

Meeting 6 - Komentāri par tekstu

Apmācības laiks nevis ātrums

Katru eksperimentu veiksi 10x reizes

TĀBĪRĀS	
Ievads.....	3
Dziļā māšīnmācīšanās.....	3
Pamata arhitektūras.....	3
Atpakaļizplatīšanās algoritms.....	9
Apmācāmo parametru optimizācijas algoritmi.....	10
Metrikas.....	15
Hiperparametri.....	16
Attēlu klasifikācija.....	17
PyTorch vide.....	17
Metodoloģija.....	19
Datu ielādes metodes.....	19
Failu sistēma.....	19
NumPy mmap.....	19
CuPy mmap.....	19
HDF5.....	20
Zarr.....	20
Datu kopas.....	20
CIFAR10.....	20
Tiny ImageNet.....	20
Images with people wearing masks.....	20
Apmācības protokols.....	21
Rezultāti.....	22
Tālākie pētījumi.....	23
Secinājumi.....	24
Bibliogrāfija.....	25

apjoms 1:1

Datu formātu attīstības vēsture

balstīties uz tiem kurus izmanto + citus, kurs nemaz nepēti

salīdzināt ar tensorflow, CNTK (static graph vs dynamic graph)

Failu metode

Testēšanas protokols - vidējais epocha ātrums, kopējais apmācības ātrums, atmiņas noslodze uz GPU, uz RAM, GPU noslodze, CPU noslodze

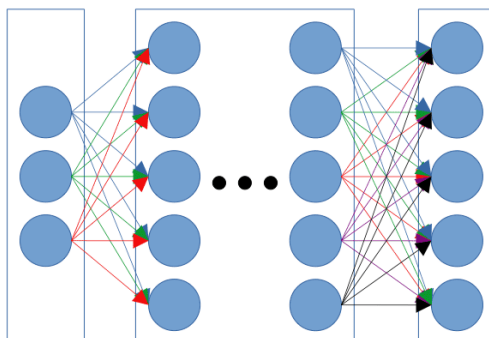
numurācija

Ievads

Dziļā māšīnmācīšanās

Pamata arhitektūras

Makslīgais neironu tīkls sastāv no viena ieejas slāņa, viena vai dažiem slēptiem slāņiem, kā arī viena izejas slāņa. Katrs slānis satur vienu vai vairākus neironus. Tie ir saistīti ar blakusslāņu neironiem.



Eksistē tīkli, kuru neironi ir pilnīgi saistīti ar visiem neironiem blakusslāņos, bet var būt saistīti tikai daļēji. Katrai saitei ir savs svars un nobīde. Tie ir apmācāmie parametri. Apmācīšanās procesa mērķis ir atrast tādus svarus un nobīdes visām saitēm, ar kuriem tīkls veidotu pareizus rezultātus izejas slānī.

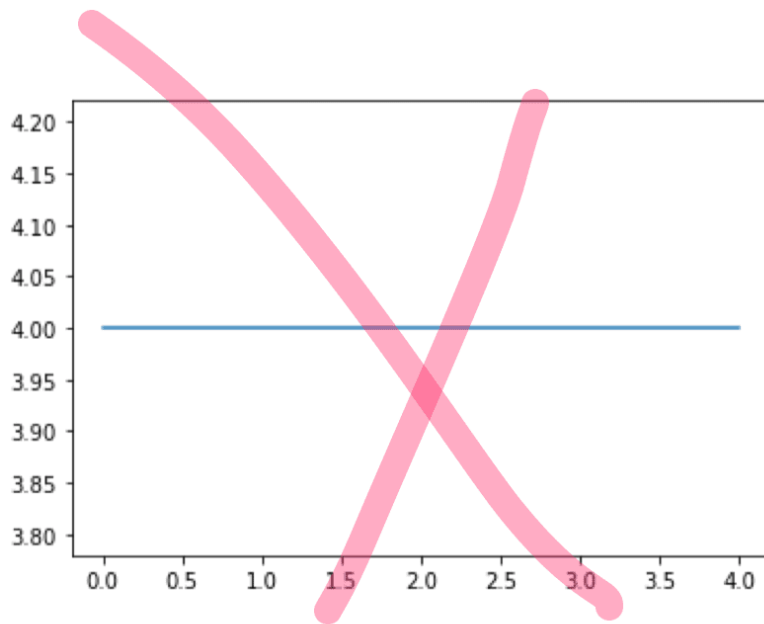
Word (fields) numerācija attēliem

Linārie slāņi

Stingras formāla teorija par to, kā atlasīt neironu tīkla slāņus un konfigurāciju neeksistē, un, lai gan vienīgais veids, kā noregulēt dažus hiperparametrus, ir tikai mēģinājumu un kļūdu metode (piemēram, meta-apmācība), taču joprojām pastāv dažas heuristikas, vadlīnijas un teorijas, kas joprojām var palīdzēt ievērojami samazināt piemērotu arhitektūru meklēšanas telpu.

Lineārais nobīdes slānis

$$y=b$$



Šis slānis būtībā iemācās konstanti. Tas spēj apgūt nobīdi, novirzi, sliekšni vai vidējo vērtību. Ja neironu tīklu izveidot tikai no šī slāņa un apmācīt to, izmantojot datu kopu, vidējās kvadrātiskās kļūdas zudums tiks šim slānim konverģēt uz izejas vidējo vērtību.

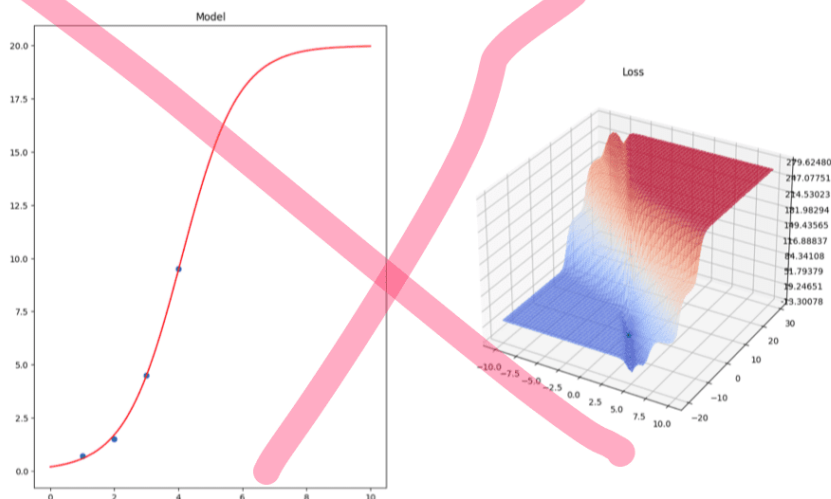
Piemēram, ja ir šāda datu kopa {1, 1, 2, 2, 3, 3} un neironu tīkls tiek spiests to saspiest līdz unikālai vērtībai b , loģiskākā konverģence būs ap vērtību $b=2$ (kas ir datu kopas vidējais lielums, lai samazinātu zaudējumus līdz maksimumam. Jebkura vērtība, kas pārsniedz šo nobīdi, būs pozitīva.

$$\frac{\partial \mathcal{L}}{\partial \theta_0} = \frac{\partial \mathcal{L}}{\partial b} = \frac{\partial \mathcal{L}}{\partial a} \cdot \frac{\partial a}{\partial b} = \frac{\partial \mathcal{L}}{\partial a} \cdot 1 = \frac{\partial \mathcal{L}}{\partial a} \cdot (1 - \frac{1}{1+e^{-a}}) \cdot 1$$

$$\frac{\partial \mathcal{L}}{\partial \theta_1} = \frac{\partial \mathcal{L}}{\partial W} = \frac{\partial \mathcal{L}}{\partial a} \cdot \frac{\partial a}{\partial W} = \frac{\partial \mathcal{L}}{\partial a} \cdot \frac{\partial}{\partial W} \left(\frac{e^{-a}}{(1+e^{-a})^2} \right) \cdot x = \frac{1}{1+e^{-a}} \cdot \left(1 - \frac{1}{1+e^{-a}} \right) \cdot x$$

Zuduma plakne lineāram modelim ar sigmoida aktivizācijas funkciju:

w=1.1346469402126882 b=-4.6482628586296135 loss=0.014018127818400896 learning_rate=0.00014900000000000007



Kubiskajai funkcijai $b+W \cdot x^3$ bez aktivizācijas funkcijas, laba lieta šajā gadījumā zuduma funkcijas atvasinājums dW, db ir tas pats kā lineārajai funkcijai. Un dx ir:

bloka beigās dažādu ceļu izejas tiek savienotas.

PyTorch vide

PyTorch ir optimizēta mašīnmācīšanas bibliotēka, kura atvieglo darbu gan ar procesoru (CPU), gan ar videokārti (GPU). Programēšanas valoda ir Python.

Modeļa definēšanas piemērs PyTorch vidē:

```
class Model(Module):
    def __init__(self):
        super().__init__()
```

**Pytorch Datasets VS
TF Record**

```

self.layers = Sequential(
    Linear(in_features=13, out_features=10, device=device),
    Tanh(),
    Linear(in_features=10, out_features=5, device=device),
    LeakyReLU(),
    Linear(in_features=5, out_features=1, device=device)
)

def forward(self, x):
    y_prim = self.layers.forward(x)

    return y_prim

```

**ar dažiem
piemēriem**

**+ tekstā aprakstīt
atšķirības starp
frameworks**

18

Datu kopas

CIFAR10

**Vidējais izmērs , min, max, median
katram attēlam**



Attēls 11: CIFAR-10 datu kopnes

8 bit^{ti} jpeg^{rs}

CIFAR-10 datukopa satur 60 tūkstošus 32x32 krāsainas bildes, kuras ir sadalītas 10 klasēs, katrai klasei atbilst 6000 bildes. Datukopa ir sadalīta uz 50000 bildēm apmācībai un 10000 bildēm pārbaudei.

Tiny ImageNet



Attēls 12: Tiny ImageNet datu kopnes attēlu piemērs

Šī datu kopa satur 100000 krāsainus attēlus, sadalītus 200 klasēs (pa 500 attēliem priekš katrai klasei) un samazinātus līdz 64x64 pikseliem. Katra klase satur 500 apmācības, 50 validācijas un 50 testa attēlus.

Failu metodes: JPEG, citi formāti (JPEG2000 vai kaut kas netika sabojāts)

JPEG => PIL => ushort => float32

kas notiek, ja dataset dati tiek sagatavoti kā float16 nevis float32

1. Apmācība ir svarīga, lai noskaidrotu kā mijiedarbojās datu ielādes metožu caching ar pytorch apmācības procesu
2. Pārbaudīt vai pareizi implemntēta datu ielādes metode

About 9,040 results for "Neural networks and deep learning: a textbook"

Fields of Study ▾

Date Range

Neural Networks and Deep Learning

C. Aggarwal · Computer Science · 26 Jul 2016

TLDR The neural networks and deep learning are the most important topics in the digital library. [Expand](#)

👍 173 PDF · 📖 Save · 🔔 Alerts

DEEP LEARNING NEURAL NETWORKS

D. Graupe · Computer Science · 7 Jul 2016

TLDR This comprehensive textbook is intended for use as a one semester

Cite Paper

BibTex

MLA

APA

Chicago

```
@inproceedings{Graupe2016DEEPLN,  
  title={DEEP LEARNING NEURAL NETWORKS: DESIGN AND CASE STUDIES},  
  author={Daniel Graupe},  
  year={2016}  
}
```

📄 Copy