



IIT KHARAGPUR
IIT GANDHINAGAR



NPTEL ONLINE
CERTIFICATION COURSES

Scalable Data Science

Lecture 21: Stochastic Optimization

Sourangshu Bhattacharya
Computer Science and Engineering
IIT KHARAGPUR

In the previous lectures:

- Outline:
 - Big Data platforms.
 - Distributed Machine Learning.



IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

Sourangshu Bhattacharya
Computer Science and Engg.

In this Lecture:

- Outline:
 - Stochastic gradient descent (SGD)
 - SGD Convergence
 - Minibatch and Distributed SGD
 - Practical considerations
 - Advancements from SGD.



IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

Sourangshu Bhattacharya
Computer Science and Engg.

Much of ML is optimization

Linear Classification

$$\arg \min_w \sum_{i=1}^n ||w||^2 + C \sum_{i=1}^n \xi_i$$

$$\text{s.t. } 1 - y_i x_i^T w \leq \xi_i$$

$$\xi_i \geq 0$$

Maximum Likelihood

$$\arg \max_{\theta} \sum_{i=1}^n \log p_{\theta}(x_i)$$

K-Means

$$\arg \min_{\mu_1, \mu_2, \dots, \mu_k} J(\mu) = \sum_{j=1}^k \sum_{i \in C_j} ||x_i - \mu_j||^2$$



IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

Stochastic optimization

- Goal of machine learning :
 - Minimize expected loss

$$\min_h L(h) = \mathbf{E} [\text{loss}(h(x), y)]$$

given samples (x_i, y_i) $i = 1, 2 \dots m$

- This is Stochastic Optimization
 - Assume loss function is convex



IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

Batch (sub)gradient descent for ML

- Process all examples together in each step

$$w^{(k+1)} \leftarrow w^{(k)} - \eta_t \left(\frac{1}{n} \sum_{i=1}^n \frac{\partial L(w, x_i, y_i)}{\partial w} \right)$$

where L is the regularized loss function

- Entire training set examined at each step
- Very slow when n is very large

Stochastic (sub)gradient descent

- “Optimize” one example at a time
- Choose examples randomly (or reorder and choose in order)
 - Learning representative of example distribution

for $i = 1$ to n :

$$w^{(k+1)} \leftarrow w^{(k)} - \eta_t \frac{\partial L(w, x_i, y_i)}{\partial w}$$

where L is the regularized loss function



IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

Stochastic (sub)gradient descent

for $i = 1$ to n :

$$w^{(k+1)} \leftarrow w^{(k)} - \eta_t \frac{\partial L(w, x_i, y_i)}{\partial w}$$

where L is the regularized loss function

- Equivalent to online learning (the weight vector w changes with every example)
- Convergence guaranteed for convex functions (to local minimum)



IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

Stochastic gradient descent

- Given dataset $D = \{(x_1, y_1), \dots, (x_m, y_m)\}$
- Loss function: $L(\theta, D) = \frac{1}{N} \sum_{i=1}^N l(\theta; x_i, y_i)$
- For linear models: $l(\theta; x_i, y_i) = l(y_i, \theta^T \phi(x_i))$
- Assumption D is drawn IID from some distribution \mathcal{P} .
- Problem:

$$\min_{\theta} L(\theta, D)$$



IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

Stochastic gradient descent

- Input: D
- Output: $\bar{\theta}$

Algorithm:

- Initialize θ^0
- For $t = 1, \dots, T$
$$\theta^{t+1} = \theta^t - \eta_t \nabla_{\theta} l(y_t, \theta^T \phi(x_t))$$
- $\bar{\theta} = \frac{\sum_{t=1}^T \eta_t \theta^t}{\sum_{t=1}^T \eta_t}$.



IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

SGD convergence

- Expected loss: $s(\theta) = E_{\mathcal{P}}[l(y, \theta^T \phi(x))]$
- Optimal Expected loss: $s^* = s(\theta^*) = \min_{\theta} s(\theta)$
- Convergence:

$$E_{\bar{\theta}}[s(\bar{\theta})] - s^* \leq \frac{R^2 + L^2 \sum_{t=1}^T \eta_t^2}{2 \sum_{t=1}^T \eta_t}$$

- Where: $R = \|\theta^0 - \theta^*\|$
- $L = \max \nabla l(y, \theta^T \phi(x))$



IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

SGD convergence proof

- Define $r_t = \|\theta^t - \theta^*\|$ and $g_t = \nabla_{\theta} l(y_t, \theta^T \phi(x_t))$
- $r_{t+1}^2 = r_t^2 + \eta_t^2 \|g_t\|^2 - 2\eta_t(\theta^t - \theta^*)^T g_t$
- Taking expectation w.r.t $\mathcal{P}, \bar{\theta}$ and using $s^* - s(\theta^t) \geq g_t^T (\theta^* - \theta^t)$, we get:

$$E_{\bar{\theta}}[r_{t+1}^2 - r_t^2] \leq \eta_t^2 L^2 + 2\eta_t(s^* - E_{\bar{\theta}}[s(\theta^t)])$$

- Taking sum over $t = 1, \dots, T$ and using

$$E_{\bar{\theta}}[r_{t+1}^2 - r_0^2] \leq L^2 \sum_{t=0}^{T-1} \eta_t^2 + 2 \sum_{t=0}^{T-1} \eta_t (s^* - E_{\bar{\theta}}[s(\theta^t)])$$



IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

SGD convergence proof

- Using convexity of s :

$$\left(\sum_{t=0}^{T-1} \eta_t \right) E_{\bar{\theta}} [s(\bar{\theta})] \leq E_{\bar{\theta}} \left[\sum_{t=0}^{T-1} \eta_t s(\theta^t) \right]$$

- Substituting in the expression from previous slide:

$$E_{\bar{\theta}} [r_{t+1}^2 - r_0^2] \leq L^2 \sum_{t=0}^{T-1} \eta_t^2 + 2 \sum_{t=0}^{T-1} \eta_t (s^* - E_{\bar{\theta}} [s(\bar{\theta})])$$

- Rearranging the terms proves the result.



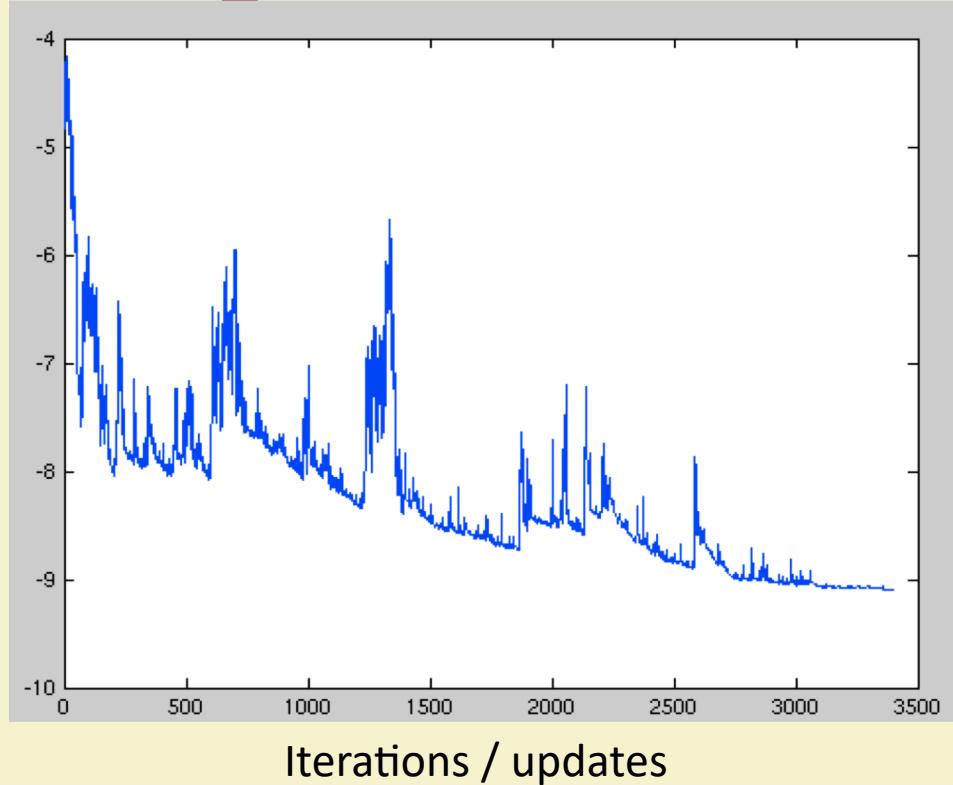
IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

SGD convergence

Objective function value



IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

SGD - Issues

- Convergence very sensitive to learning rate (η_t) (oscillations near solution due to probabilistic nature of sampling)
 - Might need to decrease with time to ensure the algorithm converges eventually
- Basically – SGD good for machine learning with large data sets!



IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

Mini-batch SGD

- Stochastic – 1 example per iteration
- Batch – All the examples!
- Mini-batch SGD:
 - Sample m examples at each step and perform SGD on them
- Allows for parallelization, but choice of m based on heuristics



IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

Example: Text categorization

- Example by Leon Bottou:

- Reuters RCV1 document corpus
 - Predict a category of a document
 - One **vs.** the rest classification
- $n = 781,000$ training examples (documents)
- 23,000 test examples
- $d = 50,000$ features
 - One feature per word
 - Remove stop-words
 - Remove low frequency words



IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

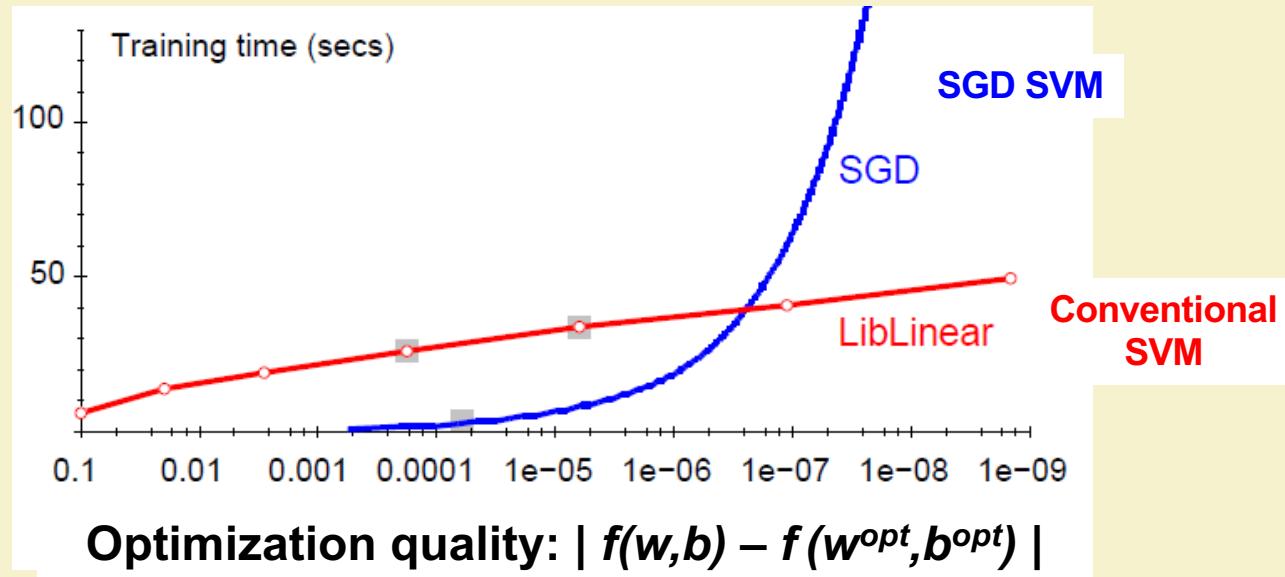
Example: Text categorization

- **Questions:**
 - (1) Is SGD successful at minimizing $f(w,b)$?
 - (2) How quickly does SGD find the min of $f(w,b)$?
 - (3) What is the error on a test set?

	<i>Training time</i>	<i>Value of $f(w,b)$</i>	<i>Test error</i>
Standard SVM	23,642 secs	0.2275	6.02%
“Fast SVM”	66 secs	0.2278	6.03%
SGD SVM	1.4 secs	0.2275	6.02%

- (1) SGD-SVM is successful at minimizing the value of $f(w,b)$
- (2) SGD-SVM is super fast
- (3) SGD-SVM test set error is comparable

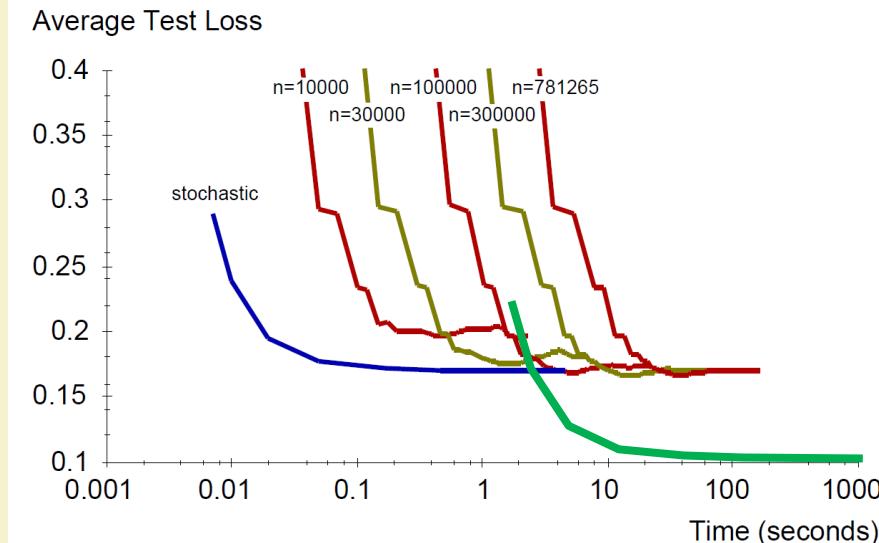
Optimization “Accuracy”



For optimizing $f(w,b)$ *within reasonable* quality
SGD-SVM is super fast

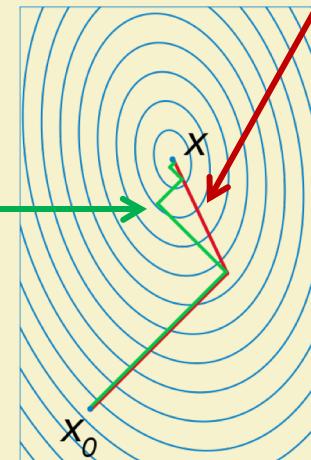
SGD vs. Batch Conjugate Gradient

- SGD on full dataset vs. Conjugate Gradient on a sample of n training examples



Bottom line: Doing a simple (but fast) SGD update many times is better than doing a complicated (but slow) CG update a few times

Theory says: Gradient descent converges in linear time k . Conjugate gradient converges in \sqrt{k} .
 k ... condition number



Practical Considerations

- Need to choose learning rate η and t_0

$$w_{t+1} \leftarrow w_t - \frac{\eta_t}{t + t_0} \left(w_t + C \frac{\partial L(x_i, y_i)}{\partial w} \right)$$

- Leon suggests:

- Choose t_0 so that the expected initial updates are comparable with the expected size of the weights
- Choose η :
 - Select a small subsample
 - Try various rates η (e.g., 10, 1, 0.1, 0.01, ...)
 - Pick the one that most reduces the cost
 - Use η for next 100k iterations on the full dataset



IIT KHARAGPUR



NPTEL ONLINE
CERTIFICATION COURSES

Practical Considerations

- Sparse Linear SVM:
 - Feature vector x_i is sparse (contains many zeros)
 - Do not do: $x_i = [0, 0, 0, 1, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, \dots]$
 - But represent x_i as a sparse vector $x_i = [(4, 1), (9, 5), \dots]$
 - Can we do the SGD update more efficiently?

$$w \leftarrow w - \eta \left(w + C \frac{\partial L(x_i, y_i)}{\partial w} \right)$$

- Approximated in 2 steps:

$$w \leftarrow w - \eta C \frac{\partial L(x_i, y_i)}{\partial w}$$

cheap: x_i is sparse and so few
coordinates j of w will be updated

$$w \leftarrow w(1 - \eta)$$

expensive: w is not sparse, all
coordinates need to be updated



IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

Practical Considerations

■ Solution 1: $w = s \cdot v$

- Represent vector w as the product of scalar s and vector v
- Then the update procedure is:
 - (1) $v = v - \eta C \frac{\partial L(x_i, y_i)}{\partial w}$
 - (2) $s = s(1 - \eta)$

• Solution 2:

- Perform only step (1) for each training example
- Perform step (2) with lower frequency and higher η

Two step update procedure:

$$(1) w \leftarrow w - \eta C \frac{\partial L(x_i, y_i)}{\partial w}$$

$$(2) w \leftarrow w(1 - \eta)$$



IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

Practical Considerations

- **Stopping criteria:**

How many iterations of SGD?

- **Early stopping with cross validation**

- Create a validation set
 - Monitor cost function on the validation set
 - Stop when loss stops decreasing

- **Early stopping**

- Extract two disjoint subsamples **A** and **B** of training data
 - Train on **A**, stop by validating on **B**
 - Number of epochs is an estimate of **k**
 - Train for **k** epochs on the full dataset



IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

Stochastic gradient descent

- Idea: Perform a parameter update for each training example $x(i)$ and label $y(i)$
- Update: $\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x(i), y(i))$
- Performs redundant computations for large datasets



IIT KHARAGPUR



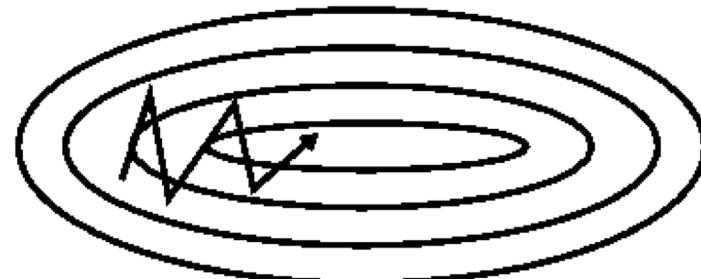
NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

Momentum gradient descent

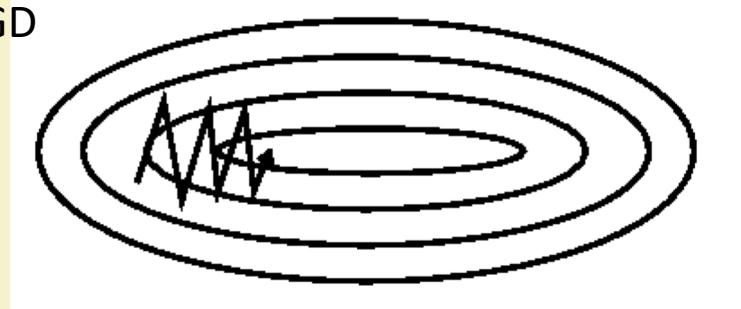
- Idea: Overcome ravine oscillations by momentum
- Update:

$$\begin{aligned} \bullet \quad & v_t = \gamma v_{t-1} + \eta \cdot \nabla_{\theta} J(\theta) \\ \bullet \quad & \theta = \theta - v_t \end{aligned}$$

SGD with
momentum



SGD



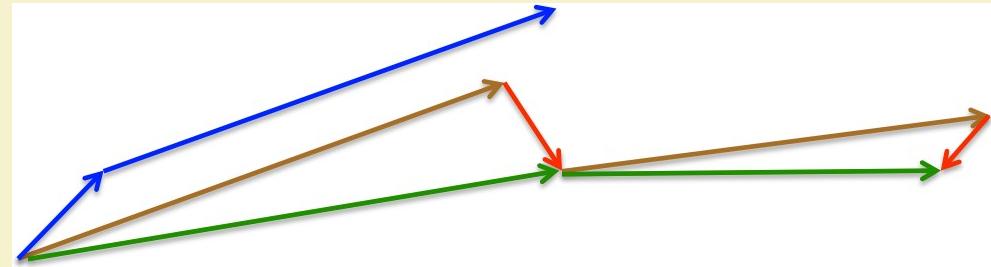
IIT KHARAGPUR



NPTEL ONLINE
CERTIFICATION COURSES

Nesterov accelerated gradient

- Ideas:
 1. Big jump in the direction of the previous accumulated gradient & measure the gradient
 2. Then make a correction.
- Update:
 - $v_t = \gamma v_{t-1} + \eta \cdot \nabla_{\theta} J(\theta - \gamma v_{t-1})$
 - $\theta = \theta - v_t$



RMSprop

- Idea: Use the second moment of gradient vector to estimate the magnitude of update in a given direction.
- Update:
 - $E[g^2]_t = 0.9 E[g^2]_{t-1} + 0.1 g_t^2$
 - $\Delta\theta_t = -\eta / \sqrt{E[g^2]_t + \epsilon} \odot g_t$



IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

ADAM (Adaptive moment)

- Idea: In addition to storing an exponentially decaying average of past squared gradients like RMSprop, Adam also keeps an exponentially decaying average of past gradients.
- Updates:
 - $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$
 - $v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$
 - $\hat{m}_t = m_t / (1 - \beta_1^t)$
 - $\hat{v}_t = v_t / (1 - \beta_2^t)$
 - $\vartheta_{t+1} = \vartheta_t - (\eta / (\sqrt{\hat{v}_t} + \epsilon)) \hat{m}_t$

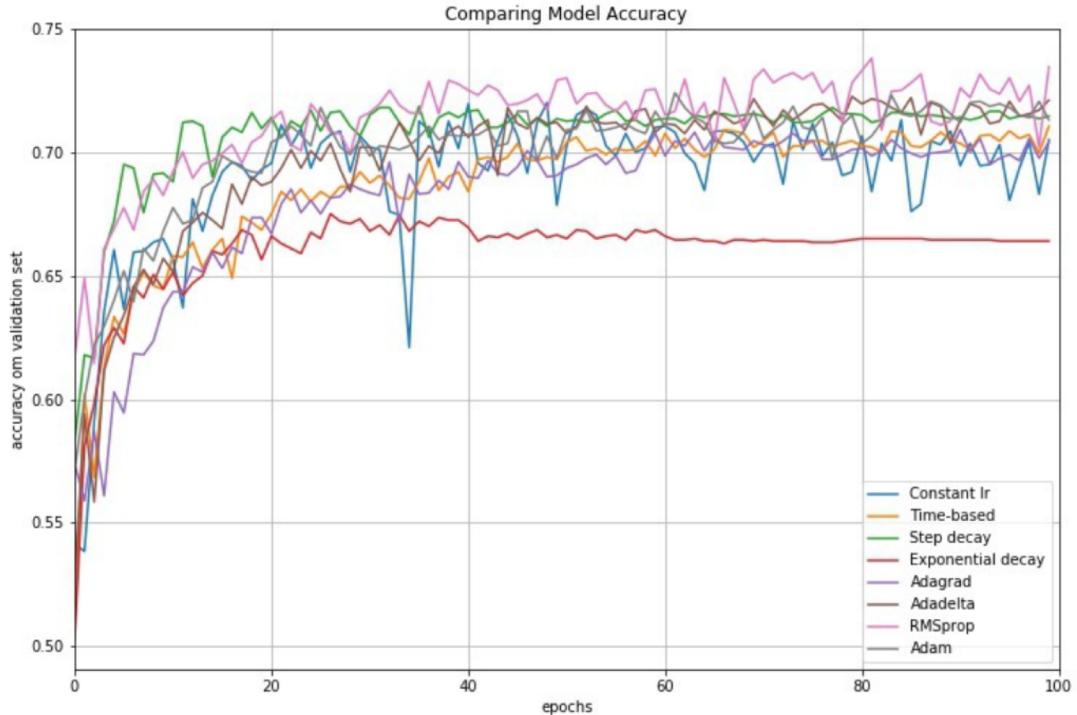


IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

Enhancements comparison



IIT KHARAGPUR



NPTEL
ONLINE
CERTIFICATION COURSES

Conclusion:

- We have seen:
 - Spark: Programming examples:
 - Alternating least squares
 - User log mining
 - Partitioning
 - Accumulators
 - Scheduling of tasks



IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

Sourangshu Bhattacharya
Computer Science and Engg.

References:

- SGD proof by Yuri Nesterov.
- MMDS <http://www.mmds.org/>
- *Blog of Sebastian Ruder* <http://ruder.io/optimizing-gradient-descent/>
- *Learning rate comparison* <https://towardsdatascience.com/learning-rate-schedules-and-adaptive-learning-rate-methods-for-deep-learning-2c8f433990d1>



IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

Sourangshu Bhattacharya
Computer Science and Engg.

Thank You!!



IIT KHARAGPUR



NPTEL
NPTEL ONLINE
CERTIFICATION COURSES

Sourangshu Bhattacharya
Computer Science and Engg.