Длабоко учење

Преглед

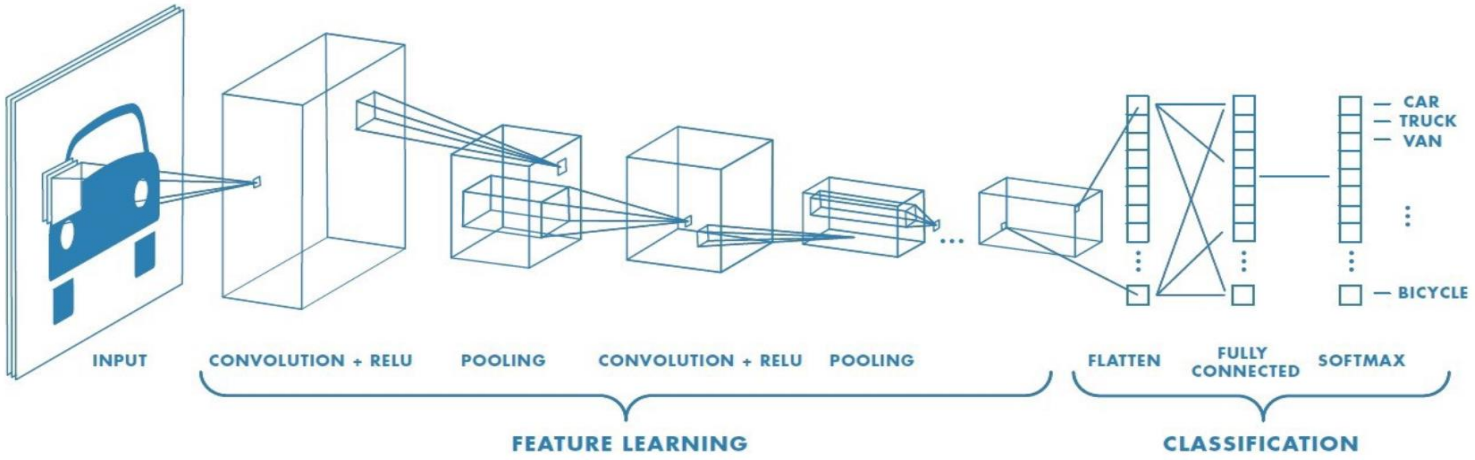
- Вовед во длабоко учење
- Конволуциски невронски мрежи
- Рекурентни невронски мрежи
- Автоенкодери
- Трансформери
- Граф невронски мрежи

Длабоко учење

- Длабоко учење опфаќа методи од машинско учење кои се базираат на вештачки невронски мрежи со повеќе од два слоја
- Многу функции кои не може да бидат претставени со плитки невронски мрежи се претставуваат добро со длабоки невронски мрежи кои поседуваат поголема експресивност
- Мозокот има длабока архитектура во која што се разменуваат информации помеѓу повеќе различни функционални единици кои генерално се слабо поврзани
- Когнитивниот процес кај луѓето е комплексен и хиерархиски организиран во мноштво едноставни и комплицирани концепти

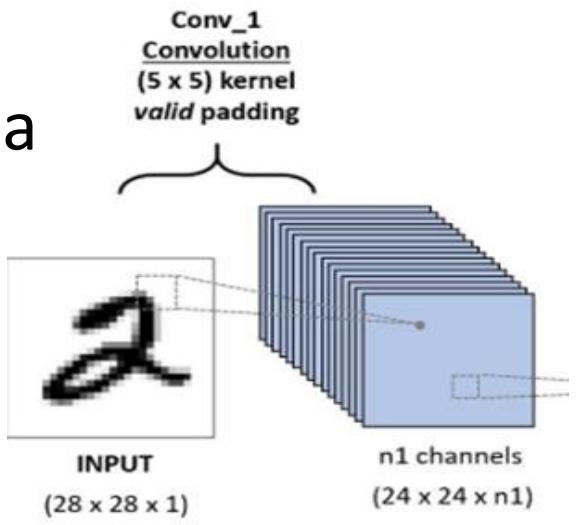
Конволуциски невронски мрежи (CNNs)

- CNN ја отсликуваат просторната корелација помеѓу пикселите во сликите со примена на соодветни филтри и на тој начин успеваат да извлечат соодветни карактеристики со кои може да се прави класификација на сликите
  - **Извлекување карактеристики** со неколку конволуциски и агрегирачки слоеви
  - **Класификација** со користење на извлечените карактеристики и стандарден MLP



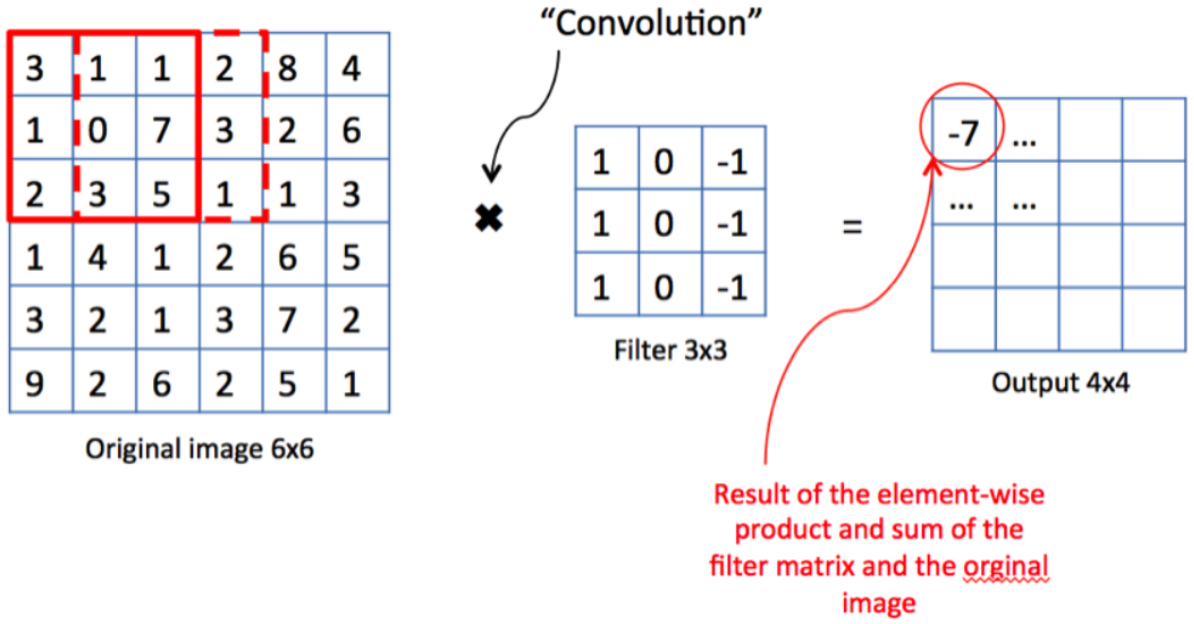
Конволуциски слој

- Во конволуцискиот слој се употребуваат повеќе филтри со кои се пресметува конволуција со што се исполнува одредена задача, а потоа се комбинираат нивните излези за да се реши одреден проблем, на пример класификација
  - Што повеќе филтри, толку повеќе детали и параметри
  - Секој филтер може да се гледа како еден неврон во MLP
- Треба да се избере и соодветна активациска функција
  - вообичаено ReLU
- Ако имаме 32 филтри на излез ќе добиеме 32 различни излезни матрици (feature maps)



Конволуција

- Конволуцијата е операција на трансформација на една влезна матрица со друга филтер матрица (кернел)
- Целта е да се извлечат некои карактеристики од сликите претставени како матрици, кои би биле релевантни за решавање на проблемот
- Вообичаено се користат филтри со димензии 3x3 или 5x5
- Димензиите на излезот се помали
  - Ако сакаме да се исти треба да додадеме нули на краевите ‘padding’



Филтри

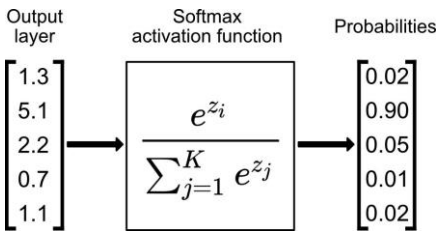
- Филтрите се користат стандардно во компјутерска визија за извлекување на карактеристики и процесирање на слики
- Во CNN параметрите на филтрите се учат од самите податоци
- **Пример:** Собел филтри за детекција на рабови

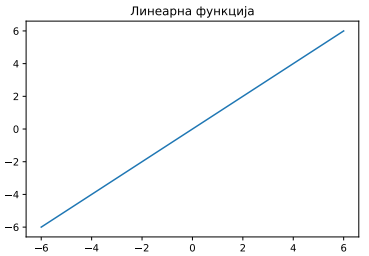
Vertical Sobel filter

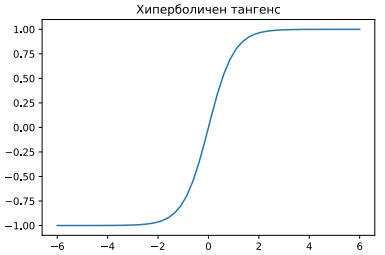
Horizontal Sobel filter

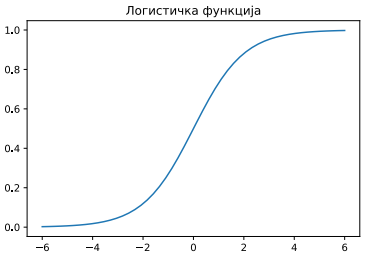
Активациски функции

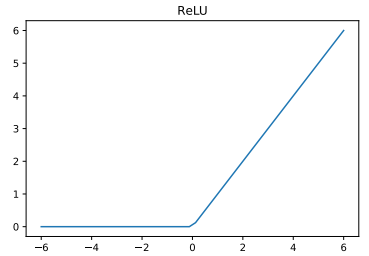
- Треба да се избере соодветна активациска функција
- Активацискатa функција игра значајна улога во функционирањето на невронските мрежи, и постојат неколку популарни активациски функции во скриените и излезните слоеви
  - Активациски функции во скриениот слој
    - Логистичка функција (често нарекувана сигмоид)
    - Хиперболичен тангенс
    - Исправена линеарна единица (Rectified Linear unit – ReLU) – популарен избор
  - Активацискатa функција во излезниот слој зависи од примената
    - Линеарна функција (регресија)
    - Логистичка функција (бинарна класификација)
    - Софтмакс (повеќекласна класификација)

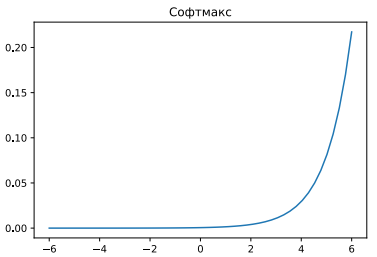






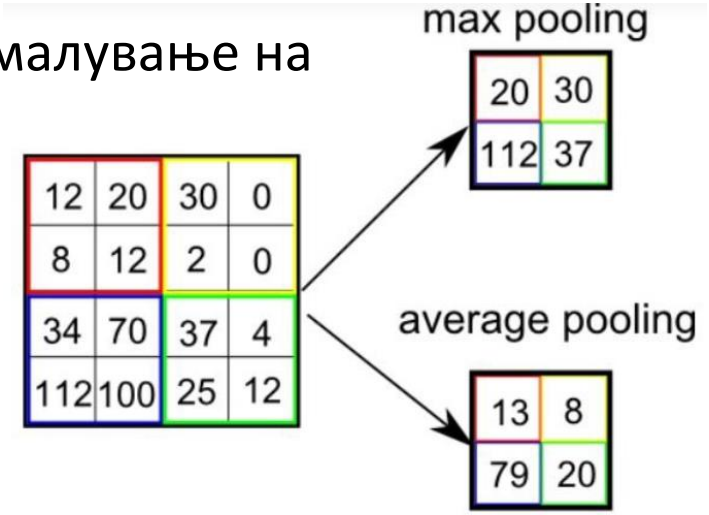






Здружувачки (pooling) слој

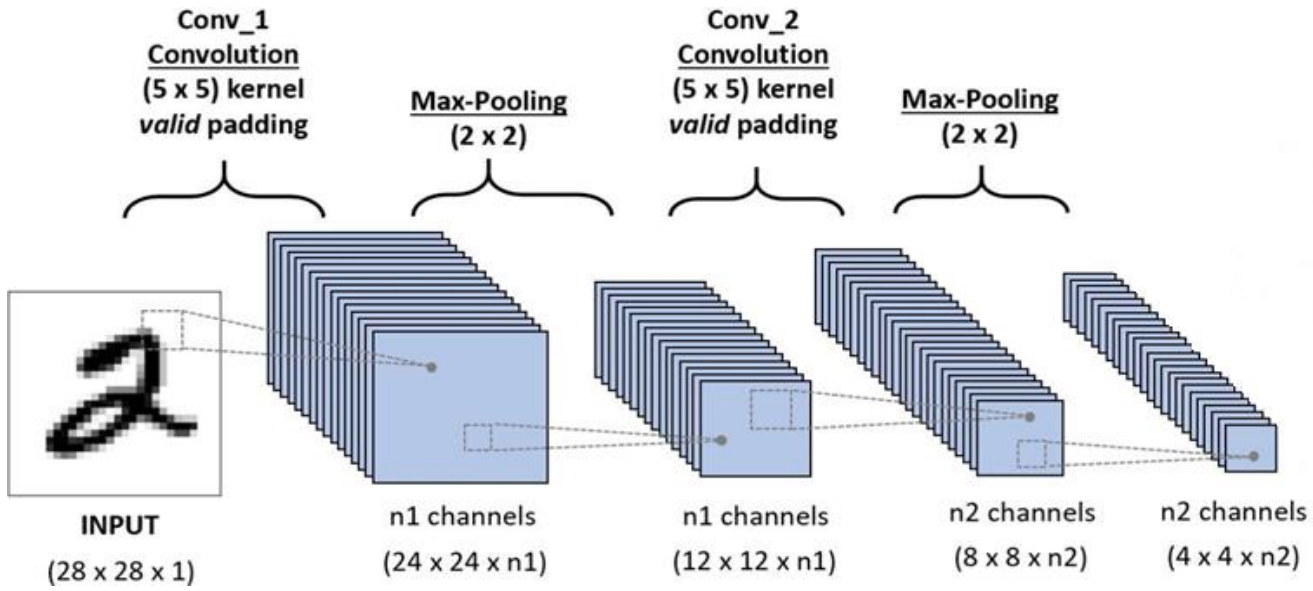
- Во сликите има голема корелација помеѓу соседните пиксели, па може да се намали на димензионалноста со избирање на репрезентативна вредност за секој блок од излезните матрици - **здружување**
  - Големина на блок – типично 2x2 што резултира во двојно намалување на димензионалноста
  - Тип на здружување
    - **Max pooling** – се избира максималната вредност од блокот
    - **Average pooling** – се наоѓа просечната вредност во блокот
  - Се уште се задржуваат главните карактеристики на сликата



Истата слика од претходно после две здружувања

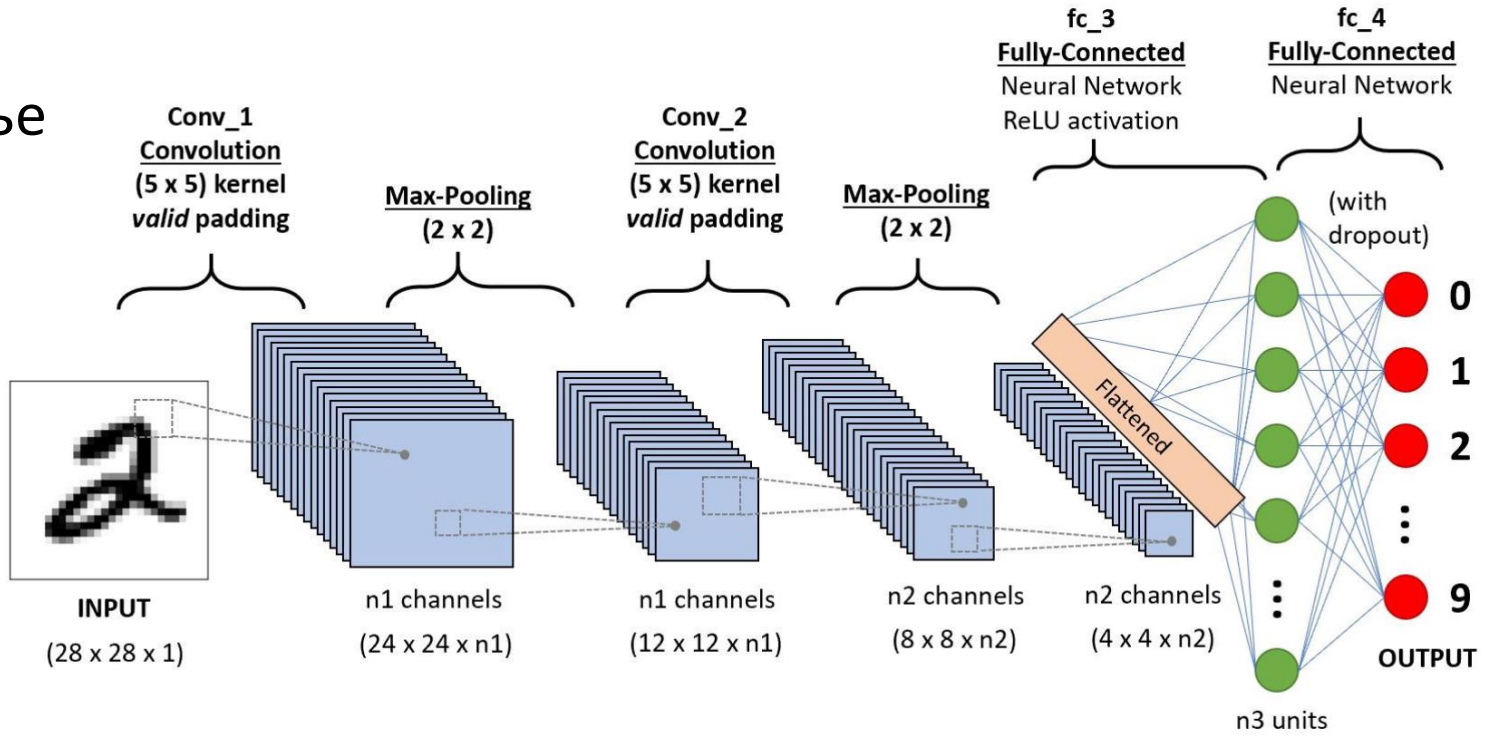
Извлекување карактеристики во CNN

- Може да се додадат повеќе конволуциски и здружувачки слоеви, со што се зголемува комплексноста на моделот и неговата експресивност со што може да се извлечат повеќе карактеристики
- Може да се додаваат и други слоеви
  - Dropout,
  - BatchNormalization,
  - Attention
  - и други



Излезен слој

- На крајот треба да се додаде излезен слој кој одговара на конкретниот проблем кој се разгледува
  - пр. за класификација може да се додаде трослоен целосно поврзан MLP со ReLU активација во скриениот слој и софтмакс активација во последниот слој
- Пред излезниот слој се прави порамнување



Регуларизација

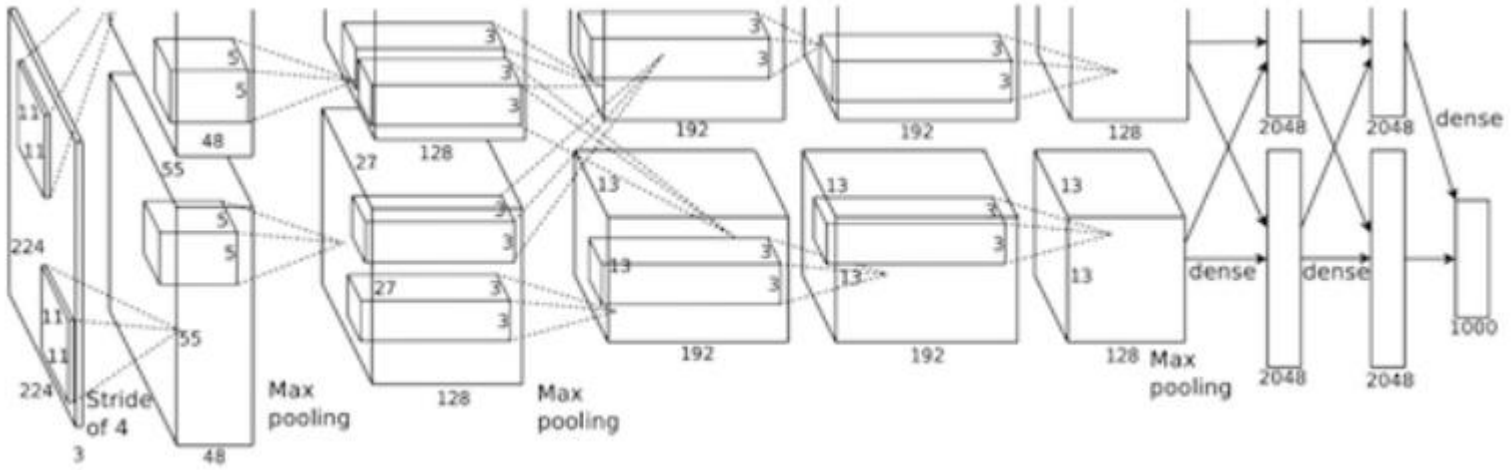
- За подобрување на точноста во тест фазата, односно, за избегнување на пренагодување (overfit) на моделот на тренинг множеството, се користат методи за регуларицација
- ‘Dropout’ е еден од најпопуларните и наједноставни за имплементација методи за регуларизација
  - Функционира така што дел од влезовите во јазлите се поставуваат на нула
  - Ратата со која се прави ресетирањето на нула (dropout rate) е хипер-параметар во моделот и се задава при неговиот дизајн

Збогатување на податочно множество

- Друга техника за тоа која може да се користи кога податочното множество е мало е негово збогатување (Data augmentation)
  - Кај CNN се додаваат нови слики при тренирањето добиени со примена на случајни трансформации на слики од постоечкото податочно множество
  - Мрежата е изложена на поголем број на варијации и затоа може подобро генерализира
  - Типови на трансформации
    - Ротација
    - Превртување (хоризонтално, вертикално)
    - Зголемување (zoom)
    - Сечење на дел од слика, и.т.н
  - Слични трансформации може да се применат и за други типови на податоци и за други модели на длабоко учење

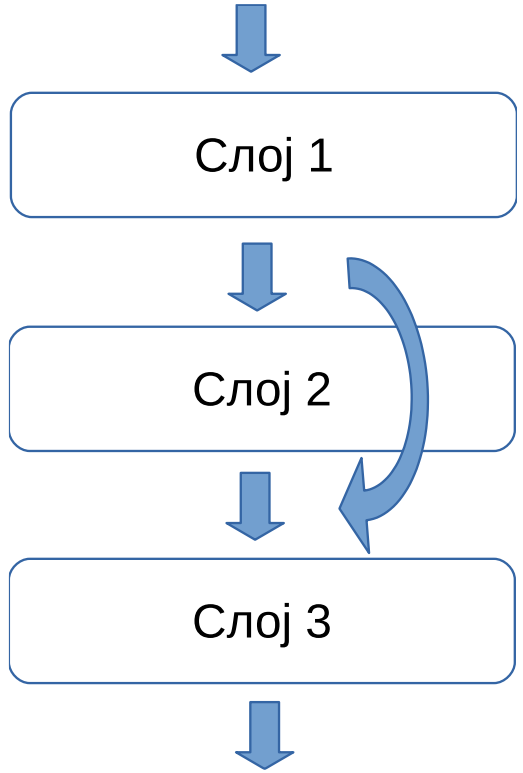
AlexNet архитектура на CNN

- **AlexNet (**Alex Krizhevsky, et. al., 2012)
  - Победник со голма разлика на ImageNet натпреварот
  - Придонес со употреба на:
    - ReLU
    - Dropout за избегнување на пренагодување
    - Препокривачки ‘max pooling’, наместо разводнувачкиот ‘average pooling’
    - GPUs NVIDIA GTX 580 за побрзо тренирање



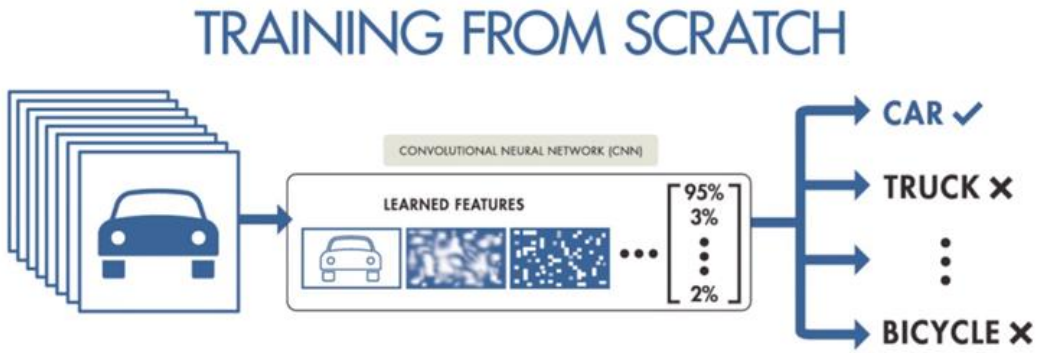
Резидуални врски

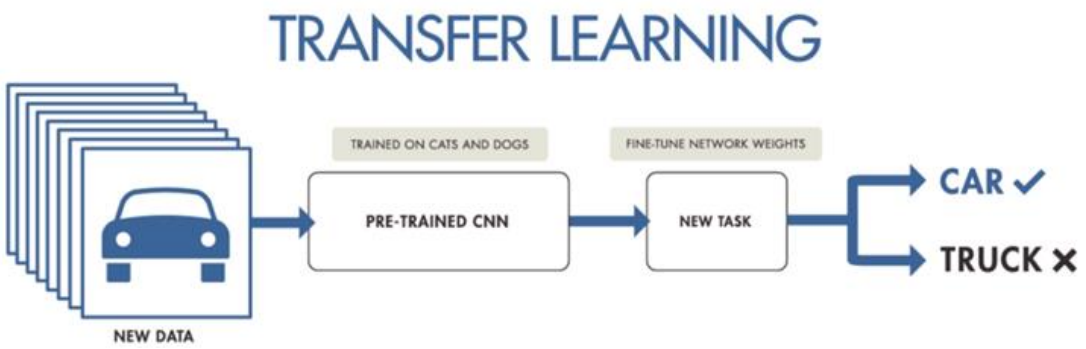
- Кај резидуалните невронски мрежи постојат врски со кои се “скокаат” некои од слоевите
  - Сличен концепт постои и во биолошките невронски мрежи
- На некој начин тие слоеви предвидуваат како треба да се промени излезот, наспроти кој треба да биде излезот
- Со овие врски се постигнува побрзо тренирање на погорните слоеви на почетокот, со што се избегнува исчезнувањето на градиентот
- Со текот на тренирањето влијанието на овие врски се намалува, а мрежата се приспособува на просторот на влезните карактеристики



Трансфер учење

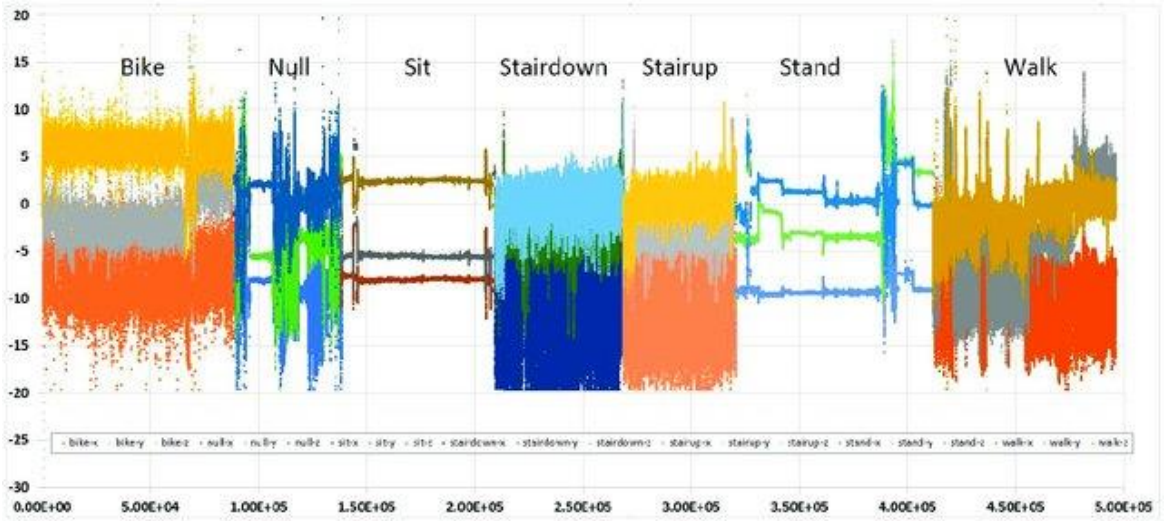
- Овозможува да се користат изворни модели кои се истренирани за решавање на еден проблем, за развој на целен модел со кој се решава друг сличен проблем
  - Погодно за решавање на целни проблеми за кои има малку достапни податоци, па тоа се надоместува со податоците достапни за првиот проблем
  - Изворните модели може да бидат некои постојни генерални модели, или модели развиени сепцијално со намера да се применат за решавање на втората целна задача
  - Вообичаено целните модели треба да се дотренираат за да бидат добри за решавање на втората задача





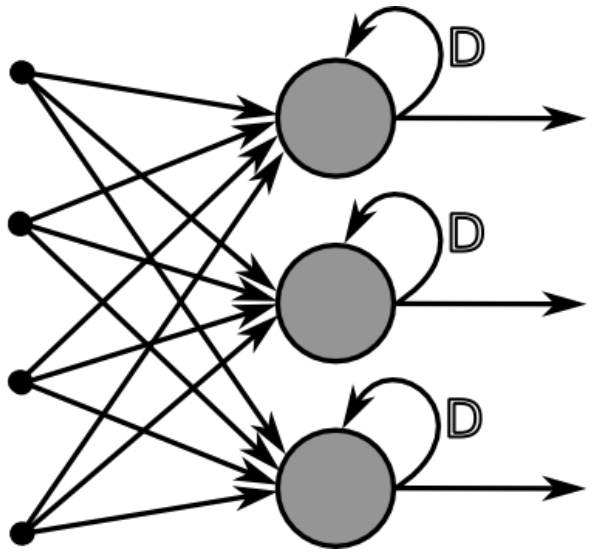
Секвенцијални податоци

- Некои податоци се временски или последователни, на пример временски серии од сензорски мерења, текстови, говор, итн.
- Вообичаено проблемите кои вклучуваат секвенцијални податоци потешко се решаваат со класични вештачки невронски мрежи
  - Предвидување на временски серии и детекција на аномалии
  - Препознавање на човечки активности од сензорски мерења
  - Препознавање на говор
  - Обработка на текст
  - итн.



Рекурентни невронски мрежи

- Рекурентни невронски мрежи (РНМ) се вештачки невронски мрежи во кои има и повратни врски, т.ш. излезите ќе зависат не само од тековните влезови туку и од претходните, со што се зачува одредена состојба на невронската мрежа
- Претставуваат нелинеарен дискретен динамички модел со надворешни влезови, некоја почетна состојба и излези
- Постојат повеќе типови на РНМ како:
  - Целосно рекурентни
  - Хопфилдови мрежи
  - Long-short term memory (LSTM)
  - Gated recurrent networks (GRU)
  - итн.

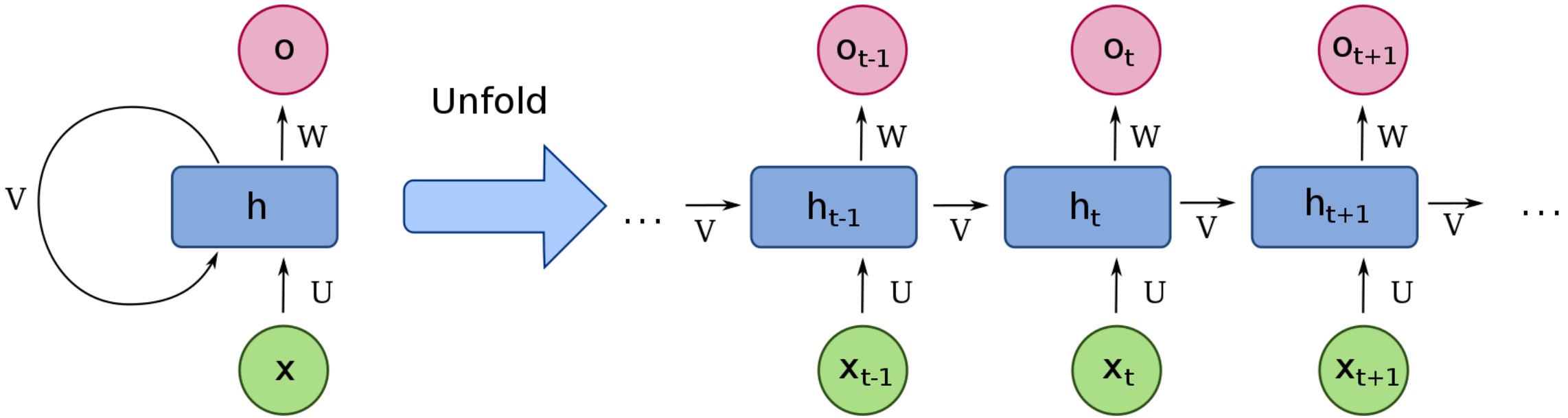


Дијаграм на РНМ

- x – влезови, o – излези, h – скриена состојба, v – тежини на повратни врски, U – влезни тежини, W – излезни тежини.

рекурентен приказ

одмотан приказ



Целосно рекурентни невронски мрежи

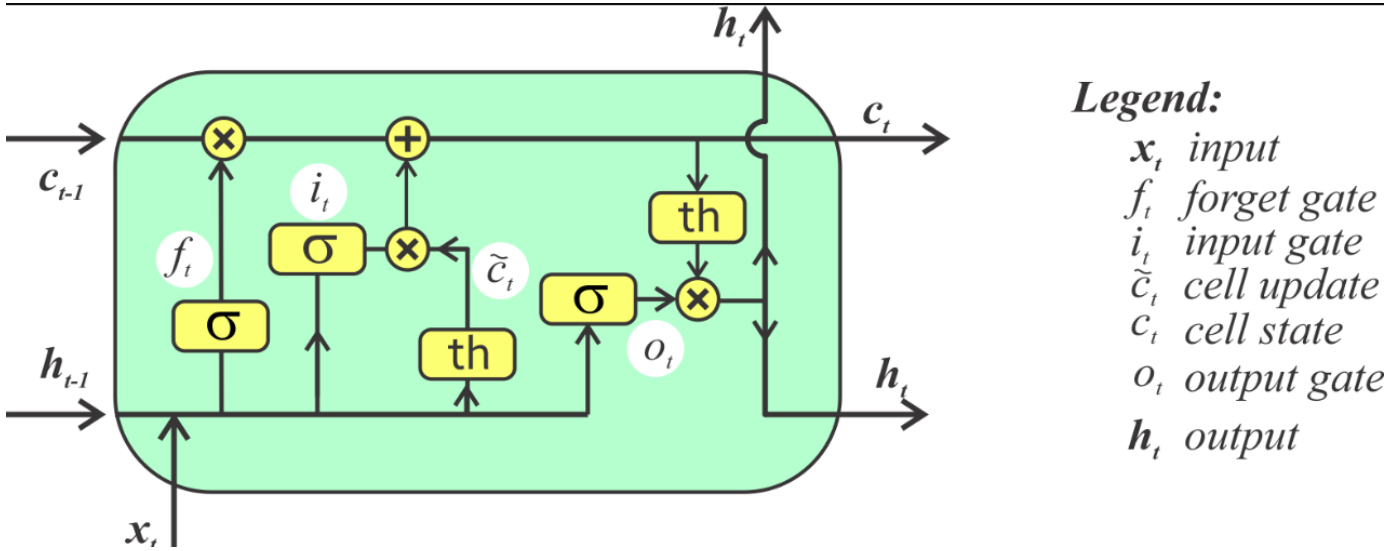
- Во целосно рекурентните невронски мрежи (ЦРНМ) излезите од невроните се проследуваат наназад до сите влезови, што значи постојат сите можни рекурентни врски
- Концептуално се наједноставен тип на РНМ, но пресметковно интензивни како што се зголемува бројот на неврони
- Останатите типови на РНМ имаат конкретни организации во кои постојат само одредени одбрани повратни врски

Проблеми во РНМ

- Класичните рекурентни невронски мрежи имаат долга историја на употреба, но при нивното тренирање со алгоритмот со пропагација-наназад може да се појават проблеми
- **Исчезнување на градиентот** – при тренирање на мрежата се прават промени на тежините зависно од парцијалниот извод од функцијата на грешка.
  - Овие изводи може да станат премногу мали како што се пропагираат наназад низ мрежата, посебно ако се користат класични активациски функции како сигмоид или хиперболичен тангенс
- **Експлодирање на градиентот** – спротивно од претходниот проблем, се појавува кога парцијалните изводи стануваат премногу големи при тренирањето на мрежата

Long short-term memory (LSTM)

- За надминување на проблемите во PHM е предложен LSTM
  - овозможува информациите да се чуваат долг временски период, не е преосетлив на промени во параметрите и дозволува градиентите да се одржат доволно големи за успешно тренирање
  - една LSTM единица се состои од информациска ќелија и три порти: влезна порта, излезна порта и порта за заборавање



Legend:

x_t *input*
f_t *forget gate*
i_t *input gate*
$\tilde{c}_t$ *cell update*
c_t *cell state*
o_t *output gate*
h_t *output*

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$

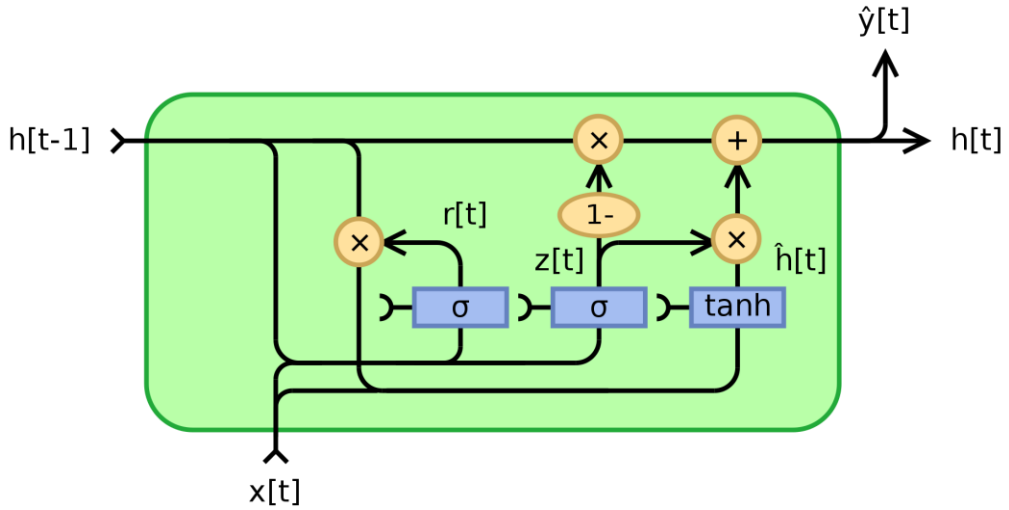
$$\tilde{c}_t = tahn(W_c x_t + U_c h_{t-1} + b_c)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$

$$h_t = o_t \odot tahn(c_t)$$

Gated recurrent unit (GRU)

- Слична на LSTM, но без регулатор на излезот и со <параметри
  - Често е попогодна за мали и небалансирани податочни множества
  - **Портата за ажурурање** (z_t) регулира колку од новите информации ќе бидат зачувани во состојбата
  - **Портата за ресетирање** (r_t) регулира колку од минатите информации ќе бидат заборавени



$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$

$$\hat{h}_t = tahn(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h)$$

$$h_t = (z - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t,$$

Двонасочни РНМ

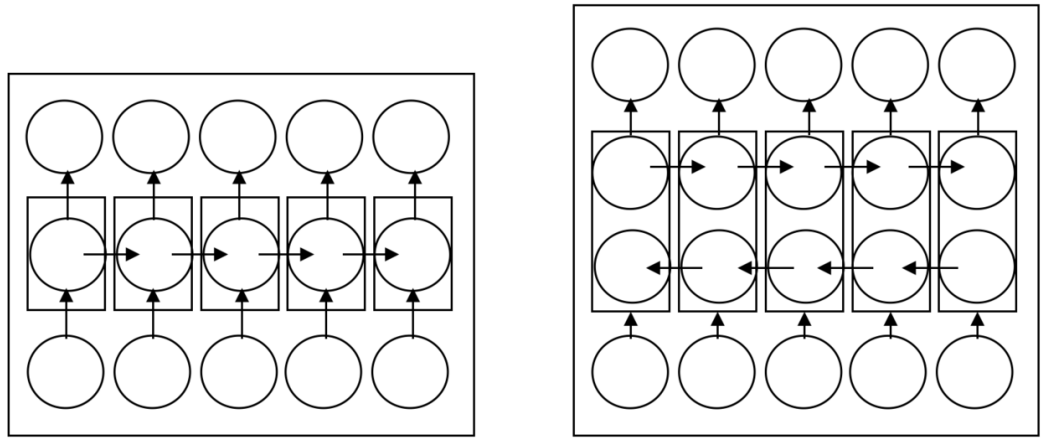
(a) Еднонасочна РНМ

(b) Двонасочна РНМ

- Има два скриени слоја во две спротивни насоки
- Тековната состојба ги зема предвид и минатите и идните влезови

Погодни за апликации како

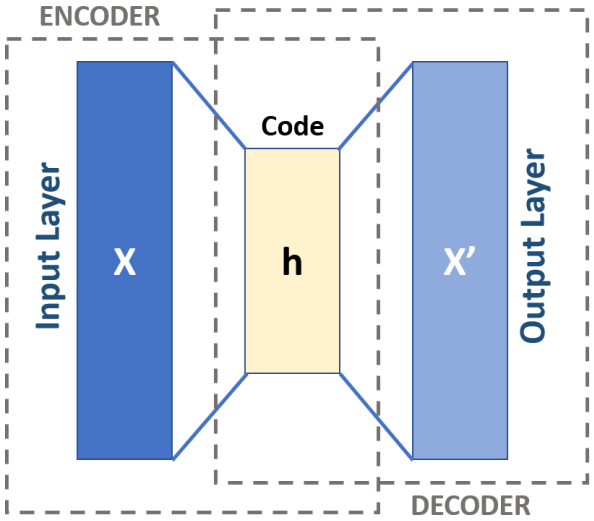
- Превод или сумаризација на текст
- Препознавање на говор или ракопис
- Екстракција на ентитети
- итн.



Structure overview
(a) unidirectional RNN
(b) bidirectional RNN

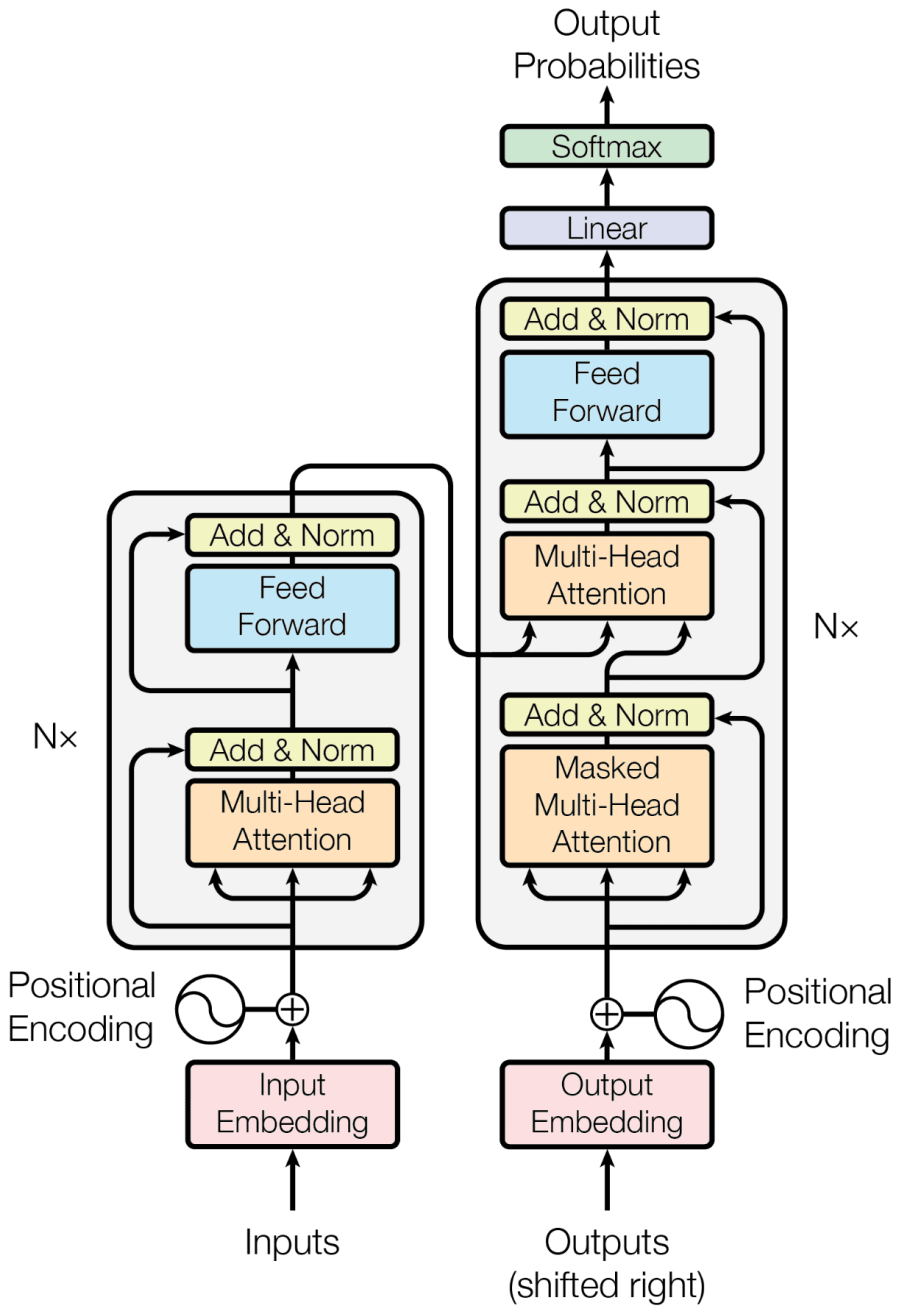
Авто-енкодери

- Автоенкодерите се невронски мрежи кои користат ненадгледувано учење за да се репрезентација со помала димензионалност на влезните податоци
- **Енкодерот** го мапира влезот во помала репрезентација h
- **Декодерот** пробува да го реконструира влезот од помалата репрезентација
- Има различни примени:
  - Редукција на податоци
  - Репрезентациско учење
  - Отстранување шум
  - Колаборативно филтрирање
  - Детекција на аномалии
  - итн.



Трансформери

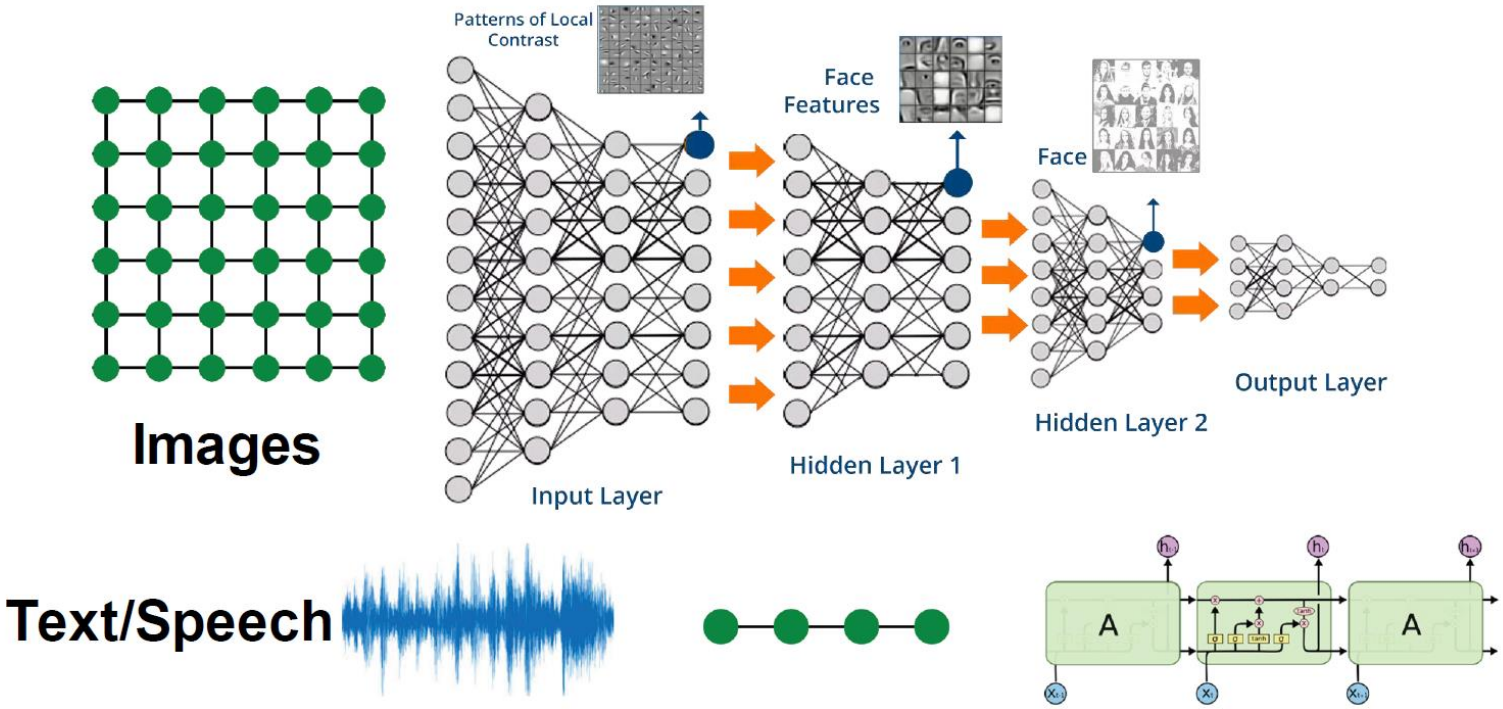
- Множество на кодери и декодери
- Има механизам за вниманите (attention)
- Нема рекурентни врски ни конволуции
- Трансфер учење
- На влез прима зборови (токени) претставени со ембедирани вектори
- Воглавно се користат во
  - **NLP**: класификација на текст, препознавање на именски ентитети, одговарање на прашања ...
  - **Компјутерска визија**: класификација на слики, детекција на објекти, сегментација на слики ...
- Постојат многу различни архитектури со разнолики апликации
- Познати примери:
  - BERT (Bidirectional Encoder Representations from Transformers) by Google
  - GPT-n (Generative Pretrained Transformer) by OpenAI



Attention is all you need, Vaswani et. al 2017

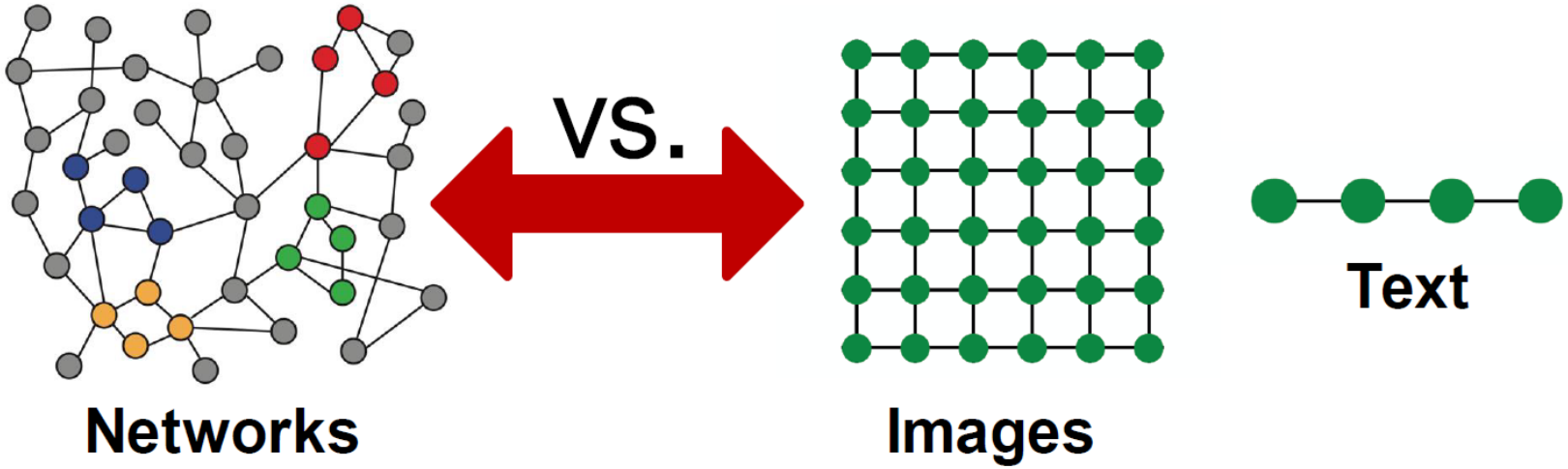
Модерни методи за длабоко учење

- Стандардните модерни методи за длабоко учење се дизајнирани за временско-просторни податоци како слики, текст, говор, временски серии ...



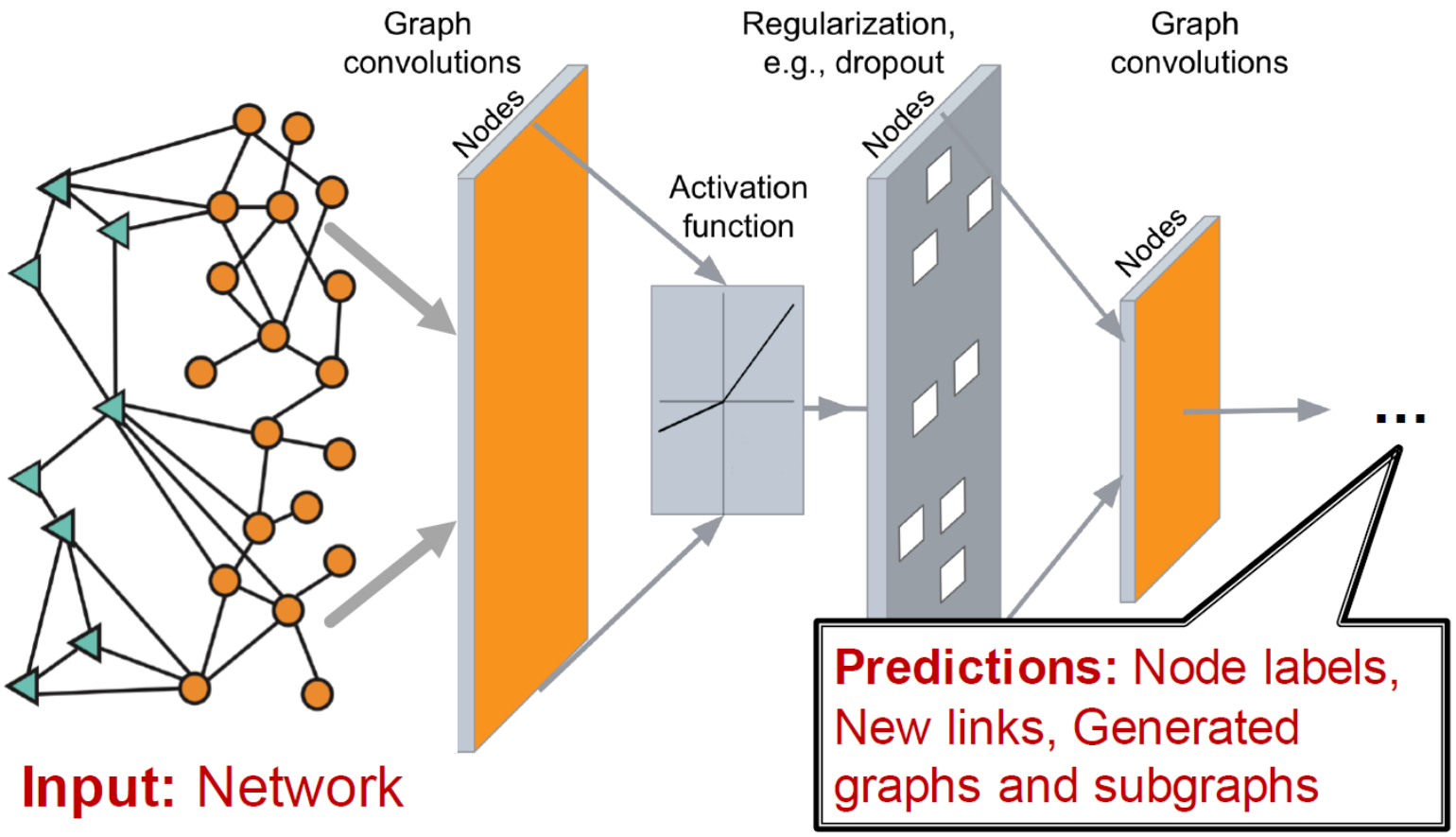
Граф длабокото учење е тешко

- Мрежите се комплексни
  - Произволна големина и комплексна тополошка структура



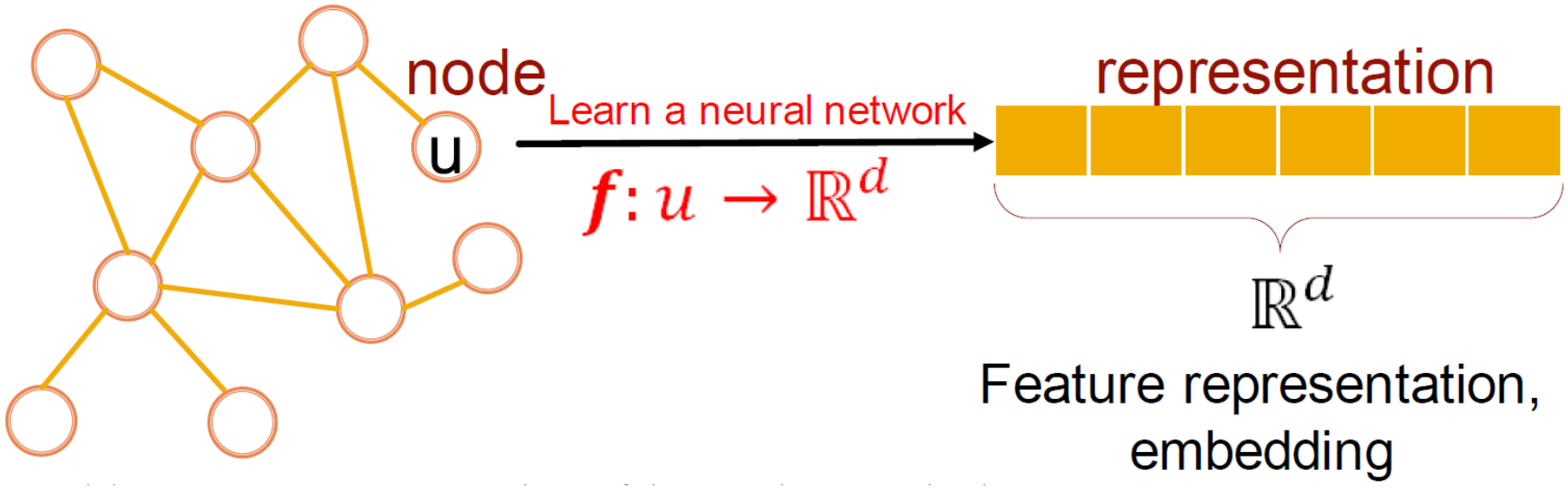
- Немаат фиксно подредување на јазлите или референтна точка
- Често се динамични и имаат мултимодални карактеристики

Длабоко учење во графови

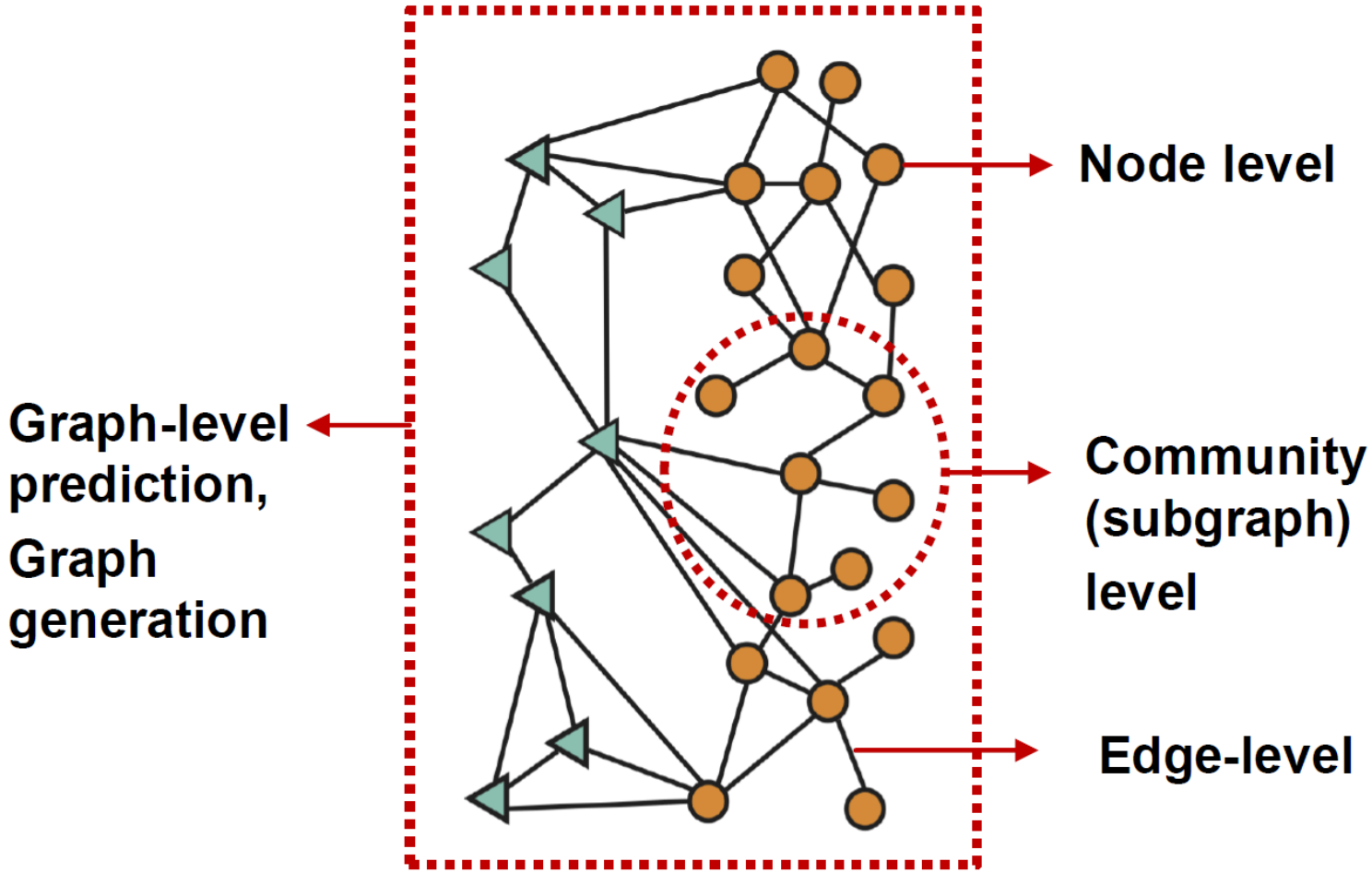


Граф репрезентациско учење

- Мапирање во d-димензионални ембединзи така што сличните јазли во мрежата се ембедираат блиску еден до друг



Различни типови на проблеми

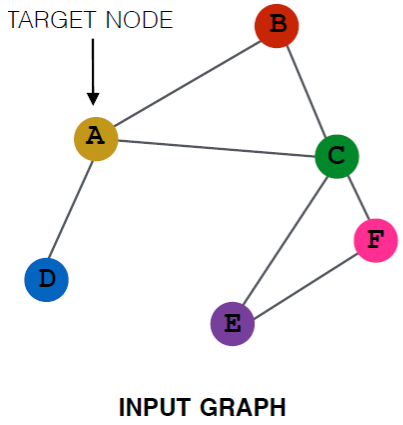


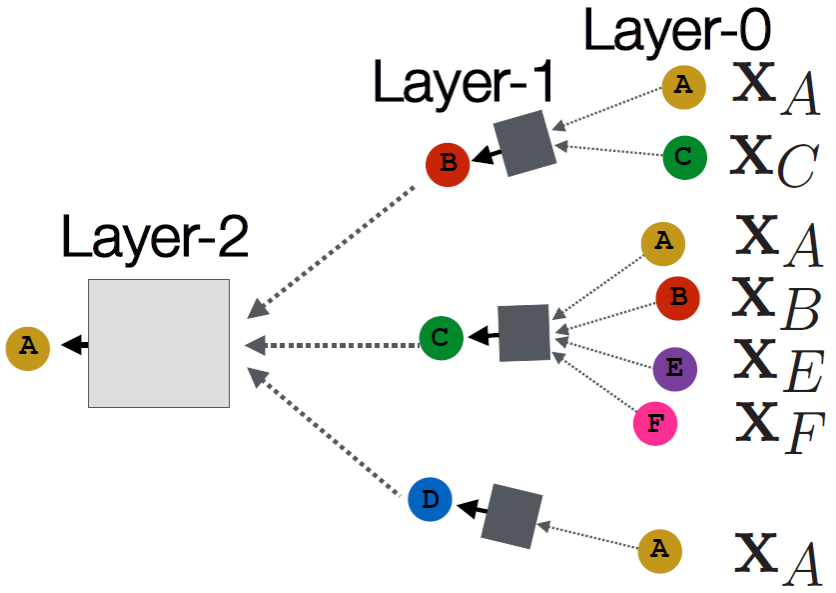
Класични задачи во граф машинско учење

- **Класификација на јазли**: предвидување на својство на јазол
  - Пример: категоризација на онлајн корисници, предвидување на превиткување (3D структура) на протеини - AlphaFold
- **Предвидување на линкови**: предвидување дали има линк помеѓу два јазли
  - Пример: комплетирање на графови на знаење, системи за препораки, предвидување на несакани ефекти од земање комбинации на лекови
- **Класификација на графови**: Категоризација на различни графови
  - Пример: откривање на лекови преки предвидување на молекуларни својства или генерирање на нови молекули
- **Кластерирање**: Откривање на заедници
  - Пример: откривање на социјални кругови
- Други задачи:
  - Генерирање на графови
  - Еволуција на графови

Граф невронски мрежи

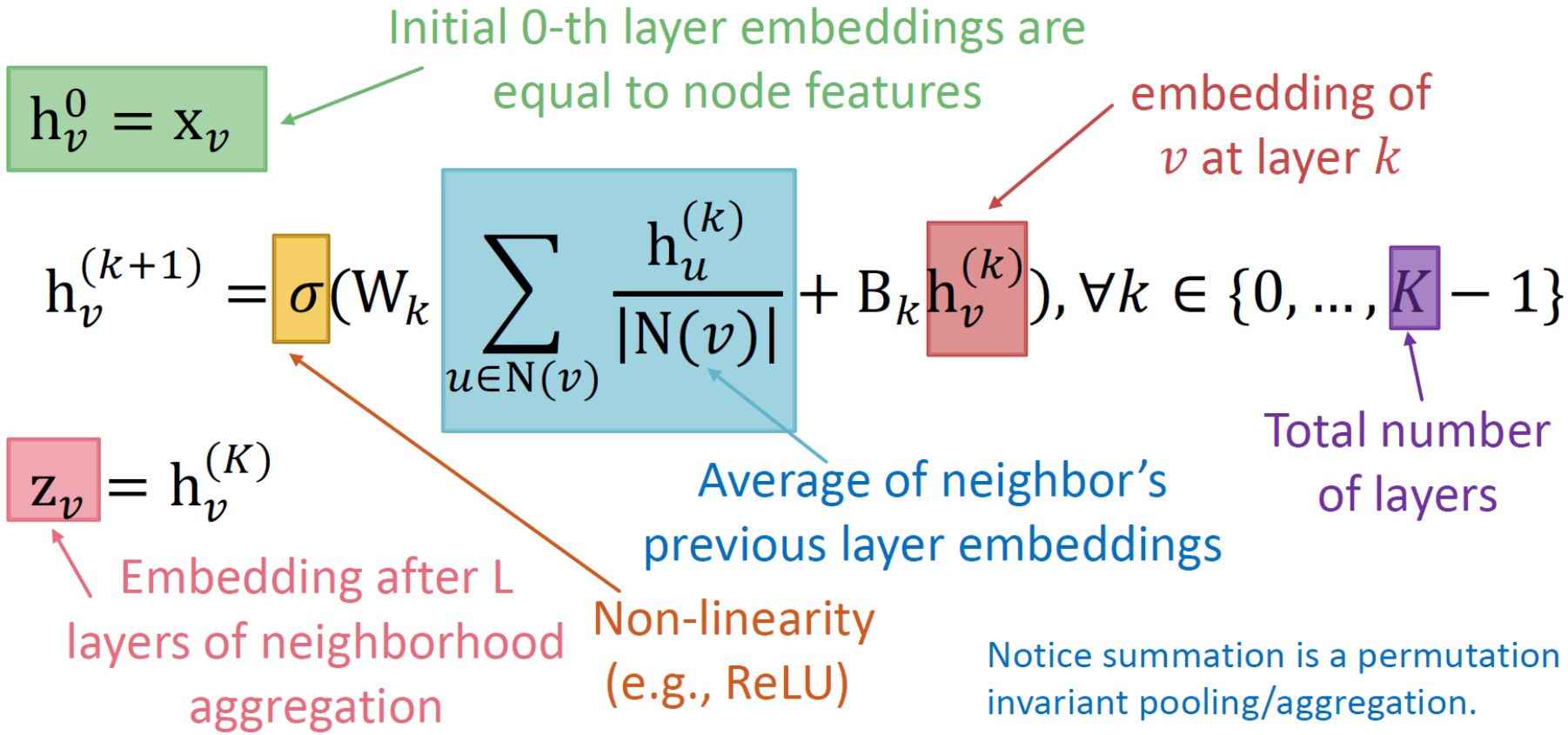
- Моделот може да има произволна длабочина
  - На секое ниво јазлите имаат соодветен ембедирачки вектор
  - На ниво-0 ембедингот на јазолот v е неговата карактеристика x_v
  - На ниво-k ембедингот се добива од информациите од соседите кои се k-скока далеку





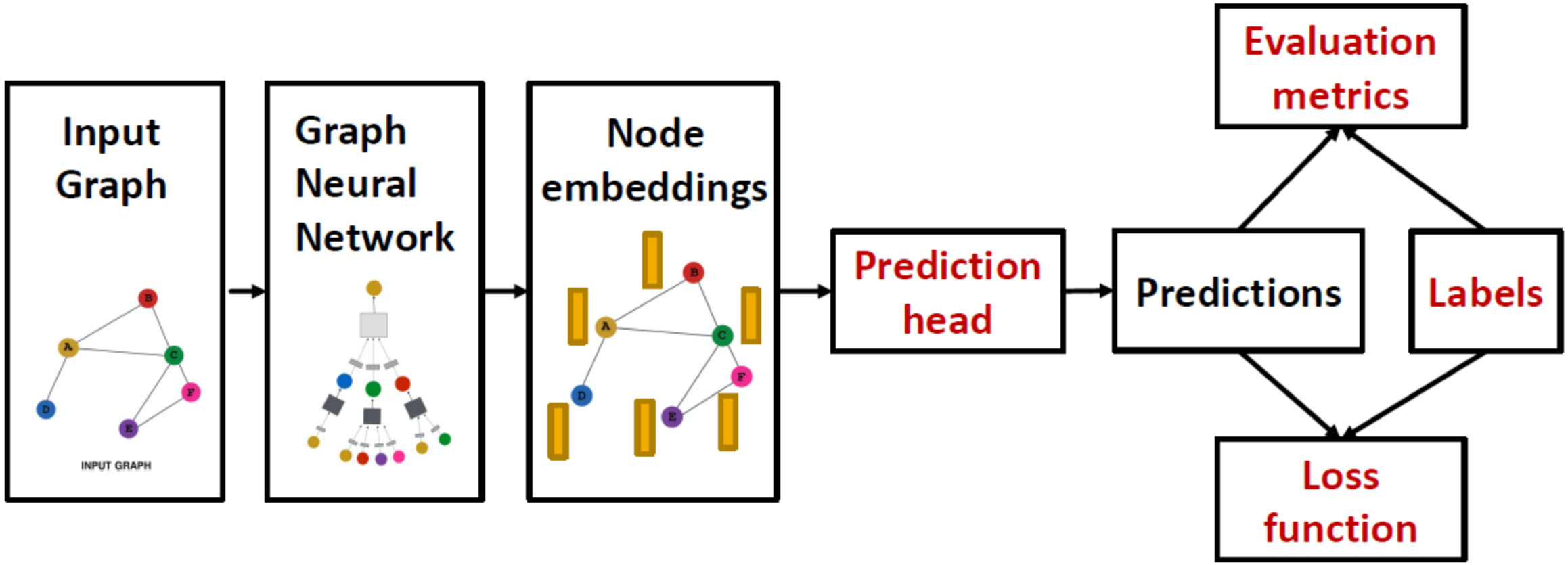
Граф конволуциска мрежа

- Едноставна граф невронска мрежа со пресметка на просек од информациите добиени од соседите



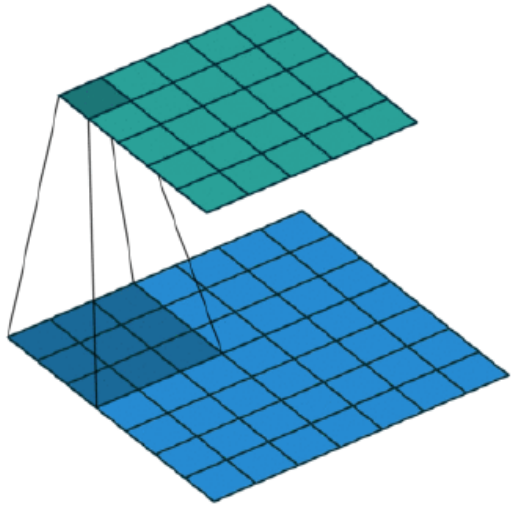
Целосна архитектура на GNN

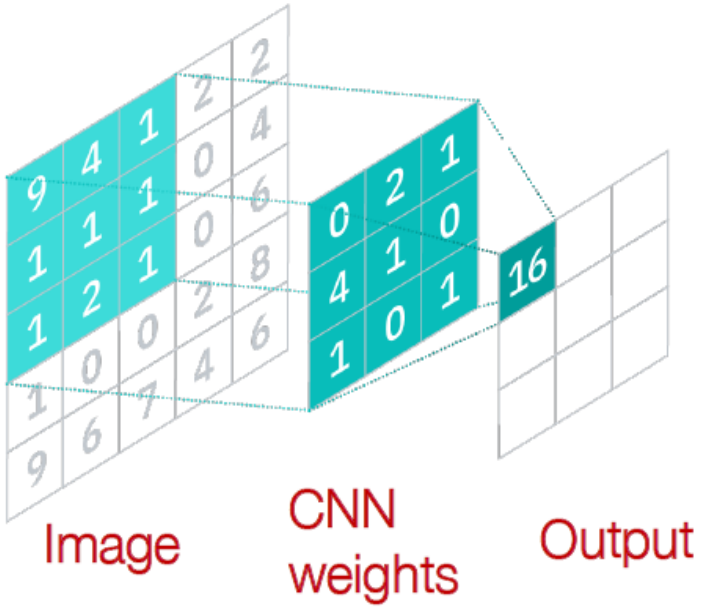
- Дополнително треба да се избере глава за предвидување со соодветна функција на загуба и евалуациска метрика зависно од проблемот



Конволуциска невронска мрежа

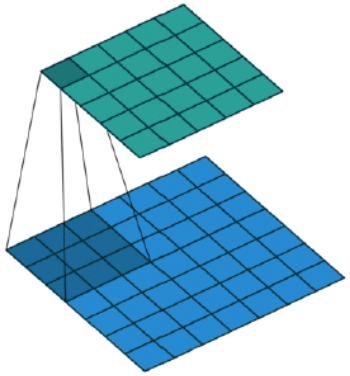
- CNN со филтер 3x3
  - N(v) ги опфаќа осумте соседни пиксели на v



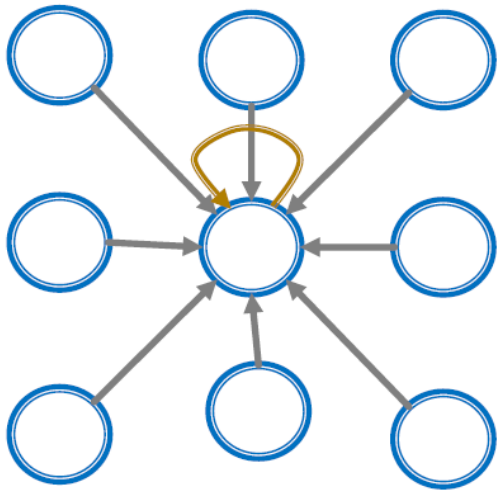


CNN formulation: $\mathrm{h}_v^{(l+1)} = \sigma(\sum_{u \in \mathrm{N}(v) \cup \{v\}} \mathrm{W}_l^u \mathrm{h}_u^{(l)}), \quad \forall l \in \{0, \ldots, L-1\}$

CNN vs. GNN



Image



Graph

- GNN formulation (previous slide): $\mathrm{h}_v^{(l+1)} = \sigma(\mathbf{W}_l \sum_{u \in \mathrm{N}(v)} \frac{\mathrm{h}_u^{(l)}}{|\mathrm{N}(v)|} + \mathrm{B}_l \mathrm{h}_v^{(l)}), \forall l \in \{0, \ldots, L-1\}$
- CNN formulation: $\mathrm{h}_v^{(l+1)} = \sigma(\sum_{u \in \mathrm{N}(v) \cup \{v\}} \mathrm{W}_l^u \mathrm{h}_u^{(l)}), \forall l \in \{0, \ldots, L-1\}$

  if we rewrite: $\mathrm{h}_v^{(l+1)} = \sigma(\sum_{u \in \mathrm{N}(v)} \mathbf{W}_l^u \mathrm{h}_u^{(l)} + \mathrm{B}_l \mathrm{h}_v^{(l)}), \forall l \in \{0, \ldots, L-1\}$

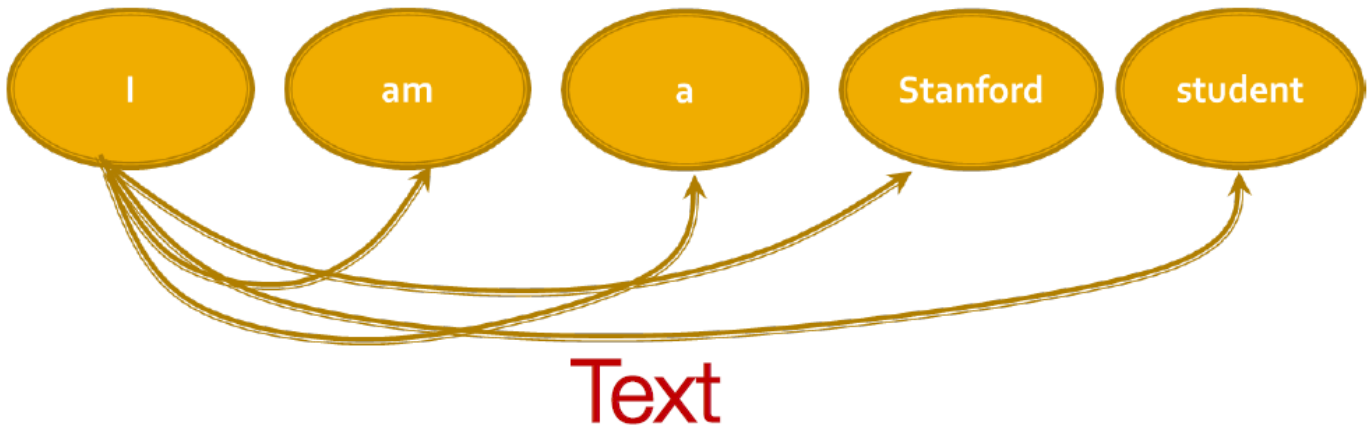
Кај CNN W_l^u може да е различна за сите соседи u зашто тие се подредени

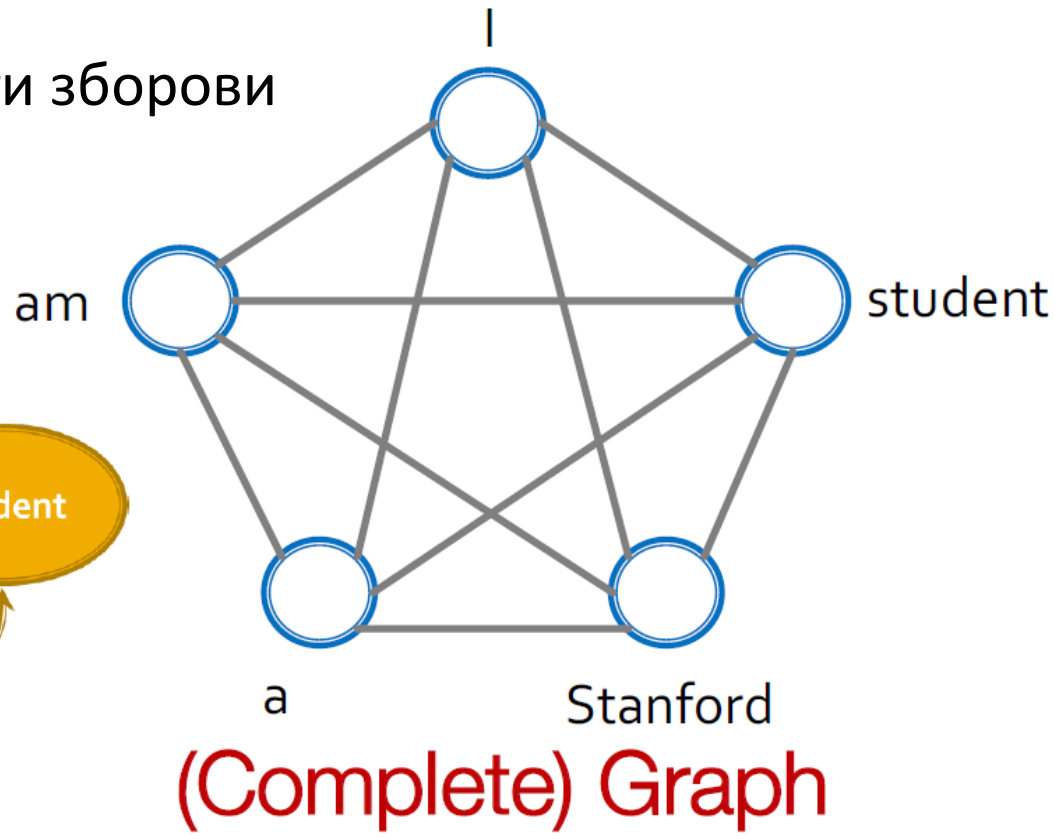
CNN vs. GNN

- CNN може да се гледа како специјална GNN со фиксно соседство и подреденост
  - Големината на филтерот е предефинирана во CNN
  - Предност на GNN е што може да процесира графови со различни степени за секој јазол
- CNN не е пермутациски инваријантна
  - Менувањето на редоследот на пикеслите води кон различен излез

Трансформери vs. GNN

- Трансформерите може да се гледаат како специјална GNN на целосно поврзан граф од зборови
  - Секој збор “внимава” на сите останати зборови





Билбиотеки за длабоко учење

- TensorFlow
- PyTorch
- Huggingface (најпопуларна за трансформери)
- PyG – PyTorch Geometric (развиена за GNNs)