Convergence Analysis of Optimizers

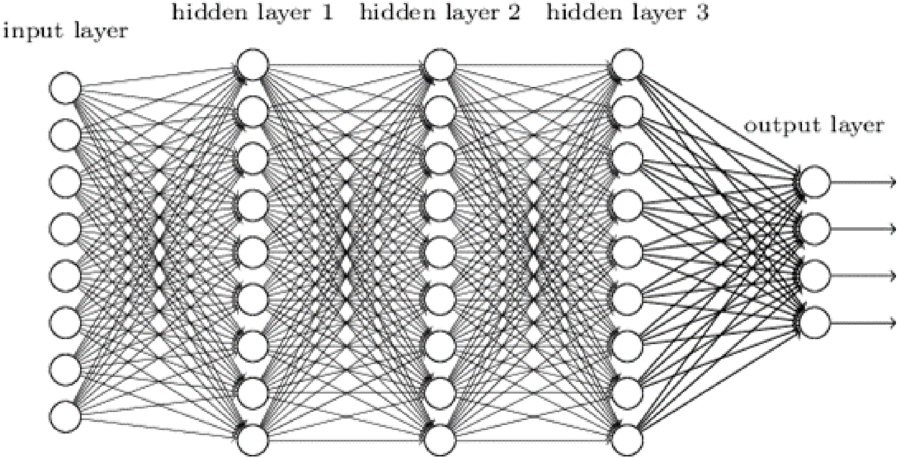
Anthony Garcia

Contents (Contribution)

- 1. Deep Learning Optimization Overview
- 2. Mathematical Approach - Unified Perspective for Optimizers
- 3. Virtual Experiment - Polynomial Example
- 4. Empirical Evaluation - Broad Range of Controlled Experiments
- 5. Conclusion

1. Deep Learning Optimization Overview

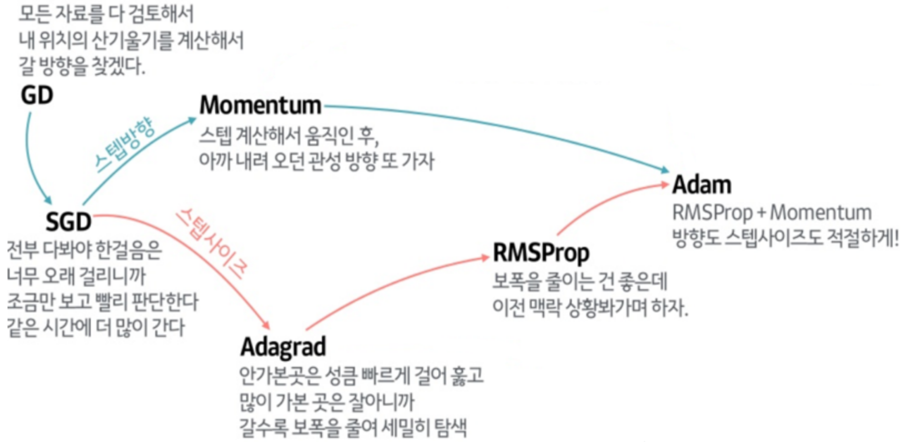
- The optimization problem in deep learning is non-convex.
- Through initialization, (dynamic) learning rate, and stochastic elements, we can make gradient descent-based methods converge to the global minimum.



2. Unified Perspective for Optimizers (GD)

$$\mathbf{x_{t+1}} = \mathbf{x_t} - \alpha \nabla \mathbf{f}(\mathbf{x})$$

- Step size (learning rate) : α
- Step direction : $\nabla f(x)$



2. Unified Perspective for Optimizers (SGD)

Stochastic Gradient Descent (Mini-batch)

- $\mathbf{g_t = \nabla \tilde{f}_t(x) = \frac{1}{t}\sum_{i=1}^{t} \nabla f_i(x_t)}$
- $\mathbf{x_{t+1} = x_t - \alpha g_t}$

Mini-batch: A training method dividing the entire dataset into small subsets

- Noise Injection $\rightarrow$ escape local minima, find global minima
- Stochastic Learning
- Computational Efficiency
- Memory Efficiency

2. Unified Perspective for Optimizers (SGDM)

Stochastic gradient descent with momentum

- $\mathbf{g_t} = \frac{\mathbf{1}}{\mathbf{t}} \sum_{\mathbf{i=1}}^{\mathbf{t}} \nabla \mathbf{f_i}(\mathbf{x_t})$
- $\mathbf{m_t} = \beta \mathbf{m_{t-1}} + (\mathbf{1} - \beta)\mathbf{g_t}$
- $\mathbf{x_{t+1}} = \mathbf{x_t} - \alpha \mathbf{m_t}$

Momentum: Inertia from previous gradients + current gradient

- $m_t = (0.081 g_{t-2} + 0.09 g_{t-1}) + 0.1 g_t$, $(\beta = 0.9)$
- SGDM is SGD with inertia (SGD is special case of SGDM in $\beta = 1$)
- Reduction of oscillations

2. Unified Perspective for Optimizers (Adagrad)

Adagrad : dynamic learning rate based on the accumulated square sum of gradients

- $\mathbf{g_i^{(k)} = \frac{1}{t}\sum_{j=1}^{k} \nabla f_i(x_j)}$
- $\mathbf{v_i^{(k)} = \sum_{j=1}^{k} (g_i^{(j)})^2}$
- $\mathbf{x_i^{(k+1)} = x_i^{(k)}} - \frac{\alpha}{\sigma + \sqrt{\mathbf{v_i^{(k)}}}}\mathbf{g_i^{(k)}}$

  (σ is a small value to prevent division by zero)

Dynamic learning rate: less sensitive to learning rate

- Adagrad is SGD with dynamic learning rate
  (SGD is special case of Adagrad in $v_i^{(k)} = 0$)
- large step size initially, small step size near optimum

2. Unified Perspective for Optimizers (RMSProp)

RMSProp : exponentially weighted moving average

- $\mathbf{g_i^{(k)}} = \mathbf{\frac{1}{t} \sum_{j=1}^{k} \nabla f_i(x_j)}$
- $\mathbf{v_i^{(k+1)}} = \beta_\mathbf{2} \mathbf{v_i^{(k)}} + (\mathbf{1} - \beta_\mathbf{2})(\mathbf{g_i^{(k)}})^\mathbf{2}$
- $\mathbf{\hat{v}_i}^{\mathbf{(k+1)}} = \dfrac{\mathbf{v_i^{(k+1)}}}{\mathbf{1} - \beta_\mathbf{2}^{\mathbf{k+1}}}$
- $\mathbf{x_i^{(k+1)}} = \mathbf{x_i^{(k)}} - \dfrac{\alpha}{\sigma + \sqrt{\mathbf{\hat{v}_i^{(k+1)}}}} \mathbf{g_i^{(k)}} = \mathbf{x_i^{(k)}} - \dfrac{\alpha}{\sigma + \mathrm{RMS}(\mathbf{g_i})} \mathbf{g_i^{(k)}}$

Exponentially weighted moving average: not monotonically decreasing

- to prevent the learning rate from decreasing too rapidly in Adagrad

2. Unified Perspective for Optimizers (Adam)

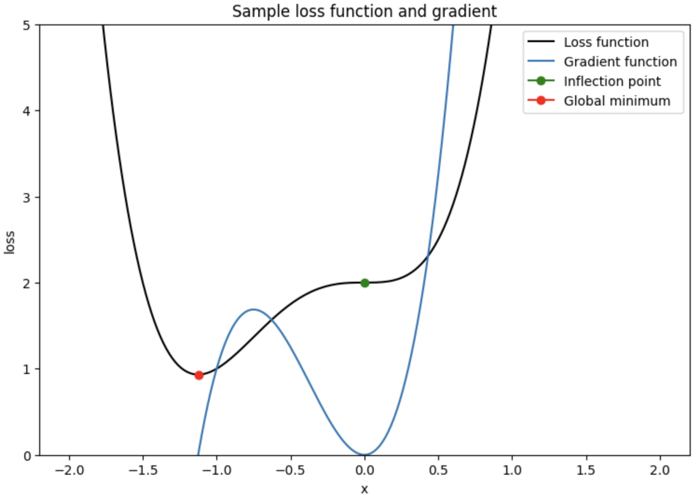
Adam(Momentum + RMSProp)

- $\mathbf{g_i^{(k)} = \frac{1}{t} \sum_{j=1}^{k} \nabla f_i(x_j)}$
- $\mathbf{m^{(k+1)} = \beta_1 m^{(k)} + (1 - \beta_1) g_i^{(k)}}$
- $\mathbf{v_i^{(k+1)} = \beta_2 v_i^{(k)} + (1 - \beta_2)(g_i^{(k)})^2}$
- $\mathbf{\hat{m}^{(k+1)} = \frac{m^{(k+1)}}{1 - \beta_1^{k+1}}}$
- $\mathbf{\hat{v}_i^{(k+1)} = \frac{v_i^{(k+1)}}{1 - \beta_2^{k+1}}}$
- $\mathbf{x_i^{(k+1)} = x_i^{(k)}} - \frac{\alpha}{\sigma + \sqrt{\mathbf{\hat{v}_i^{(k+1)}}}} \mathbf{m^{(k+1)}} = x_i^{(k)} - \frac{\alpha}{\sigma + \mathsf{RMS}(g_i)} \mathbf{m^{(k+1)}}$

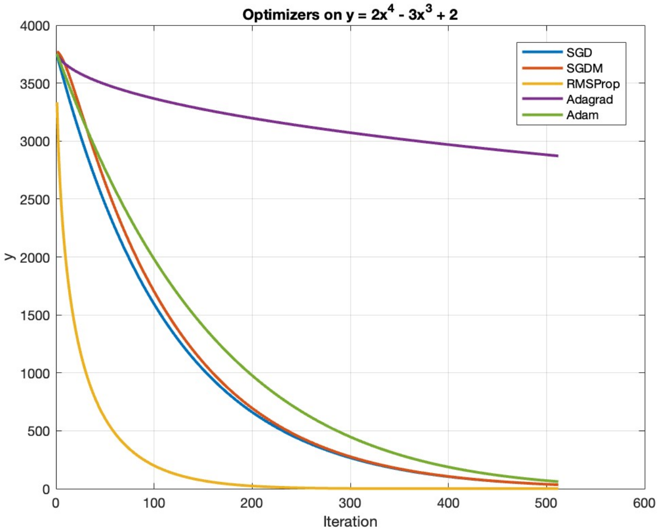
Momentum + RMSProp:

- Adam is the most generalized form among all optimizers (SGDM, Adagrad, RMSProp are special cases of Adam)

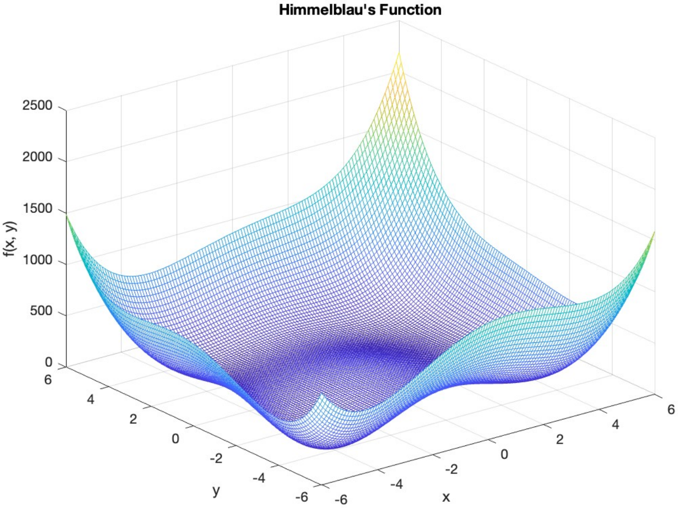
3. Polynomial Example



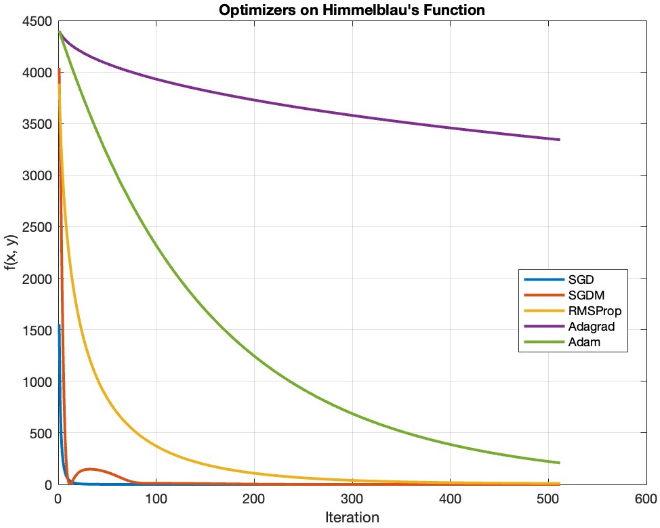
3. Polynomial Example



3. Polynomial Example



3. Polynomial Example

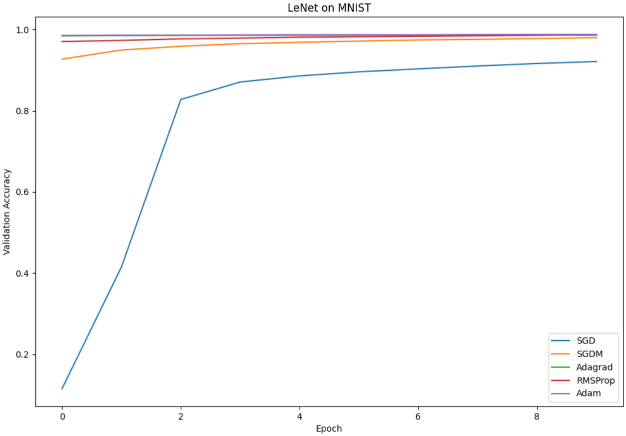


4. Broad Range of Experiments

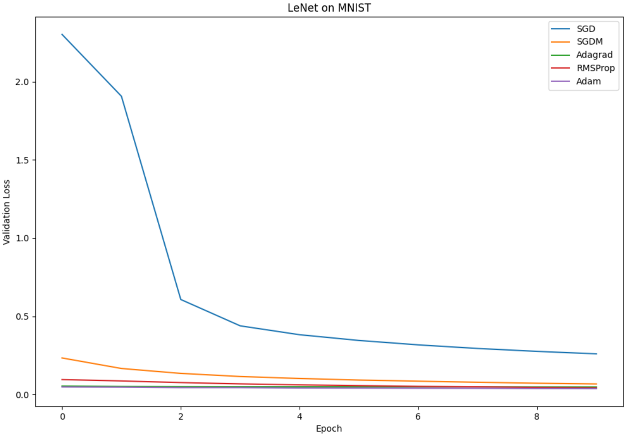
Multiple Datasets, Different Model Architectures, and various optimizer

Dataset	**Model Architecture**
Mnist	LeNet-5
Fashion-mnist	AlexNet
Cifar-10	VGGNet

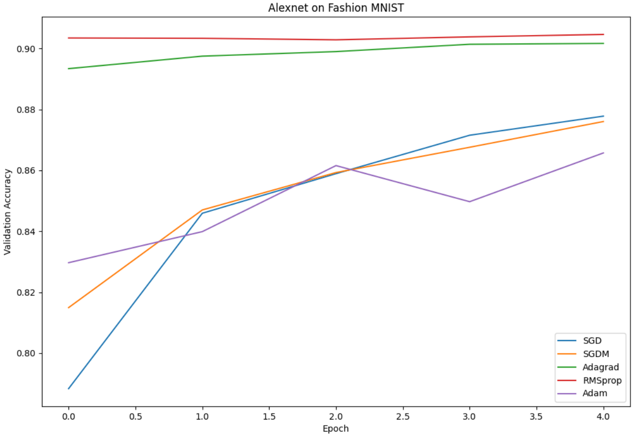
4. Broad Range of Experiments



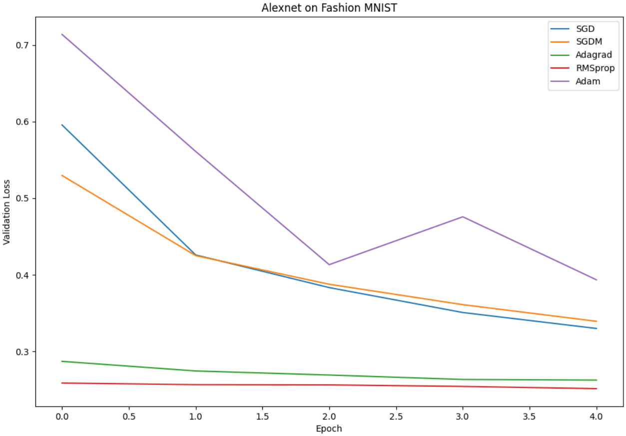
4. Broad Range of Experiments



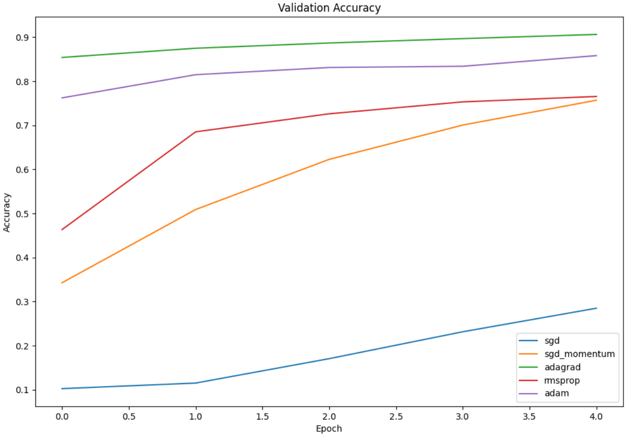
4. Broad Range of Experiments



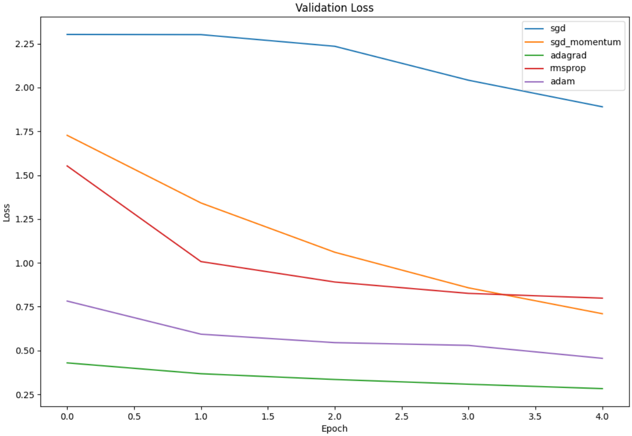
4. Broad Range of Experiments



4. Broad Range of Experiments



4. Broad Range of Experiments



References

- Oh Ilseok, Machine Learning
- Mykel J. Kochenderfer, Algorithms for Optimization
- Yanli Liu, An Improved Analysis of Stochastic Gradient Descent with Momentum(2020)
- John Duchi, Adaptive Subgradient Methods for Online Learning and Stochastic Optimization(2011)
- Alexandre Défossez, A Simple Convergence Proof of Adam and Adagrad(2022)
- Rachel Ward, AdaGrad stepsizes: Sharp convergence over nonconvex landscapes(2020)
- Fangyu Zou, A Sufficient Condition for Convergences of Adam and RMSProp
- Diederik P. Kingma, Adam: A Method for Stochastic Optimization(2015)
- Pierre Foret, Sharpness-Aware Minimization for Efficiently Improving Generalization(2021)