

Adaptive principal component extraction - APEX

Jakub Bahyl¹

FMFI, mFTL - 4.ročník

(Dated: 10 December 2017)

V tejto práci prezentujem výsledky kompresie čiernobieleho 256×256 obrázku pomocou jednovrstvovej neurónovej siete s rekurzívnym algoritmom APEX. Po rozsekaní obrázku na partície s rozmerom 8×8 pixelov hľadáme ich vlastné vektory (PCs), do ktorých sa následne tieto 64-rozmerné dáta rozkladám, čím vykonávame dekompresiu. Po natrénovaní siete ukazujem rekonštrukcie modleového obrázku *Elaine* pre rôzne počty použitých neurónov. V diskusii vysvetľujem ich reprezentáciu a kvalitu aplikácie pre nové obrázky.

Keywords: principal component, adaptive NN

"You had to decompress the pressure before the race. I taught my heart to relax. I lay down before the race. It gave me more energy just before the race."

Emerson Fittipaldi

I. INTRO

Programovacia časť bola tvorená v jazyku Julia v0.5.1. Prílohou k tomuto projektu je zdrojový kód, ktorý je spustiteľný bez akýchkoľvek inštalácií cez Jupyter na [JuliaBox](#). Prihlásiť sa dá voľne cez Google+, Github alebo LinkedIn. Po uploadnutí kódu a vstupného obrázku by mal byť kód bežať.

Využívam knižnice *PyPlot* na grafy, *Images* na prácu s png obrázkom a *Pycall* na zneužívanie Pythonu.

II. IMPORT & PRE-PROCESSING

Obrázok, s ktorým budeme pracovať vyzerá takto:



FIG. 1. 256×256 Elaine z jednorozmerných pixelov.

Načítanie *Elaine.png* do pamäte Julie generuje floatové pole s rozmerom 256×256 , pričom každý člen matice leží v intervale $(0, 1)$, čo reprezentuje tmavosť (0=biela, 1=čierna). Následne obrázok normalizujem (škálujem na interval $(-0.5, 0.5)$), aby všetky PCs v neskoršej fáze prechádzali cez nulový bod.

III. APEX-OVÁ NEURÓNOVÁ SIETĚ

APEXový neurónový model je zobrazený na obrázku nižšie (FIG.2). N inputov je prepojených s M outputmi pomocou váhovej matice P . Navyše, pridané sú anti-hebbovské váhy vo vektore W , ktoré upravujú informáciu putujúcu so neurónu y_M . Predpokladám, že vstupné dáta sú stacionárne a ich korelačná matica má N rozdielnych eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_N$. Ďalej predpokladám, že $M - 1$ neurónov charakterizuje (už boli optimalizované) $M - 1$ normalizovaných PCs. Najvýznamnejšia *feature* APEX algoritmu je, že dokáže rekurzívne extrahovať najväčšiu ortonormálnu M -tú PC využívajúc vedomosť predchádzajúcich $M - 1$ PC.

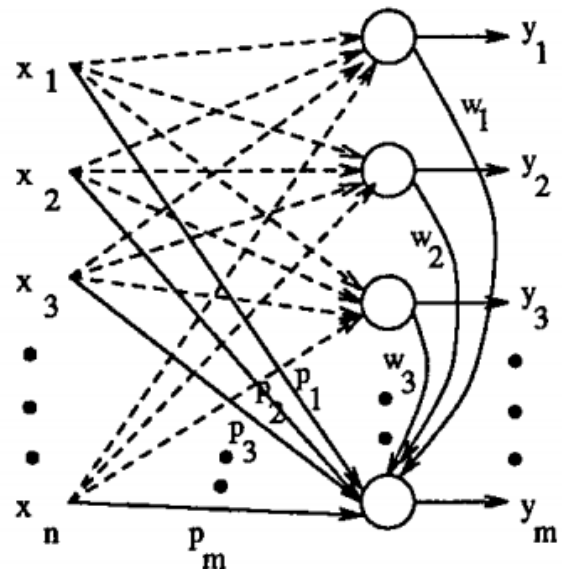


FIG. 2. Schéma neurónovej APEX siete.

APEX Algoritmus

- 1) Update-neme všetkých M neurónov pomocou matice P a k M -tému neurónu pripočítame anti-hebbovskú korekciu:

$$y = Px, \quad y_M := y_M + W^T x$$

- 2) K M -tému riadku matice P a k celému vektoru W pripočítame korekcie:

$$\Delta P_M = \beta(y_M x^T - y_M^2 P_M)$$

$$\Delta W = -\gamma(y_M y^T + y_M^2 W)$$

Všimnime si, že korekcia k riadku P_M je totožná ako sa používa v jednoduchšom *Oja's algorithm* a preto je táto časť APEX algoritmu hebbovská.

V limite nekonečnej iterácie by riadky matice P mali konvergovať postupne k normalizovaným PCs usporiadaným zostupne podľa veľkosti prislúchajúcej eigenvalue λ . Vektor W by mal v takej istej limite konvergovať k nulovým hodnotám. Tieto fundamentálne vlastnosti nám môžu poslúžiť ako rýchly test správnosti výsledkov dekompresie.

Nakoniec, veľkosť krokov β , γ volím ako $\beta = \gamma = 1/(\#Inputs)$. Dôkaz o konvergencii algoritmu, voľba kroku a iné podrobnosti možno nájsť priamo v článku Kostasa Diamantarasa, ktorý APEX algoritmus prezentoval v článku: *A neural network learning algorithm for adaptive principal component extraction (APEX)* (1990)

IV. TRÉNING NEURÓNOVEJ SIETE

Vstupnými dátami do APEX siete budú 8×8 -pixelové výseky (partície) z celého Elaine obrázku. Dokopy teda máme $32 \times 32 = 1024$ dát, ktoré sú 64-rozmerné. V každej epoche tréningu siete (hľadania PC) vyberám poradie vstupujúcich partícií náhodne. Motiváciou je nájsť ľubovoľný počet (M) najsilnejších PCs, takže začať treba hľadaním tej prvej :-). Na to úplne postačuje *Oja's algorithm*, čiže podmnožina APEX algoritmu.

Tréning teda beží nasledovne:

- 1) Inicializujem váhy $P \in \mathcal{R}^{M \times 64}$ a $W \in \mathcal{R}^8$ ako dostatočne úzku homogénnu distribúciu (šírka $\epsilon = 10^{-3}$). Nastavím krok $\beta = \gamma = 1/(32 \cdot 32) \approx 10^{-3}$, počet epoch= 5000 a počet hľadaných PCs ako $M = 20$.
- 2) Nájdem prvú PC pomocou Oja-ovho algoritmu.
- 3) Každú ďalšiu PC nájdem pomocou APEXu.
- 4) Na konci, keď už je natrénovaná matica P ako súbor PC vektorov, dá sa pristúpiť k testovaniu vlastností, počítaniu chýb a rekonštrukcií obrázku Elaine.

V. VÝSLEDKY

Rekonštrukciu obrázku robím tak, že postupne každú partíciu z pôvodného obrázku reshape-nem na vektor, transformujem na novú partíciu, re-shapnem späť a zošijem do nového obrázka:

$$\text{img} \rightarrow \text{part} \xrightarrow{\text{reshape}} X \mapsto P^T P X \xrightarrow{\text{reshape}} \text{part}^* \rightarrow \text{img}^*$$

Nakoniec de-normalizujem a výsledok porovnám s originálnym obrázkom. Vyhodnocovať budem klasickú L2 chybu na jeden pixel:

$$\text{Err} = \frac{1}{256^2} \sum_{\text{pixels}} (\text{img} - \text{img}^*)^2$$

V tabuľke nižšie uvádzam 10 hodnôt L2 chyby zrekonštruovaného obrázku, ktorý vznikol použitím rôzneho počtu PCs:

| PCs | Err [$\times 10^{-3}$] |
|-----|--------------------------|
| 1 | 6.06 |
| 2 | 4.68 |
| 3 | 3.57 |
| 4 | 2.74 |
| 5 | 2.44 |
| 6 | 2.18 |
| 8 | 4.26 |
| 10 | 4.17 |
| 12 | 4.43 |
| 15 | 2.38 |

Korelačná matica a vlastné hodnoty

Pre celý obrázok Elaine skonštruujeme korelačnú maticu podľa definičného vzťahu:

$$\text{Corr} = \frac{1}{32^2} \sum_{\text{parts}} X X^T,$$

kde X sú 64-rozmerné vektory danej partície obrázku. Eigenvalues matice obrázku Elaine porovnáme s eigenvalues najlepšieho zrekonštruovaného obrázku. V tabuľke nižšie uvádzam hodnoty zostupne zoradených najväčších desať eigenvalues spolu s chybou:

| Eig [$\times 10^{-4}$] | Eig* [$\times 10^{-4}$] | Diff [$\times 10^{-4}$] |
|--------------------------|---------------------------|---------------------------|
| 348.02 | 353.90 | +5.89 |
| 28.67 | 28.96 | +0.29 |
| 22.14 | 21.82 | -0.33 |
| 5.76 | 5.70 | -0.06 |
| 3.00 | 2.97 | -0.04 |
| 2.17 | 2.15 | -0.03 |
| 1.76 | 1.74 | -0.02 |
| 1.16 | 1.15 | -0.01 |
| 1.01 | 1.00 | -0.01 |
| 0.94 | 0.92 | -0.01 |

Vizualizácie:

Na poslednej strane možno nájsť niekoľko vizualizácií:

FIG. 3: Tu možno vidieť 10 obrázkov, pričom 9 z nich bolo zrekonštruovaných pre 9 rôznych počtov neurónov PCs $\in (1, 2, 3, 4, 5, 7, 9, 12, 15)$. Posledný obrázok je original. Veľmi dobre vidno, ako sa postupne (so zvyšujúcim počtom zapojených neurónov) kvalita rekonštrukcie blíži ku kvalite originálu.

FIG. 4: Vizualizácia najsilnejších 10 neurónov (PCs s najväčšími eigenvalues). Dá sa identifikovať, že prvý neurón nesie informáciu o intenzite, druhý a tretí uhlopriečných hraníc, štvrtý a piaty špeciálne ohraničenia, ostatné už využívajú komplikovanejšie, ťažko interpretovateľné vzory.

FIG. 5: Výsledky rekonštrukcie určitými skupinami neurónov. Naľavo je rekonštrukcia prvým neurónom, teda každý 8×8 rozmerná partícia je celá zafarbená nejakým odtieňom. Na druhom obrázku je použitý druhý a tretí neurón, čiže dokopy vidíme všetky dôležité hranice. Tretí obrázok je vystavaný druhým až piatym neurónom, teda všetkými (okrem intenzitného neurónu), ktoré nesú pre ľahko interpretovateľnú informáciu. Vo štvrtom obrázku je použitý 2-8 neurón a v úplne poslednom všetky neuróny okrem prvého.

FIG. 6: Tu aplikujem natrénovanú neurónovú sieť na cudzie obrázky. Použil som štyri čiernobiele obrázky s rozmerom 256×256 . Okrem faktu, že obrázky sieť zrekonštruovala pomerne kvalitne (aj keď bola natrénovaná na odlišnom obrázku), si môžeme všimnúť, že sieť má problém s veľmi komplikovanými tvarmi ako napr americká vlajka v pozadí Obamu alebo strapaté vlasy Einsteina. Na tréningovom obrázku elaine také vzory neboli, takže sieť na niečo také nie je pripravená. Ešte si možno všimnúť, že obrázky sú akoby celé trochu zasněžené. To je spôsobené štvrtým neurónom, ktorý nemal ideálnu konvergenciu počas tréningu (viac to komentujem v Záverečných poznámkach).

VI. ZÁVEREČNÉ POZNÁMKY

Počas hľadania každého PC som počas prechádzania epochami sledoval priemernú hodnotu vektora W ako kontrolu, či vektor naozaj konverguje k nule tak, ako by mal. Pri niektorých prípadoch (konkrétne 5-ty PC) sa stalo, že vektor W vôbec nekonvergoval k nule ale k nejakej konkrétnej, pomerne vysokej hodnote. Veľkosť kroku, rozdielna inicializácia, ani počet epoch tomuto problému nepomohol. 5ta PC jednoducho vždy vychádzala veľmi zle. Všetky dáta okolo 5-tej komponenty sú teda z výsledkov zámerne vynechané, čo sa žiaľ určite prejavilo na kvalite rekonštrukcií, ktoré zahŕňali viac ako 5 neurónov.

Na obranu 5-eho neurónu treba podotknúť, že ani jeden z ostatných neurónov nemal tendenciu konvergovať k nule pri obrovskom počte epoch či miniatúrnom kroku. Každý konvergoval k inej (niekedy aj rádovo inej) hodnote, čo vysvetľuje aj to, prečo sú najlepšie rekonštrukcie zašumené (4-ty neurón po 5-tom druhý najhoršie určený neurón. Súdim podľa toho, ako skonvergoval vektor W . Preto v každej 8×8 partícii vidno výraznú "stopu" tohto neurónu.). Chyba však nebola natolko iritujúca, aby som aj tento z prezentácie odstránil.



FIG. 3. Rekonštrukcia pre rôzny počet neurónov + original.

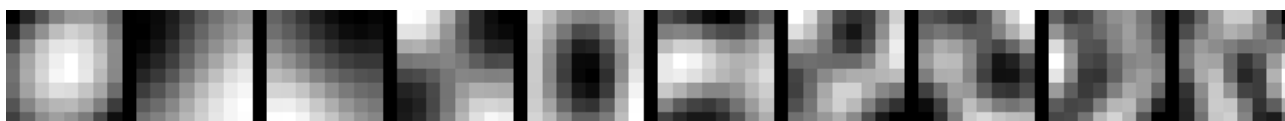


FIG. 4. Náhľad na prvých 10 zmysluplných neurónov.

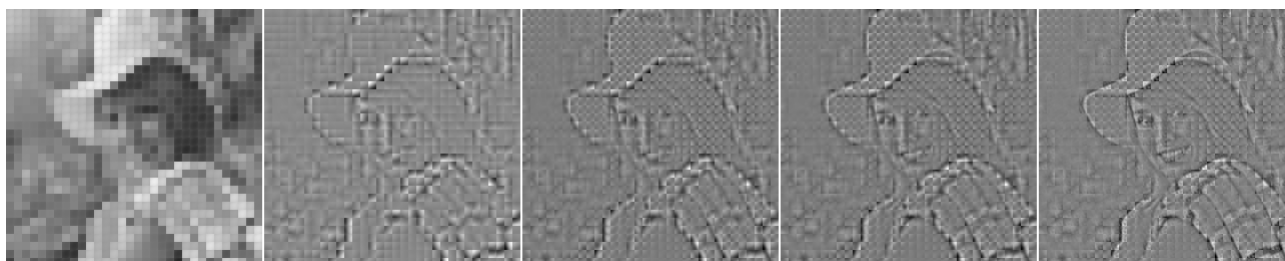


FIG. 5. Rekonštrukcia pomocou niekoľkých skupín neurónov.



FIG. 6. Rekonštrukcia iných obrázkov + original.