神经网络超参数

赵鉴

简介

超参

- 神经网络结构
- 神经网络优化器
- 神经网络激活函数
- Batchsize
- 损失函数loss function

神经网络结构

- 主线
  - Alexnet->Vggnet,Googlenet->Resnet->Densenet->Senet
- 分支
  - 谷歌
    - ✓ Inception系列
    - ✓ Mobilenet系列
    - ✓ Nasnet系列
    - ✓ Deeplab系列
  - 旷视
    - ✓ Shufflenet系列
  - MSRA：
    - ✓ Deformable系列
    - ✓ IGC系列
  - Pjreddie
    - ✓ Yolo系列

Resnet

- 2015年LSVRC 2012 分类竞赛冠军
- 2016 CVPR best paper
- 思考：
  - 假如你发现了Resnet比一般CNN效果好，你会怎么写这个paper
    - ✓ 非常苦恼，因为不知道为什么Resnet效果比一般CNN要好
    - ✓ 容易被review质疑，是不是只适用于特定任务

Resnet

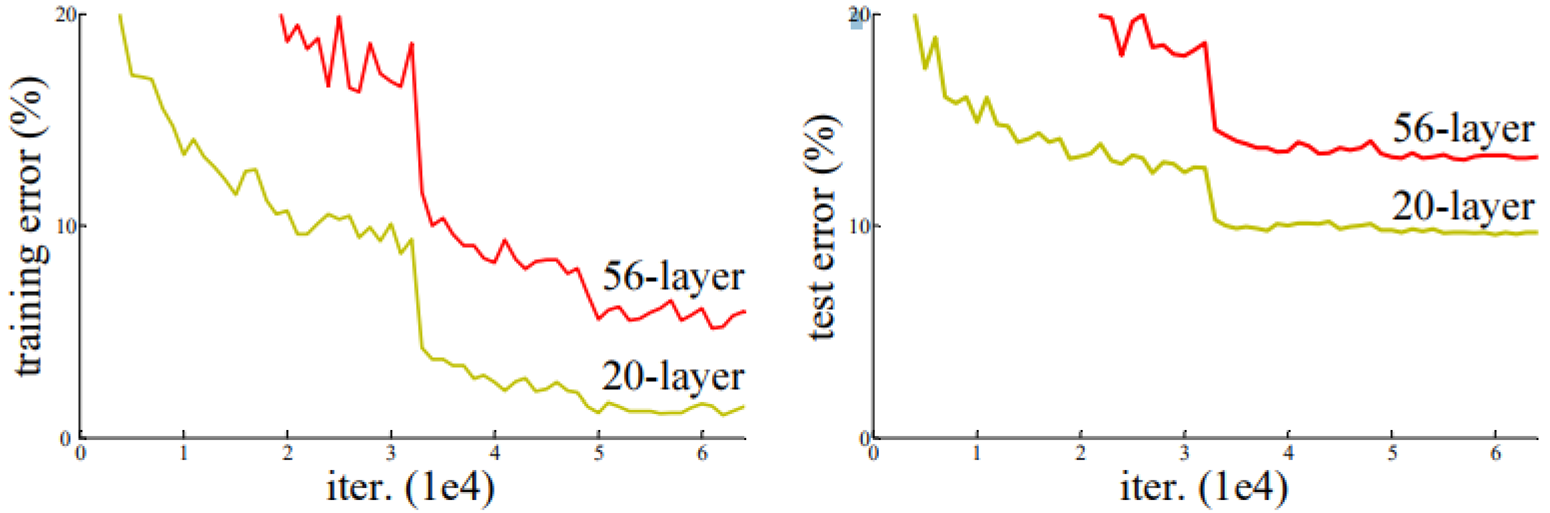


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

Resnet

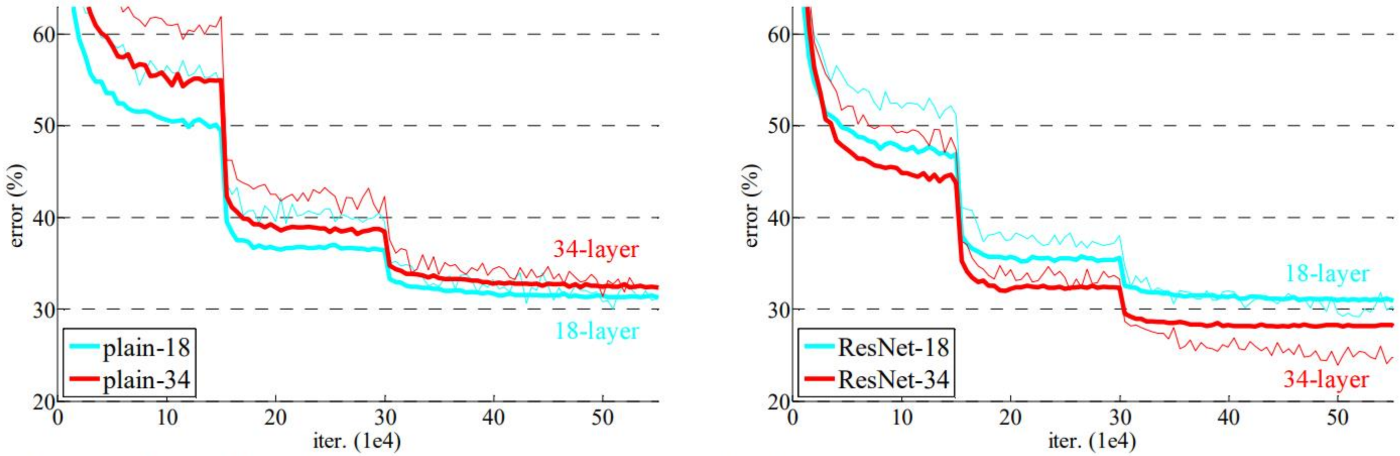


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

Resnet

- 问题：
  - Resnet为什么泛化效果好
- 问题转移：
  - Resnet结构随深度增加，效果变好不变差
- 原因：
  - 梯度消失，梯度爆炸
- 结果：
  - Best paper！！！

Adam

- 问题：
  - 为什么Adam训练神经网络效果好
- 问题转移：
  - Adam训练凸问题收敛
- 原因：
  - 一堆数学推导（错）
- 结果：
  - 2015 ICLR best paper！！！

AMSGRAD（Adam变种）

- 问题：
  - 为什么AMSGRAD训练神经网络效果好
- 问题转移：
  - Adam对于某些凸问题不收敛
  - AMSGRAD对于凸问题是收敛的
- 原因：
  - 一堆数学推导
- 结果：
  - 2018 ICLR best paper！！！

Densenet

- **Each layer has direct access to the gradients from the loss function and the original input signal, leading to an implicit deep supervision**

- 结果：
  - 2017 CVPR best paper

- 影响:
  - 开启了sota的浪潮

Nasnet

- ☐ Neural Architecture Search With Reinforcement Learning

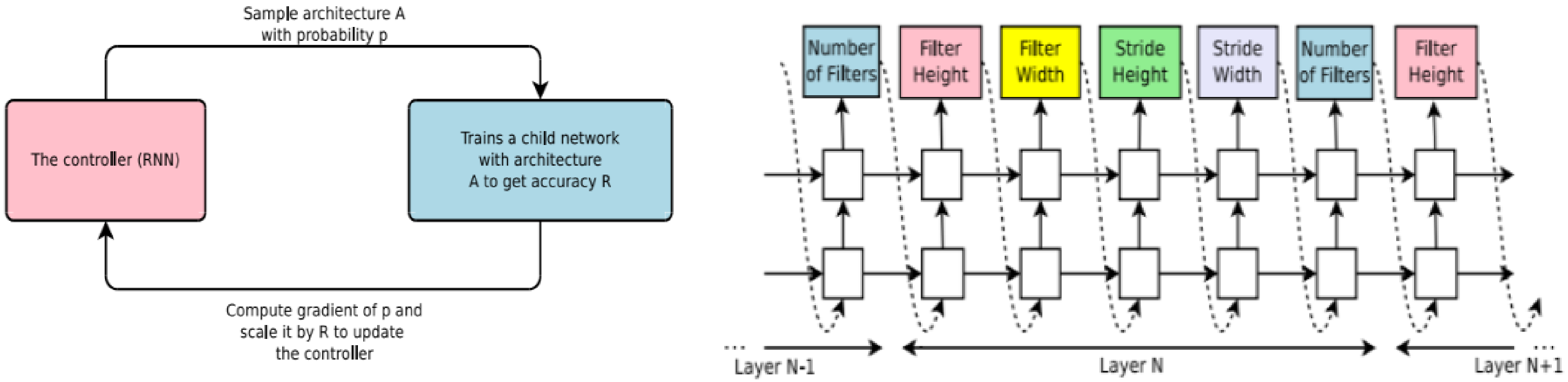
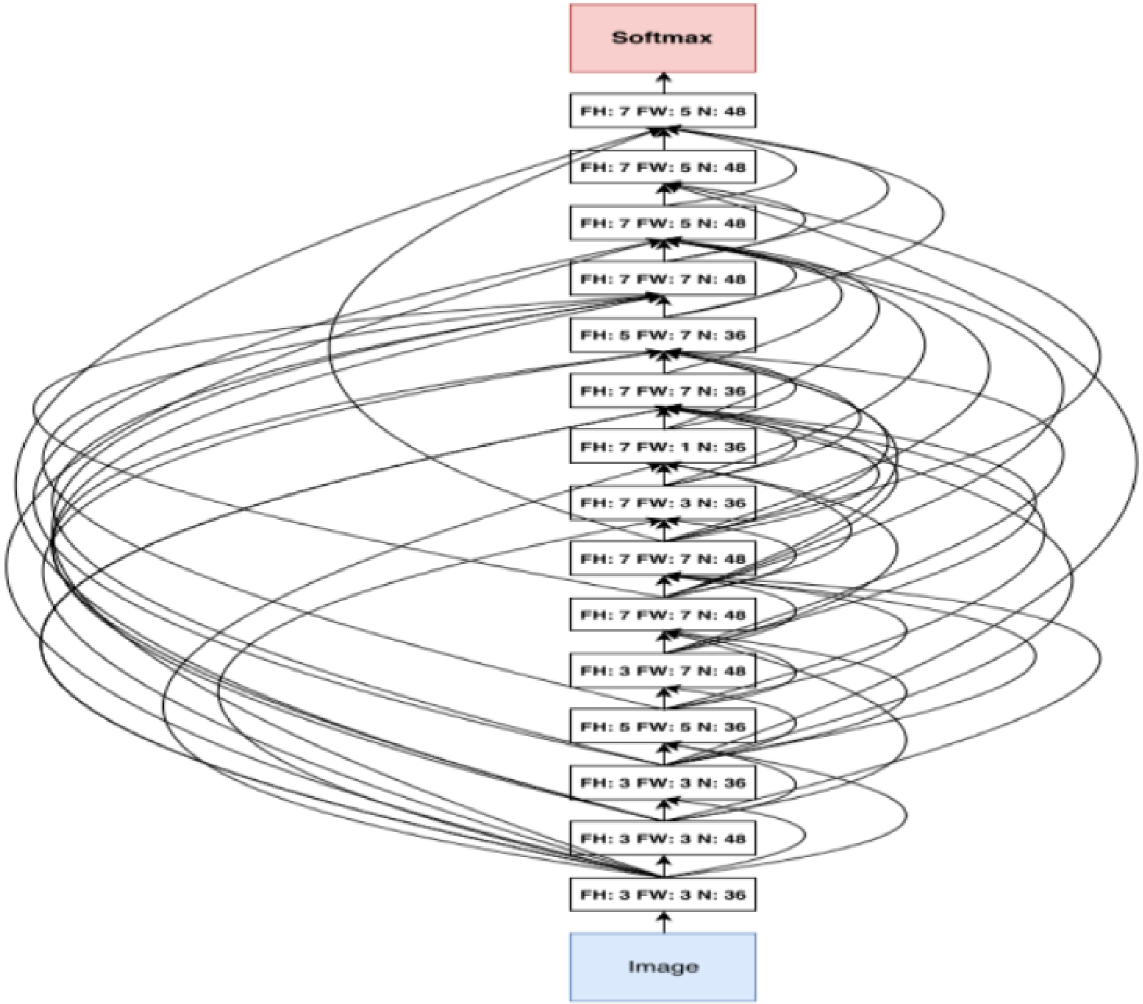


Figure 1: An overview of Neural Architecture Search.

Nasnet



Softmax
FH: 7 FW: 5 N: 48
FH: 7 FW: 5 N: 48
FH: 7 FW: 5 N: 48
FH: 7 FW: 7 N: 48
FH: 5 FW: 7 N: 36
FH: 7 FW: 7 N: 36
FH: 7 FW: 1 N: 36
FH: 7 FW: 3 N: 36
FH: 7 FW: 7 N: 48
FH: 7 FW: 7 N: 48
FH: 3 FW: 7 N: 48
FH: 5 FW: 5 N: 36
FH: 3 FW: 3 N: 36
FH: 3 FW: 3 N: 48
FH: 3 FW: 3 N: 36
Image

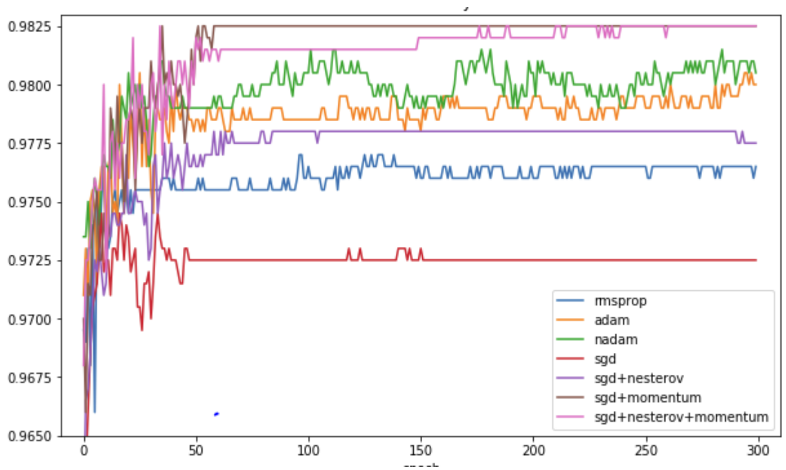
all Keras optimizers

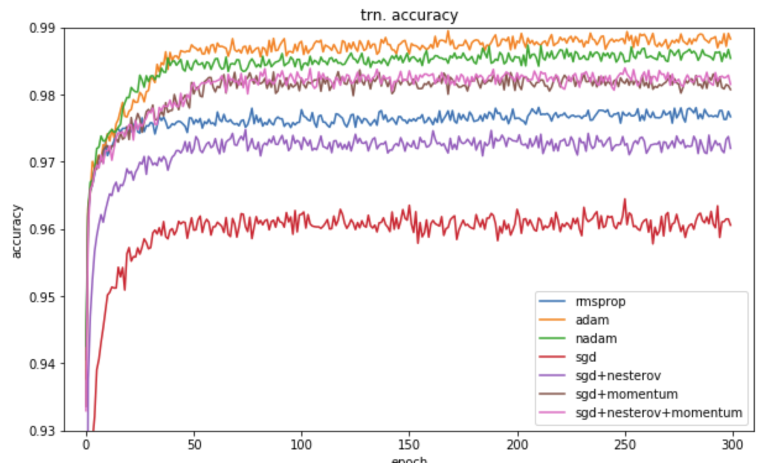
- SGD
- RMSprop
- Adagrad
- Adadelta
- Adam
- Adamax
- Nadam

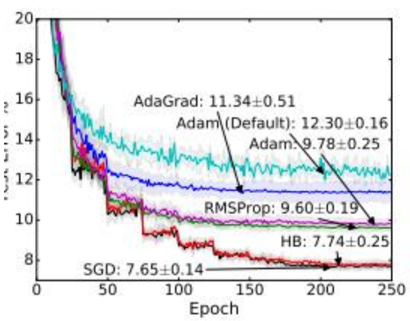
$$W = W - LearningRate * dW$$

	dW	**Learning rate**
SGD	/	/
SGD + momentum	Momentum	/
SGD + nesterov	Nesterov	/
Adagrad	/	L2
RMSprop	/	Average L2
Adadelta	/	*
Adam	Momentum	Average L2
Adamax	Momentum	Average L∞
Nadam	Nesterov	Average L2

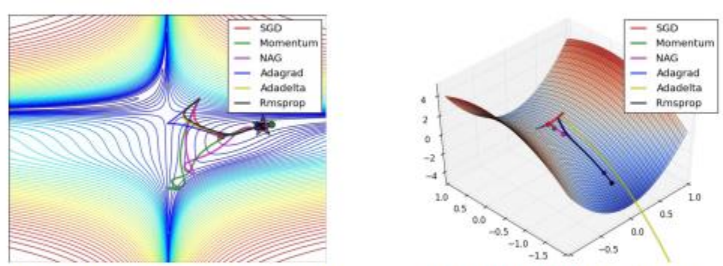
performance







(b) CIFAR-10 (Test)



(a) SGD optimization on loss surface contours

(b) SGD optimization on saddle point

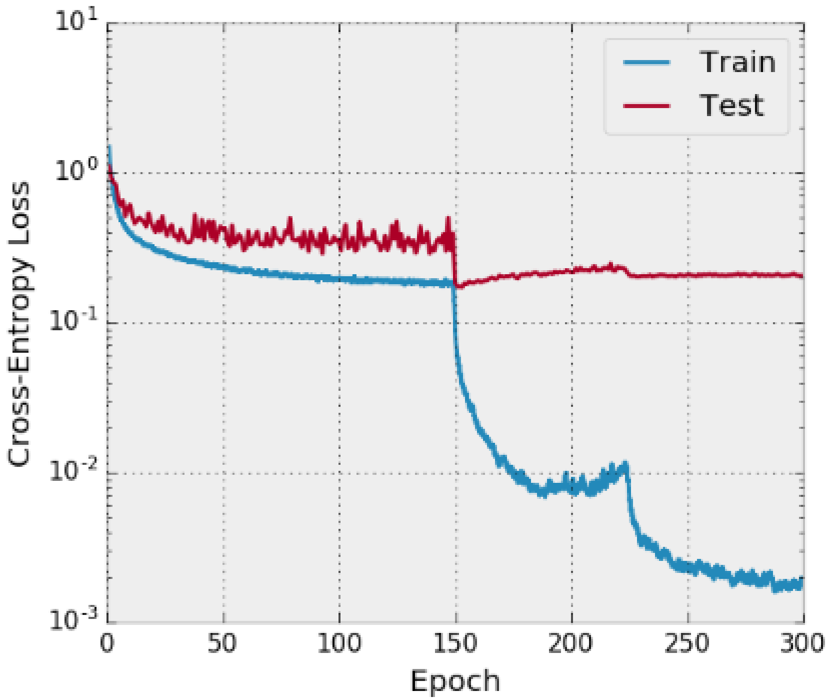
Figure 4: Source and full animations: Alec Radford

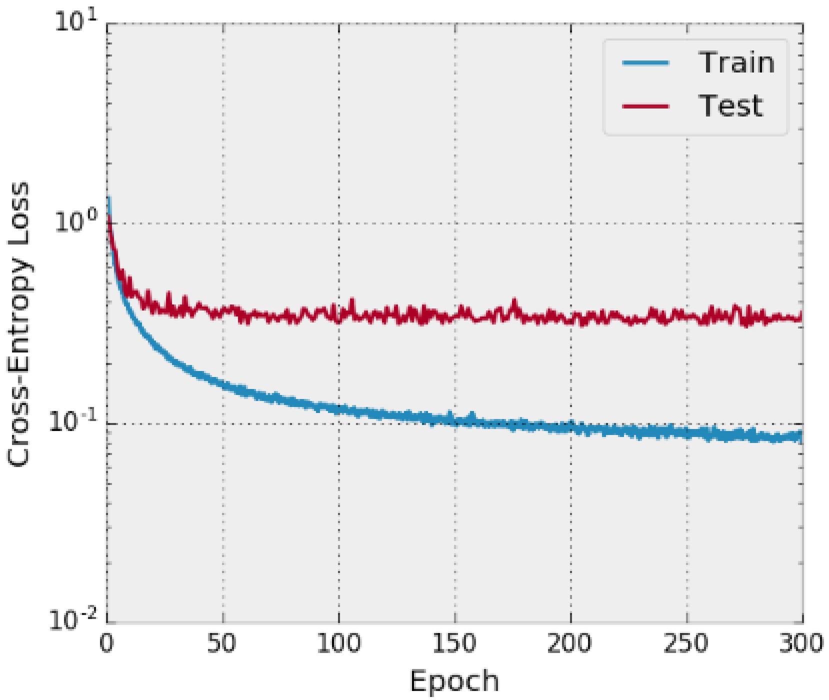
An overview of gradient descent optimization algorithms∗
https://arxiv.org/pdf/1609.04747.pdf
The marginal value of adaptive gradient methods in machine learn
arXiv preprint arXiv:1705.08292, 2017

优化器

- 假如神经网络是个凸函数
  - 不同优化器最后优化的值确定
  - 不同优化器只有收敛速度的问题
- 但是神经网络是非凸的
  - 不同优化器收敛到不同的局部最小值
  - 不同的局部最小值泛化能力不同
- 总之，不同优化器，训练一个相同的神经网络，达到相同的train loss，test accuracy差距很大
  - 无法描述
  - 产生了一系列调参黑科技
  - ½ epoch lr/=10; ¾ epoch lr/=10; (7/8 epoch lr/=10)

优化器


Train
Test
Cross-Entropy Loss
Epoch
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
0
50
100
150
200
250
300


Train
Test
Cross-Entropy Loss
Epoch
10^{1}
10^{0}
10^{-1}
10^{-2}
0
50
100
150
200
250
300

激活函数

全靠猜

Neural [*] Search with Reinforcement Learning

- Paper 1: [*] = Architecture ICLR2017
- Paper 2: [*] = Optimizer ICML2017
- Paper 3: [*] = Activation Function ICLR2018

Neural [*] Search with Reinforcement Learning

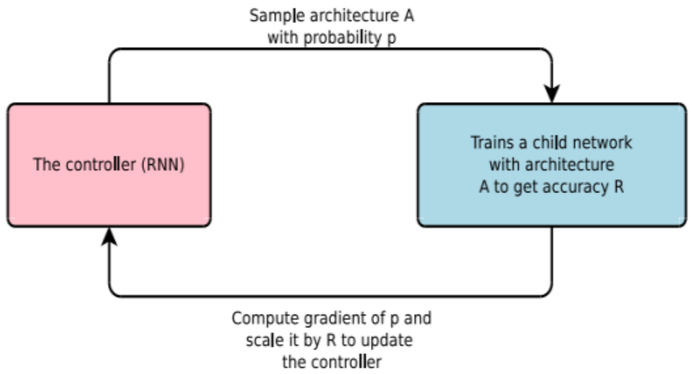
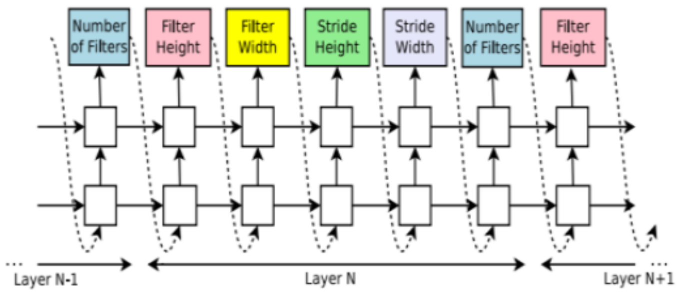


Figure 1: An overview of Neural Architecture Search.



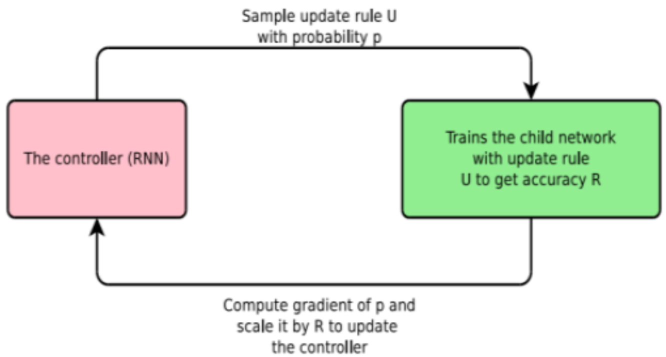
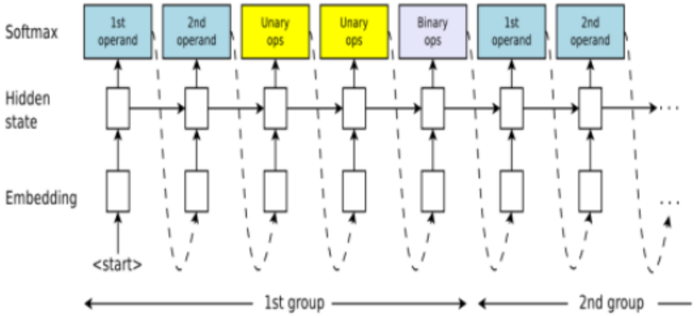


Figure 1. An overview of Neural Optimizer Search.



论文图表对比

总语

- 冲sota越来越难

- 调参这件事可能会被大量计算资源替代
  - AutoML
  - AutoKeras
  - AutoAzure

- 目标可能要回归数学理论

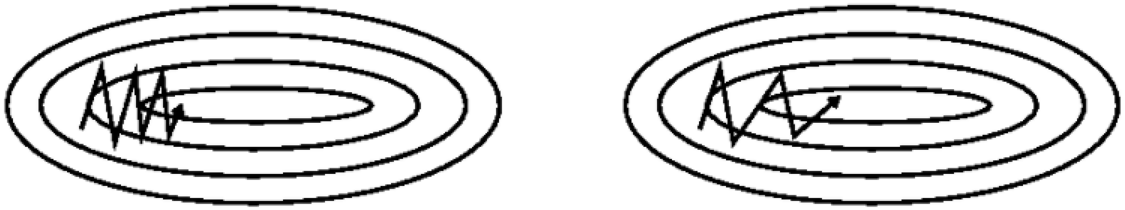
SGD

- $SGD(lr = 0.01, momentum = 0.0, decay = 0.0, nesterov = \boldsymbol{False})$
  - **lr**: Learning rate.
  - **momentum**: Parameter updates momentum.
  - **decay**: Learning rate decay over each update.
  - **nesterov**: Whether to apply Nesterov momentum.

  - $W = W - \alpha dW$
  - Disadvantages:
    - ✓ Converge slowly(momentum, nesterov)
    - ✓ The learning rate unchanged and is the same for each dimension(Adagrad …)
    - ✓ converge to a local optimum and saddle point

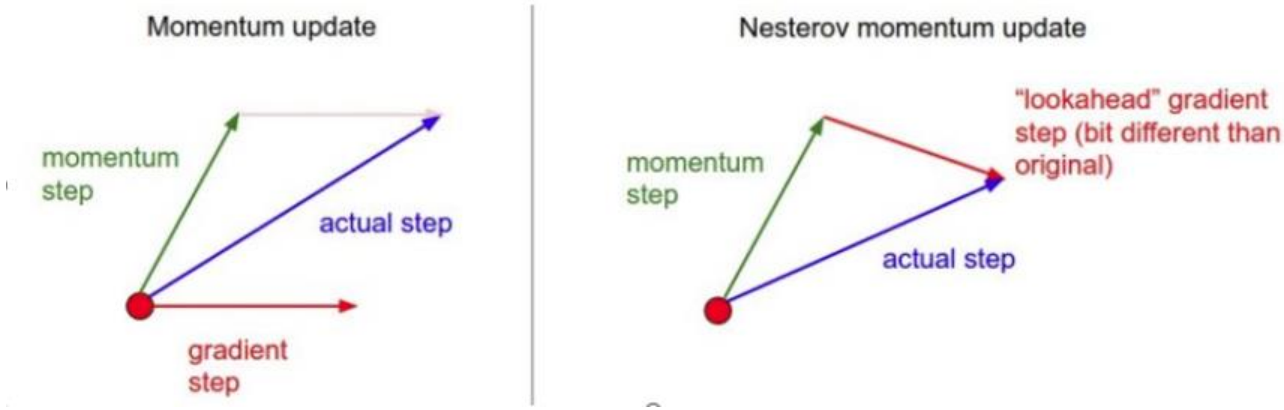
SGD + momentum (average L1)

- $V = \beta V + \alpha dW$
- $W = W - V$
- exponential weighted average
  - The weight of each value decreases exponentially with time
  - Only need to keep V



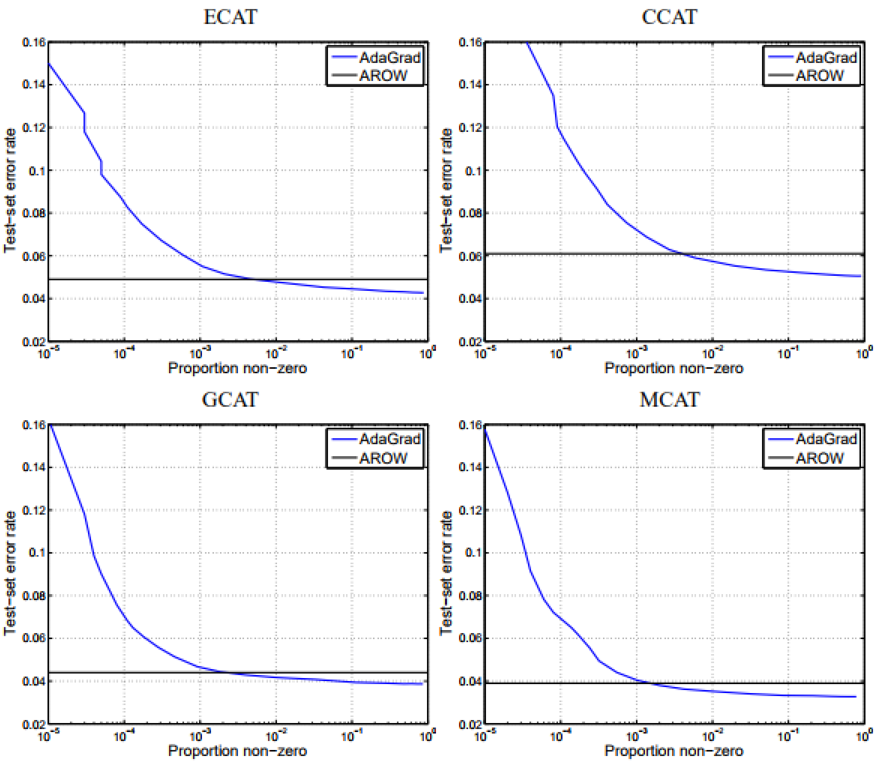
SGD + nesterov + momentum

- $V_t = \beta V_{t-1} + \alpha \nabla_\theta J(\theta - \beta V_{t-1})$
- $W = W - V_t$
- stronger theoretical converge guarantees for convex functions
- in practice works slightly better than standard momentum



Adagrad (L2)

- $G += (dW)^2$
- $W = W - \alpha * \frac{dW}{(\sqrt{G} + \epsilon)}$
- Disadvantages:
  - stops learning too early(R
  - Different units(Adadelta)



[Adaptive Subgradient Methods for Online Learning and Stochastic Optimization]

RMSprop (average L2)

- $G = \beta * G + (1 - \beta) * (dW)^2$
- $W = W - \alpha * \frac{dW}{(\sqrt{G} + \epsilon)}$

Adadelta

- $G = \beta * G + (1-\beta) * (dW)^2 => RMS(dW) = \sqrt{G + \epsilon}$
- RMSprop :$W = W - \alpha * \dfrac{dW}{RMS(dW)}$
- Adadelta:
  - $W_t = W_t - \dfrac{RMS(\Delta W_{t-1})}{RMS(dW_t)} * \mathrm{dW_t}$
  - $\Delta W_t = \dfrac{RMS(\Delta W_{t-1})}{RMS(dW_t)} * \mathrm{dW_t}$

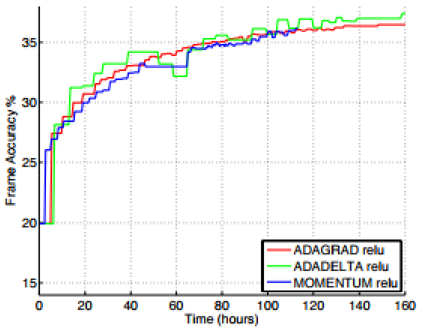


Fig. 4. Comparison of ADAGRAD, Momentum, and ADADELTA on the Speech Dataset with 200 replicas using rectified linear nonlinearities.

•Adadelta - an adaptive learning rate method

Adam = SGD + momentum + RMSprop

- RMSprop:
  - $G = \beta * G + (1-\beta) * (dW)^2$
  - $W = W - \alpha * \frac{dW}{(\sqrt{G} + \epsilon)}$
- Momentum:
  - $V = \beta V + \alpha dW$
  - $W = W - v_{dW}$
- Adam:
  - $G = \beta_1 * G + (1-\beta_1) * (dW)$
  - $G' = \frac{G}{1-\beta_1^t}$
  - $V = \beta_2 V + (1-\beta_2) dW$
  - $V' = \frac{V}{1-\beta_2^t}$
  - $W = W - \alpha * \frac{V\prime}{(\sqrt{G\prime} + \epsilon)}$

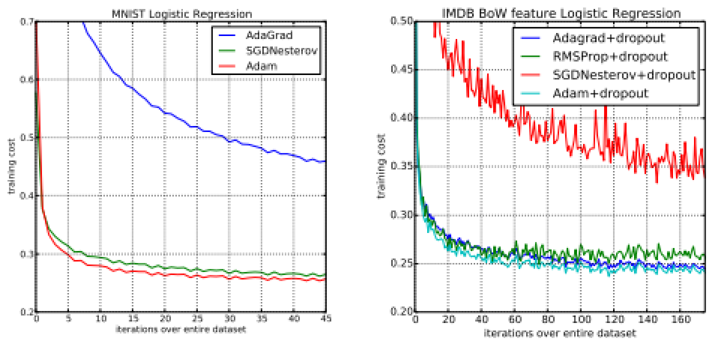


Figure 1: Logistic regression training negative log likelihood on MNIST images and IMDB movie reviews with 10,000 bag-of-words (BoW) feature vectors.

- Adam – A Method for Stochastic Optimization

Adamax (average L_∞)

- Adam:
  - $G = \beta_1 * G + (1 - \beta_1) * (dW)^2$
  - $V = \beta_2 V + (1 - \beta_2) dW$
  - $W = W - \alpha * \frac{V}{(\sqrt{G} + \epsilon)}$
- Adamax:
  - $G = \beta_1^\infty * G + (1 - \beta_1^\infty) * (dW)^\infty \Rightarrow \ u = \max(\beta_1 * G, |dW|)$
  - $V = \beta_2 V + \beta_2 dW$
  - $W = W - \alpha * \frac{V}{(\sqrt[\infty]{G} + \epsilon)} \Rightarrow W = W - \alpha * \frac{V}{u}$

Nadam = SGD + nesterov + RMSprop

- Adam:
  - $G = \beta_1 * G + (1-\beta_1) * (dW)^2$
  - $V = \beta_2 V + (1-\beta_2) dW$
  - $W = W - \alpha * \frac{V}{(\sqrt{G} + \epsilon)}$
- Nadam:
  - $G = \beta_1 * G + (1-\beta_1) * (\nabla_\theta J(\theta - \beta I$
  - $V = \beta_2 + (1-\beta_2) * \nabla_\theta J(\theta - \beta V_{t-1})$
  - $W = W - \alpha * \frac{V}{(\sqrt{G} + \epsilon)}$

Word2Vec

Batch

	GD	Mom	NAG
Test loss	.368	.361	.358
	RMS	Adam	Nadam
Test loss	.316	.325	**.284**
	Maxa	A-max	N-max
Test loss	.346	.356	.355

Image Recognition

	GD	Mom	NAG
Test loss	.0202	.0263	.0283
	RMS	Adam	Nadam
Test loss	**.0172**	.0175	.0183
	Maxa	A-max	N-max
Test loss	.0195	.0231	.0204

LSTM Language Model

	GD	Mom	NAG
Test perp	100.8	**99.3**	99.8
	RMS	Adam	Nadam
Test perp	106.7	111.0	105.5
	Maxa	A-max	N-max
Test perp	106.3	108.5	107.0

•Incorporating Nesterov Momentum into Adam