Kaheteistkümnes peatükk

TEHISINTELLEKTI KASUTAMINE TERVISHOIUS

Elli Valla

Sisukord

Sissejuhatus tehisintellekti maailma

„Artificial intelligence is the new electricity.“
– Andrew Ng, TI pioneer, Coursera kaasasutaja, ex-Google Brain

Kujutage ette maailma, kus masinad aitavad arstil avastada haiguse varem, kui inimene seda märkaks, kus nutikad süsteemid hoiavad liikluse sujuvana ja kus õpikeskkonnad kohanevad iga õppija vajaduste järgi. See ei ole enam tuleviku utoopia, vaid tehisintellekti (TI) reaalsus, mis vormib meie elu juba täna. Kuid mis see TI õigupoolest on? Unustagem ulmefilmidest tuttavad kõikvõimsad robotid. **Tegelikkuses on TI algoritm – nutikas matemaatiline juhis, mis õpetab arvutit lahendama konkreetseid probleeme, analüüsima andmeid ja isegi looma midagi täiesti uut.**

Viimase kümnendi jooksul on algoritmipõhised süsteemid teinud revolutsioonilisi läbimurdeid teaduses, meditsiinis ja meie igapäevaelus, muutes peaaegu iga valdkonda, mida me tunneme. Üks silmapaistvamaid näiteid TI võimekusest on **AlphaFold** (Google DeepMind, s.a.-a). See DeepMindi loodud TI suudab enneolematu täpsusega **ennustada valkude kolmemõõtmelist struktuuri**. See on tohutu teaduslik saavutus, sest valgud on elu ehituskivid, mis vastutavad peaaegu kõigi bioloogiliste protsesside eest meie kehas. Aastakümneid oli valkude struktuuri kindlaksmääramine aeganõudev, kallis ja keeruline eksperimentaalne protsess. Enne AlphaFoldi oli teadlastel kindlaks tehtud vaid umbes 17% kõigi teadaolevate valkude 3D-struktuurist. Tänu AlphaFoldile on see osakaal tõusnud juba üle 99%, mis on toonud kaasa uusi lootusi ravimiarenduses, haiguste mõistmisel ja 2024. aastal ka Nobeli keemiaauhinna. See tehnoloogia tugineb **sügavõppele** (eesti keelses kirjanduses kasutatakse ka terminit **süvaõpe**) – TI meetodile, mis on erakordselt võimekas leidmaks mustreid ja seoseid suurtes ja keerulistes andmehulkades.

Viimaste aastate suurim läbimurre on aga kahtlemata **generatiivne TI**. Mudelid nagu **ChatGPT** (OpenAI, 2025), **Gemini** (Google, 2025), **Claude** (Anthropic, 2025) jt on muutnud seda, kuidas me masinatega suhtleme. Need keelemudelid ei vasta lihtsalt küsimustele, vaid suudavad luua uut sisu: kirjutada e-kirju, tõlkida meditsiinilisi dokumente, koostada patsientide haiguslugude kokkuvõtteid, isegi luua loovtekste ja kunstiteoseid. Nende mudelite taga on murranguline **transformer-tüüpi arhitektuur**, mis sündis alles 2017. aastal ja on nüüdseks kujunenud kaasaegse TI alustehnoloogiaks, sillutades teed uutele murrangulistele võimalustele.

Joonis 12.1 Tehisintellekti harud (E. Valla, 2025).

Tehisintellekt on lai ja mitmetahuline mõiste, mis hõlmab erinevaid alavaldkondi. Kõige üldisemalt on TI **igasugune tehnika, mis õpetab arvuteid jäljendama inimkäitumist**. TI näiteks võib tuua isesõitva auto. Kuigi selline süsteem kasutab masinõppemudeleid, ei koosne see ainult neist. Isesõitvates autodes rakendatakse ka palju muud tehnoloogiat, mis simuleerib inimlikku taju ja käitumist, näiteks klassikalist reeglipõhist otsustusloogikat, andurite andmete töötlemise algoritme ja ohutusprotokolle. Reeglitele ja loogikale tuginevat lähenemist nimetatakse **sümboolseks tehisintellektiks (ingl k *symbolic AI*)**. See põhineb inimese loodud selgetel reeglitel ja teadmiste esitamisel loogiliste seoste kaudu. Näiteks liiklusreeglid ja ohutusnõuded saab vormistada „kui-siis" tüüpi otsustusreegliteks: kui ees on stoppmärk, siis peatu; kui jalakäija astub tee servale, siis vähenda kiirust. Sümboolse TI eeliseks on läbipaistvus ja kontrollitavus, kuid see ei suuda kohaneda uute olukordadega, mida reeglites ei ole ette nähtud. Seepärast kombineeritakse sümboolset TI-d üha sagedamini andmepõhise masinõppega, et ühendada loogiline juhitavus ja õppimisvõime ühtseks süsteemiks. Masinõpe on TI üks oluline alamharu, mis seisneb võimes õppida andmetest ja teha prognoose või otsuseid ilma, et kõiki reegleid oleks vaja enne sõnaselgelt programmeerida. Nii keerukad süsteemid nagu ChatGPT või Gemini kui ka lihtsamad algoritmid, nagu näiteks lineaarne regressioon (statistiline mudel, mis leiab andmetest sirgjoonelisi seoseid) või isegi malet mängiv programm, mis on loodud spetsiifiliste reeglite järgi, kuuluvad TI laiema definitsiooni alla, vt joonis 12.1.

- **Masinõpe (ingl k *machine learning*)** – TI haru, kus süsteemid õpivad andmetest iseseisvalt, ilma et inimene peaks neile reegleid ette kirjutama.
- **Sügavõpe (ingl k *deep learning*)** – masinõppe haru, mis kasutab nn närvivõrke. See algoritm võimaldab leida keerulisi ja varjatud mustreid väga suurtes ja komplekssetes andmehulkades, näiteks piltidel või helis. Valdkond, kus arenevad transformer-tüüpi arhitektuurid, mis on tänapäevase generatiivse TI aluseks.
- **Generatiivne TI (ingl k *generative AI)*** – TI mudelid, mis loovad uusi sisendeid (nt tekste, pilte, muusikat või andmeid), imiteerides inimloomingut. Siia alla kuuluvad tuntud generatiivsed mudelid ChatGPT, Gemini, Claude, jne.

12.1 Tehisintellekti mõistmine – alustest rakendusteni

„Artificial intelligence is not science fiction – it's an engineering discipline."
– Dario Amodei (Anthropicu kaasasutaja)

Et paremini mõista tänapäeva TI-süsteeme, sukeldume samm-sammult nende telgitagustesse. Käsitleme esmalt põhilisi ehitusplokke, seejärel liigume edasi keerukamate kontseptsioonide ja praktiliste rakendusteni, keskendudes tervishoiu kontekstile.

12.1.1 Tehisintellekti põhimõtted ja õppimise tüübid

Mida TI vajab, et „mõelda"?

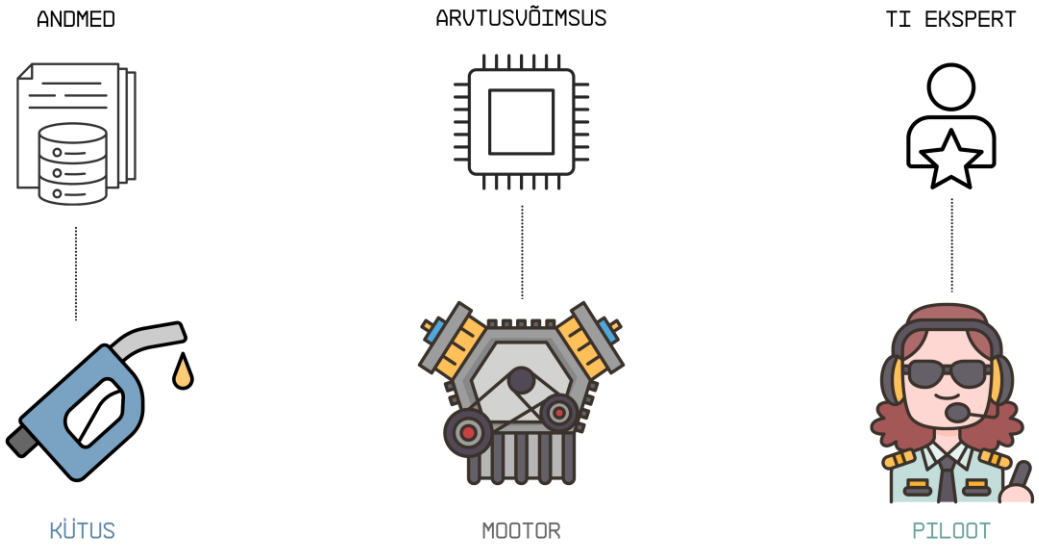
Nagu eespool rõhutasime, ei ole TI maagia ega ulmefilmide fantaasia. See on **algoritmide kogum** – nutikate, süsteemsete reeglite süsteem, mis suudab õppida ja otsuseid teha. Kuid isegi kõige tõhusam algoritm on kasutu, kui sellel puuduvad kolm elutähtsat eeldust (vt joonis 12.2).

1. **Andmed – TI „kütus".** Kujutage ette, et proovite õpetada last lugema ilma ühtegi raamatut näitamata. Samamoodi ei saa ka TI õppida ja tegutseda ilma kvaliteetsete ja piisavalt suurte andmehulkadeta. Need andmed on TI-süsteemi „kütus". Tervishoius tähendab see tohutut hulka digitaalseid haiguslugusid, meditsiinilisi pilte (nt röntgenpildid, MRT-d, KT-d), sensorite kogutud eluliste näitajate andmeid, patsiendi enesekohaseid sisendeid ja palju muud. Mida rohkem ja kvaliteetsemaid andmeid TI-süsteem saab, seda paremini suudab ta mustreid leida ja täpseid ennustusi teha.

2. **Arvutusvõimsus – TI „mootor".** Suurte andmehulkade töötlemiseks ja keeruliste algoritmide käitamiseks on vaja tohutut arvutusvõimsust. See on TI „mootor". Eriti **sügavõppe** puhul, mis kasutab hiiglaslikke närvivõrke, on vaja spetsialiseeritud riistvara nagu võimsad **graafikaprotsessorid** (ingl k *Graphics Processing Unit, GPU***)** või Google'i loodud **tensorprotsessori üksused (**ingl k *Tensor Processing Unit, TPU***)**. Need kiibid on loodud paralleelsete arvutuste jaoks, võimaldades treenida suuri närvivõrke päevade või tundidega, mitte nädalate või kuudega. Õnneks on pilveplatvormid (nagu Google Cloud, Microsoft Azure, Amazon Web Services) selle kättesaadavust oluliselt lihtsustanud, pakkudes vajalikku arvutusressurssi.

3. **Inimekspertiis ehk talent – TI „piloot".** Isegi parim auto ja kõige parem kütus vajavad osavat juhti. Samamoodi vajab TI inimesi – andmeteadlasi, masinõppeinsenere, valdkonna- ehk domeeniekspert, kes oskavad andmeid puhastada, mudeleid üles ehitada, nende tööd optimeerida ja tulemusi kriitiliselt hinnata. Tervishoius on see eriti oluline, sest siin on tegemist inimeludega. See tähendab arstide, andmeteadlaste ja arendajate tihedat koostööd tagamaks, et loodud algoritmid vastavad tegelikele kliinilistele vajadustele ja on turvalised kasutada.

Kõige selle keskmes ongi **algoritmid** – reeglite süsteemid, mis otsustavad, kuidas andmeid töödelda, neist mustreid leida ja tulemusi ennustada. Need võivad olla äärmiselt lihtsad, nagu



Joonis 12.2. Tehisintellekti juhttegurid (E. Valla, 2025).

näiteks **lineaarne regressioon** (matemaatiline mudel, mis aitab ennustada ühe muutuja väärtust teise alusel, näiteks patsiendi vanuse ja vererõhu seoseid), või ülimalt keerukad, nagu **transformer-arhitektuurid**, mida kasutatakse näiteks ChatGPT-s loomuliku keele mõistmiseks ja genereerimiseks ning meditsiiniliste piltide täpseks tõlgendamiseks.

Mõistmine, mis tüüpi algoritme ja andmeid erinevates olukordades kasutatakse, aitab paremini hinnata TI vahendite sobivust, usaldusväärsust ja piiranguid just tervishoiu tundlikus ja kriitilises valdkonnas.

Mille poolest erineb TI ehk masinõpe klassikalisest programmeerimisest? Et paremini mõista masinõppe revolutsioonilist olemust, on kasulik võrrelda seda traditsioonilise ehk **klassikalise programmeerimisega**. See erinevus on TI arengu tuumaks ja aitab selgitada, miks masinad suudavad tänapäeval teha asju, mis varem tundusid ulmelisena. **Klassikalises programmeerimises** on protsess lihtne ja loogiline (vt joonis 12.3):

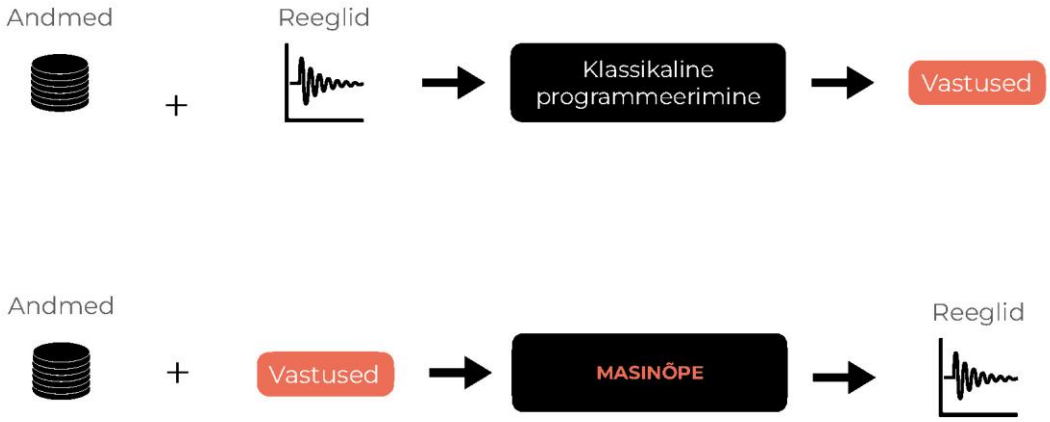
- **andmed** sisestatakse arvutisse;
- **inimene kirjutab reeglid** ehk juhised, kuidas neid andmeid töödelda (näiteks „kui kehatemperatuur on üle 37,5 kraadi, siis on palavik“);
- arvuti rakendab neid reegleid andmetele ja **annab vastuse**.

See on suurepärane lahendus paljude ülesannete jaoks, kus reeglid on selged ja püsivad, näiteks kalkulaatorid, pangatehingud või lihtsad andmebaasipäringud. Aga mis juhtub siis, kui reeglid on liiga keerulised, et neid käsitsi kirja panna, või kui need pidevalt muutuvad? Näiteks, kuidas programmeerida reeglid, mis kirjeldaksid, kas pildil on kass või koer? Või millised mustrid viitavad vähile mammogrammil?

Masinõpe muudab protsessi järgmiselt:

- süsteemile antakse ette **andmed** (näiteks tuhanded pildid kassidest ja koertest);
- iga andmejuhtumi kohta antakse ka **õiged vastused** (näiteks „see pilt on kass“, „see pilt on koer“);
- **masinõppe algoritm analüüsib neid andmeid ja vastuseid ning „õpib“ ise reeglid** ehk mustrid (matemaatilises keeles funktsioonid), mis andmeid ja vastuseid seovad.

See on fundamentaalne nihe. Masin ei pea enam ootama, et inimene talle kõikvõimalikud reeglid ette kirjutaks. Selle asemel **suudab masin ise õppida mustreid ja luua reegleid andmete põhjal**. Just see võimekus õppida keerulisi ja sageli inimestele nähtamatuid seoseid andmetest on teinud masinõppest nii võimsa tööriista, mis on pannud aluse ka TI-revolutsioonile tervishoius ja paljudes teistes valdkondades.



Joonis 12.3. Masinõppe erinevus klassikalisest programmeerimisest (juhendatud õppe näitel).

Kui me räägime masinõppest, siis räägime tegelikult protsessist, kus mudel õpib seoseid andmete ja nendele vastavate tulemuste vahel. Olenevalt sellest, millist infot süsteem õppimiseks saab ja kuidas ta tagasisidet vastu võtab, eristatakse kolme peamist masinõppe tüüpi.

1. **Juhendatud õpe (**ingl k *supervised learning***)**

See on masinõppe kõige levinum ja ehk ka intuitiivsem vorm. Kujutage ette õpilast, kes õpib eksamiks vanade eksamite ja nende õigete vastuste abil. Juhendatud õppe puhul saab süsteem õppimiseks märgendatud andmed – igale sisendile on lisatud ka „õige vastus“ ehk märgend (ingl k *label*). Süsteem õpib seoseid sisendi ja väljundi vahel, et suudaks seejärel ennustada uute, seni nägemata andmete puhul õigeid vastuseid.

Näide tervishoiust

Algoritmile antakse tuhandeid nahapilte, millest igaühele on dermatoloog märkinud, kas tegemist on healoomulise pigmendilaigu, pahaloomulise melanoomi või millegi muuga. Mudel õpib tuvastama, millised pildimustrid viitavad millisele diagnoosile, ja suudab seejärel aidata arstidel uusi nahapilte hinnata.

2. **Juhendamata õpe (**ingl k *unsupervised learning***)**

Erinevalt juhendatud õppest, kus süsteemile antakse ette nii küsimus kui ka vastus, antakse juhendamata õppes süsteemile andmed ilma etteantud vastusteta. Eesmärk on, et süsteem avastaks ise andmetes peituvad mustrid, struktuurid, seosed või rühmitused.

Näide tervishoiust

Patsientide haiguslugude põhjal algoritm klasterdab ehk rühmitab patsiente sarnaste sümptomite, laborinäitajate või haiguskulude alusel. See aitab leida uusi alarühmi diabeedi või südamehaiguste puhul, mis võivad vajada erinevat ravi, isegi kui neil ei ole veel selget diagnoosi või nad näivad esmapilgul sarnased.

3. **Tugevdusõpe (**ingl k *reinforcement learning***)**

See masinõppe tüüp sarnaneb kõige enam sellega, kuidas me ise lapsena õpime – katse ja eksituse kaudu. Süsteem tegutseb keskkonnas ning saab oma tegevuse eest tagasisidet – „tasu“ edukate sammude eest ja „karistuse“ ebaõnnestumiste eest. Eesmärk on maksimeerida saadud „tasu“ pikema aja jooksul. Seda lähenemist kasutatakse olukordades, kus süsteem peab tegema järjestikuseid otsuseid, arvestades pidevalt keskkonna reaktsiooni.

Näide tervishoiust

Virtuaalne assistent õpib optimeerima intensiivravi patsiendi ravimite doose. Süsteem annab ravimi, jälgib patsiendi eluliste näitajate (nt vererõhk, pulss) muutumist ja saab tagasisidet selle kohta, kas patsiendi seisund paranes („tasu“) või halvenes („karistus“). Aja jooksul õpib süsteem leidma optimaalset ravistrateegiat iga patsiendi jaoks.

Oluline on märkida, et praktikas kasutatakse neid lähenemisi sageli **kombineeritult**. Näiteks võib süsteem esmalt kasutada juhendamata õppimist, et avastada struktuuri ja anomaaliaid tohututes meditsiinilistes andmehulkades, ning seejärel juhendatud õppimist, et seda avastatud struktuuri kliiniliselt interpreteerida ja rakendada. See mitmekesisus annab TI-le tohutu paindlikkuse ja võimekuse lahendada väga erinevaid ülesandeid. Selles õpikus keskendume eelkõige **juhendatud õppele**, sest selle põhimõtete mõistmine on kogu masinõppe loogika mõistmise alus. Kui need põhimehhanismid on selged, on lihtne edasi liikuda ka teistesse suundadesse – näiteks **enesejuhendatud õppesse (ingl k *self-supervised learning*)** või **tugevdusõppesse**, mida kasutatakse robootikas ja autonoomsetes süsteemides.

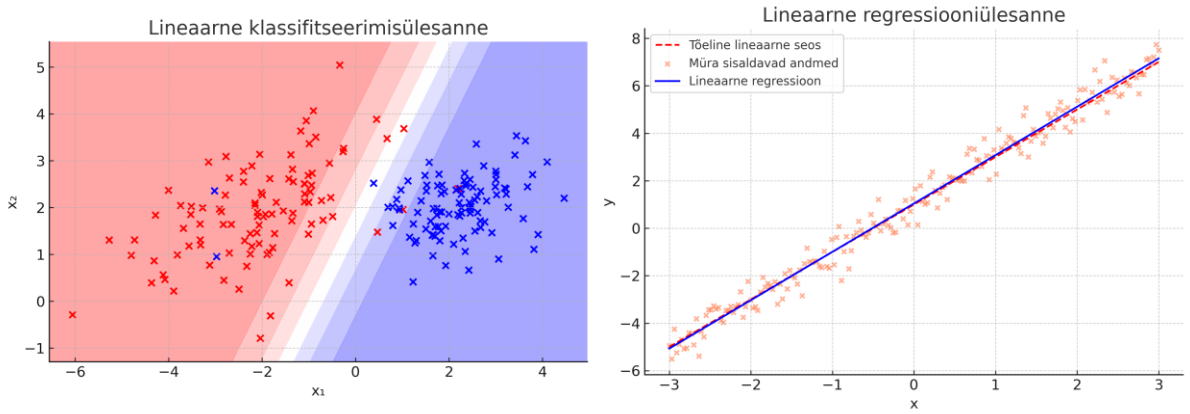
Pärast masinõppe tüüpide tutvustamist on oluline mõista ka **ülesannete liigitust**. Enamik praktilisi TI-mudeleid on loodud kas **klassifitseerimis-** või **regressiooniülesande** lahendamiseks.

Tabel 12.1. Ülesannete liigitus

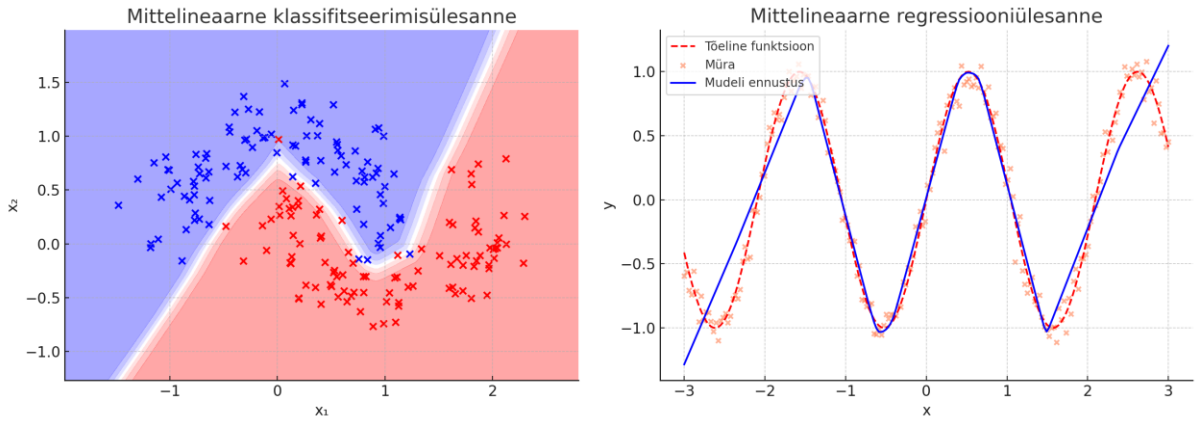
Ülesande tüüp	Väljund	Näited tervishoiust
Klassifitseerimine	Diskreetne (jah/ei, klassi tuvastamine)	Diagnoos (kas pilt on melanoom või mitte?)
Regressioon	Pidev (arvuline väärtus)	Prognoos (nt vereglükoosi tase 3 h pärast)

Peale ülesande tüübi (klassifitseerimine või regressioon) on oluline vahe, kas mudel püüab leida **lineaarseid** või **mittelineaarseid** seoseid sisendi ja väljundi vahel. Lihtsustatud juhtudel piisab sirgjoonelisest eraldusjoonest või trendist, nagu näha joonisel 12.4, kus nii klassifitseerimis- kui regressiooniülesanded on lahendatavad lineaarse mudeliga. Meditsiinis võib lineaarne seos ilmneda näiteks patsiendi vanuse ja teatud riskiskoori vahel, kus risk kasvab vanuse suurenedes enam-vähem ühtlases tempos.

Enamik tegelikke tervishoiuandmeid on aga **mittelineaarsed** – andmetevaheline seos ei ole sirgjooneline ja seda ei ole võimalik kirjeldada lihtsa lineaarse mudeliga. Nagu näha joonisel 12.5, ei piisa enam sirgjoonest – vaja on paindlikumaid algoritme, mis suudavad eristada mittelineaarseid otsustuspiire või järgida kõveraid sõltuvusi. Näiteks meditsiinilistes pildiandmetes (MRT, KT, histopatoloogia) võivad kasvaja piirid olla ebakorrapärased ja keerukate kontuuridega. Mittelineaarsus avaldub ka ajas muutuva kliinilise informatsiooni puhul, näiteks veresuhkru kõikumises, mis sõltub korraga mitmest tegurist (toitumine, stress, füüsiline aktiivsus, insuliin) ega allu lineaarsele seosele.



Joonis 12.4. Lineaarne andmestik (E. Valla, 2025).



Joonis 12.5. Mittelineaarne andmestik (E. Valla, 2025).

Masinõppemudelite treenimisel jagatakse kogu andmestik tavaliselt kaheks või kolmeks osaks, et tagada mudeli võimekus teha usaldusväärseid ennustusi ka uute, seni nägemata andmete puhul.

- **Treeningandmed** (ingl k *training data*) on andmestiku suurim osa (tavaliselt 60–80%), mida kasutatakse mudeli **õpetamiseks**. Mudel „vaatab" neid andmeid, tuvastab neis mustrid ja kohandab oma parameetreid, et minimeerida viga. See on justkui õpiku materjal, mille põhjal õpilane õpib.
- **Testandmed** (ingl k *test data*) on eraldi hoitud andmestiku osa, mida **mudel ei ole õppeprotsessi käigus kunagi näinud**. Seda kasutatakse mudeli lõpliku tulemuslikkuse hindamiseks pärast treenimise lõppu. See on justkui lõpueksam, mis testib õpilase tegelikku võimekust uute ja tundmatute probleemidega toime tulla, teisisõnu mudeli üldistamisvõimet ja ennustustäpsust.

Tulemuslikkus (ingl k *performance*) viitab sellele, kui hästi mudel täidab oma ülesannet – tavaliselt ennustamist. Tulemuslikkust hinnatakse mitme **mõõdiku** abil.

Täpsus (ingl k *accuracy*) on kõige lihtsam mõõdik: see on õigesti klassifitseeritud näidete osakaal kõikide näidete hulgas.

$$\text{Täpsus} = \frac{\text{õigesti klassifitseeritud vaatluspunktid}}{\text{kõik vaatluspunktid}}$$

Miks see ei ole alati hea mõõdik? Kui andmestik ei ole tasakaalustatud (nt 95% inimestest on terved ja 5% haiged), võib mudel ennustada kõigi puhul „terve" ja saavutada 95% täpsuse. Kuigi täpsus on suur, ei ole mudel tegelikult õppinud haigeid inimesi tuvastama (ta on 100% halb haigete tuvastamisel). Seetõttu vajame keerukamaid mõõdikuid.

Tundlikkus (ingl k *sensitivity* ehk *recall***)** ja **spetsiifilisus** (ingl k *specificity*) on eriti olulised mõõdikud diagnostikas. Need keskenduvad mudeli võimele tuvastada iga klassi.

- Tundlikkus mõõdab, kui suure osa positiivse klassi näiteid (nt haigeid inimesi) kõigist tegelikult positiivsetest näidetest suudab mudel tuvastada. See vastab küsimusele: „Kui suure osa haigetest me üles leiame?"

$$\text{Tundlikkus} = \frac{\text{tõesed positiivsed}}{\text{tõesed positiivsed} + \text{valenegatiivsed}}$$

- Spetsiifilisus mõõdab, kui suure osa negatiivse klassi näiteid (nt terveid inimesi) kõigist tegelikult negatiivsetest näidetest suudab mudel tuvastada. See vastab küsimusele: „Kui suure osa tervetest me õigesti tuvastame?"

$$\text{Spetsiifilisus} = \frac{\text{tõesed negatiivsed}}{\text{tõesed negatiivsed} + \text{valepositiivsed}}$$

Sama tähtis on mõista, **kuidas mudel sellise tulemuseni jõuab** ja **kas mudel suudab edukalt üldistada ka uutele andmetele**. Siin tulevad mängu mudeli keerukus ja õppimisvõime.

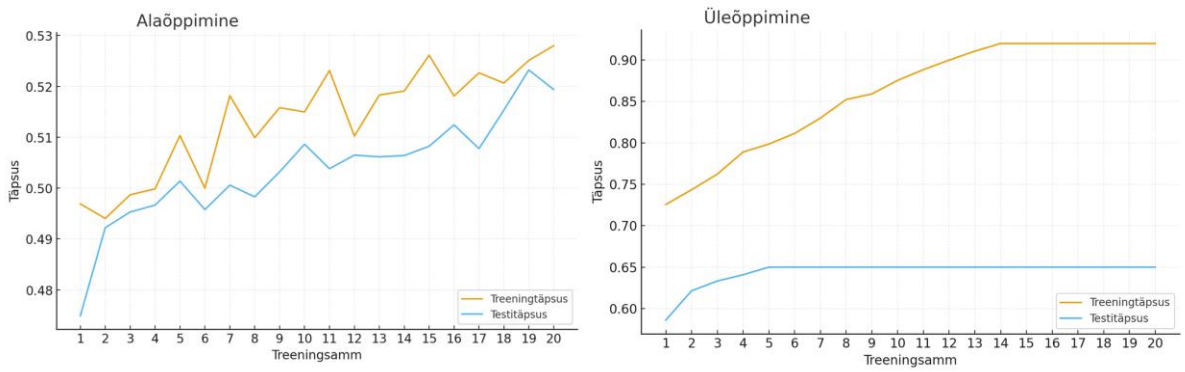
Ideaalsel mudelil on hea tasakaal **keerukuse** (piisavalt detailne, et mustrid ära õppida) ja **üldistamisvõime** (suuteline töötama uute andmetega) vahel. **Hälbimine** sellest tasakaalust viib kahe peamise probleemini: **alaõppimine** ja **üleõppimine**.

Alaõppimine või alasobitamine (ingl k *underfitting*) toimub siis, kui mudel on **liiga lihtne**, et õppida andmetes peituvaid olulisi mustreid. See juhtub tavaliselt, kui mudelil on liiga vähe parameetreid või

kasutatakse liiga tugevat lihtsustust. **Alaõppinud mudelil on madal tulemuslikkus** nii treening- kui ka testandmetel (väike tundlikkus, spetsiifilisus ja täpsus).

Üleõppimine või ülesobitamine (ingl k *overfitting*) toimub siis, kui mudel on **liiga keeruline** ja püüab treeningandmeid liiga täpselt meelde jätta, kaasa arvatud juhuslik müra ja anomaaliad. Mudel kaotab seetõttu **üldistusvõime**. **Üleõppinud mudelil on väga kõrge tulemuslikkus** treeningandmetel, kuid **madal tulemuslikkus** testandmetel. Teisisõnu, õpilane õpib eksamiküsimused ja vastused pähe, kuid ei suuda väljaspool neid uutele küsimustele vastata ning põrub eksamil.

Ala- ja üleõppimist saab treenimise ajal ära tunda, võrreldes mudeli täpsust treening- ja testandmetel: kui mõlema täpsus on väike, on tegu alaõppimisega, kui treeningtäpsus on oluliselt suurem kui testitäpsus, on tegu üleõppimisega (vt joonis 12.6).

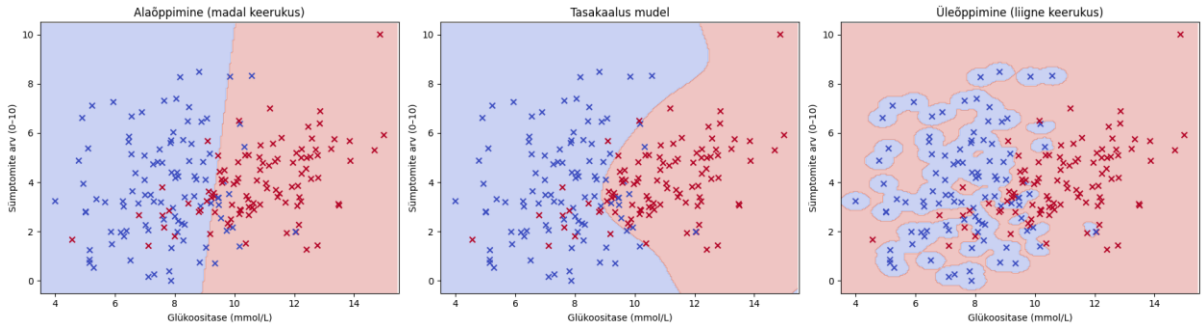


Joonis 12.6. Ala- ja üleõppimise treenimise kõverad (E. Valla, 2025).

Hea mudel on piisavalt paindlik, et haarata andmetes olulisi mustreid, kuid mitte nii paindlik, et järgida kõiki juhuslikke kõikumisi. Sellisel mudelil

- on mõõdukas keerukus,
- saavutatakse hea tulemuslikkus nii treening- kui ka testandmetel,
- suudetakse teha üldistusi uute patsientide ja olukordade kohta.

Joonis 12.7 illustreerib kolme erineva keerukustasemega klassifitseerimismudeli võimet ennustada, kas patsiendil esineb diabeediriski, kasutades kahte tunnust: glükoositase veres (*x*-telg, mmol/l) ja sümptomite arv (*y*-telg, skaalal 0...10). Punased punktid tähistavad riskiga patsiente, sinised riskita isikuid. Mudelite ülesanne on leida eraldusjoon (otsustuspiir), mis eristab kahte klassi parimal võimalikul viisil. Vasakpoolne graafik kujutab alaõppinud mudelit. See mudel on lineaarne ja liiga lihtne, et arvestada andmestikus peituvat keerukust. Tulemuseks on sirgjooneline otsustuspiir, mis ignoreerib olulisi erinevusi klasside vahel. Selline mudel alahindab sümptomite ja glükoositaseme koosmõju, seega ebaõnnestub sageli nii diabeediriski tuvastamine kui ka selle välistamine. Keskmine graafik kujutab tasakaalus mudelit. Otsustuspiir järgib mõõdukalt glükoosi ja sümptomite põhjal tekkivat mustrit, säilitades üldistusvõime. Mudel võtab arvesse olulisi mustreid, ent ei liialda nendega. Mudel oskab eristada tähenduslikku variatiivsust juhuslikust mürast. Tulemuseks on usaldusväärne üldistus uutele patsientidele. Parempoolne graafik esitab üleõppinud mudelit, mis sobitab end agressiivselt treeningandmete struktuuriga. Iga väike kõrvalekalle, olgu see juhuslik mõõteviga või erandlik patsient, sunnib mudelit keeruliselt „vingerdama“ otsustuspiiri. Visuaalselt on näha ebaloomulikult keeruline eraldusjoon. Kuigi mudel saavutab peaaegu täiusliku täpsuse olemasoleval andmestikul, ei suuda see enam üldistada.



Joonis 12.7. Kolme erineva keerukusega klassifitseerimismudeli otsustuspiirid glükoositaseme (x-telg, mmol/l) ja sümptomite arvu (y-telg, skaalal 0...10) alusel. Alaõppinud mudel (vasakul) ignoreerib keerukust ega suuda klasse eristada; tasakaalus mudel (keskel) leiab usaldusväärse otsustuspiiri; üleõppinud mudel (paremal) sobitab end liigse detailsusega treeningandmetele, kaotades üldistusvõime ***(E. Valla, 2025)****.*

Sellised probleemid toovad esile vajaduse mudelite mõtestatud valiku ja õige keerukusastme järele. Edasi vaatleme, kuidas statistilised mudelid pakuvad raamistiku mudelite lihtsustamiseks ja selgitamiseks, luues aluse masinõppe meetodite arengule.

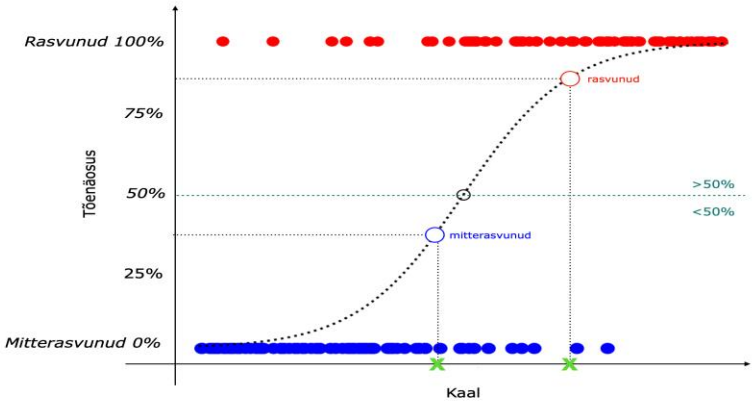
12.1.2 Statistilised masinõppemudelid

Iga TI rakendus ei vaja tingimata keerulisi sügavaid närvivõrke. Paljud igapäevased otsustustoe tööriistad tervishoius põhinevad endiselt lihtsamatel ja hästi tõlgendatavatel mudelitel, mis sobivad suurepäraselt väiksemate andmehulkade ja korrastatud (struktureeritud) andmetega tööks. Need mudelid pakuvad tihti kiiremaid, selgemaid ja vähem ressursinõudlikke lahendusi võrreldes sügavõppega.

Selles peatükis süveneme ühte klassikalisse masinõppemudelisse, **logistilisse regressiooni**, et näidata, kuidas TI saab aidata meil teha otsuseid, kus vastus on „jah“ või „ei“.

12.1.2.1 Logistiline regressioon: kuidas ennustada JAH või EI?

Kujutage ette olukorda, kus arst soovib hinnata, kas patsiendil on suurem risk olla **rasvunud** või **mitterasvunud**. Selle hindamiseks ei piisa tihti ainult ühest näitajast nagu kaal. Reaalsuses võtab mudel arvesse mitut **tunnust (sisendparameetrit)** – näiteks patsiendi kaal, pikkus, vanus, sugu, toitumisharjumused, kehaline aktiivsus. Mudeli eesmärk on nende tunnuste põhjal välja anda **ennustus**, kas patsient kuulub klassi „rasvunud“ (mille võime koodis tähistada näiteks väärtusega 1) või klassi „mitterasvunud“ (väärtus 0).



Joonis 12.8. Rasvumise klassifitseerimine logistilise regressiooni näitel (E. Valla, 2025).

Meil on hulk varasemaid andmepunkte: iga patsiendi puhul teame tema kaalu ja teisi tunnuseid ning teavet selle kohta, kas ta on tegelikult rasvunud või mitte. Joonisel 12.8 on need andmed visualiseeritud lihtsustatud kujul, kus keskendume ainult ühele tunnusele – kaalule. Sinised X-märgid tähistavad mitterasvunud patsiente ja punased punktid rasvunud patsiente. Vertikaaltelg näitab kategooriat („Rasvunud“ või „Mitterasvunud“), horisontaaltelg aga patsiendi kaalu. Klassikalise programmeerimisega oleks keeruline kirjutada kindlat reeglit (nt „kui kaal on üle X kg JA vanus üle Y aasta JA teeb Z korda nädalas trenni, siis on rasvunud“), sest tunnused kattuvad ja iga inimene on erinev. Mida teeb masinõpe? Ta otsib andmetest mustreid just nende erinevate tunnuste põhjal.

Oluline märkus visualiseerimise kohta: inimesena suudame me graafiliselt visualiseerida andmeid efektiivselt kuni kolmes mõõtmes (näiteks *X*-, *Y*- ja *Z*-teljel). See tähendab, et me saame hõlpsasti näidata seoseid nt kaalu, pikkuse ja rasvumise vahel. Kuid TI-mudelid, nagu logistiline regressioon, ei ole piiratud meie visuaalse tajuga. Nad suudavad arvesse võtta ja töödelda sadu või isegi tuhandeid erinevaid tunnuseid korraga (näiteks lisaks kaalule ja pikkusele veel vanuse, soo, vererõhu, kolesteroolitaseme, geneetilised markerid, elustiili harjumused jpm).

Kuidas aga arvuti otsustab, kuhu tõmmata piir rasvunute ja mitterasvunute vahele? Masinõppemudelid püüavad leida andmeruumist nn **otsustuspiiri**. See on hüpoteetiline joon (või kõrgemates dimensioonides tasapind), mis eraldab eri klassidesse kuuluvaid andmepunkte. Eesmärk on leida selline piir, mis jaotaks andmed kõige paremini, nii et mudel suudaks uute, seni nägemata andmete puhul õigesti ennustada, millisesse klassi need kuuluvad. Logistilise regressiooni puhul on see otsustuspiir **sirgjooneline**. Esimene osa mudelist toimib sarnaselt **lineaarse regressiooniga**, kus arvutatakse sisendmuutujate põhjal lineaarne kombinatsioon:

$$z = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n .$$

See annab igale andmepunktile ühe arvulise väärtuse $z \in (-\infty; \infty)$, mis võib olla ükskõik kui suur või väike. Kuid sellist väärtust ei saa veel tõlgendada tõenäosusena. Siin tuleb mängu logistilise regressiooni eripära – **sigmoidfunktsioon:**

$$\hat{y} = \frac{1}{1 + e^{z}}$$

Sigmoid „painutab“ sirgjoonelise tulemuse S-kujuliseks kõveraks ja teisendab selle väärtuseks vahemikus **0...1**. Just see omadus teeb sigmoidist ideaalse tööriista tõenäosuste ennustamiseks, sest tõenäosus ongi alati arv 0 ja 1 vahel (0% kuni 100%). Just sigmoidi tõttu on **logistiline regressioon** (Hosmer et al., 2013) küll nime poolest regressioon, kuid **sisuliselt klassifikatsioonimudel** – see ei prognoosi pidevat väärtust, vaid otsustab, kummas klassis andmepunkt asub (nt tõenäosus > 0,5 → „rasvunud“, < 0,5 → „mitterasvunud“).

Joonisel 12.8 on illustreeritud, kuidas logistiline regressioon suudab andmete põhjal ennustada, kas inimene kuulub klassi „rasvunud“ või „mitterasvunud“. Sinised ja punased punktid kujutavad varasemaid andmepunkte, mille pealt mudel õppis – need on andmed, mille puhul me juba teame, kas inimene oli rasvunud või mitte. **Sinised punktid** tähistavad „mitterasvunuid“ (klass 0). **Punased punktid** tähistavad „rasvunuid“ (klass 1).

Nende kahe klassi andmepunktide põhjal otsib mudel parima **S-kujulise joone (sigmoidi)**, mis kirjeldab tõenäosust, et inimene kuulub rasvunute hulka. Sigmoidjoon näitab, kuidas tõenäosus „rasvunud“ olla kasvab koos kaaluga. Kui mudel on treenitud, teisisõnu, joon on andmetele sobitatud (ingl k *the line is fitted*), saab seda kasutada uute patsientide puhul prognoosimiseks. Joonisel märgivad **rohelised X-id** patsiendi kaalu, mille põhjal mudel arvutab tõenäosuse rasvunud olemiseks.

- Kui punkt jääb **alla 50% joone**, liigitatakse patsient klassi „mitterasvunud“.
- Kui punkt jääb **üle 50% joone**, liigitatakse ta klassi „rasvunud“.

Igal masinõppealgoritmil on omad tugevused ja piirangud, logistilise regressiooni korral on need esitatud tabelis 12.2.

Tabel 12.2 Logistilise regressiooni tugevused ja piirangud

Tugevused	Piirangud
Lihtne ja läbipaistev Mudelit on suhteliselt lihtne mõista ja selle tulemusi tõlgendada, mis on meditsiinis usalduse loomiseks ülitähtis. Me saame aru, millised tegurid (nt kaal) ja kui tugevalt tõenäosust mõjutavad.	**Lineaarsuse eeldus** Logistiline regressioon püüab klasse eraldada **sirgjoonega** (või kõrgemates dimensioonides tasapinnaga), isegi kui ennustatav tõenäosuse kõver on S-kujuline. See eeldab, et andmed on lineaarselt eraldatavad, mis ei pruugi olla piisav, kui mustrid on väga keerukad ja mittelineaarsed.
Kiire ja lihtne arvutada Ei vaja tohutut arvutusvõimsust ja töötab hästi ka väiksemate andmehulkadega	**Ei sobi keerulistele andmetele** Piltide või heli analüüsiks, kus mustrid on ülimalt keerulised ja abstraktsed, on see mudel ebapiisav.
Hea alguspunkt Sobib suurepäraselt paljudes lihtsamates klassifitseerimisülesannetes, kus on kaks võimalikku tulemust.	

12.1.2.2 Otsustuspuu: samm-sammult otsuseni jõudmine

Kui logistiline regressioon annab meile tõenäosuse, siis **otsustuspuu** (Breiman et al., 1984) on otsekohene ja pakub vastuseid samm-sammult, justkui vastaks järjestikustele jah/ei küsimustele. Kujutage ette, et arst soovib hinnata, kas patsiendil on teatud haigus, näiteks diabeet ehk suhkurtõbi tuginedes erinevatele näitajatele.

Otsustuspuu joonis avaneb siit.

Joonis 12.9. *Otsustuspuu.*

Mis on otsustuspuu? See on justkui teekaart või vooskeem, mis algab ühe küsimusega ja suunab meid edasi uute küsimuste juurde, kuni jõuame lõpliku otsuseni. Iga haru on reegel ja iga „leht“ (lõpp-punkt) on ennustus või klassifikatsioon. Vaatame joonisel 12.9 olevat otsustuspuud, mis on treenitud meie näidisandmetega, kus patsiendi riski määramiseks kasutati mitut tunnust: vanus, kehamassiindeks (KMI), liikumise tunnid nädalas, suitsetamise staatus ja süstoolne vererõhk. Puu eesmärk on jagada patsiendid kahte riskiklassi: „madal risk“ (oranž) ja „kõrge risk“ (sinine). Iga ristkülik joonisel on sõlm ja sisaldab olulist infot:

- ⇒ tingimus: reegel, mille alusel andmed jagunevad (nt „vanus <= 62,5“);
- ⇒ *gini*: see arv näitab sõlme „puhtust“ ehk seda, kui segane klass selles punktis veel on. Mida lähemal 0-le, seda puhtam on sõlm (st kõik punktid kuuluvad samasse klassi);
- ⇒ *samples*: mitu andmepunkti (patsienti) selles sõlmes on;
- ⇒ *value* näitab, mitu andmepunkti kuulub igasse klassi (nt [93, 107] tähendab 90 „madala riskiga“ ja 107 „kõrge riskiga“ patsienti);
- ⇒ *class*: domineeriv klass selles sõlmes.

Graafikut loetakse ülevalt alla, alustades ülemisest kastist, mida nimetatakse juureks. Seal esitatakse esimene otsustav küsimus, mis jagab andmed kaheks. Näiteks antud graafikus on esimeseks küsimuseks „Vanus ≤ 62,5“. See tähendab, et mudel kontrollib, kas patsiendi vanus on väiksem või võrdne 62,5 aastaga. Kui vastus on tõene (*true*), liigume vasakule harusse, kui väär (*false*), liigume paremale. Järgmiseks täpsustab otsustuspuu vasakus harus riski näiteks kehamassiindeksi põhjal: „KMI ≤ 23,551“. Kui patsiendi KMI on alla selle piiri, liigume jälle vasakule ja risk jääb madalaks. Kui KMI on suurem kui 23,551, liigume paremale ja risk suureneb. Paremal ülemisel harus hinnatakse esmalt samuti KMI väärtust („KMI ≤ 25,972“), mis näitab, et ka kõrge vanuse korral on kehamassiindeks oluline riskitegur. Järgmine otsus võib sõltuda näiteks vererõhust („Vererõhk_süstoolne ≤ 148,775“), mis aitab riski veelgi täpsemalt klassifitseerida.

Graafiku värvid aitavad tõlgendamist lihtsustada: oranžid kastid viitavad madala ja sinised kõrge riskiga rühmadele. Mida tumedam toon, seda kindlam on mudel oma otsuses, sest vastavas harus domineerib üks klass selgelt.

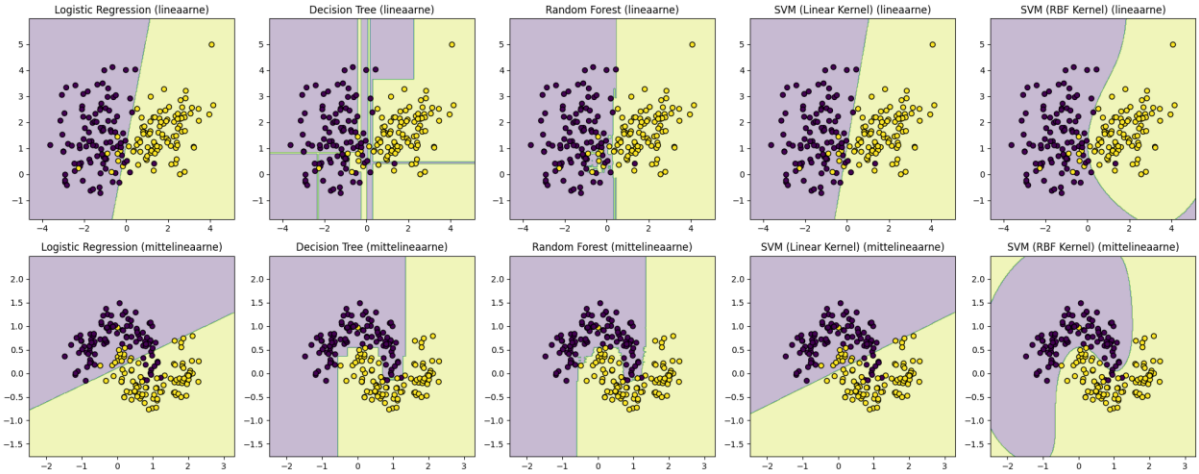
Kokkuvõttes töötab otsustuspuu nagu struktureeritud jah/ei-küsimuste jada. Otsustuspuu leiab kõige olulisemad reeglid ja nende järjestuse, mis aitavad patsiendid riskikategooriatesse jagada. Nagu näha, ei ole see lihtsalt „kui KMI on üle X, siis on risk“, vaid arvestab mitme teguri koosmõju. Arst saab jälgida igat sammu ja näha, miks TI sellise ennustuse tegi, mis on tervishoius äärmiselt oluline usalduse ja selguse seisukohalt. Otsustuspuu tugevused ja piirangud on esitatud tabelis 12.3.

Tabel 12.3 Otsustuspuu tugevused ja piirangud

Tugevused	Piirangud
Lihtsasti tõlgendatav ja visualiseeritav Arstid ja muud spetsialistid saavad puu loogikat kergesti jälgida ja mõista, milliste kriteeriumide alusel mudel otsuse tegi. See suurendab usaldust TI-tööriista vastu.	**Üksiku puu tundlikkus ja üleõppimise risk** Üksik otsustuspuu võib olla tundlik andmete väikestele muutustele ja „üle õppida“ ehk muutuda liiga spetsiifiliseks treeningandmetele. See võib viia halva üldistusvõimeni uutel andmetel. Selle piirangu ületamiseks kasutatakse sageli **otsustusmetsa** (ingl k *random forest*) meetodit. See tähendab sadade või tuhandete otsustuspuude kombineerimist, kus iga puu õpib pisut erineva andmevalimiga. Lõpptulemus saadakse kõigi puude „hääletamise“ teel, mis teeb mudeli palju robustsemaks ja täpsemaks.
Kasulik reeglite ja mustrite avastamiseks Otsustuspuud võivad paljastada seoseid, mida inimesed ei pruugi märgata – näiteks millised tunnused eristavad erinevaid haigusrühmi.	**Vähem sobiv pidevate seoste modelleerimiseks** Otsustuspuu lõikab andmeid diskreetselt, mistõttu ei pruugi ta tabada pidevaid seoseid.
Töötab nii numbriliste kui ka kategooriliste andmetega Ei vaja keerukat eeltöötlust, sobib hästi segatüüpi sisenditele.	
Ei eelda andmete normaaljaotust Võib hästi toimida ka ebakorrapäraste või ebaühtlaselt jaotunud andmete korral.	

Millal eelistada klassikalist masinõpet sügavõppele? Ja miks on valik oluline?

Nagu nägime logistilise regressiooni ja otsustuspuu näitel, on klassikaline masinõpe võimas tööriist paljude probleemide lahendamiseks. Kuid need on vaid kaks näidet paljudest mudelitest, mis klassikalise masinõppe alla kuuluvad. Mudeli valik on oluline, sest igal mudelil on oma tugevused ja nõrkused ning nad sobivad eri tüüpi andmete ja probleemide jaoks.



Joonis 12.10. Erinevate klassikaliste masinõppemudelite (logistiline regressioon, otsustuspuu, otsustusmets, tugivektorklassifitseerija) otsustuspiirid kahes eri olukorras. Ülemine rida: lineaarselt eristatav andmestik. Alumine rida: mittelineaarne andmestik (moons) (E. Valla, 2025).

Joonis 12.10 visualiseerib seda olulist erinevust, näidates, kuidas erinevad klassikalised masinõppemudelid jaotavad andmeid kahes eri olukorras.

- **Ülemine rida:** andmestik on **lineaarse struktuuriga**, kahte klassi (lillad ja kollased täpid) on võimalik eraldada sirgjoonega. Mõned mudelid (nt logistiline regressioon) leiavadki sirgjoonelise piiri, kuid paindlikumad mudelid (nt otsustuspuud, ansamblid või mittelineaarsed kernelid) võivad kujundada keerukama kuju ka lineaarse andmestiku puhul.
- **Alumine rida:** andmestik on **mittelineaarne (nt *two moons*)**, klassid on põimunud ja vajavad eraldamiseks kõverjoonelist piiri.

Vaatame lähemalt, mida joonis meile ütleb.

- **Logistiline regressioon:** nagu juba teame, leiab see lineaarse otsustuspiiri. Ülemisel real töötab see hästi, sest andmed on lineaarselt eraldatavad. Alumisel, mittelineaarsel andmestikul on aga selgelt näha, et sirge joon ei suuda kumeraid „kuusid“ eraldada – mudel jääb hätta.
- **Otsustuspuu** jagab andmeruumi ristkülikuteks. Ülemisel real luuakse sirgjooned. Alumisel real loob see ka ristkülikuid, püüdes keerulist kuju mitme sirge jaotusega jäljendada. Kuigi see suudab paremini kohaneda kui logistiline regressioon, võib tulemus olla üsna „nurgeline“.
- **Otsustusmets** (Breiman, 2001) kombineerib otsustuspuude tarkust. Nagu jooniselt näha, suudab see luua palju sujuvama ja keerukama otsustuspiiri kui üksik otsustuspuu, eriti mittelineaarse andmestiku puhul. See näitab kollektiivse tarkuse eelist. See on **ansambelmeetod** – kombineerib sadu või tuhandeid otsustuspuid, millest igaüks töötab pisut erineva andmevalimiga. Lõpptulemus saadakse kõigi puude „hääletamise“ teel. **Miks „ansambel“?** Nii nagu orkestris annab iga pillimees oma osa ja tervik kõlab paremini, annavad ka siin paljud iseseisvalt treenitud mudelid (puud) oma panuse ja nende kollektiivne tarkus on usaldusväärsem kui ühegi üksiku puu ennustus. See aitab oluliselt vähendada üksiku puu piiranguid, nagu tundlikkus müra suhtes ja üleõppimine.

- **Tugivektorklassifitseerija** (ingl k *support vector machine, SVM*) (Cortes & Vapnik, 1995) on võimas mudel, mis otsib parimat võimalikku piiri kahe klassi vahel.
    - **Lineaarse kerneliga (üleval):** sarnaselt logistilise regressiooniga leiab see lineaarse piiri;
    - **RBF kerneliga (all):** siin tuleb esile tugivektorklassifitseerija eriline võimekus. Kasutades nn **kerneli trikki**, suudab see leida ka **mittelineaarseid otsustuspiire**. Nagu näha joonisel 12.10, eraldab see kõverjooneline piir „kuud" ideaalselt, mis näitab, et tugivektorklassifitseerija suudab ilma närvivõrke kasutamata toime tulla ka keerukate, mittelineaarsete andmemustritega.

Peale nende mudelite, mida näeme joonisel 12.10, on olemas ka **uusima põlvkonna klassikalised masinõppemudelid**, mis on tuntud oma erakordse jõudluse poolest keerukate andmete ja mittelineaarsete seoste korral. Üks neist on **XGBoost (**ingl k ***extreme gradient boosting*****)** (Chen & Guestrin, 2016). See on samuti ansambelmeetod, mis kombineerib mitme lihtsama mudeli (tavaliselt otsustuspuude) tulemusi, et saavutada suurem täpsus. XGBoost on leidnud laialdast kasutust mitmes valdkonnas, sealhulgas meditsiinis, tänu oma võimele käsitleda suuri andmehulki, puuduvaid väärtusi ja keerulisi mustreid, sageli ületades teisi klassikalisi algoritme.

Lõppkokkuvõttes ei ole üks lähenemine tingimata teisest parem. Õige tööriista valimine sõltub olukorrast, andmete tüübist ja oodatavatest tulemustest. Tervishoius on selgus ja tõlgendatavus sageli elulise tähtsusega, mis annab klassikalistele masinõppe meetoditele endiselt olulise rolli. Lühikokkuvõte klassikalise masinaõppe ja sügavõppe eelistest on esitatud tabelis 12.4.

Tabel 12.4. Klassikalise masinõppe ja sügavõppe eelised

Klassikaline masinõpe sobib hästi	Sügavõpe on sobivam
Väiksemad andmestikud	Suured, rikkalikud andmekogud (nt pildid)
Struktureeritud andmed (nt tabelid)	Struktureerimata andmed (nt tekst, pildid)
Vajadus tõlgendatavuse järele	Vajadus väga keerukate mustrite leidmiseks

Millised mudelid on veel võimelised mittelineaarsete andmestikega töötama ja kuidas nad seda teevad, vaatame järgmises osas, kus sukeldume sügavõppe põnevasse maailma.

12.1.3 Sügavõppemeetodid

12.1.3.1 Tehisnärvivõrgud

Oletame, et meil on patsiendi kohta suur hulk meditsiinilisi andmeid: vanus, vererõhk, südame löögisagedus, vereanalüüside tulemused, piltdiagnostika näitajad jpm. Arstide eesmärk võib olla nende andmete põhjal otsustada, kas patsiendil esineb südameveresoonkonna haiguse risk. Selline otsustamine tähendab sisuliselt, et meil on **sisendandmed x** ja soovitud **väljund y** (diagnoos). Me ei tea aga täpset matemaatilist seost nende vahel.

Kujutleme, et meil on sadade nuppudega masin (vt joonis 12.11). Iga nupu väärtus on nagu **kaal** närvivõrgus – ta määrab, kui palju üks sisend mõjutab väljundit. Kui nupud on õigesti paika keeratud, siis suudab masin sisendandmete põhjal anda õige otsuse (diagnoosi).

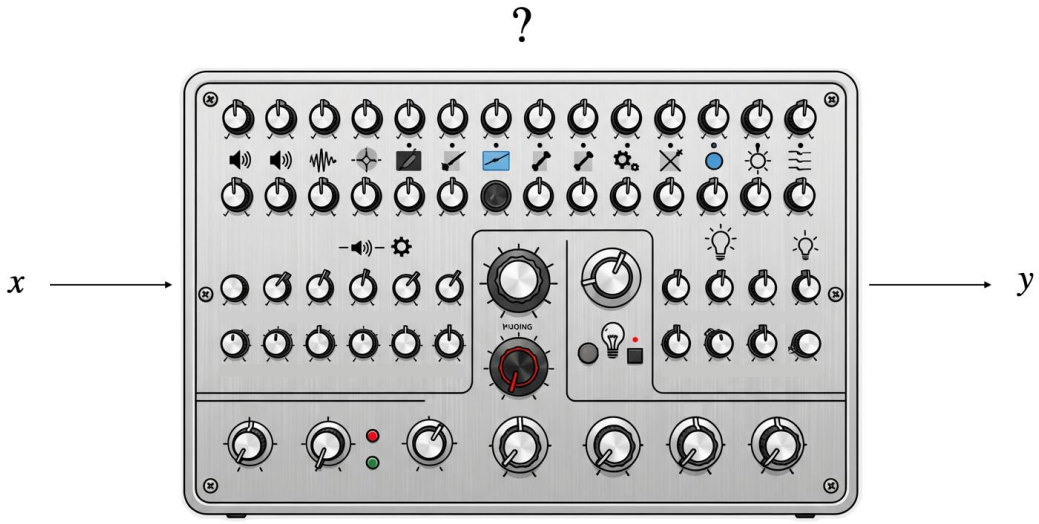
- **Sisend (x)** – patsiendi andmed, mis lähevad masinasse.

- **Väljund (y)** – masin ennustab, kas patsiendil on risk kõrge või madal.
- **Nupud (kaalud)** – süsteemi sees kohandatavad parameetrid, mida õpitakse andmete põhjal.

Alguses on nupud keeratud juhuslikult ja masin töötab halvasti. Aga kui me laseme tal õppida paljudest näidetest (patsientidest, kelle tegelik diagnoos on teada), siis samm-sammult reguleeritakse nupud nii, et masin oskab sisendi põhjal õiget väljundit ennustada. Seda visualiseerib järgnev GIF.

Närvivõrgu õppimise protsess GIF

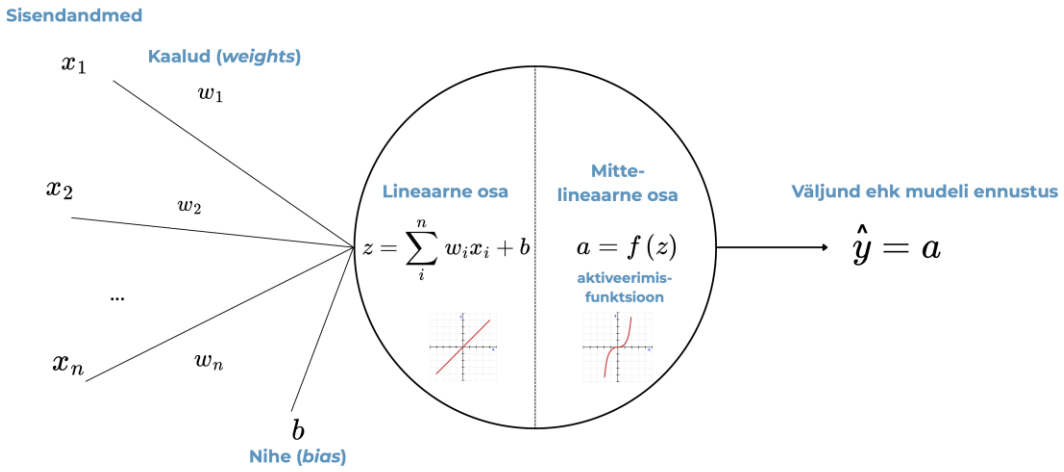
Alguses annab mudel juhusliku kuju, kuid näidete põhjal õppides reguleerib ta sammhaaval oma parameetreid, kuni ennustus järgib tegelikku mustrit.



Joonis 12.11. Närvivõrkude metafoor: tuunimisnuppudega masin (E. Valla, 2025).

Iga nupp ei muuda väärtust suvaliselt, vaid võtab vastu sisendi, korrutab selle oma kaaluga, liidab teistega kokku ja laseb läbi „otsustuskriteeriumi“. Kui panna selliseid lihtsaid elemente kokku kümneid või tuhandeid, siis tekib keerukas süsteem, mis suudab õppida mustreid, mida inimene käsitsi kirjeldada ei oska.

Tehisnärvivõrk koosneb väga paljudest väikestest arvutusüksustest, mida nimetatakse neuroniteks. Üksik neuron võtab sisendandmeid (näiteks mõõteväärtusi), korrutab need kaaludega, liidab tulemused kokku, lisab nihke (*bias*) ja teisendab tulemuse aktiveerimisfunktsiooniga. Sellist mudelit nimetatakse **pertseptroniks** ja see vastab ühe lihtsa tehisneuroni tööloogikale. Pertseptron on tehisnärvivõrkude ehituskivi, suuremad mudelid koosnevad sadadest või miljonitest pertseptronitest, mis on ühendatud kihtide kaupa. Pertseptroni ehitust on kujutatud joonisel 12.12.



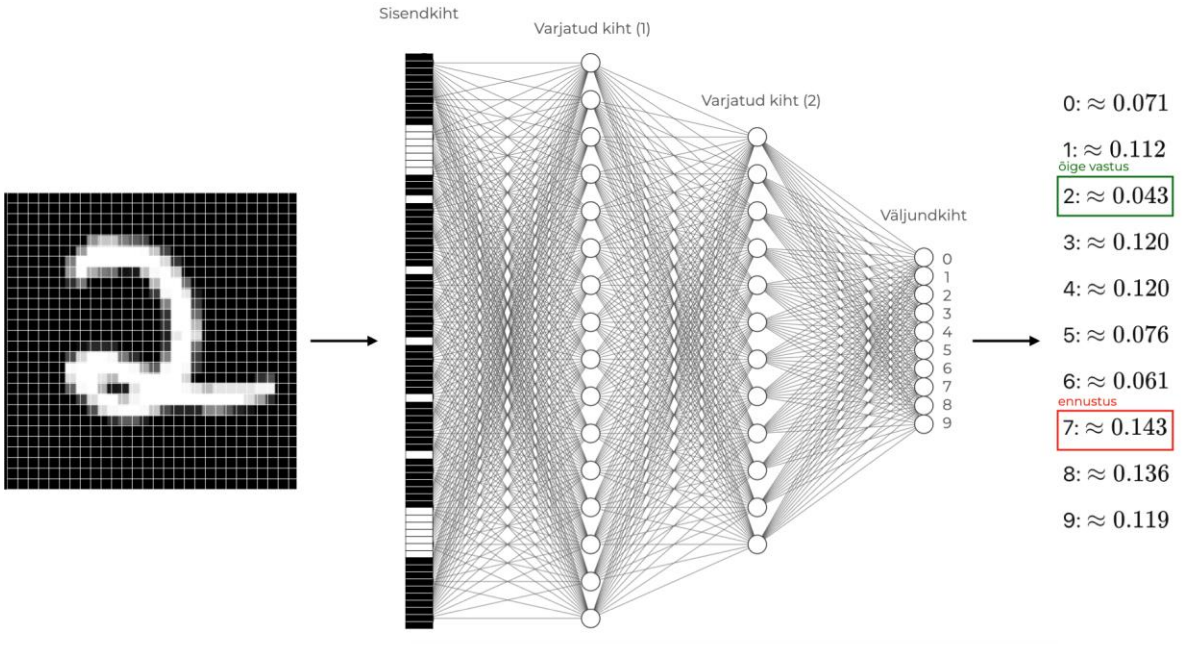
Joonis 12.12. Pertseptron ehk üks neuron (E. Valla, 2025).

Lineaarne osa tähendab sisendi kaalutud summeerimist. Iga sisendandme väärtus korrutatakse vastava kaaluga, mis näitab selle sisendi olulisust. Näiteks kui südame löögisagedusel on diagnoosimisel suurem mõju kui patsiendi vanusel, siis omistatakse sellele suurem kaal. Nii modelleerib neuron lihtsat seost kujul „kui oluline mingi tunnus on“. Kui kõik need kaalutud sisendid kokku liita (ja lisada nihe), saamegi lineaarse kombinatsiooni. **Mittelineaarne osa**: saadud summale rakendatakse **aktiveerimisfunktsioon**, mis otsustab, kas neuron „aktiveerub“ või mitte. Näiteks võib see funktsioon olla nagu valguslüliti: kui signaal on piisavalt tugev, läheb tuli põlema. Just see teine samm teeb neuronist huvitava elemendi. Aktiveerimisfunktsioone on mitut tüüpi ja need määravad, kuidas neuron reageerib sisendile. Levinumad näited:

- **sigmoidfunktsiooni** kasutatakse tihti binaarse klassifikatsiooni korral. Sigmoidi väärtus jääb vahemikku 0...1, mistõttu tõlgendatakse seda sageli tõenäosusena. Sigmoidi tutvustati ka logistilise regressiooni osas (vt 1.2.1 Logistiline regressioon: kuidas ennustada JAH või EI?);
- **tanh-funktsioon (hüperboolne tangensfunktsioon)** sarnaneb sigmoidiga, kuid selle väärtused jäävad vahemikku –1...1. See funktsioon on sümmeetriline nulli suhtes ja sobib hästi juhtudel, kus on vaja modelleerida ka negatiivseid väärtusi;
- **mittenegatiivne lineaarfunktsioon (**ingl k ***rectified linear unit, ReLU*****)** (Nair & Hinton, 2010) on kõige populaarsem aktiveerimisfunktsioon sügavates närvivõrkudes. ReLU annab väärtuse 0 kõikide negatiivsete sisendite korral ja jätab positiivsed väärtused muutmata. ReLU on arvutuslikult lihtne ja aitab lahendada probleemi, kus sigmoidi või tanh-funktsiooni kasutamisel mudel hakkab õppimise käigus „aeglustuma“.

Kui meil oleks ainult lineaarne summa, ei saaks võrk õppida keerulisi mustreid. Mittelineaarsus ongi see, mis lubab võrguosadel koos töötades leida mustreid, mida inimene ei oskaks otseselt kirjeldada.

Alustame lihtsast näitest, oletame, et meie eesmärk on tuvastada, kas pildil on number 2. Siis me vajame mudelit, mis suudab selliseid pilte mõista ja õigesti liigitada. Selleks õpetatakse närvivõrku varem sildistatud andmete abil – meil on suur hulk pildinäiteid, mille kohta on juba teada, millist numbrit (0...9) need kujutavad.



Joonis 12.13. Närvivõrgu treenimise visualiseerimine (E. Valla, 2025).

Treeningu alguses ei oska võrk veel midagi mõistlikku öelda. Selle sees olevad kaalud, mis määravad, kuidas sisendandmed väljundit mõjutavad, on alguses määratud juhuslikult, seega on ka võrgu esialgne vastus suvaline. Kui vaatame sellise treenimata võrgu väljundit, saame iga numbri kohta väärtuse, mida võib tõlgendada kui selle klassi tõenäosust, ning kõik väärtused moodustavad kokku ühe tõenäosusjaotuse. Joonisel 12.13 on näha, et kõrgeim väärtus tuleb vale numbri kohta – näiteks klass 7, kuigi pildil oli number 2. See tähendab, et võrk eksib. Meie eesmärk on nüüd teda treenida, st kohandada tema sisemisi parameetreid nii, et järgmistel kordadel oleks vastus täpsem: soovime, et võrgu väljundis oleks klassi 2 tõenäosus suur ja kõigi teiste oma väike. Teisisõnu, me peame **minimeerima vea** võrgu ennustuse ja tegeliku vastuse vahel.

Aga kuidas võrk saab teada, millised kaalud olid „valesti keeratud“ ja kuidas neid parandada? Siin tulebki mängu treenimine (optimeerimine). See tähendab, et võrgu sees arvutatakse samm-sammult, millises suunas peaks iga kaalu muutma, et **kaofunktsiooni** kaudu defineeritud viga väheneks. Valime kaofunktsiooni $L(y, \hat{y})$, mis mõõdab, kui kaugel on mudeli väljund tegelikust vastusest.

Näited:

- regressioon: $L = \frac{1}{2}(y - \hat{y})^2$;
- mitme klassi liigitus: ristsentroopia $L = -\sum_i y \log \hat{y}$.

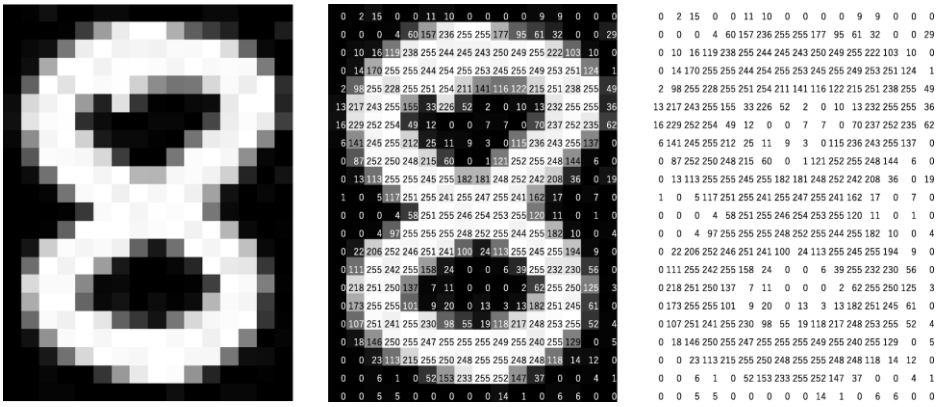
Tuletame meelde keskkooli matemaatikast: kui tahame leida funktsiooni **miinumumpunkti**, siis vaatame selle tuletist. Tuletis näitab, kas funktsioon kasvab või kahaneb ja kui järsult see muutub. Närvivõrgu puhul teeme sama, aga palju mõõtmeid korraga – iga kaalu ja nihke kohta. Tuletis (ehk gradient) ütleb meile, mis suunas ja kui palju peaks seda konkreetset parameetrit muutma, et viga väheneks. Seejärel võtame kõik parameetrid ja liigutame neid korraga **negatiivse gradiendi suunas** – see on suund, mis vähendab viga kõige kiiremini, kui samm on sama pikk. Seda samm-sammult korrates liigume järjest lähemale punktile, kus viga on minimaalne. Kõigi nende parameetrite (kaalude ja nihete) osakaal ehk mõju vea tekkele arvutatakse **tagasilevi (**ingl k ***backpropagation*****)** algoritmi abil.

See on närvivõrkudes põhiline õppemehhanism, mis teeb võimalikuks kaofunktsiooni tuletiste leidmise iga parameetri suhtes ja võimaldab seeläbi võrku treenida.

Seni kirjeldatud mudel on **pärilevivõrk**, kus informatsioon liigub kiht-kihilt edasi ühes suunas ja iga neuron on ühendatud eelmise kihi kõigi neuronitega. Selline arhitektuur sobib hästi tabelandmete ja lihtsamate ülesannete jaoks, kuid piltandmete puhul on see ebatõhus. Järgnevalt liigume edasi keerukamate andmetüüpide juurde, mille hulka kuuluvad ka piltandmed. Piltide töötlemisel ei ole mõistlik neid lihtsalt üheks pikaks vektoriks teisendada, sest sel viisil kaob ära nende ruumiline struktuur ja seosed naaberpikslite vahel. See raskendab võrgu õppimist ja muudab mustrite avastamise keerukamaks. Järgmises osas selgitame, kuidas piltide struktuurset infot säilitada ja miks kujunesid kujutistega töötamisel oluliseks standardiks konvolutsioonilised närvivõrgud.

12.1.3.2 Konvolutsioonilised närvivõrgud

Kuidas arvuti näeb pilti? Arvuti jaoks ei ole pilt midagi muud kui **maatriks**, kus iga element vastab piksli väärtusele. Mustvalge pildi puhul on iga piksli väärtus vahemikus 0...255, kus 0 tähistab musta ja 255 valget (vt joonis 12.14). Värvilise pildi puhul on iga piksli kohta kolm väärtust – punase, rohelise ja sinise kanali intensiivsus.



Joonis 12.14. Number 8 pikslite väärtustega, väljavõte MNIST andmebaasist (LeCun et al., 1998) (Yamashita et al., 2018)

Kui paneme kõik piksli väärtused ühte pikka vektorisse (vt joonis 12.13), kaotame olulise info – **piksli asukoha ja ruumilise seose teiste pikslitega**. Naaberpikslite muster ja paiknemine on aga pildi mõistmiseks otsustava tähtsusega. Tavaline sügav närvivõrk ei kasuta ruumilist infot ära – iga sisendi väärtus on võrgu jaoks lihtsalt üks arv. Et pildist kasulikku infot kätte saada, peame leidma viisi, kuidas

1) säilitada **ruumiline struktuur** (ingl k *spatial information*);

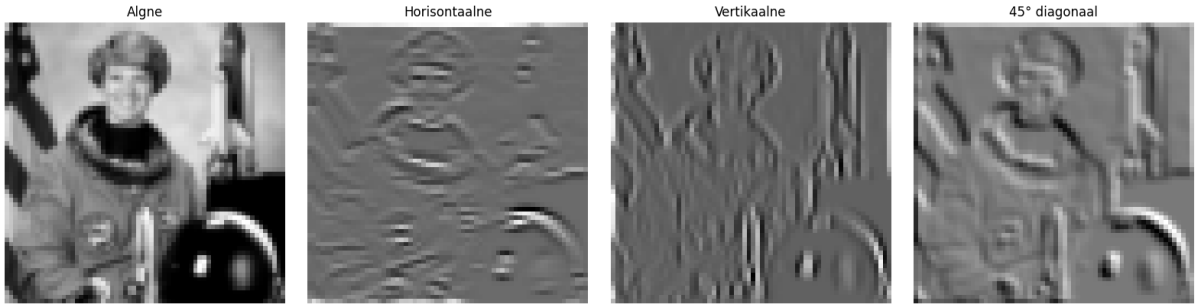
2) vähendada parameetrite hulka (vältida liiga suuri mälunõudeid ja üleõppimist);

3) muuta võrgu töö tolerantseks väikeste nihete, pööramiste ja valgustuse muutuste suhtes.

Reaalses maailmas ei näe sama objekt iga kord pildil välja täpselt ühtemoodi. Näiteks kass võib olla teises asendis, teistpidi pööratud, teise valgusega, jne. Inimese jaoks on see ikka kass, aga lihtne algoritm läheks sellistes olukordades kiiresti segadusse. Seetõttu peab närvivõrk olema invariantne väikeste nihete, pööramiste ja valgustuse muutuste suhtes ning oskama eraldada olulisi tunnuseid sõltumata sellest, kuidas objekt täpselt kaadrisse sattus. Näited joonisel 12.15.

Joonis 12.15. Variatsioonid klassis „kass“ (E. Valla, 2025).

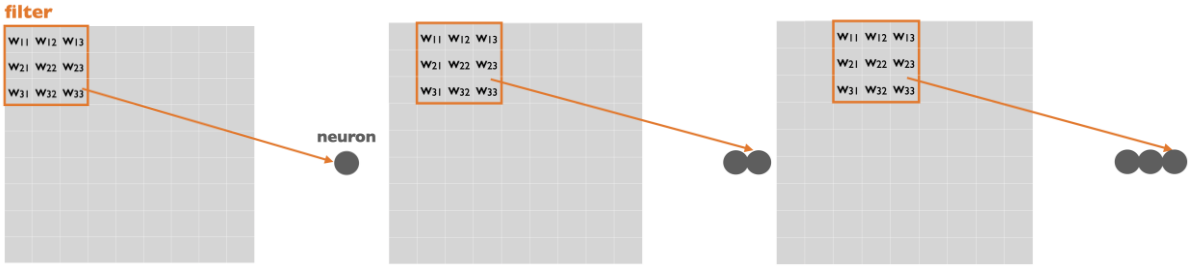
Konvolutsioonilised närvivõrgud (ingl k ***convolutional neural networks*****, CNN)** on loodud selliste muutustega paremini toime tulema ning õppima sama objekti ära tundma olenemata selle asukohast ja kujust pildil. Erinevalt pärilevivõrkudest ei teisendata sisendpilti kohe üheks vektoriks, vaid seda töödeldakse konvolutsioonikihtide kaudu, mis analüüsivad pilti lokaalselt piirkondade kaupa. CNN-i töö põhineb **filtritel** ehk **tunnuste detektoritel**, mis liiguvad üle sisendpildi ja otsivad korduvaid mustreid. Treeningu käigus õpivad filtrid eristama esmalt madala taseme tunnuseid, nagu servad ja nurgad, ning sügavamates kihtides ka keerukamaid mustreid, nagu tekstuurid, objektiosad ja lõpuks isegi terved objektid. Filtri väärtused määravad, millist mustrit (serv, tekstuur, joon) see suudab tuvastada. Joonisel 12.16 on näidatud, kuidas erinevad konvolutsioonifiltrid muudavad sisendpilti, et esile tõsta olulisi tunnuseid. Vasakul on algne pilt, mille detailsus on CNN-mudeli sisendiks sobivalt vähendatud. Järgmised kolm pilti näitavad filtri tulemust pärast konvolutsiooni rakendamist. Horisontaalservade filter toob esile pildi horisontaalseid jooni ja kontraste, näiteks riietuse ja tausta piirjooni. Vertikaalservade filter rõhutab üles-alla-suunalisi servi, näiteks keha ja taustal oleva lipu



Joonis 12.16. Filtrite mõju sisendpildile (E. Valla, 2025).

kontuure. 45-kraadise diagonaalnurga filter tuvastab kaldjooni ja diagonaalseid mustreid. Sellised madala taseme tunnused on aluseks keerukamate struktuuride tuvastamisele, sest sügavamates CNN-i kihtides kombineeritakse need mustrid kõrgema taseme esitusteks, kuni mudel suudab eristada terveid objekte.

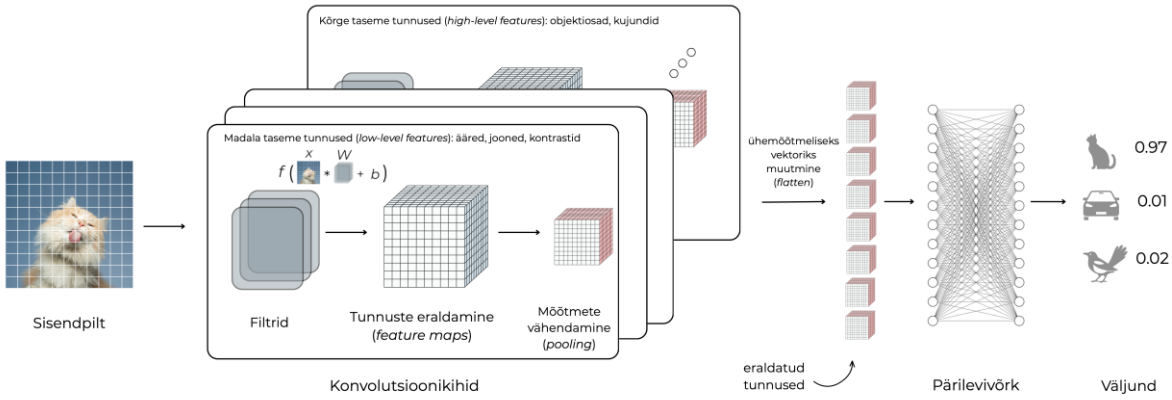
Joonisel 12.17 on kujutatud, kuidas filter liigub üle pildi ja filtri iga nihutuse järel tekib väljundis uus neuron.



Joonis 12.17. Filtri nihutamisel tekib neuronitest tunnuste kogum (ingl k feature map) järgmise kihi jaoks (E. Valla, 2025).

Filtri liigutamine üle pildi tekitab **tunnusekaardi** (ingl k *feature map*), kus iga väärtus näitab, kui tugevalt see muster vastavas asukohas esines. Kuna filtreid on mitu, tekib sisendpildist mitu tunnusekaarti, millest igaüks esindab erinevat mustrit või tunnust. Filtri ja sisendväärtuste (esialgu pikslivääruste) vahel teostatakse **konvolutsiooni operatsioon**: filtri väärtused korrutatakse vastavate sisendi väärtustega ja tulemused liidetakse, saades arvu, mis kirjeldab mustri sobivust selles asukohas. Pärast konvolutsioonioperatsiooni kasutatakse tihti **mõõtmete vähendamist (**ingl k ***pooling*)**, et vähendada andmemahtu ja muuta mudel stabiilsemaks väikeste nihete ja müra suhtes. CNN-i lõpposas muudetakse tunnusekaardid ühemõõtmeliseks vektoriks (*flatten*) ja saadetakse edasi **pärilevivõrku**, mis teeb lõpliku otsuse näiteks klassifitseerimisülesandes. Joonisel 12.18 on kujutatud konvolutsioonilise närvivõrgu töövoog pildi klassifitseerimisel. See lähenemine

- hoiab alles pildi ruumilise struktuuri;
- vähendab parameetrite hulka võrreldes tiheda ühendusega kihtidega;
- aitab võrgul paremini üldistada.



Joonis 12.18. Konvolutsioonilise närvivõrgu arhitektuur (E. Valla, 2025).

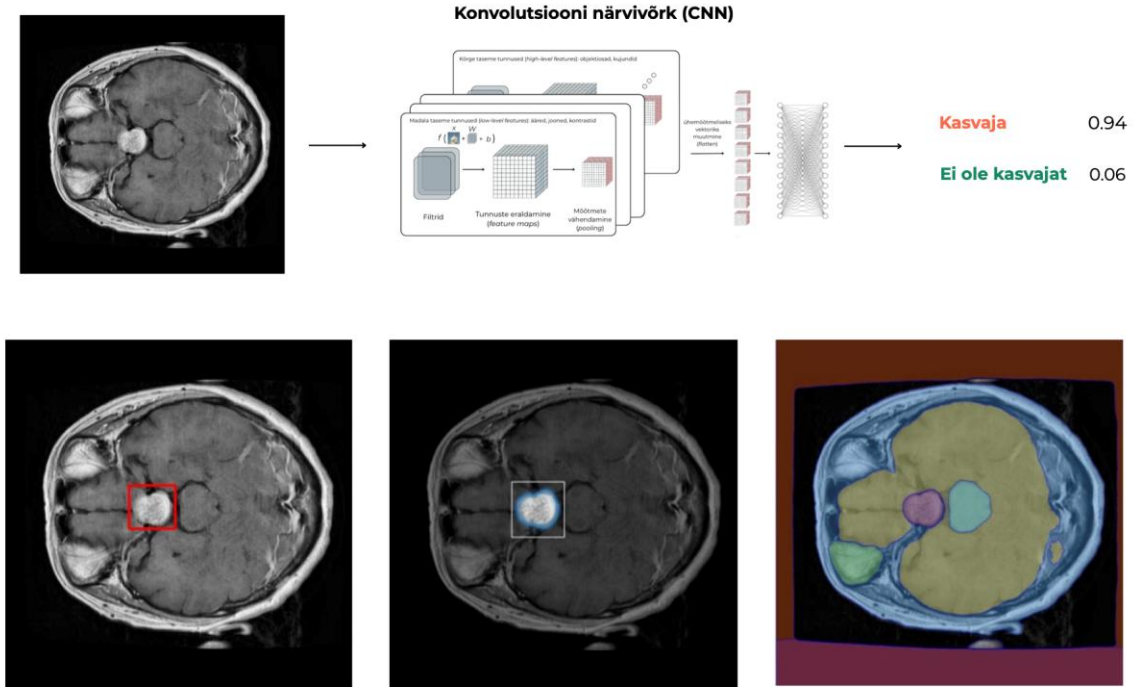
CNN-id on olnud kõige olulisem pilditöötluse ja arvutinägemise läbimurre. Nende edukus põhineb just automaatsel filtrite õppimisel, mis võimaldavad tuvastada järjest keerukamaid mustreid – alustades servadest ja kontrastist kuni tervikliku objekti või haigusliku mustrini välja. Need algoritmid ei asenda arsti, kuid nad suudavad analüüsida sadu või tuhandeid pilte kiiresti ja järjekindlalt, pakkudes arsti tööks hindamatut abivahendit. Seetõttu on CNN-id tänapäeval paljude meditsiiniliste piltdiagnostikasüsteemide nurgakivi.

Kuna arstidel on vaja teha pildiandmete põhjal mitmesuguseid otsuseid, alates kiirest sõeluuringust kuni detailse ravi planeerimiseni, on pildituvastuses mitu spetsiifilist ülesannet. Iga ülesanne vastab

erinevale diagnostilisele vajadusele lihtsast olemasolu tuvastamisest (klassifitseerimine) keeruka anatoomilise detailide kaardistamiseni (segmenteerimine). Joonisel 12.19 on kujutatud meditsiinilise pilditöötluse erinevad ülesanded, mida sügavõppemudeleid kasutades lahendatakse. Ülemises osas on näidatud konvolutsioonilise närvivõrgu (CNN) tööpõhimõte, kus mudel analüüsib sisendina antud MRT peaaju pilti ja teeb esmase otsuse, **kas kasvaja on olemas** või mitte. Sellist ülesannet nimetatakse **pildiklassifitseerimiseks**, kus vastus on jah/ei või „klass_1, klass_2, klass_3, ..., klass_n“ tüüpi prognoos.

Alumises reas on kujutatud järgmised kolm pildituvastuse taset, mis võimaldavad liikuda lihtsast otsusest detailsema diagnoosini.

1. **Objekti asukoha tuvastamine** (ingl k *object detection*) – pilt vastab küsimusele: **„Kus on kasvaja?“** Mudel leiab kasvaja asukoha pildil ja märgib selle piirava kastiga (ingl k *bounding box*). Seda kasutatakse sageli sõeluuringutes ja kiiretes esialgsetes analüüsides.
2. **Ühe klassi segmenteerimine** (ingl k *segmentation*) – pilt vastab küsimusele: **„Mis kujuga on kasvaja?“** Mudel eraldab kasvaja ülejäänud ajust piksli täpsusega, luues kasvaja täpse maski. See ülesanne on vajalik kasvaja mahu arvutamiseks, muutuste jälgimiseks ajas ja kirurgilise planeerimise toetamiseks.
3. **Mitmeklassiline segmenteerimine** – pilt vastab küsimusele: **„Millised teised struktuurid pildil asuvad?“** Mudel eraldab peale kasvaja ka muud anatoomilised struktuurid (nt erinevad koetüübid, elundid, õõnsused) ja määrab igaühele eraldi värvi ehk klassi. Seda kasutatakse ravi kavandamisel, kiiritusravi doseerimisel ja anatoomiliste piiride täpseks määramiseks.



*Joonis 12.19. Pildituvastuse liigid. Ülemine osa määrab tõenäosuse, kas on kasvaja, alumine rida aga visualiseerib vasakult paremale: kus on kasvaja (piiritlev kast), mis kujuga see on (täpne kontuur) ning millised osad on veel pildil (terviklik anatoomiline kaart) (*E. Valla*, 2025).*

Nende ülesannete lahendamiseks on välja kujundatud ka spetsialiseeritud mudelid. Näiteks **YOLO** (Redmon et al., 2015) ja selle täiendatud versioonide mudeliperekond on üks enam kasutatud

lähenemisi nii meditsiiniliste kui ka tavapiltide **objektituvastuseks**. **U-Net** (Ronneberger et al., 2015) ja selle täiendatud versioonid on saanud **biomeditsiinilise segmenteerimise** vaikimisi standardiks tänu oma täpsusele ja sobivusele piiratud meditsiiniliste andmete korral. Klassifitseerimisülesannetes kasutatakse **ResNeti** (He et al., 2015) või **EfficientNeti** (Tan & Le, 2019) tüüpi mudeleid, mis sobivad hästi näiteks haigusastmete eristamiseks ja histopatoloogia klassifitseerimiseks.

Kuigi siiani keskendusime staatilistele piltidele, on CNN-idvõimelised töötlema ka videot – mis on oma olemuselt lihtsalt kiiresti vahetuvate piltide jada. See võimekus on loonud aluse kaugtaastusravile (ingl k *telerehabilitation*).

Kasutades meetodit nimega poosituvastus (ingl k *pose estimation*), suudab TI-mudel tuvastada videopildilt patsiendi liigesed (nt küünarnukk, põlved, õlad) ja jälgida nende liikumist reaalajas. See võimaldab patsiendil sooritada terapeutilisi harjutusi kodus arvuti- või telefonikaamera ees, samal ajal kui algoritm kontrollib liigutuste amplituudi ja kiirust. Kui patsient teeb harjutust valesti, annab süsteem kohest tagasisidet, aidates vältida vigastusi ja tagada ravi efektiivsuse ka ilma terapeudi füüsilise kohaloluta.

Poosituvastuse demo rehabilitatsioonis

Selles vallas on teerajajaks **OpenPose** (Cao et al., 2019), mis suutis esimesena täpselt kaardistada mitme inimese kehaasendeid keerukates olukordades. Tänapäeva kodustes lahendustes on aga standardiks saanud Google'i arendatud **MediaPipe** (Lugaresi et al., 2019), mis on optimeeritud töötama tõhusalt just nutitelefonides ja tahvelarvutites.

Kokkuvõttes on CNN-id tõestanud end võimekate ja usaldusväärsete mudelitena meditsiiniliste piltide analüüsimisel ning neist on saanud sügavõppe standard pildiandmete töötlemisel. Kuigi CNN-id on väga tugevad ruumiliste mustrite tuvastamisel, on nende põhijõud koondunud lokaalsele infotöötlusele. See tähendab, et nad ei ole alati parimad järjestikuste või tekstiliste andmete puhul, kus on oluline mõista konteksti ja pikaajalisi seoseid.

Järgmises osas liigume pildiandmetest edasi keeleandmete juurde ja tutvume uue närvivõrkude arhitektuuriga – **transformeriga**, mis on põhjalikult muutnud seda, kuidas masinad teksti töötlevad ja keelt mõistavad.

12.1.3.3 Transformer-tüüpi närvivõrgud

Kujutage ette, et te loete raamatut. Te ei vaata iga sõna eraldi, vaid püüate mõista tähendust kontekstis – vahel meenutate juba varem loetud peatükke, vahel keskendute ainult ühele olulisele lausele. Sama oskust püüavad matkida ka tänapäeva võimsaimad tehisnärvivõrgud, mida nimetatakse **transformeriteks**.

Nimi ChatGPT, mida te kindlasti kuulnud olete, tuleb tegelikult inglise keelsetest *sõnadest generative pre-trained transformer*. See tähendab lihtsas keeles: suur keelemudel (ingl k *large language model*, LLM), mis on tohutul hulgal tekstil õppinud ja suudab „sõnu" genereerida. Aga transformerid ei piirdu ainult vestlusrobotitega. Sama tehnoloogiat kasutatakse kõne tekstiks muutmisel, automaattõlkes, piltide loomisel ja ka muusika genereerimisel.

2017. aastal ilmus teadusartikkel pealkirjaga „Attention is All You Need" (Vaswani et al., 2017), millest sai tänapäevase TI murranguline verstapost. Artikli autorid näitasid, et tähelepanumehhanismiga saab luua arhitektuuri, mis on parem ja kiirem kui varasemad masintõlke ja keeletöötluse mudelid. See uus arhitektuur sai nimeks **transformer**. Mis teeb transformerid eriliseks?

- Paralleelarvutused.

- Parem pikkade sõltuvuste mõistmine.
- Positsiooniline kodeering (ingl k *positional encoding*).

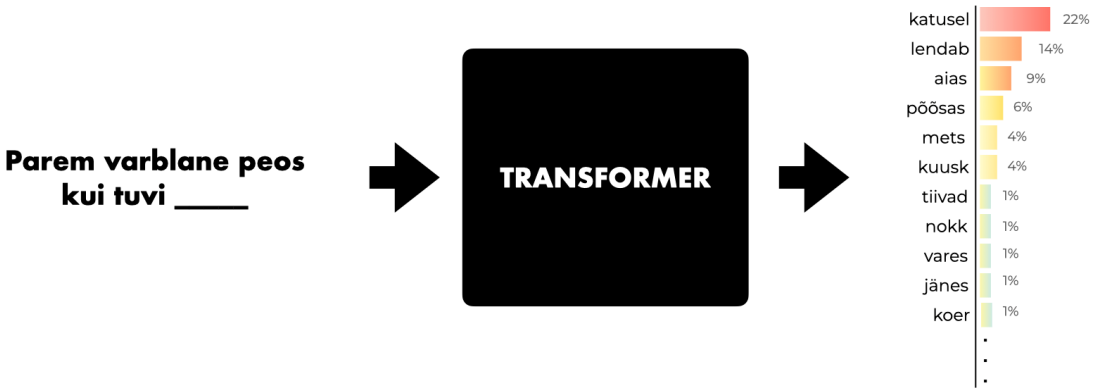
Erinevalt varasematest mudelitest ei töötle transformer andmeid järjestikku, sõnahaaval, vaid suudab analüüsida kogu sisendlauset korraga. Selleks peab mudel siiski teadma sõnade järjekorda. Kuna transformer ei loe lauset vasakult paremale nagu inimene, lisatakse igale sõnale positsiooniline kodeering. See annab mudelile info selle kohta, kus iga sõna lauses paikneb, näiteks kas tegemist on lause alguse või lõpuga ja millised sõnad on omavahel ühendatud. Positsiooniline kodeering on kui koordinaatsüsteem, mis lisab tekstile struktuuri.

Peamine uuendus transformerite puhul on tähelepanumehhanism ehk ***self-attention***. Kui CNN-id õpivad otsima olulisi tunnuseid filtrite abil, siis transformerid kasutavad tähelepanu juhtimiseks kolme vektorit: *query* (Q), *key* (K) ja *value* (V). *Query* küsib, mistaoline info on hetkel oluline, *key* aitab otsida teisi sõnu, mis on sellega seotud, ja *value* kannab sisu, mis edasi antakse. Mõnes mõttes võib tähelepanumehhanismi võrrelda otsustusfiltritega, kuid erinevalt CNN-i filtritest ei otsi need ruumilisi mustreid, vaid tähenduslikke seoseid. Treeningu käigus õpitakse, kui tugevalt peaks üks sõna pöörama tähelepanu teistele sõnadele lauses. Näiteks lauses „patsient rääkis arstile, et tema vend lõpetas ravi, sest ta ei tundnud end enam halvasti“ peab mudel otsustama, kelle kohta käib sõna „ta“ – kas patsiendi, tema venna või arsti kohta. Õige tõlgendus tuleb konteksti põhjal: „ta ei tundnud end enam halvasti“ viitab loogiliselt patsiendi vennale, mitte arstile. Tähelepanumehhanism teeb sellised seosed nähtavaks, hinnates sõnade omavahelist tähenduslikku seotust isegi suurema vahemaa korral.

Transformerite generatiivne võimekus tuleneb sellest, kuidas arhitektuur on üles ehitatud. Transformer võib töötada kolmel viisil:

- ***encoder-only*** variant (nt tekstist arusaamine) ei genereeri ise midagi, vaid mõistab ja tõlgendab. Sobib näiteks klassifitseerimiseks ja info leidmiseks tekstist;
- ***decoder-only*** variant **kasutab eelnevat konteksti järgmise sõna ennustamiseks** ja seetõttu on **loomult generatiivne**;
- ***encoder–decoder*** variant suudab **nii mõista kui ka genereerida** ning seetõttu sobib tõlkimiseks ehk tekstist tekstiks teisendamiseks.

Generatiivsus tähendab lihtsustatult, et mudel ehitab väljundi samm-sammult üles, ennustades iga järgmise sõna tõenäosuse alusel. Seda on näidatud joonisel 12.20, kus mudel jätkab antud lauset, valides tõenäosuste järgi jätkusõna. Sellise oskuse omandab mudel treeningu käigus, kus talle antakse väga suur hulk tekstinäiteid ja ta peab igal sammul ennustama, milline sõna või sümbol (*token*) tuleb järgmise sisendina. Kui mudel eksib, korrigeeritakse tema sisemisi kaalusid – nii nagu CNN-id õpivad pilte, õpivad transformerid keelemustreid. Pikaajalise treeningu tulemusel omandab mudel statistilise tunnetuse keeleloogikast ja suudab järgmist sõna prognoosida mitte juhuslikult, vaid konteksti arvestades.

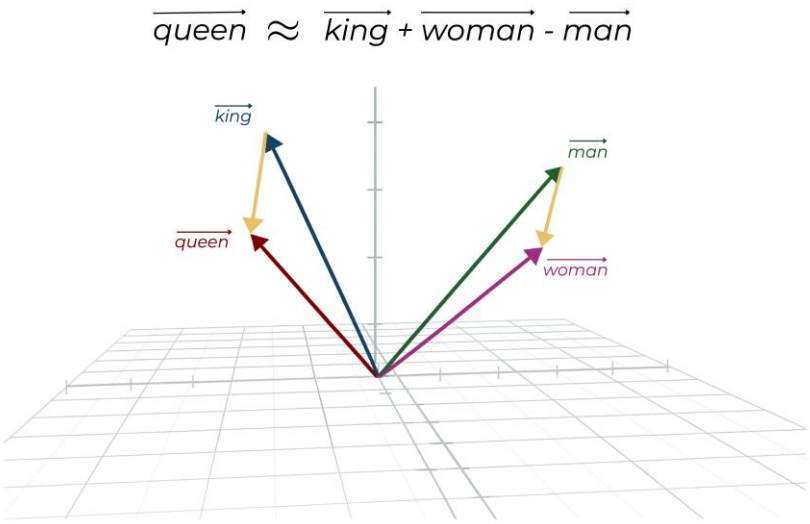


*Joonis 12.20. Transformer-tüüpi mudel ennustab järgmist sõna (täpsemalt tokenit) jadas (*E. Valla*, 2025).*

Aga kuidas transformerid „näevad“ sõnu? Piltide puhul olid sisendiks pikslid, aga tekstis? Arvuti ei tööta ju otseselt tähtede või sõnadega, vaid numbritega. Selleks jagatakse iga sõna väiksemateks ühikuteks, mida nimetatakse ***tokeniteks***. Näiteks lause „Kass jooksis kiiresti“ võib muutuda viieks tokeniks: „Kass“, „jooks“, „is“ „kiir“, „esti“. Tokeniseerimine tehakse sellisel kujul põhjusel, et **täissõnade kaupa töötamine oleks ebapraktiline**: treeningkorpus sisaldab miljoneid erinevaid sõnu, erinevaid käändeid ja isegi trükivigu, mistõttu muutuks sõnavara liiga suureks ning mudel ei suudaks harva esinevaid sõnu üldse mõista. **Teisalt ei ole ka ühe tähe kaupa tokeniseerimine efektiivne**, sest see tekitaks väga pika sisendi ja nõuaks tohutult parameetreid. Seetõttu kasutatakse vahepealset lahendust – **alamüksuspõhist tokeniseerimist**, kus sõnad jaotatakse tähenduslikeks tükkideks.

Igal tokenil on oma **numbriline esitus**. Kuid lihtsalt suvalised numbrid ei annaks mudelile mingit tähendust. Selle asemel õpib mudel iga tokeni jaoks **vektori** – reaalarvude jada, mida võib ette kujutada punktina kõrgedimensioonilises ruumis.

Mida see tähendab? Kujutage ette, et igal sõnal on oma koht suures kaardis. Lähestikku asuvad sõnad on tähenduselt sarnased, üksteisest kaugel asuvad sõnad aga erinevad. Üks kuulus näide 2013. aastal avaldatud artiklis (Mikolov et al., 2013) näitab, et kui võtta vektor „kuningas“, lahutada „mees“ ja seejärel liita „naine“, siis tulemuseks on vektor, mis asub väga lähedal sõnale „kuninganna“ (joonis 12.21). See näitab, et mudel ei õpi ainult sõnu, vaid ka nende **seoseid ja mustreid**. Selline sõnade geomeetriline kujutamine vektorruumis ongi alus, mis võimaldab transformeritel hiljem tähendusi mõista ja uusi lauseid genereerida.



Joonis 12.19. Klassikaline näide sõnavektoritest näitab, et vektor „kuningas – mees + naine“ asub lähedal sõnale *„kuninganna“ (E. Valla, 2025).*

Kui mõista, kuidas transformerid töötavad, tekib loomulikult järgmine küsimus: milliseid mudeleid on sellest arhitektuurist loodud ja miks need nii võimsaks muutusid? Kõige tuntum näide on **ChatGPT**, mis on **suur keelemudel**. Seda tüüpi mudelid on treenitud **tohutu hulga tekstide peal**, mis pärinevad avalikest allikatest, nagu raamatud, teadusartiklid, veebilehed ja vestlusandmed. Treeningu eesmärk on lihtne, kuid tõhus: mudel peab igal sammul **ennustama järgmise sõna** antud kontekstis. Just tänu sellele õppemeetodile suudavad suured keelemudelid luua sidusat teksti, vastata küsimustele ja jätkata vestlust loogiliselt.

Suurte keelemudelite võimekus tuleneb nende suurusest. Neis on **miljardeid parameetreid** – need on mudeli sisemised „nupud“, mida optimeeritakse treeningu käigus, et mudel õpiks keelemustreid. Näiteks GPT-3 sisaldab **175 miljardit parameetrit**, GPT-4 hinnanguliselt **üle 1 triljoni parameetri**. Sellise mastaabi tõttu vajavad need mudelid treenimiseks **suurt arvutusvõimsust (GPU klastreid)** ja **petabaitide ulatuses treeningandmeid**. Üks oluline tegur suurte keelemudelite kasutamisel on litsents ja ligipääs**.** Mudelid erinevad selle poolest, kas neid saab vabalt alla laadida ja lokaalselt kasutada või on juurdepääs võimalik ainult tootja kaudu. Seda eristust on tähtis mõista, sest see mõjutab mudeli kohandatavust, andmekaitset, kulusid ja sõltuvust teenusepakkujast. Nagu näha tabelis 12.5, erinevad mudelid nii parameetrite mahu kui ka avatusastme poolest.

Avatud lähtekoodiga mudel tähendab, et mudeli ülesehitus (arhitektuur) ja kaalud on avalikult kättesaadavad ning neid võib alla laadida, oma andmetega edasi treenida ja oma rakendustes kasutada. Selliste mudelite puhul on oluline litsents, mis määrab kasutusreeglid, kuid üldiselt on need paindlikud ning sobivad nii teadustööks kui ettevõtetele, kes soovivad lahendusi ise kontrollida ja arendada.

Kommertslitsentsiga mudelid on suletud ehk nende sisemisi kaalusid ja täpset ülesehitust ei avaldata. Neid saab kasutada ainult API (ingl k *application programming interface*) kaudu teenusena ja litsentsitingimused määrab tootja. Selliseid mudeleid kasutatakse sageli tööstuses, kuna need pakuvad tugevat jõudlust ja stabiilsust, kuid nende kasutus on vähem kohandatav ja sõltub tootjast.

Tabel 12.5. Suurte keelemudelite võrdlus 2024–2025

Mudel	Päritolu	Parameetrite maht (ligikaudu)	Tugevused	Avatus / litsents
GPT-4 (OpenAI, 2023)	USA	hinnanguliselt 1–1,7 triljonit	Üldotstarbeline keeleoskus, loovus, analüütiline mõtlemine	Kommertslitsents
GPT-5 (OpenAI, 2025)	USA	ametlikult avaldamata	Laiendatud loogika ja keerukamate ülesannete lahendamine	Kommertslitsents
Gemini 2.5 (Google, 2025)	USA	~1 triljon	Multimodaalsus: tekst, pilt, heli, video, kood	Kommertslitsents
Claude 3 (Anthropic, 2024)	USA	üle 200 miljardi	Väga pikk kontekst, turvalisus ja kasutajatoe dialoog	Kommertslitsents
LLaMA 3 (Meta AI, 2024)	USA	8–70 miljardit	Avatud ja kohandatav teadusele ja arendusele	Avatud lähtekoodiga
Mistral (Mistral AI, 2024)	Euroopa	7–22 miljardit	Väga tõhus, töötab ka väiksematel serveritel.	Avatud lähtekoodiga
DeepSeek V2 (DeepSeek-AI, 2024)	Hiina	67 miljardit	Tugev matemaatikas, loogikas ja koodi genereerimisel.	Avatud lähtekoodiga

Üldised keelemudelid võivad pakkuda tuge kliiniliste tekstide lugemisel ja tõlkimisel, kuid meditsiinis on vaja suuremat täpsust, domeeni-spetsiifilist sõnavara ja konteksti mõistmist. Seetõttu on kasulikud peenhäälestatud mudelid, näiteks:

- **MedGemma** (Google DeepMind, s.a.-b) – Google DeepMindi avatud lähtekoodiga, meditsiinilisel korpusel treenitud mudel, millel on tugev võime kliinilistes küsimustes ja piltide-tekstide kombineerimisel;
- **PubMedBERT** (Gu et al., 2020) – spetsiaalselt PubMedi artiklitel treenitud, et mõista teaduskeelt;
- **BioGPT** (Luo et al., 2022) – kohandatud biomeditsiini jaoks, oskab luua teaduslikult põhjendatud teksti.

Peenhäälestamine võimaldab mudelit kohandada konkreetse haigla või eriala vajadustega, näiteks vähidiagnostika raportite koostamiseks või elektrooniliste haiguslugude (EHR) struktureerimiseks.

Üks praktiline rakendus on **virtuaalne või simuleeritud arstivisiit**, mida saab kasutada nii hariduses kui ka patsiendisuhtluses.

- Näide: demo „Appoint Ready“ võimaldab TI abil simuleerida patsiendi ja arsti vestlust.
- Selline süsteem võib toetada
    - **õppimist ja koolitust** – meditsiinitudengid saavad harjutada anamneesi võtmist riskivabas keskkonnas;

- **patsiendisuhtlust** – avatar või tekstipõhine assistent selgitab protseduure, vastab korduvatele küsimustele ning pakub reaalajas tõlget, aidates ületada keelebarjääre arsti ja patsiendi vahel;
- **dokumenteerimist** – TI koostab struktureeritud kokkuvõtte (anamnees, sümptomid, soovitused), mida arst saab üle vaadata ja täiendada.

Sellised rakendused näitavad, et transformerid ei ole pelgalt tehnoloogiline uudsus, vaid neil on potentsiaal muuta tervishoiu igapäeva: vähendada arstide halduskoormust, toetada täpsemat kommunikatsiooni ja parandada patsiendikogemust.

Üks suurimaid probleeme suurte keelemudelite puhul on nähtus, mida nimetatakse **hallutsinatsiooniks** – mudel genereerib enesekindlalt vale või väljamõeldud vastuse. Oluline on mõista, et mudel ei „valeta" meelega ega varja teadmatusest tingitud ebakindlust. Vastupidi, keelemudelid on treenitud nii, et nad püüavad **alati jätkata teksti kõige tõenäolisema järgmise sõna põhjal**, isegi siis, kui sisendandmetes puudub vastus või see on ebaselge. Loomupäraselt ei ütle nad „ma ei tea", kui just eraldi ei õpetata neid seda tegema. See statistiline pimenurk on eriti ohtlik detailides. Võtame näiteks olukorra, kus tekstist selgub vaid, et patsiendi seisund muutus. Mudeli jaoks võib olla statistiliselt sama tõenäoline kirjutada, et „temperatuur tõusis", kui et see „langes", kuid meditsiinilises kontekstis on tegu vastandlike stsenaariumidega. Selline tõenäosuslik oletamine on kliinilises töös lubamatu, kuna üks vale sõna võib raviotsuse sisu täielikult muuta.

Just seetõttu on meditsiinis olulised peenhäälestatud mudelid. Nende puhul ei hinnata ainult keelelist sidusust, vaid ka täpsust ja usaldusväärsust meditsiinilise teksti mõistmisel. Lisaks klassikalisele täpsusskoorile testitakse neid meditsiinispetsiifilistes ülesannetes: sümptomite seostamine diagnoosidega, uuringute tõlgendamine, teadusartiklite refereerimine. Selline erialane peenhäälestus aitab vähendada hallutsinatsioonide riski olukordades, kus eksimine võib olla eluohtlik.

Hallutsinatsioone aitab vähendada ka allikapõhine genereerimine (ingl k *retrieval-augmented generation, RAG*) (Lewis et al., 2020) lähenemine. Selle asemel, et mudel vastaks ainult oma „mälust", otsib ta asjakohast infot kõigepealt välisest allikast (nt teadusartiklist või haigla andmebaasist) ja kasutab seda vastuse alusena. Näiteks võib RAG-põhine süsteem patsiendi küsimusele ravimite kõrvaltoimete kohta toetuda otse ravimi infolehele, mitte mudeli enda oletusele.

Vaadates tulevikku, liiguvad transformerid lihtsast tekstianalüüsist märksa keerukama **kliinilise partnerluse** suunas, kus kesksel kohal on **multimodaalsus** – võime analüüsida üheaegselt nii haiguslugu, röntgenpilte kui ka biosignaale. Et selline süsteem oleks kliiniliselt aktsepteeritav, peab see olema **selgitatav** (ingl k explainable AI, XAI), avades otsuste tagamaid ja viidates konkreetsetele tõenditele, mitte pakkuma vaid „musta kasti" vastuseid. See tehnoloogiline potentsiaal realiseerub täielikult vaid läbi **sujuva integratsiooni**, kus TI muutub haigla infosüsteemides (EHR) „nähtamatuks abiliseks", mis toetab arsti reaalajas, järgides samal ajal rangeid andmekaitse- ja Euroopa Liidu tehisintellekti määruse nõudeid.

Kokkuvõte

TI rakendamine tervishoius pakub märkimisväärseid võimalusi alates töövoogude automatiseerimisest ja diagnoositäpsuse parandamisest kuni kliiniliste otsuste toetamise ja patsiendi kaasamise suurendamiseni. Samal ajal kaasneb selle kasutuselevõtuga hulk riske. Peamised ohud on seotud andmekaitse ja privaatsusega, algoritmilise ebaõigluse ja kallutatusega, liigsest automatiseerimisest tingitud usaldusprobleemidega ning meditsiinilise vastutuse hajumisega. Suur osa praegustest lahendustest ei ole veel kliiniliselt valideeritud ja nende usaldusväärsus võib oleneda olukorrast.

Riske tuleb tunnistada, kuid neist ei saa järeldada, et TI kasutamisest tervishoius tuleks loobuda. Nagu iga meditsiinilise uuenduse puhul, tuleb hinnata riski ja tulu suhet. Riske on võimalik juhtida ja vähendada. Kõige tähtsam on parandada tervishoiutöötajate TI-kirjaoskust, mis aitab mõista tehnoloogia toimimist ja piiranguid. Teiseks tuleb soodustada kontrollitud eksperimenteerimist ja arendustööd, mis võimaldab uusi mudeleid katsetada turvalises keskkonnas ning tõenduspõhiselt. Nii saame suurendada tehnoloogia usaldusväärsust ja kasu ning samal ajal maandada sellega seotud ohte.

Kokkuvõttes ei ole küsimus selles, kas kasutada tervishoius TI-lahendusi, vaid kuidas teha seda vastutustundlikult. Läbimõeldud riskijuhtimise ja teadliku kasutuselevõtu kaudu on võimalik saavutada olukord, kus TI pakutav väärtus ületab selgelt võimalikud ohud ja toetab nii tervishoiutöötajaid, patsiente kui ka nende lähedasi.

Kasutatud kirjandus

Anthropic. (2025). Claude [suur keelemudel]. https://claude.ai

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5 32. https://doi.org/10.1023/A:1010933404324

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. Wadsworth.

Cao, Z., Hidalgo, G., Simon, T., Wei, S. E., & Sheikh, Y. (2019). OpenPose: Realtime multi-person 2D pose estimation using Part Affinity Fields. arXiv. https://arxiv.org/abs/1812.08008

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. arXiv. https://arxiv.org/abs/1603.02754

Cortes, C., & Vapnik, V. (1995). Support vector networks. Machine Learning, 20(3), 273 297. https://doi.org/10.1007/BF00994018

DeepSeek-AI. (2024). *DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model*. arXiv. https://arxiv.org/abs/2405.04434

Google DeepMind. (2025). *Gemini 2.5: Our most intelligent AI model*. https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/

Google DeepMind. (s.a-b). MedGemma. Loetud November 20, 2025, aadressil https://deepmind.google/models/gemma/medgemma/ Google DeepMind+1

Google DeepMind. (s.a.-a). AlphaFold. Loetud November 20, 2025, aadressil https://deepmind.google/science/alphafold/ Google DeepMind

Google. (2025). Gemini [suur keelemudel]. https://gemini.google.com

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., & Poon, H. (2020). Domain specific language model pretraining for biomedical natural language processing. arXiv. https://arxiv.org/abs/2007.15779 arXiv

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. arXiv. https://arxiv.org/abs/1512.03385 arXiv

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression (3rd ed.). Wiley.

LeCun, Y., Cortes, C., & Burges, C. J. C. (1998). The MNIST database of handwritten digits. Loetud November 20, 2025, aadressil http://yann.lecun.com/exdb/mnist/ SCIRP+1

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Kuttler, H., Lewis, M., Yih, W. T., Rocktaschel, T., Riedel, S., & Kiela, D. (2020). Retrieval augmented generation for knowledge intensive NLP tasks. In Advances in Neural Information Processing Systems (Vol. 33, pp. 9459-9474). Curran Associates. discovery.ucl.ac.uk+1

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., ... & Grundmann, M. (2019). MediaPipe: A framework for building perception pipelines. arXiv. https://arxiv.org/abs/1906.08172

Luo, R., Sun, L., Xia, Y., Qin, T., Zhang, S., Poon, H., & Liu, T. Y. (2022). BioGPT: Generative pre-trained transformer for biomedical text generation and mining. arXiv. https://arxiv.org/abs/2210.10341 arXiv

Meta AI. (2024). *The Llama 3 herd of models*. arXiv. https://arxiv.org/abs/2407.21783

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv. https://arxiv.org/abs/1301.3781 arXiv

Mistral AI. (2024). *Mixtral 8x22B* [Model release]. https://mistral.ai/news/mixtral-8x22b

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning* (pp. 807–814). Omnipress. https://www.cs.toronto.edu/~fritz/absps/reluICML.pdf

OpenAI. (2023). *GPT-4 technical report*. arXiv. https://arxiv.org/abs/2303.08774

OpenAI. (2025). ChatGPT [suur keelemudel]. https://chat.openai.com

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You only look once: Unified, real time object detection. arXiv. https://arxiv.org/abs/1506.02640 arXiv

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. arXiv. https://arxiv.org/abs/1505.04597 arXiv+1

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. arXiv. https://arxiv.org/abs/1905.11946 arXiv

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. arXiv. https://arxiv.org/abs/1706.03762 arXiv

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: An overview and application in radiology. Insights into Imaging, 9(4), 611-629. https://doi.org/10.1007/s13244-018-0639-9 PubMed