*Проект BG05M2OP001-2.016-0003 „Модернизация на Национален военен университет "В. Левски" - гр.*

*Велико Търново и Софийски университет "Св. Климент Охридски" - гр. София, в професионално направление 5.3 Компютърна и комуникационна техника“*

Тема 1/ Занятие 1/Лекция

# Съвременни схващания за същността и перспективите за изкуствения интелект.

1. **Въведение в изкуствения интелект**

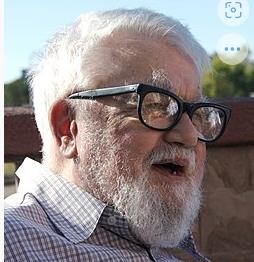
***В***исокопроизводителните изчисления и технологиите за събиране, представяне, съхранение и защита на голям обем от данни позволяват създаването на „интелигентни“ приложения, доближаващи се до възможностите на човешкото мислене при вземане на решение в условия на неопределеност и риск при решаването на задачи, изискващи множество познания в широки области.

Няма общоприето определение на термина „изкуствен интелект“, в литературата се срещат много такива, които се различават по някои детайли, но по своя смисъл всички препращат към идеята за изграждане на машини/компютърни системи, които са способни да изпълняват задачи, които обикновено изискват човешка интелигентност, като визуално възприятие, разпознаване на реч и вземане на решения [3], т.е. да „мислят като хора“.

1

*Дисциплина „Невронни мрежи“*

Терминът „изкуствен интелект е предложен от американския учен Джон Маккарти през 1956г по време на научна конференция в Колежа „Дартмут“ като наименование на „наука за концепциите, методите и средствата за създаване на интелигентни компютърни програми, които могат да извършват действия, които за хората изглеждат разумни, и за изследване на естествения (човешкия) интелект чрез компютърни системи“.

Джон Маккарти (на английски: John McCarthy) е американски информатик и

когнитивен изследовател. Създател е на програмния език Лисп, който след публикуването му през 1960 г. е широко използван за програмиране на приложения прилагащи изкуствен интелект.

Според дефиницията, използвана в съобщението на Европейската комисия

„Изкуствен интелект за Европа“[**Error! Reference source not found.**], понятието „включва системи, които показват интелигентно поведение чрез анализ на средата и предприемане на действия – с известна степен на самостоятелност – за постигане на конкретни цели. Системите, базирани на изкуствен интелект, могат да бъдат изцяло основани на софтуер, действащи във виртуалния свят (напр. гласови асистенти, анализ на изображения софтуер, системи за разпознаване на реч и лице и др.), или да бъдат вградени в хардуерни устройства (напр. модерни роботи, дронове и др.).

Дефиницията за изкуствен интелект е доразвита и прецизирана от независимата експертна група на високо равнище по въпросите на изкуствения интелект, създадена от Европейската комисия през 2018г.[2], както следва:

„Системите с изкуствен интелект (СИИ) са софтуерни (а вероятно и хардуерни) системи, създадени от хора, които с оглед на дадена цел действат в рамките на физическото или цифровото измерение, като възприемат заобикалящата ги среда чрез събиране на данни, тълкуват събраните структурирани или неструктурирани данни, разсъждават въз основа на познанието или обработват информацията, получена от тези данни, и вземат решение за предприемане на най-правилните действия за постигане на дадена цел. Системите с ИИ могат или да използват символно представени правила, или да усвояват цифров модел и могат да адаптират поведението си като анализират начина, по който средата е засегната от предишни техни действия.“ За изкуствен интелект може да се говори когато технически системи наблюдават околната си среда, получават данни (които са подготвени от другиго или които набират сами), преработват ги и извършват действия,

свързани с постигането на конкретна цел.

Системите с изкуствен интелект са способни да адаптират поведението си до известна степен, като анализират резултати от предишни действия, и работят автономно.

# Предимства на системите с изкуствения интелект

Изкуствения интелект притежава всички **предимства, които имат машините над хората**.

Изкуствения интелект позволява човекът да бъде изведен от производствения цикъл, за сметка на машините, които функционират автоматизирано, по-бързо и масово.

Машините с изкуствен интелект заместват човека в изпълнението на дейности с висока степен на трудност и риск в множество области. Тяхната способност за адаптация към условията на външната среда намалява

вероятността за получаване на щета. Докато за хората същата дейност може да е причина за травми и да застраши човешкия живот.

Теоретично, ако не е налице неизправност, ИИ има възможност да работи непрекъснато при сравнително прогнозируеми разходи. Например не се налагат допълнителни разходи като транспортни разходи, заплати, бонуси, отпуск по болест или годишен отпуск, не възникват и психологически, личностни и социални проблеми, свързани с трудово правните отношения, характерни за работните колективи – машините нямат оплаквания от претоварване и не се уморяват като хората.

Изкуственият интелект се използва при вземане на решения и може да мисли по-бързо от хората и не е лишен от субективност в преценката си. Изкуствения интелект изпълнява възложените му задачи по начина, по който е програмирана, ако възникне грешка в работата му, то тя се дължи на пропуск в конструирането или програмирането му.

Изкуственият интелект взема решения безпристрастно за разлика от хората, които могат да бъдат повлияни от емоции и субективни съображения.

**Системи с изкуствен интелект имат и редица недостатъци.** Ето защо не може да се говори за пълна замяна за хората в повечето сфери.

За разлика от човешкия мозък и въпреки големия технологичен напредък, системите с изкуствен интелект все още срещат ограничение за тяхната креативност, разбиране, мислене и т.н. Т.е. те изпълняват предназначението си според начина по който са програмирани, но нищо повече от това.

Нарастването на използването на системи с изкуствен интелект е възможно да създаде безработица в определени сектори.

Разходите за използване на изкуствен интелект големи. Свързани са с покупка, поддръжка и ремонт, това са разходи и се изисква висока инвестиция. Също така трябва да се включат и таксите за актуализиране на софтуера според

променящите се изисквания. Така че само някои организации могат да си позволят разходите за изкуствен интелект.

Изкуственият интелект би могъл да бъде използван и за постигане на злонамерени и престъпни цели. Няма технологични предпоставки, които да възпрепятстват използването на изкуствения интелект за военни цели, в това число и разработването на т.нар. „роботи убийци“, това кара експертите да не изключват възможността технологията да може да допринесе за разрастването на глобален военен конфликт. Според някои разработването на смъртоносни автономни оръжия и използването на изкуствен интелект във вземането на военни решения може да предизвика войни, водени от изкуствения интелект.

1. **Развитие на теорията на изкуствения интелект.**

Идеята за изкуствения интелект се заражда преди около век и благодарение на научно-техническия прогрес се развива с изумително бързи темпове и еволюира.

Първите модерни компютри са военните машини през и веднага след Втората световна война, базирани на теория, създадена от Алън Тюринг, смятан за баща на теоретичната информатика и теорията на изкуствения интелект, и разработени от Джон фон Нойман.

Британецът Алън Матисън Тюринг (на английски: Alan Mathison Turing) е британски математик, логик, криптоаналитик, информатик и философ. По време на Втората световна война той създава няколко метода за криптоанализ и прави подобрения на използваната

изчислителна електромеханична машина „Бомба“, което прави възможно дешифрирането на кодовете на Енигма.

След Втората световна война Тюринг се занимава с начините за обработване на информация, както от човека, така и от машината. Създава т.нар. Тест на Тюринг, според който интелигентна е тази машина, която в общуването си чрез естествен език с човек (събеседник), е неразличима от него за втори човек (наблюдател). **Тестът на Тюринг** се превръща в еталон за идентифициране на интелигентността на машината, който помага за оформянето на философията около изкуствения интелект, и е все още популярен.

До 1955 г. макар в научните среди да се мисли на концептуално ниво за **невронни мрежи и естествен** език, обединяваща концепция, която да обхване различните видове машинен интелект, все още не съществува. Такава се появява на първата научна конференция по изкуствен интелект. Смята се, че тогава се заражда изкуственият интелект като научноизследователска област. На тази конференция се създава обща рамка на изследователските области, в които машинната интелигентност може да окаже влияние.

Американският учен Франк Розенблат разработва **изкуствена невронна мрежа** наречена **Perceptron**, която притежава възможност „да запаметява“ светлинни поредици. Предназначена е да моделира начина по който човешкият мозък обработва визуална информация. Perceptron може да бъде

„обучен“ да разпознава модели от светлини, също както човек прави връзка между дадено име и лице. Въпреки многото си недостатъци програмата е прототип на мислеща машина. Първото устройство, основано на принципа на

невронната мрежа – Mark I Perceptron – се появява през 1958 г. То бързо се научава да разпознава прости геометрични форми.

През 60-те години прогресът започва да се забавя и тази научна област започва да изглежда все по-малко обещаваща поради подценяване на сложността на предизвикателствата.

Към средата на 70-те години, поради прекалено оптимистичните очаквания и липса на важни научни открития, правителствата на САЩ и Обединеното кралство за известно време намаляват финансирането на изследванията в областта на изкуствения интелект. През следващите години настъпва т.нар. „зима“ за изкуствения интелект – период, в който такива изследвания не са финансирани.

От 80-те години на XX век започва нов етап в развитието на изкуствения интелект - „**машинно обучение**“, при който системата имитира човешкото поведение в определени граници и има способността и да се учи.

По нататък усилията се концентрират върху използването и изработването на т.нар. „**експертни системи**“. Те са програми, които отговарят на въпроси и решават проблеми в конкретна област, използвайки зададени правила. За много задачи тези системи се справят значително по- качествено и по-бързо от хората.

Втората „зима“ за изкуствения интелект идва в края на 80-те и началото на 90-те години след поредица от финансови загуби. Липсата на алтернативни открития и загубата на популярност на експертните системи, отново довеждат до спад в интереса към изкуствения интелект от страна на обществото и инвеститорите.

В навечерието на новото хилядолетие ситуацията се променя и въпросите за изкуствения интелект отново стават актуални и перспективни. Постига се значителен напредък след период на неуспехи. Това се случва през 1997г., когато **Deep Blue на IBM** става първата машина за шах, която успява да победи световния шампион по шах – Гари Каспаров на 11 май 1997г. Проектираният компютър Deep Blue е бил способен да изчислява по 100 милиона позиции в секунда, което е уникално постижение за тези години. Играта, в която Deep Blue прави първата победа на машина срещу човек в шах, е първият от общо шест мача. Каспаров успява да постигне две равенства и три победи, което го прави победител в общата игра, но по-късно през 1997г., на реванш машината разгромява гросмайстора. Победата връща интереса към изкуственият интелект по впечатляващ начин.

От 10-те години на XXI век се заражда така нареченото „**дълбоко обучение**“, което представлява следващия етап в развитието на изкуствения интелект и машинното обучение, при който системата има способността да осигурява висока точност при задачи като идентифициране на обекти, разпознаване на реч, превеждане от един език на друг, автоматично извличане на информация от различни видове източници – текст, изображения, аудио, видео и др.

През 2011г. година инженерът Джеф Дийн и Андрю Нг - професор по компютърни науки в Станфордския университет представят идеята за изграждане на **голяма невронна мрежа**, захранвайки я с огромна изчислителна мощ, използвайки сървърните ресурси на Google и запълвайки я с масивен набор от изображения.

Невронната мрежа обработва данни от изображенията в продължение на три дни. След това връща резултат, съдържащ три изображения – човешко лице, човешко тяло и котка. Това изследване е голям пробив в изследването на невронни мрежи и автономното обучение без човешка намеса в задачите за т.нар. „компютърно зрение“, насочено към автоматична обработка на изображения от реалния свят с цел извличане и интерпретиране на визуалната информация в тях. Така могат да се откриват и разпознават обекти и визуални измерения.

Професор Джефри Хинтън от Университета на Торонто и двама негови студенти изграждат система за **компютърно зрение**, наречена AlexNet, за да се състезават в конкурс за разпознаване на изображения, наречен ImageNet. И успяват да го спечелят. Системата анализира хиляди снимки и се самообучава да разпознава общите предмети като цветя и автомобили. Това може би повече от всяко друго събитие дава силния старт в изследванията на дълбоките невронни мрежи и съответно машинното обучение.

Джефри Хинтън, Иошу Бенджио и Ян Лекун са наречени „кръстниците“ на изкуствения интелект и през 2018г. печелят приза „Тюринг“, известен като технологичната версия на Нобелова награда.

През 2016 г. компютърната програма AlphaGo реализира първата победа срещу може би най-добрия играч в света на древната японска настолна игра Go. Така 18-кратният световен шампион Седол Ли губи за по-малко от два часа. Играта е една от най-сложните, измисляни някога, но компютърната програма е разработена така, че не само да подражава на хората, но и да разработва нови стратегии, имитирайки човешката интуиция в справянето със сложни задачи. По време на играта подобрената версия на AlphaGo – AlphaGo

Zero дори разработва свои специфични стратегии, които разиграва и впоследствие развива предпочитание към серия от ходове и варианти, непознати досега. С четирите си процесора, изкуствения интелект тренира системите си само за 40 дни, което е изключително постижение за учените. От подразделението за изкуствен интелект на Google-Deep Mind обявява тогава, че разработената технология ще бъде използвана в областта на роботиката.

Литература:

1. Winston, P., Artificial Intelligence, Third Edition, Addison-Wesley, 1992.
2. Rich E., Knight K., Artificial Intelligence, McGraw-Hill, (2017), ISBN- 10: 0070087709
3. Minsky M. and Papert S., “Perceptrons,” MIT Press, Cambridge, 1969
4. Dimiter Dobrev. Formal Definition of Artificial Intelligence. International Journal "Information Theories & Applications", vol.12, Number 3, 2005, pp.277-285. <http://www.dobrev.com/AI/>
5. [15] Dimiter Dobrev. Parallel between definition of chess playing program and definition of AI. International Journal "Information Technologies & Knowledge ", vol.1, Number 2, 2007, pp.196-199.

Тема 1/Занятие 2/Лекция

# Основни теоретични концепции в областта на изкуствения

**интелект**

Увод: Изкуственият интелект е дял от информатиката, изучаващ възможностите на изчислителни системи да реализират интелектуални процеси, характерни за хората - възприятие, анализ на данни, формулиране на решения въз основа на този анализ, откриване на смисъл, обобщаване и самообучение на базата на предишен опит.

# Основни характеристики на изкуствения интелект

Човешкият интелект е комплекс от разнообразни способности, които най-общо са: разсъждаване, анализ и синтез на информация. Ето защо за да се твърди, че една компютърна система притежава интелект, тя трябва да притежава способности: учене, разсъждаване, решаване на проблеми, възприятие и използване на език [**Error! Reference source not found.**].

# Способност за учене (обучение)

Според Хърбърт Саймън обучението на компютърната система се състои в такива промени в системата, които и позволяват при всеки следващ опит да извършва дадена работа по-ефективно, отколкото при предходните опити. Към изкуствения интелект са приложими различни форми на обучение, но най-удобен за компютърна реализация е методът за учене „проба-грешка“. Пример за използване на такъв подход е компютърна програма за игра на шах, решаваща задача за мат в един ход, която реализира произволни движения на фигурите по дъската, докато не намери решение. Решението с позицията се съхранява, така че следващият път, когато програмата срещне същата позиция, прилага същото решение. Механичното запаметяване на казуси и процедури

за обработката им, познато като „обучение чрез наизустяване“ - rote learning е сравнително лесно за програмна реализация. По-трудно се решава проблемът с прилагането на миналия опит в аналогични нови ситуации.

**Способност за регистриране на данни за околната среда (Сетивност)** Изкуственото възприятие се състои в сканиране на средата с помощта на различни сетивни органи (сензори), а сцената се разлага на отделни обекти в

различни пространствени отношения.

# Използване на език

Езикът е система от знаци с предварително дефинирано значение. Важна характеристика на човешките езици е тяхната производителност, която се изразява във възможността за формулиране на неограничено разнообразие от изречения. Създаването на компютърни програми, които в силно ограничен контекст реагират свободно на въпроси и изявления на човешки език е сравнително лесно. Те биха могли да се усъвършенстват/самоусъвършенстват до степен, че владеенето на език от тяхна страна да не се различава от това на нормалния човек.

# Вземане на решение

Разсъждението се състои във формулирането на изводи от определена ситуация. Вземането на решение може да се разглежда като систематично търсене в пространството на възможните действия, с оглед постигне на предварително определена цел или решение. Постигнат е значителен успех в програмирането на компютри за извличане на дедуктивни заключения.

Съществуват специализирани методи и методи с общо предназначение. Методите със специално предназначение са пригодени за решаването на конкретен проблем и са съобразени с редица специфични характеристики, докато универсалните са приложими за решаване на голямо разнообразие от проблеми.

Процедурите за вземане на самостоятелно решение от системите с изкуствен интелект се осъществяват чрез два основни подхода – **конвенционален**, който използва методи за самообучение на [машината](https://bg.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%B0), основани на специален формализъм и [статистически](https://bg.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B0) анализ и **изчислителен**, който се свързва с използване на интерактивно обучение на системите с изкуствен интелект, основано на емпирични данни и асоциирани с тях

„гъвкави“ изчисления.

# Системи с изкуствен интелект, разработени на основата на конвенционалния поход

На основата на конвенционалния подход са изградени:

# [Експертни системи](https://bg.wikipedia.org/wiki/%D0%95%D0%BA%D1%81%D0%BF%D0%B5%D1%80%D1%82%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0)

Експертните системи програми, които са предназначени да заместват експерт в определена област. Те работят по определени правила, обработват голямо количество информация и в резултат извеждат заключение по нея.

Те решават практически проблеми, които възникват в слабо структурирана и трудна за формализиране предметна област. Те са едни първите системи с изкуствен интелект.

Тук да сложа моя доклад за оценяването с Марго и от сайта на Ганчева

* + Разсъждение на основата на аналогични случаи (англ. Case- based reasoning).
  + Вероятностни ([Бейсови) мреж](https://bg.wikipedia.org/wiki/%D0%91%D0%B5%D0%B9%D1%81%D0%BE%D0%B2%D0%B0_%D0%BC%D1%80%D0%B5%D0%B6%D0%B0)и (Belief Networks).
  + [Дърво на решенията](https://bg.wikipedia.org/wiki/%D0%94%D1%8A%D1%80%D0%B2%D0%BE_%D0%BD%D0%B0_%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F%D1%82%D0%B0)
  + Поведенчески подход: модулен метод за изкуствен интелект, при който системата се разделя на няколко сравнително автономни програмни поведения, които се задействат в зависимост от измененията във външната среда.

# На основата на изчислителния подход са архитектури като:

* + [**Невронни мреж**](https://bg.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B2%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BC%D1%80%D0%B5%D0%B6%D0%B0)**и**: използват математически модел за обработка на данни, сходен с начина, по който функционира човешката нервна система. Намират приложение при разпознаване на изображения.

# Генетични алгоритми

Генетичните Алгоритми (ГА) са част от еволюционното изчисление, което е бързо нарастваща част от изкуствения интелект. Те са универсална вероятностна техника на търсене, която имитира естествената биологична еволюция и използва понятията от естествената генетика. Различават се съществено от традиционните оптимизационни методи. Основните разлики между тях и познатите оптимизационните процедури могат да се синтезират по следния начин:

1. ГА работят в кодирано параметрично пространство, а не със самите параметри;
2. ГА извършват търсенето в множество от точки, а не само в отделна точка.
3. ГА извършва оптимизация на базата на предварително дефинирана потребителска функция, а не на базата на производни или други допълнителни знания.
4. ГА работят на базата на вероятностни преходни правила, а не на базата на детерминирани правила.

Идеята за еволюционното изчисление бе представена през 1960 от **И. Реченбърг** в неговата разработка *“Еволюционни стратегии”*. Неговата идея след това е разработена от други изследователи. Генетичните алгоритми са изобретени от Джон **Холанд** и разработени от него, негови студенти и колеги. Това е довело до книгата на Холанд *“Адаптация в Естествени и Изкуствени Системи”* публикувана през 1975.

През 1992 Джон Козза е ползвал генетични алгоритми да развие програми изпълняващи определени задачи. Той нарекъл своя метод “генетично програмиране” (ГП). Използвани са програми на LISP, понеже програмите на този език могат да се изразяват под формата на “дървета с граматичен разбор”, каквито са обектите върху които работят ГА.

Генетичните алгоритми са вдъхновени от Дарвиновата теория за еволюцията. Решението на задача изчислявана чрез генетичните алгоритми става чрез развитие.

Генетичните алгоритми, основаващи се само на трите основни генетични оператори възпроизвеждане, кросоувър и мутация, се наричат обикновени ГА. Наред с тях редица автори като Bagley(1967), Rosenberg

(1967), Hollstien (1971), Brindle (1981), Smith (1987, 1988), Goldberg & Smith (1987) и други, разглеждат и въвеждат и ред модернистични оператори и съвременни техники в ГА, целящи подобряване на качествата на обикновените ГА.

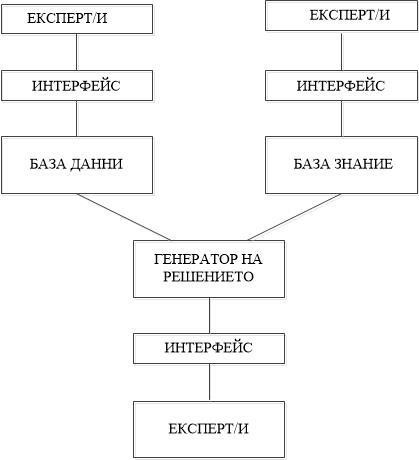
Алгоритъма започва с набор от решения (представени от хромозоми) наречен популация. Решенията от една популация се вземат и се ползват да се формира нова популация. Това се мотивира от надеждата, че новата популация ще е по-добра от старата. Решенията които са избрани да формират новото решение (потомството) се избират според тяхната устойчивост – колкото по подходящи са те, толкова повече са шансовете им да се репродуцират.

Това се повтаря докато някое условие (например брои популации или напредък към най-доброто решение) е удовлетворено.

# Системи с [размита](https://bg.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D0%BC%D0%B8%D1%82%D0%BE_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%BE) логика (от англ. Fuzzy Logic): методи за разсъждение в условия на неопределеност.

С понятието размита система (Fuzzy System) се означават експертни или основани на знания системи, базирани върху размити формализми. Теорията на размитите множества предоставя подходящ инструментариум за моделиране на неопределеностите от различен тип.

Примерна архитектура на размита система за автоматизирано оценяване на съвместимостта на две информационни системи е представена на фигура 1.



фигура 15. Архитектура на размита система за автоматизирано оценяване на съвместимостта

Предназначението на основните структурни елементи е следното:

***Базата данни*** (БД) съдържа данните за оценяваните системи, които могат да се съхраняват аналогично на записите в БД от релационен тип. Всеки запис може да се разглежда като реализация на експертната оценка на степента на съответствие между моделите на съвместимост на две информационни системи. Предполага се, че всички атрибути са прости и съответстват на оценките в скалата M = {<ниска степен на сходство>, <средна степен на сходство>, <висока степен на сходство>}.

БД се попълва от експерт/и посредством специализиран интерфейс, който осигурява връзката с експерта и извършва синтактичен и семантичен контрол на въвежданите данни.

***Базата знания*** (БЗ) съдържа експерно формулирани правила за:

* + конволюция на експертната оценка;
  + вземане на решение относно степента на съвместимост, представени в таблица 7.

***Генераторът на решението*** формира оценката на съвместимостта на системите, като използваданните, съдържащи се в базата данни и на правилата описани в базата знания. Съдържа два модула, които реализират последователно етапите на вземане на решение:

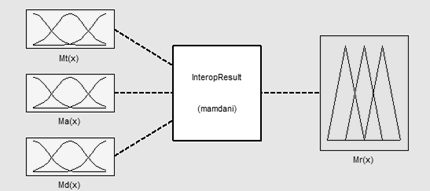
## *Модул за свиване на векторнозначната оценка.*

Реализира композиционното правило на Заде за преобразуване на многоразмерната оценка от вида (7) във вида: ***µR = (µT(х), µA(y), µD(z))***

* + ***Модул за определяне на степента на съвместимост -*** формира оценка на съвместимостта в скалата {<много ниска степен на съвместимост>,

< ниска степен на съвместимост>, < средна степен на съвместимост>, < висока степен на съвместимост>, <много висока степен на съвместимост>}.

На фигура 16 е представена блокова диаграма на модула за определяне на степента на съвместимост, построена в Matlab. Избран е механизъм на Мамдани за извеждане на заключение. Входни данни са оценките ***µT(х), µA(y)*** и ***µD(z).*** Решението се взема по правилата, съдържащи се в БЗ. Екран от формата за въвеждане на правилата в базата знания е представен на фигура 17.



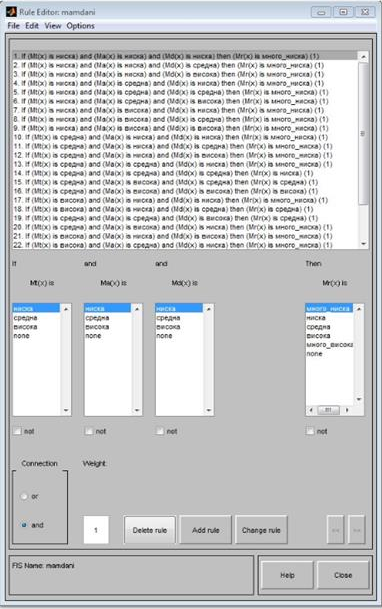
фигура 16. Блокова диаграма на модула за определяне на степента на

съвместимост

## *Функциониране на системата*

1. Попълване на базата знания от експерт или група експерти.
2. Въвеждане на данни за оценяваните системи.
3. Проверка на въвежданите данни за пълнота и непротиворечивост.
4. Генериране на решението.
5. Извеждане на резултатите в удобен за потребителя вид.

Реализирането и използването на размита система за автоматизирано оценяване дава възможност за ускоряване на процеса и обективизиране на резултата посредством намаляване на неточностите, присъщи на експертното мнение.



фигура 17. Екран от редактора за въвеждане на правила в Matlab

## *Възможност за развитие на системата*

Възможно е разширяване на функционалността на експертната система чрез добавяне на функция за определяне на степента на сходство между съответните показатели. С това ще се автоматизира формирането на оценката от вида (7), което ще доведе до минимизиране работата на експерта и свеждане

на задачата му до въвеждане на формализираните описания на системите (5) и

1. в БД.

Реализирането на тази функционалност е възможно чрез добавяне на нов модул в ***Генератора на решението - Модул за определяне на степените на сходство***. Целта на работата на този модул е да се получи многомерен вектор с оценки в скалата {<ниска степен на сходство>, <средна степен на сходство>,

<висока степен на сходство>} от постъпилите на входа на системата експертно изготвени описания.

***Базата знания*** на усъвършенстваната експертна система следва да се допълни с правилата за вземане на решение относно определянето на степента на сходство на показателите. Формулирането им поражда необходимостта да се моделира мисленето на експерта, а това е неформализована задача, която трудно се поддава на алгоритмизация.

Реализацията на нова функционалност е свързана със значително усложняване на архитектурата на системата, увеличаване на времето за разработка, изискваща висококвалифициран труд на експерти.

* + Системи с [еволюционни](https://bg.wikipedia.org/w/index.php?title=%D0%95%D0%B2%D0%BE%D0%BB%D1%8E%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B8_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%B8&action=edit&redlink=1) и [генетични алгоритми](https://bg.wikipedia.org/wiki/%D0%93%D0%B5%D0%BD%D0%B5%D1%82%D0%B8%D1%87%D0%BD%D0%B8_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%B8): в техната теоретична основа се използват понятия от [биологията](https://bg.wikipedia.org/wiki/%D0%91%D0%B8%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%8F), като [популация](https://bg.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BF%D1%83%D0%BB%D0%B0%D1%86%D0%B8%D1%8F), [мутация](https://bg.wikipedia.org/wiki/%D0%9C%D1%83%D1%82%D0%B0%D1%86%D0%B8%D1%8F) и [естествен подбор](https://bg.wikipedia.org/wiki/%D0%95%D1%81%D1%82%D0%B5%D1%81%D1%82%D0%B2%D0%B5%D0%BD_%D0%BE%D1%82%D0%B1%D0%BE%D1%80) за усъвършенстване на решенията в задачите. В тях е заложен природния закон за естествения подбор, като специфични оптимизационни процедури осигуряват намиране на оптимално решение в условията на многомодалност.
  + [Обучение с утвърждение](https://bg.wikipedia.org/wiki/%D0%9E%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D1%81_%D1%83%D1%82%D0%B2%D1%8A%D1%80%D0%B6%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5): Група от методи за автоматично обучение, отличаващи се със способността си да функционират без необходимост от примерни решения на поставения проблем.

Невронните мрежи, генетичните алгоритми и размитата логика заимстват от природните закони и феномени, но принципно се различават. Те стоят в основата на изкуствения интелект като научно направление.

Тема 1/ Занятие 3/Лекция

# Морални, етични и правни аспекти на изкуствения интелект

Увод: Свидетели сме на бързо развитие на дигитализацията, която предизвика промени във всички аспекти на обществените отношения. В настоящето не съществува област на човешката дейност, която да не е засегната от процеса. Наблюдавания технологичен напредък е предпоставка за преминаването до нов модел на сътрудничество и взаимодействие между човека и машините, известен като **пета индустриална революция.** Отличителният белег на тази нова епоха според мнение на Франсиско Кесадо, специалист по иновации и конкуренция, изказано на конференция, организирана от Консултативната комисия по индустриални промени на Европейския икономически и социален комитет на 22 ноември 2019 г е съчетанието от конкуренция и сътрудничество между хората и роботитe [**Error! Bookmark not defined.**].

От друга страна изкуствения интелект поставя и редица предизвикателства, както пред международните институти, така и пред властите на отделните държави.

Развитието на изкуствения интелект и широкото му навлизане в множество сфери на обществото поставят на дневен ред редица морални и етични въпроси, за които тепърва се разработват правни регулации.

Благодарение на изкуствения интелект за първи път в историята на човечеството съществува възможност автономни системи да изпълняват сложни задачи, като обработка на големи количества информация, изчисляване и прогнозиране, учене и адаптиране към променящата се среда, разпознаване на определени ситуации и вземането на решения на базата на

всички тези данни. Това от своя страна открива нови възможности за развитие, модернизиране и усъвършенстване на всички области на човешкия живот, но същевременно крие и много рискове.

Рисковете произтичат най-вече от факта, че обществените отношения, които възникват по повод използването на ИИ все още не са цялостно правно регламентирани. Законодателят е изправен пред предизвикателство в опитите се да урегулира тази материя, поради факта, че изкуственият интелект повдига редица въпроси, които нямат еднозначен отговор.

Първата стъпка е хармонизирането им със съществуващите етични норми. В Европейския съюз, например, през март 2018 Комисията на Европейския съюз създава работна група по въпросите, свързани с изкуствения интелект като една от основните ѝ задачи е да съвмести новите технологии с етичните принципи. Еднозначно се посочва, че един от начините за вписването на изкуствения интелект в съществуващото законодателство минава през съгласуването му с етичните норми. В израз на желанието си да предприемат общ подход в тази сфера, на 10-ти април 2018 г. държавите членки подписват Декларация за сътрудничество в областта на изкуствения интелект[5], като в нея отново се обръща специално внимание на етичните принципи, които служат като гарант за отговорното използване на изкуствения интелект.

Съветът на Европа също предприема подобна стратегия по отношение на изкуствения интелект, а именно урегулира го чрез етичните принципи. Европейската комисия за ефикасност на правосъдието към Съвета на Европа създава проект за Европейска етична харта за използването на изкуствен интелект в съдебните системи и тяхната среда, определящ етичните принципи, свързани с използването на изкуствен интелект в съдебните системи[6].

Независимо в какъв план се разглежда употребата на изкуствения интелект, на ниво Европейски съюз, на ниво Съвет на Европа или на едно по- глобално ниво в рамките на международното публично право, винаги могат да се изведат следните положения, общи за всяко едно ниво:

* изкуственият интелект не бива да влиза в противоречие със съществуващата вече законодателна уредба;
* трябва да съществуват гарантира без всякакво съмнение, че употребата изкуственият интелект му няма да е в разрез с основните човешки права;
* трябва да съществуват достатъчно гаранции, че системите с изкуствен интелект няма да причинят преднамерена вреда.

# 2. Рамка за етичните аспекти на изкуствения интелект, роботиката и свързаните с тях технологии

Рамката за етичните аспекти на изкуствения интелект, роботиката и свързаните с тях технологии е документ на Европейския парламент от 20 октомври 2020 г. Този регламент се прилага по отношение на изкуствения интелект, роботиката и свързаните с тях технологии, когато каквато и да е част от тях се разработва, внедрява или използва в Съюза, независимо дали софтуерът, алгоритмите или данните, използвани или генерирани от такива технологии, се намират извън Съюза или нямат конкретно географско местоположение.

В документа са изведени на първо място следните съображения относно влиянието на изкуствения интелект:

Разработването, внедряването и използването на т.нар. „изкуствен интелект“, роботиката и свързаните с тях технологии се извършва от хора и

техният избор определя потенциала на тези технологии да носят полза на обществото. Посочените технологии, които имат потенциал да създават възможности за предприятията и ползи за гражданите и които могат да оказват пряко въздействие върху обществото, включително основните права и социалните и икономическите принципи и ценности, както и да оказват трайно влияние върху всички области на дейност, се насърчават и се развиват бързо;

Развитието на изкуствения интелект, роботиката и свързаните с тях технологии може да допринесе за опазването на околната среда и увеличаване на енергийната ефективност. Тези технологии могат да имат сериозни последици върху физическата и нематериалната неприкосновеност на отделни лица. ИИ са от стратегическо значение за транспортния сектор, включително поради това, че те повишават безопасността и достъпността на всички видове транспорт и създават нови възможности за заетост и по-устойчиви бизнес модели.

I„Регламентът относно етичните принципи за разработването, внедряването и използването на изкуствен интелект, роботика и свързаните с тях технологии“ се основава на следните принципи:

* ориентирани към човека, направени от човека и контролирани от човека изкуствен интелект, роботика и свързани с тях технологии;
* задължителна оценка на съответствието на високорисковия изкуствен интелект, роботика и свързаните с тях технологии;
* безопасност, прозрачност и отчетност;
* предпазни мерки и средства за правна защита срещу предубеденост и дискриминация;
* право на правна защита;
* социална отговорност и равенство между половете в изкуствения интелект, роботиката и свързаните с тях технологии;
* екологично устойчиви изкуствен интелект, роботика и свързани с тях технологии;
* зачитане на неприкосновеността на личния живот и ограниченията за използването на биометрично разпознаване;
* добро управление на изкуствения интелект, роботиката и свързаните с тях технологии, включително данните, използвани или генерирани от такива технологии.

Изтъква се необходимостта Европейския Съюз и държавите членки носят отговорност да гарантират ИИ е безопасен

**Ориентиран към човека и създаден от човека изкуствен интелект** Изкуствения интелект, следва да зачита изцяло човешкото достойнство, самостоятелността и самоопределянето на личността, да предотвратява вреди, да насърчава справедливостта, приобщаването и прозрачността, да премахва предубедеността и дискриминацията, включително по отношение на малцинствените групи, и да зачита и спазва принципите за ограничаване на отрицателните външни ефекти на използваната технология. Възможностите, основаващи се на изкуствения интелект, разчитат на големи информационни масиви, като е необходима критична маса от данни за обучение на алгоритмите и прецизиране на резултатите; приветства във връзка с това предложението на Комисията за създаване на общо пространство на данни в Съюза, за да се

засили обменът на данни и да се подпомогнат научните изследвания при пълно зачитане на европейските правила за защита на данните;

1. счита, че изкуственият интелект, роботиката и свързаните с тях технологии трябва да бъдат съобразени с нуждите на човека в съответствие с принципа, че тяхното развитие, внедряване и използване следва винаги да бъдат в услуга на хората – никога обратното – и следва да се стремят към подобряване на благосъстоянието и личната свобода, както и към опазване на мира, предотвратяване на конфликти и укрепване на международната сигурност, като същевременно се максимизират предлаганите ползи и се предотвратяват и намаляват рисковете от тях;
2. заявява, че разработването, внедряването и използването на високорисков изкуствен интелект, роботика и свързани с тях високорискови технологии, включително, но не само от хора, следва винаги да се водят от етични принципи и да са проектирани така, че да зачитат и допускат човешкия фактор и демократичен надзор, както и да дават възможност за възвръщане на човешкия контрол, когато е необходимо, чрез прилагане на подходящи контролни мерки;

Изкуственият интелект, роботиката и свързаните с тях технологии следва да бъдат разглеждани като високорискови, когато разработването, внедряването и използването им водят до значителен риск за причиняване на нараняване или вреда на отделни лица или на обществото в нарушение на основните права и правилата за безопасност,

;

освен това отбелязва, че за да се зачита човешкото достойнство, самостоятелност и безопасност, следва да се обърне дължимото внимание на жизненоважните модерни медицински уреди и на необходимостта от независими ползващи се с доверие органи, които да запазят средствата, необходими за предоставяне на услуги на лицата, носещи такива уреди, когато първоначалният разработчик или внедрител вече не ги предоставя;

високорисковите изкуствен интелект, роботика и свързани с тях технологии, включително софтуер, алгоритми и данни, използвани или генерирани от такива технологии, независимо от отрасъла, в който са разработени, внедрени и се използват, следва по проект да бъдат разработвани по сигурен, проследим, технически солиден, надежден, етичен и правно обвързващ начин и да бъдат предмет на независим контрол и преглед; счита по-специално, че всички участници във веригите за разработване и доставка на продукти и услуги с изкуствен интелект следва да носят правна отговорност, и подчертава необходимостта от механизми за гарантиране на отговорност и отчетност;

1. счита, че гражданите, включително потребителите, следва да бъдат информирани, когато си взаимодействат със система, използваща изкуствен интелект, по-специално за персонализиране на продукт или услуга за нейните потребители, дали и как могат да изключат или да ограничат такова персонализиране;
2. изтъква във връзка с това, че за да бъдат надеждни, изкуственият интелект, роботиката и свързаните с тях технологии трябва да бъдат технически устойчиви и точни;

Непредубеденост и недискриминация

27. припомня, че изкуственият интелект, в зависимост от начина, по който се разработва, използва и прилага, има потенциал да създава и засилва предубедеността, включително чрез предубеденост в базовите набори от данни, и следователно да създава различни форми на автоматизирана дискриминация, включително непряка дискриминация,;

30. твърдо вярва, че основните права на човека, залегнали в Хартата, следва да се спазват стриктно, за да се гарантира, че тези нововъзникващи технологии не водят до пропуски по отношение на защитата;

36. следователно счита, че всяко физическо или юридическо лице следва да може да потърси правна защита срещу решение, взето в негов ущърб чрез използването на изкуствен интелект,

Социална отговорност и баланс между половете

38. подчертава, че социално отговорният изкуствен интелект, роботика и свързани с тях технологии трябва да изпълняват роля, допринасяща за намирането на решения, които да защитават и насърчават основните права и ценности на нашето общество, като например демокрацията, върховенството

на закона, разнообразните и независими медии и обективната и свободно достъпна информация, здравето и икономическия просперитет,

1. счита, че изкуственият интелект може да спомогне за по-доброто използване на уменията и компетентностите на лицата с увреждания, и че прилагането на изкуствения интелект на работното място може да допринесе за приобщаващи пазари на труда и по-високи равнища на заетост за лицата с увреждания;

Околна среда и устойчивост

1. заявява, че изкуственият интелект, роботиката и свързаните с тях технологии следва да се използват от правителствата и от бизнеса в полза на хората и планетата, да допринасят за постигането на целите за устойчиво развитие, за опазването на околната среда, за неутралност по отношение на климата и за целите на кръговата икономика; счита, че разработването, внедряването и използването на тези технологии следва да допринасят за прехода към екологосъобразна икономика, за опазване на околната среда, за минимизиране и компенсиране на вредите, причинени на околната среда в рамките на жизнения им цикъл и по цялата верига за доставки на тези технологии, в съответствие с правото на Съюза;
2. предвид значителното им въздействие върху околната среда, за целите на предходния параграф въздействието върху околната среда във връзка с разработването, внедряването и използването на изкуствения интелект, роботиката и свързаните с тях технологии може да се оценява през целия им жизнен цикъл от органите, отговарящи за съответния сектор, когато това е приложимо и целесъобразно; тази оценка би могла да включва оценка на

въздействието във връзка с добива на необходимите материали, потреблението на енергия и емисиите на парникови газове, причинени от тяхното разработване, внедряване и използване;

1. предлага за целите на разработването на отговорни авангардни решения във връзка с изкуствения интелект потенциалът на изкуствения интелект, роботиката и свързаните с тях технологии да бъде проучен, стимулиран и максимизиран чрез отговорни научни изследвания и развойни дейности, което изисква мобилизирането на ресурси от Съюза и от неговите държави членки;
2. подчертава факта, че разработването, внедряването и използването на тези технологии осигурява възможности за насърчаване на целите за устойчиво развитие, очертани от ООН, глобалния енергиен преход и декарбонизацията;
3. счита, че целите за социална отговорност, равенство между половете, опазване на околната среда и устойчивост следва да не засягат съществуващите общи и секторни задължения в тези области; счита, че следва да бъдат определени насоки за изпълнение с незадължителен характер, които да се прилагат за разработчиците, внедрителите и ползвателите, по-специално по отношение на високорисковите технологии, във връзка с методологията за оценка на тяхното съответствие с настоящия регламент и постигането на тези цели;
4. призовава Съюза да насърчава и финансира развитието на ориентирани към човека изкуствен интелект, роботика и свързани с тях технологии, насочени към предизвикателствата в областта на околната среда и климата и

гарантиращи зачитането на основните права, чрез използването на стимули в сферата на данъчното облагане, възлагането на обществени поръчки и др.;

1. подчертава, че въпреки настоящия значителен въглероден отпечатък в резултат от разработването, внедряването и използването на изкуствения интелект, роботиката и свързаните с тях технологии, включително автоматизираните решения и машинното самообучение, тези технологии могат да допринесат за намаляване на настоящия екологичен отпечатък на сектора на ИКТ; подчертава, че тези и други подходящо регулирани сродни технологии следва да бъдат критични фактори за постигане на целите на Зеления пакт, целите на ООН за устойчиво развитие и Парижкото споразумение в много различни сектори и следва да засилят въздействието на политиките за опазване на околната среда, например политиките във връзка с намаляването на отпадъците и влошаването на състоянието на околната среда;
2. призовава Комисията да извърши проучване относно въздействието на въглеродния отпечатък от използването на технологии с ИИ, както и относно положителните и отрицателните последици от прехода към използването на технологии с ИИ за потребителите;
3. отбелязва, че предвид все по-честото разработване на приложения с ИИ, които изискват изчислителни ресурси, ресурси за съхранение и енергийни ресурси, въздействието на системите с ИИ върху околната среда следва да се разглежда през целия им жизнен цикъл;
4. счита, че в някои области като здравеопазването крайната отговорност трябва да се носи от физическо или юридическо лице; подчертава необходимостта от проследими и публично достъпни данни за обучение за алгоритмите;
5. подчертава ползите от ИИ за профилактиката, лечението и контрола на заболяванията, пример за което е фактът, че изкуственият интелект прогнозира епидемията от COVID-19 преди СЗО; настоятелно призовава Комисията да предостави на Европейския център за профилактика и контрол върху заболяванията (ECDC) регулаторната рамка и ресурсите за независимо събиране на необходимите анонимизирани здравни данни в реално време, в сътрудничество с държавите членки, за да се справи, наред с другото, с проблемите, които се появиха при кризата с COVID-19;

Неприкосновеност на личния живот и биометрично разпознаване

1. отбелязва, че генерирането и използването на данни, включително и лични данни, като например биометрични данни, в резултат на разработването, внедряването и използването на изкуствен интелект, роботика и свързани с тях технологии, бързо се увеличават, като по този начин се подчертава необходимостта от зачитане и прилагане на правата на гражданите за неприкосновеност на личния живот и за защита на личните данни, в съответствие с правото на Съюза;

Добро управление

72. подчертава, че подходящото управление при разработването, внедряването и използването на изкуствен интелект, роботика и свързани с тях технологии, и по-специално високорискови технологии, чрез въвеждането на мерки, насочени към отчетността и преодоляването на потенциални рискове във връзка с предубеденост и дискриминация, може да повиши безопасността и доверието на гражданите в тези технологии;

# Сигурност и отбрана

подчертава, че политиките на Европейския съюз и неговите държави членки в областта на сигурността и отбраната се ръководят от принципите, залегнали в Хартата, и от тези на Устава на Организацията на обединените нации, както и от общо разбиране за универсалните ценности за зачитане на ненакърнимите и неотчуждими права на човешката личност, на човешкото достойнство, свободата, демокрацията, равенството и правовата държава; подчертава, че тези универсални ценности трябва да се зачитат при всички свързани с отбраната усилия, полагани в рамките на Съюза, като същевременно се насърчават мирът, сигурността и напредъкът в Европа и в света;

# Транспорт

103. изтъква потенциала на използването на изкуствения интелект, роботиката и свързаните с тях технологии за всички автономни средства за автомобилен, железопътен, воден и въздушен транспорт, както и за поощряването на преминаването към други видове транспорт и интермодалността, тъй като тези технологии могат да допринесат за постигането на оптималното съчетаване на

видовете транспорт при транспортиране на стоки и пътници; освен това подчертава, че те имат потенциал да повишат ефикасността на транспорта, логистиката и потоците на движение и да повишат безопасността, интелигентността и екологосъобразността на всички видове транспорт; посочва, че етичният подход към ИИ може да се разглежда и като система за ранно предупреждение, по-специално по отношение на безопасността и ефикасността на транспорта;

# Образование и култура

Необходимо е изготвяне на критерии за разработването, внедряването и използването на ИИ, като се има предвид въздействието им върху образованието, медиите, младежта, научните изследвания и секторите на културата и творчеството, чрез разработване на показатели и определяне на принципи за етично отговорно и прието ползване на технологиите с ИИ, които могат да бъдат прилагани по подходящ начин в тези области, включително необходимостта от ясен режим на отговорност за продукти, които са резултат от използването на ИИ;

Съдържанието на Регламента, представено в съкратен обем е следното::

# Етични принципи на изкуствения интелект, роботиката и свързаните с тях технологии

1. В рамките на Съюза изкуственият интелект, роботиката и свързаните с тях технологии, включително софтуерът, алгоритмите и данните, използвани или генерирани от такива технологии, се разработват, внедряват и използват в съответствие с правото на Съюза, при пълно зачитане на човешкото достойнство, автономия и безопасност, както и на останалите основни права, определени в Хартата.
2. Всяко обработване на лични данни, извършвано в контекста на разработването, внедряването и използването на изкуствен интелект, роботика и свързани с тях технологии, включително лични данни, извлечени от нелични данни, и биометрични данни, се извършва в съответствие с Регламент (ЕС) 2016/679 и Директива 2002/58/ЕО.
3. Съюзът и неговите държави членки насърчават научноизследователските проекти, насочени към намиране на решения, основаващи се на изкуствен интелект, роботика и свързани с тях технологии, които имат за цел насърчаване на социалното приобщаване, демокрацията, плурализма, солидарността, справедливостта, равенството и сътрудничеството.

# Ориентиран към човека и създаден от човека изкуствен интелект

* 1. Всички високорискови технологии, включително софтуер, алгоритми и данни, използвани или генерирани от такива технологии, се разработват, внедряват и използват по начин, който гарантира пълен надзор от страна на човека по всяко време.
  2. Технологиите, посочени в параграф 1, се разработват, внедряват и използват по начин, който дава възможност при необходимост човешкият контрол да се възвърне, включително чрез промяна или спиране на тези технологии.

# Безопасност, прозрачност и отчетност

1. Всички високорискови технологии с изкуствен интелект, роботика и свързаните с тях технологии, включително софтуерът, алгоритмите и данните,

използвани или генерирани от такива технологии, се разработват, внедряват и използват по начин, който гарантира, че те:

а) се разработват, внедряват и използват по устойчив начин, така че да гарантират подходящо ниво на сигурност, като се придържат към минимално базово равнище, което е пропорционално на установения риск, и също така ниво на сигурност, което предотвратява експлоатацията на технически слабости за злонамерени или незаконни цели;

б) се разработват, внедряват и използват по сигурен начин, който гарантира наличието на предпазни механизми, включващи резервен план и резервни действия в случай на риск за безопасността или сигурността;

в) се разработват, внедряват и използват по начин, който гарантира надеждни резултати, каквито ползвателят разумно може да очаква, по отношение на постигането на целите и изпълнението на дейностите, за които са били създадени, включително като се гарантира, че всички операции са възпроизводими;

г) се разработват, внедряват и използват по начин, който гарантира, че изпълнението на целите и дейностите на конкретните технологии е точно; ако не могат да се избегнат отделни неточности, системата показва, доколкото е възможно, на внедрителите и ползвателите вероятността от грешки и неточности чрез подходящи средства;

д) се разработват, внедряват и използват по лесно обясним начин, така че да се гарантира, че е възможно техническите процеси на технологиите да бъдат обект на преглед;

е) се разработват, внедряват и използват по такъв начин, че да информират своите ползватели, че взаимодействат със системи за изкуствен интелект, като разкриват надлежно и цялостно своите способности, точност и ограничения пред разработчиците, внедрителите и ползвателите на изкуствен интелект;

ж) се разработват, внедряват и използват в съответствие с член 6 по начин, който позволява, в случай на несъответствие с характеристиките за безопасност, посочени в букви а) – ж), съответните функции да бъдат временно изключени и да се възвърне предишното им състояние, възстановяващо безопасните функционални възможности.

1. В съответствие с член 6, параграф 1, технологиите, посочени в параграф 1 от настоящия член, включително софтуер, алгоритми и данни, използвани или генерирани от такива технологии, се разработват, внедряват и използват по прозрачен и проследим начин, така че техните елементи, процеси и фази да са документирани съгласно възможно най-високите стандарти и да е възможно националните надзорни органи, посочени в член 18, да оценяват съответствието на тези технологии със задълженията, установени в настоящия регламент. По-конкретно, разработчикът, внедрителят или ползвателят на тези технологии отговаря за съответствието им с характеристиките за безопасност, посочени в параграф 1, и трябва да бъде в състояние да го докаже.
2. Разработчикът, внедрителят или ползвателят на технологиите, посочени в параграф 1, гарантира, че предприетите мерки за осигуряване на

съответствие с характеристиките за безопасност, посочени в параграф 1, могат да бъдат одитирани от националните надзорни органи, посочени в член 18, или, при целесъобразност, от други национални или европейски надзорни органи, свързани с този сектор.

# Непредубеденост и недискриминация

1. Софтуерите, алгоритмите или данните, използвани или генерирани от високорисков изкуствен интелект, роботика и свързани с тях технологии, които са разработени, внедрени или използвани в Съюза, не съдържат предубеждения и – без да се засяга параграф 2, не дискриминират на основания като раса, пол, сексуална ориентация, бременност, увреждане, физически или генетични характеристики, възраст, национално малцинство, етнос или социален произход, език, религия или убеждения, политически възгледи или гражданско участие, гражданство, гражданско или икономическо състояние, образование или съдимост.
2. Чрез дерогация от параграф 1 и без да се засяга правото на Съюза относно незаконната дискриминация, всяко диференцирано третиране на лица или групи от лица може да бъде оправдано само ако съществува обективна, разумна и законна цел, която е едновременно пропорционална и необходима, доколкото не съществува алтернатива, която би засегнала в по-малка степен принципа на равно третиране.

# Социална отговорност и равенство между половете

изкуствен интелект, роботика и свързани с тях технологии, включително софтуер, алгоритми и данни, използвани или генерирани от такива технологии, разработени, внедрени и използвани в Съюза, се разработват,

внедряват и използват в съответствие с приложимото право, принципи и ценности на Съюза по начин, който не допуска намеса в избори и не допринася за разпространението на дезинформация, зачита правата на работниците, насърчава качественото образование и цифровата грамотност, не увеличава неравнопоставеността между половете чрез предотвратяване на равните възможности за всички и не нарушава правата на интелектуална собственост и ограниченията или изключенията от тях.

# Екологична устойчивост

Високорисковият изкуствен интелект, високорисковата роботика и свързаните с тях високорискови технологии, включително софтуер, алгоритми и данни, използвани или генерирани от такива технологии, се оценяват по отношение на тяхната екологична устойчивост от националните надзорни органи, като се гарантира, че са въведени мерки за смекчаване и преодоляване на общото им въздействие по отношение на природните ресурси, потреблението на енергия, генерирането на отпадъци, въглеродния отпечатък, извънредната ситуация с изменението на климата и влошаването на околната среда, за да се гарантира съответствие с приложимото право на Съюза или национално право, както и с всички други международни екологични ангажименти, поети от Съюза.

# Зачитане на неприкосновеността на личния живот и защита на личните данни

Използването и събирането на биометрични данни с цел дистанционна идентификация на обществени места, като биометрично разпознаване или разпознаване на лица, носи специфични рискове за основните права и се внедрява или използва единствено от публичните органи на държавите членки за цели от съществен обществен интерес. Тези органи гарантират, че това

внедряване или използване е оповестено публично и е пропорционално и целенасочено, ограничено до конкретни цели и места, а също така и по времетраене, в съответствие с правото на Съюза и националното право, по- специално Регламент (ЕС) 2016/679 и Директива 2002/58/EО, и при надлежно зачитане на човешкото достойнство и автономност и на основните права, определени в Хартата, а именно правото на зачитане на неприкосновеността на личния живот и защитата на личните данни.

# Право на правна защита

Всяко физическо или юридическо лице има право да търси правна защита за нараняване или вреда, причинени от разработването, внедряването и използването на високорисков изкуствен интелект, роботика и свързани с тях технологии, включително софтуер, алгоритми и данни, използвани или генерирани от такива технологии, в нарушение на правото на Съюза и на задълженията, определени в настоящия регламент.

За разработването на тази лекция са използвани текстове от „Рамка от етични аспекти на изкуствения интелект, роботиката и свързаните с тях технологии“.

ЛИТЕРАТУРА

* 1. Рамка от етични аспекти на изкуствения интелект, роботиката и свързаните с тях технологии, достъпно на <https://www.europarl.europa.eu/>, посетено на 22.08.2022 г.

1. Тодорова, М., Видове изкуствен интелект – тесен, общ и супер изкуствен интелект. Рискове и етични проблеми, Институт за изследване на обществата и знанието, БАН, достъпен на [https://jesbg.com/mtodorova-](https://jesbg.com/mtodorova-vidove-izkustven-intelekt-tesen-obsht-i-super-izkustven-intelekt-riskove-i-eticheski-problemi/)

[vidove-izkustven-intelekt-tesen-obsht-i-super-izkustven-intelekt-riskove-i-](https://jesbg.com/mtodorova-vidove-izkustven-intelekt-tesen-obsht-i-super-izkustven-intelekt-riskove-i-eticheski-problemi/) [eticheski-problemi/](https://jesbg.com/mtodorova-vidove-izkustven-intelekt-tesen-obsht-i-super-izkustven-intelekt-riskove-i-eticheski-problemi/)

1. Bostrom, N. (2014). Superintellignece: Paths, Dangers, Strategies. Oxford university press.
2. [Chalmers, David](https://en.wikipedia.org/wiki/David_Chalmers). (2010).The Singularity: A Philosophical Analysis. (PDF). Journal of Consciousness Studies. 17: 7–65.
3. Goertzel, B., Penachin, C. (2007). Artificial general intelligence. Springer.
4. Harai, U. (2016). Hommo deus. A brief history of tomorrow. Vintage.
5. Kahneman, D. (2011). Thinking fast and slow. MacMillan.
6. Kurzweil, R. (2005). The Singularity is near. New York. The New York Times.
7. Leonhard,

G. (2016). Technology vs. Humanity. The coming clash between man and machine. Fast Future Publishing Ltd.

1. Vikhar, P. (2017). Evolutionary algorithms: A critical review and its future prospects. Computer Science. IEEE

Xplore, <https://ieeexplore.ieee.org/document/7955308/metrics#metrics>

Тема 1/ Занятие 4/Семинар

# Морални, етични и правни аспекти на изкуствения интелект

Увод:

Развитието на изкуствения интелект и широкото му навлизане в множество сфери на обществото поставят на дневен ред редица морални и етични въпроси, за които тепърва се разработват правни регулации.

Изкуствения интелект поставя и редица предизвикателства, както пред международните институти, така и пред властите на отделните държави.

Благодарение на изкуствения интелект за първи път в историята на човечеството съществува възможност автономни системи да изпълняват сложни задачи, като обработка на големи количества информация, изчисляване и прогнозиране, учене и адаптиране към променящата се среда, разпознаване на определени ситуации и вземането на решения на базата на всички тези данни. Това от своя страна открива нови възможности за развитие, модернизиране и усъвършенстване на всички области на човешкия живот, но същевременно крие и много рискове.

# Теми за дискусия по време на семинарното занятие:

* 1. **Предизвикателства и рискове, произтичащи от навлизането на изкуствения интелект във всички сфери на съвременното общество**.

Рисковете произтичат най-вече от факта, че обществените отношения, които възникват по повод използването на ИИ все още не са цялостно правно регламентирани. Законодателят е изправен пред предизвикателство в опитите се да урегулира тази материя, поради факта, че изкуственият интелект повдига редица въпроси, които нямат еднозначен отговор.

Обучаемите трябва да:

* Дефинират и да категоризират съществуващите и потенциалните предизвикателства и рискове, произтичащи от навлизането на изкуствения интелект;
* Да съпоставят по степен на вероятност и по степен на влияние съществуващите и потенциалните предизвикателства и рискове, произтичащи от използването на изкуствения интелект;
* Да посочат ползите за обществото и отделния гражданин и групи граждани от засиленото използване на изкуствен интелект.

# Морални измерения на разработването и употребата на изкуствен интелект.

Обучаемите трябва да:

* Дефинират и да категоризират потенциалните морални въпроси, възникващи по повод масовото използване на системи с изкуствени интелект изкуствения интелект;
* Да посочат и анализират публикации по тези въпроси;
* Да формулират и защитят своето виждане по формулирания въпрос.

# Законодателни средства за регулиране на влиянието на изкуствения интелект върху обществото.

* + 1. Обучаемите трябва да изложат своите знания по отношение на правните документи, регламентиращи употребата на изкуствения интелект на ниво Европейски съюз, на ниво Съвет на Европа, както и на глобално ниво в рамките на международното публично право.
    2. Да анализират документи и да изведат общи положения, характерни за всяко едно ниво по отношение на изискванията за полезност и безопасност на изкуствения интелект. Да познават наложените изисквания изкуствения интелект да създава възможности за предприятията и ползи за гражданите в областите:
* Сигурност и отбрана
* Екология и опазване на околната среда
* Транспорт
* Здравеопазване и медицина
* Образование и култура
* Управление
  + 1. Да анализират и да предложат правна основа по отношение на следните регламенти:
* изкуственият интелект не бива да влиза в противоречие със съществуващата вече законодателна уредба;
* трябва да съществуват гарантира без всякакво съмнение, че употребата изкуственият интелект му няма да е в разрез с основните човешки права;
* трябва да съществуват достатъчно гаранции, че системите с изкуствен интелект няма да причинят преднамерена вреда.

# Да познават и анализират съдържанието на „Рамка за етичните аспекти на изкуствения интелект, роботиката и свързаните с тях технологии“

Рамката за етичните аспекти на изкуствения интелект, роботиката и свързаните с тях технологии е документ на Европейския парламент от 20 октомври 2020 г. Този регламент се прилага по отношение на изкуствения интелект, роботиката и свързаните с тях технологии, когато каквато и да е част от тях се разработва, внедрява или използва в Съюза, независимо дали софтуерът, алгоритмите или данните, използвани или генерирани от такива технологии, се намират извън Съюза или нямат конкретно географско местоположение.

Разработването, внедряването и използването на изкуствен интелект, роботика и свързаните с тях технологии“ се основава на следните етични принципи:

* ориентирани към човека, направени от човека и контролирани от човека изкуствен интелект, роботика и свързани с тях технологии;
* задължителна оценка на съответствието на високорисковия изкуствен интелект, роботика и свързаните с тях технологии;
* безопасност, прозрачност и отчетност;
* предпазни мерки и средства за правна защита срещу предубеденост и дискриминация;
* право на правна защита;
* социална отговорност и равенство между половете в изкуствения интелект, роботиката и свързаните с тях технологии;
* екологично устойчиви изкуствен интелект, роботика и свързани с тях технологии;
* зачитане на неприкосновеността на личния живот и ограниченията за използването на биометрично разпознаване;
* добро управление на изкуствения интелект, роботиката и свързаните с тях технологии, включително данните, използвани или генерирани от такива технологии.

Изтъква се необходимостта Европейския Съюз и държавите членки носят отговорност да гарантират ИИ е безопасен

ЛИТЕРАТУРА

1. Рамка от етични аспекти на изкуствения интелект, роботиката и свързаните с тях технологии, достъпно на <https://www.europarl.europa.eu/>, посетено на 22.08.2022 г.
2. Тодорова, М., Видове изкуствен интелект – тесен, общ и супер изкуствен интелект. Рискове и етични проблеми, Институт за изследване на обществата и знанието, БАН, достъпен на [https://jesbg.com/mtodorova-vidove-izkustven-intelekt-tesen-obsht-i-super-](https://jesbg.com/mtodorova-vidove-izkustven-intelekt-tesen-obsht-i-super-izkustven-intelekt-riskove-i-eticheski-problemi/) [izkustven-intelekt-riskove-i-eticheski-problemi/](https://jesbg.com/mtodorova-vidove-izkustven-intelekt-tesen-obsht-i-super-izkustven-intelekt-riskove-i-eticheski-problemi/)
3. Bostrom, N. (2014). Superintellignece: Paths, Dangers, Strategies. Oxford university press.
4. [Chalmers, David](https://en.wikipedia.org/wiki/David_Chalmers). (2010).The Singularity: A Philosophical Analysis. (PDF). Journal of Consciousness Studies. 17: 7–65.
5. Goertzel, B., Penachin, C. (2007). Artificial general intelligence. Springer.
6. Harai, U. (2016). Hommo deus. A brief history of tomorrow. Vintage.
7. Kahneman, D. (2011). Thinking fast and slow. MacMillan.
8. Kurzweil, R. (2005). The Singularity is near. New York. The New York Times.
9. Leonhard,

G. (2016). Technology vs. Humanity. The coming clash between man and machine. Fast Future Publishing Ltd.

1. Vikhar, P. (2017). Evolutionary algorithms: A critical review and its future prospects. Computer Science. IEEE

Xplore, <https://ieeexplore.ieee.org/document/7955308/metrics#metrics>

Тема 1/ Занятие 5/ Лекция

# Обща концепция за изкуствени невронни мрежи. Особености на биологична невронна мрежа.

Концепцията за изкуствените невронни мрежи възниква паралелно с изучаването на процесите, протичащи в мозъка с цел тяхното моделиране.

Изкуствените невронни мрежи подобно останалите направления на изкуствения интелект, използват възможностите на съвременните софтуерни и хардуерни средства за да възпроизведат основните характеристики на биологичната нервна система. За разлика от тях обаче, те не само притежават свойствата, но и копират структурата на живата нервна система и принципа на нейното функциониране.

Изкуствените невронни мрежи са интелигентни системи за паралелни изчисления, която работят по начин близък до механизмите на протичане на основните процеси човешкия мозък. изчислителни структури. Те копират елементарните биологични процеси за обработка и съхранение на информацията и с помощта на редица опростявания симулират отделни функции на човешката нервна система.

Въпреки, че компютърната система разполага с по-голям брой и с по- бързодействащи изчислителни единици от човешкия мозък, човешката и изобщо биологичната нервната система я превъзхожда със способността за самоорганизация, за обучение, за обобщаване на данни и толерантност към грешки.

**Способността за учене** е една от основните разлики между процесора и биологичната невронна мрежа на живите организми, които изкуствените невронни мрежи имат за цел да премахнат.

Нервната система на живите организми има способност да се реорганизира през целия им живот и следователно е способна да учи и да се самообучава (без предишен опит, на базата на външни стимули). За решаване на различни задачи в традиционната компютърна наука се използва т.нар. алгоритмичен подход, който се състои в създаване на абстрактен модел на разглежданата задача, разработване алгоритъм за решаването Ѝ и последващо изпълнение на заложената последователност от стъпки от компютърна система. За преодоляване на тази статичност в изкуствените невронни мрежи е вградена способност да се учат (има се предвид и двете: обучение и самообучение), разработени са различни алгоритми с тази цел.

Следствие на способността за учене е развитието на способностите за обобщаване и асоцииране на данни. След успешно обучение невронната мрежа може да намери разумни решения за подобни проблеми от същия клас, за които не са били изрично обучени.

Това от своя страна води до висока степен на устойчивост на грешки възникващи в следствие на неточни, непълни и или неправилно формулирани входни данни.

Толерантността към грешки е характерно свойство на биологичните невронни мрежи. Изследванията показват, че биологичния мозък запазва когнитивни способности при унищожаване на част от мозъчните клетки – устойчивост към вътрешни грешки. Мозъчни клетки могат да бъдат в редица случаи – в следствие на заболяване, в резултат от въздействие на околната среда или на злоупотреба с алкохол или други субстанции. Мозъкът е толерантен и към външни грешки – човешкото същество е способно да различи неясен образ - например нечетлив почерк или замъглен силует.

Същата тази способност е характерна за технологията на изкуствените невронни мрежи.

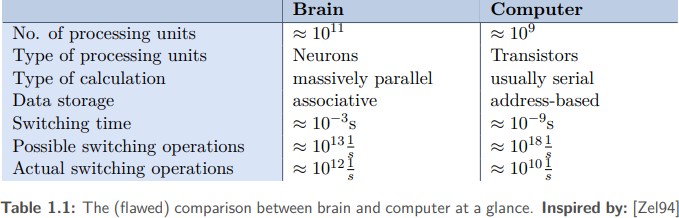
Друга съществена разлика между биологичния мозък и процесора се състои в това, че по-голямата част от мозъка **работи непрекъснато**, докато по- голямата част от компютърът е само пасивно хранилище на данни.

И не на последно място важна характеристика на мозъчните клетки е

# паралелното функциоиране.

Таблица 1. Сравнителна характеристика между биологичен мозък и

компютър



# Биoлогична невронна мрежа

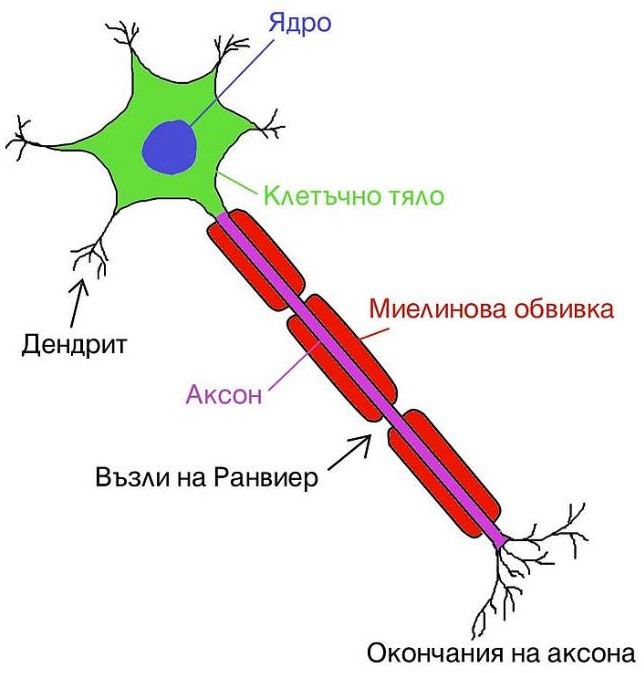
Според неврофизиологията човешкият мозък се състои от приблизително 1011 неврона, свързани в паралелно работеща структура посредством около 1014 нервни връзки.

Предаването на сигнали между невроните е сложен физико-химичен процес, при който се повишава или намалява електричният потенциал в клетката. Ако той премине определен праг, се изпраща сигнал с определена чувствителност и сила и тогава невронът се счита за активиран. Времето за превключване на неврона е 10−3 секунди.

След известен период на покой, невронът може да получи ново активиране. Всички неврони действат паралелно и асинхронно, което обяснява високата скорост на реагиране при човека.

В сравнение с него компютърната система се състои от 109 транзистори с време на превключване 10-9 секунди.

Невронът (от старогръцки: νεῦρον – влакно, нерв), наричан още нервна клетка, е основна структурно-функционална единица на нервната система. Изпълнява функцията да приема, обработва и предава информация посредством електрически и химически сигнали.



Фиг. 1. Структура на нервна клетка

Както е изобразено на фигура 1, се състои се от звездовидно тяло, в което се намира ядрото на клетката, голям брой къси израстъци - дендрити и дълъг израстък - аксон.

Дендритите приемат сигналите на другите нервни клетки. Аксонът, може да достигне дължина до един метър. Неговата функция е да предава сигналите към другите неврони. Аксонът може да притежава хиляди разклонения, което позволява един аксон да предава сигнали до множество неврони. Краищата на тези разклонения се намират върху синапсите, чрез които се извършва предаването на сигналите към други нервни клетки, мускулни клетки или клетки на жлезите. Най-често това се реализира чрез използването на невротрансмитери (по химически път) и по-рядко по електрически начин. Някои неврони могат да предават сигнални вещества и в кръвните пътища.

Пространството между аксона на един неврон и дендрита на друг неврон се нарича синапс. Синапсът е изолиран от останалото междуклетъчно пространство. Синапсите са два вида: електрични - по-бързи, но малко разпространени в живите организми и химични - широко разпространени. Химичните синапси могат да бъдат както стимулиращи, така и потискащи в зависимост от невротрансмитера, отделян от пресинаптичния неврон.

Функционирането на неврона се обуславя от неговата способност да приема и предава електрически импулси.

Биологичните неврони са свързани един с друг посредством претеглени връзки и когато биват стимулирани те предават (електрически или химичен) импулс през аксона. От аксона импулса не се прехвърля директно към следващите неврони, а първо трябва да преодолее синаптичната цепнатина, където сигналът се променя отново чрез променливи химични процеси.

В приемащия неврон различните входни импулси, след като са били обработени в синаптичната цепнатина, се обобщават в един единствен импулс. В зависимост от това как невронът е стимулиран от сумираните входни сигнали, самият неврон излъчва импулс или не – по този начин изходът е нелинеен и не е пропорционален на натрупания вход.

Изкуствените невронни мрежи са изчислителни структури, които копират описаните в т.2 биологичните процеси без да са точни и пълни модели на техните аналози. Силно опростени са и реализират само малък брой от техните добре изучени и изяснени структурни и функционални характеристики, свързани с за обработка и съхранение на информацията.

Работата на човешкия мозък се моделира посредством паралелно изпълнение на прости числови операции от множество процесорни елементи

– изкуствени неврони, свързани по определен начин. Ето защо по своята същност изкуствените невронни мрежи са многопроцесорни архитектури, които в които всеки отделен елемент обменя резултати от обработката със съседните на него посредством претеглени връзки. Структурата на връзките и теглата им са носителите на знания в изкуствените невронни мрежи.

Един силно опростен абстрактен модел на изкуствена невронна мрежа се състои от елементарни единици за обработка - невроните, и насочени, претеглени връзки между тях.

Основните градивни елементи на изкуствените невронни мрежи представляват силно опростени модели на биологичните неврони, от които мозъкът е съставен.

Всеки неврон приема входни сигнали от други неврони (или от външната среда) и ги използва за изчисляване на изходен сигнал. Всеки неврон работи

**паралелно** с останалите без **централен контрол.** Така се реализират масивни паралелни изчисления. Обновяването на сигналите води до обновяване на невроните, което може да се осъществява синхронно или асинхронно. При синхронното актуализиране всички процесорни елементи променят активностите си едновременно, докато при асинхронното обновяването е по различно време за всеки от тях. Броят на невроните в един невронен модел зависи от спецификата на разработваното приложение.

Теглото на всяка връзка определя силата на сигнала, предавана между два процесорни елемента. Например, ако теглото е 0.5, това означава, че само половината от сигнала се предава на получаващия неврон. Ако стойността на теглото е положителна величина, връзката се счита за възбуждаща, а когато е отрицателна – за подтискаща. Тегловите коефициенти на връзките между процесорните единици позволяват представяне на знания, които трудно се извличат в дефинируема форма.

Типична характеристика на невронната мрежа е природата и на “черна кутия”. Поради вграждането на знанията в структурата от връзки и тегла е трудно да се каже, защо се генерира определен резултат. Това изисква периодична проверка на модела. Необходимостта от преоценка на сигурността им на работа е особено наложителна за динамични области.

Литература:

* 1. Winston, P., Artificial Intelligence, Third Edition, Addison-Wesley, 1992.
  2. Rich E., Knight K., Artificial Intelligence, McGraw-Hill, (2017), ISBN- 10: 0070087709
  3. Minsky M. and Papert S., “Perceptrons,” MIT Press, Cambridge, 1969
  4. Dimiter Dobrev. Formal Definition of Artificial Intelligence. International Journal "Information Theories & Applications", vol.12, Number 3, 2005, pp.277-285. <http://www.dobrev.com/AI/>
  5. [15] Dimiter Dobrev. Parallel between definition of chess playing program and definition of AI. International Journal "Information Technologies & Knowledge ", vol.1, Number 2, 2007, pp.196-199.

# Тема 1/Занятие 6/Лекция

**Архитектура и функционални особености на изкуствените невронни мрежи. Структура на изкуствен неврон.**

# Увод

Изкуствените невронни мрежи са изчислителни структури, които

копират описаните процесите в биологичната нервна система без да са точни и пълни модели на техните аналози. Силно опростени са и реализират само малък брой от техните добре изучени и изяснени структурни и функционални характеристики, свързани с за обработка и съхранение на информацията.

Работата на човешкия мозък се моделира посредством паралелно изпълнение на прости числови операции от множество процесорни елементи

* изкуствени неврони, свързани по определен начин. Ето защо по своята същност изкуствените невронни мрежи са многопроцесорни архитектури, които в които всеки отделен елемент обменя резултати от обработката със съседните на него посредством претеглени връзки. Структурата на връзките и теглата им са носителите на знания в изкуствените невронни мрежи.

# Структурно и функционално описание на изкуствена невронна мрежа

Един силно опростен абстрактен модел на изкуствена невронна мрежа се състои от елементарни единици за обработка - невроните, и насочени, претеглени връзки между тях.

Основните градивни елементи на изкуствените невронни мрежи представляват силно опростени модели на биологичните неврони, от които мозъкът е съставен.

Всеки неврон приема входни сигнали от други неврони (или от външната среда) и ги използва за изчисляване на изходен сигнал. Всеки неврон работи **паралелно** с останалите без **централен контрол.** Така се реализират масивни паралелни изчисления. Обновяването на сигналите води до обновяване на невроните, което може да се осъществява синхронно или асинхронно. При синхронното актуализиране всички процесорни елементи променят активностите си едновременно, докато при асинхронното обновяването е по различно време за всеки от тях. Броят на невроните в един невронен модел зависи от спецификата на разработваното приложение.

Теглото на всяка връзка определя силата на сигнала, предавана между два процесорни елемента. Например, ако теглото е 0.5, това означава, че само половината от сигнала се предава на получаващия неврон. Ако стойността на теглото е положителна величина, връзката се счита за възбуждаща, а когато е отрицателна – за подтискаща. Тегловите коефициенти на връзките между процесорните единици позволяват представяне на знания, които трудно се извличат в дефинируема форма.

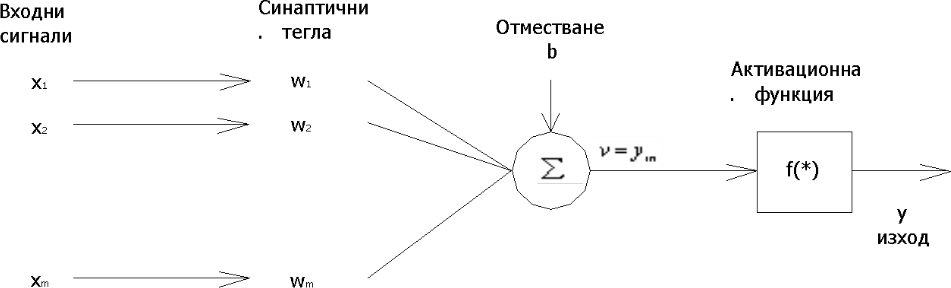
Типична характеристика на невронната мрежа е природата и на “черна кутия”. Поради вграждането на знанията в структурата от връзки и тегла е трудно да се каже, защо се генерира определен резултат. Това изисква периодична проверка на модела. Необходимостта от преоценка на сигурността им на работа е особено наложителна за динамични области.

Математически невронната мрежа може да се опише с наредената тройка *(N, V, w)* , където *N* е множество от неврони, *V* е множеството {(i, j)| i,jϵℕ}, чийто елементи са връзки между неврон *i* и неврон j. Функцията *w: V→ℝ* дефинира теглата *w*, където *w((i, j)),* теглото на връзката между неврон *i* и неврон *j*. При липсата на връзка между двойката неврони *i* и *j* тегло *wi,j* или се дефинира или му се присвоява стойност 0.

Теглата могат да бъдат представени в теглови вектор W или квадратна теглова матрица W, в която номера на реда на съответства на неврона източник, а номерът на колоната указва неврона - цел.

# Модел на изкуствен неврон

Елементарните преобразуватели на сигналите в изкуствените невронни мрежи са изкуствените неврони. Те са силно опростени модели на биологичните неврони и реализират ограничен брой от техните изучени структурни и функционални характеристики. Техническото приближение на биологичния неврон, представен на фиг. 2.



Фиг.1. Модел на изкуствен неврон

В природата един неврон получава средно импулси от 103 до 104 други неврони, това определя факта, че входния сигнал на изкуствените неврони се състои от много компоненти, следователно е **вектор**. На входовете x1, x2,…,xm постъпват сигналите към неврона. Те могат да са сигнали от външната среда или сигнали от изходите на други неврони. **Входовете** са модел на дендритите на биологичния неврон.

Синапсите променят входния сигнал, като го усилват или потискат. В техническите невронни мрежи входовете също се обработват предварително като се умножават по число (теглото) – претеглят се.

На всеки вход (фиг. 2) е присвоено **тегло** (синаптично тегло) wj, j=1,2,…,m, което моделира силата на връзката при предаване на сигнала посредством синапса, свързан със съответния дендрит.

Наборът от тези тегла представлява съхранението на информация в невронната мрежа - както в биологичен оригинал, така и в техническата му адаптация.

**Изходът** е модел на аксона. Изходните сигнали са скаларни, което означава, че се състоят само от един компонент. Няколко скаларни изхода на свой ред формират векторния вход на друг неврон.

За да се генерира скаларен резултат от векторни входни компоненти е необходимо те да обобщени.

В тялото на биологичния неврон входните данни се обобщават в един импулс в съответствие с химическата промяна, т.е. те се натрупват. От техническа страна това често се реализира чрез претеглената сума.

**Сумиращата функция** (**суматор)** за входа и специфичната за мрежата активираща нелинейна функция моделират агрегирането на входните сигнали в тялото на неврона.

С цел по-лесно формиране на праговата активационна функция, при модела на неврона се въвежда и сигнал с постоянна стойност b, наречен отместване (bias). В общия случай активационната функция е нелинейна и изходния сигнал на суматора се изчислява с формулата :

*m*

*v*  *yin*

 *wj xj*  *b*

*j* 1

(1)

При тези опростени модели не се отчитат редица характеристики на реалните неврони като:

* + възможността да се реагира по непрекъснат начин на входните сигнали;
  + предава се последователност от импулси, а не един изходен сигнал. В резултат на опростяването при изкуствения неврон се генерира само един изходен сигнал и така се пренебрегва голяма част от реално подаваната информация. Това е допустимо, тъй като повечето експерти считат тази допълнителна информация за несъществена в модела.

С развитието на изследванията в областта на невронните мрежи се заменя праговата функция с по-обща нелинейна функция, наречена функция на активност (предавателна функция). Известни са и се прилагат следните основни видове предавателни функции (фиг. 3):

* линейна предавателна функция (Linear Transfer Function); - линейна предавателна функция за положителните стойности с фиксирана долна граница (Positive Linear Transfer Function);
* линейна предавателна функция за положителните стойности с фиксирани граници на насищане (Saturating Linear Transfer Function);
* линейна предавателна функция с фиксирани граници на насищане, симетрични спрямо абсцисната ос (Symmetric Saturating Linear Transfer Function);
* релейна предавателна функция с нулево и положително ниво (Hard- Limit Transfer Function);
* релейна предавателна функция със симетрични нива спрямо абсцисната ос (Symmetric Hard-Limit Transfer Function);
* логаритмично-сигмоидална предавателна функция (Logarithmic Sigmoid Transfer Function);
* тангенциално-сигмоидална предавателна функция (Tangent Sigmoid

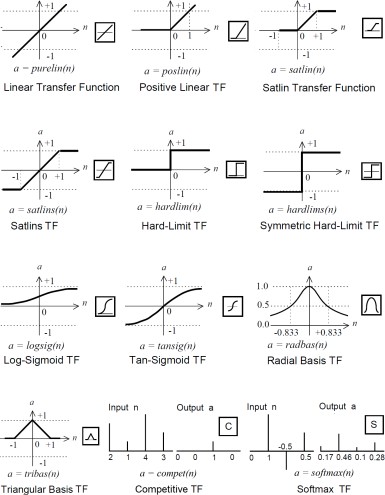
TF);

* предавателна функция с радиален базис (Radial Basis Transfer Function);
* предавателна функция с триъгълен базис (Triangular Basis Transfer

Function);

* + „състезателна“ предавателна функция (Competitive Transfer Function);
  + „softmax” предавателна функция (SoftМax Transfer Function: a = en /

Σen)



Фиг. 2. Видове предавателни функции

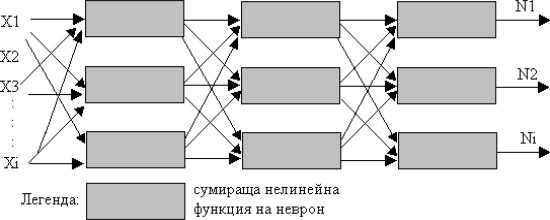
# Топология на изкуствените невронни мрежи

Невронните мрежи биват еднослойни, двуслойни и многослойни. Най- често се срещат невронни мрежи, съставени от няколко обособени слоя елементи, при които невроните от най-ниския слой играят роля на входни устройства на мрежата, а тези от най-горния слой-на изходни /т.е. извеждат резултата, получен въз основа на входните сигнали и теглата на връзките в системата/.

При многослойните мрежи има поне един скрит междинен слой.

Изходният сигнал от един неврон може да бъде вход за други неврони.

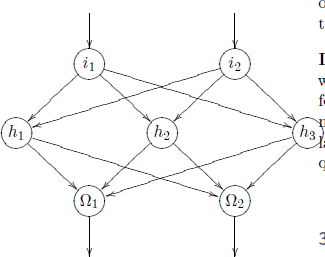
По подобен начин човек може да използва една и съща информация, отдавайки и различно значение за отделни цели. Така могат да се създават многослойни невронни мрежи. Модел на такава мрежа е представен на фиг. 5.



Фиг. 3. Многослойна изкуствена невронна мрежа

При многослойна невронна мрежа някои от нивата се разглеждат като скрити и това са тези от тях, чиито входни и изходни сигнали остават в рамките на мрежата. Например, на схемата скрито ниво е междинното, второ ниво.

# Многослойна мрежа с право предаване на сигнала.

Този тип мрежи се състоят от два или повече слоеве и връзки между слоевете, като посоката на предаване на информацията е от по-долен кум по- горен слой, както е показано на фиг.4

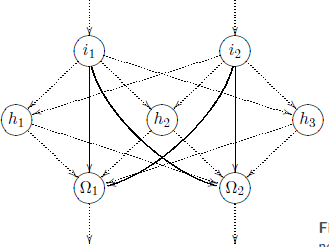
Фиг. 4. Трислойна невронна мрежа с право предаване на сигнала

В многослойните мрежи с право предаване на сигнала невроните са групирани в следните слоеве: един входен слой, един или повече междинни (изчислителни), слоеве, които са недостъпни от среда, външна за системата и един изходен слой.

Всеки неврон от даден слой има връзка пряка само с неврони от следващия по-горен слой, по посока към изходния слой. Не съществуват връзки между невроните от един и същи слой. Възлите на графа на фигура ….. изобразяват неврони, съответно *i1, i2* – принадлежащи на входния слой, *h1, h2* и *h3* – принадлежащи на междинния слой и Ώ1 и Ώ2 – принадлежащи на изходния слой. А стрелките показват връзките между тях.

Съществуват невронни мрежи при които всеки неврон от даден слой е свързан с всички неврони от следващия слой. Такива мрежи се наричат напълно свързани.

Друга разновидност на описваните мрежи, са тези при които съществуват връзки между неврони от несъседни слоеве (фиг )



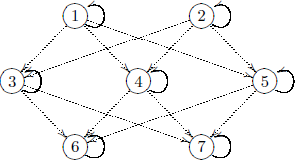
Фиг. 5

# Невронни мрежи с рекурентни връзки

Рекурсията е процес при който неврон влияе върху себе си по някакъв начин или чрез някаква връзка. Рекурсивните мрежи не винаги имат изрично дефинирани входни или изходни неврони.

# Мрежи с директна рекурсия

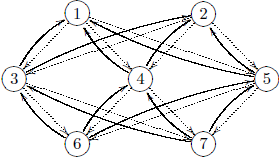
При мрежите с директна рекурсия невроните са свързани със самите себе си (фигура 6 ). В резултат на това те потискат или увеличават сигнала си до достигане на границата си на активиране.



Фиг. 6. Мрежи с директна рекурсия

# Мрежи с индиректна рекурсия

Този тип мрежи са мрежи с право разпространение на сигнала, при които се допускат връзки към невроните от предходни слоеве (фиг. 7), включително и връзки към входния слой.



# Фиг.7. Мрежи с индиректна рекурсия

**Мрежи с хоризонтални връзки (странична рекурсия)**

При мрежите от този тип съществуват връзки между невроните от един и същи слой. Тази начин на свързване довежда до това всеки неврон да потиска останалите от същото ниво и да усилва собствения си сигнал. Така на състезателен принцип се определя кой неврон да бъде активиран.

# Класификация на изкуствените невронни мрежи.

* Според **вида на входните данни** изкуствените невронни мрежи са:
* аналогови (на входа се подават реални числа);
* двоични (на входа се подават двоични числа);
* фигуративни (на входа се подават символи).
* Според **начина на обучение**:
* контролирано обучение (известно е изходното пространство на решенията на невронната мрежа);
* обучение без учител (изходното пространство на решенията се формира само въз основа на входни действия; такива мрежи се наричат самоорганизиращи се);
* подсилващо обучение (използване на система за определяне на наказания и награди в резултат на взаимодействието на мрежата с околната среда).

# според начина на настройката на тегловите коефициенти на връзките между невроните:

* мрежи с фиксирани връзки (тегловните коефициенти на невронната мрежа се избират веднага въз основа на условията на проблема);
* мрежи с динамични връзки (тези мрежи коригират своите синаптични връзки по време на процеса на обучение) .

# Според времето на предаване на сигнала:

* синхронни мрежи (времето за предаване за всяка синаптична връзка е или нула, или фиксирана константа);
* асинхронни мрежи (времето за предаване за всяка връзка между елементите е различно, но също постоянно).

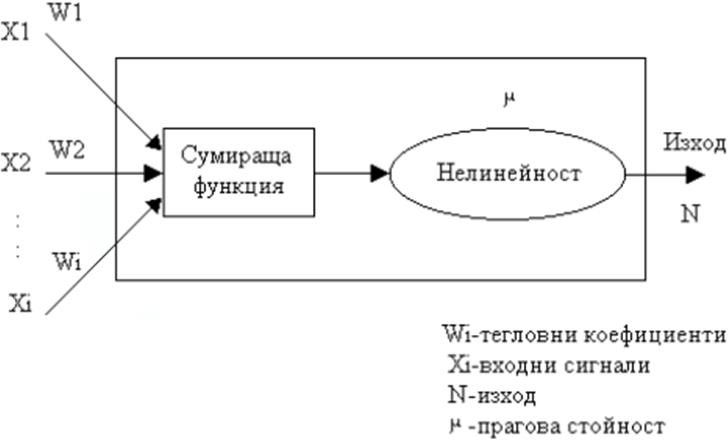
# Според естеството на връзките:

* мрежи за директно разпространение (всички връзки са насочени стриктно от входните неврони към изходните);
* повтарящи се мрежи (сигналът от изходните неврони или невроните на скрития слой се предава частично обратно към входовете на невроните на входния слой) ,
* рекурентна мрежа на **Хопфийлд** (филтрира входните данни, връщайки се в стабилно състояние и по този начин позволява решаване на проблемите с компресирането на данни и изграждането на асоциативна памет);
* двупосочни мрежи (има връзки между слоевете както в посока от входния слой към изходния и в обратната посока).
* Извън тази класификация съществуват мрежи с радиална основа или хипер базисна функционална мрежа, на англ. *Hyper Basis Function network RBF мрежи*), самоорганизиращи се карти (по-специално самоорганизиращата се карта на Кохонен) и мрежи от други класове, които все още не са се оформили напълно.
* **Според топологията** на мрежата невронните мрежи биват еднослойни, двуслойни и многослойни.

# Неврон на McCulloch-Pitt

Невронът на McCulloch-Pitt, показан на фигура 3, е първият изчислителен модел на неврон. Предложен е през 1943 г. от Уорън МуКълок

(невролог) и Уолтър Питс (математик). Чрез него се илюстрира идеята на използването на невронните мрежи за различни изчисления. Моделът се счита за универсален тъй като всяка логическа функция може да бъда изчислена с мрежа от неврони на McCulloch-Pitt. Същевременно всяка крайна последователност от дискретни действия може да бъде симулирана с рекурентни невронни мрежи от такива неврони. Ето защо те представляват основа за създаване на други процесорни елементи чрез използване на различни нелинейни функции.



Фиг. 4. Неврон на McCulloch-Pitt

Литература:

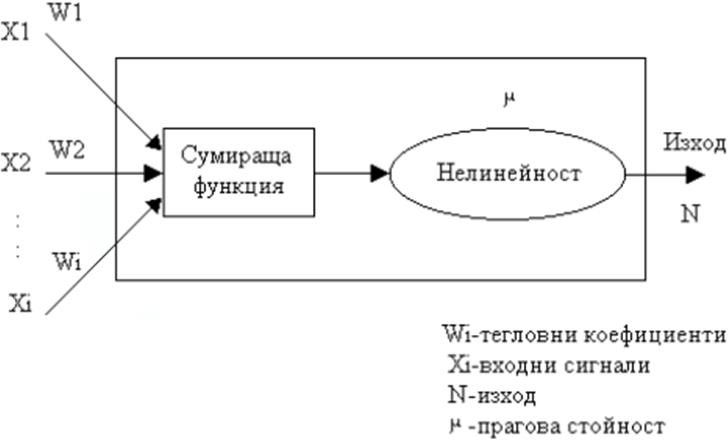
1. David Kriesel, A Brief Introduction to Neural Networks, достъпно на <http://www.dkriesel.com/en/science/neural_networks>, посетено на 12.08.2022 г.
2. Терехов В. А., Ефимов Д. В., Тюкин И. Ю. Нейросетевые системы управления. — М.: Высшая школа, 2002. — 184 с. — ISBN 5-06-004094-1.
3. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика = Neural Computing. Theory and Practice. — М.: Мир, 1992. — 240 с. — ISBN 5- 03-002115-9.
4. Хайкин С. Нейронные сети: полный курс = Neural Networks: A Comprehensive Foundation. 2-е изд. — М.: Вильямс, 2006. — 1104 с. — ISBN 0- 13-273350-1.

# Тема 1/Занятие 7/Аудиторно упражнение

**Обучение на невронни мрежи.**

# Неврон на McCulloch-Pitt

Невронът на McCulloch-Pitt, показан на фигура 3, е първият изчислителен модел на неврон. Предложен е през 1943 г. от Уорън МуКълок (невролог) и Уолтър Питс (математик). Чрез него се илюстрира идеята на използването на невронните мрежи за различни изчисления. Моделът се счита за универсален тъй като всяка логическа функция може да бъда изчислена с мрежа от неврони на McCulloch-Pitt. Същевременно всяка крайна последователност от дискретни действия може да бъде симулирана с рекурентни невронни мрежи от такива неврони. Ето защо те представляват основа за създаване на други процесорни елементи чрез използване на различни нелинейни функции.



Фиг. 3. Неврон на McCulloch-Pitt

Както показва фиг. 3, на входа на неврона постъпват входните сигнали,

X=**{х1, х2, ….., хi},** които както и изходния сигнал **Yса** двоични т.е.

*xi* 0, 1

и *y* 0, 1.

Входовете биват два вида – **възбудителни** и **забранителни**. Теглата,

свързани със възбудителните входове

*x*1, *x*2 , ,

*xk* са отрицателни. Всеки

ненулев входен сигнал на някой от забранителните входове генерира нулев изходен сигнал. Те се обобщават от сумираща функция и постъпват на входа на блок, който реализира нелинейна функция. Нелинейната функция трансформира входа от сумиращата функция в изход, който определя изходния сигнал на неврона. Нелинейната функция има определена прагова стойност (параметърът  на фиг. 3). Когато сумата от произведенията на входовете **Xi** с теглата **Wi** надхвърли този праг, невронът се активира (извежда 1), а в противен случай остава в покой (извежда 0).

 *k* 

 *y*1,

*y*  

*когато wj x j* **

*j*1

*k*

*или xi* 0,

*i**k* 1,...,*m*

  *j j i*

0, *когато w* *x* **  *или i**k* 1,...,*m*, *такова че x* 1



*j*1

(3)

За правилната работа на неврона трябва да е изпълнено условието

*k*

*w*  *w*  ** ,

  *i*  *k* 1,..., *m*

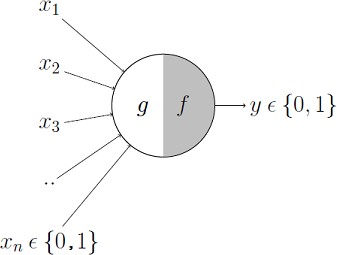
*j i*

(4)

*j*1

При невронът на McCulloch-Pitt теглата са фиксирани, т.е. не се настройват с обучение. [[5](#_bookmark0)]

Неговата работа може да бъде описана със следващите прости примери.

Нека да разглеждаме неврона условно като съставен от две части, както е показано на фиг. 4. Лявата част *g* приема входните сигнали, сумира ги и предава резултата на дясната част *f,* която взема решението.

Фиг. 4. Опростен модел на неврон на McCulloch-Pitt Например**:** Нека да неврона да вземе решение *дали да бъде гледан или*

*не футболен мач по телевизията.*

Всички входни и изходни променливи са булеви, т.е. {0: Ще го гледам, 1: Няма да го гледам}.

Нека:

* ***x\_1*** има значение *Играе най-любимия ми отбор*
* ***x\_2*** има значение *Играе отбор, който харесвам*
* ***x\_3*** има значение *Не съм вкъщи*
* ***x\_4*** има значение *Играе отбор, които много харесвам*

В този пример входа ***x\_3*** е забранителен. Ако *x\_3 е 1 (не съм вкъщи*) изходната функция ще има стойност 0 независимо от стойността на сигнала на останалите входове.

Входовете ***x\_1, x\_2 и x\_4*** са възбудителни, но нито един от тях не може самостоятелно да задейства неврона. Това се случва само, когато сумата им надвиши определена прагова стойност, както е описано в израз (3).

# Реализиране на булеви функции с помощта на неврон на McCulloch- Pitt

М-Р неврона е подходящ да се използва за реализация на булеви функции тъй като входовете и изходите му са булеви.

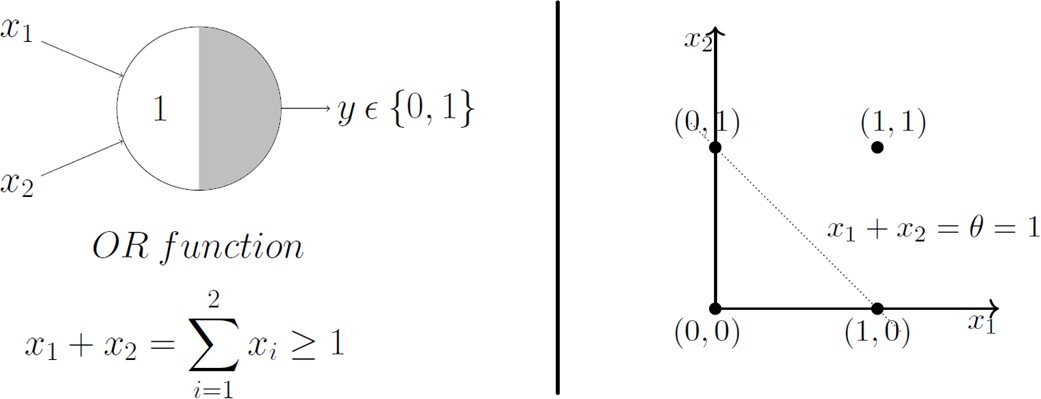
В следващите примери да приемем, че неврона има два входа и в зависимост от това ли сумата от техните стойности е надвишава или не определена прагова стойност, неврона ще генерира или не изходна функция.

# Реализация на логическа функция ИЛИ (OR)

Реализирането на булевата функция ИЛИ от неврона (фигура 5) е равнозначно на генерирането на изходен сигнал ***y = 1*** когато на поне един от входовете има подадена логическа 1 (за справка вж. таблица 1) т.е., ***g(x) ≥ 1***.

Табл. 1. Таблица на истинност на функция ИЛИ за две входни променливи.

|  |  |  |
| --- | --- | --- |
| **x1** | **x2** | **ИЛИ** |
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

Фигура 5 представя четирите възможни комбинации от стойностите на булевите входните променливи **x1** и **x2** като координати на точки в равнина. Уравнението: ***x1 + x2 ≥ 1*** е агрегиращото уравнение на неврона и показва границата на решението. Всички комбинации от стойности на входните променливи, които са координати на точки, лежащи **ПОД линията** ще изведат решение **0**, а тези които лежат **НА** или **НАД** нея ще изведат **1** като резултат.

Фиг. 5 Реализация на логическа функция ИЛИ

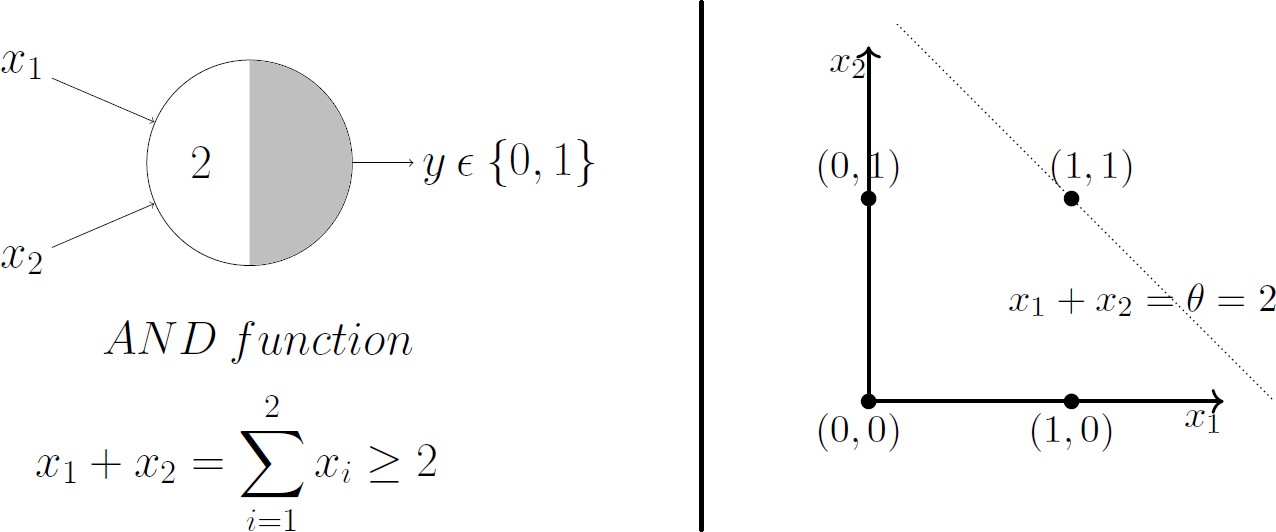
По този начин M-P невронът е обучен на линейна граница на решение. Той разделя входните набори на два класа - положителни и отрицателни. Положителните (които извеждат 1) са тези, които се намират **ВЪРХУ** или **НАД** границата на вземане на решение, а отрицателните (които извеждат 0) са тези, които се намират ПОД границата на вземане на решение.

# Реализация на логическа функция И (AND)

Реализирането на булевата функция И от неврона е равнозначно на генерирането на изходен сигнал ***y = 1*** само когато има подадена логическа 1 и на двата входа едновременно (за справка вж. таблица 2) т.е., ***g(x) = 2***.

Табл. 2. Таблица на истинност на функция И за две входни променливи.

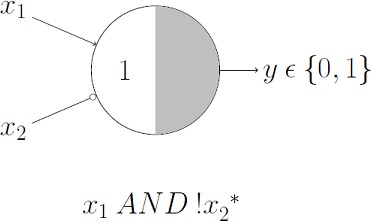
|  |  |  |
| --- | --- | --- |
| **x1** | **x2** | **AND** |
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |



Фиг. 5 Реализация на логическа функция И

В този случай агрегиращото уравнение на неврона, което показва границата на решението е ***x1 + x2 =* 2**. Всички точки лежащи **ВЪРХУ** или **НАД** линията – в случая точката с координати (1,1), извеждат 1.

# Реализация на функция с потискащ вход



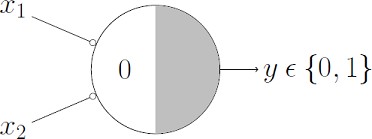
Фиг. 6. Реализация на логическа функция ИЛИ

Входът ***x2*** е потискащ, така че винаги когато ***x2 = 1***, изходът ще бъде 0. В резултат изходния сигнал ще бъде равен на 1 само когато ***x1=1 и x2=0***, (табл. 2). Очевидно е, че праговият параметър трябва да бъде 1.

Таблица 2. Таблица на истинност на логическа функция ??????

|  |  |  |
| --- | --- | --- |
| **x1** | **x2** | **x1 *AND! x2*** |
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

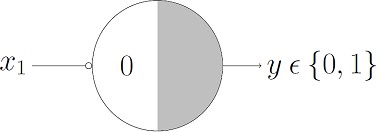
# Реализация на логическа функция ИЛИ - НЕ (NOR)



Фиг. 7. Реализация на логическа функция ИЛИ-НЕ

За да се генерира на изхода сигнал ***y =*** 1е необходимо на всички входове да се подаде 0 и праговия параметър да се определи като 0.

# Реализация на логическа функция НЕ (NOТ)



Фиг. 8. Реализация на логическа функция НЕ

Този неврон извежда 0 когато на входа му е подадена 1 и извежда 1, когато на входа му е подадена 1. За да се постигне такова „поведение“ на неврона входа му трябва да се определи като инхибиращ вход и праговия параметър да се зададе като 0.

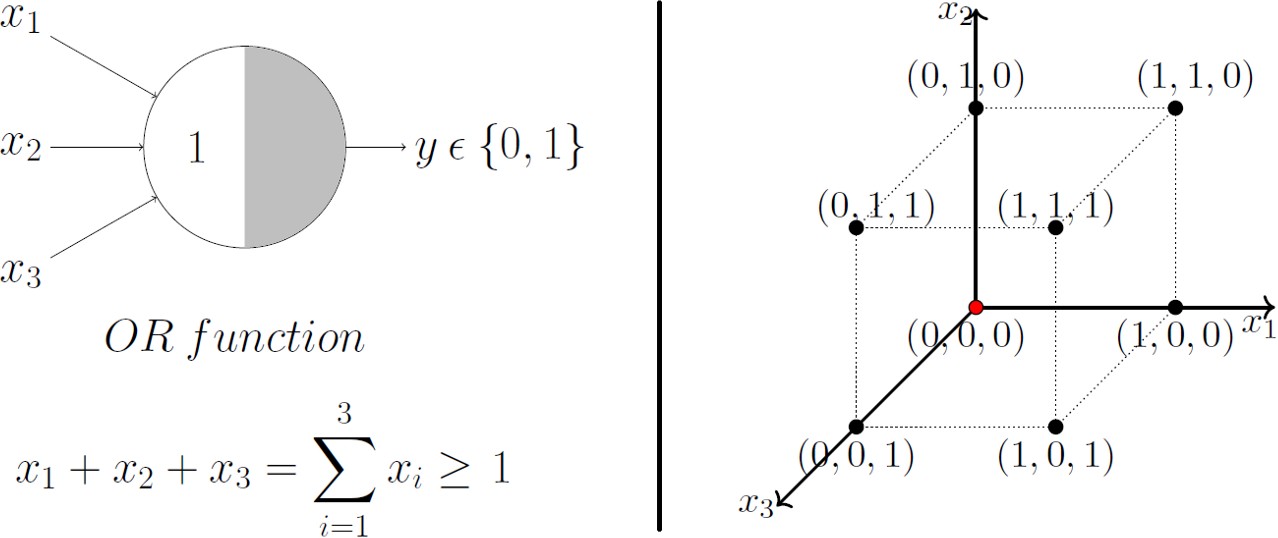
**Реализация на логическа функция ИЛИ (OR)** с три входни променливи

Реализирането на булевата функция ИЛИ с три входни променливи от неврона (фигура 9) е равнозначно на генерирането на изходен сигнал ***y = 1*** когато на поне един от входовете има подадена логическа 1 т.е., ***g(x) ≥ 1***.

Табл. 3. Таблица на истинност на функция ИЛИ за три входни променливи.

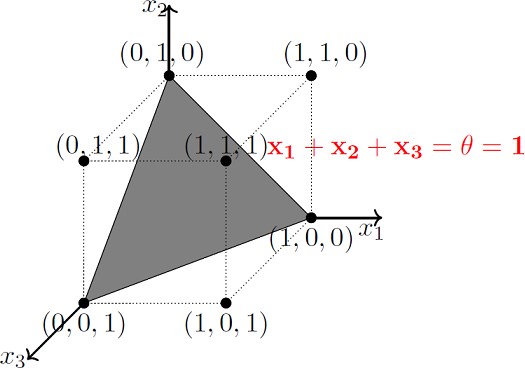
|  |  |  |  |
| --- | --- | --- | --- |
| **x1** | **x2** | **x3** | **ИЛИ** |
| 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 |



Фиг. 9. Реализация на логическа функция ИЛИ на три променливи

Както показва таблица 3, възможни са 8 комбинации от стойности на входните променливи. На фигура 9 е показана границата на решение в триизмерното пространство.



Фиг. 10. Граница на решение на логическа функция ИЛИ на три променливи Всички състояния на входните променливи, определящи точки в пространството, които лежат **ВЪРХУ** или **НАД** равнината, определена от уравнението: ***x1 + x2 + x3 =* 1** (положителното полупространство), ще доведат до изход 1, и всички които лежат **ПОД** тази равнина (отрицателно

полупространство) ще доведе до изход 0.

Анализът на представените примери води до извода, че правилното определяне на праговия параметър на M-P неврона позволява реализацията на линейно разделими булеви функции.

M-P неврона притежава редица недостатъци:

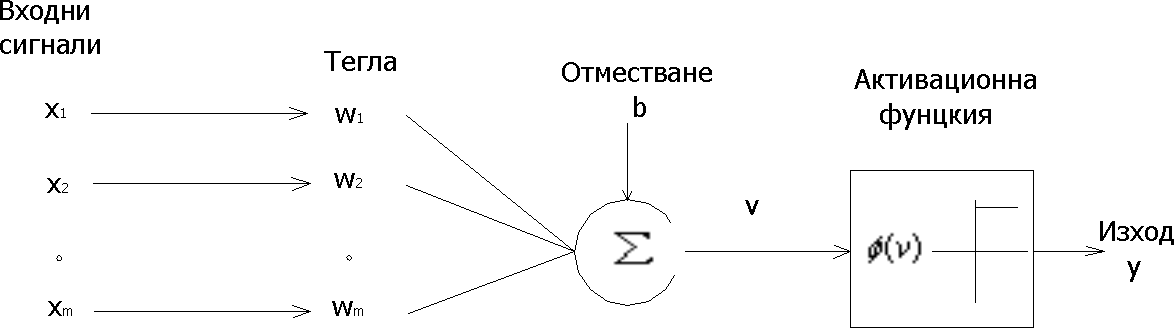
* Не се разпознават входни променливи, които не са бинарни;
* Определянето на стойността на праговия параметър не може да стане автоматично;
* Не е възможно да се определят различни приоритети на различните входове;
* Не е възможно да се реализират функции, които не са линейно

отделими, например функцията XOR.

# Перцептрон

Описаните ограничения са преодолени в предложеният през 1957 г. от Frank Rosenblatt обобщен изчислителен модел на неврон при който теглата и праговете могат да бъдат научени с течение на времето.

Перцептронът, представен е на фиг. 4, представлява еднослойна невронна мрежа с един елемент, който често е негов сумиращ процесор. Може да работи както с двоични така и с аналогови входни величини. Използва нелинейна активационна функция (прагова, сигмоидна и др.)



Фиг. 10.

При него изходът е явна функция на входа. Това е типично за всички прави мрежи. Изходът на суматора се изчислява като

**  *wj xj*  *b*  *wj xj* ,

*m m*

(5)

*j*1 *j* 0

където

*w*0  *b* , а

*x*0 е сигнал с постоянна стойност 1.

Изходният сигнал y на перцептрона се определя с формулата :

*y*  **(*v*) 1,

0,

*ako ako*

** **

** **

(6)

където ** е праг на активация на перцептрона, и ако стойността му не е

спомената специално се счита, че е нула **  0 .

При перцептрона се използва обучение с учител. При този тип обучение критерият за обучението се задава с комплект примерни входно-изходни последователности за желаната работа на мрежата от типа

където

{ *p*1, *d*1},{ *p*2 , *d*2}, ,{ *pQ* , *dQ* },

*pQ* е образ, подаван на входа на мрежата, а *dQ*

е съответната правилна

(желана) стойност на изхода на мрежата при подаване на

*pQ* . С прилагане на

входните елементи към мрежата се сравняват получената стойност на изхода на мрежата с желаната стойност, зададена от съответното d. След това се стартира алгоритъм за обучение, посредством който се променят (настройват) теглата и отместванията на мрежата така, че получаваната стойност на изходите на мрежата да се доближават до желаната.

Алгоритъмът, прилаган при обучението на перцептрона е следният :

ако имаме d=0 и

ако имаме d=1 и

*f* (*wT x*)  1, то тогава

*f* (*wT x*)  0 , то тогава

*w*\_ *new*

*w*\_ *new*

 *w*\_ *old*

 *w*\_ *old*

* *x* ,
* *x* ,

ако имаме d= *f* (*wT x*) , то тогава

*w*\_ *new*

 *w*\_ *old* ,

където с

*w*\_ *new*

се означава новата, а с

*w*\_ *old*

старата стойност на тегловия

коефициент от преходната интерация. С x се означава съответната стойност на

*pQ* , подадена на входа, а с d- съответното

*dQ* . Функцията

*f* (*wT x*) , имаща

релеен характер, приема в случая стойности 0 или 1, т.е.

*y*  *f* (*v*)  1, *v***

0, *v***

Ако бъде въведена грешка

*e*  *d*  *f* (*wT x*),

правилото за обучение може

да бъде представено в следния компактен вид :

*w*\_ *new*  *w*\_ *old*  *ex* .

Горното правило може да се обобщи за един слой от k на брой

перцептрони, използващи едни и същи входни сигнали

*x*  [*x*1 *x*2

...

*x* ]*T* . В

този случай формулите за настройка на вектора с теглата *wi*

*m*

и отместване

*bi* ,

свързани с i-тия перцептрон, са равни на :

*wi* \_ *new*  *wi* \_ *old*  *ei x*,

*bi*  *bi*  *ei* ,

където

*e*  *d*  *f* (*wT x*  *b* ).

*i i*. *i i*

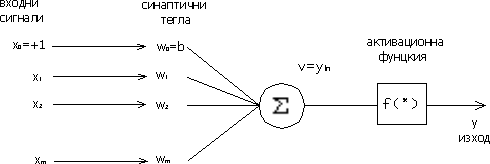
# Мрежите на Hopfield

Мрежите на Hopfield обикновенно се състоят от едно ниво, което приема на вход 1 или 0. Тегловните коефициенти са цели числа. Всички неврони са напълно взаимносвързани и всеки от тях се явява като входен, изходен и междинен. Изходът на всеки неврон внася сигнал на входа му.

Направени са подробни изследвания и някои разширения на мрежата на Hopfield. Установено е, че тя може да се използва за оптимизационни задачи (например задачата за търговския пътник), реконструкция на образ от зашумени данни и други. При оптимизационните задачи се тъси конфигурация на мрежата, която да минимизира т.нар. енергийна функция. **Енергийната функция е функция на състоянието, която винаги намалява(или остава непроменена)при еволюция на мрежата съгласно промяната в активността на невроните.** Тази функция има локален минимум1 при стабилно състояние на мрежата. Приложима е, ако тегловните коефиценти в мрежата са симетрични,т.е. силите на връзките са симетрични - *wij=wji.*

# Адаптивен линеен елемент - ADAptive LINear Element (ADALINE)

Адаптивният линеен елемент - ADAptive LINear Element (ADALINE) е предложен през 1962 г. от Bernard Widrow. Той представлява единичен неврон (фиг.3.2) с линейна активационна функция и настройващи се тегла. Алгоритъмът на Windrow-Hoff за настройка на теглата се базира на метода на най-малките квадрати и е пример за използване на обучение при настройката. Той е в основа за различни видове адаптивна филтрация използвана при обработка на сигнали.



Фиг. 3.2

За обучението на мрежите от тип ADALINЕ е необходимо да се разпалага с обучаваща извадка, представляваща набори от входни сигнали и желан (еталонен) изход на мрежата за всяки от тях. При подаване на всеки *k*- ти комплект входни сигнали, се определя изходът от невронната мрежа и

грешката *ek*

между него и еталонния изход при зададената комбинация на

входа . Целта на обучението е да бъдат настроени теглата *w*  [*w*1 *w*2 ... *w* ]

*T*

*m*

*p*

така, че да бъде минимизирана интегралната грешка  *ei* за цялата обучаваща

*i* 1

съвкупност. В различни варианти на обучаващите алгоритми промяната на теглата се прави след всеки компонент от обучаващата серия или след едно цялостно нейно изчерпване (т.е. след всяка епоха). Във всички случаи

промяната на теглата е пропорционална на съответния входен сигнал, грешката и параметър ** , наречен скорост на обучение.

Мрежите Адалайн принадлежат към общия клас алгоритми, наречени адаптивни линейни филтри. Те намират приложение в много области като например за проектиране на изравняващи филтри при високоскоростни модеми, за адаптивни ехокомпенсатори при телефонни разговори на големи разстояния и сателитни комуникации, както и за предсказване на сигнали. Намират приложение и в медицината например за подтискане на шума от биене на сърцето на майката при ембрионална електрокардиография. [3]

Литература:

1. David Kriesel, A Brief Introduction to Neural Networks, достъпно на <http://www.dkriesel.com/en/science/neural_networks>, посетено на 12.08.2022 г.
2. Терехов В. А., Ефимов Д. В., Тюкин И. Ю. Нейросетевые системы управления. — М.: Высшая школа, 2002. — 184 с. — ISBN 5-06-004094-1.
3. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика = Neural Computing. Theory and Practice. — М.: Мир, 1992. — 240 с. — ISBN 5- 03-002115-9.
4. Хайкин С. Нейронные сети: полный курс = Neural Networks: A Comprehensive Foundation. 2-е изд. — М.: Вильямс, 2006. — 1104 с. — ISBN 0- 13-273350-1.
5. McCulloch-Pitts Neuron — Mankind’s First Mathematical Model Of A Biological Neuron | by Akshay L Chandra | Towards Data Science

# Тема 1/Занятие 8/Лекция

**Обучение на невронни мрежи.**

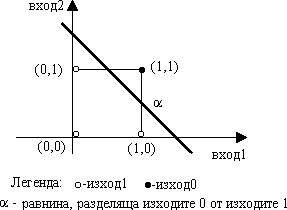
# Съдържание на процеса на обучение на изкуствена невронна мрежа.

Обучението на невронната мрежа е процес, при който тя се настройва да реагира по определен начин на определени входни въздействия. По този начин се постига получаване на достоверни резултати от работата ѝ.

Най-общо обучението на невронната мрежа се изразява в промяна на нейните компоненти, така че тя да реагира по желания начин на определени входни въздействия. Теоретично една невронна мрежа може да се учи чрез:

* създаване на нови връзки между невроните;
* изтриване на съществуващи връзки;
* промяна на тегловите коефициенти на връзките;
* промяна на праговите стойности на невроните;
* промяна на една или повече от следните три функции (активиращата, функцията на разпространение и изходната функция),
* създаване на нови или изтриване на съществуващи неврони (и на връзките помежду им).

Не всички задачи са решими чрез обучението на една невронна мрежа. Подходящи са тези от тях, които са **линейно отделими**. Това означава, че за задачата може да се намери равнина в пространството, която да разделя множествата от различни изходни резултати, получени при отделните комбинации входни сигнали.



Фиг. 4. Линейна отделимост на логическата функция конюнкция

Ако такава равнина не може да бъде построена, задачата не може да бъде решена чрез невронна мрежа. Пример за линейно отделима задача е логическата функция конюнкция (фиг.4). Входните сигнали и резултатът, които трябва да се получат са посочени в таблица.1.

таблица.1. Таблица на истинност на логическа функция конюнкция

|  |  |  |
| --- | --- | --- |
| **Вход 1** | **Вход 2** | **изход** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

От фигурата се вижда, че може да се построи равнина, отделяща изход 1 от изходите 0. Такава задача може да бъде решена от обикновен неврон.

Способността за обучението е основна особеност на невронните мрежи. То се реализира по подходящи алгоритми чрез реализация на многократна промяна на тегловните коефициенти на връзките между невроните.

Комбинацията от входни сигнали за които се търси правилната реакция от мрежата се нарича **обучаващо множество.**

Въздействията на входа, които невронната мрежа получава по-късно могат да се различават от сигналите с които тя е обучена, но получената реакция трябва да бъде правилна.

Обучаващия процес започва с въвеждане на първия пример на входното ниво на мрежата. Произвежда се изход, който обикновено е далеч от целта, защото е базиран на текущото състояние на връзките. Поради произволно избраните първоначално тегла, резултатът се различава от желания. Сравнява се получения и желания изход. В зависимост от размера на грешката теглата на всички връзки се настройват, така че отклонението да се намали. За тази цел се използват подходящи алгоритми (обучаващи правила). Процесът се повтаря със следващия пример и продължава докато се минимизира грешката или се постигне поставената цел. Грешката се намалява чрез постепенна настройка на теглата в невронната мрежа. Когато обучаващия процес се преустанови, мрежата е готова за използване. Постига се стабилно състояние**.** Обикновено теглата не се променят за времето за което мрежата се използва по задачата, за която е обучена.

От съществено значение при обучението на изкуствени невронни мрежи е да се определи кога да се спре процеса на обучение. Тъй като способността да се правят обобщения е от фундаментално значение за способността на мрежите, прекомерното обучение, прекомерно нагаждане или прекаляването с процеса на трениране е сериозен проблем.

Друга важна особеност в определянето и осигуряването на необходимото количество данните за обучение. Необходимо е да се използва максимално голяма база данни, с която системата може да бъде обучена. Добре е да бъдат използвани достатъчно голям набор от данни, поради факта че някои шаблони на поведение не могат да бъдат открити в малки бази данни.

# Обучение с учител

Контролираното обучение изисква много шаблони, които да служат като образци. Всеки шаблон принадлежащ от комплекта за обучение съдържа входни стойности и съответните целеви изходни. Мрежата изчислява целевата стойност на базата на зададените входни данни чрез минимизиране на грешката на изхода си до достигане на целевия изход. Това се реализира посредством непрекъснато коригиране на теглата на връзките си чрез един повтарящ се процес на обучение, който на практика представлява трениране.

Най-често срещаният алгоритъм за обучение на мрежата е **алгоритъм с обратно разпространение на грешката**.

Обратното разпространение на грешката по време на обучението на изкуствена невронна мрежа е процеса по разпространение на грешка в системата в посока, обратна на придвижването на входящата информация, т.е. от изходния към входния слой на мрежата. Изходящия слой на мрежата е единствения, който има целева стойност, с която да се сравнява получения резултат и докато грешката се разпространява по обратният път към входящия слой чрез възлите, тежестите на връзките между отделните неврони се променят. Обучението продължава докато грешката в теглата е достатъчно незначителна, за да бъде приета. Видът на активиращата функция, използвана

във възлите на невронната мрежа може да окаже влияние при определяне на вида на данните, с които се обучава мрежата.

# Обучение без учител

Използват се и **самообучаващи се невронни мрежи**. Този тип мрежи сами откриват най-характерните черти на категориите, които трябва да различават и изработват собствено представяне на входните сигнали.

и не се нуждае от конкретни стойности на изхода за провеждане на обучението. За всеки един от шаблоните подавани на входния слой на мрежата се допуска, че принадлежи към отделен клас параметри. По този начин, процесът на обучение се свежда до това да се остави мрежата да разкрие тези класове. Този метод не е толкова популярен колкото контролираното обучение и не е подходящ да се използва в този труд. Поради тази причина няма да бъде разглеждан по-нататък.

Два популярни метода за самообучение на невронни мрежи са: **обучение, основано на главните компоненти на данните и състезателно обучение**.

# Усъвършенстващо обучение

Усъвършенстващото обучение е хибриден метод за обучение. При него на мрежата не се задават целеви резултати, но в същото време мрежата се информира, ако изчислените изходни данни се движат в правилната посока или не.

# Алгоритъм за обучение на еднослоен перцептрон

Перцептронът трябва да реши проблема с класификацията чрез двоични входни сигнали. Наборът от входни сигнали ще бъде означен с n-мерния вектор x. Всички елементи на вектора са булеви променливи (променливи, които приемат стойностите "True" или "False"). Понякога обаче е полезно да се оперира с числени стойности. Ще приемем, че стойността "false" съответства на числовата стойност 0, а стойността "True" съответства на 1.

Перцептронът е устройство, което изчислява следната система от функции:



( 1)

където wI са теглата на персептрона,

 е прагът, xI са стойностите на входните сигнали, скобите [] означават прехода от булеви (логически) стойности към числени стойности според правилата, посочени по-горе.

Обучението на Perceptron се състои в коригиране на коефициентите на тегло. Нека има набор от двойки вектори  ,  , , наречен набор за обучение.

Невронната мрежа ще бъде обучена на даден тренировъчен набор, ако, когато всеки вектор, който се подава към мрежовите входове, съответният изходен вектор се получава всеки път на изходите.

Методът на обучение, предложен от Ф. Розенблат, се състои в итеративно коригиране на матрицата на теглото, което последователно намалява грешката в изходните вектори. Алгоритъмът включва няколко стъпки:

**Стъпка 0** Първоначалните стойности на теглата на всички неврони W(t=0) се приемат за случайни

**Стъпка 1** Входното изображение се представя на мрежата, в резултат на което се формира изходното изображение.

**Стъпка 2** Изчислява се векторът на грешката  направена от мрежата на изхода. Другата идея е, че промяната на тегловния вектор в областта на малките грешки трябва да бъде пропорционална на изходната грешка и равна на нула, ако грешката е нула.

**Стъпка 3** Тегловият вектор се модифицира по следната

формула:  където е

скоростта на обучение.

**Стъпка 4 Стъпки 1-3** се повтарят за всички вектори за обучение. Един цикъл на последователно представяне на цялата проба се нарича епоха.

Обучението завършва след няколко епохи: а) когато итерациите се сближат, т.е. тегловният вектор спира да се променя, или б) когато общата абсолютна грешка, сумирана за всички вектори, стане по-малка от някаква малка стойност.

Посочените в алгоритъма стъпки могат да бъдат интерпретирани по следния начин:

подаваме на входа на персептрона вектор x, за който верният отговор вече е известен. Ако изходът на перцептрона съвпада с правилния отговор, тогава не е необходимо действие. В случай на грешка е необходимо да обучите перцептрона да реши правилно този пример. Грешките могат да бъдат два вида.

Първият тип грешка: изходът на персептрона е 0, а верният отговор е 1. За да може перцептронът да даде правилен отговор, е необходимо сумата от дясната страна на (1) да стане по-голяма. Тъй като променливите приемат стойности 0 или 1, може да се постигне увеличение на сумата чрез увеличаване на теглата wi. Въпреки това, няма смисъл да се увеличават теглата, когато променливите xi са равни на нула. Следователно теглата wi трябва да бъдат увеличени за тези променливи xi, които са равни на 1.

Първо правило. Ако изходът на перцептрона е 0, а правилният отговор е 1, тогава е необходимо да се увеличат теглата на връзките между едновременно активни неврони. В този случай изходният перцептрон се счита за активен.

Вторият тип грешка: изходът на перцептрона е 1, а верният отговор е нула. За да научите правилното решение на този пример, сумата от дясната страна на (1) трябва да бъде намалена. Следователно е необходимо да се намалят теглата на връзките w\_i за тези променливи, които са равни на 1 (тъй като няма смисъл да се намаляват теглата на връзките за променливи x\_i, равни на нула). Също така е необходимо тази процедура да се извърши за всички активни неврони от предишните слоеве. В резултат на това получаваме второто правило.

Второ правило. Ако изходът на перцептрона е единица, а правилният отговор е нула, тогава е необходимо да се намалят теглата на връзките между едновременно активни неврони.

По този начин процедурата за обучение се свежда до последователно изброяване на всички примери от набора за обучение, като се използват правилата за обучение за погрешно решени примери. Ако след следващия цикъл на представяне на всички примери се окаже, че всички са решени правилно, тогава процедурата за обучение е завършена.

Теорема за конвергенцията на персептрон. Ако има параметър вектор w, за който персептронът решава правилно всички примери от набора за обучение, тогава, когато персептронът се обучава съгласно описания по-горе алгоритъм, решението ще бъде намерено в краен брой стъпки.

Теорема за зацикляне на персептрон. Ако няма параметърен вектор w, за който перцептронът решава правилно всички примери от обучителната извадка, тогава, когато перцептронът се обучава съгласно това правило, векторът на теглото ще започне да се повтаря след краен брой стъпки.

По този начин тези теореми гласят, че след стартиране на процедурата за обучение на персептрон, след крайно време получаваме или обучен персептрон, или отговорът, че този перцептрон не може да научи задачата.

Литература:

1. David Kriesel, A Brief Introduction to Neural Networks, достъпно на <http://www.dkriesel.com/en/science/neural_networks>, посетено на 12.08.2022 г.
2. Терехов В. А., Ефимов Д. В., Тюкин И. Ю. Нейросетевые системы управления. — М.: Высшая школа, 2002. — 184 с. — ISBN 5-06-004094-1.
3. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика = Neural Computing. Theory and Practice. — М.: Мир, 1992. — 240 с. — ISBN 5- 03-002115-9.
4. Хайкин С. Нейронные сети: полный курс = Neural Networks: A Comprehensive Foundation. 2-е изд. — М.: Вильямс, 2006. — 1104 с. — ISBN 0- 13-273350-1.
5. Гульнара Яхъяева, Лекция 3. Персептроны. Обучение персептрон, достъпно на https://intuit.ru/studies/courses/88/88/print\_lecture/20531

Тема 1/Занятие 9/Упражнение

# Обучение с учител. Задачи за апроксимация на многомерни функции и класификация на образи. Обучение с право и обратно разпространение

**на грешката. Обучение без учител.**

# Същност на процеса на обучение на изкуствена невронна мрежа.

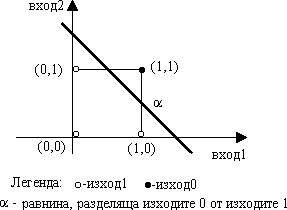
Способността за обучението е основна особеност на невронните мрежи. То се реализира по подходящи алгоритми чрез реализация на многократна промяна на тегловните коефициенти на връзките между невроните.

Обучаващия процес започва с въвеждане на първия пример на входното ниво на мрежата. Произвежда се изход, който обикновено е далеч от целта, защото е базиран на текущото състояние на връзките. Поради произволно избраните първоначално тегла, резултатът се различава от желания. Сравнява се получения и желания изход. В зависимост от размера на грешката теглата на всички връзки се настройват, така че отклонението да се намали. За тази цел се използват подходящи алгоритми (обучаващи правила). Процесът се повтаря със следващия пример и продължава докато се минимизира грешката или се постигне поставената цел. Грешката се намалява чрез постепенна настройка на теглата в невронната мрежа. Когато обучаващия процес се преустанови, мрежата е готова за използване. Постига се стабилно състояние**.** Обикновено

теглата не се променят за времето за което мрежата се използва по задачата, за която е обучена.

# Линейна отделимост на задача.

**Линейна отделимост** означава, че за задачата може да се намери равнина в пространството, която да разделя множествата от различни изходни резултати, получени при отделните комбинации входни сигнали.



Фиг. 1. Линейна отделимост на логическата функция конюнкция

Ако такава равнина не може да бъде построена, задачата не може да бъде решена чрез невронна мрежа. Пример за линейно отделима задача е логическата функция конюнкция

# Видове обучение на изкуствени невронни мрежи

* 1. **Същност и особености на обучението с учител**

Контролираното обучение изисква много шаблони, които да служат като образци. Всеки шаблон принадлежащ от комплекта за обучение съдържа входни стойности и съответните целеви изходни. Мрежата изчислява целевата стойност на базата на зададените входни данни чрез минимизиране на грешката на изхода си до достигане на целевия изход. Това се реализира посредством непрекъснато коригиране на теглата на връзките си чрез един повтарящ се процес на обучение, който на практика представлява трениране.

* 1. **Същност и особености на обучението без учител Самообучаващите се невронни мрежи** сами откриват най-характерните черти на категориите, които трябва да различават и изработват собствено представяне на входните сигнали. и не се нуждае от конкретни стойности на изхода за провеждане на обучението. За всеки един от шаблоните подавани на входния слой на мрежата се допуска, че принадлежи към отделен клас параметри. По този начин, процесът на обучение се свежда до това да се остави мрежата да разкрие тези класове. Този метод не е толкова популярен колкото контролираното обучение и не е подходящ да се използва в този труд. Поради тази причина няма да бъде разглеждан по-нататък.

# Задачи за апроксимация на многомерни функции и класификация на образи.

Разпознаването се състои в предварително получаване на пълен вектор от характеристики за всеки отделен разпознаваем обект, избран в изображението, и едва след това в определяне на кой от стандартите съответства този вектор. Стандартите най-често се изграждат като статистически или като геометрични обекти. В първия случай обучението може да се състои, например, в получаване на матрица от честоти за появата на всяка характеристика във всеки клас обекти и разпознаване при определяне на вероятностите, че векторът на характеристиките принадлежи към всеки от стандартите. При геометричния подход резултатът от обучението най-често е разделянето на пространството на характеристиките на региони, съответстващи на различни класове разпознаваеми обекти, а разпознаването се състои в определяне на кой от тези региони попада входният вектор от признаци, съответстващ на разпознаваемия обект. Трудности при присвояването на вектора на входните признаци на която и да е област могат да възникнат в случай на пресичане на области, а също и ако областите, съответстващи на отделни разпознаваеми класове, не са изпъкнали и са разположени в пространството на признаците по такъв начин, че разпознаваемият клас не е отделени от другите класове с една хиперравнина. Тези проблеми най-често се решават евристично, например чрез изчисляване и сравняване на разстояния (не непременно евклидови) в пространството на характеристиките от тестовия обект до центровете на тежестта на подмножествата на обучителната извадка, съответстващи на различни класове. Възможни са и по-радикални мерки, например промяна на азбуката на характеристиките или групиране на обучителната извадка, или и двете едновременно.

Структурният подход съответства на референтни описания, изградени по отношение на структурни части на обекти и пространствени отношения между тях. Структурните елементи се разграничават, като правило, върху контура или върху "скелета" на обекта. Най-често структурното описание може да бъде представено чрез графика, която включва структурни елементи и връзки между тях. По време на разпознаването се изгражда структурно описание на входния обект. Това описание се сравнява с всички структурни стандарти, например се открива изоморфизъм на графики.

Растерните и структурните методи понякога се свеждат до индикативен подход, разглеждайки в първия случай точките на изображението като знаци, а във втория - структурни елементи и връзки между тях. Веднага отбелязваме, че има много важна фундаментална разлика между тези методи. Растерният метод има свойството цялост. Метод struct може да има свойство за цялост. Методът на атрибута няма свойството цялост.

Какво е почтеност и каква роля играе във възприятието?

Класическото разпознаване на образи обикновено се организира като последователен процес, който се разгръща "отдолу нагоре" (от изображение към разбиране) при липса на контрол на възприятието от горните концептуални нива. Етапът на разпознаване се предхожда от етапа на получаване на априорно описание на входното изображение. Операциите за извличане на елементи от това описание, например характеристики или структурни елементи, се извършват локално върху изображението, части от изображението получават независима интерпретация, тоест няма холистично възприятие, което в общия случай може да доведе до грешки - фрагмент от изображение, разглеждан изолирано, често може да бъде интерпретиран напълно различно в зависимост от хипотезата на възприятието, т.е. от това какъв вид интегрален обект трябва да се види.

Традиционните подходи са фокусирани върху разпознаването (класифицирането) на обекти, разглеждани поотделно. Етапът на действително разпознаване трябва да бъде предшестван от етапа на сегментиране (разделяне) на изображението на части, съответстващи на изображения на отделни разпознаваеми обекти. Методите за априорно сегментиране обикновено използват специфични свойства на входното изображение. Няма общо решение на проблема с предварителната сегментация. Освен в най-простите случаи, критерият за разделяне не може да бъде формулиран от гледна точка на локалните свойства на самото изображение, тоест преди то да бъде разпознато.

Изкуствените невронни мрежи също могат да апроксимират нелинейни разделителни повърхности с равнини и като резултат да комбинират несвързани области на пространството на характеристиките. За да изпълняват тази задача те трябва да са многослойни.

Най-важният аспект на възприятието е предвиждането, основано на йерархичен модел на света и многостепенен процес на възприятие. В позната среда и познати ситуации възприятието протича на нивата на обобщения (общо-частно) и разширения (цяло-част) и се състои в потвърждаване на предвиждането на тези нива. Привличането към нивото на детайлно възприятие се случва само до степента на поведенческа необходимост или в случай на несъответствие между предвиждане и реален вход.

Във физиологията е известен принцип на акцептора, според който в нервната система винаги (непрекъснато) във всяко действие има модел на очакваната обратна аферентация, идваща от резултата от действието. Несъответствието между модела и реалната обратна аферентация предизвиква ориентировъчно-изследователска реакция.

Най-важният момент от възприятието е формирането на хипотеза за съдържанието на изображението. Хипотезата произтича от взаимодействието на процеса "отгоре надолу", който се разгръща на базата на модела на средата, модела на текущата ситуация и текущия резултат от възприятието, и процеса "отдолу нагоре", базиран на преки грубо, преди всичко показателно, възприятие.

След това се потвърждава хипотезата или се изяснява възприятието. На този етап си взаимодействат и двата процеса - операции върху информация от

модела и операции върху изображение. В същото време, в рамките на настоящата хипотеза, използвайки модела на средата и информацията за контекста на възприятието, се извършва целенасочено търсене, включващо сегментиране на изображението на части, търсени в съответствие с хипотезата и съвместна интерпретация от разпределените части.

Най-важният аспект на възприятието е неговата цялост: резултатите от локалните операции се интерпретират само съвместно в процеса на интерпретиране на интегрални фрагменти и цялото изображение като цяло. Холистичният модел, използван при възприятието, трябва да бъде структурен, пълен и видим. Последното означава, че трябва да е възможно мислено да си представим обект според неговия модел.

Моделът на проблемната среда, използван при възприятието, трябва да включва йерархия от интегрални представяния. По отношение на задачата за автоматично възприемане на изображения може да се каже следното. В допълнение към съвместната интерпретация на елементите на изображението като част от разпознаваеми обекти, принципът на целостта на възприятието предполага и интерпретацията на самите разпознаваеми обекти като част от по-големи интегрални образувания - структури, които показват тези взаимоотношения от външна задача, в която участват разпознаваеми обекти . Тези взаимоотношения формират външния контекст на разпознаване. Използването на външен контекст на разпознаване позволява не само да се интерпретират правилно онези обекти, чиито образи позволяват двусмислена интерпретация при отделното им възприемане, но и да се повиши надеждността на разпознаването на всички обекти, участващи в определена

семантична конструкция, поради тяхната целенасоченост и съвместност. интерпретация.

В съответствие с принципа на целостта на възприятието, в общия случай можем да говорим за йерархия от нива на интерпретация на елементите на разпознаваем образ: от интерпретация като част от самите разпознаваеми обекти до интерпретация като част от най-големите семантични структури, представени в това изображение.

Такава организация на процеса на разпознаване в система за машинно зрение е необходима, ако искаме да получим наистина ефективно решение на сложни практически проблеми. Естествено, машинното визуално възприятие все още не може да се конкурира с човешкото зрително възприятие. Основната причина е, че не знаем как да изградим и използваме пълен машинен семантичен модел на перцептивната среда. Въпреки това, за подобряване на ефективността на системите за машинно зрение и по-специално на системите за машинно четене, гореспоменатите принципи на двупосочност (от изображение към модел и от модел към изображение), предвиждане, формиране на хипотеза, цялостност, целенасоченост и максимално използване на информация за проблемната среда трябва да се прилага до известна степен.не само възможно, но и необходимо.

Тези принципи, доколкото е възможно, са внедрени в софтуерния пакет Graphite, в програмите FineReader-manuscript и FormReader за разпознаване на ръкописни знаци. Същите принципи са частично внедрени в програмата FineReader, предназначена за разпознаване на печатен текст. Пакетът Graphite е разработен в NICEVT през 80-те години, програмите FineReader и FormReader - в ABBYY.

Литература:

1. David Kriesel, A Brief Introduction to Neural Networks, достъпно на <http://www.dkriesel.com/en/science/neural_networks>, посетено на 12.08.2022 г.
2. Терехов В. А., Ефимов Д. В., Тюкин И. Ю. Нейросетевые системы управления. — М.: Высшая школа, 2002. — 184 с. — ISBN 5-06-004094-1.
3. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика = Neural Computing. Theory and Practice. — М.: Мир, 1992. — 240 с. — ISBN 5- 03-002115-9.
4. Хайкин С. Нейронные сети: полный курс = Neural Networks: A Comprehensive Foundation. 2-е изд. — М.: Вильямс, 2006. — 1104 с. — ISBN 0- 13-273350-1.
5. Гульнара Яхъяева, Лекция 3. Персептроны. Обучение персептрон, достъпно на https://intuit.ru/studies/courses/88/88/print\_lecture/20531

*Тема 1/ Занятие 10. Лекция*

# Приложение на изкуствените невронни мрежи

Увод: В недалечен исторически план създаването на големи невронни мрежи е изисквало солиден обем изчислителни ресурси. Например Google създава мрежа с около 1000 сървъра - приблизително 16 000 процесорни ядра със 1,7 милиарда връзки между невроните, за да създаде мрежа, която се научи да разпознава котки само в поредица от видеоклипове в YouTube. По-късно Google създадава мрежа с 11,2 милиарда връзки между невроните.

През 2015 г Digital Reasoning създава мрежа със 160 милиарда връзки, която е обявена за най-голямата в света. Тя има възможности да чете и да разбира човешка реч, като взема предвид контекста с точност 85,8%.

През 2017 г. Movidius, създава и разпространява сред

потребителите Neural Compute Stick. Той има размери, сравними с обикновено флаш устройство, докато вътре

има мощна невронна мрежа с функция за дълбоко машинно обучение. Това е мини компютър с размерите на преносима USB памет, който следи трафика на информация между две мрежи.

# Характеристики на изкуствените невронни мрежи, определящи областите на тяхното приложение

Основните **предимства** на невронните мрежи пред традиционните изчислителни методи са:

# Паралелна обработка на информацията.

За разлика от последователното изпълнение на инструкции при машината на фон-Нойман (конвенционалния компютър), моделите на невронните мрежи използват паралелизмът на многопроцесорната система.

# Способност за решаване на проблеми в условия на несигурност и недостиг на информация.

Благодарение на способността за учене, невронната мрежа има възможност да решава проблеми с неизвестни модели и зависимости между входни и изходни данни, а от там и работа с непълни данни.

# Устойчивост на шум във входните данни.

Невронната мрежа може самостоятелно да идентифицира параметри, които не са информативни за анализа, и да ги филтрира, поради което няма нужда от предварителен анализ на входните данни.

# Гъвкавост на структурата на невронните мрежи.

Компонентите на неврокомпютрите - невроните и връзките между тях - могат да се комбинират по различни начини. Благодарение на това един неврокомпютър може да се използва за решаване на различни проблеми, често несвързани помежду си. Висока производителност. Входните данни се обработват от много неврони едновременно, благодарение на което невронните мрежи решават проблеми по-бързо от повечето други алгоритми.

# Адаптиране към промените в околната среда.

Невронните мрежи, които се учат от данни, са в състояние да се адаптират към променящата се среда (например към промените в пазарната ситуация, ако задачата на невронната мрежа е да прогнозира колебанията на цените на фондовата борса). Ако е необходимо да се реши някакъв проблем в нестационарна среда, тогава могат да бъдат създадени невронни мрежи, които се обучават в реално време. Колкото по-висок е адаптивният капацитет на системата, толкова по-стабилна ще бъде нейната работа в нестационарна среда.

# Устойчивост на грешки.

Невронната мрежа реагира на неблагоприятна промяна в условията само с лек спад в производителността. Тази функция се обяснява с разпределения характер на съхранението на информация в невронната мрежа, така че само сериозно увреждане на структурата може значително да повлияе на работата на невронната мрежа.

* **възможност за обобщаване** – извличат общото от множество данни. Невронните мрежи имат редица сериозни **недостатъци**.
* Не са способни да решават задачи, за които са необходими точни и недвусмислени отговори.
* Не могат да решат проблеми, които изискват последователно изпълнение на няколко стъпки. Те са в състояние да решат проблема само "на един дъх". Следователно една невронна мрежа не може например да докаже математическа теорема.
* Не са способни да решават изчислителни проблеми. Например не са способни да получат всички решения на математическо уравнение за различни параметри.
* Трудоемкост и продължителност на обучението. За да може невронната мрежа да решава правилно задачите, е необходимо тя да се обучава на десетки милиони набори от входни данни. Разработени са различни технологии за ускорено обучение, съвременните видеокарти позволяват да се обучават невронни мрежи стотици пъти по-бързо. Съществуват и готови, предварително обучени невронни мрежи, предназначени за разпознаване на изображения, въз основа на които могат да бъдат създавани приложения без продължително обучение. Препоръчва се, когато недостигат данни за създаване на обучаващо множество или набирането им е с много висока цена, да се предпочете използването на експертни системи.
* Необходимост от интеграцията на голям брой процесорни елементи в структурата на невронната мрежа, което предполага затруднена техническа реализация при решаване на сложни задачи.
* трудности при генериране на обяснение на решението и проблеми при изграждане на естествено езиковия интерфейс.

Невронните мрежи са подходящи за решаване на задачи при които е трудно да се изрази с формални средства логическия процес на вземане на решение от експертите, но може да се подготви репрезентативно множество от данни. Най-общо това са задачи, свързани с класификация и разпознаване на образи, диагностика, персонална индентификация, анализ на естествени езици и оптимизационни задачи.

# Приложение на изкуствените невронни мрежи според вида на решаваните задачи.

Основните **приложения** на изкуствените невронни мрежи са в следните направления:

# класификация и клъстеризация на данни

Под клъстеризация се разбира групиране (автоматична класификация) на данните, състояща се в избор на категории, разделяне на набор от ситуации на няколко групи. Другите термини, които се срещат в литературата са клъстерен анализ, неконтролирана класификация и автоматична класификация.

Такива процес е определянето дали набор от анализирани данни може да се причисли към една или друга група, на базата на сходство, така че сходството с една група да е по-голямо отколкото с други групи.

Например идентифициране на наличието на няколко варианта на хода на заболяването,идентифициране на наличието на няколко групи потребители, автоматично категоризиране на текстове или новини по теми. За решаване на тази задачи се подхожда по следния начин: идентифицират няколко групи ситуации и се идентифицират типични представители за всяка група. Изследва се степента на различие между идентифицираните групи и степента на сходство на екземплярите в рамките на определена група.

Клъстерният анализ подобрява точността на решаване на проблемите с прогнозирането или класификацията, тъй като в рамките на всеки клъстер проблемът с прогнозирането или класификацията може да бъде решен отделно

* чрез намаляване на областите на компетентност на прогнозните модели.
  + **разпознаване на обекти и изображения (**в това число и разпознаване на символи – например оптични като ръкописен или печатен текст;

Разпознаване на изображения (например ръкопис, пръстови отпечатъци, разпознаване на лица), намиране и разпознаване на различни обекти в изображение теоретични може да бъде сведена до присвояване на изображението на един или друг референтен клас.

* + сегментиране и обработка на реч
  + филтриране на шума (ехото) от при пренос на телефонни сигнали на големи разстояния;
  + прогнозиране;

# за визуализация на данни

Двуизмерната и триизмерната визуализация на многомерни данни позволява анализиране на зависимости между тях като: откриване на вътрешната им структура, откриване на качествено различни групи или подмножества от данни, оценка и интерпретация на резултатите от автоматична класификация на данните за подпомагане на процеса на вземане на решение.

Пример за невронна мрежа, изпълнваща такава задача е самоорганизиращата се карта на Кохонен. Предназначена е да проектира многоизмерни данни в по-нискоизмерно пространство да ги визуализира, така че да се установи наличието или отсъствието на клъстерна структура и да се определи броят на клъстерите, законите за съвместно разпределение на характеристиките, зависимостите между променливите.

# Изисквания потребителя на невронна мрежа и възможности програмиране

С термина потребител“ се означава не крайния потребител на обучена и настроена за работа система, а експерт, който разработва модела и обучава невронни мрежа.

Едно от основните изисквания е правилната формулировка на проблема, която се заключава в коректна семантичната формулировка на проблема, правилен избор на математическия метод за решение и неговите настройки - избор на адекватна невронна мрежа структура, алгоритъм на обучение, критерий за качество на решаване на проблема и др. Автоматичните схеми за избор на оптимални настройки за невронни мрежови методи, налични в софтуера, ориентиран към крайния потребител, не винаги могат да намерят правилните решения за повече или по-малко сложни задачи. Например, съответните процедури в невронните мрежи на Statistica не са добри при решаването на проблема с прогнозирането на времеви редове, тъй като те не използват редуциране на редове до стационарна форма.

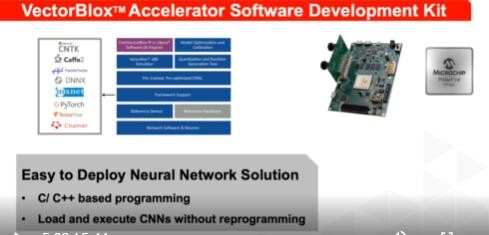
Най-често използваните програми използващи методите на невронните мрежи са т. нар. симулатори на невронни мрежи - те работят на обикновени компютри. Обикновено те включват набор от основни операции за създаване, обучение и манипулиране на невронни мрежи, първоначални данни, свойства на невронни мрежи и невронни решения, и автоматизирани процедури за извършване на най-рутинните вериги от действия, например за определяне на оптималната мрежови настройки и алгоритъм за обучение.

Универсалните съвременни невропрограми имат способността да генерират описание на обучена невронна мрежа на определен език за

програмиране, така че полученият софтуерен модул да може да бъде вмъкнат в програмата на потребителя. Също така потребителската програма може да използва стандартни инструменти за междупрограмна комуникация (DDE, OLE, COM в средата на Windows), за да осигури достъп до невронни мрежи в универсални невронни програми. Невронната мрежа остава да работи в средата на програмата за невронна мрежа, която я е създала , а потребителската програма само издава необходимите команди за данни и получава резултатите. Специфични изчислителни архитектури (например сигнални процесори) обикновено се прехвърлят към описание на обучена невронна мрежа, генерирана от невросимулаторна програма на език за програмиране (например на C). За FPGA / PLM / FPGA-кристали има и CAD системи, в които номенклатурата на невроните може да бъде създадена и описана като набор от модули и след това от тези модули, чрез избиране и плъзгане с мишката, да се сглоби желаната топология- архитектура на невронната мрежа, след това компилирайте проекта и " флашнете кристала. Понастоящем е възможно да се създават неврокомпютри, базирани на видеокарти (графични процесори),

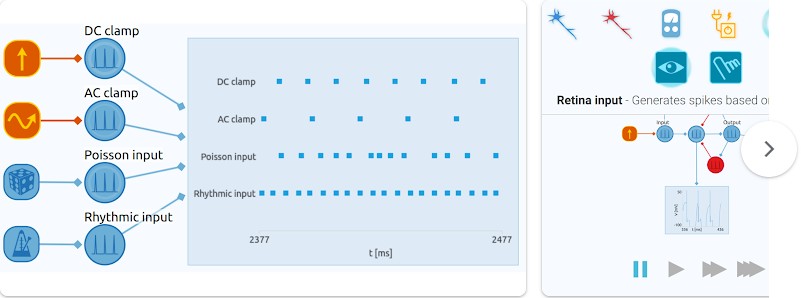
използвайки технологиите CUDA и OpenCL.

VectorBlox Accelerator [[3](#_bookmark2)] е софтуерна развойна среда, разработена от Microchip чрез която програмируемите логически матрици (FPGA) от фамилията PolarFire могат да бъдат използвани за създаване на интелигентни приложения за невронни мрежи с ниско енергопотребление.



Тя предоставя на разработчиците готов набор от софтуерни инструменти и не прави засъджителни задълбочените познания в областта на FPGA програмирането, като проектирането на мрежи с изкуствен интелект може да се осъществи чрез код на C/C++. Комплектът от гъвкави софтуерни инструменти може да изпълнява модели в TensorFlow и платформата за невронни мрежи с отворен код ONNX, която предлага най-широката фреймуърк съвместимост (Caffe2, MXNet, PyTorch, MATLAB). За разлика от алтернативни FPGA решения, VectorBlox Accelerator се поддържа от операционните системи Linux и Windows, а също така включва и bit-accurate симулатор, който предоставя на потребителя възможност да валидира хардуера в софтуерна среда. IP-адресът на невронната мрежа, включен в комплекта, също поддържа възможността за зареждане на различни мрежови модели по време на изпълнение.

Neuronify [[4](#_bookmark3)] е образователен инструмент, който онагледява поведението на неврони и невронни мрежи. Можете да се използва за симулация и експериментиране на невронни мрежи с различни параметри.



Широкия набор от софтуерни инструменти не изключва възможността за ръчно програмиране (като се използва готово невроядро (софтуерна библиотека) и да добавите необходимите механизми за достъп до собствените данни, интерфейс за манипулиране на данни и / или настройки за методи за обработка и средства за показване / визуализиране на резултати.

# Oбласти на приложение на изкуствените невронни мрежи [[2](#_bookmark1)]

* + **изготвяне на прогнози за времето;**
    - **във финансовия сектор,** за автоматизирано решаване на следните практически задачи:
      * за прогнозиране на пазара на акции,
      * определяне на пазарни показатели, борсови цени,
      * класификация на акции,
      * прогнозиране на времеви редове (обменни курсове, цени на суровини, търсене, обеми на продажби);
      * осъществяване на автоматична търговия (търговия на валутна, фондова или стокова борса);
      * оценка на риска от неизпълнение на кредити, прогнозиране на фалит, оценка на недвижими имоти;
      * определяне на рейтинга на компании;
      * оптимизиране на стокови и парични потоци;
      * четене и разпознаване на чекове и документи;
      * обезпечаване на сигурността на транзакции с банкови карти.

# в образованието;

* + - **в криптографията**
    - **в медицината и здравеопазването** за автоматизирано решаване на следните практически задачи:
      * диагностициране на заболявания;
      * обработка на медицински изображения;
      * изчистване на показанията на инструментите от шум;
      * наблюдение на състоянието на пациента;
      * прогнозиране на резултатите от използването на различни методи на лечение,;
      * анализиране на ефективността на лечението.
    - **Авионика** за автоматизирано решаване на следните практически задачи:
      * обучение на автопилоти и безпилотни летателни апарати;
      * разпознаване на радарни сигнали;
      * адаптивно пилотиране на тежко повредени самолети;
    - **Комуникации** за автоматизирано решаване на следните практически задачи:
      * компресиране на видео информация;
      * бързо кодиране-декодиране;
      * оптимизация на клетъчни мрежи и схеми за маршрутизиране на пакети;
    - **Интернет** за автоматизирано решаване на следните практически задачи:
* асоциативно извличане на информация;
* филтриране и блокиране на нежелана поща;
* автоматично категоризиране на съобщения от новинарски канали;
* целенасочена реклама и маркетинг;
  + - **Автоматизация на производството** за автоматизирано решаване на следните практически задачи:
* за контрол и оптимизиране на режимите на производствени процеси;
* контрол на качеството на продукта;
* предотвратяване на аварийни ситуации.
  + - **Роботика** за автоматизирано решаване на следните практически задачи:
* разпознаване на обстановка, обекти и препятствия пред робота;
* определяне на маршрута на движение;
* управление на манипулатори;
* поддържане на баланс.
  + - **Политология и социологически изследвания** за автоматизирано решаване на следните практически задачи:
* прогнозиране на изборни резултати;
* анализ на анкети;
* прогнозиране на динамиката на рейтинга;
* идентифициране на значими фактори;
* групиране на електората;
* изследване и визуализация на социалната динамика на населението.
  + - **Сигурност** за автоматизирано решаване на следните практически задачи:
* за изграждане на системи за сигурност: лицево разпознаване; идентифициране на лице чрез пръстов отпечатък, глас, подпис или лице; разпознаване на регистрационни номера;
* за мониторинг на информационни пакети и информационни потоци в компютърна мрежа за откриване на проникване;
* анализ на данни от видеокамери и различни сензори;
* анализ на аерокосмически изображения (например за откриване на горски пожари).
  + - **Въвеждане и обработка на информация** за автоматизирано решаване на следните практически задачи:
* разпознаване на ръкописни текстове, сканирани пощенски, платежни, финансови и счетоводни документи;
* разпознаване на говорни команди, говорно въвеждане на текст в компютър.
  + - **Геоложки проучвания** за автоматизирано решаване на следните практически задачи:
* анализ на сеизмични данни;
* асоциативни методи за търсене на полезни изкопаеми;
* оценка на ресурсите на находищата.
  + - **Компютърни и настолни игри** за автоматизирано решаване на следните практически задачи:
* създаване на невро-играчи в дама и шах (оценки, потвърдени от игра с хора - на ниво майстори и международни майстори);
* победа в Го срещу шампионите на Европа и света; средно по-добре от човек, преминаването на почти петдесет стари класически игри с Atari (всички видове Pongs, Pac-Mans).

# Етапи на подготовка на изкуствени невронни мрежи за решаване на приложни задачи.

Етап 1: Подготовка на данни за обучение на мрежата Етап 2. Подготовка и нормализиране на данни

Етап 3. Избор на мрежова топология;

Етап 4. Експериментален избор на мрежови характеристики; Етап 5. Експериментален подбор на тренировъчни параметри; Етап 6 реално обучение;

Етап 7. Проверка на адекватността на обучението;

Етап 8. Настройка на параметрите, окончателно обучение; Етап 9. Мрежова вербализация[18] за по-нататъшна употреба.

Изборът на данни за мрежово обучение и тяхната обработка е най- трудната стъпка в решаването на проблема. Наборът от данни за обучение трябва да отговаря на критериите за представителност, достоверност и съгласуваност.

Всеки запис във файла с обучаващи данни се нарича вектор за обучение. Обучаващият вектор съдържа една стойност за всеки вход на мрежата и в зависимост от вида на обучението (контролирано или неконтролирано), една стойност за всеки мрежов изход. Преди да бъдат подадени към входовете на мрежата е необходимо данните за обучение да бъдат преобразувани по определен начини:

Нормализацията се извършва, когато към различни входове се подават данни с различни размерности.

Квантуването се извършва върху непрекъснати величини, за които ограничен набор от дискретни стойности. Например, квантуването се използва за задаване на честотите на звука сигнали при разпознаване на реч;

Филтрирането се извършва за "шумни" данни. Освен това представянето както на входните, така и на изходните данни играе важна роля.

Да предположим, че мрежата е обучена да разпознава букви в изображения и има един цифров изход − номер на буквата в азбуката. В този случай мрежата ще получи фалшиво впечатление, че буквите с цифрите 1 и 2 са по-сходни от буквите 1 и 3, което по принцип не е вярно. Да се избягвайте тази ситуация, използвайте мрежова топология с голям брой изходи, когато всеки изход има свой собствен смисъл. Колкото повече изходи в мрежата, толкова по-голямо е разстоянието между класовете и толкова е по-трудно да бъдат объркани.

Google: Технологията AlphaGo побеждава световния шампион в играта Go; през март 2016 г. корпорацията продаде на търг 29 картини, нарисувани от невронни мрежи.

Microsoft: проектът CaptionBot разпознава изображения в снимки и автоматично генерира надписи за тях, проектът WhatDog определя породата на куче от снимка, услугата HowOld определя възрастта на човек на снимка.

Компанията Yandex: Приложението Auto.ru разпознава автомобили в снимки; невронната мрежа записа музикален албум; проектът LikeMo.net се учи да рисува в стила на известни художници.

Такива развлекателни услуги се създават главно за демонстриране на способностите на определена невронна мрежа и провеждане на нейното обучение. Но резултатите са впечатляващи.

* Подборът му е от съществено значение и изисква време. Фирми, които обработват голям брой транзакции, осигуряват улеснено набиране на обучаващи примери.

GMDH Shell, професионален софтуер за невронни мрежи, решава задачи за прогнозиране на времеви серии и извличане на данни, като изгражда изкуствени невронни мрежи и ги прилага към входните данни

За разработването на тази лекция са използвани материали от източниците посочени е списъка с по-долу.

ЛИТЕРАТУРА

1. Intel® Neural Compute Stick 2 (Intel® NCS2)
2. Невронни мрежи, методи за анализ на данни от изследването до разработката и внедряването, достъпен на <http://www.neuropro.ru/>, посетен на 14.08.2022 г.
3. [https://www.microchip.com/en-us/products/fpgas-and-plds/fpga-and-soc-](https://www.microchip.com/en-us/products/fpgas-and-plds/fpga-and-soc-design-tools/vectorblox) [design-tools/vectorblox](https://www.microchip.com/en-us/products/fpgas-and-plds/fpga-and-soc-design-tools/vectorblox)
4. https://play.google.com/store/apps/details?id=net.ovilab.neuronify&hl=bg& gl=US

Тема 1/Занятие 11/Семинар

# Семинарно занятие по тема 1

По предложения списък от въпроси обучаемите се подготвят предварително и представят по време на зянятието доклад в обем до четири страници и по тяхна преценка презентация, в обем до десет слайда.

По всеки от въпросите се провежда дискусия в рамките на тридесет минути.

# Основни теоретични концепции в областта на изкуствения интелект.

* 1. **Морални, етични и правни аспекти на изкуствения интелект.**

# Архитектура и функционални особености на изкуствените невронни мрежи.

* 1. **Същност на процеса на обучение на изкуствена невронна мрежа.**

# Решавани задачи и области на приложение на изкуствените невронни мрежи.

Литература:

1. David Kriesel, A Brief Introduction to Neural Networks, достъпно на <http://www.dkriesel.com/en/science/neural_networks>, посетено на 12.08.2022 г.
2. Терехов В. А., Ефимов Д. В., Тюкин И. Ю. Нейросетевые системы управления. — М.: Высшая школа, 2002. — 184 с. — ISBN 5-06-004094-1.
3. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика = Neural Computing. Theory and Practice. — М.: Мир, 1992. — 240 с. — ISBN 5- 03-002115-9.
4. Хайкин С. Нейронные сети: полный курс = Neural Networks: A Comprehensive Foundation. 2-е изд. — М.: Вильямс, 2006. — 1104 с. — ISBN 0- 13-273350-1.
5. Гульнара Яхъяева, Лекция 3. Персептроны. Обучение персептрон, достъпно на https://intuit.ru/studies/courses/88/88/print\_lecture/20531

# Тема 2/Занятие 1/Лекция

**Базови функции на програмната среда MATLAB за изграждане и обучение на изкуствени невронни мрежи.**

# Предназначение и общо описание на програмната среда MATLAB.

MathWorks предоставя няколко продукта, които са особено подходящи за видове задачи, които могат да бъдат изпълнявани с NeuralNetwork Toolbox, описани в таблица 1.

MATLAB е програмна среда за автоматизиране на числени пресмятания и самостоятелен програмен език от четвърто поколение. Както подсказва името, системата MATLAB (от MATrix LABoratory“ - на български: „матрична лаборатория“) е специално създадена за извършване на матрични изчисления: решаване на системи линейни уравнения, разлагане на матрици и т.н.

Освен числен анализ софтуерната среда позволява изчертаване графики на функции, представяне на данни, програмна реализация на алгоритми, разработка на човеко-машинни интерфейси и интерфейси с други програмни продукти, написани на различни програмни езици.

MatLab се състои от ядро, което представлява съвкупност от програми с различно предназначение и допълнителни компоненти (ToolBox), разширяващи възможностите на ядрото с широк спектър от възможности в областта на анализа на данни, изследването на системи за регулиране, идентификацията, оптимизацията и др.

|  |  |  |
| --- | --- | --- |
| Продукт | Описание | |
| Control System Toolbox | Инструмент за моделиране, анализиране  проектиране | и |
| Data Acquisition Toolbox | MATLAB функции за директен достъп измерени данни от MATLAB | до |
| Database Toolbox | Инструмент за свързване и взаимодействие с повечето ODBC/JDBC бази данни от MATLAB | |
| Financial Toolbox | Функции на MATLAB за количествено финансово  моделиране и аналитично прототипиране | |
| Fuzzy Logic Toolbox | Инструмент за подпомагане на овладяването на техниките на размита логика и тяхното приложение при практически проблеми с  управлението | |
| Image Processing Toolbox | Пълен пакет за цифрова обработка изображения и  инструменти за анализ за MATLAB | на |
| Optimization Toolbox | Инструмент за обща и мащабна оптимизация на нелинейни задачи, както и за линейно  програмиране, квадратично програмиране, нелинейни най-малки квадрати | |
| Signal Processing Toolbox | Инструмент за разработване на алгоритъм, сигнал и линеен системен анализ и данни от времеви  редове моделиране | |
| Simulink | Интерактивна, графична среда за моделиране, симулиране и създаване на прототипи  динамични системи | |
| Spline Toolbox | Инструмент за конструиране и използване на  части полиномиални функции | |
| Stateflow | Инструмент за графично моделиране и симулация  на сложна контролна логика | |
| Statistics Toolbox | Инструмент за анализ на исторически данни, моделиране системи, разработване на  статистически алгоритми и статистика на обучението и преподаването | |

|  |  |
| --- | --- |
| System Identification Toolbox | Инструмент за изграждане на точни, опростени модели на сложни системи от шумни времеви  серии от данни |

Основния начин за представяне на данните в Matlab е матрицата. Това дава възможност да се решават задачи, свързани със сложни технически изчисления, изискващи матрично представяне на данните.

В Matlab е вграден специално разработеният собствен програмен език от високо ниво. Той с ориентиран към създаване на технически разчети и инженерно проектиране.

Matlab се състои от следните компоненти:

* развойна среда Matlab;
* програмен език Matlab;
* графична среда с възможности за интерактивен интерфейс;• библиотеки с готови функции-команди (toolboxes);•

програмен интерфейс с възможност за компилация на разработените потребителски програми на други програмни езици, например C++.

Simulink е съпътстваща на Matlab програма. Сама по себе си тя представлява графична среда за симулационно моделиране на системи и процеси. В Симулинк която библиотеките се състоят не от текстови файлове, а от графични блокове, които се явяват симулационни модели на съответни реални физически елементи, обекти и дори системи.

По подразбиране изчисленията се извършват при стандарт двойна точност (double precision), при което за всяка въведена числена стойност (литерал) или променлива се отделят по 64 бита (8 байта) в режим на плаваща запетая, а за комплексни величини – по 64 бита съответно за реалната и имагинерната компоненти (общо 16 байта). Този факт, наред с използваните модерни числени методи, позволява извършването на сложни математически изчисления в много широк числов диапазон при наличието на минимална грешка от закръгления на междинните резултати.

# Дефиниране и обучаване на невронна мрежа с използване на вградени функции от Neural Network ToolboxTM.

За да започнем, трябва да декларираме обект вид мрежа от избраната функция, която съдържа променливи и методи за извършване на процес на оптимизация.

Функцията очаква два незадължителни аргумента, представляващи броя на скрити единици (и след това на скритите слоеве) и алгоритъмът за обратно разпространение, който трябва да се използва по време на фазата на обучение. Броят на скритите единици трябва да бъде предоставен като едно цяло число, изразяващ размера на скрития слой, или като вектор на цял ред, чиито елементи показват размера на съответните скрити слоеве. Командата: ***nn = patternne t ( 3 )***

# Създаване на перцептрон . Функция newp

Перцептрон може да бъде създаден с функцията newp

# net = newp(PR, S)

където входни аргументи са:

PR е R-на-2 матрица на минимални и максимални стойности за R вход елементи.

S е броят на невроните.

Обикновено функцията **hardlim** се използва в перцептрони, така че е по подразбиране. Кодът по-долу създава **пецептронна мрежа** с един вход от един елемент вектор и един неврон. Диапазонът за единичния елемент на единичния вход

# векторът е [0 2]. net = newp([0 2],1);

Можем да видим каква мрежа е създадена, като изпълним следния код

# inputweights = net.inputweights{1,1}

което дава: входни тегла = закъснения: 0 initFcn: 'initzero' научи: 1 learnFcn: 'learnp'

# learnParam: [] размер: [1 1]

потребителски данни: [1x1 структура]

# weightFcn: 'dotprod'

Мрежовият вход към трансферната функция на hardlim е dotprod, който генерира произведението на входния вектор и матрицата на теглото и добавя отклонението за изчисляване на мрежовият вход.

Функцията за инициализация по подразбиране, **initzero**, се използва за задаване на начални стойности на теглата до нула.

# Симулиране. Функция (sim)

Да предположим, че вземем персептрон с един входен вектор от два елемента. Дефинираме мрежата с

net = newp([-2 2;-2 +2],1);

Както отбелязахме по-горе, това ни дава нулеви тегла и отклонения, така че ако искаме конкретно набор, различен от нули, трябва да ги създадем. Можем да зададем двете тегла и едното отместване към -1, 1 и 1, както бяха във фигурата на границата на решението със следния код.

# net.IW{1,1}= [-1 1];

**net.b{1} = [1];**

За да сме сигурни, че тези параметри са зададени правилно, ние ги проверяваме с

# net.IW{1,1}

**ans =**

# -1 1

**net.b{1} ans =**

Сега нека видим дали мрежата отговаря на два сигнала, по един от всяка страна на перцептронна граница.

# p1 = [1;1];

**a1 = sim(net,p1) a1 =**

# 1

**и за**

# p2 = [1;-1]

**a2 = sim(net,p2) a2 =**

# 0

Перцептронът класифицира правилно двата входа.

Имайте предвид, че можем да представим двата входа последователно и да получим изходите също последователно.

# p3 = {[1;1] [1;-1]};

**a3 = sim(net,p3) a3 =**

# [1] [0]

.

# Инициализация (init)

Можете да използвате функцията init, за да нулирате мрежовите тегла и отклонения към техните оригинални стойности. Да предположим например, че започвате с мрежата

# net = newp([-2 2;-2 +2],1);

**За проверка на теглото му с**

wts = net.IW{1,1}

което дава, както се очаква, wts =

0 0

По същия начин можете да проверите дали отклонението е 0 с bias **= net.b{1}**

което дава bias =

# 0

Сега задайте теглата на стойностите 3 и 4 и стойността отклонението на

5 с

# net.IW{1,1} = [3,4];

**net.b{1} = 5;**

Проверете отново теглата и отклонението, както е показано по-горе, за да проверите дали има промяна. Тя е:

wts = 3 4

bias = 5

Сега използвайте init, за да нулирате теглата и отклонението към първоначалните им стойности.

# net = init(net);

**Можем да проверим, както е показано по-горе, за да потвърдим**

# това.

**wts = 0 0**

# bias = 0

Можем да променим начина, по който се инициализира перцептрон с **init**. Например, можем да предефинираме входните тегла на мрежата и bias initFcns като рандове и след това приложете init, както е показано по-долу.

# net.inputweights{1,1}.initFcn = 'rands'; net.biases{1}.initFcn = 'rands';

**net = init(net);**

Сега проверете теглата и отклонението.

# wts =

**0,2309 0,5839**

# bias =

**-0,1106**

Литература:

1. Demuth, H., M., Beale, Neural Network Toolbox For Use with MATLAB®
2. David Kriesel, A Brief Introduction to Neural Networks, достъпно на <http://www.dkriesel.com/en/science/neural_networks>, посетено на 12.08.2022 г.
3. Терехов В. А., Ефимов Д. В., Тюкин И. Ю. Нейросетевые системы управления. — М.: Высшая школа, 2002. — 184 с. — ISBN 5-06-004094-1.
4. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика = Neural Computing. Theory and Practice. — М.: Мир, 1992. — 240 с. — ISBN 5- 03-002115-9.
5. Хайкин С. Нейронные сети: полный курс = Neural Networks: A Comprehensive Foundation. 2-е изд. — М.: Вильямс, 2006. — 1104 с. — ISBN 0- 13-273350-1.
6. Гульнара Яхъяева, Лекция 3. Персептроны. Обучение персептрон, достъпно на https://intuit.ru/studies/courses/88/88/print\_lecture/20531

Тема 2/Занятие 2/Упражнение

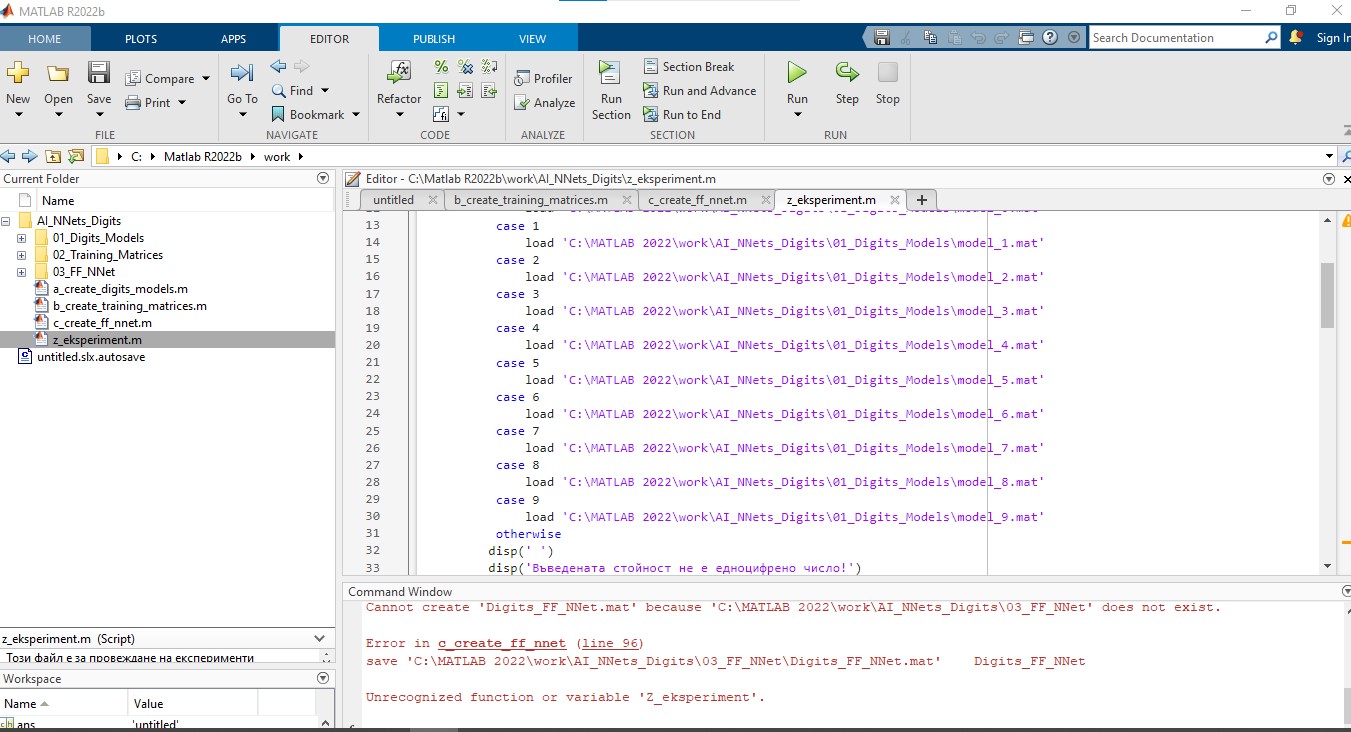
# Изграждане и обучение на изкуствени невронни мрежи в среда на MATLAB.

MATLAB е програмна среда за автоматизиране на числени пресмятания и самостоятелен програмен език от четвърто поколение. Както подсказва името, системата MATLAB (от MATrix LABoratory“ - на български: „матрична лаборатория“) е специално създадена за извършване на матрични изчисления: решаване на системи линейни уравнения, разлагане на матрици и т.н.

Освен числен анализ софтуерната среда позволява изчертаване графики на функции, представяне на данни, програмна реализация на алгоритми, разработка на човеко-машинни интерфейси и интерфейси с други програмни продукти, написани на различни програмни езици.

# Дефиниране и обучаване на невронна мрежа с използване на вградени функции от Neural Network ToolboxTM.

Общ изглед на работния екран



% Създаваме двуслойна FeedForward Neural Network (по стар стил)

Digits\_FF\_NNet = newff( minmax(M), [ S1 S2 ], { 'logsig' 'logsig' }, 'traingdx' );

%

% NET = NEWFF creates a new network NET with a dialog box.

% Obsoleted in R2010b NNET 7.0.0.

% Last used in R2010a NNET 6.0.4.

% The recommended new function is "feedforwardnet".

%

% NET = newff( PR, [S1 S2...SN ], {TF1 TF2...TFN}, BTF, BLF, PF)

% where the parameters are:

% PR - Rx2 matrix of min and max values for R input elements.

% Si - Size of i-th layer ( where i from 1 of N ).

% TFi - Transfer function of i-th layer, default = 'tansig'

% BTF - Backprop network training function, default = 'trainlm'

% BLF - Backprop weight/bias learning function, default = 'learngdm'

% PF - Performance function, default = 'mse'.

% Possible training functions are:

% 'trainlm' - Levenberg-Marquardt (Default)

% 'trainbr' - Bayesian Regularization

% 'trainbfg' - BFGS Quasi-Newton

% 'trainrp' - Resilient Backpropagation

% 'trainscg' - Scaled Conjugate Gradient

% 'traincgb' - Conjugate Gradient with Powell/Beale Restarts

% 'traincgf' - Fletcher-Powell Conjugate Gradient

% 'traincgp' - Polak-Ribiere Conjugate Gradient

% 'trainoss' - One Step Secant

% 'traingdx' - Variable Learning Rate Gradient Descent

% 'traingdm' - Gradient Descent with Momentum

% 'traingd' - Gradient Descent

%

%

% To create a comon shallow NNet we can use:

% NNet = network( NumInputs, NumLayers, BiasConnect, ...

% InputConnect, LayerConnect, OutputConnect )

% where parameters are:

% NumInputs - Number of inputs, 0

% NumLayers - Number of layers, 0

% BiasConnect - NumLayers-by-1 Boolean vector, zeros

% InputConnect - NumLayers-by-NumInputs Boolean matrix, zeros

% LayerConnect - NumLayers-by-NumLayers Boolean matrix, zeros

% OutputConnect - 1-by-NumLayers Boolean vector, zeros

%

% Example how to create a One-input, Two-layer, FeedForward network:

% net = network( 1, 2, [1; 0], [1; 0], [0 0; 1 0], [0 1] )

%

% Here only Layer 1 has a bias.

% An input weight connects to layer 1 from input 1.

% A layer weight connects to layer 2 from layer 1.

% Layer 2 is a network output and has a target.

%

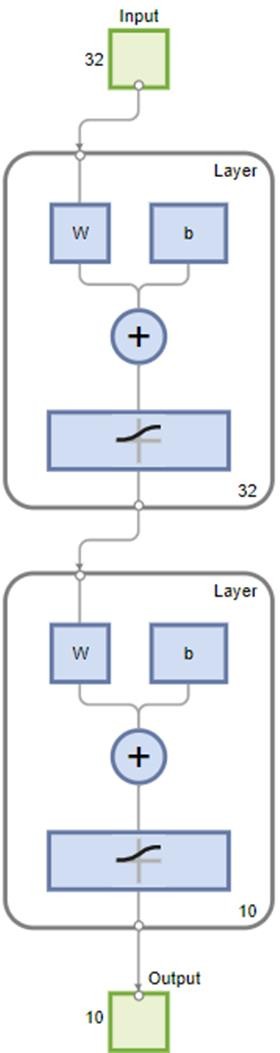
%

% GUI NNet Interface:

% nnstart (old command was "nntool")

% Можем да визуализираме невроннат мрежа: view( Digits\_FF\_NNet )

Архитектурата на изградената мрежа има вида:



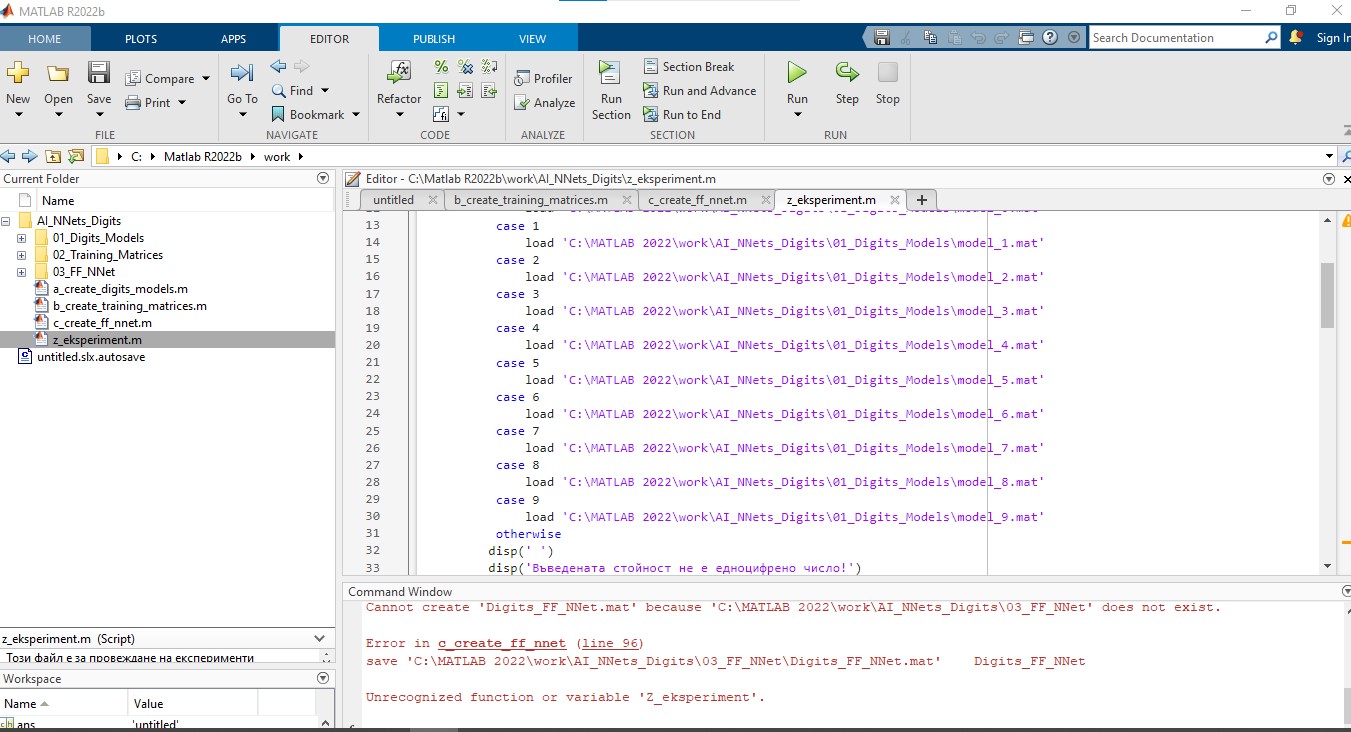
Литература:

1. David Kriesel, A Brief Introduction to Neural Networks, достъпно на <http://www.dkriesel.com/en/science/neural_networks>, посетено на 12.08.2022 г.
2. Терехов В. А., Ефимов Д. В., Тюкин И. Ю. Нейросетевые системы управления. — М.: Высшая школа, 2002. — 184 с. — ISBN 5-06-004094-1.
3. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика = Neural Computing. Theory and Practice. — М.: Мир, 1992. — 240 с. — ISBN 5- 03-002115-9.
4. Хайкин С. Нейронные сети: полный курс = Neural Networks: A Comprehensive Foundation. 2-е изд. — М.: Вильямс, 2006. — 1104 с. — ISBN 0- 13-273350-1.
5. Гульнара Яхъяева, Лекция 3. Персептроны. Обучение персептрон, достъпно на https://intuit.ru/studies/courses/88/88/print\_lecture/20531

Тема 2/Занятие 3/Лекция

**Функции на програмната среда MATLAB за симулиране на работата на изкуствени невронни мрежи.**

Общ изглед на работния екран



1. Създаване на модели на цифрите

% Този файл е за провеждане на експерименти

% clearvars -except Digits\_FF\_NNet clear

close all clc

try

disp(' ')

model = input('Моля, въведете цифра от 0 до 9 по ваше желание:\n'); switch model

case 0

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\01\_Digits\_Models\model\_0.mat' case 1

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\01\_Digits\_Models\model\_1.mat' case 2

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\01\_Digits\_Models\model\_2.mat'

end catch

end

case 3

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\01\_Digits\_Models\model\_3.mat' case 4

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\01\_Digits\_Models\model\_4.mat' case 5

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\01\_Digits\_Models\model\_5.mat' case 6

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\01\_Digits\_Models\model\_6.mat' case 7

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\01\_Digits\_Models\model\_7.mat' case 8

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\01\_Digits\_Models\model\_8.mat' case 9

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\01\_Digits\_Models\model\_9.mat' otherwise

disp(' ')

disp('Въведената стойност не е едноцифрено число!') pause(3)

z\_eksperiment z\_eksperiment

M\_help = [zeros(1,5); M zeros(8,1)];

% Фигура, изобразяваща числото figure

set( gcf, 'Position', [10 350 155 255] ) pcolor(flipud(M\_help)) colormap(flipud(gray))

axis off drawnow

% Потребителят задава математическото очаване на смущението

% (по подразбиране е 0.05) disp(' ')

m = input('Моля, въведете стойност за математическото очакване\nна смущенията, в диапазона от 0 до 0.1:\n');

% Потребителят задава дисперсията на смущението

% Ако не е въведено нищо,

% да се възприеме стойността по подразбиране 0.05 if exist( num2str(m), 'var' ) == 0

m = 0.05;

end

% Потребителят задава дисперсията на смущението

% (по подразбиране е 0..1) disp(' ')

v = input('Моля, въведете стойност за дисперсията на смущенията,\nв диапазона от 0 до 0.2:\n');

% Ако не е въведено нищо,

% да се възприеме стойността по подразбиране 0.1 if exist( num2str(v), 'var' ) == 0

v = 0.1;

end

% Внасяне на смущения в изображението: M\_noise=imnoise(M,'gaussian',m,v);

M\_noise\_help = [zeros(1,5); M\_noise zeros(8,1)];

% Фигура, изобразяваща числото close(gcf)

figure

set(gcf, 'Position', [10 350 155 255]) pcolor(flipud(M\_help)) colormap(flipud(gray))

axis off drawnow

% Фигура, изобразяваща зашуменото изображение figure

set(gcf, 'Position', [180 350 155 255]) pcolor(flipud(M\_noise\_help)) colormap(flipud(gray))

axis off drawnow

clear M\_noise\_help

% Зареждаме невронната мрежа, която е обучена за работа при шум:

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\03\_FF\_NNet\Digits\_FF\_NNet.mat'

% Процес на разпознаване: reshape\_M\_noise = reshape(M\_noise,32,1);

Result = sim( Digits\_FF\_NNet, reshape\_M\_noise);

% Фигура, изобразяваща резултата от разпознаването: figure

set(gcf, 'Position', [350 350 301 255])

bar( [0: 9], Result ) axis square

axis([-1 10 0 1])

drawnow

% Този файл създава бинарни модели на цифрите от 0 до 9 clear M

close all clc

% Тук дефинираме желаната цифра:

M = [ 0 1 1 0 ;...

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 1 | ;... |  |
| 1 | 0 | 0 | 1 | ;... |  |
| 1 | 0 | 0 | 1 | ;... |  |
| 0 | 1 | 1 | 1 | ;... |  |
| 0 | 0 | 0 | 1 | ;... |  |
| 1 | 0 | 0 | 1 | ;... |  |
| 0 | 1 | 1 | 0 |  | ] ; |

% Запомням матрицата M на хард диска

% във \*.mat файл със съответно име:

save 'C:\Matlab R2022b\work\AI\_NNets\_Digits\01\_Digits\_Models'\model\_9.mat M

% load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\01\_Digits\_Models\model\_9.mat'

% Допълвам матрицата на числото с по един ред и един стълб

% за да мога да получа хубава фигурата, изобразяваща числото: M\_help = [ zeros(1,5); M zeros(8,1)];

% Съзадавам фигура, изобразяваща числото:

figure

set( gcf, 'Position', [300 300 100 200] ) pcolor( flipud(M\_help) )

colormap( flipud(gray) ) axis off

Литература:

* 1. David Kriesel, A Brief Introduction to Neural Networks, достъпно на <http://www.dkriesel.com/en/science/neural_networks>, посетено на 12.08.2022 г.
  2. Терехов В. А., Ефимов Д. В., Тюкин И. Ю. Нейросетевые системы управления. — М.: Высшая школа, 2002. — 184 с. — ISBN 5-06-004094-1.
  3. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика = Neural Computing. Theory and Practice. — М.: Мир, 1992. — 240 с. — ISBN 5- 03-002115-9.
  4. Хайкин С. Нейронные сети: полный курс = Neural Networks: A Comprehensive Foundation. 2-е изд. — М.: Вильямс, 2006. — 1104 с. — ISBN 0- 13-273350-1.
  5. Гульнара Яхъяева, Лекция 3. Персептроны. Обучение персептрон, достъпно на https://intuit.ru/studies/courses/88/88/print\_lecture/20531

Тема 2/Занятие 4/Упражнение

# Симулиране на работата на изкуствени невронни мрежи и изследване на влиянието на основни параметри, задавани от потребителя върху качествените характеристики на получените решения.

% Този файл е за провеждане на експерименти

% clearvars -except Digits\_FF\_NNet clear

close all clc

try

disp(' ')

model = input('Моля, въведете цифра от 0 до 9 по ваше желание:\n'); switch model

case 0

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\01\_Digits\_Models\model\_0.mat' case 1

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\01\_Digits\_Models\model\_1.mat' case 2

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\01\_Digits\_Models\model\_2.mat' case 3

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\01\_Digits\_Models\model\_3.mat' case 4

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\01\_Digits\_Models\model\_4.mat' case 5

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\01\_Digits\_Models\model\_5.mat' case 6

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\01\_Digits\_Models\model\_6.mat' case 7

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\01\_Digits\_Models\model\_7.mat' case 8

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\01\_Digits\_Models\model\_8.mat' case 9

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\01\_Digits\_Models\model\_9.mat' otherwise

disp(' ')

disp('Въведената стойност не е едноцифрено число!') pause(3)

z\_eksperiment

end catch

z\_eksperiment

end

M\_help = [zeros(1,5); M zeros(8,1)];

% Фигура, изобразяваща числото figure

set( gcf, 'Position', [10 350 155 255] ) pcolor(flipud(M\_help)) colormap(flipud(gray))

axis off drawnow

% Потребителят задава математическото очаване на смущението

% (по подразбиране е 0.05) disp(' ')

m = input('Моля, въведете стойност за математическото очакване\nна смущенията, в диапазона от 0 до 0.1:\n');

% Потребителят задава дисперсията на смущението

% Ако не е въведено нищо,

% да се възприеме стойността по подразбиране 0.05 if exist( num2str(m), 'var' ) == 0

m = 0.05;

end

% Потребителят задава дисперсията на смущението

% (по подразбиране е 0..1) disp(' ')

v = input('Моля, въведете стойност за дисперсията на смущенията,\nв диапазона от 0 до 0.2:\n');

% Ако не е въведено нищо,

% да се възприеме стойността по подразбиране 0.1 if exist( num2str(v), 'var' ) == 0

v = 0.1;

end

% Внасяне на смущения в изображението: M\_noise=imnoise(M,'gaussian',m,v);

M\_noise\_help = [zeros(1,5); M\_noise zeros(8,1)];

% Фигура, изобразяваща числото close(gcf)

figure

set(gcf, 'Position', [10 350 155 255]) pcolor(flipud(M\_help)) colormap(flipud(gray))

axis off drawnow

% Фигура, изобразяваща зашуменото изображение figure

set(gcf, 'Position', [180 350 155 255]) pcolor(flipud(M\_noise\_help)) colormap(flipud(gray))

axis off

drawnow

clear M\_noise\_help

% Зареждаме невронната мрежа, която е обучена за работа при шум:

load 'C:\MATLAB 2022\work\AI\_NNets\_Digits\03\_FF\_NNet\Digits\_FF\_NNet.mat'

% Процес на разпознаване:

reshape\_M\_noise = reshape(M\_noise,32,1); Result = sim( Digits\_FF\_NNet, reshape\_M\_noise);

% Фигура, изобразяваща резултата от разпознаването:

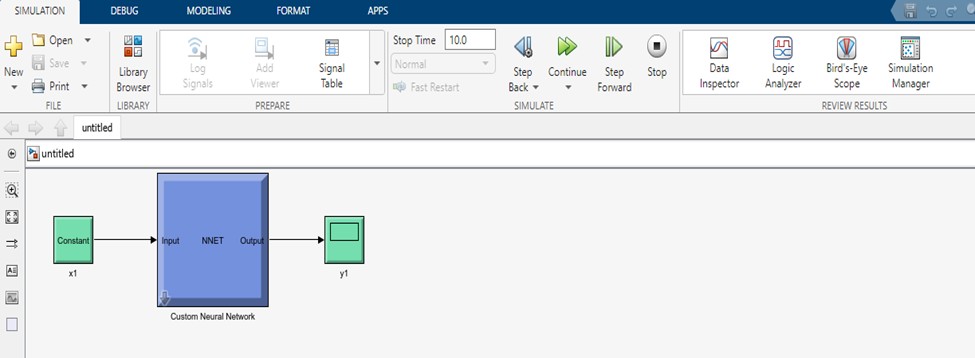
figure

set(gcf, 'Position', [350 350 301 255])

bar( [0: 9], Result ) axis square

axis([-1 10 0 1])

drawnow



Литература:

1. David Kriesel, A Brief Introduction to Neural Networks, достъпно на <http://www.dkriesel.com/en/science/neural_networks>, посетено на 12.08.2022 г.
2. Терехов В. А., Ефимов Д. В., Тюкин И. Ю. Нейросетевые системы управления. — М.: Высшая школа, 2002. — 184 с. — ISBN 5-06-004094-1.
3. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика = Neural Computing. Theory and Practice. — М.: Мир, 1992. — 240 с. — ISBN 5- 03-002115-9.
4. Хайкин С. Нейронные сети: полный курс = Neural Networks: A Comprehensive Foundation. 2-е изд. — М.: Вильямс, 2006. — 1104 с. — ISBN 0- 13-273350-1.
5. Гульнара Яхъяева, Лекция 3. Персептроны. Обучение персептрон, достъпно на https://intuit.ru/studies/courses/88/88/print\_lecture/20531

# Тема 2/Занятие 5/Лекция

**Възможности на програмната среда MATLAB за разработване на програмни решения на типични задачи от областта на идентификацията и разпознаване на образи с използване на невронни мрежи..**

# лекция

**Подходи на обучение на изкуствени невронни мрежи**

В постепенно обучение (***incremental training***), теглата и отклоненията на мрежата се актуализират всеки път, когато входът е представен на мрежата.

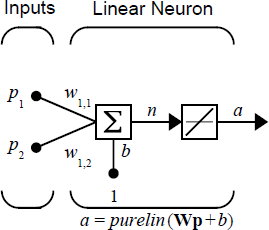
При групово обучение (***batch training***) теглата и отклоненията се актуализират само след представяне на всички входове.

Постепенно обучение (на адаптивни и друг вид мрежи)

Постепенното обучение може да се прилага както за статични, така и за динамични мрежи, въпреки че се използва по-често с динамични мрежи, като адаптивни филтри.

# Постепенно обучение на статични мрежи

Нека разглеждаме следната статична мрежа.



Искаме да я обучим постепенно, така че теглата и отклоненията да се актуализират след представяне на всеки вход. В този случай използваме функцията **adapt**, и представяме входовете и целите като последователности.

Да предположим, че искаме да обучим мрежата да създава линейната функция

***t* = 2*p*1 + *p*2**

Тогава входовете ще бъдат:



А изходите ще бъдат:



Първоначално настройваме мрежата с нулеви първоначални тегла и отклонения. Задаваме скоростта на обучение на нула, за да покаже ефекта от нарастващото обучение.

# net = newlin([-1 1;-1 1],1,0,0);

**net.IW{1,1} = [0 0];**

# net.b{1} = 0;

За постепенно обучение искаме да представим входовете и целите като последователности:

# P = {[1;2] [2;1] [2;3] [3;1]};

**T = {4 5 7 7};**

Знае се, че при симулация статичната мрежа мрежата произвежда едни и същи изходи, независимо дали входовете са представени като матрица от едновременни вектори или като клетъчен масив от последователни вектори.

# Това обаче не се отнася за обучение на мрежата.

Когато използвате функцията за адаптиране, ако входовете са представени като клетъчен масив на последователни вектори, тогава теглата се актуализират при всеки вход представен (постепенен режим).

Както се вижда в следващия раздел, ако входовете са представени като матрица от едновременни вектори, тогава теглата са актуализира се само след представяне на всички входове (пакетен режим).

За постепенно обучение на мрежата се въвежда командата:

# [net,a,e,pf] = adapt(net,P,T);

Мрежовите изходи ще останат нула, тъй като скоростта на обучение е нула, и теглата не се актуализират. Грешките ще бъдат равни на целите:

# a = [0] [0] [0] [0]

**e = [4] [5] [7] [7]**

Ако сега зададем скоростта на обучение на 0,1, можем да видим как е мрежата коригирани при представяне на всеки вход:

# net.inputWeights{1,1}.learnParam.lr=0,1; net.biases{1,1}.learnParam.lr=0.1; [net,a,e,pf] = adapt(net,P,T);

**a = [0] [2] [6,0] [5,8]**

# e = [4] [3] [1,0] [1,2]

Първият изход е същият, както беше с нулева скорост на обучение, тъй като не актуализацията се извършва до представяне на първия вход. Вторият изход е различни, тъй като теглата са актуализирани. Тежестите продължават да да се променят при изчисляването на всяка грешка. Ако мрежата е способна и скоростта на обучение е зададена правилно, грешката в крайна сметка ще бъде доведена до нула.

# Постепенно обучение на динамични мрежи

Можем също да обучаваме динамични мрежи постепенно. Всъщност това би било най-честата ситуация. Да вземем линейната мрежа с едно закъснение на входа, който използвахме в предишен пример. Инициализираме тегла на нула и задайте скоростта на обучение на 0,1.

# net = newlin([-1 1],1,[0 1],0.1);

**net.IW{1,1} = [0 0];**

# net.biasConnect = 0;

За да обучим тази мрежа постепенно, ние представяме входовете и целите като елементи от клетъчни масиви.

# Pi = {1};

**P = {2 3 4};**

# T = {3 5 7};

Тук се опитваме да обучим мрежата да сумира текущото и предишното входове за създаване на текущия изход. Това е същата входна последователност, която ние използвани в предишния пример за използване на sim, с изключение на това, че присвояваме първия термин в последователността като начално условие за закъснението. Вече можем последователно обучаване мрежата:

# [net,a,e,pf] = adapt(net,P,T,Pi);

**a = [0] [2,4] [7,98]**

# e = [3] [2,6] [-1,98]

Първият изход е нула, тъй като теглата все още не са актуализирани.

Теглата се променят при всяка следваща времева стъпка.

# Пакетно обучение

Пакетно обучение, при което теглата и пристрастията се актуализират само след всичко входовете и целите са представени, могат да се прилагат както към статични, така и към динамични мрежи. В този раздел обсъждаме и двата типа мрежи.

# Пакетно обучение на статични мрежи

Пакетното обучение може да се извърши с помощта на адаптиране или обучение, макар и обучение като цяло е най-добрият вариант, тъй като обикновено има достъп до по-ефективни тренировъчни алгоритми. Постепенното обучение може да се извърши само с adapt;

Нека започнем със статичната мрежа, която използвахме в предишните примери. Скоростта на обучение ще бъде зададена на 0,1.

# net = newlin([-1 1;-1 1],1,0,0.1);

**net.IW{1,1} = [0 0];**

# net.b{1} = 0;

За групово обучение на статична мрежа с adapt, входните вектори трябва да бъдат поставени в една матрица от едновременни вектори.

# P = [1 2 2 3; 2 1 3 1];

**T = [4 5 7 7];**

Когато извикаме **adapt**, тя ще извика влакове (което е по подразбиране адаптационна функция за линейната мрежа) и learnwh (което е функция за обучение по подразбиране за тегла и отклонения). Следователно, Използва се обучение на Widrow-Hoff.

# [net,a,e,pf] = adapt(net,P,T); a = 0 0 0 0

**e = 4 5 7 7**

Обърнете внимание, че всички изходи на мрежата са нула, защото теглата са такива не се актуализира, докато не бъде представен целият комплект за обучение. Ако показваме теглата, които намираме:

# »net.IW{1,1}

**ans = 4,9000 4,1000**

# »net.b{1} ans = 2,3000

Това е различно от резултата, който получихме след едно преминаване на адаптация постепенно актуализиране.

Сега нека извършим същото партидно обучение с помощта на влак. Тъй като Правилото на Widrow-Hoff може да се използва в инкрементален или пакетен режим, може да бъде извиква се от **adapt** или **train**. Има няколко алгоритъма, които могат да бъдат само използвани в пакетен режим (напр. Levenberg-Marquardt), и така тези алгоритми може да се извика само с **train**.

Мрежата ще бъде настроена по същия начин.

# net = newlin([-1 1;-1 1],1,0,0.1);

**net.IW{1,1} = [0 0];**

# net.b{1} = 0;

В този случай входните вектори могат или да бъдат поставени в матрица на едновременни вектори или в клетъчен масив от последователни вектори. Във **train** всеки клетъчен масив от последователни вектори се преобразува в матрица от едновременни вектори. Това е така, защото мрежата е статична и защото тренирайте винаги работи в пакетен режим. Обикновено се използва

работа в паралелен режим когато е възможно, защото има по-ефективен MATLAB изпълнение.

# P = [1 2 2 3; 2 1 3 1];

**T = [4 5 7 7];**

Сега сме готови да обучим мрежата. Ще го обучим само за една епоха, тъй като използвахме само едно преминаване на **adapt**. Функцията за обучение по подразбиране за линейната мрежа е **trainc**, а функцията за обучение по подразбиране за тегла е **learnwh**, така че трябва да получим същите резултати като нас получен с помощта на **adapt** в предишния пример, където по подразбиране адаптационната функция беше train.

# net.inputWeights{1,1}.learnParam.lr = 0,1;

**net.biases{1}.learnParam.lr = 0,1;**

# net.trainParam.epochs = 1; net = train(net,P,T);

Ако покажем теглата след една епоха на обучение, намираме:

# »net.IW{1,1}

**ans = 4,9000 4,1000**

# »net.b{1} ans = 2,3000

Това е същият резултат, който получихме с обучението в пакетен режим в adapt.

При статични мрежи функцията за адаптиране може да реализира инкрементални или пакетно обучение в зависимост от формата на входните данни. Ако данните са представен като матрица от едновременни вектори, ще

се извърши пакетно обучение. Ако данните се представят като последователност, ще се извърши постепенно обучение. Това не е вярно за **train**, която винаги изпълнява пакетно обучение, независимо от това от формата на входа.

# Пакетно обучение на динамични мрежи

Обучението на статични мрежи е относително лесно. Ако използваме **train** мрежата се обучава в пакетен режим и входовете се преобразуват в едновременни вектори (колони на матрица), дори и да са първоначално предавани като последователност (елементи от клетъчен масив).

Ако използваме **adapt**, форматът на входа определя метода на обучение. Ако входовете се предават като последователност, тогава мрежата се обучава в инкрементален режим. Ако входовете се предават като едновременни вектори, тогава се използва обучение в пакетен режим.

При динамичните мрежи обикновено се извършва обучение в пакетен режим.

Нека разгледаме отново линейната мрежа със закъснение. Ние използваме коефициент на обучение от 0,02. (Когато се използва алгоритъм с градиентно спускане е необходимо да се използва по-малка скорост от тази за обучение в пакетен режим.

Градиентите се сумират заедно, преди да се определи стъпката на промяна на тегловите коефициенти.

# net = newlin([-1 1],1,[0 1],0.02);

**net.IW{1,1}=[0 0];**

# net.biasConnect=0; net.trainParam.epochs = 1;

**Pi = {1};**

# P = {2 3 4};

**T = {3 5 6};**

Искаме да обучим мрежата със същата последователност, която използвахме за постепенно обучение по-рано, но този път искаме да актуализираме само теглата след като всички входове са приложени (пакетен режим).

Мрежата се симулира в последователен режим, тъй като входът е последователен, но теглата се актуализират в пакетен режим.

# net=train(net,P,T,Pi);

**Тежестите след една епоха на тренировка са**

# »net.IW{1,1}

**ans = 0,9000 0,6200**

Това са различни тегла от тези, които бихме получили с инкременталнa тренировка, където тежестите ще се актуализират три пъти по време на една преминаване на обучение. За пакетно обучение тежестите са само актуализиран веднъж във всяка епоха.

Литература:

1. David Kriesel, A Brief Introduction to Neural Networks, достъпно на <http://www.dkriesel.com/en/science/neural_networks>, посетено на 12.08.2022 г.
2. Терехов В. А., Ефимов Д. В., Тюкин И. Ю. Нейросетевые системы управления. — М.: Высшая школа, 2002. — 184 с. — ISBN 5-06-004094-1.
3. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика = Neural Computing. Theory and Practice. — М.: Мир, 1992. — 240 с. — ISBN 5- 03-002115-9.
4. Хайкин С. Нейронные сети: полный курс = Neural Networks: A Comprehensive Foundation. 2-е изд. — М.: Вильямс, 2006. — 1104 с. — ISBN 0- 13-273350-1.
5. Гульнара Яхъяева, Лекция 3. Персептроны. Обучение персептрон, достъпно на https://intuit.ru/studies/courses/88/88/print\_lecture/20531

Тема 2/Занятие 6/Упражнение

# Софтуерни средства за моделиране на невронни мрежи.

Разработване на програмни решения на типични задачи от областта на идентификацията и разпознаване на образи с използване на невронни мрежи.

Да се анализира как форматът на структурите на входните данни влияе върху симулацията на мрежи. Мрежите могат да са статични и динамични.

# Анализ на статични мрежи.

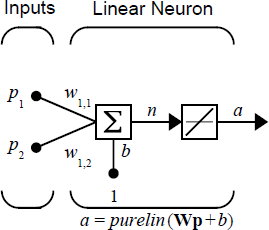
Ще бъдат разгледани два основни типа входни вектори: тези, които възникват едновременно (по едно и също време или в неопределена времева последователност) и тези, които се появяват последователно във времето.

За едновременни вектори редът не е важен и ако имахме няколко мрежи, работещи паралелно, можем да представим един входен вектор към всяка от мрежите. За последователни вектори, редът, в който се появяват векторите, е важен.

# Симулация с едновременни входове в статична мрежа

Най-простата ситуация за симулиране на мрежа възниква, когато мрежата, която ще се симулира, е статична (няма обратна връзка или закъснения). В такъв случай, не е от значение дали входните вектори се случват в определена времева последователност, така че можем да третираме входовете като едновременни. В допълнение, ние правим проблема още по- прост, като приемем че мрежата има само един входен вектор.

Нека за пример да се използва следната мрежа.



За да настроим тази препращаща мрежа, можем да използваме следната команда.

# net = newlin([1 3;1 3],1);

За простота задайте матрицата на теглото и отклонението, което трябва да бъде

# W = [1 2 ] *b* =[ 0]

Командите за тези назначения са

# net.IW{1,1} = [1 2];

**net.b{1} = 0;**

Да предположим, че наборът от данни за мрежова симулация се състои от Q = 4 едновременни вектори:

Едновременните вектори се представят на мрежата като единична матрица: **P = [1 2 2 3; 2 1 3 1];**

Вече можем да симулираме мрежата:

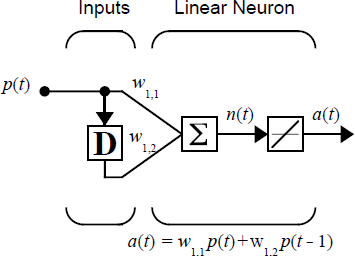
# A = sim(net,P) А =

**5 4 8 5**

Единична матрица от едновременни вектори се представя на мрежата и на мрежата произвежда единична матрица от едновременни вектори като изход. Резултатът би бил същият, ако имаше четири работещи мрежи паралелно и всяка мрежа получава един от входните вектори и произвежда един от изходите. Подреждането на входните вектори не е важно, тъй като те не взаимодействат помежду си.

# Симулация с последователни входове в динамична мрежа

Когато мрежата съдържа закъснения, входът към мрежата е последователност от входни вектори, които се появяват в определен времеви ред. За да илюстрираме този случай, използваме проста мрежа, която съдържа едно забавяне.



Следните команди създават тази мрежа:

# net = newlin([-1 1],1,[0 1]);

**net.biasConnect = 0;**

Задайте тегловната матрица, която да бъде W = [1 2]

Командата е

# net.IW{1,1} = [1 2];

Да предположим, че входната последователност е



Последователните входове се представят на мрежата като елементи от клетъчен масив:

# P = {1 2 3 4};

Вече можем да симулираме мрежата:

# A = sim(net,P) А =

**[1] [4] [7] [10]**

Въвеждаме клетъчен масив, съдържащ последователност от входове, и мрежата произвежда клетъчен масив, съдържащ последователност от изходи. Имайте предвид, че редът на входовете е важен, когато са представени като последователност.

В този случай текущият изход се получава чрез умножаване на текущия вход по 1 и предходния вход по 2 и сумиране на резултата. Ако променим реда на входовете, това ще промени числата, които ще получим в изхода.

# Симулация с едновременни входове в динамични мрежи

Ако трябваше да приложим същите входове от предишния пример като набор на едновременни входове вместо последователност от входове, бихме получили a напълно различен отговор.Би било така, сякаш всеки вход бяха приложени едновременно към отделна паралелна мрежа. За предишния пример, ако използваме паралелен набор от входове, които имаме

който може да бъде създаден със следния код:

# P = [1 2 3 4];

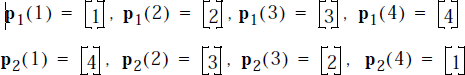
Когато симулираме с едновременни входове, получаваме

# A = sim(net,P) А =

**1 2 3 4**

Резултатът е същият, както ако бяхме приложили едновременно всеки един от входове към отделна мрежа и изчислен един изход. Имайте предвид, че оттогава ние не присвоихме никакви начални условия на мрежовите закъснения, те бяха се приема за нула. В този случай резултатът просто ще бъде 1 пъти по-голям вход, тъй като теглото, което умножава текущия вход, е

1. В някои специални случаи може да искаме да симулираме отговора на мрежата към няколко различни последователности едновременно. В този случай бихме искате да представите мрежата с едновременен набор от последователности. За например, да речем, че искаме да представим следните две последователности на мрежата:



Входът P трябва да бъде клетъчен масив, където всеки елемент от масива съдържа двата елемента от двете последователности, които се появяват едновременно:

# P = {[1 4] [2 3] [3 2] [4 1]};

Вече можем да симулираме мрежата:

# A = sim(net,P);

Полученият мрежов изход ще бъде

# A = {[ 1 4] [4 11] [7 8] [10 5]}

Както можете да видите, първата колона на всяка матрица съставлява изхода последователност, произведена от първата входна последователност, която беше тази, която беше използвана в предходен пример. Втората колона на всяка матрица съставлява изходната последователност, произведена от втората входна последователност. Няма взаимодействие между двете едновременни последователности. Все едно са били приложени към отделни мрежи, работещи паралелно.

Литература:

* 1. David Kriesel, A Brief Introduction to Neural Networks, достъпно на <http://www.dkriesel.com/en/science/neural_networks>, посетено на 12.08.2022 г.
  2. Терехов В. А., Ефимов Д. В., Тюкин И. Ю. Нейросетевые системы управления. — М.: Высшая школа, 2002. — 184 с. — ISBN 5-06-004094-1.
  3. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика = Neural Computing. Theory and Practice. — М.: Мир, 1992. — 240 с. — ISBN 5- 03-002115-9.
  4. Хайкин С. Нейронные сети: полный курс = Neural Networks: A Comprehensive Foundation. 2-е изд. — М.: Вильямс, 2006. — 1104 с. — ISBN 0- 13-273350-1.
  5. Гульнара Яхъяева, Лекция 3. Персептроны. Обучение персептрон, достъпно на <https://intuit.ru/studies/courses/88/88/print_lecture/20531>
  6. David Kriesel, A Brief Introduction to Neural Networks, достъпно на <http://www.dkriesel.com/en/science/neural_networks>

Тема 2/Занятие 7/Семинар

# Софтуерни средства за изграждане и обучение и симулация на изкуствени невронни мрежи

Задача 1. По предложения списък от въпроси обучаемите се подготвят предварително и представят по време на зянятието доклад в обем до четири страници и по тяхна преценка презентация в обем до десет слайда.

По всеки от въпросите се провежда дискусия в рамките на тридесет минути.

1. Възможности на програмната среда MATLAB за изграждане и обучение на изкуствени невронни мрежи.
2. Необходимост от използване на софтуерни средства в помощ на проектирането и симулацията на изкуствени невронни мрежи.

Задача 2. Обучаемите самостоятелно да проучат и разработят представяне на софтуерни средства за изграждане и обучение и симулация на изкуствени невронни мрежи.

Литература:

1. Минчев, Ч., Основи на програмирането в Матлаб, Шумен, 2017, ISBN 978-954-9681-35-2;

2.

1. David Kriesel, A Brief Introduction to Neural Networks, достъпно на <http://www.dkriesel.com/en/science/neural_networks>, посетено на 12.08.2022 г.
2. Терехов В. А., Ефимов Д. В., Тюкин И. Ю. Нейросетевые системы управления. — М.: Высшая школа, 2002. — 184 с. — ISBN 5-06-004094-1.
3. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика = Neural Computing. Theory and Practice. — М.: Мир, 1992. — 240 с. — ISBN 5- 03-002115-9.
4. Хайкин С. Нейронные сети: полный курс = Neural Networks: A Comprehensive Foundation. 2-е изд. — М.: Вильямс, 2006. — 1104 с. — ISBN 0- 13-273350-1.
5. Гульнара Яхъяева, Лекция 3. Персептроны. Обучение персептрон, достъпно на https://intuit.ru/studies/courses/88/88/print\_lecture/20531