

DL - есејски

Wednesday, January 29, 2025

9:41 PM

- 1) Зошто во современата практика се користат **длабоки невронски мрежи наместо плитките мрежи**, и кои се клучните предности на длабоките мрежи во однос на плитките

Shallow Networks -> тоа се мрежи кои имаат мал број на скриени слоеви (1-2)

- Тоа што располагаат со мал број на скриени слоеви ги спречува да решаваат комплексни задачи

Deep Networks -> тоа се мрежи кои имаат поголем број на скриени слоеви во споредба со плитките мрежи

- Многу се поефективни во извршување на комплексни задачи затоа што работат така што комбинираат карактеристики научени во пониските слоеви и тоа ги прави многу поефективни
- Во пракса, кога се соочуваме со проблеми од областа на **препознавање говор, финансиски предвидувања или покомплексна анализа** и предвидување на податоци почесто се користи DNN
- Првично затоа што во реалниот свет сме преплавени со голем број на податоци, ни треба мрежа која добро ќе се справува со големо множество на податоци и покрај тоа што бараат повеќе ресурси

- 2) Како **forward propagation** и **backpropagation** работат во процесот на учење на невронски мрежи? Дискутирајте го значењето на **функциите за загуба и оптимизација**.

- Ова се механизми за **мапирање на влезни податоци во излезни предвидувања**. Работат заедно за да ја минимизираат грешката при предвидување и оптимизација на невронската мрежа
 - **Forward propagation** --> Се проследуваат влезните податоци, се пресметуваат тежините и активацијата и се генерира излез за крајно да се пресмета загубата: разликата меѓу предвидената вредност и точниот излез
 - Forward propagation **не ги ажурира тежините** туку само ја пресметува загубата што ќе биде основа за backpropagation
 - **Backpropagation** --> **Пресметка на грешката** (MSE за регресија или Cross-entropy за класификација), па се пресметуваат градиентите (со ова разбираме како секоја тежина влијае на загубата), па **се ажурираат тежините** користејќи оптимизациски алгоритам со стапка на учење (learning rate), овој процес се повторува
 - Backpropagation прави **оптимизирање на тежините** преку методи како **Gradient Descent**
- Функциите за загуба ја мерат грешката и влијаат на насоката на учење.
Оптимизациските алгоритми го одредуваат темпото на подобрување на мрежата.

- 3) Како концептот на хиерархиско учење во длабокото учење се споредува со традиционалните машинско-учечки методи во однос на автоматската екстракција на карактеристики?

- **Deep Learning хиерархиско учење** => влезните податоци се проследуваат низ мултислојни невронски мрежи (CNN, RNN) и секој **слој учи различно ниво на карактеристики**, како што се гради мрежата така се оптимизира (Gradient descent, backpropagation) и користи **голем број на податоци** и моќни хардверски ресурси

- Невронската мрежа **сама ги идентификува релевантните репрезентации** на различни нивоа без човечка интервенција
- Предностите на deep learning е можноста за **автоматска екстракција на карактеристики (feature learning)** преку **хиерархиското учење**. Во паралела со традиционалните ML методи (**feature engineering**) **се прави рачно**
- Во традиционалните ML методи рачно се извршува инженерството на карактеристиките. Треба да се **анализираат податоците** и да се изберат најкорисните атрибути, па да се **тренира моделот**.
 - Исто така **не постои механизам за хиерархиско учење**, постојат фиксни карактеристики

Deep Learning примена:

- Автоматско препознавање на лица (CNN)** --> сам учи кои карактеристики се важни (форма на очи, големина на нос) и нема потреба од рачно програмирање на специфични карактеристики

Традиционален ML примена:

- Предвидување на болест** --> рачно избирање на релевантни карактеристики (крвен притисок, возраст) каде без човечка интервенција моделот не може сам да открие релевантни параметри

4) Популарни Deep Learning архитектури и нивната примена во реалноста

- CNN** (Convolutional Neural Networks) --> Специјализира за **обработка на слики**
- RNN** (Recurrent Neural Networks) --> Анализира **секвенцијални податоци** (текст, временски серии)
- Transformers, BERT** --> Обработка на јазик (NLP)

5) На кој начин проблемите со „експлодирачки“ и „исчезнувачки“ градиенти влијаат на процесот на тренирање на невронски мрежи? Дискутирајте ги техниките како што се нормализација на батч, различните функции за активација и нивното влијание врз решавање на овие проблеми.

- Градиентите играат клучна улога во тренирањето на невронските мрежи бидејќи тие го одредуваат **ажурирањето на тежините во секој слој преку backpropagation**

При процесот на backpropagation може да настанат 2 проблеми

- Исчезнувачки градиент** - кога градиентите стануваат премногу мали што доведува до бавно или воопшто никаква ажурирање на тежините
 - Последици:
 - ✗ Невроните во почетните слоеви не добиваат значајни ажурирања, што ги прави некорисни во учењето.
 - ✗ Мрежата учењето го концентрира само на последните слоеви, додека претходните слоеви остануваат скоро неизменети.
 - ✗ Тренирањето е многу бавно и неефикасно.
- Експлодирачки градиент** - кога градиентите стануваат премногу големи што доведува до нестабилно учење и преголеми вредности на тежините
 - Последици:
 - ✗ Губење на стабилноста на моделот – вредностите на тежините стануваат огромни.
 - ✗ Губење на корисни информации во слоевите поради премногу големи градиенти.
 - ✗ Неуспех во конвергенција – моделот никогаш не учи правилно.

Стабилизација на градиенти:

- **Batch Normalization** - нормализира активирања на секој слој, со што се стабилизира процесот на учење и се спречуваат екстремни вредности на градиентите.
- **Намалување на стапката на учење** (Learning Rate)

- 6) Зошто длабокото учење е особено ефикасно во обработка на слики и природен јазик во споредба со традиционалните алгоритми за обработка на податоци? Наведи конкретни примери и објасни улогата на конволуциските и рекурентните невронски мрежи.
- 7) Како техниките за регуларизација како dropout, L1 и L2 нормирање придонесуваат за подобрување на генерализацијата на моделите? Дискутирајте ги потенцијалните предизвици и компромиси во нивната примена.
- 8) **Кои се основните разлики помеѓу Boosting методите и Deep Learning во контекст на перформансите и апликациите? Дискутирајте ги предностите и недостатоците на секој пристап.**
- Основните разлики се тоа што
 - **Boosting** методите комбинираат слаби модели за да се добие посилен модел, може да работи добро и со помалку податоци, погодни се за **структурирани податоци** (табели, сетови)
 - **Примена во реалноста: Финансии, детекција на измами, здравство**
 - **Deep Learning** користи длабоки невронски мрежи за извлекување на значајни карактеристики, може да **учи комплексни нелинеарни релации** потребни се **големи количини на податоци** за да се истакнат квалитетни резултати, погодно е за **неструктурирани податоци** како слики, аудио, текст
 - **Примена во реалноста: Обработка на јазик, препознавање на говор, автономни возила**
- 9) Како големите количини податоци и унапредените хардверски ресурси (GPU, TPU) го овозможуваат подемот на длабокото учење? Дискутирајте го значењето на Big Data во развојот на модерните архитектури за длабоко учење.
- 10) **Зошто изборот на функција за активација (ReLU, Leaky ReLU, Sigmoid, Tanh, Swish) е критичен за перформансите на длабоките невронски мрежи? Објаснете ги предностите и недостатоците на секоја функција.**

- Функциите за активација се клучен дел од длабоките невронски мрежи бидејќи тие ја одредуваат нелинеарноста на моделот, можноста за учење на комплексни нелинеарни релации, без нив мрежата би била линеарен модел
 - Дополнително го разгледува присуството на исчезнувачки и експлодирачки градиенти

SIGMOID функција: Ги претвора влезните вредности во интервал (0,1) и често се користи во излезните слоеви за бинарна класификација

- PROS:

✓ корисен за задачи каде резултатот мора да е во опсег (веројатност)

- CONS:

✗ не е центрирана околу 0 (неефикасно тренирање)

✗ исчезнувачки градиент за големи или многу мали вредности

TANH функција: слична на SIGMOID но враќа вредности во интервал (-1,1)

- PROS:
 - ✓ Центрирана околу 0
- CONS:
 - ✗ исчезнувачки градиент
 - ✗ не се користи во модерни мрежи (заменета со RELU)

RELU функција: елиминира негативни вредности и остава само позитивни

- PROS:
 - ✓ нема исчезнувачки градиент
 - ✓ забрзано тренирање
 - ✓ едноставна и ефективна (подобри перформанси)
 - ✓ воведува SPARSITY (0 значи дека невронот не е активен што води до поефикасна обработка)
- CONS:
 - ✗ Dying RELU --> ако невроните добијат само негативни влезови тие секогаш враќаат 0 и не се ажурираат
- CONS:
 - ✗ не може да се користи во задачи каде опсегот на вредности мора да е ограничен (веројатност)

LEAKY RELU: го решава проблемот на dying RELU

- PROS:
 - ✓ ги елиминира мртвите неврони - дури и за негативни влезови
 - ✓ ги задржува добрите аспекти (брзина и едноставност)
- CONS:
 - ✗ нема оптимален избор на параметарот α

SWISH функција: слична на RELU но има мала нелинеарност

- PROS:
 - ✓ нема мртви неврони
 - ✓ флексибилна нелинеарност (подобри перформанси)
 - ✓ подобрена контрола врз градиентите
- CONS:
 - ✗ посложена за пресметување

- 11) Како функциите за активација влијаат врз проблемот на исчезнувачки и експлодирачки градиенти? Дискутирајте како одредени функции (на пр. ReLU) помагаат во справување со овие проблеми
- 12) Објаснете ја разликата меѓу различните функции за активација (ReLU, Sigmoid, Tanh, Swish) и нивното влијание врз перформансите на невронските мрежи. Во кои сценарија би се користела секоја од нив?
- 13) Како примената на Batch Normalization ја подобрува стабилноста и брзината на тренирање на длабоките невронски мрежи? Дискутирајте го концептот на „Internal Covariate Shift“ и неговите импликации врз тренирањето.
- 14) Објаснете го концептот на „feature learning“ во длабокото учење. Како автоматската екстракција на карактеристики се споредува со традиционалните техники за инженеринг на карактеристики во машинското учење?
 - Погоре е објаснето истото, споредба feature learning во DL VS engineering во ML
- 15) Како техниките како Batch Normalization, Dropout и Early Stopping помагаат во

оптимизација на невронските мрежи? Дискутирајте го нивното влијание врз стабилноста на тренингот и генералната способност на моделите.

- Оптимизација на невронски мрежи опфаќа справување со предизвици како нестабилност на градиенти, преголемо приспособување (overfitting) и бавно конвергирање на моделот

- 1) **Batch Normalization** --> ги нормализира влезните вредности на секој слој така што просекот на излезите останува околу 0 а стандардната девијација околу 1

- ✓ решава исчезнувачки/експлодирачки градиенти
- ✓ го забрзува тренингот (**стабилни градиенти овозможуваат поголеми стапки на учење**)

- Се користи пред активациска функција

- 2) **Dropout** --> случајно исклучува (**поставува на 0**) одреден процент од невроните во секој тренинг чекор што **спречува невроните да станат премногу зависни едни од други**

- ✓ спречува overfitting
- ✓ подобрена генерализација
- ✓ различни неврони учат различни репрезентации
- ✓ поголема флексибилност на моделот

- Се користи кога моделот има висок тренинг accuracy но низок validation

- 3) **Early Stopping** --> го прекинува тренингот кога моделот станува премногу приспособен на тренинг податоците односно прекинува да се подобрува а загубата расте

- Го следи валидацискиот сет, ако не се намалува валидациската загуба тогаш тренингот се прекинува и моделот ја задржува најдобрата верзија

- ✓ спречува overfitting
- ✓ го намалува времето на тренинг (не тренира непотребно)
- ✓ спречува деградација на моделот

- 16) Како изборот на функција за загуба (Mean Squared Error, Cross-Entropy) влијае на процесот на учење во длабоките мрежи? Кои фактори треба да се земат предвид при изборот на функција за загуба?
- 17) Како феноменот „Internal Covariate Shift“ влијае на тренирањето на невронските мрежи и како Batch Normalization помага во негово ублажување?
- 18) Како техниките како L1 и L2 регуларизација помагаат во намалување на преголемото прилагодување на моделите? Дискутирајте како тие влијаат врз вредностите на тежините во невронските мрежи.
- 19) Како Dropout регуларизацијата ја подобрува генерализацијата на длабоките мрежи? Дискутирајте го концептот на „stochastic regularization“.