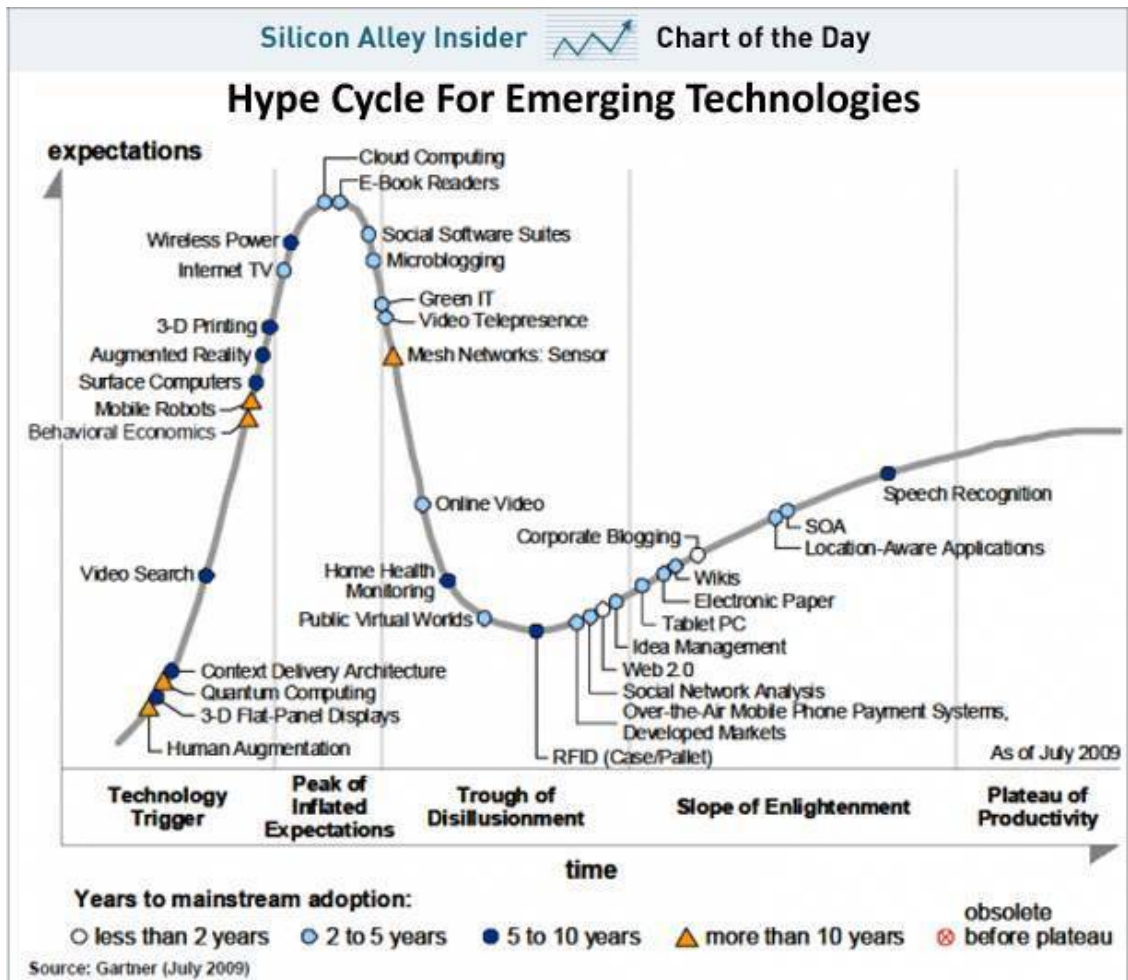


**An example of A.I. systems integration using message protocols and blackboard systems such as OpenAIR and Psyclone**



[Источник фото](#)

Также, см. [https://en.wikipedia.org/wiki/Applications\\_of\\_artificial\\_intelligence](https://en.wikipedia.org/wiki/Applications_of_artificial_intelligence)



[Источник фото](#)

## 2 Венчурная аналитика AI-проектов

Насчет выбора курса по AI (см. предыдущий пост). При выборе темы учебного проекта, думаю, стоит ориентироваться на AI-проекты, получившие инвестиции в последнее время. Ниже последние AI-проекты ФРИИ, YC, 500 Startups и TechStars.

Т.к. дальше будет лонгрид и много описаний проектов, вынесу наверх мысли насчет проектов YC:

1) США новый Вавилон. В последнем наборе Summer 2019 – 175 компаний 27 стран. США, Европа, Азия, ЛатАм, даже Африка. России нет, но есть Украина. Никаких других центров стартапов как-будто нет.

2) Идея брать 7% за \$120К от 175 потенциально прорывных компаний каждые полгода может быть невероятно прибыльной. Чисто по статистике 3-5 единорогов у них получится. Похоже, у YC нет никаких проблем со сбором \$\$\$ на финансирование очередного набора фонда. Возможно в РФ с венчуром проблема не в рынке, недостатке инвесторах, политике и др, а тупо просто нет столько людей-предпринимателей кто хочет сделать свой проект.

3) Похоже, венчур в России в глубокой ж. 11 vs 175 стартапов (ФРИИ vs YC) в последнем наборе. Много стартапов во ФРИИ – унылые локальные проекты без полета фантазии, которые никогда не выйдут на внешний рынок и не станут единорогами. Онлайн ГИБДД, Помощник в исполнении 149-ФЗ по безопасности данных, оборудование для соляных пещер :- (Зачем это вообще?

4) Очень крутые проекты YC с миллионной выручкой, понятной бизнес-моделью, не боятся испачкать руки. Многие проекты для узкой аудитории – мигранты-айтишники, венесуэльцы, наркоманы, женщины, секс-фетишисты, гики.

5) В YC классные описания проектов. В одном абзаце донесена суть, главные цифры, команда, перспективы. Можно за час прочитать описание всех 175 проектов в наборе и принять инвестиционное решение.

6) AI не волшебная пилюля. Во многих проектах AI упомянута только для галочки. Весь цимес в хорошем удовлетворении реального спроса, бизнес-модели и перспективах (растущие тренды, взрывной рост какой-либо характеристики 1-10 тыс раз).

7) Главное в стартапах YC – бабло, а не красивая идея или хайповый набор технологий. Клеить танчики умеют многие, важно осознанно и к месту их клеить.

8) Блин, какие они крутые! Проекты по продаже марихуаны через дроны, создание новых лекарств, VR для хирургов, запись инфо в ДНК, запуск орбитальных спутников – легко. Начитались, твари, Жюль Верна, никаких тормозов! Походу Илон Маск там еще не самый упорный :-)

Итак, AI-проекты ведущих акселераторов.

1) ФРИИ набор #19, 3 Октября 2019 (11 проектов):

<https://www.iidf.ru/media/articles/accelerator/19-nabor-akseleratora/>

Leveli.ng (Казань) – AI-сервис для автоматизации работы с отзывами пользователей в интернете. Сервис автоматически обрабатывает, генерирует и публикует ответы на отзывы по разным сценариям: нивелирование негатива, нейтрализация отзыва, работа с постоянными клиентами.

2) ФРИИ набор #18, 1 Июля 2019 (19 проектов):

<https://www.iidf.ru/media/articles/accelerator/v-18-nabor-akselatora/>

Clover Group (Москва) – разработчик решений в области прогнозного обслуживания для промышленных предприятий с применением технологий искусственного интеллекта, машинного обучения и предиктивной аналитики.

FoxTail (Ростов-на-Дону) – онлайн-платформа для формирования и управления распределенной командой разработки. Заказчики IT-услуг и аутсорсинговые компании благодаря умному поиску, основанному на big data и machine learning, смогут быстро находить подходящих проверенных специалистов, а с помощью инструментов совместной работы управлять проектом.

Music Scan (Москва) – сервис, используя технологии Big Data, помогает правообладателям пресекать нелегальное использование аудио-произведений, увеличивать размер вознаграждений за использование контента и системно выстраивать стратегию продвижения произведений на музыкальном рынке.

Инфобот (Уфа) – телефонный робот, которого не отличить от человека. Инфобот может отвечать на входящие и делать исходящие звонки, полностью распознает человеческую речь и на основании диалога принимает решение о следующем действии.

3) YC Summer 2019 Batch (175 companies):

<https://blog.ycombinator.com/yc-summer-2019-batch-stats/>

<https://techcrunch.com/2019/08/19/all-84-startups-from-y-combinators-s19-demo-day-1/>

<https://techcrunch.com/2019/08/20/here-are-the-82-startups>

Intersect Labs: Intersect Labs is building CoreML for enterprise, letting its customers easily build machine learning models to help make sense of their historical data and deliver insights without having to hire data scientists. The monthly subscription is aiming to deliver a product that doesn't require much technical knowledge. "If you can use a spreadsheet, you can use Intersect Labs."

Traces: As privacy-conscious consumers speak up against the proliferation of facial recognition tech, there's still a clear need for a product that enables smart camera tracking for customers. Traces is building computer vision tracking tech that relies on cues other than facial structure like clothing and size to help customers integrate less invasive tracking tech. It was built by former Ring engineers.

Soteris: Soteris is a startup building machine learning software for insurance pricing. Within six months of their pilot, they already have two insurers under contract, giving them \$500K in guaranteed annual revenue.

Well Principled: This is an AI-driven management consultant that says it wants to "replace MBAs with software." Companies spend \$200 billion on management consultants every year. Well Principled wants to replace that expensive and cumbersome system with its tech that has culled growth and revenue learnings from academic research and turned it into enterprise software. The company wants to eliminate the need for outside consultants by integrating its software into the daily operations of businesses as they launch new products. Well Principled is advised and invested in by early Palantir leaders, and claims \$840,000 ARR from its first Fortune 200 customer.

**Dashblock:** Dashblock creates APIs from any web page using machine learning. Drop in a URL, select the data you want from a page, and it will figure out how to automatically extract it and provide it via API. It has have more than 1,500 users since launching two weeks ago.

**EARTH AI:** This full stack AI-powered mining exploration company built a technology to predict the location of un-mined rare metals. EARTH AI's mission is to improve the efficiency of mineral exploration to provide enough metals and minerals for current and future generations. The company predicts where metals may exist, actually mines the ore and then sells it. The team credits themselves with discovering the world's first AI-predicted mineral deposit, and says it has also secured the rights to \$18 billion worth of ore.

**Holy Grail:** Holy Grail says it has built a cheaper and faster way to manufacture batteries. The company is using AI to find the next generation of batteries at what it claims is 1,000x faster and hundreds of million dollars cheaper than traditional R&D processes. Holy Grail's software designs batteries and predicts their performance – then manufactures them using a robot it built. Traditional R&D relies on trial and error and spreadsheets, and this company thinks it can harness AI to “do something good for the world while also making money.”

**Zenith:** This company is building a new virtual world that blends AI, VR and its backend tech to immerse users in new lives online. Zenith, which raised \$120,000 on Kickstarter in one week, is the first cross platform world to exist on VR desktop and console. Essentially every screen you own is a window into their world. The company plans to monetize by taking cuts of every item bought or sold on their platform, like property and clothing. The founders have worked at Google and Unity, and co-produced with Oculus.

**Lofty AI:** Lofty AI is building what they claim to be the first reliable method for tracking neighborhood demand to help real estate investors make more informed investment decisions. Lofty AI recommends properties to investors and if the investors decide to purchase, they enter into a contract that gives them 20% of the profit. However, if the value of a property goes down, Lofty says they will cover all of the investor's losses.

**Treble.ai:** This is a customer support platform that lets companies get feedback from users through SMS and WhatsApp. The company describes itself as similar to Qualtrics and Zendesk, but with one big difference: Qualtrics and Zendesk were built for desktop web and email. Treble is built for mobile-first, chat-based communication. Treble says there are 100,000 companies that serve their users through mobile apps, and it wants to be the startup that manages their customer support. The startup scored Colombian logistics unicorn Rappi as their largest customer, and is seeing \$16,000 in MRR.

**Symplex:** The team at Symplex is developing an AI-based doctor that can diagnose you using your smartphone. The startup says they've signed up 15 doctors in the first few weeks, with a goal of expanding into a \$2.6 billion market. Here's how it works: First, you tell Symplex how you're feeling, then, the company's machine learning algorithm gauges your condition and provides a detailed initial diagnosis, which is then stored and saved.

**Percept.AI:** This startup is creating an AI support agent that immediately resolves common customer support tickets. Other solutions can take over three weeks of onboarding, quality is often insufficient and the AIs only end up resolving between 10-30% of tickets. Percept.AI says their tech could work to identify 1.2 billion support tickets that go outstanding. They say they can immediately

resolve up to 50% of tickets without human intervention, what it describes as an exciting \$22 billion market.

4) 500 Startups, Batch #25, JULY 22, 2019:

<https://500.co/meet-the-startups-of-batch-25/>

<https://500.co/latest-ai-applications-from-500s-batch-25/>

<https://500.co/startups/>

Chemtech: An AI-product for manufacturing plant automatization

Curie: A camera-based AI shopping assistant

EINO: An AI platform that produces meaningful predictive and historical insights on localized population movement and their intention in urban areas for enterprise business users

Heartex: Helping companies quickly build AI products and features

InnerTrends: A data science service for SaaS that uncovers deep insights in customer onboarding, retention, and engagement without the need of data scientists

LucidAct Health: An AI assistant for nurses and case managers to help them know what to do faster and eliminate errors

RestAR: 3D capturing and product visualization for e-commerce using AI with any mobile device

Visionful: Connecting smart cities and autonomous vehicles leveraging AI and computer vision to provide full automation for parking and traffic monitoring

5) TechStars, индустрии AI/ML и Analytics:

<https://www.techstars.com/companies-in-program/>

Asgard.ai: We think any sales intelligence solutions with a broad scope might want to acquire us as they have difficulties to distinguish from one to another. They would typically buy our signal detection engine. Currently, we are able to detect 32 kinds of signals. Our goal is to improve it so it can track more specific buying signals for every new customer. And even maybe to design a product that lets the customer easily set up a very relevant signal by giving feedback/interacting on the platform.

FLOD: FLOD develops a multi-sensor based learning algorithm for structural maintenance detection and prediction.

Heuristech: Heuristech is about dealing with pipeline maintenance the right way.

Lightmass Dynamics: An application framework that easily integrates with any software that manipulates n-dimensional data for real-time simulation or visualization of physical forces.

MorphL AI: MorphL drastically reduces the amount of time required to infuse AI into your applications so you can predict user behaviors, enable personalized digital experiences and increase product KPIs.

ReelData AI: AI for aquaculture. Delivers real time weight distributions, feeding analysis, health analysis and growth analysis.

Revelio Labs: Revelio Labs provides an in-depth view into the workforces of companies around the world. We leverage advancements in AI technologies to turn hundreds of millions of online

profiles, resumes, and job postings into clear insights. Our clients include: corporate strategists, HR, VCs, and investors.

Shipright: Shipright helps software businesses track customer feedback in one organized place, so they can build outstanding products.

По-прежнему жду советов какой AI-курс выбрать, отзывы, промокоды и прочее (см. предыдущий пост).

Ссылки:

1) Новые бизнес-модели AI “The New Business of AI (and How It’s Different From Traditional Software)”:

<https://a16z.com>

2) Какой стартап мне запустить завтра?:

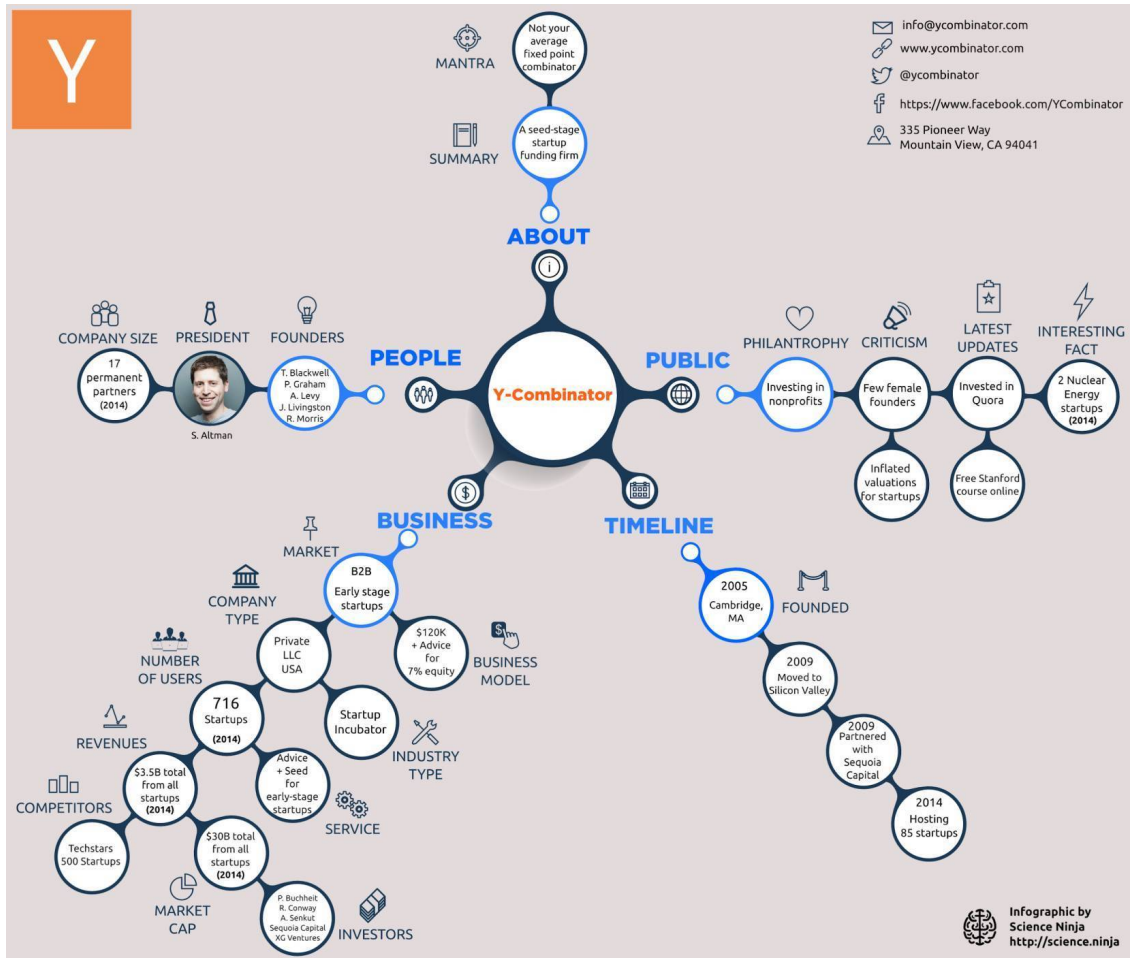
<https://habr.com/ru/company/mailru/blog/481256/>

Суть статьи:

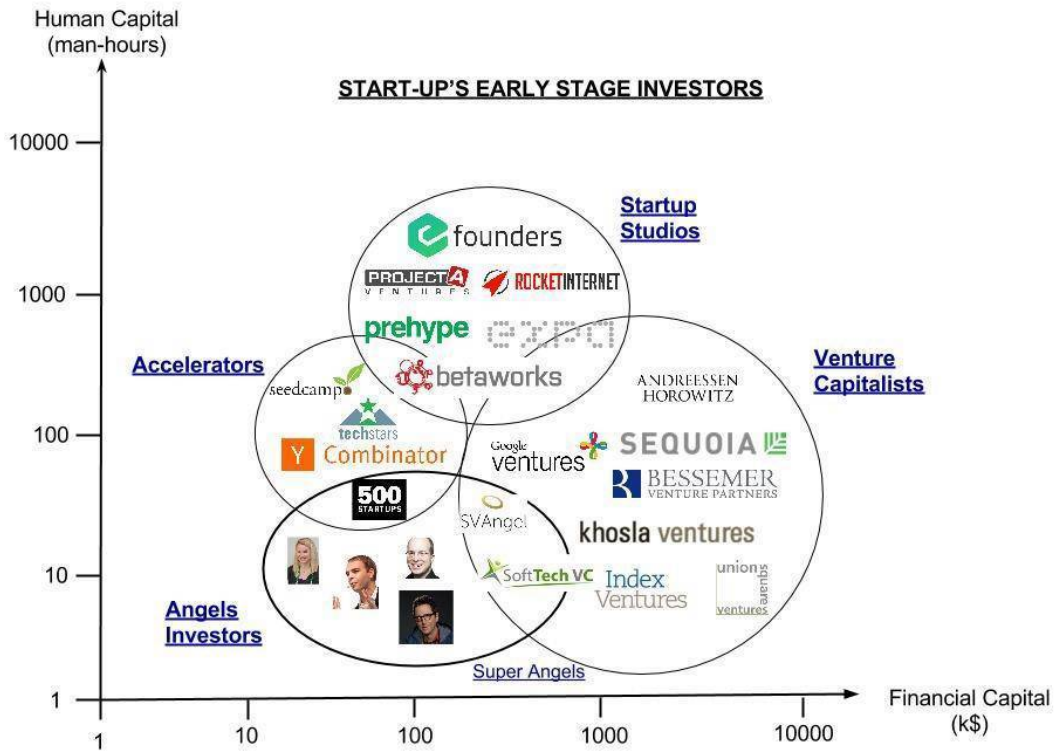
а) AI это принципиально новый инструмент, который позволит взломать кучу рынков и бизнесов. В аджайл, крутой UX, кастдев, метрики, перфоманс маркетинг уже все умеют, а в AI пока нет.

б) AI открывает окно возможностей для дизайнеров. Следующее окно, возможно, будет лет через 10-15 (как ранее был момент доткомов или потом появления смартфонов), поэтому имеет смысл сесть в эту лодку.

в) Сфер применения AI дофига (нет смысла ныть что все хорошие идеи уже реализованы). Это все сферы, где много рутинной повторяющейся работы и много работников. Кассиры, водители, кладовщики, бухгалтеры, трейдеры, колл-центры и даже творческие профессии – журналисты, программисты :grimacing:, музыканты и художники. Попробовать применить AI, всех уволить и перевести на безусловный базовый доход. Ну или появятся новые профессии – разметка данных для нейросетей, колонисты экзопланет или чистое творчество. Раньше технологии развития AI не позволяли заменить человека, сейчас уже позволяют.



Источник фото



[Источник фото](#)

### 3 Погружение в ИИ

Итак, начал разбираться с этим вашим ИИ. Для начала прочел на вики основную статью по нейронным сетям (НС) и связанные с ней темы:

[https://ru.wikipedia.org/wiki/Искусственная\\_нейронная\\_сеть](https://ru.wikipedia.org/wiki/Искусственная_нейронная_сеть)

<https://ru.wikipedia.org/wiki/Перцептрон>

[https://ru.wikipedia.org/wiki/Розенблатт,\\_Фрэнк](https://ru.wikipedia.org/wiki/Розенблатт,_Фрэнк)

[https://ru.wikipedia.org/wiki/Искусственный\\_интеллект](https://ru.wikipedia.org/wiki/Искусственный_интеллект)

[https://ru.wikipedia.org/wiki/Зима\\_искусственного\\_интеллекта](https://ru.wikipedia.org/wiki/Зима_искусственного_интеллекта)

Написано довольно сложным языком, с формулами и фразами вида "... очевидно, что нелинейная характеристика нейрона может быть произвольной: от сигмоидальной до произвольного волнового пакета или вейвлета, синуса или многочлена". Ага, очевидно... для доцентов физмат вузов! Остро не хватает знаний по матану, теорверу, дифурам и прочим задротским ништякам для очкариков и яйцеголовых. Гуманитариям, думаю, читать будет совсем тяжело.

В итоге, как я понял работает НС.

Допустим есть картинка 100\*100 точек. Нужно определить, какая цифра 0..9 на ней изображена.

Делаем 1-ый слой (сенсорный S) – двумерный битовый массив 100\*100. В каждую позицию записываем 0 или 1 в соответствии с тем, закрашена ли точка в той же позиции в файле.

Делаем 2-ой слой (ассоциативный A) – трехмерный числовой массив N\*100\*100. Где N – число образов, которые должна распознавать наша НС. В данном случае N=10, т.к. нам нужно распознать числа 0..9, т.е. всего 10 цифр / образов. В каждую позицию [N, i, j] записываем число x в соответствии с тем, насколько часто встречается закрашенная точка в той же позиции для цифры 0..9.

Как посчитать  $X_{ij}$  для каждого образа? Прогоняем много файлов 100\*100 где по-разному записана цифра 0..9. Если для цифры точка закрашена прибавляем 1 к  $X_{ij}$ , иначе убавляем 1.

Т.е. для цифры 1, например, матрица будет такая, что по краям будут сильно отрицательные значения. А в полоске посередине все более положительные.

И наконец, делаем 3ий слой (реагирующий R) – одномерный числовой массив размерности N. При поступлении картинки для распознавания НС считает число совпадений точек для каждого из образов 0..9. Т.е. для каждой точки смотрим, если она закрашена, то в позицию  $R[n]$  прибавляем значение из 2го слоя  $A[n, i, j]$ .

В итоге получаем для каждого эталонного образа 0..9 финальную оценку  $R[n]$ , которая говорит насколько картинка похожа на эти образы. Для удобства, эти баллы можно нормализовать и получить, например, что на "8" картинка похожа с вероятностью 85%, а на "3" с вероятностью 60% и так далее. Можно настроить выход третьего слоя, чтобы он давал только одно значение – число с максимальной полученной вероятностью в случае если эта вероятность >80%. А если вероятность меньше, то считать что цифра не распознана.

Вот в принципе и всё. Кстати, эта штука с 3-мя слоями называется перцептрон. Он считается простейшим, т.к. у него только один внутренний А-слой.

Можно сделать НС с несколькими внутренними А-слоями. Например, один А-слой распознает отдельно лапы, хвост, морду и шерсть собак. А 2ой А-слой пользуется 1м и знает, какие лапы и другие причуды у определенных пород собак.

Или первый слой распознает буквы, 2-ой слова из букв, а 3ий понимает, полученный текст спам или нет. Плюс 2-ой слой влияет на 1ый. Например, в английском языке есть артикль the и если есть слово из 3х букв и первые 2 – "th", то 3ий слой ненавязчиво советует 2-му слою, что 3-я буква "e". Это рекуррентные НС.

В реальности считаются не "число попаданий точки и умножение на весовой коэффициент", а хитрая математическая магия: сигмоидальная функция, метод градиентного спуска, метод обратного распространения ошибки и прочая нечисть.

Что из этого всего следует:

- 1) для приемлемого качества НС нужно обучать гигантским количеством данных.
- 2) обучающие данные должны быть размеченными, т.е. человек должен просмотреть все эти файлы с изображениями, например, собак и указать, что это именно собака, а не кошка.
- 3) для обучения нужны огромные вычислительные ресурсы на парсинг данных, подсчет коэффициентов и др.
- 4) исходные данные нужно нормализовать – при распознавании текста из графического файла резать текст на слова и буквы, поворачивать изображение, чтобы оно было без наклона, увеличивать или уменьшать, сделать монохромным, убирать шумы и др.
- 5) цикл обучения НС нужно повторять с разными настройками (размер матриц, пороговые вероятности), чтобы получить лучшую НС.
- 6) что хорошо – обучить НС можно один раз на мощностях ИТ-гиганта типа Гугл, а пользоваться результатами смогут все.
- 7) алгоритм распознавания образов аналогично работает для всех других видов распознавания и предсказания – речи, спама, будущего курса акций, прогноза погоды, самоуправляемые авто и др.
- 7) большой минус – НС, в-основном, могут решать всего 2 основных класса задач:
  - а) распознавание – отнести объект к известному классу, т.е. сказать, что на фото число 1 или 2, танк или корова.
  - б) кластеризацию – если классов объектов нет, то создать эти классы, т.е. из миллионов фоток цифр сгруппировать единички в один класс, двойки в другой и т.д.

В вики хорошо описаны этапы решения задач при помощи НС:

- 1) Сбор данных для обучения;
- 2) Подготовка и нормализация данных;
- 3) Выбор топологии сети;
- 4) Экспериментальный подбор характеристик сети;
- 5) Экспериментальный подбор параметров обучения;
- 6) Собственно обучение;
- 7) Проверка адекватности обучения;
- 8) Корректировка параметров, окончательное обучение;
- 9) Вербализация сети с целью дальнейшего использования.

Короче, никаким ИИ здесь не пахнет. Сара Коннор может спать спокойно и придется ей придумать другую легенду появления внебрачного ребенка. Кстати, в каком-то смысле Терминатор вполне себе библейский сюжет. Мальчик, рожденный не в браке. Он будущий король мира. Появился от Архангела и Святого Духа, возникших из ниоткуда. Он вполне бы умер за наши грехи, но на эту роль подвернулся старина Арни во 2ой и последующих частях.

А путешествий во времени, к сожалению, нет, т.к. это тупо нарушает закон Ломоносова-Лавуазье. Из ничего не может появиться что-то. А жаль, я так хотел прокатиться верхом на трицератопсе! Это кроме того факта, что невозможно точно выяснить местоположение планеты в прошлом, т.е. при путешествии даже в минус неделю мы окажемся в космосе.

Кроме того НС не равны ИИ еще и потому, что НС могут различить объекты, но не могут понять их суть, т.е. процесса мышления не происходит. Т.е. НС по интеллекту на уровне тостера. Как следствие, машина не способна осознать себя отдельной личностью и начать выпиливать конкурирующие виды. Но это уже плюс для нас. Хотя эволюции в общем-то, плевать, кто будет на вершине пищевой цепочки.

По сути НС это жестко заданный алгоритм. Если прогнать через нее один и тот же набор данных, результат будет на 100% известен. А там где есть алгоритм нет разума, т.к. разум гибкий и иногда противоречащий правилам и ожидаемой выгоде. Кто из нас не сядил играть в Героев накануне экзамена в универе или не съедал торт забив на диету?

Как выяснилось, разочарование в ИИ стандартная штука. Называется "зима ИИ". Это периоды, когда человечество разочаровывалось в ИИ (из-за завышенных ожиданий), сокращались бюджеты, рушились карьеры и снижался на годы интерес к ИИ. Самая известная зима ИИ – в 1969 г началась после того, как одnogруппник Фрэнка Розенблатта (изобретателя перцептрона) Марвин Минский доказал принципиальную невозможность решения многих классов задач перцептронами. А также после многочисленных провалов проектов для военных по машинному переводу (мгновенному переводу секретных переговоров русских), автономному танку, распознаванию на фото бронетехники, краху Lisp-машин (еще одна вундервафля не оправдавшая ожиданий) и др.

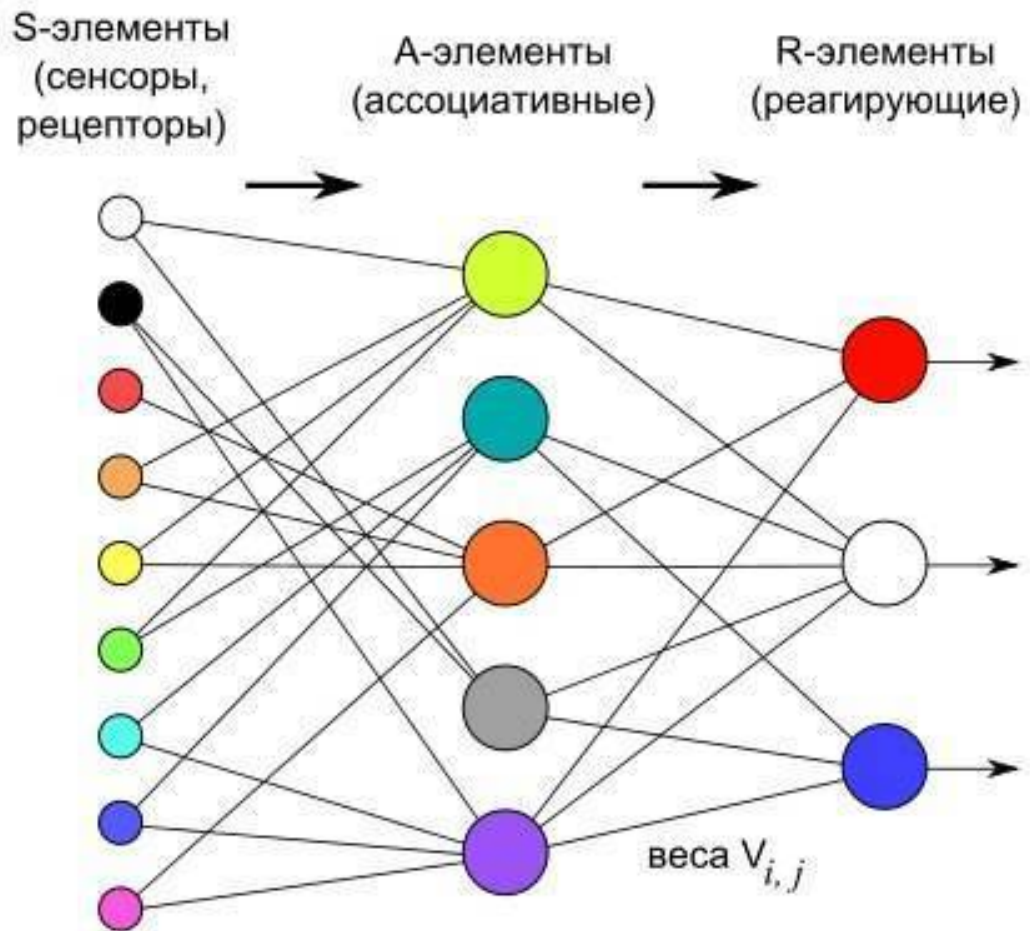
Да, а придумал перцептрон Фрэнк Розенблат. Наш человек. Его еврейская семья очень своевременно свалила в США из СССР/Украины как раз в годы между 1 и 2-ой мировыми войнами. Хорошая интуиция. Стал известным ученым в области ИИ, профессором и звездой Корнеллского университета. Исследовал мозг, включая такие спорные опыты на мышах, как обучение мышей навыкам, затем переработку их мозгов в миксере и ввод этой жидкости в мозг другим мышам, а у тех внезапно появлялись навыки прошлых мышей. Мозг странная штука.

Умер в 43 года очень странной смертью после кораблекрушения своей яхты в Чесапикском заливе в 1971 г. Надеюсь, это все-таки была операция по смене личности от ЦРУ и он жил долго и счастливо работая в секретной лаборатории и придумывая всякие интересные штуки. Хоть я и не верю в мировые заговоры. Миром правит явная лажа, а не тайная ложа.

Крупный недостаток НС – они не могут предсказать будущее на основе прошлого если данные из прошлого не коррелируют с будущим. Например они не способны угадать перелом тренда и предсказать курс доллара на завтра. Также, можно скормить НС хоть миллиард знаков числа  $\pi$ , но на их основе НС не сможет предсказать следующую цифру. Впрочем, никто из хомо сапиенс это тоже сделать не в состоянии.

Ссылки:

- 1) Машинное обучение для людей: [https://vas3k.ru/blog/machine\\_learning/](https://vas3k.ru/blog/machine_learning/)
- 2) Русское датасаенс комьюнити: <https://ods.ai>



[Источник фото](#)

## 4 Распознавание цифр без нейросетей

Продолжаю изучать нейронные сети (НС). Если вам неинтересны технические детали НС (ну, вдруг), а в посте ищите только смехуечки, промотайте текст до середины (фразы "тупой комп"), там пара абзацев для гуманитариев.

Итак, прочитал еще пару статей. Много прояснилось, но вопросы остались. Сигмоидная функция  $f(x) = 1/(1+e^{-x})$ . По описанию страшная вещь! А по факту – просто преобразователь данных. Чтобы значения от  $[-\#..+\#]$  преобразовать в  $[0..1]$ .

Нейрон смещения – тот же преобразователь, чтобы сдвинуть функцию (полученные значения) влево или вправо по оси  $x$ . Т.е. из диапазона  $[0..1]$  перейти, например, в диапазон  $[3..4]$ .

Но главное, я по-прежнему не понимаю как на физическом уровне устроен процесс обучения и распознавания НС.

Работу НС обычно описывают так. Есть НС с 3мя слоями: входным, средним и выходным. Присваиваем нейронам среднего слоя случайные веса. Подаем на входной слой образец для обучения. Накладываем каждую точку входного изображения на входной слой. Если на нейроне входного слоя есть сигнал, умножаем его на вес связанного нейрона из 2-го слоя и передаем на 3-ий выходной слой. Выходной слой суммирует пришедшие сигналы со 2-го слоя и пропускает его через функцию активации (ту самую сигмоиду) чтобы преобразовать сигнал в  $[0..1]$ .

По сигналу 0 или 1 НС говорит на фото кошка или таки собака. Если НС ошиблась, вычисляем "методом градиентного спуска" какие веса должны быть у нейронов 2-го слоя, чтобы минимизировать ошибку. Меняем веса нейронов через "метод обратного распространения ошибки". Подаем на вход НС все больше данных, НС учится, мутки мутятся, ошибка уменьшается и НС всё лучше распознает данные.

Вроде понятно, но что конкретно? Как сделать НС с нуля? Как она научится распознавать изображение? Зачем расставлять случайные веса нейронов (приносить шум в систему) в начале обучения? Зачем менять веса нейронов на каждой итерации через градиентный спуск? Разве НС не будет перенастраиваться каждый раз и запоминать последний образец? Как методом обратного распространения ошибки менять веса всех нейронов так, чтобы общая вероятность распознавания НС увеличивалась после каждой итерации? Разве возможно уменьшать выходную ошибку  $f\_error(x)$  если она зависит от кучи параметров  $x_1...x_n$ , а сами параметры  $x$  не должны влиять друг на друга?

В общем, я не понял как создать НС с нуля. Поэтому решил сделать промежуточное грубое решение исходя из того, что ясно на данный момент.

В чудесном новом мире датасаенс и нейронных сетей есть свой аналог простейшего приложения HelloWorld, как это принято в остальном ИТ. Задача состоит в том, чтобы написать программу, которая распознает рукописные цифры 0..9. Всего-то.

Я уж было решил взять тетрадь в клеточку, написать по страничке каждую цифру, отсканировать и создать таким образом набор данных для распознавания. Но тут открыл для себя

прекрасный сайт [kaggle.com](https://www.kaggle.com) с кучей бесплатных датасетов, конкурсами и datascience-кудесницами. Оттуда скачал \*.csv датасет MNIST с 60+10 тыс рукописных цифр в размере 28x28 точек собранных из сканов контрольных работ американских школьников.

Реализовал простой алгоритм. Назвал его "метод вероятностного накопления". La méthode de l'accumulation de probabilité (fr). На всякий случай забуду название, вдруг докторскую еще по ней защищать.

Суть в том, чтобы в режиме обучения для каждой цифры просуммировать веса каждой точки ее изображения для всех тренировочных образов. После тренировки НС у нас будут 10 матриц 28x28 с весами каждой точки для цифр 0..9.

В режиме распознавания подаем на вход скан цифры и определяем пересечения с матрицами цифр 0..9. Если точка закрашена в образце и матрице, суммируем вес из матрицы. Если в образце точка закрашена, а в матрице нет, вычитаем штрафные очки. Это защита от кейса "закрасим всю область черным и получим цифру 8". Матрица с максимальной суммой считается распознанной цифрой 0..9.

В общем долго объяснять, а кода получилось всего ничего ~100 строк. См. код в скринах и ссылку в конце.

Можно улучшить алгоритм. Если точка не попала в пересечение, находить ближайшую и добавлять ее вес с неким коэффициентом. Можно делать изображения черно-белыми (сейчас серое). При совпадении точек умножать ее вес на вес из матрицы и др.

Но даже с исходными условиями получилось довольно неплохо. НС корректно определило цифру в 57% случаев при тренировке 5 тыс изображений и 5тыс для теста.

Правда на вики пишут, если правильно построить НС, можно добиться 99.8% корректно распознанных цифр. На кривой козе к Тьюрингу не подъедешь.

В общем, надо копать тему дальше. Есть инструменты обучения НС более высокого порядка – Google/TensorFlow, Microsoft/Azure, Amazon/AWS, Яндекс/DSVM. Где нужно задать только данные и параметры обучения и можно использовать НС не зная какие алгоритмы в ней работают. Но все-таки хочу знать что у этой штуки под капотом. Без этого пегого дудочника не напишешь и в техкранч не попадешь.

Как я сказал выше, основная проблема в том, что я не понимаю как НС должна работать "по-настоящему". Что бесит – даже создатели НС, той же DeepBlue (побила Каспарова в шахматах) или AlphaGo (разгромила чемпиона мира по ГО) не могут внятно объяснить, как работает их НС и почему она сделала определенный ход.

Еще смешнее с появившимися последнее время интеллектуальными голосовыми/чат помощниками – Сири, Алисой и Олегом. На жалобу клиентки, что ей не удается войти в моб банк по отпечатку пальца, Олег посоветовал отрезать ей пальцы.

Лол, кек. А что в итоге? Авторы Олега не смогли объяснить его кровожадность. Типа, Олег еще учится, потом станет умнее. Имхо, это не ИИ, а просто перебор фраз, части которых использовались в похожем контексте, а Олег тупо их склеил в примерно осмысленную фразу.

Это не интеллектуальный помощник, а буллит генератор. По развитию Олег не ушел от ELIZA, видел такую программу поддерживающую разговор с человеком еще в 1996 г (создана в 1966 г!).

Это же алгоритм и тупой комп! Что ему дали, то он и посчитал. В ИТ нет ответа "я кнопку нажала и все пропало, не знаю что произошло". Всегда есть конкретная причина у всего и всегда есть ответ на поставленный вопрос. Может трудно получить ответ, но он всегда есть.

Математика – точная наука, не то что эти ваши гуманитарные сопли:

– География – бессмысленное заучивание списка стран и столиц.

– История – зазубривание списка фактов и сознательное искажение этих фактов в угоду текущей повестке.

– Литература – изучение контента, написанного алкоголиками, самоубийцами и депрессивными интеллигентами, не нашедших своего места в жизни и рефлексирующих над проблемами давно утративших актуальность.

– и прочие недонауки (медицина/биология, философия, психология и, мое любимое – естествознание! никогда не понимал, что это такое и зачем это в школе).

Не боюсь оскорбить гуманитариев этим постом, они его не прочтут, т.к. стоят на свободной кассе.

Со своей стороны признаюсь, что несмотря на техническое образование, не понимаю как работают многие вроде бы элементарные вещи.

Особенно плохо в школе нам давали физику. Препод был 2х-метровый боксер-еврей. Похоже он просто замещал свою жену, тоже учителя физики, но сам совершенно не любил, не понимал и не умел интересно объяснять свой предмет. Но сообщить ему сей прискорбный факт желающих не нашлось.

А еще говорят евреи круты в физике. В универе, правда, я сдал ее на высший бал и только за счет этого поступил на бюджет. Тупо зазубрил формулы и моментально все забыл на следующий день. А вот математику и особенно химию нам давали неожиданно хорошо для почти сельской школы.

Итак, ТОП-12 непонятных для меня вещей. Для того, кто плохо знает физику, мир полон чудес.

1) Электричество. Дырки и электроны, которые ползут в разные стороны в зависимости от разности потенциалов. Ну фигня же полная! Как это посмотреть в микроскоп?

2) Конденсаторы. Заряженный аккумулятор Теслы весит как пустой, но двигает авто на сотни километров. Чудо же!

3) Гравитация и закон всемирного тяготения. Кидаешь яблоко вверх и оно приземляется тебе на голову. А если его кинуть с 1-ой космической скоростью 8 км/сек, оно уйдет на стационарную орбиту вокруг планеты и будет крутиться там очень долго. А с 11.2км/сек яблоко покинет планету (16.7 – Солнечную систему). Но блин, почему вещество притягивает другое вещество? Почему людей тянет друг к другу?

4) Гравитация на микроуровне. Почему электрон крутится вокруг ядра и не улетает к чертям?

5) Критическая масса и ядерная реакция. Почему если быстро и сильно сжать всего 50 кг урана-235 будет бабах и выделяется энергия, которая сожжет целый город? А если сжать всего на килограмм меньше или не так быстро, то ничего не будет?

6) Преобразование химических элементов. Раз элементы настолько сильно отличаются всего от одного электрона на орбите ядра, почему нельзя сдуть пару электронов и свинец превратить в золото? Почему вообще идут химические реакции?  $H_2+O_2=H_2O$  и энергия. Как вещество превращается в энергию и наоборот? Концепция смешивания веществ ясна, а вот преобразование весьма туманно.

7) Как фотоны могут передавать энергию и свет, не имея массы? Как что-то может не иметь массы? Если масса = 0, то этого чего-то нет. Так вижу!

8) Ну, с невозможностью построения летательных аппаратов тяжелее воздуха вроде разобрался.

9) ОТО (общая теория относительности). Почему скорость частиц или космического корабля не может быть быстрее 300 тыс км/сек? Почему размеры предметов уменьшаются от увеличения скорости? Почему близнецы (космонавт и неудачник на Земле) стареют по-разному? Почему для элементарной частицы можно определить положение или импульс, но не оба параметра сразу?

10) Корпускулярно-волновой дуализм. Вот это совершенно наркоманская вещь! Свет одновременно набор частиц и волна. При передаче энергии это частицы, а при прохождении через дифракционную решетку (расческу) уже волна. Даже один неделимый (!) фотон, прошедший через решетку с 2мя отверстиями типа расщепляется (!!!) и оставляет след за каждой из щелей, т.е. становится волной.

11) Большой взрыв. Ага, прям из точки с нулевым радиусом и бесконечной массой все бабахнуло и разлетается в бесконечной вселенной.

12) Происхождение жизни. Из неживого создалось живое?! Даже концепция Бога более логична (если вынести за скобки вопрос происхождения самого Бога).

И возвращаясь к науке DataScience. Как я понял, главное в ней – данные. Нет data – нет science.

Ссылки:

1) Мой код нейросети распознающей рукописные цифры (49 КБ): [https://drive.google.com/file/d/1g1Owp6PLOPE6\\_ChbJoe8paLDviRczHPR](https://drive.google.com/file/d/1g1Owp6PLOPE6_ChbJoe8paLDviRczHPR)

<https://github.com/berlicon/SimpleNeuralNetworkMNIST>

2) MNIST database в удобном формате \*.csv:

<https://www.kaggle.com/oddrational/mnist-in-csv>

3) MNIST database исходные данные:

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

[https://en.wikipedia.org/wiki/MNIST\\_database](https://en.wikipedia.org/wiki/MNIST_database)

4) [https://en.wikipedia.org/wiki/List\\_of\\_datasets\\_for\\_machine-learning\\_research](https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research)

5) 52 датасета для тренировочных проектов:

<https://habr.com/ru/company/edison/blog/480408/>

6) Нейронные сети для начинающих. Часть 1:

<https://habr.com/ru/post/312450/>

7) Нейронные сети для начинающих. Часть 2:

<https://habr.com/ru/post/313216/>

8) Машинное обучение для людей:

[https://vas3k.ru/blog/machine\\_learning/](https://vas3k.ru/blog/machine_learning/)

9) [https://ru.wikipedia.org/wiki/Метод\\_обратного\\_распространения\\_ошибки](https://ru.wikipedia.org/wiki/Метод_обратного_распространения_ошибки)

10) [https://ru.wikipedia.org/wiki/Градиентный\\_спуск](https://ru.wikipedia.org/wiki/Градиентный_спуск)

11) <https://en.wikipedia.org/wiki/ELIZA>

```
//Program.cs
```

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.IO;
```

```
using System.Text;
```

```
namespace SimpleNeuralNetworkMNIST
```

```
{
```

```
class Program
```

```
{
```

```
const int IMAGE_SIZE = 28; //each image 28*28 pixels
```

```
const int SAMPLE_COUNT = 10; //analyse 10 images – numbers 0..9
```

```
const int TRAIN_ROWS_COUNT = 5000; //first rows to train;
```

```
const int TEST_ROWS_COUNT = 5000; //other rows to test
```

```
const int INCORRECT_PENALTY = byte.MaxValue * TRAIN_ROWS_COUNT; //penalty
```

```
for incorrect overlap
```

```
    //const string FILE_PATH = @"C:\Users\3208080\Downloads\mnist-in-csv\nnist_test_200_rows.csv"; //43% 100+100
```

```
    //const string FILE_PATH = @"C:\Users\3208080\Downloads\mnist-in-csv\nnist_test_2000_rows.csv"; //53% 1000+1000
```

```
    //const string FILE_PATH = @"C:\Users\3208080\Downloads\mnist-in-csv\nnist_test_2000_rows.csv"; //56% 1900+100
```

```
    //const string FILE_PATH = @"C:\Users\3208080\Downloads\mnist-in-csv\nnist_test.csv"; //50% 9900+100
```

```
    //const string FILE_PATH = @"C:\Users\3208080\Downloads\mnist-in-csv\nnist_test.csv"; //56% 9000+1000
```

```
    const string FILE_PATH = @"C:\Users\3208080\Downloads\mnist-in-csv\nnist_test.csv"; //57% 5000+5000
```

```
    //const string FILE_PATH = @"C:\Users\3208080\Downloads\mnist-in-csv\nnist_test.csv"; //49% 1000+9000
```

```
    //const string FILE_PATH = @"C:\Users\3208080\Downloads\mnist-in-csv\nnist_test.csv"; //41% 100+9900
```

```

    //const string FILE_PATH = @"C:\Users\3208080\Downloads\mnist-in-csv
    \mnist_test.csv";//55% 5000+5000 black/white

    private static long[, ] layerAssotiations = new long[SAMPLE_COUNT, IMAGE_SIZE,
    IMAGE_SIZE];
    private static Dictionary<long, long> layerResult = new Dictionary<long, long>();
    private static long correctResults = 0;

    static void Main(string[] args)
    {
        train();
        test();

        Console.WriteLine("Правильно распознано {0}% вариантов",
        100 * correctResults / TEST_ROWS_COUNT);
    }

    private static void train()
    {
        Console.WriteLine("Начало тренировки нейросети");
        var index = 1;
        var rows =
        File.ReadAllLines(FILE_PATH).Skip(1).Take(TRAIN_ROWS_COUNT).ToList();

        foreach (var row in rows)
        {
            Console.WriteLine("Итерация {0} из {1}", index++, TRAIN_ROWS_COUNT);
            var values = row.Split(',');
            for (int i = 1; i < values.Length; i++)
            {
                var value = byte.Parse(values[i]); //var value = (values[i] == "0") ? 0 : 1;
                layerAssotiations[
                byte.Parse(values[0]),
                (i - 1) / IMAGE_SIZE,
                (i - 1) % IMAGE_SIZE]
                += value;
            }
        }

    private static void test()
    {
        Console.WriteLine("Начало тестирования нейросети");
        var index = 1;
        var rows =
        File.ReadAllLines(FILE_PATH).Skip(1
        TRAIN_ROWS_COUNT).Take(TEST_ROWS_COUNT).ToList();

        foreach (var row in rows)
    {

```

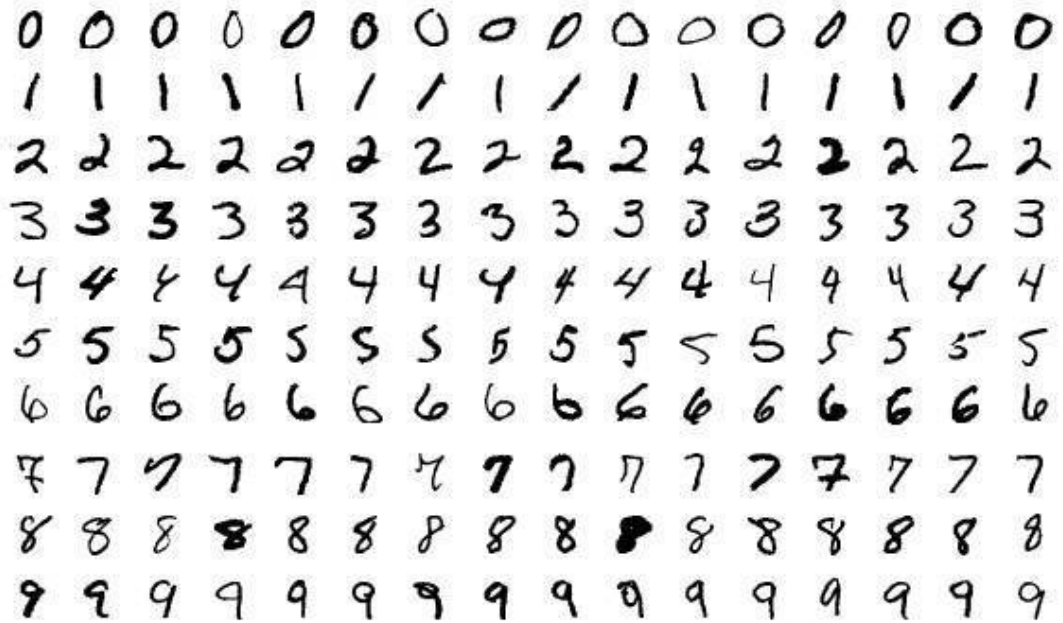
```
Console.WriteLine("Итерация {0} из {1}", index++, TEST_ROWS_COUNT);
clearResultLayer();

var values = row.Split(',');
for (int i = 1; i < values.Length; i++)
{
    var value = byte.Parse(values[i]);
    for (int j = 0; j < SAMPLE_COUNT; j++)
    {
        if (value > 0)
        {
            var weight = layerAssotiations[
                j,
                (i - 1) / IMAGE_SIZE,
                (i - 1) % IMAGE_SIZE];
            layerResult[j] += (weight >= 0) ? weight : -INCORRECT_PENALTY;
        }
    }
}

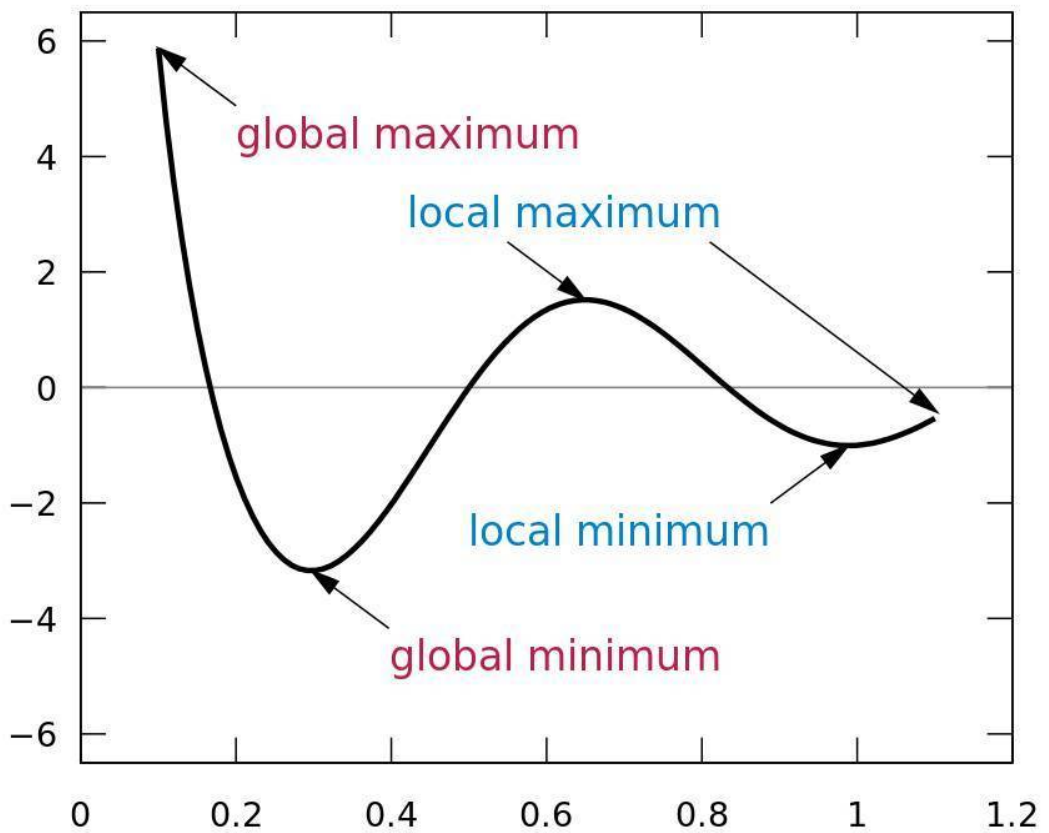
calculateStatistics(byte.Parse(values[0]));
}
}

private static void clearResultLayer()
{
    layerResult = new Dictionary<long, long>();
    for (int i = 0; i < SAMPLE_COUNT; i++) layerResult[i] = 0;
}

private static void calculateStatistics(byte correctNumber)
{
    var proposalNumber = layerResult.OrderByDescending(p => p.Value).First().Key;
    Console.WriteLine("Число {0} определено как {1} {2}", correctNumber, proposalNumber,
        proposalNumber == correctNumber ? "УСПЕХ" : "НЕУДАЧА");
    if (proposalNumber == correctNumber) correctResults++;
}
}
}
```



[Источник фото](#)



[Источник фото](#)

## 5 Нейросеть Matt Mazur

Сделал, наконец, рабочий пример нейросети (НС). Сначала прочитал теорию – объяснение механизма back propagation из русской вики. Вроде все понятно. За вечер написал программу для анализа цифр 0..9 из базы MNIST. Программа запускается, веса обновляются, работа кипит. У любого программиста есть вера в чудо. Что программа, которая компилируется при первом запуске сразу сработает корректно. Да, но нет, девочки. На выходе – 10% распознанных цифр. Ерунда, короче. Как если бы цифра определялась случайным образом.

Стал разбираться. Выяснилось, что в русской вики алгоритм описан сложно, неточно, замылено и недостаточно понятно для реализации без PhD в математике:

1) При изменении весов ребер НС их нужно умножать на коэффициент обучения. Иначе, метод градиентного спуска может застрять в локальном минимуме, так и не попав в глобальный минимум. Но не сказано, как подбирать этот коэффициент.

2) Не сказано, что после подсчета дельт для изменения весов на каждом слое, обновлять веса ребер нужно один раз для всех слоев сразу, а не сразу после подсчета каждой дельты.

3) Для скрытых слоев в правиле подсчета дельт сказано, что нужно умножить на веса исходящих ребер на их дельты. Это не так. Не дельты, а 3 из 4-х множителей в дельте. Т.е. из дельт нужно исключить сигнал на предыдущем узле. Т.е. умножить на:  $(T_j - O_j) * O_j * (1 - O_j)$ , а не на  $(T_j - O_j) * O_j * (1 - O_j) * O_n$ , где  $n = j - 1$

4) Ничего не сказано про архитектуру слоев и ребер. Должны ли они быть Full-connected [FC] (каждый узел соединен с каждым следующим) и как изменить алгоритм если у нас не FC-слои.

5) Пишут про функцию ошибки НС, но никак в финальных расчетах алгоритма ее не используют (хотя из нее через производные выводят формулы подсчета).

6) Полно туманных разъяснений вида "ну, тут очевидно – надо взять частную производную  $dE/dw$ ". Ага, очевидно. Целыми днями только этим и занимаюсь.

7) Нет нормального примера простой НС с проставленными числами для каждого сигнала, веса, узла и расчета как это считается хотя бы на одной итерации.

8) Неясно, нужно ли нормализовать входные сигналы (из 0..255, например, делать числа в интервале 0..1).

9) Говорят, в начале нужно выставить маленькие значения весов ребер. Неясно, насколько малые. 0..1 или 0..0,001?

Правильно говорят, нужна своя, православная вики за 2 млрд руб!

Надо бы залить на вики свой алгоритм работы НС с примером подсчета. Не знаю, почему так ужасно объясняют на русской вики. Возможно это диверсия. Т.е. админы все написали

понятно, но каждую ночь на страницу заходят враги отечества из-за границы (и не только) и запутывают описание алгоритма.

Есть такая теория заговора, что в школах и университетах СССР специально так ужасно преподают английский, чтобы народ не разбежался по заграницам сразу после выпуска. Федеральная программа по противодействию утечке мозгов. Или как-то так. Товарищ майор, если я случайно раскрыл гос. тайну, подайте знак.

В итоге, я сделал все ошибки из возможных, поэтому считался мусор. После гугления и изучения многих примеров НС, нашел, наконец, отличный пример простой реализации НС. С формулами и числовым расчетом. Все понятно, хоть и на английском.

Ссылка: A Step by Step Backpropagation Example by Matt Mazur  
<https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/>

Сделал реализацию по статье и вуаля! Все работает! См. код в конце статьи и на скринах.

Заметил, что когда нужно разобраться в какой-либо айтишной теме, то на английском описание лучше и на треть короче, чем русский текст. На-русском, обычно, объясняют от общего к частному, т.е. методом дедукции, как Шерлок Холмс завещал. Сначала дают кучу формул, а потом цифры. Это очень неудобно, особенно для тех, у кого слабо развито абстрактное мышление. На английском наоборот, дают от частного к общему, т.е. индукцию. Сначала на пальцах считают цифры, а затем говорят какие формулы за этим стоят и почему формулы именно такие.

Наверно есть плюсы в обоих подходах. В дедуктивном подходе (условно, советском), если школьник не врубается в тему, он отваливается на уроке и на него не тратят время. Те же, кто прошел дальше, идут во всякие физматы, становятся перельманами, уезжают преподавать матан в американские университеты, получают нобелевку и отказываются от нее, сходят с ума и уезжают жить назад в Купчино и т.п. В индуктивном же подходе (американском) кривая обучения более пологая, больше людей можно научить. Опять же, это подходит для эмигрантов со слабым знанием языка. Сильно прокачаться в теории сложно, но быстро овладеть практическими знаниями вполне.

Есть такая байка про русского программиста в США. Он заметил, что данные записываются в одну таблицу, а потом прибавляются в нее же и нужно выдать результат. Он переписал функцию и просто выдал наружу `return x*x`; И все такие – вау! Гениальный русский придумал ФОРМУЛУ! Хотя смех смехом, но знакомого программиста с руками оторвали в какую-то контору в США, когда ему дали простую задачу найти элемент массива по условию, а он такой "ба, так это же задача на сортировку графа по теореме такой то!", применил алгоритм и вуаля. А не стал писать свой велосипед, что ОЧЕНЬ ценится в ИТ.

Ура! Заново понял, что такое производная и обратный процесс – интегрирование (восстановление функции по её производной). Геометрический смысл производной – тангенс угла наклона касательной к графику функции в заданной точке. Т.е. мера "крутизны" графика функции, насколько быстро она ускоряется. Так все просто! Все эти:

Для производной:  $(x^2)' = 2x$ , т.е. степень "x" становится множителем и степень уменьшается на 1.

Для интеграла:  $d(x^2) = (x^3)/3$ , т.е. интеграл  $f(x)$  – это найти такое значение функции, производная которой равна нашей (нашему родимому интегралу)

... а для сложных функций вообще чума: [https://ru.wikipedia.org/wiki/Производная\\_функции#Таблица\\_производных](https://ru.wikipedia.org/wiki/Производная_функции#Таблица_производных)

Эта как по телику когда говорят "В последний квартал уменьшились темпы снижения роста вывода капитала за границу". Чтобы понять, что сказал диктор, надо всего-то, взять 5ую производную от денежного потока.

Надо копать дальше. Делать анализ 0..9 для MNIST. Для простого примера текущий код ОК, но вообще, выбрал плохую архитектуру структуры данных. Неудобно обновлять кучу массивов.

## **Конец ознакомительного фрагмента.**

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.