Deep Learning Szeminárium
5. előadás: Mély hálózatok

Zombori Zsolt

MTA, Rényi Alfréd Matematikai Kutatóintézet

A hálózat mélysége

- A 2000-es évekig 1-2 rejtett rétegű, sekély hálózatokkal kísérleteztek
- Több réteg növeli a reprezentáció rugalmasságát
  - Kevesebb neuron is elég lehet
  - Hierarchikus adatreprezentációk tudnak létrejönni
  - Gazdagabb reprezentációs struktúra, jobb általánosítás
- Sajnos mélyebb hálózatokat nehezebb tanítani!

Instabil gradiensek problémája

Mély hálózatokban a különböző rétegek más sebességgel tanulnak

http://neuralnetworksanddeeplearning.com/chap5.html

- Vanishing gradients: bemenet felé egyre kisebb gradiensek
- Exploding gradients: bemenet felé egyre nagyobb gradiensek

A korai rétegek gradiense egy hosszú szorzat eredménye $W_i^T \sigma'(z_i)$ alakú tényezőkkel

Ahhoz, hogy a különböző rétegek gradiensei hasonló nagyságrendűek legyenek, valamilyen mechanizmusnak gondosan ki kell(ene) egyensúlyoznia ezeket a tényezőket.

Instabil gradiensek problémája

Sok trükk segíthet ennek kezelésében:

- Más aktivációs függvények keresése: ReLU
- Súlyok és eltolások gondos inicializálása
- Több tanító adat, augmentáció
- Különféle SGD variánsok átméretezhetik a gradienst
- Konvolúciós hálók: kevesebb paraméter, gazdagabb gradiens jel
- **Batch Normalization**: stabil eloszlás a rétegek bemenetein
- **Reziduális hálózatok**

Batch Normalization

- A rétegek bemenetének eloszlása változik a tanulás során
- Törékenyebb és lassabb a tanulási folyamat
- A rétegeknek folyamatosan adaptálódniuk kell a megváltozott adatkörnyezethez
- Ötlet:
  - Normalizáljuk a retegek bemeneteit
  - Mini batch alapján, komponensenként normalizálunk
  - A normalizált eloszlás paraméterei tanulhatóak

Batch Normalization

- Sergey Ioffe, Szegedy Krisztián (2015)
- Google Scholar szerint 4020 hivatkozás

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

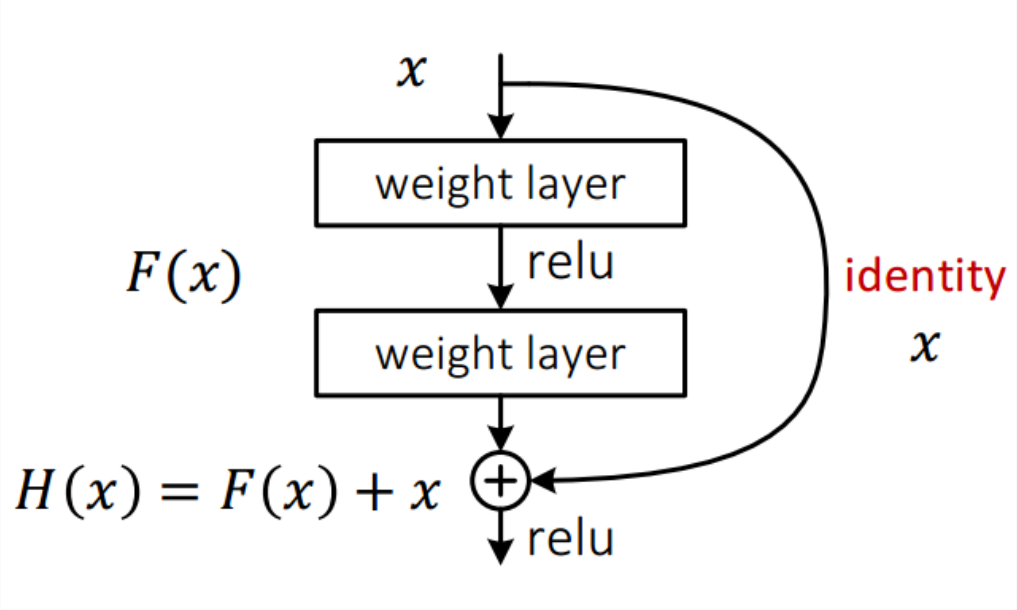
Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

Reziduális hálózatok

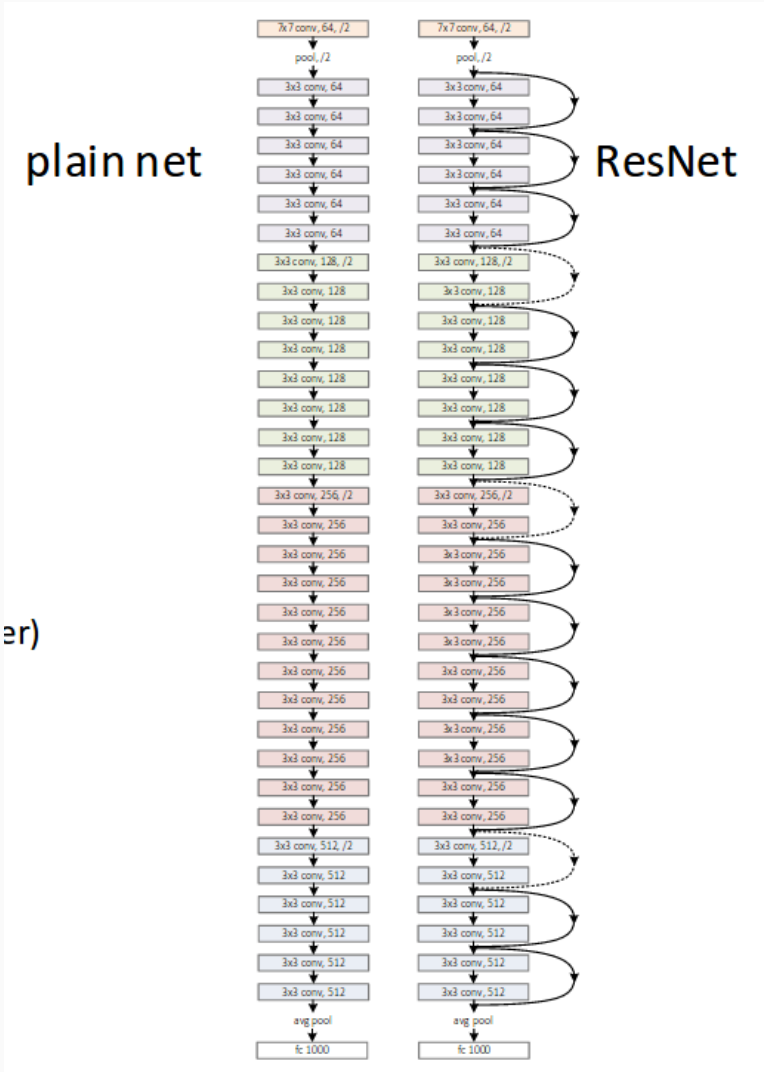
- Degradation problem: Egy hálózathoz új rétegeket adva egy ideig nő a pontosság, de gyorsan telítődik és aztán elkezd csökkenni.
- Pedig a hálózat reprezentációs képessége nő!
- Hiszen az identitás függvényt kellene megtanulni, hogy az új réteg ne ártson.

Reziduális hálózatok

- Nem könnyű SGD-vel identitás függvényt tanulni.
- Lényesen egyszerűbb elérni, hogy a hálózat konstans nulla függvényt tanuljon: csupa nulla súly.
- Ha az optimális leképezés az identitás közelében van: könnyebb megtalálni a konstans nullához viszonyítva.



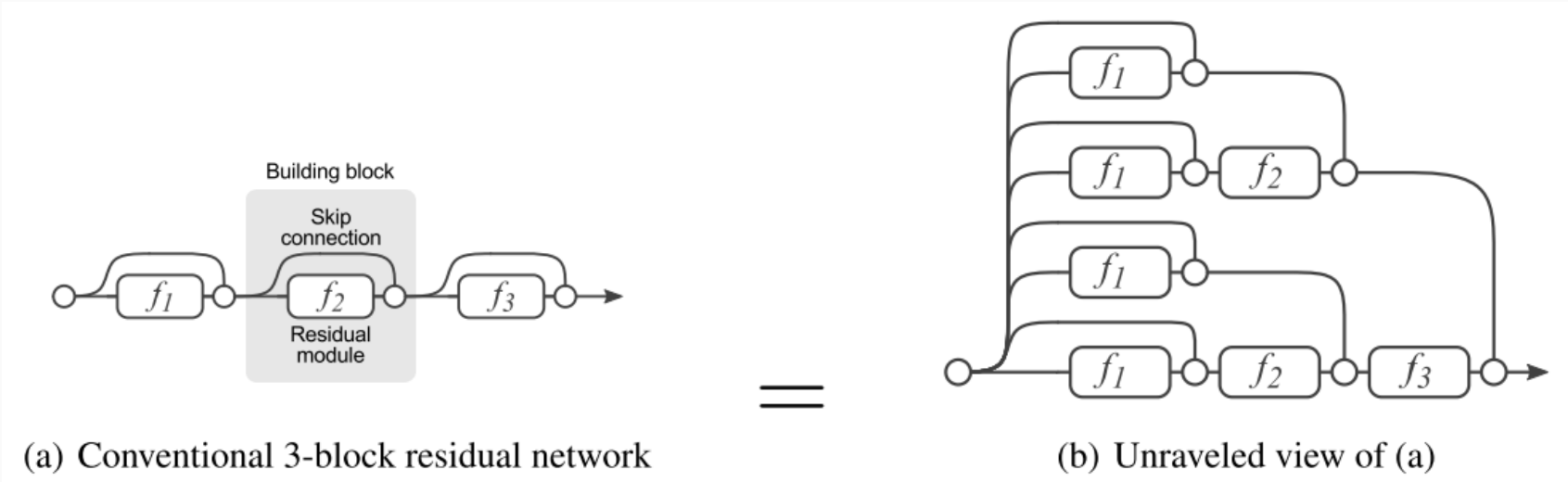
Reziduális hálózatok



- Nem jön létre új reprezentációs tér
- A gradiens exponenciálisan sok úton terjed
- A közvetlen kapcsolatokon keresztül erős gradiens tud áramlani

Reziduális hálózatok

- Tekinthetünk egy reziduális hálóra úgy, mint sok különböző mélységű hálózat együttesére

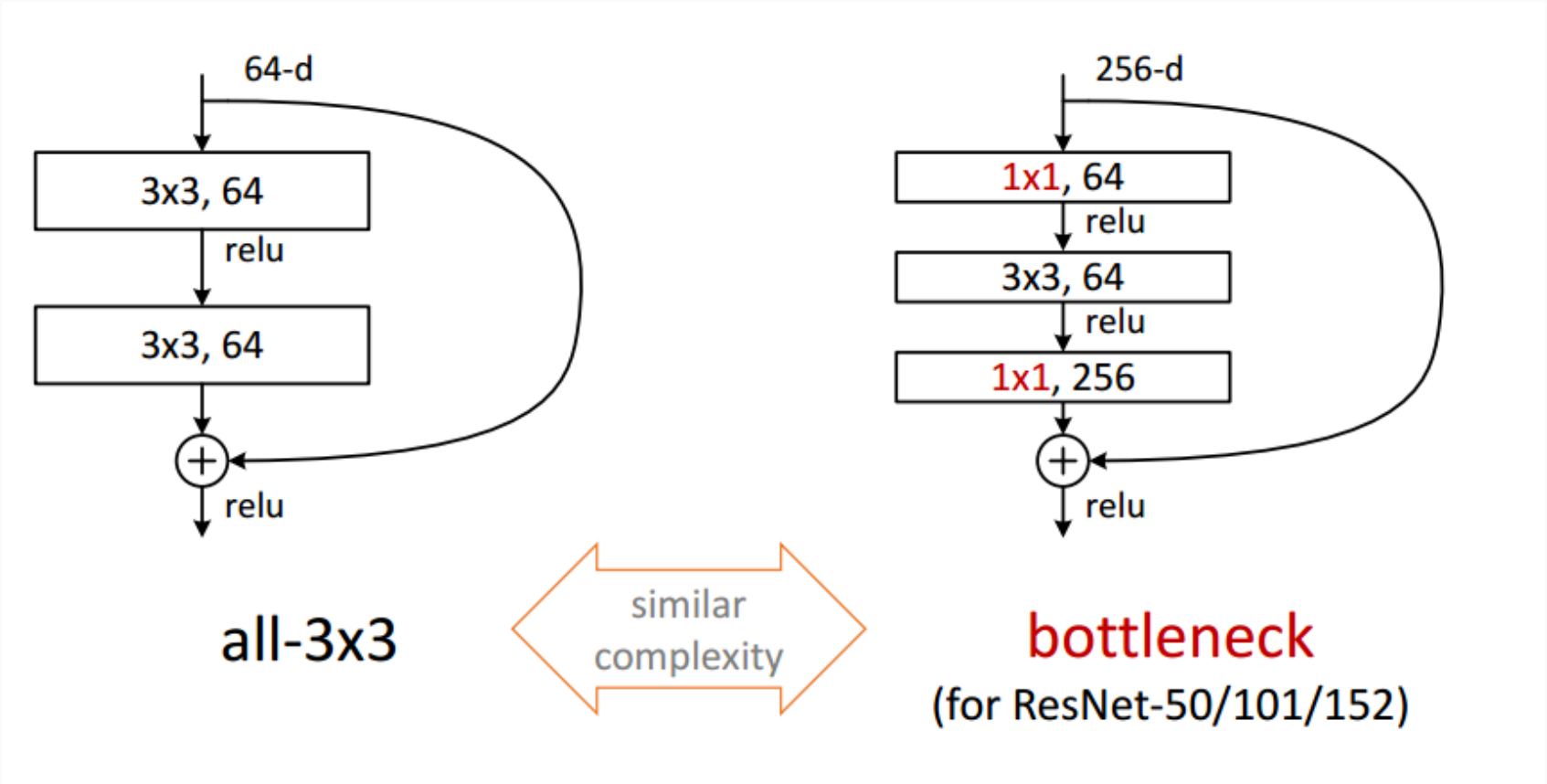


(a) Conventional 3-block residual network

(b) Unraveled view of (a)

Reziduális hálózatok

- Egy praktikus megfontolás: paraméterszám csökkentése
- Bottleneck reziduális blokk



Reziduális hálózatok

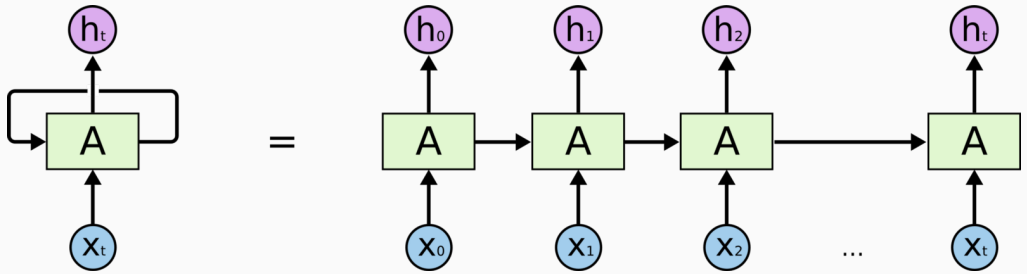
- Akár 1000 felett is tudnak javulást hozni új rétegek.
- Egy idő után nagyon kicsit növekedés nagyon sok paraméterrel
- A háló figyelmen kívül tudja hagyni a reziduális blokkokat, ahelyett, hogy hasznosítaná őket.
- A szélesség növelésével (a mélység rovására), ugyanannyi paraméterrel jobb eredményt tudunk elérni
- Wide Residual Networks

Reziduális hálózatok

- Egyszerű implementálni
- https://github.com/zsoltzombori/keras_fashion_mnist_tutorial/blob/master/fashion_mnist_resnet.py

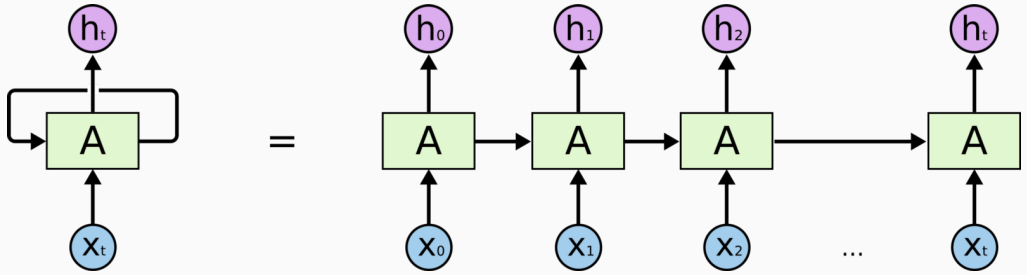
Rekurrens hálózatok

Rekurrens hálózatok



- Visszacsatolt hálózat
- Egymást követő adatpontok közti kapcsolatok kezelhetőek
- Idősorok, dinamikus rendszerek modellezésére alkalmas
- Egyfajta memóriát valósít meg
- Képes egy vagy több bemenetből egy vagy több kimenet előállítására

Rekurrens hálózatok



- Hasonlít egy olyan előre csatolt hálózathoz, amiben bizonyos rétegek paraméterei meg vannak osztva
- Tanítás: Backpropagation Through Time (BPTT)
- Használhatósága nagyban függ attól, hogy milyen léptékű időbeli függőségek vannak az adatokban
- Hosszú függőségek esetén: eltűnő-felrobbanó gradiensek problémája

Rekurrens hálózatok

- Long-Short-Term Memory Unit (LSTM): Hosszútávú függőségek kezelésére
  http://colah.github.io/posts/2015-08-Understanding-LSTMs/
- Keras LSTM tutorial
  http://adventuresinmachinelearning.com/keras-lstm-tutorial/
- Keras implementáció
  https://github.com/zsoltzombori/keras_fashion_mnist_tutorial/blob/master/keras_lstm.py