یادگیری عمیق با تنسورفلو و کراس در پایتون

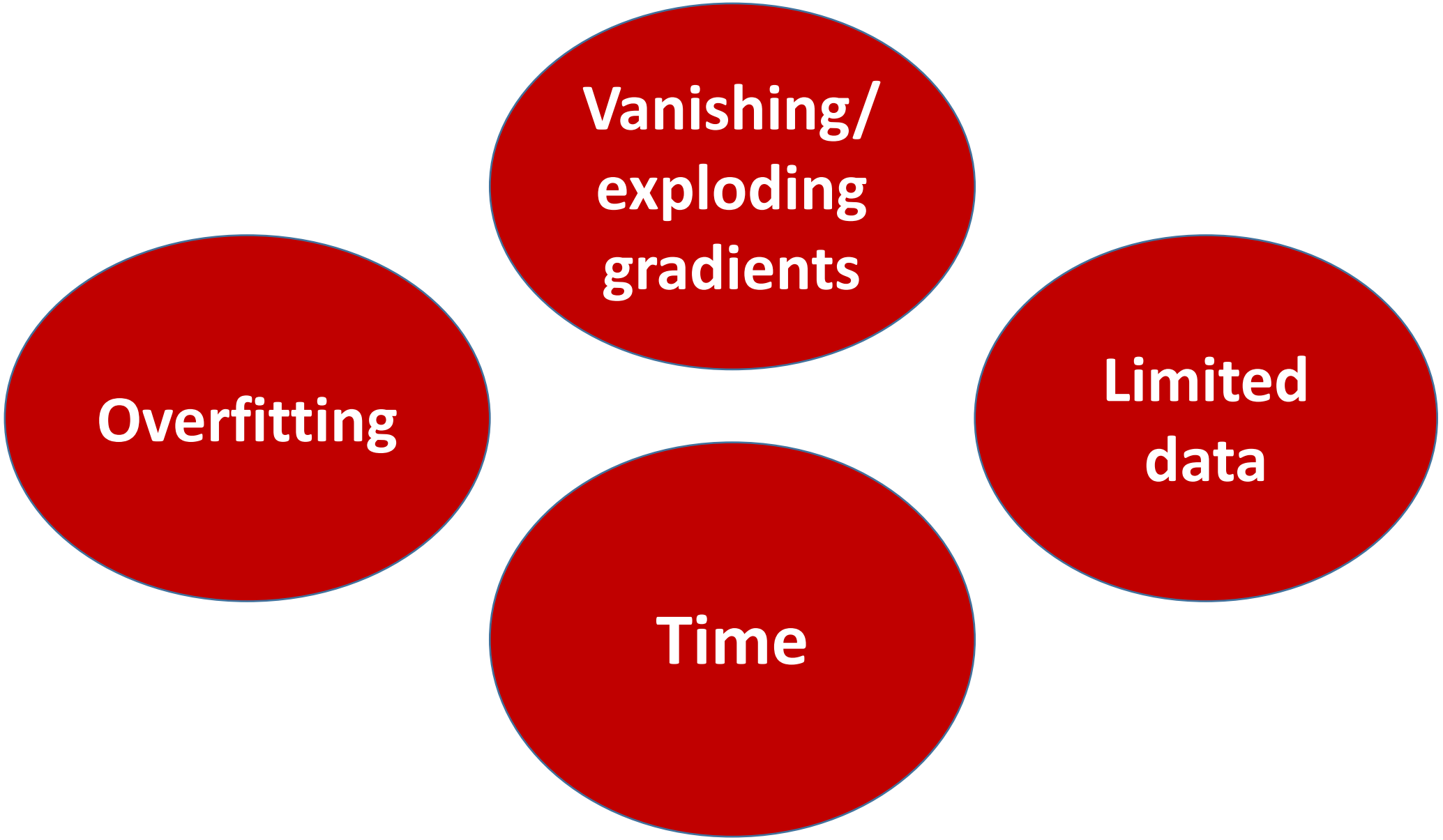
فصل سوم: آموزش شبکه های عصبی عمیق

پژمان اقبالی

PhD Student in Biomechanics

EPFL

چالش های آموزش شبکه های عصبی عمیق



چالش های آموزش شبکه های عصبی عمیق

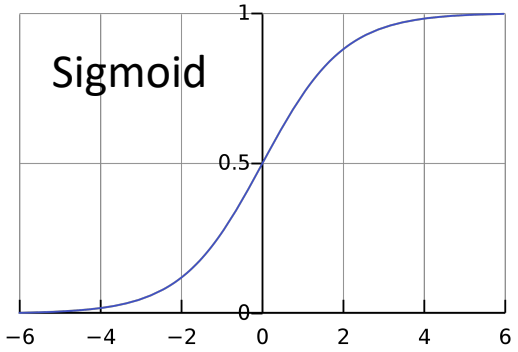
Vanishing/exploding gradients

- ❑Initialization
- ❑Nonsaturating Activation Functions
- ❑Batch Normalization
- ❑Gradient Clipping

چالش های آموزش شبکه های عصبی عمیق

Vanishing/exploding gradients

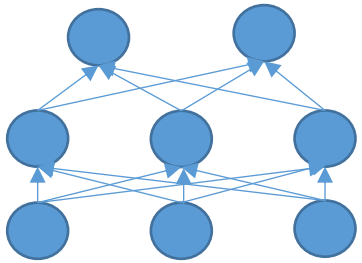
❑Initialization



wikipedia.org/wiki/Activation_function

Understanding the difficulty of training deep feedforward neural networks

Xavier Glorot **Yoshua Bengio**
DIRO, Université de Montréal, Montréal, Québec, Canada



$\rightarrow \begin{cases} fan_{in} = 3 \\ fan_{out} = 2 \\ fan_{avg} = \dfrac{fan_{in} + fan_{out}}{2} \end{cases}$

Glorot Initialization

- Normal distribution with mean 0 and $\sigma^2 = \dfrac{1}{fan_{avg}}$
- Uniform distribution between $-r$ and $+r$, with $r = \sqrt{\dfrac{3}{fan_{avg}}}$

Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia La-guna Resort, Sardinia, Italy.

چالش های آموزش شبکه های عصبی عمیق

Vanishing/exploding gradients

❑Initialization

Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun

arXiv:1502.01852

چالش های آموزش شبکه های عصبی عمیق

Vanishing/exploding gradients

- **Initialization**

	Activation functions	σ^2 (Normal)
Glorot	tanh, logistic, softmax	$1/fan_{avg}$
He	RELU (and variants)	$2/fan_{in}$
LeCun	SELU	$1/fan_{in}$

چالش های آموزش شبکه های عصبی عمیق

Vanishing/exploding gradients

❑ Nonsaturating Activation Functions

Empirical Evaluation of Rectified Activations in Convolutional Network

Bing Xu, Naiyan Wang, Tianqi Chen, Mu Li

- ❑ Leaky RELU (α)
- ❑ Randomized leaky RELU (RRELU)
- ❑ Parametric leaky RELU (PRELU)

arXiv:1505.00853

چالش های آموزش شبکه های عصبی عمیق

Vanishing/exploding gradients

❑ Nonsaturating Activation Functions

Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)

Djork-Arné Clevert, Thomas Unterthiner, Sepp Hochreiter

arXiv:1511.07289

چالش های آموزش شبکه های عصبی عمیق

Vanishing/exploding gradients

❑ Nonsaturating Activation Functions

Self-Normalizing Neural Networks

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, Sepp Hochreiter

- ❑ **Scaled ELU (SELU)**

arXiv:1706.02515

چالش های آموزش شبکه های عصبی عمیق

Vanishing/exploding gradients

❑ Nonsaturating Activation Functions

SELU > ELU > leaky RELU > RELU > tanh > logistic

Not self normalizing: ELU

Speed: RELU

چالش های آموزش شبکه های عصبی عمیق

Vanishing/exploding gradients

❑ Batch Normalization

Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift

Sergey Ioffe, Christian Szegedy

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

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

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

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

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

arXiv:1502.03167

چالش های آموزش شبکه های عصبی عمیق

Vanishing/exploding gradients

❑ Gradient Clipping

On the difficulty of training Recurrent Neural Networks

Razvan Pascanu, Tomas Mikolov, Yoshua Bengio

arXiv:1211.5063

چالش های آموزش شبکه های عصبی عمیق

Limited data → **لایه های از قبل آموزش دیده**

- **Transfer learning**
- **What Else?**
  - **Unsupervised pretraining**

چالش های آموزش شبکه های عصبی عمیق

Time → **Optimizer های سریعتر**

- ❑ **Momentum optimization**
- ❑ **Nesterov**
- ❑ **AdaGrad**
- ❑ **RMSProp**
- ❑ **Adam and Nadam**
- ❑ **Learning rate scheduling**

چالش های آموزش شبکه های عصبی عمیق

Time → **Optimizer های سریعتر**

❑Stochastic Gradient Descent (SGD)

Require: Learning rate schedule $\epsilon_1, \epsilon_2, \ldots$

Require: Initial parameter θ

k ← 1

while not stoping criteria:

Sample a minibatch of size m from the training set $\{x^{(1)}, \ldots, x^{(m)}\}$ with corresponding targets $y^{(i)}$

Compute gradient estimate: $\hat{g} \leftarrow \frac{1}{m} \nabla_\theta \Sigma_i L\left(f\left(x^{(i)}; \theta\right), y^{(i)}\right)$

update: $\theta \leftarrow \theta - \epsilon_k \hat{g}$

k ← k + 1

Goodfellow et al, Deep learning

چالش های آموزش شبکه های عصبی عمیق

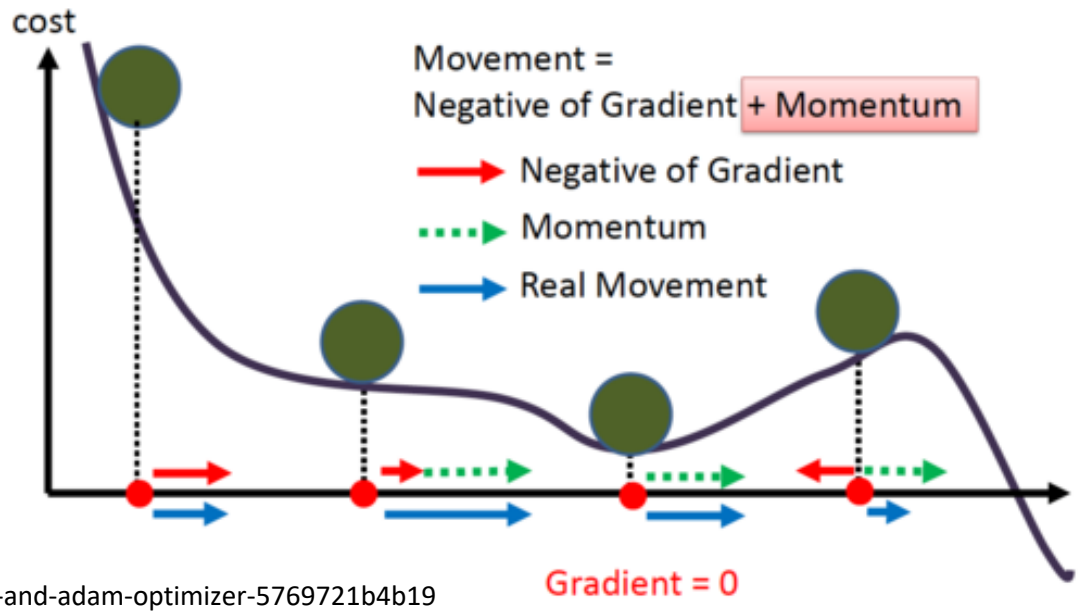
Time → **Optimizer های سریعتر**

❑ Momentum optimization

Some methods of speeding up the convergence of iteration methods ☆

B.T. Polyak

https://doi.org/10.1016/0041-5553(64)90137-5



https://medium.com/analytics-vidhya/momentum-rmsprop-and-adam-optimizer-5769721b4b19

چالش های آموزش شبکه های عصبی عمیق

❑Momentum

Require: Learning rate schedule ϵ, momentum parameter α
Require: Initial parameter θ, initial velocity v

while not stoping criteria:

Sample a minibatch of size m from the training set $\{x^{(1)}, \dots, x^{(m)}\}$ with corresponding targets $y^{(i)}$

Compute gradient estimate: $\hat{g} \leftarrow \frac{1}{m} \nabla_\theta \Sigma_i L\left(f\left(x^{(i)}; \theta\right), y^{(i)}\right)$

Compute velocity update: $v \leftarrow \alpha v - \epsilon \hat{g}$

update: $\theta \leftarrow \theta + v$

Goodfellow et al, Deep learning

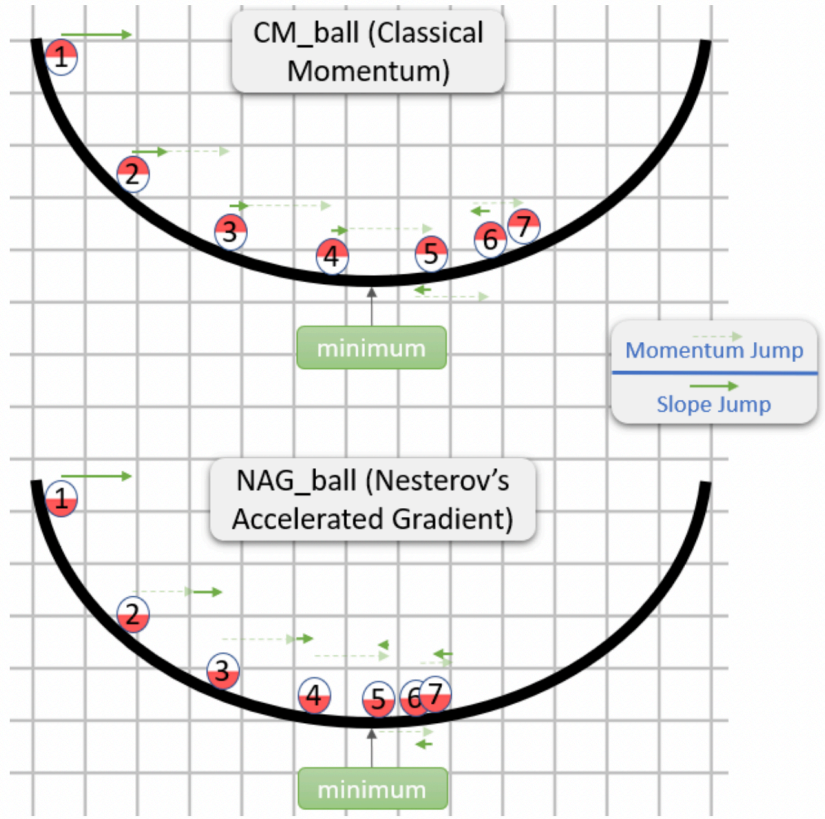
چالش های آموزش شبکه های عصبی عمیق

Time → **Optimizer های سریعتر**

❑ Nesterov

A method for unconstrained convex minimization problem with the rate of convergence o(1/k^2)

Yurii Nesterov



Doklady AN USSR 269 (1983)

https://math.stackexchange.com/questions/1145155/intuition-for-gradient-descent-with-nesterov-momentum

چالش های آموزش شبکه های عصبی عمیق

❑Nesterov Momentum

Require: Learning rate schedule ϵ, momentum parameter α
Require: Initial parameter θ, initial velocity v

while not stoping criteria:

Sample a minibatch of size m from the training set $\{x^{(1)}, \dots, x^{(m)}\}$ with corresponding targets $y^{(i)}$

Apply interim update: $\tilde{\theta} \leftarrow \theta + \alpha v$

Compute gradient estimate: $g \leftarrow \frac{1}{m} \nabla_{\tilde{\theta}} \Sigma_i L\left(f\left(x^{(i)}; \tilde{\theta}\right), y^{(i)}\right)$

Compute velocity update: $v \leftarrow \alpha v - \epsilon g$

update: $\theta \leftarrow \theta + v$

Goodfellow et al, Deep learning

چالش های آموزش شبکه های عصبی عمیق

Time → **Optimizer های سریعتر**

❑ AdaGrad

Adaptive Subgradient Methods for Online Learning and Stochastic Optimization

Authors: John Duchi, Elad Hazan, Yoram Singer Authors Info & Claims

The Journal of Machine Learning Research, Volume 12 • 2/1/2011 • pp 2121–2159

چالش های آموزش شبکه های عصبی عمیق

❑AdaGrad

Require: Global learning rate ϵ, Initial parameter θ, Small constant δ (for numerical instability = 1e-7)

Initialize gradient accumulation variable r=0

while not stoping criteria:

Sample a minibatch of size m from the training set $\{x^{(1)}, \dots, x^{(m)}\}$ with corresponding targets $y^{(i)}$

Compute gradient estimate: $g \leftarrow \frac{1}{m} \nabla_\theta \Sigma_i L\left(f\left(x^{(i)}; \theta\right), y^{(i)}\right)$

Accumulate squared gradient : $r \leftarrow \mathbf{r} + g \odot g$

Compute update: $\Delta\theta \leftarrow -\frac{\epsilon}{\delta + \sqrt{r}} \odot g$

update: $\theta \leftarrow \theta + \Delta\theta$

Goodfellow et al, Deep learning

چالش های آموزش شبکه های عصبی عمیق

Time → **Optimizer های سریعتر**

❑ RMSProp

Neural Networks for Machine Learning

Lecture 6a
Overview of mini-batch gradient descent

Geoffrey Hinton
with
Nitish Srivastava
Kevin Swersky

چالش های آموزش شبکه های عصبی عمیق

❑RMSProp

Require: Global learning rate ϵ, decay rate ρ

Require: Initial parameter θ, Small constant δ (1e-6)

Initialize accumulation variable r=0

while not stoping criteria:

Sample a minibatch of size m from the training set $\{x^{(1)}, \dots, x^{(m)}\}$ with corresponding targets $y^{(i)}$

Compute gradient estimate: $g \leftarrow \frac{1}{m} \nabla_\theta \Sigma_i L\left(f\left(x^{(i)}; \theta\right), y^{(i)}\right)$

Accumulate squared gradient : $r \leftarrow \rho r + (1-\rho) g \odot g$

Compute update: $\Delta\theta = -\frac{\epsilon}{\sqrt{\delta + r}} \odot g$

update: $\theta \leftarrow \theta + \Delta\theta$

Goodfellow et al, Deep learning

چالش های آموزش شبکه های عصبی عمیق

❑RMSProp with Nesterov

Require: Global learning rate ϵ, decay rate ρ, momentum coefficient α

Require: Initial parameter θ, initial velocity v

Initialize accumulation variable r=0

while not stoping criteria:

Sample a minibatch of size m from the training set $\{x^{(1)}, \dots, x^{(m)}\}$ with corresponding targets $y^{(i)}$

Compute interim update: $\tilde{\theta} \leftarrow \theta + \alpha v$

Compute gradient estimate: $g \leftarrow \frac{1}{m} \nabla_{\tilde{\theta}} \Sigma_i L\left(f\left(x^{(i)}; \tilde{\theta}\right), y^{(i)}\right)$

Accumulate squared gradient: $r \leftarrow \rho r + (1-\rho) g \odot g$

Compute velocity update: $v \leftarrow \alpha v - \frac{\epsilon}{\sqrt{r}} \odot g$

update: $\theta \leftarrow \theta + v$

Goodfellow et al, Deep learning

چالش های آموزش شبکه های عصبی عمیق

Time → **Optimizer های سریعتر**

- **Adam**

Adam: A Method for Stochastic Optimization

Diederik P. Kingma, Jimmy Ba

arXiv:1412.6980

چالش های آموزش شبکه های عصبی عمیق

❑Adam

Require: Step size ϵ(suggested default:0.001), Exponential decay rates for moment estimates, ρ_1 and ρ_2 in [0, 1) (suggested defaults: 0.9 and 0.999), Small constant δ (1e-8)

Require: Initial parameters θ

Initialize first and second moment variables s=0 and r = 0 and time step t = 0

while not stoping criteria:

Sample a minibatch of size m from the training set $\{x^{(1)}, \dots, x^{(m)}\}$ with corresponding targets $y^{(i)}$

Compute gradient estimate: $g \leftarrow \frac{1}{m} \nabla_\theta \Sigma_i L\left(f\left(x^{(i)}; \theta\right), y^{(i)}\right)$

$t \leftarrow t + 1$

Update biased first moment estimate: $s \leftarrow \rho_1 s + (1 - \rho_1) g$

Update biased second moment estimate: $\mathrm{r} \leftarrow \rho_2 r + (1 - \rho_2) g \odot g$

Correct bias in first and second moment: $\hat{s} \leftarrow \frac{s}{1-\rho_1^t}$, $\hat{r} \leftarrow \frac{r}{1-\rho_2^t}$

Compute update: $\Delta\theta = -\epsilon \frac{\hat{s}}{\sqrt{\hat{r}} + \delta}$

update: $\theta \leftarrow \theta + \Delta\theta$

Goodfellow et al, Deep learning

چالش های آموزش شبکه های عصبی عمیق

Time ⟶ **Optimizer های سریعتر**

❑ Adam -> AdaMax

while not stoping criteria:

Sample a minibatch of size m from the training set $\{x^{(1)}, \dots, x^{(m)}\}$ with corresponding targets $y^{(i)}$

Compute gradient estimate: $g \leftarrow \frac{1}{m} \nabla_\theta \Sigma_i L\left(f\left(x^{(i)}; \theta\right), y^{(i)}\right)$

$t \leftarrow t + 1$

Update biased first moment estimate: $s \leftarrow \rho_1 s + (1 - \rho_1) g$

Update the exponentially weighted infinity norm : $\mathrm{r} \leftarrow max(\rho_2 r, |g|)$

Compute update: $\Delta\theta = -\frac{\epsilon}{1-\rho_1^t} \frac{s}{r}$

update: $\theta \leftarrow \theta + \Delta\theta$

چالش های آموزش شبکه های عصبی عمیق

Time → **Optimizer های سریعتر**

❑ Adam -> Nadam

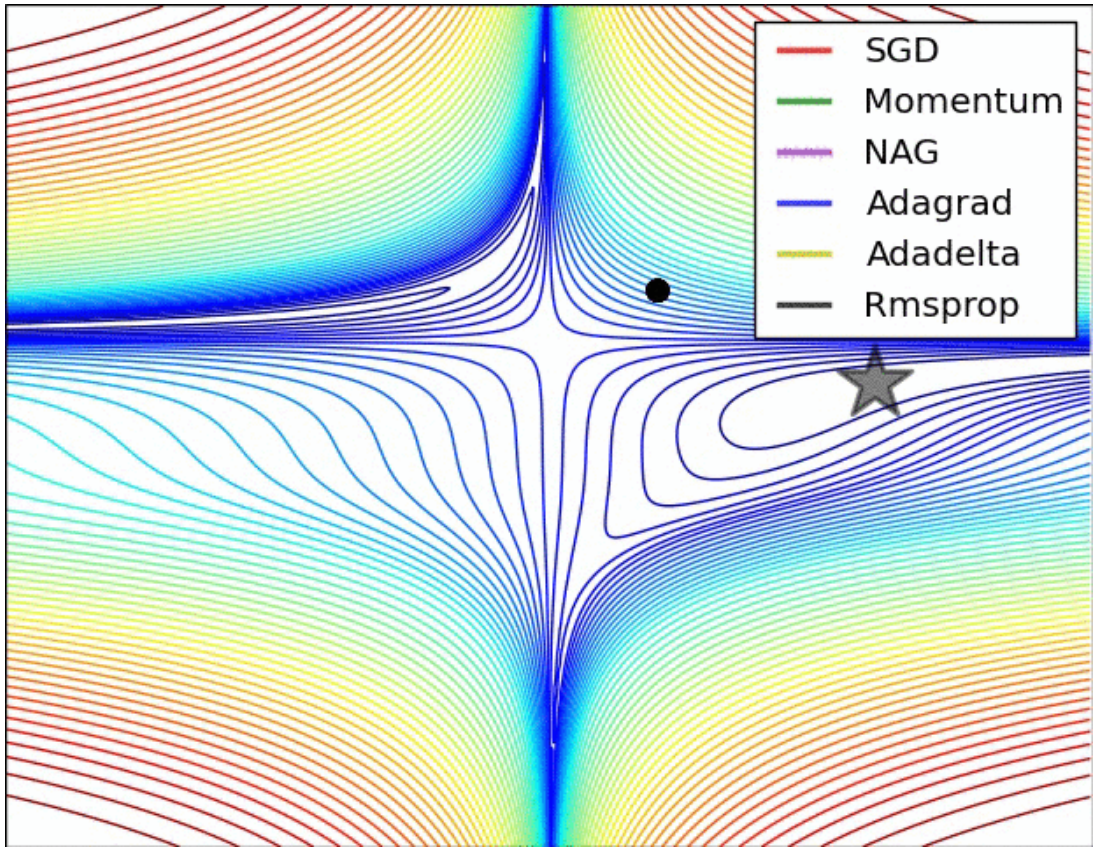
Workshop track - ICLR 2016

INCORPORATING NESTEROV MOMENTUM INTO ADAM

Timothy Dozat

چالش های آموزش شبکه های عصبی عمیق

Time → **Optimizer های سریعتر**



https://towardsdatascience.com/why-visualize-gradient-descent-optimization-algorithms-a393806eee2

چالش های آموزش شبکه های عصبی عمیق

Time → **Optimizer های سریعتر**

Optimizer	سرعت همگرایی	کیفیت همگرایی
SGD	1	3
SGD(momentum)	2	3
SGD(momentum, nesterov)	2	3
Adagrad	3	1
RMSprop	3	2-3
Adam	3	2-3
Nadam	3	2-3
AdaMax	3	2-3

چالش های آموزش شبکه های عصبی عمیق

Time → **برنامه های زمانی نرخ یادگیری**

❑ Learning Rate

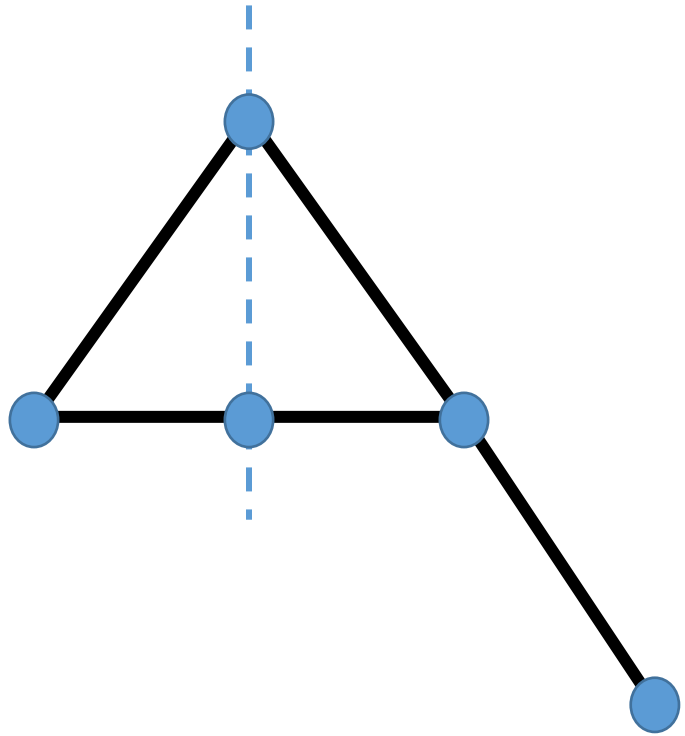
- ❑ **Power scheduling** $\eta(t) = \eta_0/(1 + t/s)^c$
- ❑ **Exponential scheduling** $\eta(t) = \eta_0 0.1^{t/s}$
- ❑ **Piecewise constant scheduling**
- ❑ **Performance scheduling**
- ❑ **1cycle scheduling** N. Smith, 2018 arXiv:1801.09821

چالش های آموزش شبکه های عصبی عمیق

Time → **برنامه های زمانی نرخ یادگیری**

❑ Learning Rate

- ❑ 1cycle scheduling



چالش های آموزش شبکه های عصبی عمیق

Overfitting → **Regularization**

- ❑ **l1 and l2 regularization**
- ❑ **Dropout**
- ❑ **Max-Norm**

چالش های آموزش شبکه های عصبی عمیق

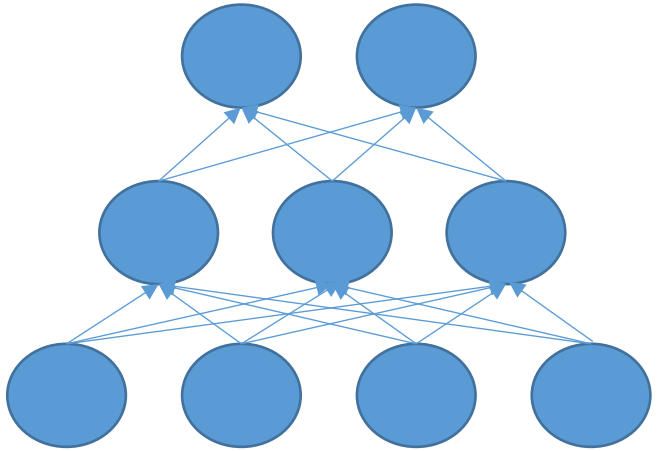
Overfitting → **Regularization**

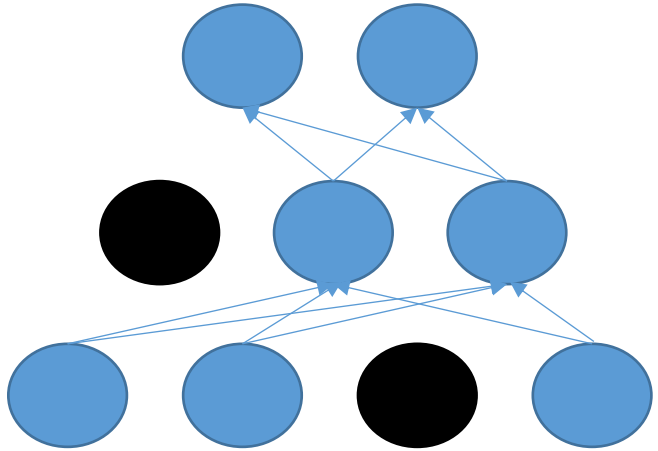
❑ Dropout

JMLR

Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov; 15(56):1929–1958, 2014.





چالش های آموزش شبکه های عصبی عمیق

Overfitting → **Regularization**

❑ Dropout -> Monte Carlo

Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning

Yarin Gal, Zoubin Ghahramani

arXiv:1506.02142

چالش های آموزش شبکه های عصبی عمیق

Overfitting → **Regularization**

- **Max-Norm**

$$\boldsymbol{w} \leftarrow \boldsymbol{w} * \boldsymbol{c} / \| \boldsymbol{w} \|_2$$

چالش های آموزش شبکه های عصبی عمیق

Default Config

Regular Network

Hyperparameter	Default
Kernel initializer	He initialization
Activation function	ELU
Normalization	Batch (None for shallow)
Regularization	Early stopping (l2)
Optimizer	Momentum (RMSProp or Nadam)
Learning rate schedule	1cycle

Self-normalizing Network

Hyperparameter	Default
Kernel initializer	LeCun
Activation function	SELU
Normalization	None
Regularization	Alpha dropout
Optimizer	Momentum (RMSProp or Nadam)
Learning rate schedule	1cycle

Aurelien Geron, Hands-on machine learning with scikit-learn, keras and tensorflow. Orielly, 2019